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

MOTOR DE BUSCA PARA DADOS ABERTOS

IAN CARNEIRO TEIXEIRA DE ARAÚJO

Cajazeiras-PB
2021

IAN CARNEIRO TEIXEIRA DE ARAÚJO

MOTOR DE BUSCA PARA DADOS ABERTOS

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

Orientador

Prof. Dr. Fabio Gomes de Andrade.

Cajazeiras-PB
2021

Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Daniel Andrade CRB-15/593

A663m

Araújo, Ian Carneiro Teixeira de

Motor de busca para dados abertos / Ian Carneiro Teixeira de Araújo; orientador Fabio Gomes de Andrade.- 2021.
59 f.: il.

Orientador: Fabio Gomes de Andrade.
TCC (Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2021.

1. Dados abertos governamentais 2. Recuperação da informação 3. Anotação de dados I. Título.

004.6(0.067)

SERVIÇO PÚBLICO FEDERAL
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
COORDENAÇÃO DA UNIDADE ACADÊMICA DA ÁREA DE INFORMÁTICA - CAMPUS CAJAZEIRAS

Às **16:00** horas do dia **04** do mês de **fevereiro** do ano de **2021,** via Google Meet, compareceu para defesa pública do **Trabalho de Conclusão de Curso**, requisito obrigatório para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, o(a) aluno(a) **IAN CARNEIRO TEIXEIRA DE ARAÚJO**, matrícula **201712010016**, tendo como Título do Trabalho **MOTOR DE BUSCA PARA DADOS ABERTOS**. Constituíram a Banca Examinadora os professores **FABIO GOMES DE ANDRADE** (orientador), **FRANCISCO PAULO DE FREITAS NETO** (examinador) e **FRANCISCO DALADIER MARQUES JÚNIOR** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **85,** com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho, via processo eletrônico à coordenação de curso. A versão deve conter a ficha catalográfica e atender às sugestões feitas pelos membros da banca. O código fonte desenvolvido no trabalho (caso haja) deve ser enviado para o e-mail da coordenação do curso (cads.cz@ifpb.edu.br).

Cajazeiras-PB, 5 de fevereiro de 2021.

Documento assinado eletronicamente por:

- **Francisco Daladier Marques Junior, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 02/03/2021 15:17:30.
- **Ian Carneiro Teixeira de Araújo, ALUNO (201712010016) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 01/03/2021 19:43:50.
- **Fabio Gomes de Andrade, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 12/02/2021 12:29:06.
- **Francisco Paulo de Freitas Neto, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 05/02/2021 14:10:22.

Este documento foi emitido pelo SUAP em 04/02/2021. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 154827
Código de Autenticação: 25b3a10f87

AGRADECIMENTOS

A Deus que me proporcionou viver esse momento tão importante e que tem me ajudado em todas as áreas da minha vida.

Aos meus pais, que me apoiaram e incentivaram de várias maneiras durante toda a minha participação no curso de análise e desenvolvimento de sistemas.

Ao Professor Dr. Fabio Gomes de Andrade, que me deu a alegria de ser seu orientando e participou de todo o processo de desenvolvimento deste trabalho.

Aos amigos que fiz durante o curso, em especial aos do grupo de WhatsApp "Paulo Renjes", que participaram comigo dessa longa, exaustiva e gratificante jornada.

RESUMO

Nos últimos anos, o movimento para a publicação de dados abertos governamentais tem conquistado uma popularidade cada vez maior. Para facilitar o compartilhamento destes dados, muitos governos implementam portais de dados abertos, que são utilizados tanto por provedores quanto por clientes desse tipo de dado. Os provedores de dados abertos usam esses portais para anunciar os seus conjuntos de dados. Para isso, eles cadastram os seus conjuntos de dados, fornecendo uma série de metadados acerca dos dados que são oferecidos e as URLs a partir das quais os arquivos podem ser acessados. Já os clientes usam os portais para localizar os conjuntos de dados do seu interesse. Embora os portais de dados abertos tenham facilitado a localização dos dados que se encontram disponíveis, eles ainda têm grandes limitações. O maior problema ocorre porque as suas ferramentas de busca geralmente resolvem as consultas apenas com base em palavras-chaves, o que reduz a qualidade dos resultados. Com o objetivo de resolver este problema, este trabalho propõe um motor de busca para a recuperação de dados abertos governamentais. Para melhorar a qualidade das consultas, a ferramenta permite a resolução de consultas tanto em nível de conjunto de dados quanto em nível de arquivos. Além disso, ela está apta a resolver consultas com restrições espacial, temporal e temática.

Palavras-chave: Dados Abertos Governamentais. Recuperação da Informação. Anotação de Dados.

ABSTRACT

In recent years, the open government data movement has gained increasing popularity. To facilitate the sharing of these data, many governments have implemented open government data portals, which are used by both providers and clients of this type of data. Open data providers use these portals to advertise their datasets. For this, they register their datasets, providing several metadata about the data that are being offered and the URL from which they can be downloaded. On the other hand, clients use these portals to find out the datasets of their interest. Although open data portals have made it easier to find the data that is available, they still have major limitations. The biggest limitation occurs because their search engines usually solve queries based only on keywords, which reduces the quality of the results. In order to solve this problem, this work proposes a search engine to improve information retrieval in open government data portals. To enhance the quality of the queries, the tool implemented in this work can solve queries at the level of dataset and resource. Moreover, it is able to perform queries with spatial, temporal, and thematic constraints.

Keywords: Open Government Data. Information Retrieval. Data Annotation.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
ASCII	*American Standard Code for Information Interchange*
CKAN	*Comprehensive Knowledge Archive Network*
CSS	*Cascading Style Sheets*
CSV	*Comma Separated Values*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
JSON	*JavaScript Object Notation*
LAI	Lei de Acesso à Informação
OGC	*Open Geospatial Consortium*
OGD	*Open Government Data*
OKF	*Open Knowledge Foundation*
REST	*Representational State Transfer*
RPC	*Remote Procedure Call*
SGBD	*Sistema de Gerenciamento de Banco de Dados*
SQL	*Structured Query Language*
TCC	Trabalho de Conclusão de Curso
TCU	Tribunal de Contas da União
URL	*Uniform Resource Locator*
UTF-8	*8-bit Unicode Transformation Format*
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, a disponibilização de dados governamentais tem conquistado uma grande popularidade no mundo todo. O movimento *Open Government Data* (OGD) teve grande contribuição para a ocorrência desse fenômeno. Esse movimento começou a crescer nos Estados Unidos, em 2013, quando o então presidente Barack Obama assinou uma lei determinando a abertura dos dados produzidos pelas agências do governo federal (THE WHITE HOUSE, 2013), com o intuito de oferecer mais transparência e permitir a reutilização desses conjuntos de dados para o desenvolvimento de novos serviços para a população.

Uma importante característica desse movimento é que o mesmo defende o compartilhamento dos dados com poucas restrições. De acordo com a *Open Knowledge Foundation* (OKF, 2019), "o conhecimento é considerado aberto se qualquer pessoa estiver livre para acessá-lo, utilizá-lo, modificá-lo, e compartilhá-lo (restrito, no máximo, a medidas que preservam a proveniência e a abertura)". Quando os dados que representam esse conhecimento são produzidos e compartilhados por autoridades públicas, são chamados de dados abertos governamentais.

Desde então, vários portais de dados abertos têm sido desenvolvidos por governos do mundo inteiro, em diferentes níveis (federal, regional, estadual e municipal), para a disponibilização dos conjuntos de dados de suas agências. Por exemplo, atualmente, o portal de dados abertos do governo federal brasileiro[1] disponibiliza mais de 10000 conjuntos de dados, enquanto que os portais dos governos do Reino Unido[2] e dos Estados Unidos[3] disponibilizam, respectivamente, mais de 50000 e 250000 conjuntos de dados. A publicação de dados abertos é de grande importância, uma vez que os mesmos representam um canal de comunicação entre os governantes e os cidadãos. Além disso, esses dados podem ser usados como base para a descoberta de informações úteis e bastante significativas para a realidade atual de qualquer país, incluindo o Brasil.

De acordo com o Tribunal de Contas da União (TCU, 2015), no documento “5 motivos para a abertura de dados na administração pública”, existem cinco motivos para a abertura de dados na administração pública. Um dos motivos é que a abertura tornou-se obrigatória. O artigo 8º da Lei 12.527/2011, que é chamada de Lei de Acesso à Informação (LAI) (BRASIL, 2011), estabelece que as informações de interesse coletivo ou geral devem ser obrigatoriamente divulgadas pelos órgãos e entidades públicas

[1] “Portal Brasileiro de Dados Abertos”. Disponível em: http://dados.gov.br/. Acessado em 11 Fev. 2021
[2] “Find open data - data.gov.uk”. Disponível em: https://data.gov.uk/. Acessado em 11 Fev. 2021
[3] “Data.gov”. Disponível em: https://www.data.gov/. Acessado em 11 Fev. 2021

em seus sítios oficiais. Os outros quatro motivos são: promover a transparência de dados na gestão pública, a possibilidade de permitir que a própria sociedade possa contribuir com o desenvolvimento de serviços inovadores ao cidadão, o aprimoramento da qualidade dos dados governamentais e a viabilização de novos negócios. Outras vantagens que podem ser obtidas por meio da disponibilização de dados abertos por parte dos governos são apontadas por Ubaldi (2013, p. 4, tradução nossa):

> A acessibilidade aprimorada de dados pode permitir maior colaboração dentro dos governos, bem como entre agências governamentais e a sociedade em geral, incluindo o setor privado, organizações da sociedade civil e cidadãos. Isso está estimulando uma mudança na cultura organizacional do setor público, não apenas em direção à abertura, transparência e responsabilidade, mas também ao compartilhamento, colaboração e maior engajamento público.

Mediante o livre acesso aos dados abertos governamentais, desenvolvedores de aplicações e cientistas de dados podem realizar análises precisas, criar relatórios detalhados, utilizar e/ou construir ferramentas que possam automatizar a criação de relatórios de forma dinâmica, facilitando, assim, a informatização da sociedade no que se refere ao melhor entendimento de informações públicas, e fomentando o engajamento da população no planejamento e desenvolvimento de políticas públicas.

A grande quantidade de conjuntos de dados disponibilizados pelos portais de dados abertos governamentais faz surgir a necessidade de se desenvolver mecanismos que permitam ao cidadão encontrar, de forma rápida, os conjuntos de dados nos quais ele tem interesse. Para isso, cada portal de dados abertos oferece uma ferramenta de busca, que pode ser utilizada pelos seus clientes para a recuperação dos seus conjuntos de dados. Como o número de conjuntos de dados ofertados cresce a cada dia, torna-se cada vez mais necessário que essas ferramentas de busca sejam pensadas e projetadas para resolver diversos tipos de consultas de forma efetiva.

1.1 MOTIVAÇÃO

Os portais de dados abertos atuais oferecem ferramentas de busca que permitem que os seus usuários localizem os conjuntos de dados de seu interesse. Entretanto, essas ferramentas possuem limitações importantes, que reduzem a qualidade dos resultados retornados por suas consultas. Uma característica importante das ferramentas de busca desses portais é que elas resolvem as consultas apenas com base em palavras-chaves. Assim, ao realizar uma consulta, o usuário especifica um conjunto de palavras-chaves de seu interesse, e a ferramenta retorna como resultado todos os conjuntos de dados que tenham essas palavras em sua descrição.

A resolução de buscas com base apenas em palavras-chaves dificulta a resolução de diversos tipos de consultas. Por exemplo, considere um conjunto de dados hipotético chamado “Casos de Dengue na Paraíba - 2018 a 2020”, contendo a quantidade de casos de dengue em cada cidade do estado em cada um desses três anos. Caso o usuário esteja interessado em conjuntos de dados que contenham informações sobre a cidade de Itaporanga, na Paraíba, o conjunto de dados em questão, mesmo contendo dados relevantes para a consulta, não será recuperado, uma vez que o nome da cidade informada pelo usuário não aparece em sua descrição.

As ferramentas de buscas atuais também têm dificuldades para resolver consultas com restrições temporais. Por exemplo, caso o usuário realize uma consulta procurando por dados sobre o ano de 2019, o conjunto de dados citado anteriormente também será descartado, uma vez que o ano em questão não aparece na descrição do conjunto.

Finalmente, o portal se limita a realizar consultas apenas em nível de conjuntos de dados, não sendo possível realizar uma busca diretamente por recursos. O problema disso é que recursos que possuem maior relevância para a consulta podem estar em um conjunto de dados que não é tão relevante como um todo, ficando em uma posição no ranking não favorável para o recurso. Além disso, conjuntos de dados podem possuir um grande número de recursos, o que dificulta a procura por um recurso específico.

Com o intuito de resolver essas limitações, este projeto tem como objetivo o desenvolvimento de um motor de busca para conjuntos de dados abertos governamentais. Para alcançar esse objetivo, propõe-se o desenvolvimento de uma solução na qual os conjuntos de dados são descritos por metadados adicionais extraídos a partir da análise da sua descrição e do seu conteúdo.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral deste Trabalho de Conclusão de Curso (TCC) consiste no desenvolvimento de um motor de busca que visa facilitar a recuperação de informações disponibilizadas em portais de dados abertos governamentais por parte dos seus usuários.

1.2.2 Objetivos Específicos

O trabalho proposto tem ainda os seguintes objetivos específicos:

- compreender como os dados são ofertados pelos portais de dados abertos atuais;
- entender o funcionamento das ferramentas atuais utilizadas para a recuperação de dados abertos;
- identificar, a partir do conteúdo dos dados publicados em portais de dados abertos, metadados que possam ser utilizados para melhorar o processo de recuperação da informação;
- gerar um banco de dados centralizado para a recuperação de dados abertos governamentais;
- desenvolver uma ferramenta com interface *web* que permita a recuperação de dados abertos governamentais por parte de qualquer usuário;
- aplicar a solução desenvolvida, usando como estudo de caso pelo menos um portal real de dados abertos governamentais.

1.3 TRABALHOS RELACIONADOS

O trabalho proposto neste TCC está relacionado a algumas ferramentas já disponíveis na Internet. Uma dessas ferramentas é o *Google Dataset Search*, que consiste em um projeto desenvolvido por Brickley et al. (2019). O projeto tem como objetivo a implementação de um motor de busca para todos os conjuntos de dados abertos disponibilizados na web. As consultas por conjuntos de dados são resolvidas a partir dos metadados que são publicados pelos proprietários. Como esses metadados precisam ser semanticamente aprimorados, os autores incentivam a padronização de esquemas dos metadados para cada conjunto de dados proveniente de uma área específica. Uma vez disponibilizados, os metadados são agregados, normalizados e reconciliados, para que possam ser recuperados por um mecanismo de pesquisa de conjuntos de dados.

Diferente do *Google Dataset Search*, o trabalho proposto neste TCC é direcionado a portais de dados governamentais. Assim, já existe uma padronização no esquema dos metadados encontrados no portal, o que facilita a busca por metadados que serão úteis no processo de indexação. Tanto a ferramenta do Google, quanto a ferramenta proposta neste TCC utilizam, além dos metadados, os próprios dados para o processo de indexação. Entretanto, a ferramenta do Google foca mais no uso dos metadados, utilizando os dados para suprir a falta de informações obtidas em certos atributos dos metadados.

Outro trabalho relacionado foi desenvolvido por Marquez et al. (2010). Eles desenvolveram um motor de busca de dados abertos espaciais fornecidos por meio

de serviços *Open Geospatial Consortium* (OGC). Nesse trabalho, foi desenvolvido um *crawler* composto por três camadas responsáveis por: I) rastrear serviços OGC, II) buscar divisões políticas e não políticas; III) usar os dados dessas divisões para extrair informações relacionadas aos serviços e indexá-los. Para as consultas feitas pelo usuário, além da busca de serviços com divisões políticas relacionadas, é realizada uma expansão no significado das palavras-chave, utilizando uma base de conhecimento contendo a taxonomia das palavras que representam divisões não políticas e diversos domínios de conhecimento. Para a indexação de serviços relacionados a divisões políticas, uma base de dados de nome de lugares é utilizada para validar as divisões políticas encontradas nos metadados e obter a hierarquia ascendente de suas divisões políticas.

Foi utilizado uma estrutura de árvore (*R-Tree*) própria para relacionar dados espaciais. A ideia de organizar os dados espaciais dessa forma serviu como referencial para que neste trabalho de TCC fosse utilizada uma estrutura baseada em grafo. Nesse grafo, existe um conjunto de nós que representam lugares existentes no Brasil e outro representando os recursos oferecidos pelo portal. Então, arcos são usados para associar os recursos aos locais que são referenciados em seu conteúdo.

O trabalho de Stróżyna et al. (2018) faz a seleção e a detecção de diversas fontes de dados do domínio marítimo. Essas fontes de dados são abertas e podem conter dados estruturados ou não (páginas na web e na *Deep web*, arquivos PDF, CSV e XLS). Após a identificação e escolha das melhores fontes de dados é feita a recuperação automática dos dados, através do Módulo de Aquisição de Dados. Para dados estruturados como o CSV é feita uma busca, linha por linha, para se obterem dados sobre entidades relacionadas ao domínio marítimo. Para os outros tipos de dados também é feita a busca de informações no conteúdo utilizando, por exemplo, expressões regulares. Os dados recuperados de fontes heterogêneas são mesclados e armazenados numa base de dados única onde possui todas as entidades relacionadas ao domínio marítimo (como navios, portos e tipos de embarcações). Este TCC realiza, também, a busca de informações no conteúdo dos arquivos CSV e recupera dados numa base de dados específica, esses dados são interpretados e utilizados em consultas com restrição espacial.

Por fim, o trabalho Kacprzak et al. (2019) é uma pesquisa sobre as características de consultas para se obter conjuntos de dados. A pesquisa foi feita analisando os registros gerados a partir de várias consultas realizadas em portais de dados abertos além de consultas feitas em mecanismos de busca na web, que resultaram na entrada dos usuários nestes portais. Durante a pesquisa foram verificados padrões como o tamanho médio de palavras em uma consulta, a estrutura dessa consulta, quais

mecanismos de busca foram usados e quais as propriedades mais importantes em uma consulta realizada pelo usuário. Esse último padrão foi importante para este TCC, pois foi descoberto que muitas consultas feitas por usuários, comumente possuem informações espaciais e temporais. Assim, essas consultas são as que precisam de maior suporte. No trabalho proposto neste TCC, o usuário tem a possibilidade de realizar consultas com restrições espaciais (busca por nome de municípios, estados, regiões) e temporais (busca por qualquer intervalo de data composta por dia, mês e ano).

1.4 CONTRIBUIÇÕES

O trabalho oferta as seguintes contribuições:

- O desenvolvimento de um banco de dados centralizado para a recuperação de dados abertos governamentais;
- O desenvolvimento de um motor de busca capaz de recuperar informações tanto em nível de conjuntos de dados quanto em nível de recursos;
- O desenvolvimento de um motor de busca capaz de recuperar dados abertos governamentais a partir de restrições espaciais, temporais, temáticas e multidimensionais;
- A proposição de métricas de ranking para avaliar a relevância de cada recurso recuperado com relação às restrições espaciais, temporais, temáticas e multidimensionais.

1.5 METODOLOGIA

O desenvolvimento deste trabalho de conclusão de curso contou com o desenvolvimento das seguintes atividades:

- **Estudo sobre estado da arte (A1):** nessa etapa foi realizado um estudo mais aprofundado sobre como os dados abertos governamentais são ofertados e recuperados pelas ferramentas de buscas atuais. Também foram pesquisadas ferramentas desenvolvidas com o intuito de melhorar a recuperação desses dados. A fim de manter o conhecimento sempre atualizado, essa atividade foi realizada durante todo o desenvolvimento do trabalho;
- **Análise e projeto (A2):** nessa etapa foram realizadas as atividades referentes à análise e projeto da ferramenta que foi implementada. Dentre essas atividades

estão o levantamento dos requisitos funcionais, a definição da arquitetura e a elaboração do esquema do banco de dados;

- **Desenvolvimento do módulo de coleta de dados (A3):** nessa etapa foi implementado um módulo responsável por interagir com o portal de dados abertos a ser usado como estudo de caso. Esse módulo é responsável por coletar os metadados de cada conjunto de dados oferecido pelo portal. Além disso, o módulo recupera, para fins de análise, os arquivos disponibilizados em cada conjunto de dados;
- **Desenvolvimento do módulo de extração de metadados (A4):** nessa etapa, foi implementado um módulo responsável por processar as informações obtidas por meio do módulo de coleta de dados. Esse módulo é responsável por extrair os metadados que foram usados, para melhorar o processo de recuperação da informação;
- **Desenvolvimento do motor de busca (A5):** nessa etapa foi implementado o motor de busca, que é responsável por receber as consultas dos usuários e resolvê-las com base nas informações extraídas pelo módulo de extração de metadados;
- **Elaboração do documento final de TCC (A6):** nessa etapa foi elaborado o documento de TCC. Ela também foi realizada ao longo de todo o desenvolvimento do trabalho.

1.6 ESTRUTURA E ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento está dividido em três capítulos.

O capítulo 2, relacionado a fundamentação teórica, onde foram abordados tecnologias e conceitos utilizados neste trabalho e que são importantes para compreensão do desenvolvimento da ferramenta proposta.

O capítulo 3 apresenta os artefatos, resultantes da análise e projeto, bem como descreve a implementação da ferramenta, explicando os principais detalhes de cada um de seus módulos e utilizando exemplos de como eles funcionam.

Finalmente, o capítulo 4 apresenta as considerações finais e aponta possíveis trabalhos futuros que podem ser realizados para o aperfeiçoamento da ferramenta proposta.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos e tecnologias utilizados para desenvolvimento deste trabalho. Inicialmente, o capítulo apresenta o software *Comprehensive Knowledge Archive Network* (CKAN) e sua API. Depois, é apresentada a ferramenta *Solr*.

2.1 CKAN

O CKAN[4] (CKAN, 2020) é um software livre que permite a construção de sítios para a publicação de dados abertos, possibilitando o gerenciamento e a publicação de grandes coleções de dados. O CKAN foi criado por Rufus Pollock, fundador da *Open Knowledge Foundation* (OKF), e continua sendo desenvolvido por dezenas de colaboradores. Desde a sua criação, ele tem sido utilizado por governos de diferentes níveis (países, estados, regiões e municípios) para a divulgação dos seus dados para a população. Exemplos de países que usam esse software para a publicação dos seus conjuntos de dados incluem os Estados Unidos, o Reino Unido, o Canadá, a Austrália e o Brasil.

O CKAN possui diversas funcionalidades, tanto para aqueles que desejam publicar dados, quanto para aqueles que desejam consumi-los. Para a utilização de suas funcionalidades é necessário um conhecimento prévio sobre o que são *datasets*, *resources*, *metadata*, *organizations* e os níveis de autorização dos usuários em uma *organization*. Nessa seção, são discutidos apenas os conceitos necessários para a recuperação dos dados armazenados, uma vez que neste trabalho não foram utilizadas as funcionalidades de criação, remoção e atualização de dados, que exigem autenticação do usuário.

Os *datasets* (também chamados de *packages*) são os resultados que são retornados em cada pesquisa feita pelo usuário no motor de busca. Cada instância de um *dataset* representa um conjunto de dados ofertado por alguma organização. Cada *dataset* é formado por:

- ***metadata* (metadados)**: são as informações que descrevem o conjunto de dados ofertado, como, por exemplo, o título, a descrição, o autor, o identificador, o *e-mail* do autor, as *tags* e informações sobre os *resources* (recursos);

[4] "CKAN". Disponível em: https://github.com/ckan/ckan. Acessado em 11 Fev. 2021

- ***resources***: os recursos contém metadados sobre os recursos oferecidos por um conjunto de dados. Cada recurso corresponde a um arquivo que compõe o conjunto de dados. Não há uma restrição para o formato desses recursos, embora eles sejam normalmente ofertados em formatos que podem ser processados por máquina, tais como CSV, JSON e XLS. Cada conjunto de dados é comumente formado por vários recursos;

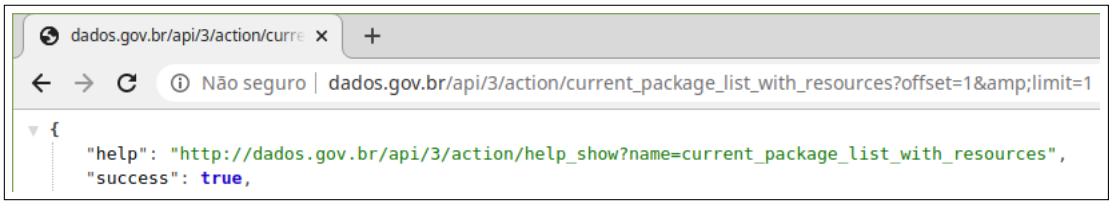
Cada conjunto de dados normalmente é relacionado a uma *organization*, que representa a organização responsável por sua publicação. Cada instância do CKAN pode ser usada para publicar os dados de diversas organizações, que podem representar, por exemplo, ministérios, institutos, universidades e agências de um país, assim como ocorre no portal de dados governamentais abertos do Brasil.

2.1.1 A API CKAN

O CKAN disponibiliza uma API, chamada "*API Action*", que é implementada na linguagem de programação Python e funciona no estilo *Remote Procedure Call* (RPC). Essa API fornece todas as funcionalidades disponíveis na interface gráfica do CKAN, além de outras operações adicionais. Nesta seção, será abordada apenas a funcionalidade de realização de requisições à API com o intuito de recuperar metadados acerca dos conjuntos de dados, organizações e recursos.

A Figura 1 mostra um exemplo de requisição para a API através do *endpoint* de recuperação dos metadados do portal de dados governamentais abertos do Brasil. A resposta da requisição é um arquivo no formato JSON. Os seus primeiros atributos são "*help*" e "*success*". O primeiro atributo tem como valor uma URL para uma página contendo uma breve descrição da funcionalidade utilizada, enquanto que o segundo atributo possui o valor indicando se a requisição foi realizada com sucesso (*true*) ou não (*false*).

Figura 1 – Exemplo de uma requisição para API *Action*



Fonte: Elaborado pelo autor

No restante do documento retornado após a requisição pode-se encontrar a descrição dos conjuntos de dados ofertados pela instância que está sendo acessada, dos seus respectivos recursos e da organização responsável por sua publicação. A Figura 2 mostra alguns metadados que descrevem um conjunto de dados. Nela, percebe-se que os metadados descrevem informações importantes como o identificador único, o mantenedor, a data de criação, a data de última modificação e o autor do *dataset*, além de outras informações adicionais.

Figura 2 – Metadados referentes a um conjunto de dados


```
"result": [
  {
    "license_title": "Creative Commons Atribuição",
    "maintainer": "Departamento de Governo Eletrônico",
    "relationships_as_object": [],
    "private": false,
    "maintainer_email": "dominios@planejamento.gov.br",
    "num_tags": 4,
    "id": "32553ebf-7a6f-40c0-bec9-5435850bcabf",
    "metadata_created": "2015-10-29T18:29:55.260609",
    "metadata_modified": "2020-01-24T06:00:35.430585",
    "author": "Departamento de Governo Eletrônico",
    "author_email": "dominios@planejamento.gov.br",
    "state": "active",
    "version": "",
    "creator_user_id": "7260a7bf-7f61-4f04-b4cb-1a46c2638a3a",
    "type": "dataset",
```


Fonte: Elaborado pelo autor

A Figura 3 mostra alguns metadados que descrevem um dos recursos oferecidos pelo conjunto de dados mostrado na Figura 2. Nela, percebe-se que os metadados apresentados permitem a identificação de informações importantes como o nome, a descrição, a URL a partir da qual o recurso pode ser acessado, a data da última atualização, o formato do arquivo no qual ele é disponibilizado, o seu identificador único e o conjunto de dados ao qual ele pertence.

Figura 3 – Metadados referentes a um recurso


```
"resources": [
  {
    "mimetype": null,
    "cache_url": null,
    "hash": "",
    "description": "Informações sobre os Domínios Gov.br registrados e seus respectivos responsáveis",
    "name": "Domínios GOV.BR",
    "format": "CSV",
    "url": "http://dominios.governoeletronico.gov.br/dados-abertos/Dominios_GovBR_basico.csv",
    "datastore_active": true,
    "cache_last_updated": null,
    "package_id": "32553ebf-7a6f-40c0-bec9-5435850bcabf",
    "created": "2015-10-29T16:31:41.004381",
    "state": "active",
    "mimetype_inner": null,
    "last_modified": "2020-01-24T00:00:00",
    "position": 0,
    "revision_id": "019ea795-7f80-4352-9fc4-778a62d73270",
    "url_type": null,
    "id": "197a0106-c93b-42fc-bb4e-c3095baee1a0",
    "resource_type": null,
    "size": null
  },
```


Fonte: Elaborado pelo autor

Finalmente, a Figura 4 mostra os metadados que descrevem a organização responsável pela publicação do conjunto de dados. Nela, percebe-se que é possível identificar informações importantes como o nome, o título, a descrição e o seu identificador único dentro da ferramenta. Por meio desse identificador único é possível, por exemplo, distinguir as organizações e classificar os conjuntos de dados.

Figura 4 – Metadados referentes a uma organização


```
"organization": {
    "description": "O [Ministério do Planejamento, Desenvolvimento e Gest
    "created": "2015-09-08T14:13:08.462246",
    "title": "Ministério do Planejamento, Desenvolvimento e Gestão - MP",
    "name": "ministerio-do-planejamento-desenvolvimento-e-gestao-mp",
    "is_organization": true,
    "state": "active",
    "image_url": "http://dados.gov.br/wp/wp-content/uploads/2015/10/minis
    "revision_id": "6cf07ead-c3e4-4c68-b930-f1d6fd8f727e",
    "type": "organization",
    "id": "fadff597-55a0-4619-8a81-cfecc9c0c2bb",
    "approval_status": "approved"
},
```


Fonte: Elaborado pelo autor

2.2 SOLR

O *Solr*[5], de acordo com a *Apache Software Foundation* (2019), é uma plataforma de busca textual completa e independente, de código fonte aberto, desenvolvida e gerenciada pela mesma. Ele foi desenvolvido utilizando-se como base o *Apache Lucene*[6], que é uma ferramenta que provê funcionalidades de indexação e busca de documentos. Além disso, o *Solr* possui uma API REST que facilita a sua integração com aplicações desenvolvidas em diversas linguagens de programação.

No contexto do *Solr*, um núcleo é uma instância de índice com seus *logs* de transação e seus arquivos de configurações. Um servidor *Solr* pode conter vários núcleos, o que permite que o usuário possa indexar diferentes estruturas. Para se definir a estrutura dos dados, o núcleo possui um arquivo chamado "*schema.xml*", no qual os atributos da estrutura são descritos.

Neste trabalho os atributos possuem as seguintes propriedades: o nome do atributo, o tipo (texto, *booleano*, inteiro, data, etc), se ele é indexado ou não (os atributos que serão utilizados como parâmetro na busca, devem ser indexados), se o atributo é armazenado (ele pode ser indexado, mas não armazenado na base de dados), se ele é obrigatório, se possui um valor único em comparação com os outros valores armazenados e se é multivalorado.

A Figura 5 mostra um exemplo de definição de três atributos de uma estrutura. Cada atributo é definido por meio de uma instância da *tag field*. O primeiro atributo foi chamado de id, o seu tipo é *string*, e foi definido que ele deve ser indexado e armazenado na base de dados do *Solr*, que ele é obrigatório e não é multivalorado. O segundo atributo, com nome *metadata*, é do tipo *text_general*. Esse tipo foi descrito no mesmo arquivo de configuração, assim como todos os outros tipos. O terceiro atributo representa um valor gerado automaticamente pelo *Solr*, e significa a versão do documento. Finalmente, a última *tag* mostrada na figura possui o nome do atributo que representa a chave única. Nesse caso, o atributo com o nome id foi escolhido.

[5] "Apache Solr". Disponível em: https://lucene.apache.org/solr/. Acessado em 11 Fev. 2021

[6] "Apache Lucene - Welcome to Apache Lucene". Disponível em: https://lucene.apache.org/. Acessado em 11 Fev. 2021

Figura 5 – Exemplo de definição de atributos

```
<field name="id" type="string" indexed="true" stored="true" required="true"
       multiValued="false" />

<field name="metadata" type="text_general" indexed="true" stored="false"
       required="true" multiValued="false"/>

<field name="_version_" type="plong" indexed="true" stored="true" multiValued="false"/>

<uniqueKey>id</uniqueKey>
```

Fonte: Elaborado pelo autor

A Figura 6 mostra a declaração dos tipos utilizados pelos atributos da Figura 5. Nela, percebe-se que a declaração dos tipos é feita utilizando-se a *tag fieldType*. Dentre todas as propriedades que podem ser aplicadas ao *fieldType*, foram utilizadas, para este trabalho, somente aquelas usadas para informar o nome do tipo e a sua classe Java correspondente. Por exemplo, a primeira *tag fieldType* é a declaração de um tipo chamado *string* (que será utilizado na definição da *tag field* no parâmetro *type*) e a classe Java do tipo será *StrField* (pertencente ao pacote Solr), que é um tipo de texto codificado no formato UTF-8 ou Unicode com tamanho limite ligeiramente menor que 32K. A última *tag fieldType*, com o nome *textgeneral*, se fecha, diferente das outras duas que não se fecham. Isso acontece porque a classe Java referente a esse tipo é a *TextField* (também do pacote *Solr*), que representa um texto de múltiplas palavras ou *tokens* que são analisáveis.

A classe *TextField* permite diversas configurações e pode ser composta de tokenizadores e filtros que são informados dentro da *tag analyzer*. Pode-se perceber que, no exemplo mostrado na Figura 6, o *fieldType* possui duas *tags analyzer*. A primeira delas possui as configurações para análise no processo de indexação e a segunda possui as configurações para análise em tempo de consulta.

Figura 6 – Exemplo de declaração do tipo do atributo

```
<fieldType name="string" class="solr.StrField"/>

<fieldType name="plong" class="solr.LongPointField"/>

<fieldType name="text_general" class="solr.TextField">
  <analyzer type="index">
    <tokenizer class="solr.OpenNLPTokenizerFactory"
               sentenceModel="./lang/pt-sent.bin"
               tokenizerModel="./lang/pt-token.bin"/>
    <filter class="solr.WordDelimiterGraphFilterFactory" generateNumberParts="0"/>
    <filter class="solr.FlattenGraphFilterFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StopFilterFactory" words="./lang/stopwords.txt" />
    <filter class="solr.OpenNLPPOSFilterFactory" posTaggerModel="./lang/pt-pos-maxent.bin"/>
    <filter class="solr.OpenNLPLemmatizerFilterFactory" dictionary="./lang/pt-br-lemmatizer.dict"/>
    <filter class="solr.ASCIIFoldingFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.OpenNLPTokenizerFactory"
               sentenceModel="./lang/pt-sent.bin"
               tokenizerModel="./lang/pt-token.bin"/>
    <filter class="solr.WordDelimiterGraphFilterFactory" generateNumberParts="0"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="./lang/stopwords.txt" />
    <filter class="solr.OpenNLPPOSFilterFactory" posTaggerModel="./lang/pt-pos-maxent.bin"/>
    <filter class="solr.OpenNLPLemmatizerFilterFactory" dictionary="./lang/pt-br-lemmatizer.dict"/>
    <filter class="solr.SynonymGraphFilterFactory" synonyms="./lang/synonyms.txt"
                                                   ignoreCase="true" expand="true"/>
    <filter class="solr.ASCIIFoldingFilterFactory"/>
  </analyzer>
</fieldType>
```

Fonte: Elaborado pelo autor

O Quadro 1 descreve cada uma das opções de filtros que são utilizadas depois do processo de *tokenização*, que é o processo de dividir um texto em uma sequência de partes (que podem ser palavras) analisáveis chamadas de *tokens*.

Quadro 1 – Descrição dos filtros

Filtro	Descrição
WordDelimiterGraphFilterFactory	Os tokens são divididos pelos delimitadores de palavras.
FlattenGraphFilterFactory	É usado após filtros que retornam grafos, transformando-os em estruturas que o indexador possa manipular.
LowerCaseFilterFactory	O Texto dos tokens são convertidos em textos com letras minúsculas.
StopFilterFactory	Os tokens que estiverem presente em uma determinada lista de palavras não serão incluídos no processo de análise.
OpenNLPPOSFilterFactory	Define o atributo de tipo do token para a classe gramatical apropriada de acordo com o modelo configurado.
OpenNLPLemmatizerFilterFactory	Troca o texto do token pelo seu respectivo lema que é encontrado no dicionário que foi definido.
ASCIIFoldingFilterFactory	Converte caracteres alfabéticos, números e símbolos em seus caracteres equivalentes na tabela ASCII.
SynonymGraphFilterFactory	Converte a palavra do token em seus respectivos sinônimos encontrados no dicionário que foi fornecido.

Fonte: Elaborado pelo autor

2.2.1 Processamento de Texto

O processamento de texto no *Solr*, utilizando os filtros descritos no Quadro 1, ocorre em uma série de etapas. A primeira etapa do processamento do texto é a tokenização. A Figura 7 exibe o resultado da tokenização da frase “Dados cadastrais das Instituições de Ensino Superior no Brasil”.

Figura 7 – Tokens extraídos de um texto

Fonte: Elaborado pelo autor

Depois da tokenização, a próxima etapa consiste em dividir as palavras por meio dos seus delimitadores. Por exemplo, a palavra “seguro-desemprego”, que é uma

palavra composta que possui um hífen, após passar pela primeira etapa, é dividida em duas palavras: “seguro” e “desemprego”. Esse processo torna possível a busca pela palavra com ou sem o hífen.

Na terceira etapa, todas as palavras são processadas de forma que sejam representadas apenas com letras minúsculas. Assim, as palavras “Dados”, “Ensino”, “Superior” e “Brasil” que aparecem na Figura 7, são transformadas, respectivamente, para “dados”, “ensino”, “superior” e “brasil”. Esse processo faz com os documentos que contenham essas palavras possam ser recuperados independentemente de possíveis diferenças de caixa utilizadas pelo usuário na hora da formulação da consulta.

Na quarta etapa são removidas as palavras que podem ser ignoradas no resultado da consulta e no processo de indexação. Nessa etapa, palavras que representam preposições, conjunções, artigos, entre outras, como “das”, “de” e “no”, são removidas com base em uma lista de palavras chamada *Stop Words*.

Na quinta etapa as palavras são classificadas gramaticalmente. A classificação gramatical das palavras é útil para a sexta etapa, que é a lematização. A lematização, segundo Toman et al. (2006), é o processo de transformar cada palavra em sua forma básica (lema). Após esse processo, as palavras “dados”, “cadastrais” e “instituições” perdem a sua flexão, e são substituídas, respectivamente, pelas palavras “dado”, “cadastral” e “instituição”. Essa etapa é útil para garantir que documentos que possuam essas palavras sejam recuperados independentemente de suas flexões.

A sétima etapa, que não está presente no processo de indexação mas somente no de resolução de consultas, retorna o(s) respectivo(s) sinônimo(s) de cada palavra. A palavra “ensino”, por exemplo, tem como sinônimos, “educação”, “instrução” e “preparo” que serão retornados, juntamente com a própria palavra, como resultado dessa etapa. Expandir as palavras da consulta para os seus sinônimos permite uma maior abrangência de dados relevantes buscados.

Por fim, na última etapa, as letras das palavras são substituídas pelos seus caracteres equivalentes da tabela ASCII. Por exemplo, a palavra “educação” é transformada para o *token* “educacao”. Com isso, mesmo que o usuário realize buscas com erros de acentuação, os documentos que contêm essa palavra serão recuperados pela ferramenta.

A Figura 8 apresenta o resultado dos *tokens*, exibidos na Figura 7, após as etapas de processamento textual para indexação. Os espaços vazios representam os *tokens* que foram removidos na terceira etapa.

Figura 8 – Tokens após o processamento

Fonte: Elaborado pelo autor

3 MOTOR DE BUSCA PARA DADOS ABERTOS

Este capítulo descreve o processo de desenvolvimento do motor de busca para dados abertos proposto por este TCC. Inicialmente, o capítulo descreve os artefatos gerados na etapa de análise, descrevendo os stakeholders e os requisitos funcionais da ferramenta. Depois, é apresentado o projeto arquitetural, com a descrição de cada módulo da ferramenta. Em seguida, o capítulo descreve os esquemas usados para a implementação dos seus bancos de dados. Finalmente, é fornecida uma descrição mais detalhada da implementação dos módulos.

3.1 ANÁLISE

Esta seção descreve os resultados obtidos na etapa de análise da ferramenta proposta por este TCC.

3.1.1 *Stakeholders*

Os *stakeholders* (as partes interessadas) identificados para a ferramenta são os seus usuários finais, que correspondem a todos os usuários que utilizarão a ferramenta com propósito de encontrar conjuntos de dados de seu interesse.

3.1.2 Requisitos Funcionais

A ferramenta proposta por este TCC deve satisfazer os seguintes requisitos funcionais:

- **Resolução de consultas em dois níveis (R1):** diferentemente das ferramentas de busca atuais, que apenas realizam consultas em nível de conjuntos de dados, a ferramenta proposta neste TCC deve resolver consultas tanto em nível de conjuntos de dados quanto em nível de recursos. No primeiro tipo de consulta, devem ser recuperados todos os conjuntos de dados que tenham pelo menos um recurso que satisfaça os critérios de seleção definidos na consulta. No segundo tipo de consulta, a ferramenta deve selecionar diretamente os recursos que satisfaçam os critérios definidos na consulta do usuário;
- **Resolução de consultas espaciais (R2):** a ferramenta deve ser capaz de resolver consultas com restrições espaciais. Nesse tipo de consulta, o usuário deve fornecer o nome de um lugar, e a ferramenta deve retornar todos os recursos cujo conteúdo tenha algum dado sobre a localidade desejada;

- **Resolução de consultas temporais (R3):** a ferramenta deve ser capaz de resolver consultas com restrições temporais. Nesse tipo de consulta, o usuário deve fornecer o intervalo de tempo de seu interesse, e a ferramenta deve retornar todos os recursos, cujo conteúdo tenha algum dado sobre o período desejado;
- **Resolução de consultas temáticas (R4):** a ferramenta deve ser capaz de resolver consultas com restrições temáticas. Nesse tipo de consulta, o usuário deve fornecer uma ou mais palavras-chaves, correspondentes aos temas de seu interesse, e a ferramenta deve retornar todos os recursos cujo conteúdo tenha algum dado sobre o tema desejado;
- **Resolução de consultas multidimensionais (R5):** a ferramenta deve ser capaz de resolver consultas com mais de um tipo de restrição. Nesse tipo de consulta, o usuário deve fornecer a região espacial, o período de tempo e o tema de seu interesse, e a ferramenta deve recuperar todos os recursos, que tenham algum dado que satisfaça todas as restrições especificadas.

O Quadro 2 mostra a comparação entre os dois primeiros trabalhos descritos na seção 1.3 (os outros dois trabalhos não eram motores de busca) e o trabalho proposto. Foi usado como critério de comparação, se as ferramentas implementam os requisitos funcionais já descritos.

Quadro 2 – Quadro comparativo

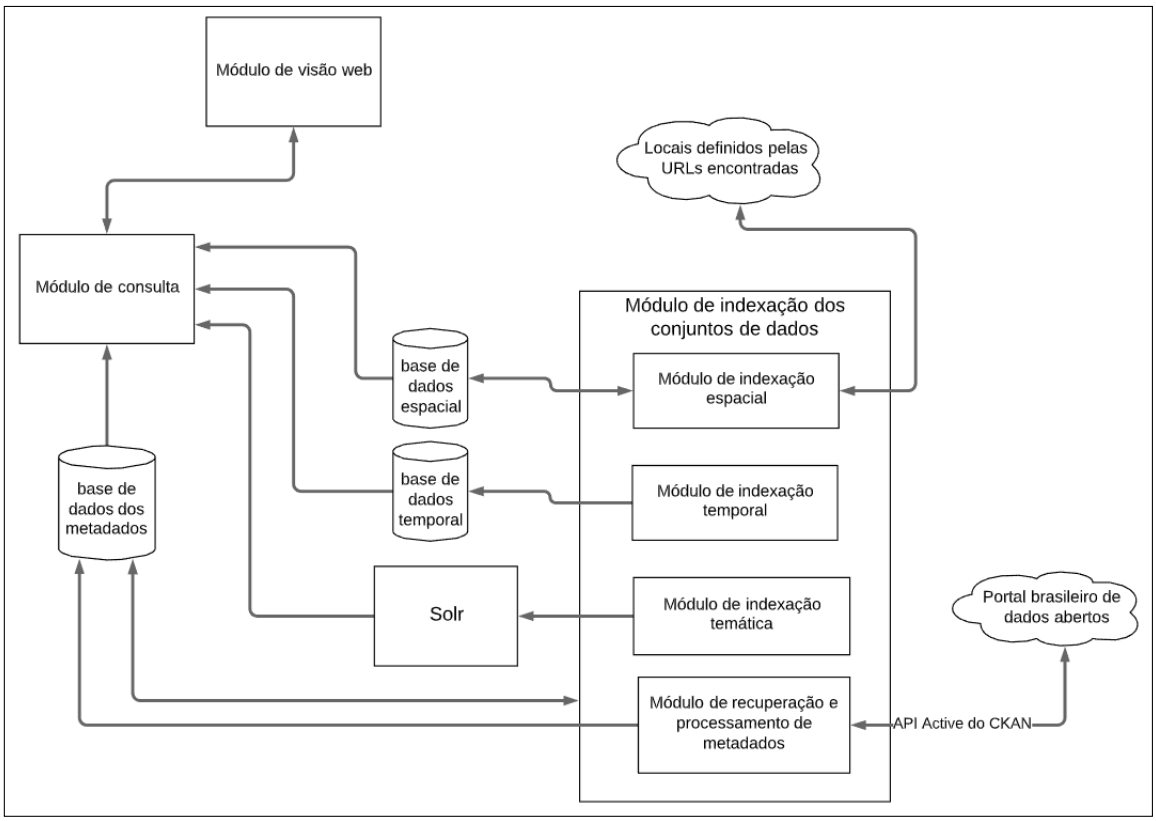
	Este Trabalho	**Google Dataset Search**	**Trabalho de Marquez et al.**
Consultas em dois níveis	sim	não	não
Consulta espacial	sim	não	sim
Consulta temporal	sim	não	não
Consulta temática	sim	sim	sim
Consulta multidimensional	sim	não	sim

Fonte: Elaborado pelo autor

3.2 PROJETO ARQUITETURAL

Esta seção apresenta uma visão geral do projeto arquitetural usado para a implementação da ferramenta, apresentando cada um dos seus módulos. A arquitetura da ferramenta, que é mostrada na Figura 9, é composta por três partes: módulo de visão web, módulo de consulta e módulo de indexação dos conjuntos de dados.

Figura 9 – Projeto Arquitetural



Fonte: Elaborado pelo autor

O módulo de indexação dos conjuntos de dados é responsável por obter e analisar os metadados dos conjuntos de dados (e seus respectivos recursos) oferecidos pelo portal de dados abertos. Esse módulo é dividido em quatro outros módulos: módulo de indexação espacial, módulo de indexação temporal, módulo de indexação temática e módulo de recuperação e processamento de metadados.

O módulo de recuperação e processamento de metadados é responsável por coletar os dados junto ao portal de dados abertos. Ele também é responsável por processar, selecionar os metadados de interesse da ferramenta e armazená-los na base de dados de metadados. O módulo de indexação espacial é responsável por identificar novos metadados a respeito da extensão espacial dos recursos identificados, que são armazenados na base de dados espacial. O módulo de indexação temporal é responsável por identificar a extensão temporal dos recursos identificados e armazená-los na base de dados temporal. Finalmente, o módulo de indexação temática é responsável por tentar encontrar metadados mais precisos para descrever o tema referente aos

dados de cada recurso. Os metadados identificados por esse módulo são armazenados usando uma base de dados que é gerenciada pela ferramenta *Solr*.

O módulo de consulta é responsável por utilizar os bancos de dados da ferramenta para resolver as consultas submetidas pelos usuários finais da ferramenta. Nesse módulo foi implementado um *web service*, que recebe requisições HTTP e retorna a resposta para o módulo de visão *web*. Essa característica é importante porque permite que a ferramenta seja utilizada por outras aplicações de software.

O módulo de visão *web* é o responsável por receber as requisições do usuário e enviá-las ao módulo de consulta. O módulo também exibe os resultados da consulta para o usuário final.

3.3 ESQUEMAS DOS BANCOS DE DADOS

Para a implementação da ferramenta, quatro bancos de dados são utilizados para armazenar os metadados recuperados do portal de dados abertos e os índices gerados durante a fase de processamento dos metadados. São eles: uma base de dados de metadados, uma base de dados espacial, uma base de dados temporal e uma base de dados temática. A decisão pela utilização de bases de dados distintas para cada dimensão dos dados teve como objetivo manter a independência dessas bases de dados de dados. Essa independência permitiu que cada base de dados fosse implementada utilizando-se do sistema de gerenciamento de banco de dados que melhor atendia às particularidades de cada dimensão.

3.3.1 Esquema da base de dados dos metadados

A Figura 10 mostra o esquema lógico usado para a implementação do banco de dados responsável por armazenar os metadados recuperados a partir do portal de dados abertos. Tal esquema, que foi implementado usando o Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL, é formado por três tabelas: portal, *dataset* e *resource*.

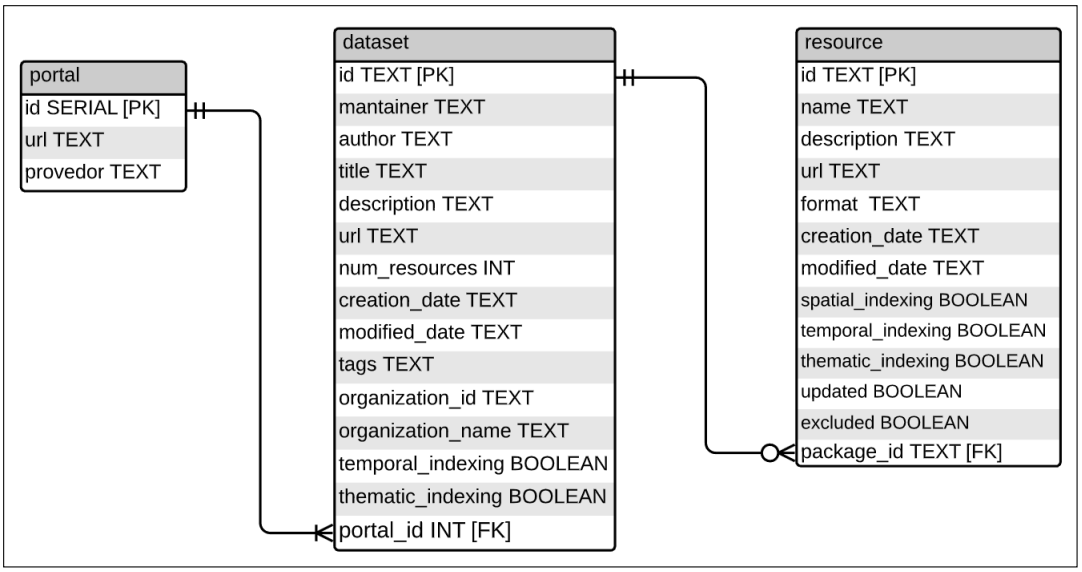
A tabela *portal* armazena os dados referentes aos portais de dados governamentais abertos cujos dados são indexados pela ferramenta. Essa tabela permite que a ferramenta possa armazenar os dados de diversos portais de dados abertos. Para cada portal, são armazenados a sua URL, o nome do seu provedor e um identificador único.

Na tabela *dataset* são armazenados os metadados dos conjuntos de dados identificados a partir de cada portal. Para cada conjunto de dados são armazenados o

seu identificador único, bem como o seu mantenedor, o autor, o título, a descrição, a URL, o número de recursos (arquivos do formato CSV), a data de criação, a data de modificação, as suas *tags*, o identificador da organização responsável por sua produção, o nome da organização, um atributo que indica se os metadados de um conjunto de dados já foram utilizados na indexação temporal, um atributo para identificar se os metadados foram utilizados na indexação temática e, por fim, um identificador único do portal de referência.

Na tabela *resource* são armazenadas as informações acerca dos recursos oferecidos por cada conjunto de dados. Para cada recurso são armazenados o seu identificador único, o nome, a descrição, a URL a partir da qual o arquivo com o seu conteúdo pode ser acessado, o formato do arquivo, a data de criação, a data de modificação, três atributos que indicam, respectivamente, se foi feita a indexação espacial, temporal e temática do recurso, dois atributos que indicam se os metadados do recurso indexado foram atualizados ou excluídos e o identificador do conjunto de dados ao qual ele pertence.

Figura 10 – Esquema lógico da base de dados dos metadados



Fonte: Elaborado pelo autor

O banco de dados possui ainda dois gatilhos: um para a tabela *resource* e outro para a tabela *dataset*. Os dois gatilhos são ativados antes das operações de inserção. O gatilho da tabela *dataset* é usado para verificar se já existe uma tupla no banco com

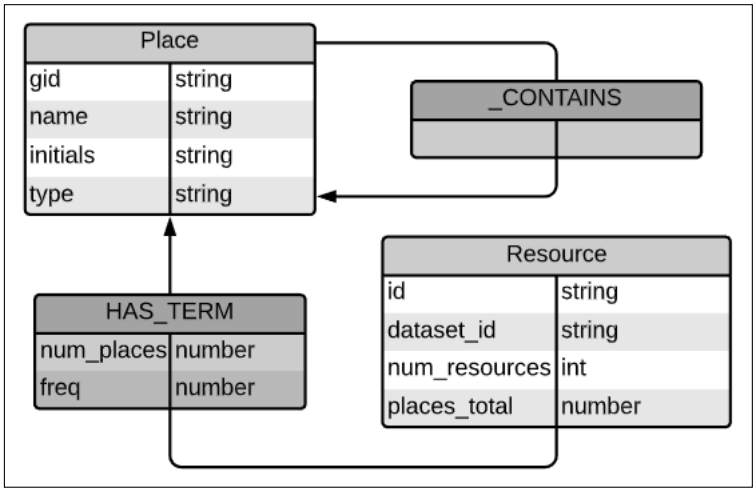
o mesmo identificador único que o da nova tupla que será inserida. Caso não exista, a tupla é simplesmente inserida na tabela. Caso contrário, o gatilho verifica se a data de última modificação da tupla atualmente armazenada na tabela é anterior à data da nova tupla. Se sim, o gatilho entende que a nova tupla contém dados mais atualizados e substitui a tupla antiga. Nesse caso, os valores dos atributos *temporal_indexing* e *thematic_indexing* são alterados para false, para indicar que esse conjunto de dados deve passar por uma nova indexação temporal e temática.

O gatilho da tabela *resource* tem a mesma função. A única diferença é que os atributos que são atualizados para “false” são o *spatial_indexing*, *temporal_indexing*, *thematic_indexing* e *excluded*. O atributo *updated* também é atualizado, passando a guardar o valor “*true*”. Essas modificações são usadas para indicar que o recurso atualizado deve passar por uma nova indexação espacial, temporal e temática.

3.3.2 Esquema da base de dados espacial

A base de dados espacial, diferentemente da base de dados dos metadados, foi implementada utilizando-se do banco de dados de grafos Neo4j. A Figura 11 exibe a estrutura dos grafos armazenados no banco.

Figura 11 – Modelo lógico da base de dados espacial



Fonte: Elaborado pelo autor

Os grafos possuem nós chamados *Place*, que representam as divisões geográficas do Brasil: municípios, unidades federativas e regiões. Os nós rotulados como *Place* possuem os seguintes atributos: identificador único, nome do lugar, siglas do nome do

lugar (aplicado somente a unidades federativas) e o tipo de divisão geográfica que o lugar representa. Esses nós podem possuir, no grafo, relacionamento com outros nós do mesmo tipo. Esse relacionamento é chamado de "*_CONTAINS*" e representa que um lugar de um tipo X contém outro lugar do tipo Y. Por exemplo, a unidade federativa Paraíba possui um relacionamento "*_CONTAINS*" com cada um dos seus respectivos municípios. Todos os municípios, unidades federativas e regiões do Brasil foram armazenados no banco de dados, junto com os seus respectivos relacionamentos.

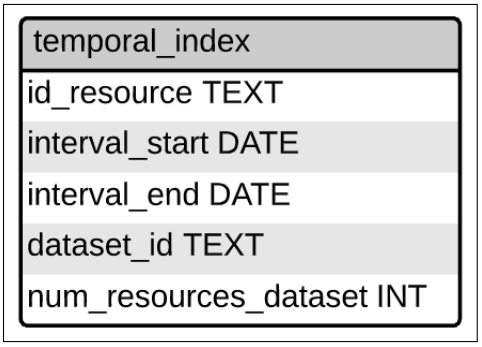
Os nós do tipo *Resource* representam os recursos que são oferecidos pelo portal de dados abertos. Eles possuem os seguintes atributos: o identificador único do recurso, o identificador único do conjunto de dados ao qual o recurso pertence, a quantidade de recursos que o conjunto de dados possui e o total de referências a localidades identificadas em seu conteúdo.

O conteúdo dos recursos pode ter diversas referências a nomes de lugares. Cada referência encontrada é representada no banco de dados por meio de um arco partindo de um nó do tipo *Resource* para um nó do tipo *Place*, indicando que o recurso contém referências a esse local. Esse tipo de relacionamento é chamado de *HAS_TERM*. O relacionamento *HAS_TERM* também possui atributos, são eles: a quantidade de referências que o recurso faz ao lugar e a frequência, que indica a relevância do lugar para o recurso. Essa frequência é calculada dividindo o valor do primeiro atributo descrito pelo total de lugares encontrados no recurso.

3.3.3 Esquema da base de dados temporal

A Figura 12 mostra o esquema lógico da base de dados temporal que foi implementada usando o SGBD PostgreSQL. O esquema é formado por uma única tabela, que guarda as informações sobre a extensão temporal dos recursos oferecidos pelo portal de dados abertos. Essas informações são extraídas durante o processo de indexação temporal. A extensão temporal de cada recurso é descrita por meio de um intervalo, que é composto pelas datas que representam o início e o fim do período de tempo ao qual o recurso se refere.

Figura 12 – Esquema lógico da base de dados temporal



Fonte: Elaborado pelo autor

A tabela usada para o armazenamento dos dados é chamada de *temporal_index*. Ela é formada pelos seguintes atributos: o identificador único, a data inicial do intervalo encontrado no processo de indexação, a data final do intervalo, o identificador único do conjunto de dados ao qual o recurso pertence e a quantidade de recursos (arquivos do formato CSV) ofertados pelo conjunto de dados. A base de dados possui também funções SQL úteis que calculam: a distância entre duas datas, a interseção, a diferença e a similaridade de dois intervalos de datas. A função que calcula a similaridade utiliza-se de todas as outras funções para realizar o cálculo.

3.3.4 Esquema da base de dados do *Solr*

Durante a implementação deste trabalho, a ferramenta *Solr* foi usada para a resolução de consultas temáticas. A escolha por essa ferramenta para a resolução das consultas temáticas se deu pelo fato de que ela é uma ferramenta de código aberto que consegue realizar consultas textuais em grandes bases de dados de documentos com boa qualidade e com bom desempenho (GRAINGER; POTTER, 2014).

Para este trabalho foram criados dois núcleos no *Solr*. Cada um desses núcleos possui um esquema que descreve como os dados serão armazenados e/ou indexados, tanto para os dados referentes aos recursos, quanto para os conjuntos de dados. O esquema do núcleo que guarda os índices dos conjuntos de dados foi mostrado na Figura 5 (seção 2.2). A estrutura definida no esquema possui atributos que representam: o identificador único do conjunto de dados, a junção de metadados do conjunto de dados e o número de versão do documento (sendo documento o tipo de estrutura que é armazenada no *Solr*). O atributo de metadados não será armazenado, somente indexado.

A Figura 13 mostra a declaração dos atributos da estrutura do núcleo *resource*. Os atributos declarados representam, respectivamente: o identificador único do recurso, o identificador único do conjunto de dados que contém o recurso, a junção dos metadados do recurso e o número de versão do documento. Assim, como no esquema do núcleo *dataset*, o único atributo que não será armazenado é o de metadados do recurso.

Para os dois esquemas, os tipos utilizados nos atributos das estruturas foram declarados, e são mostrados na Figura 6. Os detalhes do funcionamento da declaração dos tipos definidos para a implementação deste trabalho foram mostrados na seção 2.2.

Figura 13 – Definição de atributos do núcleo *resource*

```
<field name="id" type="string" indexed="true" stored="true" required="true"
    multiValued="false" />

<field name="package_id" type="string" indexed="true" stored="true"
    required="true" multiValued="false" />

<field name="metadata" type="text_general" indexed="true" stored="false"
    required="true" multiValued="false"/>

<field name="_version_" type="plong" indexed="true" stored="true"
    multiValued="false"/>

<uniqueKey>id</uniqueKey>
```

Fonte: Elaborado pelo autor

3.4 IMPLEMENTAÇÃO

Esta seção descreve o processo de implementação da ferramenta, destacando a implementação de cada módulo existente em sua arquitetura.

3.4.1 O processo de coleta dos dados

Essa seção descreve o fluxo do processo de coleta dos dados junto a um portal de dados abertos governamentais. A Figura 14 ilustra esse processo que é realizado em seis etapas:

1. O módulo de indexação dos conjuntos de dados executa o submódulo de recuperação e processamento de metadados;

2. O módulo de recuperação e processamento de metadados acessa um portal de dados e recupera todos os metadados dos conjuntos de dados oferecidos e seus respectivos recursos;

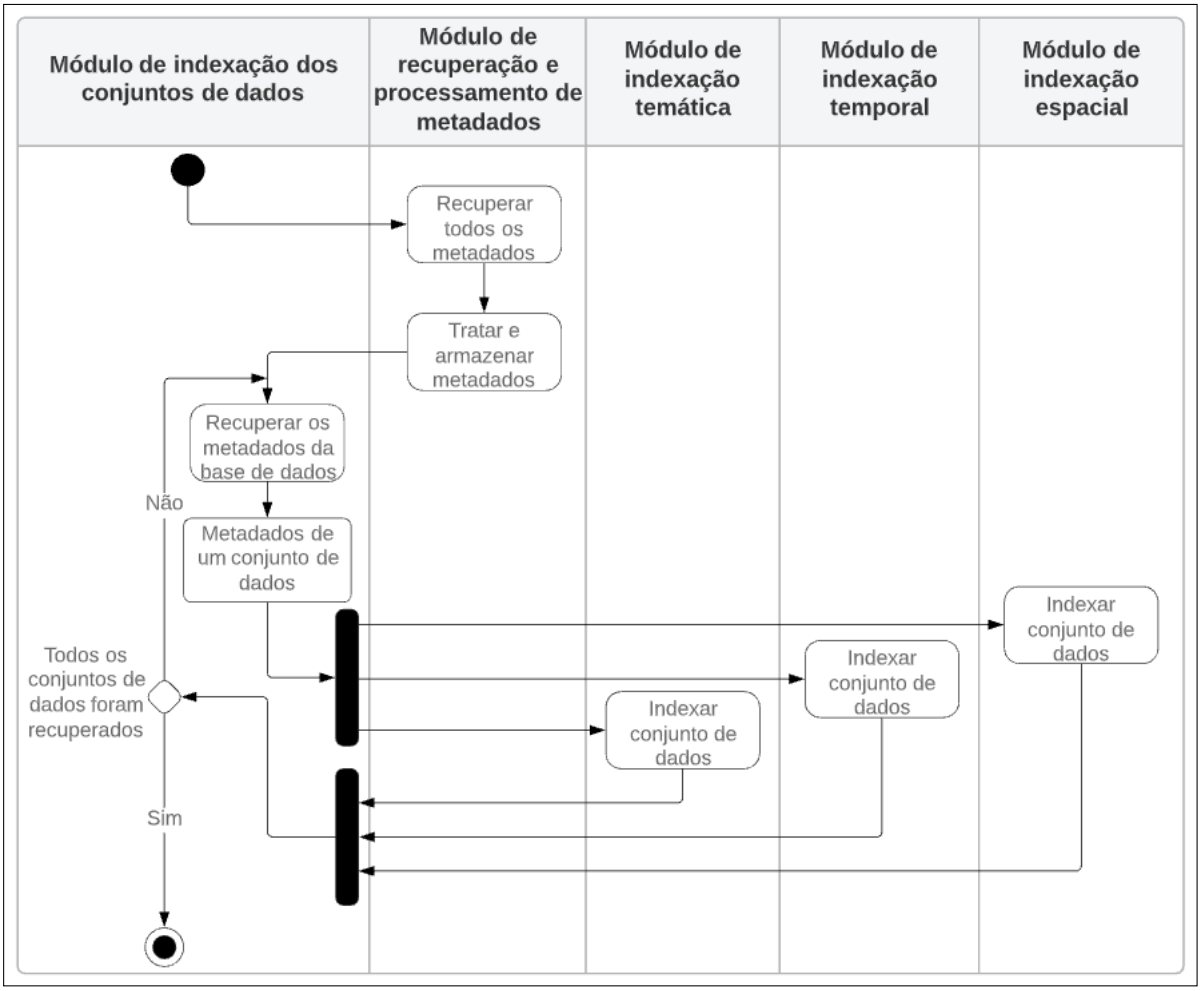
3. Os metadados recuperados são tratados e armazenados na base de dados dos metadados;

4. O módulo de indexação dos conjuntos acessa a base de dados dos metadados, para recuperar as informações acerca dos conjuntos de dados (e seus respectivos recursos) que ainda não foram indexados ou que estejam desatualizados;

5. Os dados sobre os conjuntos de dados selecionados na etapa 4 são passados como parâmetro para os submódulos de indexação espacial, temporal e temática. Este processo de recuperação e indexação se repete para todos os conjuntos de dados e seus recursos;

6. Cada um dos submódulos de indexação realiza a extração de novos metadados acerca dos conjuntos de dados e seus recursos, armazenando os resultados obtidos em suas respectivas bases de dados.

Figura 14 – Diagrama de atividades do principal fluxo da ferramenta



Fonte: Elaborado pelo autor

3.4.2 Módulo de indexação dos conjuntos de dados

O módulo de indexação dos conjuntos de dados é responsável por realizar o processo de obtenção e atualização dos metadados junto ao portal de dados abertos usado pela ferramenta. Ele é responsável por coordenar todo o processo de coleta, armazenamento e atualização dos dados, acerca dos conjuntos de dados e recursos encontrados. Além disso, esse módulo realiza testes de conexão com o *Solr* e os diversos bancos de dados usados pela ferramenta, e verifica se os dados, aqueles que devem estar pré-armazenados, de fato, existem nas bases de dados.

3.4.2.1 O módulo de recuperação e processamento de metadados

O módulo de recuperação e processamento de metadados é responsável por recuperar todos os metadados dos conjuntos de dados e recursos oferecidos por algum portal de dados abertos. Atualmente, o módulo consegue coletar os dados de portais que são implementados por meio da plataforma CKAN. As URLs dos portais que devem ser processados são fornecidas manualmente.

Para cada portal de dados, o módulo realiza o download dos metadados de todos os seus conjuntos de dados. Esse processo é feito por meio de várias requisições, sendo que em cada requisição é possível baixar os dados em lotes de no máximo mil conjuntos de dados. Assim, caso um portal tenha mais de mil conjuntos de dados, são necessárias várias requisições para se obter todos os dados disponibilizados. Para cada requisição, o portal retorna um arquivo no formato JSON contendo informações acerca dos conjuntos de dados e seus respectivos recursos.

Os metadados de cada recurso e conjunto de dados são processados e todas as informações relevantes para a ferramenta são filtradas e armazenadas na base de dados de metadados. Para a implementação desse módulo foi utilizado o *framework python ckanapi*[7], que atua como um intermediador do acesso à API *Action*[8] do CKAN. O *framework* possui uma interface de acesso à API em uma instância remota do CKAN. Para realizar o acesso é necessário apenas informar a URL da instância CKAN que deve ser acessada. A Figura 15 mostra um exemplo destacando como a API pode ser configurada para acessar os dados do portal brasileiro de dados abertos, que foi usado no desenvolvimento deste trabalho.

Figura 15 – Instância da interface de acesso a API do CKAN

```
interface = RemoteCKAN('http://dados.gov.br/')
```

Fonte: Elaborado pelo autor

Por meio da interface mostrada na Figura 15, pode-se acessar o atributo "*action*" que oferece, dentre outras, uma funcionalidade de recuperação dos grupos de metadados (cada grupo está associado a um conjunto de dados específico) publicados no

[7] "CKAN API". Disponível em: https://github.com/ckan/ckanapi. Acessado em 11 Fev. 2020

[8] "API guide". Disponível em: https://github.com/ckan/ckan/blob/master/doc/api/index.rst. Acessado em 11 Fev. 2020

portal que está sendo acessado. Nessa funcionalidade, são passados como parâmetros a quantidade limite dos grupos de metadados que devem ser recuperados, e um parâmetro que indica a partir de qual conjunto de dados se deseja consultar os metadados.

A Figura 16 exibe a instância da interface de acesso sendo atribuída à variável "*interface*". Depois, a variável "*api_action*" recebe o atributo da interface, que permite acessar as funcionalidades da *API Action*. Por último, a função de recuperação dos metadados é invocada. Na requisição ilustrada na figura, estão sendo solicitados os metadados de até mil *datasets*, sendo o primeiro grupo de metadados referente ao conjunto de dados que está depois da milésima posição.

Figura 16 – Funcionalidade de recuperação de metadados

```
interface = RemoteCKAN('http://dados.gov.br/')

api_action = interface.action

metadata = api_action.current_package_list_with_resources(limit=1000, offset=1000)
```

Fonte: Elaborado pelo autor

3.4.2.2 O módulo de indexação espacial

Como o nome já indica, o módulo de indexação espacial realiza o processo de identificar e armazenar metadados referentes à extensão espacial dos recursos oferecidos pelo portal. Antes de começar o processo de indexação, ele realiza o download dos arquivos contendo os dados de cada recurso, a fim de que seja feita a identificação de metadados adicionais que serão usados pela ferramenta.

Primeiramente é feita uma requisição HTTP para a URL do recurso. Caso o recurso requisitado esteja disponível, o seu conteúdo é retornado. Ao receber o resultado, o módulo primeiro verifica o tipo de arquivo que foi retornado. Essa ação é necessária porque, embora os metadados indiquem que conteúdo da URL é um arquivo no formato CSV, é possível que seja retornado como resposta da requisição conteúdos do tipo HTML, XML, CSS, dentre outros. Quando isso acontece, o conteúdo é descartado e o recurso não é processado. Caso seja realmente retornado um arquivo CSV, é realizado o processo de indexação espacial.

Durante o processo de indexação, o módulo procura, em cada linha do arquivo CSV referente ao recurso que está sendo processado, colunas cujos valores contenham referências a nomes de lugares. Mais especificamente, ele procura por referências a nomes de municípios, unidades federativas e regiões do Brasil. Para realizar essa tarefa, o módulo verifica os valores de cada coluna do arquivo CSV em busca desse tipo de informação. Então, é correto dizer que, o que é feito a princípio é uma preparação para de fato começar a indexação.

Ao fim dessa tarefa podem acontecer três situações:

1. Caso não seja encontrada nenhuma coluna com referências a nomes de lugares, o processo se encerra sem a identificação de metadados espaciais;
2. Caso seja encontrada uma coluna com referência a nomes de lugares, ela é usada para fazer a geocodificação das linhas do arquivo;
3. Caso seja encontrada mais de uma coluna com referência a nomes de lugares, a geocodificação é realizada com base na coluna mais específica, uma vez que ela descreve a localização com maior precisão.

A Figura 17 descreve um exemplo de um arquivo CSV hipotético. Ao se analisar as colunas do arquivo, nota-se que existem duas colunas, chamadas lugar1 e lugar2, que armazenam, respectivamente, o nome do município e da unidade federativa de cada linha de dados que existe no arquivo. Nesse caso, o módulo vai geocodificar cada linha do arquivo com base no valor da coluna lugar1, que oferece a informação de localização de forma mais precisa.

Figura 17 – Valores de um arquivo CSV fictício

id	lugar1	lugar2
123	São Paulo	São Paulo
213	Itaporanga	Paraíba
...	...	...

Fonte: Elaborado pelo autor

Na primeira linha de dados do arquivo mostrado na Figura 17, o valor São Paulo aparece em duas colunas distintas. Como esse valor pode representar o nome de um município ou de um estado brasileiro (ou nenhum deles), o módulo precisa analisar outras linhas de dados para eliminar essa ambiguidade. Ao se analisar a segunda linha, percebe-se que essas colunas possuem, respectivamente, os valores Itaporanga (que é o nome de uma cidade) e Paraíba (que é o nome de uma unidade federativa). Com essa verificação, o módulo percebe o tipo de divisão administrativa correspondente a cada coluna e, com base nesse critério, escolhe a coluna que ofereça a informação de forma mais precisa. Para garantir que a coluna de um arquivo realmente possui nomes de lugares, a verificação é feita para uma determinada quantidade de linhas do CSV.

Por fim, as referências aos nomes de lugares e as informações do recurso são armazenadas na base de dados espacial (seção 3.3.2).

3.4.2.3 O módulo de indexação temporal

O módulo de indexação temporal realiza o processo de identificar metadados que descrevem a extensão temporal dos recursos oferecidos por um portal. Durante esse processo, o módulo procura nos metadados valores que representam expressões temporais, tais como datas, ou expressões como bimestre, primeiro mês, semestre, entre outras.

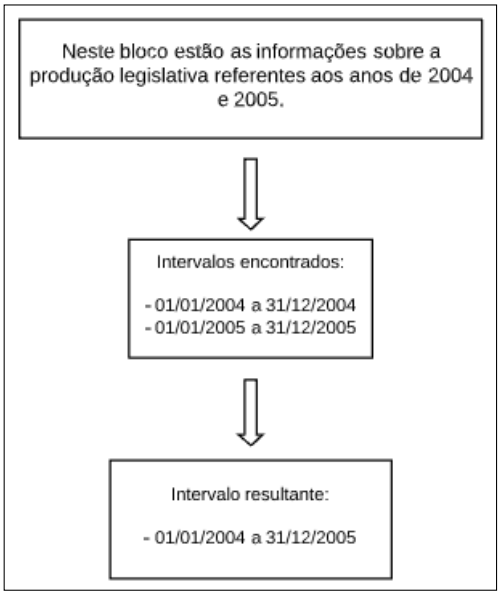
As expressões são procuradas em metadados pré-escolhidos e em uma ordem de prioridade predeterminada. Os metadados escolhidos, em ordem de prioridade, foram: o nome do recurso, a descrição do recurso, o título do conjunto de dados do qual o recurso faz parte, a descrição do conjunto de dados e a data de criação do recurso. Quando a busca encontra uma expressão temporal, ela é finalizada e não prossegue para os metadados subsequentes.

Para que seja feita a identificação de expressões temporais dos metadados foi desenvolvida, para este trabalho, uma expressão regular ou *regex*, que é uma forma de prover a identificação de conjuntos de caracteres com um determinado padrão. Neste trabalho, o *regex* identifica diversos tipos de formatos de datas, desde as mais simples, como as que contém somente o ano, quanto as mais complexas, que possuem traços, barras, o nome ou abreviação do mês e palavras como semestre, trimestre e bimestre.

As expressões encontradas são transformadas em intervalos, que possuem uma data inicial e uma data final. Esse intervalo indica a extensão temporal que o recurso representa. A Figura 18 exemplifica a extração de informações temporais, a partir da descrição de um recurso. Nos metadados existem duas expressões temporais,

que representam os anos 2004 e 2005. Elas são transformadas em um intervalo que abrange desde o primeiro dia do ano até o último dia. O intervalo resultante, que representa a extensão temporal do recurso, é dado pela menor data inicial encontrada nos intervalos e a maior data final encontrada nos intervalos.

Figura 18 – Exemplo de extração do intervalo de data



Fonte: Elaborado pelo autor

No fim do processo de indexação, o identificador único do recurso que está sendo processado, o identificador único do conjunto de dados do qual o recurso faz parte e o intervalo identificado a partir da análise dos metadados são armazenados no banco de dados juntamente com a quantidade de recursos que o conjunto de dados possui. Esse último valor será útil no processo de resolução de consultas temporais, que é descrito na seção 3.4.3.

3.4.2.4 O módulo de indexação temática

O módulo de indexação temática realiza o processo de indexação temática. Durante esse processo, ele usa a ferramenta *Solr* para realizar a indexação dos recursos e dos conjuntos de dados. Esse é o único dos módulos que realiza a indexação tanto do recurso, quanto do conjunto de dados.

Para o processo de indexação de um recurso é criado um documento de texto contendo uma parte dos seus metadados (nome e descrição) e alguns metadados do

seu respectivo conjunto de dados (título, descrição e palavras-chaves). Esse documento é enviado para o *Solr* para armazenamento em sua base de dados. O documento é associado ao identificador único do recurso. O identificador único do conjunto de dados também é armazenado, com o intuito de facilitar o processo de atualização dos índices.

Para o processo de indexação de um conjunto de dados também é criado um documento de texto, contendo os metadados do conjunto de dados e de todos os seus recursos. Esse documento é submetido ao *Solr* para armazenamento em sua base de dados. Tal documento é associado ao identificador único do conjunto de dados.

Antes de serem armazenados, os documentos são processados pelo *Solr*. Esse processamento é feito em várias etapas. Primeiro, eles são submetidos a um processo de tokenização. Depois, são aplicados filtros para: dividir as palavras com delimitadores, mudar as letras das palavras para minúsculo, remover palavras indesejadas, identificar a classificação gramatical da palavra, trocar as palavras pelos seus respectivos lemas e substituir os caracteres pelos seus equivalentes da tabela ASCII.

3.4.3 O módulo de consulta

O módulo de consulta é uma API REST que disponibiliza dois serviços. O primeiro deles executa uma consulta de acordo com os parâmetros passados pelo usuário e retorna os identificadores únicos dos recursos ou dos conjuntos de dados selecionados. O outro serviço da API recebe um conjunto de identificadores únicos (de conjuntos de dados ou de recursos) e retorna os metadados referentes a esses identificadores.

A resolução de uma consulta pode ser dividida em até três subconsultas. São elas: espacial, temporal e temática. Essas subconsultas são executadas em paralelo, com o intuito de melhorar o desempenho, e podem ser executadas em nível de recurso ou de conjunto de dados.

Para a consulta espacial em nível de recurso, é passado como parâmetro o nome do lugar de interesse do usuário. Esse nome pode se referir a um município, unidade federativa ou região. Caso haja mais de um lugar com o mesmo nome fornecido, são retornadas informações referentes aos lugares encontrados, para que o usuário possa distinguir o lugar que ele deseja usar como critério de busca. Feito isso, o identificador único do local escolhido é usado como parâmetro para a consulta. Depois disso, a consulta é executada em quatro etapas:

1. O módulo localiza no banco de dados espacial, o nó que representa o lugar

específico informado pelo usuário;

2. O módulo identifica todos os nós que representam um recurso e que possuem um relacionamento do tipo *HAS_TERM* com o nó do lugar especificado. O objetivo dessa etapa consiste em localizar todos os recursos que possuem alguma linha de dados que faça referência ao local escolhido;

3. O módulo repete o processo para cada lugar que está contido no lugar especificado. Por exemplo, se a busca foi feita para a região Nordeste do Brasil, ela será automaticamente estendida para todas as unidades federativas e municípios dessa região;

4. O módulo recupera os identificadores únicos dos recursos selecionados, juntamente com o valor que representa a frequência que o local apareceu em cada recurso. Ainda usando o exemplo de uma consulta para a região Nordeste, todos os recursos associados às unidades federativas e municípios da região terão o valor da frequência multiplicado por 0,1. Assim, os recursos que estão diretamente associados ao nó da região Nordeste tenderão a possuir o valor da frequência maior, portanto maior relevância.

Na consulta espacial em nível de conjunto de dados, as mesmas etapas para a consulta em nível de recurso são realizadas, com exceção da última etapa. Para esse nível de consulta, os identificadores únicos dos conjuntos de dados são retornados juntamente com a frequência, que agora é calculada por meio da quantidade de recursos de um mesmo conjunto de dados que foram encontrados na consulta, dividida pelo total de recursos, do formato CSV, que o conjunto de dados possui. O objetivo dessa métrica consiste em priorizar os conjuntos de dados, nos quais o local escolhido pelo usuário possui maior relevância.

Para realizar uma consulta temporal em nível de recursos, o usuário precisa informar o período de tempo do seu interesse. Esse período é representado como um intervalo de tempo, que é composto por uma data inicial e uma data final. Após o recebimento desses valores, o módulo realiza uma verificação para saber se o intervalo é válido, comparando se a data do início é menor que a data do fim. Depois, o módulo executa uma consulta no banco de dados para recuperar os identificadores únicos dos recursos cuja extensão temporal tem alguma similaridade com o intervalo especificado na consulta.

Terminada essa etapa, o módulo analisa a similaridade da extensão temporal de cada recurso selecionado e o intervalo selecionado na consulta. O objetivo dessa

etapa consiste em priorizar os recursos, cuja extensão temporal é mais parecida com o intervalo requisitado.

Para verificar essa similaridade, foi utilizada a equação de Tversky (TVERSKY, 1977), que é mostrada na Equação 1. A equação considera, para o cálculo da similaridade entre dois objetos, tanto as características que eles têm em comum quanto as suas diferenças. Nos cálculos da intersecção e da diferença é retornado um intervalo representado por uma data inicial e uma data final. A função *f* calcula a distância entre essas duas datas e o resultado é dado em meses. As constantes α e β são pesos atribuídos à equação. O resultado da equação da similaridade é um valor entre 0 e 1. O valor 1 indica que os dois intervalos são idênticos, enquanto que o valor 0 indica que não há qualquer interseção entre os intervalos.

$$Tversky(A, B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha * f(A - B) + \beta * f(B - A)} \tag{1}$$

A Figura 19 mostra uma tabela contendo alguns exemplos do cálculo da similaridade temporal entre vários intervalos de tempo com o período de 01/01/2015 a 31/12/2015, que foi usado como exemplo. As colunas representam, respectivamente, a data inicial do intervalo, a data final do intervalo e o valor da similaridade entre o intervalo representado com o intervalo usado como exemplo. O valor 0,5 foi usado para as constantes α e β.

Figura 19 – Exemplos de cálculos da similaridade.

intervalo início	intervalo fim	similaridade
01/01/2015	31/12/2015	1.0
01/01/2015	30/06/2015	0.6666666666666666
01/10/2015	31/12/2015	0.4
01/12/2015	31/12/2015	0.15384615384615385
01/01/2011	31/12/2016	0.2857142857142857
11/11/2003	05/02/2020	0.11603223117532648
01/01/1993	31/12/2018	0.07407407407407407

Fonte: Elaborado pelo autor

Após a conclusão da consulta tem-se como resultado os identificadores únicos dos recursos selecionados, juntamente com o valor do cálculo da similaridade para cada recurso encontrado.

Para a consulta temporal em nível de conjunto de dados é realizado o mesmo procedimento que a consulta em nível de recurso. Entretanto, nesse tipo de consulta são retornados os identificadores únicos dos conjuntos de dados, que contêm os recursos encontrados e um valor de ranking. Para cada conjunto de dados, esse ranking é calculado através da quantidade de recursos que possuem similaridade maior que zero, dividido pela quantidade de recursos que o conjunto de dados possui. O objetivo dessa métrica consiste em priorizar os conjuntos de dados, que possuem mais recursos relacionados ao período de tempo solicitado pelo usuário.

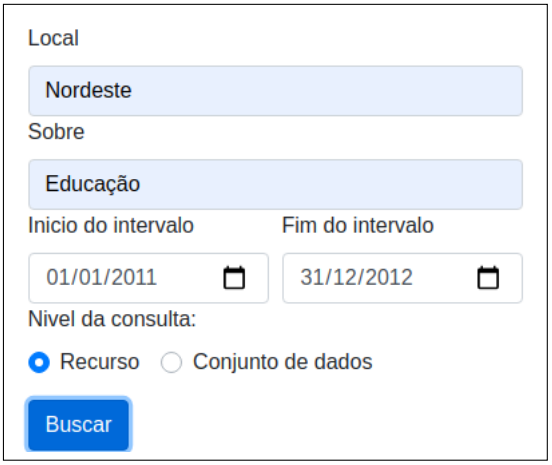
Para a consulta temática, o mesmo processo é usado para a resolução de consultas, tanto em nível de recurso, quanto em nível de conjunto de dados. A única diferença é que elas são realizadas em núcleos diferentes do banco de dados da ferramenta *Solr*. O texto que será usado na consulta, é recebido como parâmetro e é passado para a função de busca do *Solr*. Durante esse processo, também são informados quais atributos devem ser retornados para os itens selecionados. O texto usado como critério de busca passa por uma série de filtros que irão: dividir as palavras com delimitadores, mudar as letras das palavras para minúsculo, remover palavras indesejadas, buscar os sinônimos das palavras, identificar a classificação gramatical das palavras, trocar as palavras pelos seus respectivos lemas e substituir os caracteres pelos seus equivalentes da tabela ASCII. Por fim, a função de busca é executada em um dos núcleos do *Solr*, retornando os identificadores únicos dos recursos ou dos conjuntos de dados e um valor de ranking para cada item selecionado. Esse ranking é gerado automaticamente pelo motor de busca fornecido pelo *Solr*.

Finalmente, a ferramenta também permite a realização de consultas multidimensionais, que consistem em consultas que possuem mais de um tipo de restrição (espacial, temporal ou temática). Nesses casos, as consultas são divididas em subconsultas, sendo gerada uma consulta para cada restrição especificada pelo usuário. Essas subconsultas são enviadas para os módulos apropriados e são executadas em paralelo. Os resultados obtidos para cada subconsulta são analisados. Nessa etapa, são selecionados apenas os itens (conjunto de dados ou recursos), que aparecem nos resultados de todas as consultas realizadas. Após a seleção, a média aritmética dos valores de ranking retornados por cada consulta é utilizada para calcular o ranking dos resultados encontrados. Por fim, os resultados da consulta são ordenados e retornados para o usuário.

3.4.4 Módulo de visão web

O módulo de visão web oferece uma interface a partir da qual o usuário pode interagir com o módulo de consulta de forma indireta. A Figura 20 mostra os campos que podem ser preenchidos, para especificar os valores que devem ser usados como critério de busca para cada dimensão (espacial, temática, ou temporal). O usuário também pode escolher o nível da consulta, que pode ser recurso ou conjunto de dados. No exemplo mostrado na Figura 20, a consulta será multidimensional e buscará recursos que possuam dados sobre educação na região Nordeste no intervalo de 2011 a 2012.

Figura 20 – Parte da interface onde o usuário faz a consulta.

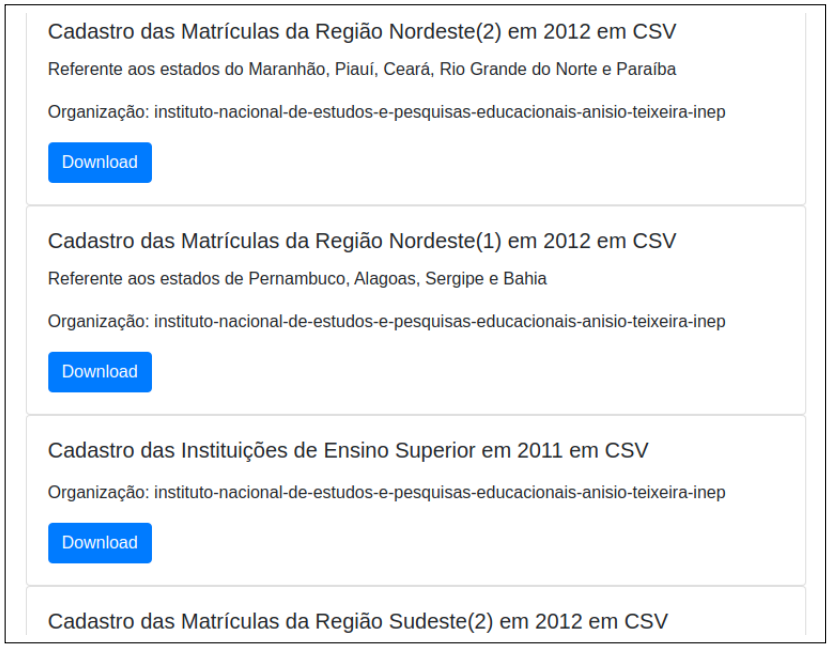


Fonte: Elaborado pelo autor

Quando a consulta é submetida, o módulo de visão usa os valores definidos pelo usuário como parâmetros de uma requisição ao módulo de consulta. No caso do exemplo da Figura 20, será realizada uma consulta espacial por dados sobre a região Nordeste, uma consulta temporal por dados do período entre 2011 e 2012, e uma consulta temática por dados sobre educação. O resultado da requisição será composto por todos os identificadores únicos dos recursos, que satisfazem todos os critérios definidos na consulta. Após o recebimento do resultado da consulta, o módulo de visão faz outra requisição para o módulo de consulta, agora, passando os identificadores únicos como parâmetro. O resultado dessa nova requisição são os metadados dos recursos, que são apresentados ao usuário em uma lista. Parte dessa lista é apresentada pela Figura 21. O usuário pode analisar os resultados retornados e escolher aquele que mais o interessa clicando no botão “Download”. Quando isso

acontece, a ferramenta o redireciona para a URL a partir da qual o arquivo pode ser obtido.

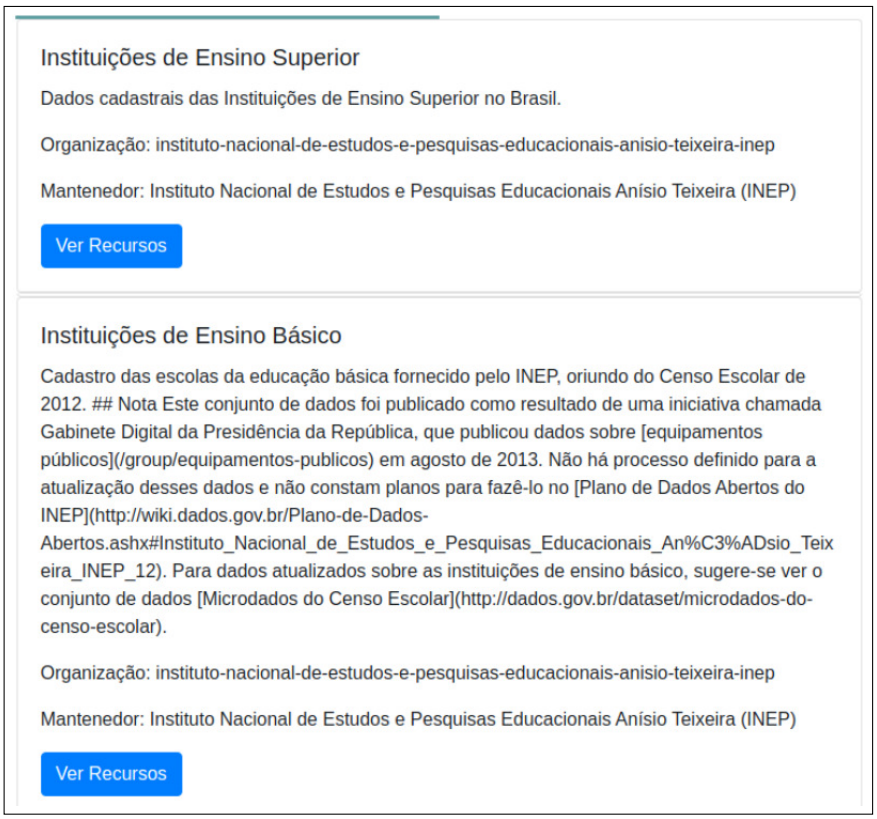
Figura 21 – Parte da lista de recursos.



Fonte: Elaborado pelo autor

Para uma consulta em nível de conjunto de dados contendo os mesmos parâmetros de consulta mostrados pela Figura 20, o resultado será composto pelos identificadores únicos dos conjuntos de dados. Ao receber esse resultado, assim como acontece nas consultas em nível de recursos, o módulo de visão realiza uma nova requisição para recuperar os metadados sobre os conjuntos de dados que foram selecionados na consulta. A Figura 22 mostra a exibição na interface gráfica de parte do resultado da consulta.

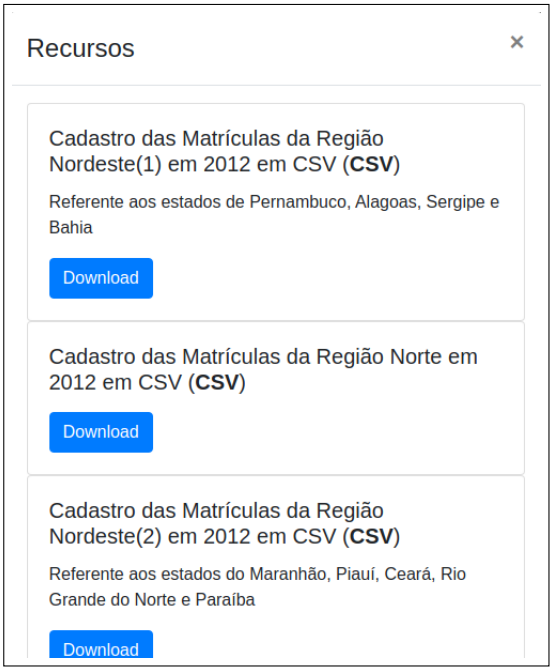
Figura 22 – Parte da lista de conjuntos de dados



Fonte: Elaborado pelo autor

Ao clicar no botão “Ver Recursos”, é exibida uma janela com uma lista de informações sobre os recursos pertencentes ao conjunto de dados. A Figura 23 mostra alguns dos recursos pertencentes ao conjunto de dados exibido na Figura 22. Um conjunto de dados pode apresentar arquivos de diversos formatos, não somente CSV.

Figura 23 – Janela com alguns dos recursos do conjunto de dados.



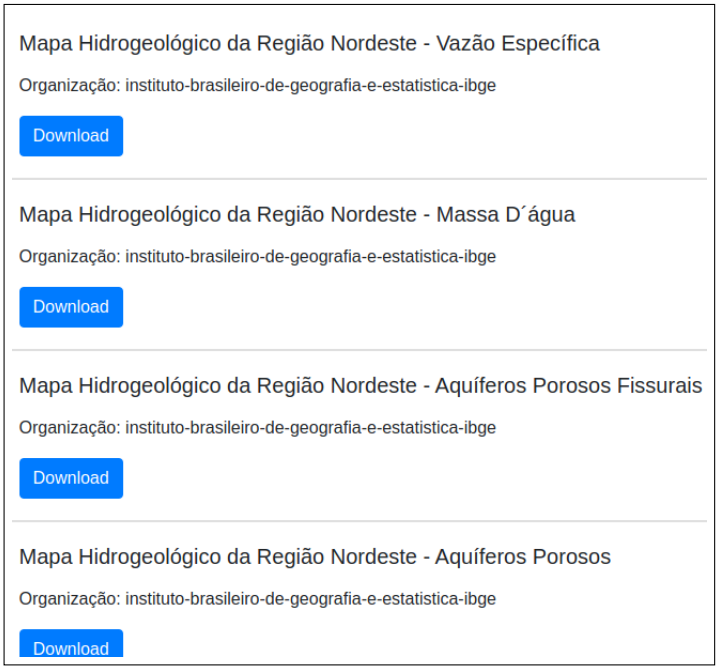
Fonte: Elaborado pelo autor

O módulo de visão permite também a paginação dos resultados. Para cada página é feita uma nova requisição. Assim, não é necessário armazenar em memória todas as informações relacionadas aos recursos ou conjuntos de dados, mas somente os valores que serão exibidos por página. Atualmente, são exibidos no máximo trinta resultados em cada página.

Consultas com uma única dimensão também podem ser realizadas, desde que seja passado o valor do parâmetro relacionado ao tipo de consulta que se deseja realizar. Os valores dos parâmetros das demais dimensões podem ser omitidos.

A Figura 24 mostra parte do resultado de uma consulta espacial. Nela foi realizada uma busca por todos os recursos sobre o município Itaporanga na Paraíba.

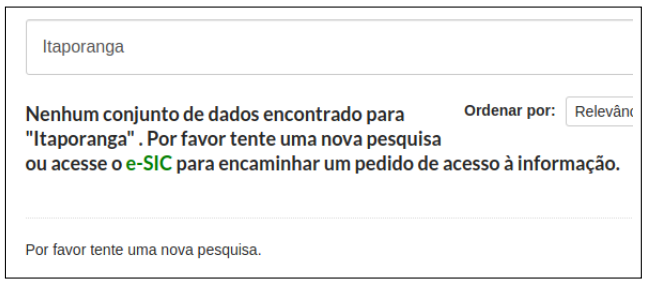
Figura 24 – Parte do resultado da consulta espacial.



Fonte: Elaborado pelo autor

A Figura 25 exibe uma consulta por “Itaporanga” no portal brasileiro de dados abertos. A consulta não retorna resultados pois, diferente da ferramenta proposta neste trabalho, a indexação não é feita utilizando os próprios dados, mas, somente os metadados. Nenhum dos conjuntos de dados possuem metadados com a palavra Itaporanga descrita neles, por isso, a ferramenta não consegue associar os conjuntos de dados à palavra-chave da consulta. Outra diferença importante é que a ferramenta proposta pôde resolver a consulta a nível de recurso facilitando que o usuário encontre mais facilmente um recurso específico.

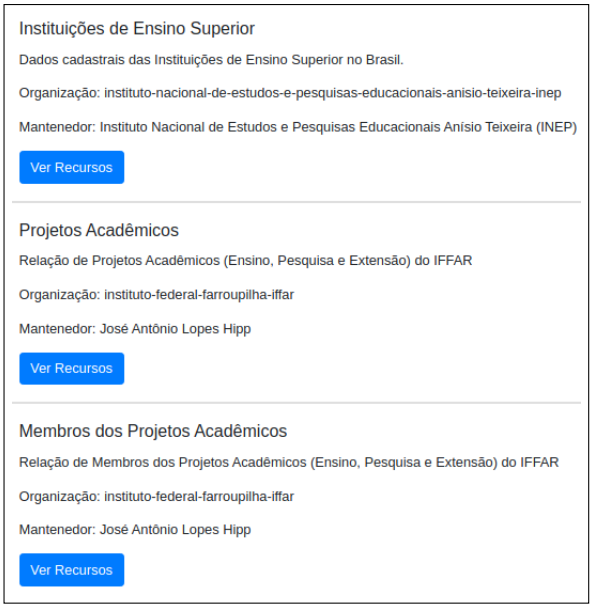
Figura 25 – Consulta em portal brasileiro de dados abertos



Fonte: Elaborado pelo autor

A Figura 26 mostra parte do resultado de uma consulta temática, na qual foram requisitados os conjuntos de dados que contêm dados sobre educação.

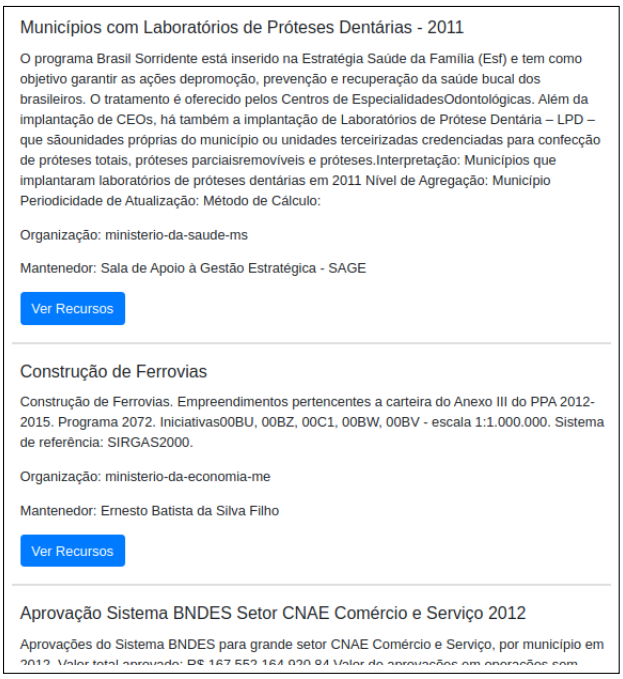
Figura 26 – Parte do resultado da consulta temática.



Fonte: Elaborado pelo autor

A Figura 27 mostra parte do resultado de uma consulta temporal. Nela, foram requisitados todos os conjuntos de dados que contenham dados do período de 2011 a 2012.

Figura 27 – Parte do resultado da consulta temporal.



Fonte: Elaborado pelo autor

4 CONSIDERAÇÕES FINAIS

Os portais de dados abertos governamentais têm o objetivo de tornar acessíveis, aos cidadãos, os conjuntos de dados gerados por órgãos públicos. Para que os recursos disponibilizados sejam acessados, o cidadão deve recorrer ao motor de busca do portal. Entretanto, nem sempre os resultados das pesquisas feitas pelo usuário são satisfatórios, pois a indexação dos conjuntos de dados é feita com base em metadados que, muitas vezes, não descrevem os dados com exatidão, tornando difícil a localização dos conjuntos de dados que são realmente relevantes para o usuário.

Com o objetivo de resolver essas limitações, este TCC propôs um motor de busca para facilitar a localização de dados abertos governamentais. O desenvolvimento do trabalho teve foco na conclusão de cinco objetivos específicos, que foram descritos na seção 1.2.2.

No primeiro objetivo, foi estudado com qual ferramenta os portais de dados governamentais abertos do Brasil são criados. O estudo resultou no entendimento de que o portal do Brasil e de vários outros governos são instâncias da ferramenta CKAN, na qual os dados disponibilizados são tratados como recursos, que estão contidos em conjuntos de dados pertencentes a organizações. Os órgãos públicos acessam a instância da ferramenta correspondente ao seu governo e lá criam suas próprias organizações e descrevem os seus conjuntos de dados.

No segundo objetivo, foi feito um estudo com o objetivo de compreender como funciona o processo de resolução de consultas nos motores de busca atuais. Durante esse processo, entendeu-se que, para realizar uma pesquisa, o usuário precisa informar um conjunto de palavras-chaves, que são utilizadas na recuperação dos conjuntos de dados que apresentem estas palavras em sua descrição.

Para alcançar o terceiro objetivo, foi implementado um módulo que interage com o portal de dados abertos e extrai os metadados relacionados às organizações, conjuntos de dados e recursos, armazenando estas informações em um banco de dados local.

Para alcançar o quarto objetivo, um banco de dados foi criado para ser usado no processo de indexação por termos que representam nomes de lugares (municípios, unidades federativas, regiões). Os termos são identificados no conteúdo de cada recurso ofertado pelo portal. Outro banco de dados foi criado, e é utilizado no processo

de indexação por termos que representam intervalo de datas. Os termos são obtidos a partir da análise dos metadados dos recursos e conjuntos de dados que estão sendo indexados. Esse banco de dados é usado para a resolução de consultas com restrições temporais. Por fim, foi utilizado o *Solr* para o processo de indexação temática. Os termos para a indexação são extraídos também dos metadados dos recursos e conjuntos de dados que serão indexados.

Finalmente, a solução desenvolvida foi aplicada ao portal de dados abertos do governo federal do Brasil, sendo indexados com sucesso, para teste, alguns de seus recursos. O módulo para resolução de consultas com restrição espacial, temporal e temática foi finalizado, bem como a interface que permite que o usuário realize consultas e visualize os seus respectivos resultados.

Alguns trabalhos futuros podem ser feitos para aprimorar a ferramenta atualmente desenvolvida. Um possível trabalho consiste em permitir que a consulta espacial não fique restrita somente a buscas de um único lugar específico, com a possibilidade de se resolver buscas por vários lugares na mesma consulta. Outro trabalho de aprimoramento seria o de criar um dicionário de palavras e sinônimos com base numa análise detalhada das palavras e dos temas abordados nas descrições dos recursos e conjuntos de dados dos portais de dados abertos do Brasil. Assim, o dicionário gerado seria utilizado para consultas temáticas mais relevantes.

REFERÊNCIAS

APACHE SOFTWARE FOUNDATION. **Apache Solr Reference Guide**. 2019. Disponível em: <http://archive.apache.org/dist/lucene/solr/ref-guide/apache-solr-ref-guide-8.1.pdf>. Acesso em: 01 ago. 2020.

BRASIL. **Regula o acesso a informações previsto no inciso XXXIII do art. 5o, no inciso II do § 3o do art. 37 e no § 2o do art. 216 da Constituição Federal; altera a Lei no 8.112, de 11 de dezembro de 1990; revoga a Lei no 11.111, de 5 de maio de 2005, e dispositivos da Lei no 8.159, de 8 de janeiro de 1991; e dá outras providências.** Brasília: LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011, 2011.

BRICKLEY, D.; BURGESS, M.; NOY, N. Google dataset search: Building a search engine for datasets in an open web ecosystem. In: **The World Wide Web Conference**. [S.l.: s.n.], 2019. p. 1365–1375.

COMPREHENSIVE KNOWLEDGE ARCHIVE NETWORK. **User guide**. 2020. Disponível em: <https://docs.ckan.org/en/latest/user-guide.html>. Acesso em: 04 fev. 2020.

GRAINGER, T.; POTTER, T. **Solr in action**. [S.l.]: Manning, 2014.

KACPRZAK, E.; KOESTEN, L.; IBÁÑEZ, L.-D.; BLOUNT, T.; TENNISON, J.; SIMPERL, E. Characterising dataset search—an analysis of search logs and data requests. **Journal of Web Semantics**, Elsevier, v. 55, p. 37–55, 2019.

MÁRQUEZ, J.; CÓRCOLES, J.; QUINTANILLA, A. A semantic index structure for integrating ogc services in a spatial search engine. In: IEEE. **2010 IEEE Conference on Open Systems (ICOS 2010)**. [S.l.], 2010. p. 103–108.

OPEN KNOWLEDGE FOUNDATION. **Definição de Conhecimento Aberto**. 2019. Disponível em: <http://opendefinition.org/od/2.0/pt-br/>. Acesso em: 04 fev. 2020.

STRÓŻYNA, M.; EIDEN, G.; ABRAMOWICZ, W.; FILIPIAK, D.; MAŁYSZKO, J.; WĘCEL, K. A framework for the quality-based selection and retrieval of open data-a use case from the maritime domain. **Electronic Markets**, Springer, v. 28, n. 2, p. 219–233, 2018.

THE WHITE HOUSE. **Executive Order Making Open and Machine Readable the New Default for Government Information**. 2013. Disponível em: <https://obamawhitehouse.archives.gov/the-press-office/2013/05/09/executive-order-making-open-and-machine-readable-new-default-government->. Acesso em: 04 fev. 2020.

TOMAN, M.; TESAR, R.; JEZEK, K. Influence of word normalization on text classification. **Proceedings of InSciT**, v. 4, p. 354–358, 2006.

TRIBUNAL DE CONTAS DA UNIÃO. **5 motivos para a abertura de dados na administração pública**. 2015. Disponível em: <https://portal.tcu.gov.br/biblioteca-digital/cinco-motivos-para-a-abertura-de-dados-na-administracao-publica.htm>. Acesso em: 04 fev. 2020.

TVERSKY, A. Features of similarity. **Psychological review**, American Psychological Association, v. 84, n. 4, p. 327, 1977.

UBALDI, B. Open government data: Towards empirical analysis of open government data initiatives. OECD, p. 4, 2013.

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

Documento Digitalizado Ostensivo (Público)

TCC - Motor de Busca para Dados Abertos - Ian Carneiro Teixeira de Araújo

Assunto: TCC - Motor de Busca para Dados Abertos - Ian Carneiro Teixeira de Araújo
Assinado por: Ian Araújo
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Ian Carneiro Teixeira de Araújo**, **ALUNO (201712010016) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 03/03/2021 15:48:29.

Este documento foi armazenado no SUAP em 03/03/2021. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 183022
Código de Autenticação: 6cce472c3c