CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Pablo Henrique de Sousa

PSIACADEMIC ANALYTICS: Desenvolvimento de um Módulo de Extração e Consulta de Dados sobre Saúde Mental de dissertações e teses acadêmicas

Palmas – TO

2022

Pablo Henrique de Sousa

PSIACADEMIC ANALYTICS: Desenvolvimento de um Módulo de Extração e Consulta de Dados sobre Saúde Mental de dissertações e teses acadêmicas

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Profª. Dra. Parcilene Fernandes de Brito.

Palmas – TO

2022

Pablo Henrique de Sousa

PSIACADEMIC ANALYTICS: Desenvolvimento de um Módulo de Extração e Consulta de Dados sobre Saúde Mental de dissertações e teses acadêmicas

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Profª. Dra. Parcilene Fernandes de Brito.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Profª. Dra. Parcilene Fernandes de Brito

Orientadora

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M.e. Jackson Gomes de Souza

Centro Universitário Luterano de Palmas – CEULP

_____

Profª. Dra. Irenides Teixeira

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2022

RESUMO

[1]DE SOUSA, Pablo Henrique. **PSIACADEMIC ANALYTICS: Desenvolvimento de um Módulo de Extração e Consulta de Dados sobre Saúde Mental de dissertações e teses acadêmicas**. 2022. 25 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2022[1].

O volume de dados existentes na internet cresce a cada dia, um formato de dado comum em diversos tipos de *softwares* é o texto. Além da internet, os repositórios de trabalhos acadêmicos (artigos, dissertações, monografias, teses e outros) também são uma fonte de dados textuais. Os textos podem fornecer muitas informações e extrair essas informações requer a utilização de técnicas de processamento de texto, por exemplo, o Processamento de Linguagem Natural. O presente trabalho apresenta o novo módulo desenvolvido para a plataforma *PsiAcademic Analytics*. Este novo módulo contém a refatoração da API utilizada para realização de consultas, inserção de novos trabalhos e aplicação de processamento de texto com Processamento de Linguagem Natural (PLN). A refatoração da API visa fornecer resultados mais precisos nas consultas, a inserção de novos trabalhos acadêmicos para aumentar a coleção de dados e por fim, a descoberta de palavras comuns com PLN para geração de informação sobre os dados.

PALAVRAS-CHAVE: PsiAcademic, API, Processamento de Linguagem Natural.

[1] Elemento incluído com a finalidade de posterior publicação do resumo na internet. Sua formatação segue a norma ABNT NBR 6023, por isto o alinhamento e o espaçamento diferem do padrão do texto

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API - (Application Programming Interface / Interface de Programação de Aplicação)

BSON - (Binary JSON / JSON binário)

JSON - (JavaScript Object Notation / Notação de Objetos JavaScript)

NLP - (Natural Language Processing / Processamento de Linguagem Natural)

REST - (Representational State Transfer / Estado Representacional de Estado)

UFAM - (Universidade Federal do Amazonas)

UFRGS - (Universidade Federal do Rio Grande do Sul)

UFMG - (Universidade Federal de Minas Gerais)

UFPA - (Universidade Federal do Pará)

UFPB - (Universidade Federal da Paraíba)

UFPE - (Universidade Federal de Pernambuco)

UFG - (Universidade Federal de Goiás)

UFRJ - (Universidade Federal do Rio de Janeiro)

UFRN - (Universidade Federal do Rio Grande do Norte)

UFSC - (Universidade Federal de Santa Catarina)

UFV - (Universidade Federal de Viçosa)

UNB - (Universidade de Brasília)

UNESP - (Universidade Estadual Paulista)

UNICAMP - (Universidade Estadual de Campinas)

URL - (Uniform Resource Locator / Localizador Uniforme de Recursos)

USP - (Universidade de São Paulo)

UTFPR - (Universidade Tecnológica Federal do Paraná)

WSGI - (Web Service Gateway Interface / Interface de Porta de Entrada do Servidor Web)

SUMÁRIO

1. INTRODUÇÃO

A *internet* e diversos sistemas de informação aumentaram o volume de dados de forma geométrica, seja em sites, em um banco de dados ou em outros dispositivos. Gradativamente, empresas entenderam a importância dos dados e os seus significados, caso queiram maximizar suas vendas ou descobrir o perfil do seu consumidor. Algumas áreas voltadas aos dados, como o *Big Data*, *Business Intelligence* ou a *Data Mining* (Mineração de Dados) ganharam mais importância através de pesquisas acadêmicas e investimentos tecnológicos (ISHIKIRIYAMA; MIRO; GOMES, 2015).

A busca por informações, também conhecida como descoberta de conhecimento, tem como etapa principal a Mineração de Dados, sendo a etapa principal na busca por conhecimento (CARVALHO; TSUNODA, 2018). Também, na descoberta de conhecimento, há um formato de dado relevante em análises, o formato texto. Uma técnica utilizada para extrair informações de dados textuais é a NLP (*Natural Language Processing*) ou PLN (Processamento de Linguagem Natural). A NLP permite classificar textos de forma automatizada, permitindo aos sistemas de computadores compreender a linguagem humana (FALCÃO; LOPES; SOUZA, 2022).

Uma das aplicações da NLP é apresentada em Chen et al. (2019), que mostra como o uso da técnica pode ser usada para classificar documentos, especificamente, relatórios sobre doenças cerebrovasculares. Em Finatto, Lopes e Ciulla (2015) também apresentam a utilização da NPL em análise de documentos com o objetivo de analisar textos científicos das áreas de medicina e linguística, gerando uma ontologia de parâmetros computacionais para textos científicos.

No Centro Universitário Luterano de Palmas (CEULP/ULBRA), o grupo de pesquisas multidisciplinar “Engenharia Inteligente de Dados” trabalha no projeto “Desenvolvimento de um Software para Análise Inteligente de Dados”. Nesse contexto, há um trabalho multidisciplinar realizado entre os cursos da área de Computação e o curso de Psicologia. Como parte das atividades de pesquisa do projeto, é realizado um levantamento de informações sobre o tema “Saúde Mental”, extraídas de repositórios acadêmicos de teses e dissertações, contendo informações sobre trabalhos de mestrado, TCCs etc.

A partir do levantamento de trabalhos sobre “Saúde Mental”, uma base de dados exclusiva sobre o tema foi criada, extraídos de repositórios de trabalhos de doze universidades brasileiras (UFPE, USP, UFRJ, UFMG, UFPB, UFRN, UNB, UFRGS, UFSC, UFAM, UFPA e UFG). Conforme Sousa (2020), essas amostras foram selecionadas pela especialista de

domínio (integrante do grupo Engenharia Inteligente de Dados), objetivando obter amostras de todas as regiões do Brasil.

As tecnologias utilizadas para a construção da base de dados com documentos sobre "Saúde Mental" são descritas nos trabalhos produzidos por Marinho et al., (2019a, 2018b). As informações sobre os trabalhos acadêmicos foram obtidas através da extração de dados nos repositórios selecionados, com a técnica de *Crawlers*. Ao final, conforme Sousa (2020), obteve-se como resultado uma base com mais de 59 mil documentos, armazenados em nuvem utilizando o serviço oferecido pela equipe do Mongodb, o Atlas. Para complementar, Sousa (2020) desenvolveu uma API de consulta personalizada sobre os dados armazenados. A API, na arquitetura REST, permite acesso padronizado a base de dados, fornece resultados no formato JSON e permite integrações com outras aplicações. Além da API, Sousa (2020) também desenvolveu um site que permite visualizar dados da base de dados utilizando a API.

Os projetos desenvolvidos por Marinho et al., (2019a, 2019b) e Sousa (2020) construíram a base do *PsiAcademic Analytics*, sendo possível adicionar novas funcionalidades através da adição de novos módulos, assim como o proposto neste trabalho. Contudo, ainda há melhorias a serem realizadas, por exemplo, no fornecimento de resultados da API, que em alguns casos não retornam os resultados conforme os parâmetros de consulta.

O presente trabalho propõe-se desenvolver um novo módulo para o *PsiAcademic Analytics* de extração, armazenamento, consultas e NLP com a base de dados sobre "Saúde Mental". As etapas de construção deste módulo envolvem a criação de *Scrapy* para extração de novos repositórios, armazenamento das novas extrações, reformulação da API e análise e obtenção de frequência de palavras nos documentos com NLP.

Com os resultados obtidos com a construção deste trabalho, espera-se atender adequadamente a consulta por informações necessárias para o grupo de pesquisa, mais especificamente, à equipe do trabalho multidisciplinar entre os cursos do Departamento de Computação e do curso de Psicologia do CEULP.

2. REFERENCIAL TEÓRICO

Nesta seção serão apresentados os conceitos sobre Extração de Dados (subseção 2.1), Banco de Dados Não Relacional (subseção 2.2) e Processamento de Linguagem Natural (2.3) e Trabalhos Relacionados. As subseções apresentam os conceitos das técnicas utilizadas neste trabalho.

2.2 EXTRAÇÃO DE DADOS

As inovações tecnológicas e a popularização da *internet* aumentaram exponencialmente o volume de dados, seja nas redes sociais, blogs, sites de notícias e outros. Os dados na internet possuem formatos variados, em textos, arquivos de música (.mp3), arquivos de vídeo (.mp4) imagens (.jpeg, .png), ou seja, os sites apresentam esses dados de diversas formas, contudo não estão estruturados. Esses dados são gerados e possuem valor quando transformados em informação, inclusive quando essas informações possuem valor comercial. Para isso necessita-se utilizar técnicas que permitam extrair dados na web de forma automatizada e uma das possibilidades é a utilização dos *Web Crawlers* (OLIVEIRA; BARACHO, 2016; MACHADO et al., 2015).

Os Web *Crawlers* ou *Crawlers* podem ser definidos como um *script* focado na *World Wide Web*. Os *crawlers* possuem diversas funcionalidades, como, atualizar o banco de dados dos motores de busca, realizar manutenções em sites, teste de *links* e outros (MACHADO et al., 2015). Os *Crawlers* funcionam obtendo dados de sites da internet, acessam *links* de forma recursiva (MARINHO et al. 2019a; OLIVEIRA; BARACHO, 2016) conforme parâmetros definidos, ou seja, captura um dado contido em algum bloco de código HTML (div, h1, p etc.) gerando um relatório após o processo (MACHADO et al., 2016).

A extração de dados na *Web* pode ser realizada de algumas formas, por exemplo, uma linguagem de programação que oferece suporte a esta atividade é a linguagem Python (MAZINI; SATO, 2019). A extração de dados pode ser feita utilizando recursos nativos da linguagem ou através de bibliotecas ou *frameworks*, destacando-se o *Beautiful Soup* e *Scrapy*. Além do acesso a *links* da internet, os *Crawlers* realizam um processo de transformação dos dados não estruturados em dados estruturados para que possam ser analisados ou armazenados em alguma base de dados (OLIVEIRA; BARACHO, 2016).

Os resultados de uma extração de dados podem ser utilizados em diversas áreas do conhecimento, uma delas é a ciência social, já os dados gerados pelos usuários de redes sociais e outros podem revelar informações sobre comportamentos (OLIVEIRA; BARACHO, 2016). Outra área beneficiada é a da economia, visto que sites de turismo podem utilizar analisar os dados, produzido no próprio site respondendo questionamento sobre os serviços oferecidos (OLIVEIRA; BARACHO, 2016).

2.3 BANCO DE DADOS NÃO RELACIONAL

O armazenamento de dados é fundamental para qualquer negócio ou pesquisa que utilize dados. As soluções comuns para o armazenamento de dados utilizam o modelo relacional e não relacional. No mercado há vários *softwares* que atendem aos modelos existentes, não há uma solução melhor que outra, mas, há soluções mais adequadas para um contexto. O volume de dados produzido e consumido por usuários da internet exige suporte para mudança e crescimento dos dados, neste sentido, os bancos de dados relacionais não são indicados para este contexto, sendo a solução não relacional mais indicada para esta situação (DIANA; GEROSA, 2010)

Os bancos de dados relacionais armazenam os dados utilizando tabelas e seus relacionamentos. Conforme Marinho et al., (2019b p.1), "tabelas para representar entidades do sistema, tradicionalmente buscam diminuir redundâncias e retratar de forma simplificada os relacionamentos entre essas entidades do Banco de Dados com a utilização de chaves primárias e estrangeiras". As características dos bancos relacionais, como realizações de transações com segurança, isolamento e integridade garantem a consistência dos dados armazenados, além de outras vantagens, contudo, não são indicados para ambientes que necessitam de escalabilidade.

A sigla NoSQL não se refere apenas a um tipo de modelo não relacional, há SGBD's, segundo Diana & Gerosa (2010), são "os tipos mais comuns de bancos de dados NOSQL: bancos de dados orientados a documentos, armazéns de chave-valor, bancos de dados de famílias de colunas e bancos de dados de grafos".

O banco de dados orientado a documentos é uma lista de documentos que não necessita de esquema ou estrutura comum, o formato de armazenamento das informações é semelhante ao JSON e permite documentos duplicados (ALMEIDA; BRITO, 2014) (DIANA; GEROSA, 2010). No banco, o tipo chave valor utiliza uma tabela *hash* que armazena os objetos indexados por chaves e permite buscas por estes objetos. No tipo orientado a colunas, conforme Lóscio, Oliveira e Pontes (2011, p. 6), "os dados são indexados por uma tripla (linha, coluna e *timestamp*), onde linhas e colunas são identificados por chaves e o *timestamp* permite diferenciar múltiplas versões de um mesmo dado". Por fim, os bancos que são orientados a grafos estão relacionados a contextos matemáticos composta por nós, vértices, relacionamentos e propriedades.

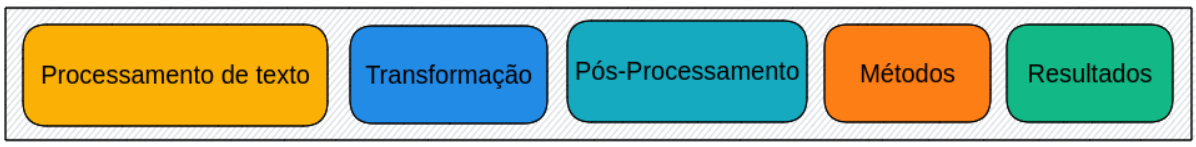
2.4 PROCESSAMENTO DE LINGUAGEM NATURAL

O Processamento de Linguagem Natural (PLN) ou Natural Language Processing (NLP), uma área do campo de Aprendizagem de Máquina, lida com a linguística e desenvolvimento de Modelos de Linguagens. Os modelos de Linguagem permitem determinar a probabilidade da ocorrência de palavras através da probabilidade (SINGH e MAHMOOD, 2021). Na análise de texto alguns fatores devem ser levados em consideração, primeiro a falta de estrutura dos dados, variando em estilos de escrita, e segundo, a ambiguidade das palavras, que podem possuir a mesma escrita, mas possuem significados diferentes em contextos diferentes.

O Processamento de Linguagem Natural envolve a análise de textos, em Niu et al. (2021), os dados textuais são o objeto de estudo principal da NLP, realizando a análise semântica e sintática para *corpus* de diferentes níveis como palavra, frase e documento. Para compreender com precisão um contexto de um texto, a língua deve ser levada em consideração, Othman et al. (2020) destaca que a transição de trabalhos do inglês para o árabe não é uma tarefa trivial, além da compreensão da própria língua árabe no qual palavras podem ser usada de diversas forma em contextos variados.

O processamento de dados textuais envolve algumas etapas e tratamento de dados para geração de informação, sendo necessário analisar os complicadores, entre eles, a falta de estrutura dos textos que usam diferentes tipos de escrita e o significado, visto que uma palavra tem sentidos em contexto variados (BEZERRA; GOLDSCHMIDT, 2010), a figura 1 apresenta fases do processamento.

Figura 1 - Fases do processamento



Conforme apresentado na figura 1, o processamento de texto, os tipos voltados para análise de conteúdo são, o processamento léxico que envolve o reconhecimento de termos, o processamento semântico que envolve extração de significados (entidades) dos textos e o processamento de características extra semânticas para identificação de sentimentos (BEZERRA; GOLDSCHMIDT, 2010). Na transformação das palavras o conjunto de palavras é obtido em um espaço vetorial para análise. Na seleção de recursos é necessário um pós-processamento de dados para analisar um grande conjunto de dados que ainda não estão estruturados. Na etapa de métodos define-se qual método, por exemplo, sumarização,

classificação etc. Por último análise dos resultados obtidos com a extração de texto (PATE; SONI, 2012; GUPTA; LEHAL, 2009). Para que haja uma análise de texto, os dados textuais precisam passar por técnicas de estruturação de texto, sendo realizado através da recuperação de informação ou extração de informação.

A NLP pode ser utilizada na análise de textos de diversas áreas, como tradução automática, análise de opiniões públicas, respostas inteligentes de perguntas, recuperação de informações, análise de sentimento e outros (NIU et al., 2021). O processamento de textos, conforme o modelo, os recursos computacionais tendem a ser utilizados em maior quantidade (SINGH e MAHMOOD, 2021).

2.5 TRABALHOS RELACIONADOS

Nesta subseção serão apresentados os trabalhos relacionados que utilizam NLP, Alawad et al. (2021), a NLP é utilizada para em conjunto com Deep Learning para analisar relatórios sobre registros de câncer populacional. Na análise dos documentos, um dos problemas encontrados pelos autores era manter a privacidade dos dados de pessoas, para isso desenvolveram um dicionário de significados desconsiderando informações pessoais de pacientes. Os resultados obtidos com dados formatos com NLP foram compartilhados com a Deep Learning, ou seja, a transferência de aprendizagem, testando alguns modelos de aprendizagem. Ao final, os pesquisadores concluíram que o modelo de *aprendizagem de transferência cíclica com preservação da privacidade* atendeu os objetivos dos pesquisadores.

No segundo artigo, Bose, Roy e Ghosh (2021) utilizaram a NLP no contexto da crise sanitária do COVID-19. O objeto de estudo dos autores eram trabalhos científicos relacionados a COVID-19, entender palavras-chave significativas sobre outros contextos afetados além da área da saúde como economia, psicologia e outros. Além de compreender contexto variados encontrados em trabalhos científicos, também foram identificadas outras categorias para países a fim de entender a resposta da comunidade científica diante da crise sanitária. A partir dos resultados obtidos, foram criadas nuvens de palavras, na classificação por países os principais resultados eram voltados para Estados Unidos, China e Itália. Os autores destacam que os resultados obtidos podem contribuir para um repositório de público de dados para construção de políticas públicas voltadas para prevenção de surtos no futuro.

3. MATERIAIS E MÉTODOS

Nesta seção serão apresentados os materiais e métodos utilizados para construção do módulo para o PsiAcademic, que inclui a captura de dados com Scrapy, utilização de NLP para análise dos trabalhos acadêmicos e uma nova versão da API de consultas com melhorias e novas funcionalidades.

3.1 MATERIAIS

3.1.1 Domínio de Dados

A base de dados utilizada neste trabalho contém dados sobre teses e dissertações sobre "Saúde Mental". Para construir esta base de dados Marinho et al., (2019a, 2019b) utilizaram técnicas de extração de dados na web para extrair dados em repositórios de trabalhos de 12 universidades, UFAM, UFG, UFMG, UFPA, UFPB, UFPE, UFRGS, UFRJ, UFRN, UFSC, UNB e USP, os dados extraídos foram: data, resumo, título, tipo, autores, palavra-chave e repositório. A tabela 01 apresenta detalhes sobre os dados armazenados.

Tabela 1 - Campos de um documento armazenado na base de dados

Atributo	Tipo de Dado	Descrição
id	ObjectID	Número gerado pelo MongoDB, não utiliza números inteiros para chaves primárias, mas um número hexadecimal
data	ISODate	Representação de um padrão internacional para datas, no formato: ano-mês-dia.
resumo	String	Campo que contém o resumo de um trabalho.
titulo	String	Campo que contém o título do trabalho.

tipo	String	Campo que contém o tipo do trabalho.
repositório	String	Campo que armazena o repositório de origem do trabalho.
autores	Array (String)	Campo que armazena os nomes dos autores do trabalho.
palavra-chave	Array (String)	Campo que armazena as palavras-chave de um trabalho.
url	String	Campo que fornece o *link* de acesso ao trabalho na internet

Após o processo de extração de dados na web e armazenamento, a base de dados disponibilizou mais de 59.000 documentos com trabalhos acadêmicos disponíveis para consulta. Para armazenamento dos dados, além da utilização do Mongodb, utilizou-se também o serviço Atlas, fornecido oficialmente pelo Mongo, um serviço de armazenamento em nuvem, sendo possível acessá-lo a qualquer momento através de um URL de acesso.

3.1.3 Intervalo de trabalhos

A base de dados atualmente contém trabalhos entre os anos de 1945 e 2020, contendo dados úteis e não úteis para o contexto do projeto. Para obtenção de resultados mais precisos, definiu-se com a especialista de domínio trabalhar com anos de publicação que continham ao menos um trabalho de cada repositório existente na base de dados, ou seja, um ano selecionado deveria conter trabalhos dos 12 (doze) repositórios extraídos até o momento.

3.1.4 API

A API desenvolvida por Sousa (2020), com exceção da consulta por índices, oferece consultas por todos os dados que existem em um documento, acessando o *link* da API e inserindo parâmetros para uma consulta retornar os resultados no formato JSON. A API permite que várias aplicações possam consumir os dados existentes na base de dados.

A API além de retornar resultados sobre consultas também retorna uma lista de informações presentes na base de dados como tipos de trabalhos, universidades que contém algum trabalho na base de dados e anos de publicações de todos os trabalhos.

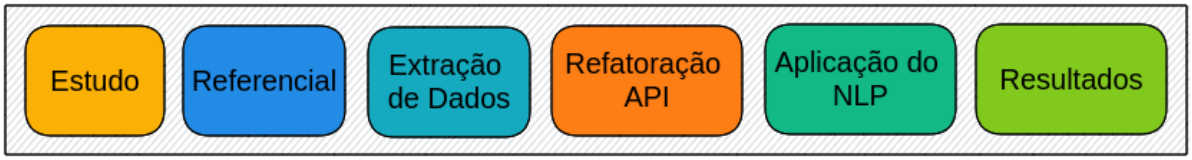
3.1.5 Tecnologias

Tecnologicamente, além das ferramentas (Python, Mongodb, Heroku) utilizadas por Marinho et al. (2019a, 2019b) e Sousa (2020), também foram utilizadas neste trabalho, com as seguintes ferramentas: FastAPI e NLTK. A linguagem de programação Python será utilizada em conjunto com o *framework* FastAPI para o desenvolvimento da lógica da aplicação, nesta lógica estarão inseridas formas de comunicação com a base de dados Mongodb e a conversão de dados para utilização na API, implantando a aplicação utilizando o serviço Heroku, disponibilizando o acesso aos dados na internet através da API.

3.2 MÉTODOS

Para o desenvolvimento do presente trabalho foram necessárias pesquisas sobre o contexto da aplicação. A figura 2 apresenta mais detalhes sobre a metodologia.

Figura 2 - Processo de desenvolvimento do trabalho



As etapas mostradas na figura 2 são partes do desenvolvimento e cumprimento dos objetivos. Para o desenvolvimento do módulo de extração, as seguintes etapas serão realizadas:

1. **Estudo**
   A partir de trabalhos anteriores já desenvolvidos, iniciou-se novos estudos para aperfeiçoar o conhecimento já existente. Pesquisas sobre trabalhos e projetos relacionados a temática principal foram identificadas e incorporadas na bibliografia do presente trabalho.

2. **Referencial**
   O Referencial Teórico foi desenvolvido com base nos estudos realizados, apresentando conceitos sobre extração de dados, armazenamento de dados com NoSql e *Text Mining*. Também, no referencial, há outros trabalhos que tratam de todos os assuntos e ao final, dois trabalhos relacionados que reúne os conteúdos do referencial.

3. **Extração de Dados**

   A extração de dados é o processo de obtenção de dados, conforme já informado, nesta etapa o *framework Scrapy* será utilizado para extrair de cada repositório dados sobre os trabalhos relacionados a “Saúde Mental”. Para cada repositório selecionado haverá um extrator de dados personalizado, visto que cada repositório possui uma estrutura diferente, mas tendo como resultado a extração dos dados. Os dados extraídos são inseridos na base de dados existente sobre "Saúde Mental".

4. **Refatoração da API**

   A refatoração dos recursos de consultas da aplicação e adição de novas funcionalidades em uma nova versão da API.

5. **Aplicação do NLP**

   Utilização de NLP, utilizando ferramentas para análise e produção de novas informações sobre a base de dados.

6. **Resultados**

   A fim de avaliar e comprovar tudo o que foi proposto pelo módulo, foram projetados a disponibilização na internet para que outras aplicações possam consumir os resultados da API juntamente com os especialistas de domínio.

4. RESULTADOS

Nesta seção serão apresentados os resultados obtidos conforme os objetivos estabelecidos. Serão descritas as melhorias referentes a refatoração da API de consultas, sanitização da base de dados sobre Saúde Mental e a utilização de Processamento de Linguagem Natural para obtenção de frequência de palavras comuns.

4.1. Base de dados

A base de dados construída por Marinho et al., (2019a, 2019b) contém mais de 59.000 (cinquenta e nove mil) documentos armazenados, resultado da extração de trabalhos em 12 (doze) repositórios universitários (UFAM, UFG, UFMG, UFPA, UFPB, UFPE, UFRGS, UFRJ, UFRN, UFSC, UNB e USP).

Em consultas realizadas na primeira versão da API de Sousa (2020), alguns resultados apresentavam documentos com dados incompletos, por exemplo, um documento sem data de publicação ou sem o tipo. Analisando a base de dados, dois tipos de inconsistências foram identificados, a primeira, documentos com dados incompletos e a segunda, documentos fora do contexto de "Saúde Mental". Para a sanitização da base de dados, realizou-se duas tarefas, a remoção de trabalhos com algum campo vazio ou nulo e a seleção de trabalhos que contenham de fato a expressão "Saúde Mental". A figura 3 apresenta um exemplo do comando do Mongo utilizado para encontrar documentos com o campo tipo vazio ou nulo.

Figura 3 - Comando Mongodb para contar documentos o campo tipo com valor nulo

```
db.trabalhos.find({"tipo": {"$eq": null}}).count();
```

Na figura 3, é realizada uma contagem de documentos no qual o campo tipo esteja vazio. Este tipo de verificação é realizado para os outros campos (autores, data, palavras-chave, repositório, resumo, título e url) na base de dados. Após a contagem dos documentos que contém valores nulos, estes documentos são removidos. A figura 4 apresenta o comando utilizado para remoção dos documentos com valores nulos.

Figura 4 - Comando Mongodb para deletar documentos que contém o campo tipo nulo

```
db.trabalhos.deleteMany({"tipo": {"$eq": null}});
```

Conforme apresentado na figura 4, o exemplo da remoção de documentos com “tipo” nulo, nos outros campos que compõem um documento, a mesma ação é realizada, ou seja, qualquer documento com um campo nulo é removido da base de dados. A verificação dos documentos utilizando a contagem também tem o objetivo de validação, o comando de deleção, na figura 5, também retorna um número natural inteiro confirmando a quantidade de documentos deletados e se ocorresse alguma divergência entre a quantidade de documentos verificados e deletados, um estado anterior da base de dados seria novamente verificado para deleção de documentos. Em dois campos dos documentos foram realizadas verificações específicas, a figura 4 apresenta o comando.

Figura 5 - Comando Mongodb que verifica o tamanho do campo “palavrachave”

```
db.trabalhos.find({"palavrachave": {"$size": 0}}).count();
```

Na figura 5, mostra a contagem de documentos que contém o campo “palavrachave” com valor zero, verifica o tamanho do *array* neste campo e caso o tamanho seja igual a 0 o documento é removido da base de dados, conforme a figura 6.

Figura 6 - Comando Mongodb que deleta documentos que contenham “palavrachave” zerada

```
db.trabalhos.deleteMany({"palavrachave": {"$size": 0}});
```

Outro campo no qual ocorreu a verificação de tamanho é o campo “autores”, os mesmos comandos utilizados nas figuras 6 e 6 também foram usados para documentos que contenham o campo “autores” com valor nulo.

A segunda etapa da sanitização dos documentos, foi a captura de documentos que continham exatamente a expressão “Saúde Mental”, visando seleção de documentos relacionados de fato à temática. Para encontrar os documentos relacionados ao tema, utilizou-se um comando do Mongodb para criação de índices de texto, a figura 7 apresenta este comando.

Figura 7 - Comando Mongodb para criação de índices de consulta

```
db.trabalhos.createIndex({"palavrachave": "text", "resumo": "text", "titulo": "text"});
```

O comando apresentado na figura 7, permite consultas no conteúdo dos textos. Logo após a criação dos índices de texto, para encontrar documentos com uma determinada expressão é utilizado um operador de consultas em texto, apresentado na figura 8.

Figura 8 - Consulta por expressão indexada

```
db.trabalhos.find({"$text": {"$search": "saude mental"}});
```

Concluída a etapa de sanitização, os documentos relacionados ao contexto de “Saúde Mental” e com dados completos são transferidos a uma nova base de dados, a figura 9 apresenta o comando de transferência e criação de uma nova base de dados.

Figura 9 - Comando Mongodb de transferência de dados para uma nova coleção

```
db.trabalhos.aggregate([
    {"$match":{"$text": {"$search": "saude mental"}}},
    {"$project": {"_id": 0, "autores": "$autores", "data": "$data", "palavrachave": "$palavrachave",
    "repositorio": "$repositorio", "resumo": "$resumo", "tipo": "$tipo","titulo": "$titulo",
    "url": "$url"}},
    {"$out": "novosTrabalhos"}]);
```

Os resultados obtidos a partir da sanitização dos documentos, com remoção de valores nulos e captura de documentos pertencentes ao domínio de “Saúde Mental”, conforme a figura 9, foram armazenados em uma nova coleção de dados, fornecendo uma base consistente e consultas com resultados mais precisos.

Os dados armazenados na base dados, apesar de sanitizados, não possuíam um critério de organização, por exemplo, havia documentos com datas de publicações de apenas um repositório, isto significa que, nos anos de publicação existentes na base tinham apenas um repositório. Com objetivo fornecer dados com maior consistência, definiu-se com a especialista um novo critério para padronização dos dados, sendo necessária a criação de uma nova coleção de documentos que atendesse ao seguinte requisito: ano que contém trabalhos de todos os repositórios. Na reformulação da base de dados foram selecionados trabalhos de anos que continham todos os repositórios, por exemplo, o ano de 2009 contém resultados de todos os repositórios, ao contrário de anos anteriores, ou seja, o ano de 2009 contém documentos elegíveis para a nova base de dados.

A seleção de trabalhos elegíveis para a nova coleção de dados foi feita realizando uma consulta na coleção sanitizada de dados, a figura 10 apresenta o comando utilizado no Mongodb.

Figura 10 - Comando agregação Mongodb

```
db.trabalhos.aggregate([
    {"$group": {"_id": {"$dateToString": {"format": "%Y", "date": "$data"}}, "repo": {"$addToSet": "$repositorio"}}},
    {"$project":{"repo": 1, "size": {"$size": "$repo"}}},
    {"$sort": {"size": -1, "_id": -1}}
]);
```

No comando da figura 10, primeiro é feito um agrupamento ($group) de repositórios por ano, os repositórios de um ano são acionados em um conjunto (para evitar resultados repetidos), segundo, é projetado ($project) um resultado que conta a quantidade de documentos de um ano e, por fim, são ordenados ($sort) por tamanho e ano decrescente, o resultado é apresentado na figura 11.

Figura 11 - Resultado da agregação de trabalhos

	_id	repo	size
1	2016	["UFSC", "UNB", "UFRN", "UFRJ", "UFRGS", "UFPE", "UFG", "UFPB", "UFPA", "USP", "UFAM", "UFMG"]	12
2	2015	["UFG", "UFSC", "UFMG", "USP", "UFRN", "UFAM", "UFPB", "UFRGS", "UFPE", "UFPA", "UFRJ", "UNB"]	12
3	2014	["UFRJ", "UFSC", "UNB", "UFMG", "UFAM", "UFPB", "USP", "UFPE", "UFG", "UFRGS", "UFRN", "UFPA"]	12
4	2013	["USP", "UFSC", "UFMG", "UFPE", "UFPA", "UFAM", "UFPB", "UFRGS", "UFG", "UFRJ", "UFRN", "UNB"]	12
5	2012	["UFSC", "UNB", "UFRJ", "UFMG", "UFAM", "UFPB", "USP", "UFPE", "UFRGS", "UFG", "UFPA", "UFRN"]	12
6	2011	["UNB", "UFSC", "UFPA", "UFRN", "UFG", "UFRGS", "UFRJ", "UFPB", "UFPE", "USP", "UFAM", "UFMG"]	12
7	2010	["UFRN", "UFSC", "UFMG", "UFPA", "USP", "UFAM", "UFPB", "UFRGS", "UFG", "UFRJ", "UFPE", "UNB"]	12
8	2009	["UFMG", "UFRN", "UFPA", "UFSC", "UFAM", "UFRGS", "UFPE", "UFG", "UFPB", "UNB", "USP", "UFRJ"]	12
9	2019	["UNB", "UFSC", "UFRN", "UFRJ", "UFRGS", "UFG", "UFPA", "UFPB", "USP", "UFPE", "UFMG"]	11
10	2018	["UFPA", "UFSC", "USP", "UFMG", "UFG", "UFPB", "UFRGS", "UFPE", "UFRJ", "UFRN", "UNB"]	11
11	2017	["UFSC", "UNB", "UFRJ", "UFPA", "UFRGS", "UFPE", "UFG", "UFPB", "UFRN", "USP", "UFMG"]	11
12	2008	["UFMG", "UFRJ", "UFPA", "UFSC", "UFG", "UFRGS", "UFPE", "UFPB", "USP", "UNB", "UFRN"]	11
13	2007	["UFSC", "UNB", "UFPA", "UFMG", "UFPB", "USP", "UFPE", "UFRGS", "UFG", "UFRN", "UFRJ"]	11
14	2006	["UFSC", "UNB", "UFRJ", "UFRN", "UFRGS", "UFG", "UFPA", "USP", "UFPB", "UFPE", "UFMG"]	11
15	2005	["UFSC", "UNB", "UFRN", "UFPA", "UFRGS", "UFRJ", "UFG", "USP", "UFPE", "UFPB", "UFMG"]	11
16	2004	["UFSC", "UFPA", "UNB", "UFMG", "USP", "UFPB", "UFPE", "UFRGS", "UFG", "UFRJ", "UFRN"]	11
17	2003	["UFSC", "UFRN", "UNB", "UFMG", "USP", "UFRGS", "UFG", "UFRJ", "UFPA"]	9
18	2001	["UFSC", "UNB", "UFPA", "UFMG", "USP", "UFRGS", "UFG", "UFRJ", "UFRN"]	9
19	2002	["UFSC", "UNB", "UFRJ", "UFMG", "UFPB", "UFRGS", "UFPA", "UFRN"]	8
20	1999	["UFMG", "UFRJ", "UFSC", "UFRGS", "UFPE", "UFG", "UNB", "UFPA"]	8

Conforme apresentado na figura 11, os anos de 2009 a 2016 contém todos os repositórios, confirmado pela coluna do resultado *size* com os resultados 12, ou seja, o número total de repositórios existentes na base de dados.

Após a seleção dos documentos, a próxima etapa foi a criação de uma nova coleção de dados com os anos selecionados. Para criação de uma nova coleção, utilizou-se o comando do Mongodb apresentado na figura 12.

Figura 12 - Transferência de dados para coleção “Saude Mental”

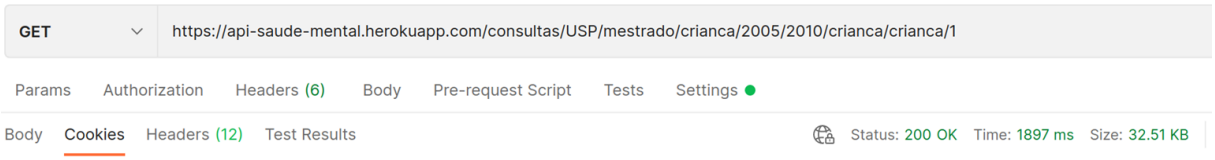
```
db.trabalhos.aggregate([
    {"$match": {"$and": [
        {"data": {"$gte": ISODate("2009-01-01")}}, {"data": {"$lte": ISODate("2016-12-31")}}
    ]}},
    {"$project":
        {"_id": 0, "data": "$data", "url": "$url", "resumo": "$resumo",
        "titulo": "$titulo", "tipo": "$tipo", "autores": "$autores",
        "palavrachave": "$palavrachave", "repositorio": "$repositorio"}
    },
    {"$out": "saudemental"}]);
```

Além da criação de uma nova coleção, outros trabalhos foram adicionados à nova base de dados, extraídos e apresentados em Silva et al. (2021). As etapas para extração de novos trabalhos e o armazenamento de dados envolveu a seleção de novos repositórios. Os novos repositórios adicionados foram UTFPR, UFV, UNICAMP, UNESP, com documentos sanitizados (sem campos incompletos ou fora do contexto) e dentro do critério de anos estabelecido. No total, a base de dados conta com 16 repositórios de trabalhos universitários com trabalhos entre 2009 e 2016.

4.2. API

Em Sousa (2020), foi desenvolvido a primeira versão API que retorna consultas personalizadas sobre a base de dados com documentos sobre "Saúde Mental" construída por Marinho et al., (2019a, 2019b). O trabalho desenvolvido por Sousa (2020) permite que aplicações Web ou Mobile, por exemplo, consultem dados através de uma interface. Assim, seguindo os parâmetros pré-definidos é possível pesquisar trabalhos acadêmicos por anos, por tipo e outros atributos. Na API, ao utilizar recurso de consulta genérica com temática "criança", tinha como resultado retorno de dados acima 1500 milissegundo ou aproximadamente 1,5 segundos, a figura 13 apresenta um exemplo.

Figura 13 - Resultado de consulta genérica da API de Sousa (2020)



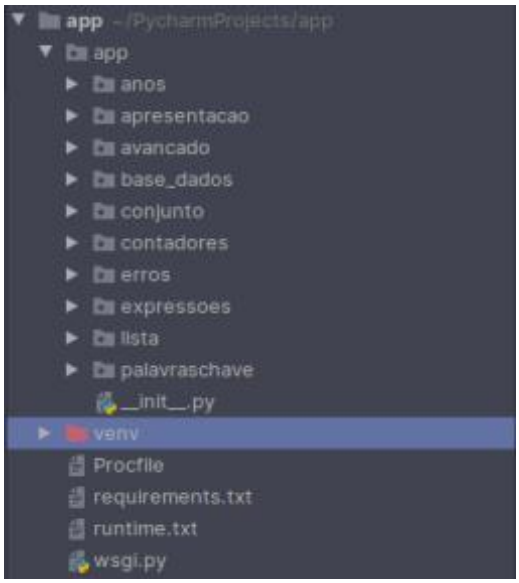
Ao utilizar outros recursos de consulta, percebeu-se que nos resultados havia documentos fora do contexto da pesquisa, utilizando novamente o exemplo da temática "criança" em consultas genéricas, continham documentos com títulos "Potencial iatrogênico da psicanálise" ou "Agravos à saúde mental dos homens envolvidos em situações de violência", a figura 14 apresenta as palavras-chave de um documento retornado como resultado na temática de "criança".

Figura 14 - Palavras-chave de um documento retornado em consulta genérica

```
"palavrachave": [
    "Efeitos iatrogênicos",
    "Formação do psicanalista",
    "Iatrogenia",
    "Poder",
    "Psicanálise",
    "Responsabilidade"
],
```

A segunda versão da API teve como foco a resolução do problema de resultados imprecisos e resultados com menor tempo de retorno. A reformulação envolveu a utilização do *framework* FastAPI, refatoração das funções existentes na primeira versão da API e novas inserções de funcionalidades. A primeira versão da API possuía uma organização modularizada, conforme a figura 15. Exceto os módulos apresentação, erros e base de dados, os módulos possuem configurações para acesso a base de dados de trabalhos acadêmicos.

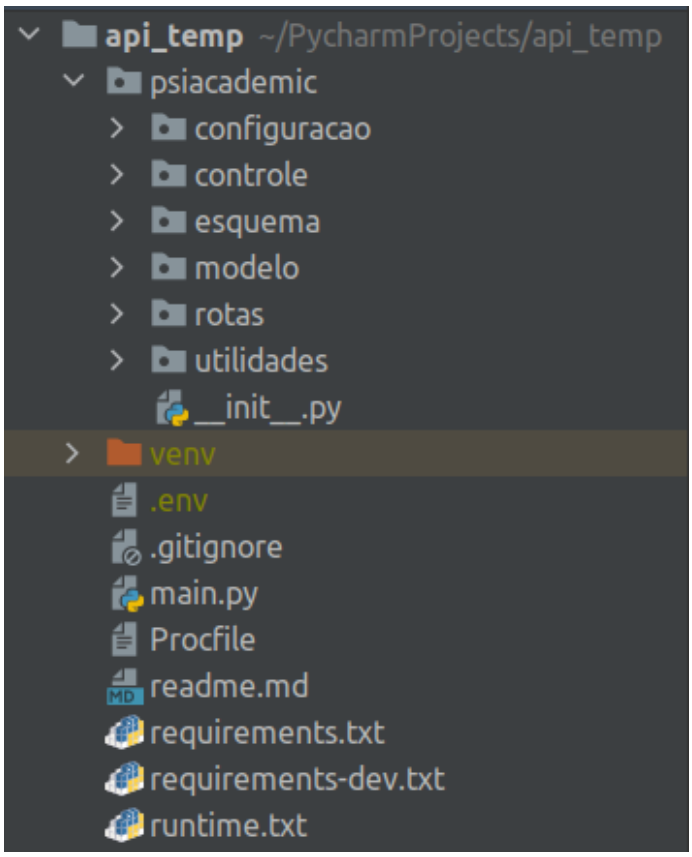
Figura 15 - Organização API de Sousa (2020)



Cada módulo é um pacote Python contendo os seguintes arquivos: *__init__.py* (arquivo necessário para criação de um módulo), *routes* (rotas para os recursos) e *views* (recursos da api).

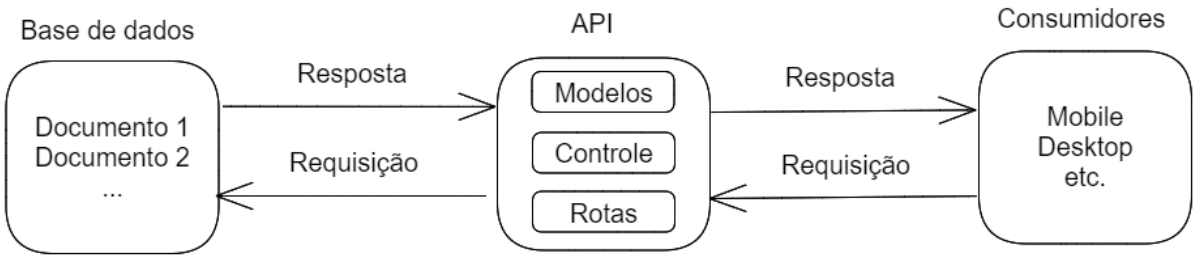
Nesta primeira versão da API do *PsiAcademic*, a aplicação não seguia um modelo de arquitetura, uma das implicações por não seguir nenhuma arquitetura está na dificuldade de realizar manutenções. Para resolver o problema de manutenção, a segunda versão do *PsiAcademic* seguiu a arquitetura MVC (*Model View Controller*) para organizar a aplicação, conforme apresentado na figura 16.

Figura 16 - Organização da API versão 2



A mudança de arquitetura adotada na segunda versão da API não alterou o formato de comunicação utilizado na primeira versão, usando o formato REST. A comunicação realizada pela API é apresentada na figura 17.

Figura 17 - Representação da comunicação realizada pela API



Conforme apresentado na Figura 17, a camada de rotas é responsável por receber requisições feitas e a API retorna uma resposta no formato JSON. O controle concentra o atendimento às requisições feitas aos recursos da API, retornando uma mensagem de sucesso ou erro. A camada de modelos contém as regras de negócios para as consultas realizadas na base de dados, os resultados obtidos utilizam os parâmetros recebidos e retornam resultados já formatados.

Com a atualização também foi realizada a refatoração de código para acesso aos recursos, as principais estão em modelos, com a separação de responsabilidades entre as funções de consultas e as funções de transformação de dados. A figura 18 mostra a função da primeira versão da API para consultar documentos por ano de publicação, que concentra a consulta e transformação de dados, respectivamente nas linhas 13 e 14. A concentração de muitas responsabilidades em uma única função não fornece boas consultas, além de aumentar a complexidade na manutenção do código.

Figura 18 - Função Python para obtenção de ano

```
1  # Consulta de documentos por ano de publicação
2  def data_ano(ano, numero_pagina):
3      if numero_pagina <= 0:
4          abort(400)
5
6      if validacao_ano(ano) is False:
7          return make_response(jsonify(''), 204, headers)
8
9      indice = (numero_pagina - 1) * 10
10     ano_inicio = datetime(ano, 1, 1)
11     ano_fim = datetime(ano, 12, 31)
12
13     documentos = __colecoes.find({'data': {'$gte': ano_inicio, '$lte': ano_fim}}).skip(indice).limit(10)
14     resposta = json.loads(json_util.dumps(documentos))
15
16     if len(resposta) == 0:
17         return make_response(jsonify(''), 204, headers)
18
19     headers['X-Total-Count'] = math.ceil((documentos.count()) / 10)
20
21     return make_response(jsonify(resposta), 200, headers)
```

A figura 19 mostra a divisão de responsabilidades para a conversão de dados que são retornados em um consultas no Mongodb que por padrão utilizam o formato BSON.

Figura 19 - Funções de conversão da segunda versão da API

```python
def retornar_dados_em_str(dados: Cursor) -> str:
    dados_str: str = json_util.dumps(dados)
    return dados_str


def retornar_dados_em_dict(dados: str) -> list:
    dados_list: list = json.loads(dados)
    return dados_list


def converter_bson_para_list(consulta: [Cursor, Collection]) -> list:
    dados_em_str = retornar_dados_em_str(consulta)
    dados_em_list = retornar_dados_em_dict(dados_em_str)
    return dados_em_list
```

Na figura 19 as funções de conversão de dados utilizada na segunda versão da API, o resultado de uma consulta (em *converter_bson_para_dict*) é convertida primeiramente para o formato *String* (*retornar_dados_em_str*) e por fim para o dicionário (*retornar_dados_em_dict*), formato suportado pelo Python. A responsabilidade por consulta de anos é apresentada na figura 20.

Figura 20 - Função de consulta por ano da segunda versão da API

```python
def consultar_por_ano(ano_inicio, ano_fim, pagina, limite):
    parametros = {"data": {"$gte": ano_inicio, "$lte": ano_fim}}
    consulta = colecao.find(parametros).skip(pagina).limit(limite)
    consulta_convertida = retornar_consulta_convertida(consulta)
    return consulta_convertida
```

A figura 19 mostra a função em modelos que utiliza a função que converte os resultados das consultas, ou seja, a responsabilidade é receber os parâmetros e realizar a consulta. As modificações na arquitetura e no código permitiram uma evolução mais estável da API, caso haja alteração nas regras de negócio ou novas funcionalidades, há menores riscos de a aplicação deixar de funcionar em caso de modificações.

Na atualização de versão da API do *PsiAcademic* houve modificações nas funcionalidades de consultas oferecidas, a tabela 2 apresenta as funcionalidades atuais da versão atual da API.

Tabela 2 - Recursos da API versão 1

Recurso	Tipo de Recurso	Resultado
Anos	Consulta por ano	Retornada trabalhos de um ano específico
	Consulta por anos anteriores	Retorna trabalhos anteriores a um ano informado pelo usuário, por exemplo, trabalhos publicados anteriores aos anos 2000
	Consulta por anos posteriores	Retorna trabalhos posteriores a um ano informado pelo usuário, por exemplo trabalhos publicados após os anos de 2010
	Consulta por período de anos	Retorna trabalhos publicados em um período
Conjunto	Consulta por quantidade de documentos	Retorna documentos, sem nenhum tipo de critério
	Consulta por repositório	Retorna documentos de um repositório específico
	Consulta por tipo	Retorna documentos de um tipo específico

Expressões	Consulta por expressões em palavras-chave	Retorna documentos conforme expressão em palavras-chaves
	Consulta por expressão em autores	Retorna documentos conforme expressão em autores
	Consulta por expressão em títulos	Retorna documentos conforme expressão em títulos
	Consulta por expressão em resumos	Retorna documentos conforme expressão em resumos
Palavras-chave	Consulta em palavras-chave (usando expressão)	Retorna documentos conforme expressão em palavras-chave
Avançada	Consulta utilizando múltiplos parâmetros (repositório, tipo, palavra-chave, período de ano, resumo, título)	Retorna documentos que atendam a pelo menos um parâmetro passado na consulta

A API também fornece o retorno de outros dados existentes na base de dados, conforme apresentado na tabela 3.

Tabela 3 - Recursos de retorno de dados da API versão 1

Recurso	Tipo de Recurso	Resultado
Lista	Anos	Retorna uma lista com todos os anos de publicação dos trabalhos
	Repositórios	Retorna uma lista com todos os repositórios
	Tipos	Retorna uma lista com todos os tipos de trabalhos
Contadores	Número de trabalhos por anos	Retorna uma lista com o número total de trabalhos por ano
	Número de trabalhos por repositórios	Retorna uma lista com número total de trabalhos por repositório
	Número de trabalhos por tipos	Retorna uma lista com o número total de trabalhos por tipo

As consultas fornecidas pela segunda versão da API são apresentadas na tabela 4. Em comparação a primeira versão, houve o acréscimo da consulta por índices.

Tabela 4 - Novos recursos da API versão 2

Genérica	Consulta por múltiplos parâmetros.	Consulta que utiliza diversos parâmetros, o resultado atende a pelo menos um parâmetro passado. Esta consulta aceita os seguintes parâmetros: título, tipo, resumo, ano, palavras-chave.
Ano	Consulta por ano, anterior ao ano e posterior ao ano	Esta consulta aceita os parâmetros ano e tipo. O parâmetro tipo não é obrigatório e aceita dois parâmetros: anterior e posterior. Caso o parâmetro esteja preenchido, serão retornados resultados anteriores ou posteriores ao ano indicado.
	Período de anos	Consulta por resultados dentro de um intervalo de anos

Tipo	Consulta por tipo de trabalho	Retorna documentos por tipo, por exemplo, apenas artigos ou trabalhos de conclusão de curso.
Repositório	Consulta em um repositório	Retorna trabalhos de um repositório
Página	Consulta indexada	Retorna um documento por seu índice
Expressão	Consulta por expressão em resumos ou títulos	Retorna documentos conforme o tipo de dados consultado (título ou resumo) e a expressão consultada.
	Consulta por expressão em resumos ou títulos	Retorna documentos conforme os dados consultados (título ou resumo) e o seu
Palavras-Chave	Consulta por palavra-chave	Retorna documentos que contenha pelo menos uma palavra-chave passada como parâmetro

Outro recurso adicionado na segunda versão da API são os totalizadores. Conforme apresentado na tabela 5, o recurso de listagem de dados foi mantido e o recurso de contagem de dados foi incorporado aos totalizadores.

Tabela 5 - Recursos de retorno de dados da API versão 2

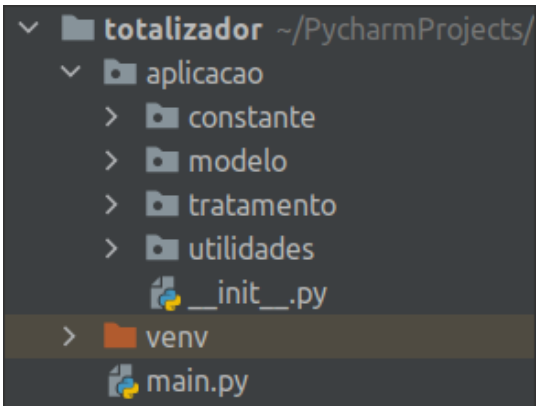
Lista	Repositórios	Retorna todos os repositórios da base de dados
	Tipos	Retorna todos os tipos de trabalhos existentes na base de dados
	Anos	Retorna todos os anos de publicação dos trabalhos
Totalizadores	Ano	Retorna uma lista com o número total de trabalho publicado em cada ano.
	Tipo	Retorna uma lista com o número de total de trabalho por tipo
	Repositório	Retorna uma lista com o número total de trabalho por repositório
	Repositório e Tipo	Retorna uma lista com o número total de trabalho por tipo em cada repositório
	Repositório e Ano	Retorna uma lista com o número total de trabalho por ano de um repositório
	Transtornos, anos e repositórios	Retorna uma lista de repositórios com o ano e o total de trabalhos de cada ano

	Transtornos e Anos	Retorna uma lista de anos com o transtorno e total de trabalhos de um transtorno
	Transtornos e ano	Retorna uma lista com todos os anos e o total de trabalhos em cada ano de um tipo de transtorno.
	Transtorno, repositório e ano	Retorna uma lista com número total de trabalhos por transtorno de um ano.

4.3. Processamento de Linguagem Natural

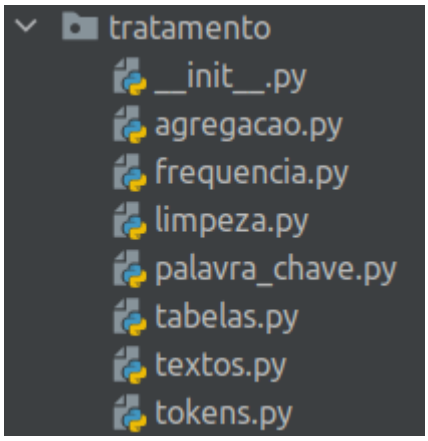
A atualização e refatoração da base de dados forneceram um ambiente para análises mais precisas, tanto para API quanto para outras aplicações que possam acessar diretamente a base de dados. Para obtenção das frequências das palavras, em conjunto com a especialista de domínio, definiu-se o retorno da frequência absoluta das 10 palavras mais comuns. A quantidade de palavras frequentes foi estabelecida como valor mínimo para as frequências. O valor mínimo de palavras na frequência também tem como objetivo a realização de testes, análise dos resultados e refatoração da aplicação de frequência em resultados não satisfatórios obtidos. A primeira análise realizada sobre a base de dados foi a descoberta de palavras com as maiores frequências nos documentos, os dados analisados foram: títulos, resumos, palavras-chave e tipo. Para analisar os documentos houve a construção de uma aplicação específica, utilizando os recursos oferecidos pela linguagem Python e a biblioteca NLTK (*Natural Language Toolkit*), a figura 21 apresenta a estrutura desta aplicação.

Figura 21 - Organização da Aplicação de frequências



A figura 21 mostra a estrutura desta aplicação, primeiramente o arquivo *main.py* é quem executa a verificação de palavras e conversa com os outros módulos da aplicação. Os módulos constante, modelo, tratamento e utilidades contém as configurações para acesso e análise de dados. O primeiro módulo, constante, contém um arquivo chamado *constantes.py*, neste arquivo há dados utilizados por toda aplicação, como o acesso a coleção de dados e as *stopwords* utilizadas no processamento de palavras. O segundo módulo, modelo, contém as funções utilizadas na consulta de dados na coleção de dados, por exemplo, consultar todos os títulos fornecidos pelo módulo constante. A figura 22 apresenta o módulo tratamento.

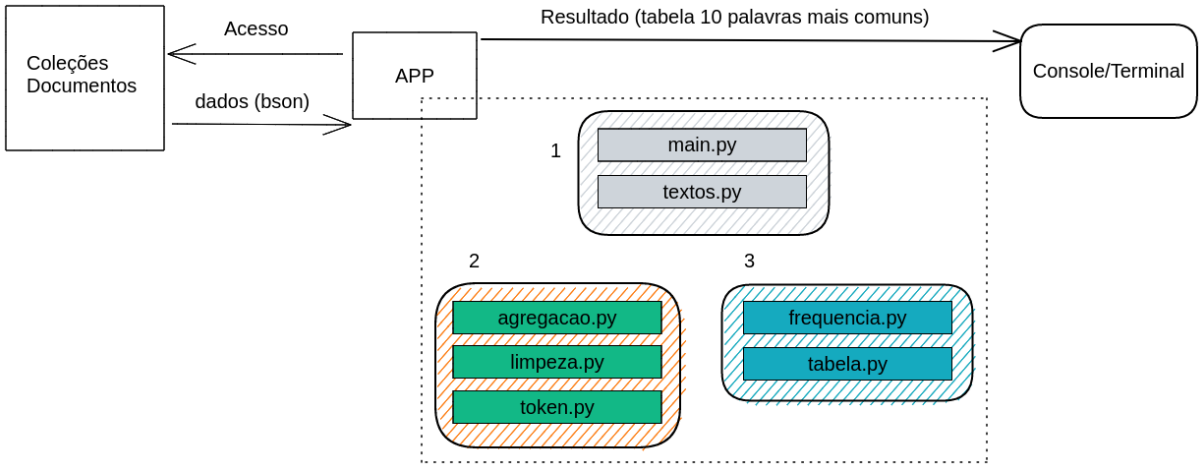
Figura 22 - Organização do módulo de tratamento



O terceiro módulo, apresentado na figura 22, contém as etapas que ocorrem durante o processamento de palavras. O arquivo *agregacao.py* contém as funções responsáveis por juntar as palavras processadas para serem utilizadas em cálculos. O arquivo *frequencia.py* contém a função para calcular a frequência das palavras, aqui é utilizada a função fornecida pelo NLTK FreqDist, que retorna a palavra e a quantidade de vezes em que ela aparece em um texto. O arquivo *limpeza.py* contém as funções responsáveis por padronizar uma palavra, ou seja, retirar

as pontuações, acentos, números, colocar a palavra em minúsculo e remover as *stopwords*. As *stopwords* são palavras que não fornecem valor semântico dentro de um contexto de análise de palavras, por exemplo, na língua portuguesa, conectivos ‘ou’, conjunções ‘e’, artigos ‘a/o’ e outros. O arquivo *palavra_chave.py* contém o processamento específico para palavras-chaves dos documentos. O arquivo *tabela.py* contém as funções responsáveis por criar a tabela de frequência das palavras, que recebe a frequência das palavras e cria uma tabela com as 10 palavras que mais aparecem dentro de um texto. O arquivo *textos.py* contém funções que fornecem tipos de tokenização de palavras e criação de tabelas. Por fim, o arquivo *token.py* contém a tokenização de palavras, com tokenização por espaço em branco ou por vírgula. A figura 23 mostra o funcionamento da aplicação de verificação de frequência.

Figura 23 - Funcionamento da aplicação de frequência



Conforme apresentado na figura 23, a primeira ação da aplicação é acessar a base de dados e obter documentos no formato BSON. Ao utilizar, como exemplo, os dados de títulos, todos os títulos existentes são retornados em um vetor, enviando o resultado ao módulo de texto conforme o tipo de “tokenização” desejada. O segundo passo é a agregação de todos os títulos em um único valor, uma *string* com todas as palavras de todos os títulos que passam para o processo de limpeza (padronização das palavras) e são “tokenizadas” por espaço em branco ou por vírgula, no caso de palavras-chave. Finalizando o processo, os *tokens* são enviados para a criação de frequência absoluta das palavras dos títulos e com este resultado é gerada uma tabela com as 10 palavras mais frequentes, neste caso, dos títulos. A figura 24 mostra a frequência de palavras para títulos.

O tipo de tokenização e os resultados foram todos validados com a especialista e, caso uma palavra ou tipo de tokenização não fizesse sentido para o contexto de saúde mental, o processamento de palavras era refatorado para geração de novos resultados. A revisão da

frequência de palavras foi sendo construída conforme a realização de teste e avaliação com a especialista de domínio, resultando em uma lista de palavras ignoradas na realização da frequência. Para cada dado analisado, uma lista de palavras ignoradas foi construída, para títulos (“avaliacao”, “avaliação”, “estudo”, “caso”, “analise”, “educacao”, “brasil”, “municipio”, “minas”, “processo”, “atencao”, “revisao”, “basica”, “vida”, “perfil”, “proposta”, “programa”, “belo”, “equipe”, “primaria”, “bucal”), palavras-chave (“amazonia”, “brasileira”, “belem”, “pa”, “historia”, “literatura”, “ensino”, “enfermagem”, “gerais”, “geral”, “pouco”, “estrategia”, “nursing”, “idosos”) e resumos (“dados”, “pesquisa”, “objetivo”, “resultados”, “pacientes”, “profissionais”, “qualidade”, “populacao”, “p”, “fatores”, “acao”, “acoes”, “uso”, “risco”, “desenvolvimento”, “estudos”). Os resultados das análises obtidos com a remoção das palavras ignoradas são apresentados nas figuras 24, 25 e 26.

Figura 24 - Frequência de palavras comuns em títulos

Frequência de Títulos - Tokenizado espaco em branco:

	termos	frequencia
6	saude	4027
58	familia	847
545	idosos	502
26	mental	496
258	intervencao	416
311	criancas	386
164	tratamento	383
301	hospital	312
260	adolescentes	296
90	social	282

Na figura 24, o tipo de “tokenização” utilizado para obtenção das frequências de palavras foi o espaço em branco. O resultado obtido nesta frequência de palavras passou por refatoração e validação pela especialista de domínio.

Figura 25 - Frequência de palavras comuns de palavras-chave

Frequencia de Palavras Chave - Tokenizado por virgula:

	termos	frequencia
77	saude	1776
5	saude familia	734
0	saude mental	580
259	saude publica	301
607	promocao saude	240
532	idoso	235
66	saude trabalhador	183
277	unico saude	164
1108	hipertensao	163
1649	diabetes mellitus	155

A frequência de palavras, figura 25, utiliza a "tokenização" por vírgula, validado com a especialista de domínio, este tipo de tokenização permite analisar uma expressão contida em palavras-chave, não apenas palavras de forma separada.

Figura 26 - Frequência de palavras comuns em tipos de trabalhos

Frequencia de Tipos - Tokenizado espaco em branco:

	termos	frequencia
0	dissertacao	5964
3	monografia	3288
4	especializacao	2880
1	tese	2633
5	relatorio	1465
2	artigo	135
6	livro	7
7	periodico	5

O resultado obtido com a "tokenização" de tipos, apresentado na figura 26, trouxe a necessidade de agrupamento para os termos monografia, dissertação e tese. Nos primeiros resultados havia uma repetição de termos com um mesmo significado, por exemplo, tese e doutorado, mestrado e dissertação, em ambos os casos as palavras possuem o mesmo sentido para dizer a qual tipo de trabalho um documento pertence. Desta forma, alinhado com a especialista de domínio, optou-se por agrupar os seguintes termos: tese ("doutorado",

“doctoralthesis”), dissertação: (“masterthesis”, “mestrado”, “profissional”) e monografia (“monografias”, “conclusao”, “graduacao”, “tcc”).

Figura 27 - Frequência absoluta de palavras em resumos

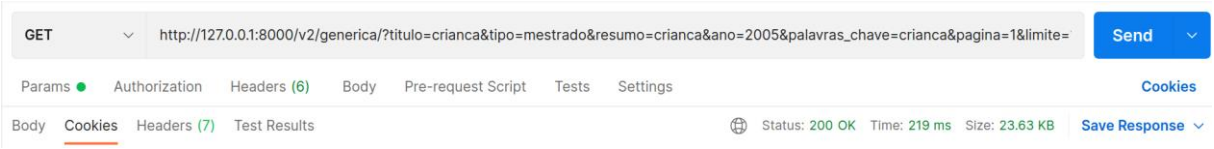
Frequencia de Resumos - Tokenizado espaco em branco:

	termos	frequencia
4	saude	34245
212	tratamento	4763
30	familia	4201
253	social	4000
5	mental	3863
239	doenca	3695
7	servicos	3564
2351	idosos	3409
1167	criancas	3222
1000	mulheres	3165

A frequência de palavras em resumos, apresentada na figura 26, também teve seu resultado validado com a especialista de domínio e refatorado. A refatoração feita em cada frequência foi realizada para o contexto das palavras existentes na base de dados, ou seja, além das *stopwords* removidas, palavras que não faziam sentido para o contexto também foram removidas. O objetivo das frequências das palavras é gerar maior conhecimento sobre a base de dados.

Os resultados obtidos com a obtenção das frequências das palavras também foram adicionados à nova versão da API. A nova versão da API possui as funcionalidades de consultas por documentos, lista de dados existentes na base de dados, quantidade total de trabalhos por tipo, repositório, anos e a frequência de palavras comuns na base de documentos. A segunda versão da API, além de retornar documentos mais precisos, em comparação a primeira versão, a segunda versão da API, utilizando o recurso de consulta genérica, retornou resultados em menos de 1 segundo, conforme a figura 28.

Figura 28 - Consulta genérica da segunda versão da API



A sanitização da base de dados também possibilitou melhorias nas consultas realizadas pela API, visto que os documentos possuem dados completos e pertencentes ao contexto de saúde mental, o que também permite outras análises de com técnicas de processamento de texto em conjunto com a frequência de palavras.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho tinha quatro objetivos principais: a extração de novos repositórios, refatoração da API de acesso a base de dados de “Saúde Mental”, sanitização dos dados armazenados e obtenção de frequência de palavras-chaves. Todos os objetivos estabelecidos e concluídos apresentados neste trabalho atende a outros contextos além deste trabalho.

O estabelecimento de um critério de organização de dados também contribuiu para a criação de uma base de dados com dados mais consistentes, outras aplicações, além da nova versão da API, poderiam fornecer resultados com outros detalhes sobre os documentos armazenados.

A obtenção da frequência absoluta das palavras mais comuns da base de dados, foi o primeiro passo para introdução da Ciência de Dados, utilizando técnicas de estatística e Inteligência Artificial para descoberta de novas informações e geração de conhecimento.

Os trabalhos futuros planejados a partir deste trabalho são a construção de um léxico de transtornos que permita ao classificador de documentos uma classificação com maior precisão, a extração de novos trabalhos para geração de novas informações sobre “Saúde Mental” e estabelecer novos períodos de trabalhos além do limite definido.

REFERÊNCIAS

FINATTO, Maria José Bocorny; LOPES, Lucelene; CIULLA, Alena. **Processamento de Linguagem Natural, Linguística de Corpus e Estudos Linguísticos: uma parceria bem-sucedida**. Domínios de Lingu@Gem**.** Porto Alegre, p. 41-59. dez. 2015. Disponível em: https://lume.ufrgs.br/handle/10183/169398. Acesso em: 25 jun. 2022.

ALAWAD, Mohammed *et al***. Privacy-Preserving Deep Learning NLP Models for Cancer Registries**. Ieee Transactions On Emerging Topics In Computing, [S.L.], v. 9, n. 3, p. 1219-1230, 1 jul. 2021. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tetc.2020.2983404. Disponível em: https://ieeexplore.ieee.org/document/9069186. Acesso em: 26 jun. 2022.

ALMEIDA, Ricardo Cardoso de; BRITO, Parcilene Fernandes de. **Utilização da Classe de Banco de Dados NOSQL como Solução para Manipulação de Diversas Estruturas de Dados**. In: ENCONTRO DE COMPUTAÇÃO E INFORMÁTICA DO TOCANTINS, Não use números Romanos ou letras, use somente números Arábicos., 2012, Palmas. **Anais [...] .** Palmas: Centro Universitário Luterano de Palmas, 2012. v. 15, p. 151-160. Disponível em: http://ulbra-to.br/encoinfo/wp-content/uploads/2020/06/Utiliza%C3%A7%C3%A3o-da-Classe-de-Banco-de-Da. Acesso em: 10 out. 2020

AMERICAN PSYCHIATRIC ASSOCIATION. **Manual diagnóstico e estatístico de transtornos mentais: DSM-5**. 5.ed. Porto Alegre: Artmed, 2014.

ATLAS. Disponível em: <https://www.mongodb.com/cloud/atlas> Acesso em:11 de abril 2022.

Batista de CARVALHO, Marcelo, Fukumi TSUNODA, Denise **Análise de dados em artigos recuperados da Web of Science (WoS)**. Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação [en linea]. 2018, 23(1), 112-125[fecha de Consulta 1 de Julio de 2021]. ISSN: . Disponible en: https://www.redalyc.org/articulo.oa?id=14762635010

BEZERRA, Eduardo; GOLDSCHMIDT, Ronaldo. **A Tarefa de Classificação em Text Mining**. Revista de Sistemas de Informação da Faculdade Salesiana Maria Auxiliadora, Macaé, v. 5, n. 0, p. 42-62, 26 abr. 2010.

BOSE, Priyankar; ROY, Satyaki; GHOSH, Preetam**. A Comparative NLP-Based Study on the Current Trends and Future Directions in COVID-19 Research**. Ieee Access, [S.L.], v. 9, p. 78341-78355, 2021. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/access.2021.3082108.

CHEN, Pengyu et al. **Automatically Structuring on Chinese Ultrasound Report of Cerebrovascular Diseases via Natural Language Processing.** Ieee Access, [S.L.], v. 7, p. 89043-89050, 2019. Institute of Electrical and Electronics Engineers (IEEE).

http://dx.doi.org/10.1109/access.2019.2923221. Disponível em: https://ieeexplore.ieee.org/document/8736947. Acesso em: 26 jun. 2022.

DIANA, Mauricio de; GEROSA, Marco Aurélio. NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0. **Workshop de Teses e Dissertações em Banco de Dados**, Belo Horizonte, v. , n. 0, p. 1-8, out. 2010. Disponível em: https://www.ime.usp.br/~mdediana/nosql_wtdbd10.pdf. Acesso em: 26 out. 2020.

FALCÃO, Luander Cipriano de Jesus; LOPES, Brenner; SOUZA, Renato Rocha. Absorção das tarefas de processamento de Linguagem Natural (NLP) pela Ciência da Informação (CI): uma revisão da literatura para tangibilização do uso de nlp pela ci. **Em Questão**, [S.L.], v. 28, n. 1, p. 13-34, 1 jan. 2022. Faculdade de Biblioteconomia Comunicacao. http://dx.doi.org/10.19132/1808-5245281.13-34. Disponível em: https://seer.ufrgs.br/index.php/EmQuestao/article/view/111323. Acesso em: 26 jun. 2022.
FASTAPI. Disponível em: <https://fastapi.tiangolo.com/> Acesso em: 15 de maio 2022.

GUPTA, Adity; TYAGI, Swati; PANWAR, Nupur; SACHDEVA, Shelly; SAXENA, Upaang. NoSQL databases: critical analysis and comparison. In: 2017 INTERNATIONAL CONFERENCE ON COMPUTING AND COMMUNICATION TECHNOLOGIES FOR SMART NATION (IC3TSN), 1., 2017, [S.I]. 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN). [S.L.]: Ieee, 2017. p. 293-299. Disponível em: https://ieeexplore.ieee.org/document/8284494. Acesso em: 01 out. 2020.

GUPTA, Vishal; LEHAL, Gurpreet S.. A Survey of Text Mining Techniques and Applications. **Journal Of Emerging Technologies In Web Intelligence.** Panjab, p. 60-76. ago. 2009. Disponível em: http://www.jetwi.us/uploadfile/2014/1230/20141230112729939.pdf. Acesso em: 01 abr. 2021.

H. Niu, C. Ma, P. Han, S. Li and Q. Ma, "A Novel Semantic Cohesion Approach for Chinese Airworthiness Regulations: Theory and Application," in IEEE Access, vol. 8, pp. 227729-227750, 2020, doi: 10.1109/ACCESS.2020.3046294.

HEROKU. Disponível em: <https://www.heroku.com/what> Acesso em: 11 de abril 2022.

ISHIKIRIYAMA, Célia Satiko; MIRO, Diego; GOMES, Carlos Francisco Simões. Business Intelligence e Big Data: um exemplo prático de aplicação de Text Mining. Simpósio de Excelência em Gestão e Tecnologia: Otimização de Recursos e Desenvolvimento, Resende, p. 1-10, out. 2015. Disponível em: https://www.researchgate.net/profile/Carlos-Francisco-Gomes/publication/283328530_Business_Intelligence_e_Big_Data_um_exemplo_pratico_de_aplicacao_de_Text_Mining/links/5633c96d08ae758841121b56/Business-Intelligence-e-Big-Data-um-exemplo-pratico-de-aplicacao-de-Text-Mining.pdf. Acesso em: 01 jun. 2021.
LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de; PONTES, Jonas César de Sousa. NoSQL no desenvolvimento de aplicações Web colaborativas. In: SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, Não use números Romanos ou letras, use somente números Arábicos., 2011, Paraty. SIMPÓSIO. Paraty: Sociedade Brasileira de Computação, 2011. p. 1-17. Disponível em: https://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf. Acesso em: 01 jun. 2021.

M. T. B. Othman, M. A. Al-Hagery and Y. M. E. Hashemi, **"Arabic Text Processing Model: Verbs Roots and Conjugation Automation,"** in IEEE Access, vol. 8, pp. 103913-103923, 2020, doi: 10.1109/ACCESS.2020.2999259.

MACHADO, C.C. et al. **Um Web Crawler para Projeções e Análise de Vulnerabilidades de Segurança e Consistência Estrutural de Páginas Web**. Revista de Empreendedorismo, Inovação e Tecnologia, [S.L.], v. 2, n. 2, p. 3-12, 30 dez. 2015. Complexo de Ensino Superior Meridional S.A.. http://dx.doi.org/10.18256/2359-3539/reit-imed.v2n2p3-12. Disponível em: http://seer.imed.edu.br/index.php/revistasi/article/view/869. Acesso em: 01 jun. 2021.

MARINHO, D. S. et al. Web **Crawlers na Extração de Informações de Teses e Dissertações sobre Saúde Mental**. XIX Jornada de Iniciação Científica. Palmas - Tocantins. Setembro, 2019.

MARINHO, D. S. et al. **Estrutura Não Relacional para Dados de Saúde Mental com NoSQL Mongodb**. XIX Jornada de Iniciação Científica. Palmas - Tocantins. Setembro, 2019.

MAZINI, Dhaniel Nunes; SATO, Renato Cesar. **Extração de dados financeiros com um web scraper: um estudo sobre a rentabilidade dos dividendos**. Waiaf - Workshop Of Artificial Intelligence Applied To Finance. São José dos Campos, p. 1-4. maio 2019. Disponível em: http://www.comp.ita.br/labsca/waiaf/papers/DhanielMazini_paper_20.pdf. Acesso em: 01 jun. 2021.

MONGO. Disponível em: <https://www.mongodb.com/pt-br> Acesso em: 16 de maio 2022.

NLTK. Disponível em: <https://www.nltk.org/> Acesso em: 15 de maio 2022.

OLIVEIRA, Rafael Almeida de; BARACHO, Renata Maria Abrantes. **Extração de dados do site TripAdvisor como suporte na elaboração de indicadores do turismo de Minas Gerais: uma iniciativa em Big Data**. Pesquisa Brasileira em Ciência da Informação e Biblioteconomia, João Pessoa, v. 11, n. 0, p. 26-37, jul. 2016. Disponível em: https://www.researchgate.net/publication/315471858_Extracao_de_dados_do_site_TripAdvisor_como_suporte_na_elaboracao_de_indicadores_do_turismo_de_Minas_Gerais_uma_iniciativa_em_Big_Data. Acesso em: 15 ago. 2020.

PATE, Falguni N.; SONI, Neha R.. Text mining: A Brief survey. **International Journal Of Advanced Computer Research.** Bhopal, p. 243-248. dez. 2021. Disponível em: https://www.researchgate.net/publication/275347413_Text_mining_A_Brief_Survey. Acesso em: 30 abr. 2021.

PYMONGO. Disponível em: <https://api.mongodb.com/python/current> Acesso em:11 de abril 2022.

S. Singh and A. Mahmood, **"The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures,"** in IEEE Access, vol. 9, pp. 68675-68702, 2021, doi: 10.1109/ACCESS.2021.3077350.

SCRAPY. Disponível em: <https://scrapy.org/> Acesso em: 16 de maio 2022.

SILVA, Felipe Silva e et al. **EXTRAÇÃO DE DADOS SOBRE "SAÚDE MENTAL" PARA ADICIONAR À BASE DE DADOS DO PSIACADEMIC ANALYTICS. In: XXI JORNADA DE INICIAÇÃO CIENTÍFICA**, 21., 2021, Palmas. Anais [...] . Palmas: Centro Universitário Luteranos de Palmas, 2021. p. 75-79. Disponível em: https://fswceulp.nyc3.digitaloceanspaces.com/jornada-de-iniciacao-cientifica/2021/artigos/ciencias-exatas/extracao-de-dados-sobre-saude-mental-para-adicionar-a-base-de-dados-do-psiacademic-analytics.pdf. Acesso em: 27 jun. 2022.

SOUSA, Pablo Henrique de. **DESENVOLVIMENTO DE UMA API PARA CONSULTA DE INFORMAÇÕES SOBRE "SAÚDE MENTAL" EM UM BANCO DE DADOS MONGODB**. 2020. 37 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas, 2020.
Tocantins. Setembro, 2019.