Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Indexação e Referências Textuais: Um Estudo de Caso com Implementação de Ferramenta para o Programa Nacional de Atividades Espaciais

Igor Pessoa Rocha

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador
Prof. Dr. Alexandre Zaghetto

Coorientador
Prof. Dr. Romualdo Alves

Brasília
2014

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Veneziano

Banca examinadora composta por:

Prof. Dr. Alexandre Zaghetto (Orientador) — CIC/UnB
Prof. Dr. Célia Ghedini Ralha — CIC/UnB
Prof. Dr. Romualdo Alves — AEB

CIP — Catalogação Internacional na Publicação

Rocha, Igor Pessoa.

Indexação e Referências Textuais: Um Estudo de Caso com Implementação de Ferramenta para o Programa Nacional de Atividades Espaciais / Igor Pessoa Rocha. Brasília : UnB, 2014.
62 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2014.

1. dicionário, 2. indexação, 3. python, 4. pnae, 5. espacial, 6. aeb, 7. pln, 8. léxico

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Indexação e Referências Textuais: Um Estudo de Caso com Implementação de Ferramenta para o Programa Nacional de Atividades Espaciais

Igor Pessoa Rocha

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Alexandre Zaghetto (Orientador)
CIC/UnB

Prof. Dr. Célia Ghedini Ralha
CIC/UnB

Prof. Dr. Romualdo Alves
AEB

Prof. Dr. Wilson Veneziano
Coordenador do Curso de Computação — Licenciatura

Brasília, 26 de março de 2014

Dedicatória

Dedico este trabalho à minha família, à minha namorada e aos meus amigos.

Agradecimentos

Agradeço primeiramente a meus pais, que sempre me apoiaram e me deram condições de estar aqui hoje. Aos meus irmãos, que conviveram todo esse tempo comigo e souberam me incentivar quando precisei. À minha namorada Ana Paula Bottecchia, que participou de todo esse período em que estive na universidade, me ajudando, apoiando e me dando forças até quando não as tinha mais, sabendo compreender os inúmeros momentos em que estive ausente para fazer este mesmo trabalho. Agradeço ao colega Lucas Rosa pela ajuda de web design. Agradeço ao meu coorientador, Romualdo Alves, pela tamanha ajuda, horas dedicadas à melhoria do trabalho e motivação em momentos que precisava e ao meu orientador Alexandre Zaghetto pela disposição e boa vontade em tornar este trabalho possível.

Resumo

O Programa Nacional de Atividades Espaciais (PNAE) é um programa brasileiro de incentivo ao desenvolvimento das atividades espaciais. Ele representa as prioridades de integração da política espacial às demais políticas públicas em execução no Brasil e foi resultado de um seminário realizado por representantes do Governo, de segmentos científicos, acadêmicos e empresariais. Por se tratar de um documento de importância nacional, o relatório do PNAE é um documento disponível a toda sociedade, porém com um vocabulário muito técnico, podendo limitar a compreensão dos leitores leigos da área espacial tais como gestores públicos e a própria sociedade. Tendo em vista esse cenário, este trabalho propõe uma aplicação, batizada de Indexer, que identifica todos os termos técnicos simples e compostos, no domínio da área espacial, do documento do PNAE. Além de identificar os termos técnicos, o Indexer também apresenta os respectivos significados ao leitor no momento em que ele posiciona o ponteiro do mouse sobre as palavras escolhidas, agregando valor descritivo aos termos e fornecendo informações contextualizadas. Os termos técnicos da área espacial são identificados de acordo com o Dicionário Enciclopédico de Astronomia e Astronáutica, que foi estruturado e preparado para a consulta. O Indexer foi desenvolvido na linguagem Python e possui também a possibilidade de indexar outros documentos da área espacial que estejam disponíveis em formato .html ou .txt.

Concluiu-se do trabalho realizado que o objetivo do desenvolvimento do Indexer foi alcançado, isto é, os resultados obtidos proveram ao leitor o enriquecimento do documento do PNAE e de outros documentos da área com informações complementares dos termos técnicos espaciais, diferenciando termos simples de termos compostos o que aumenta a contextualização dos significados.

Palavras-chave: dicionário, indexação, python, pnae, espacial, aeb, pln, léxico

Abstract

The National Program of Spacial Activities (PNAE) is a Brazilian program to encourage the development of spacial activities. The PNAE represents the integration priorities of the spacial policy to other public policies in execution at Brazil. The PNAE was launched by a Government Seminar where scientists, academic and business segments participated. Regarding the national importance of this document, the PNAE report, it is a document available to the whole society. Nevertheless, the PNAE report uses technical vocabulary that limits the understanding of citizens and public managers. Given this scenario, this work proposes an application, named Indexer, which identifies all the technical terms simple and compound, in the field of space research. In addition to identify the technical terms, the Indexer also presents their meanings to the reader by positioning the mouse pointer over the words identified, adding descriptive value to the terms and providing contextual information. Technical terms of spatial area are identified according to the Encyclopedic Dictionary of Astronomy and Astronautics, which was structured and prepared for database querying. The Indexer has been developed in Python, and has the possibility of indexing other spatial-related documents that are available in .html or .txt.

As a conclusion, the Indexer proposed and developed in this project has provided the reader the enrichment of the Program document and other documents in the spatial area with complementary information of the technical terms, differentiating simple and compound terms which enhances the meanings contextualization.

Keywords: dictionary, nlp, python, indexing, python, pnae, spacial, lexicon

Sumário

1 Introdução **1**
1.1 Problema . . . 2
1.2 PNAE . . . 2
1.2.1 A implementação e as prioridades do PNAE . . . 2
1.2.2 As ações do PNAE . . . 4
1.2.3 As considerações do programa . . . 4
1.3 O Dicionário Enciclopédico de Astronomia e Astronáutica . . . 5
1.3.1 O objetivo e verbetes específicos . . . 5
1.4 Justificativa . . . 6
1.5 Objetivo Geral . . . 6
1.6 Objetivos Específicos . . . 6

2 Referencial Teórico **7**
2.1 A Indexação de conteúdos textuais . . . 7
2.1.1 Sistema de Recuperação de Informação . . . 8
2.1.2 Sistema de Recuperação de Informação x Sistema de Gerenciadores de Bancos de Dados . . . 9
2.1.3 Processo de Indexação . . . 9
2.1.4 Modelos de SRI . . . 10
2.2 Etapas de Pré Processamento . . . 11
2.2.1 *Tokenization* . . . 11
2.2.2 Expressões Regulares . . . 12
2.2.3 *Stopwords* . . . 12
2.3 O Léxico . . . 14

3 Referencial Tecnológico **16**
3.1 Linguagem Python . . . 17
3.1.1 Python como Linguagem Interpretada . . . 18
3.1.2 Interpretador Interativo . . . 19
3.1.3 Tipagem dinâmica . . . 20
3.1.4 Controle de bloco por indentação . . . 21
3.1.5 Orientação a Objetos . . . 21
3.2 Django . . . 21
3.2.1 Os componentes do Django . . . 22
3.3 JavaScript . . . 23
3.3.1 Jquery . . . 23

3.4 CSS 25

4 Proposta **27**
4.1 O Léxico no Indexer 27
4.1.1 Extração das palavras-chave 28
4.2 Indexer 32
4.2.1 Visão Geral do Indexer 32
4.2.2 Trabalhos Correlatos 32
4.2.3 Banco de dados 33
4.2.4 Levantamento de Requisitos 34
4.2.5 Algoritmo 35
4.2.6 Apresentação do Documento Indexado 38

5 Conclusão e Trabalhos Futuros **42**

Referências **44**

I Código do Programa de Extração de Termos e Significados **46**

II Código do Programa de Indexação do Indexer **49**

Lista de Figuras

1.1 Dicionário Enciclopédico de Astronomia e Astronáutica [29]. 5

2.1 Conceitos de Indexação [14]. 7
2.2 Componentes de um sistema de recuperação de informação [8]. 8
2.3 Exemplo de *tokenization* utilizando expressão regular. 12
2.4 Exemplo do uso de expressão regular. Busca por verbos no infinitivo, caracterizados por conterem "r"como última letra. 13

3.1 Estrutura do Projeto. 16
3.2 Python representado pela área hachurada. 17
3.3 Interpretador [19]. 17
3.4 Compilador [19]. 18
3.5 Interpretador interativo IDLE [20]. 18
3.6 Prompt de comando do Windows [20]. 19
3.7 Testando retorno do método par(). 20
3.8 Django representado pela área hachurada. 22
3.9 JavaScript representado pela área hachurada. 23
3.10 Jquery contido dentro de JavaScript e representado pela área hachurada. . 24
3.11 CSS representado pela área hachurada. 25

4.1 Dicionário antes da filtragem. 29
4.2 Padrão de palavra-chave e significado. 29
4.3 Dicionário depois de passar pela primeira etapa da extração. 30
4.4 Fluxograma do algoritmo de extração. 31
4.5 Expressão regular das palavras-chave em azul e dos significados em vermelho. 31
4.6 Tela inicial do Indexer. 33
4.7 Comparativo de softwares correlatos. 34
4.8 Tabelas "STOPWORDS"e "TERMOS". 35
4.9 Fluxograma do algoritmo do Indexer. 36
4.10 Termos identificados. 40
4.11 Significado do termo composto "hemisfério sul". 41

Lista de Tabelas

2.1 Metacaracteres básicos de expressões regulares [5]. 14

Capítulo 1

Introdução

Criado em 1996 pelo governo brasileiro para apresentar as ações, expectativas e investimentos que serão feitos na área espacial do Brasil, o Programa Nacional de Atividades Espaciais (PNAE) é um documento público, que pode ser encontrado no site da Agência Espacial Brasileira (AEB). O Seminário de Revisão do Programa Espacial Brasileiro, realizado em dezembro de 2004, foi resultante de um debate aberto à sociedade, onde participaram representantes do Governo, de segmentos científicos, acadêmicos e empresariais e que deu origem à terceira revisão brasileira do PNAE 2004 - 2014 [3]. Revisão esta que foi utilizada como base para o programa apresentado na monografia. Apesar disso, em 2012, por "mudanças no cenário estratégico do país" PNAE (2012) [3], foi antecipado um novo documento cobrindo o período de 2012 a 2021. Segundo PNAE (2005) [2]:

> Somente os países que dominem a tecnologia espacial poderão ter autonomia na elaboração de cenários de evolução global capazes de levar em conta tanto os impactos da ação humana, quanto os dos fenômenos naturais. Serão estes os países em condição de sustentar posições e argumentar nas mesas de negociação diplomática.

Visando melhorar o entendimento do conteúdo específico do PNAE entre os leitores, o presente trabalho tem a intenção de identificar palavras específicas do documento e de explicitar os respectivos significados, apresentando-os ao leitor de forma clara e instantânea. O trabalho já foi apresentado no primeiro e segundo Colóquio de Arquitetura da Informação do Centro de Pesquisa de Arquitetura da Informação da Universidade de Brasíia e utiliza o Dicionário Enciclopédico de Astronomia e de Astronáutica [29] como base para detalhar melhor as palavras específicas da área espacial.

Este trabalho propõe um algoritmo de identificação e indexação de palavras, que distingue termos simples de termos compostos e que será detalhado nos próximos capítulos. Foi utilizado a linguagem Python para o desenvolvimento de toda a parte de processamento da indexação. A visualização dos significados foi feita utilizando html. Já para fazer a aplicação web, foi utilizado um framework web do Python chamado Django.

A monografia detalha as tecnologias usadas para o desenvolvimento do projeto, assim como a metodologia, quais problemas surgiram e quais tiveram de ser solucionados e os resultados obtidos.

1.1 Problema

O PNAE é um documento específico da área espacial e, por isso, contém diversos termos técnicos da área que podem ser incompreendidos por leitores que não estão inseridos no contexto em questão. Apesar do PNAE conter vários termos específicos, ele é um documento feito com a participação da sociedade e disponível a públicos de diversos tipos de perfis.

1.2 PNAE

A terceira revisão do Programa Nacional de Atividades Espaciais (PNAE) [2] é um documento elaborado pela Agência Espacial Brasileira (AEB) e pelo Ministério da Ciência e da Tecnologia que revela estatísticas, dados e informações dos anos de 2005 a 2014 e responde às orientações da Política Nacional de Desenvolvimento de Atividades Espaciais. O relatório propõe, ainda, as prioridades para os dez anos seguintes, mostrando o cronograma das atividades e o orçamento dos valores gastos para o período decenal.

Segundo o documento, o programa define missões e estabelece ações destinadas a concretizar os objetivos ali estabelecidos. "Nele se incluem, também, as prioridades e diretrizes que norteiam a execução do conjunto das atividades espaciais que deverão servir de referência para o planejamento atual e plurianual dos componentes do Sistema Nacional de Desenvolvimento de Atividades Espaciais", descreve o presidente da época da Agência Espacial Brasileira, em 2005, Sergio Gaudenzi.

De acordo com a apresentação do projeto [2], o PNAE "é estratégico para o desenvolvimento soberano do Brasil. A importância da capacitação no domínio da tecnologia espacial, que, em seu ciclo completo, abrange centros de lançamento, veículos lançadores, satélites e cargas úteis, decorre de sua relevância para o futuro do País".

1.2.1 A implementação e as prioridades do PNAE

Além de analisar o serviço espacial brasileiro em contexto nacional e internacional e de gerar informações públicas para a sociedade, o PNAE também estabelece prioridades para o Brasil para os próximos anos. Para que seja feito de forma satisfatória, o programa se baseia, de acordo com o relatório, em alguns princípios:

- Foco no atendimento às necessidades dos usuários públicos de bens e serviços espaciais.
- Autonomia na área de pequenos satélites e respectivos veículos lançadores.
- Adoção de padrões de segurança e qualidade compatíveis com as normas internacionais.
- Busca da sustentabilidade do modelo de financiamento das atividades espaciais mediante a comercialização de bens e serviços espaciais.
- Integração da indústria e da academia ao conjunto das instituições envolvidas com a implementação do PNAE.

- Fortalecimento das instituições, direta ou indiretamente envolvidas com a implementação do PNAE, com ênfase em:

    i. Formação, capacitação e alocação de recursos humanos de modo a favorecer a inovação tecnológica e o aperfeiçoamento da gestão;
    ii. Utilização de métodos, técnicas e ferramentas de gestão do conhecimento gerado no âmbito destas instituições; e
    iii. Utilização de métodos, técnicas e ferramentas de planejamento estratégico e tecnológico para a área espacial.

Além disso, a cada revisão - que teoricamente é realizada a cada dez anos, os responsáveis estabelecem metas e prioridades para os dez anos seguintes no setor espacial. Segundo o relatório apresentado em 2005, as metas para o próximo período eram:

- Continuação do desenvolvimento do Veículo Lançador de Satélites - VLS e seus sucessores, com incremento da participação industrial, e da infra-estrutura de lançamento, incluindo o Centro de Lançamento de Alcântara - CLA.
- Em todas as missões, sempre que possível, será agregada uma carga útil para manutenção e atualização do Sistema de Coleta de Dados Brasileiro, eliminando a necessidade de satélites específicos.
- Conclusão do projeto da Plataforma Multimissão e suas cargas úteis.
- Continuação do projeto dos satélites CBERS e dos meios de processamento e distribuição de imagens.
- Promoção da comercialização dos meios de acesso ao espaço, pela implantação da infra-estrutura geral do Centro Espacial de Alcântara, que inclui sítios de lançamento comerciais.
- Investimentos em Pesquisa e Desenvolvimento voltados para o domínio de tecnologias críticas, com participação dos setores acadêmico e industrial.
- Condução de projetos mobilizadores, que atendam às demandas nacionais em Observação da Terra, Missões Científicas e Tecnológicas, Telecomunicações e Meteorologia, incluindo o desenvolvimento de satélites geoestacionários, de satélites de observação por radar de abertura sintética - SAR e missões científicas com satélites, balões e foguetes de sondagem.
- Manutenção e industrialização do bem-sucedido programa de foguetes de sondagem.
- Incremento da participação da indústria nacional no desenvolvimento das atividades e projetos contemplados pelo Programa, promovendo, inclusive, a transferência de tecnologias geradas no INPE e no CTA.
- Utilização de instrumentos de cooperação internacional que envolvam transferência de tecnologia e coincidam com os interesses nacionais.

1.2.2 As ações do PNAE

Alguns dos objetivos previstos no programa é o estabelecimento de parcerias com universidades brasileiros para o desenvolvimento de produtos, tecnologias e processos para que sejam realizadas com sucesso as missões previstas pelo PNAE, principalmente as que visam o "desenvolvimento de satélites tecnológicos e a qualificação de novas tecnologias de lançadores nacionais".

Com isso, segundo o relatório, o PNAE identifica como indispensável a ampliação do conhecimento em tecnologias estratégicas e sua transferência para o setor industrial, para garantir a este setor capacitação adequada e melhor competitividade, ao responder às demandas do mercado internacional (...) Na área de veículos espaciais, é indispensável, também, a aquisição das tecnologias de guiagem e controle, de sistemas inerciais, de materiais e processos, de sistemas computacionais e de propulsão líquida".

Com isso, o programa prevê diversas ações nas seguintes áreas:

- Tecnologias estratégicas;
- Ciências espaciais e atmosféricas - com pesquisas e estudos que se baseiam principalmente na área de Aeronomia, Astrofísica e Geofísica Espacial;
- Mudanças globais - com a finalidade de apoiar projetos de pesquisa cujos temas se baseiem na utilização de meios, técnicas ou produtos espaciais;
- Microgravidade - com a ação de fomentar e coordenar no Brasil projetos de pesquisa e desenvolvimento baseados na realização de experimento em ambiente de microgravidade;
- Geoposicionamento - com o interesse em contribuir "ativamente não apenas na consecução de iniciativas mas, também, para a concepção e implementação das mesmas".

1.2.3 As considerações do programa

Quando foi apresentado em 2005, o documento afirmou existir a necessidade de se adotar estratégias para o fortalecimento da indústria espacial e apresentou as seguintes considerações:

- Algumas tecnologias necessárias ao desenvolvimento de certos componentes e subsistemas para as missões e demais projetos do PNAE não são dominados no país, além de haver restrições para sua transferência.
- Os produtos espaciais fornecidos pelo setor industrial são certificados pelos próprios órgãos setoriais do SINDAE, durante o seu processo de aquisição.
- Alguns projetos, embora passíveis de serem transferidos para a indústria, continuam sendo desenvolvidos no ambiente dos institutos de pesquisa, resultando numa baixa utilização da capacidade instalada e dos recursos humanos qualificados da indústria no desenvolvimento das missões e demais projetos do PNAE.
- A indústria não utiliza sua flexibilidade de contratação para atrair especialistas – sobretudo os recém-formados – de modo a propiciar o surgimento de inovações no ambiente das empresas.

- A incipiente circulação de conhecimentos e o quase inexistente intercâmbio de especialistas entre universidades, institutos de pesquisa e indústria.

1.3 O Dicionário Enciclopédico de Astronomia e Astronáutica

O dicionário usado para a indexação dos termos específicos dentro do Indexer ao documento do Programa Nacional de Atividades Especiais foi o Dicionário Enciclopédico de Astronomia e Astronáutica [29], veja Figura 1.1, publicado pela Editora Nova Fronteira em 1987 e co-editado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

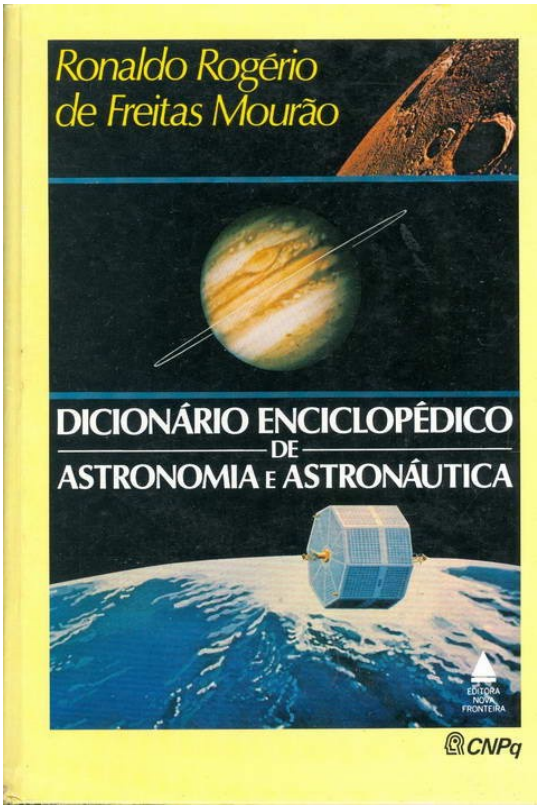


Figura 1.1: Dicionário Enciclopédico de Astronomia e Astronáutica [29].

De acordo com o então presidente do Conselho na época, Crodowaldo Pavan, "o campo da divulgação científica, embora grandemente ampliado no País através de múltiplas iniciativas, ainda está longe de suprir as demandas originadas dos vários segmentos da sociedade. Toda ação com o objetivo de ampliar os canais de acesso à informação no âmbito da Ciência e da Tecnologia deve merecer apoio".

1.3.1 O objetivo e verbetes específicos

O livro, que conta com mais de 18.000 palavras tem como objetivo, segundo a publicação, a elaboração de um dicionário especializado em termos astronáuticos, uma fonte

de pesquisa em que os usuários pudessem compreender melhor as palavras específicas da área. "Todos os verbetes foram redigidos numa linguagem simples e a mais objetiva possível. Além dos termos astronômicos, não dispensamos os das ciências afins, tais como a astrofísica, meteorologia, metrologia, geofísica, sismologia, cronometria, cronologia, hemerologia, física, etc. Incluímos até mesmo a astrologia - de grande importância histórica para aqueles que desejam compreender o desenvolvimento das ideias sobre os astros", explicou na introdução da enciclopédia o autor Ronaldo Rogério de Freitas Mourão.

Mourão explicou na enciclopédia, também, sobre os tipos de verbetes e os títulos deles. "O vocábulo é feito de acordo com as normas da União Astronômica Internacional. A denominação oficial internacional prevalece, adotando-se a ortografia dos nomes próprios usados na língua original", afirma. Além disso, no livre é possível ver as descrições dos verbetes conceituais através de instrumentos matemáticos e astronômicos, nomes próprios e até formações topográficas da Lua, de planetas e satélites do sistema solar.

1.4 Justificativa

Assim como o PNAE, existem vários documentos da área espacial que possuem termos técnicos na linguagem usada para escrever o texto, podendo limitar o entendimento por parte de alguns leitores que não estão a par desse contexto tão específico. Por isso oferecemos um recurso de auxílio ao leitor em sua compreensão com a ajuda dos significados dos termos encontrados no Dicionário Enciclopédico de Astronomia e Astronáutica [29].

1.5 Objetivo Geral

Desenvolver um software que indexe os termos simples e compostos do Dicionário de Astronomia e Astronáutica [29] ao PNAE e a qualquer documento que contenha termos específicos da área espacial, mostrando os significados de maneira clara e instantânea ao leitor.

1.6 Objetivos Específicos

- Estruturar o Dicionário Enciclopédico de Astronomia e Astronáutica para consulta.
- Criar algoritmo para associar o significado dos termos coincidentes do dicionário e do documento do PNAE.
- Mostrar o significado desses termos coincidentes ao usuário de maneira instantânea.

Capítulo 2

Referencial Teórico

2.1 A Indexação de conteúdos textuais

Segundo Borges (2009) [16], "indexar é representar um documento por uma descrição abreviada de seu conteúdo, com o intuito de sinalizar sua essência. É a atividade de selecionar ou definir termos (palavras ou expressões) que descreverão o conteúdo de um determinado documento, sempre levando em consideração uma clientela específica".

Leiva (2012) [14] descreve por meio da Figura 2.1 os diversos tipos de conceitos da Indexação para vários autores

Analisar o conteúdo informacional dos registros do conhecimento e expressar o conteúdo informacional na linguagem do sistema de indexação.	Borko; Bernier (1978, p. 8)
Representação pelos elementos de uma linguagem documentária ou natural, das noções resultantes da análise do conteúdo de um documento para facilitar sua localização	NF Z 47-102 1978
Descrever o conteúdo de documentos ou demandas documentárias para possibilitar a elaboração de estratégias de recuperação mediante conceitos ou assuntos.	García Gutiérrez (1984, p. 105)
Ação de descrever ou identificar um documento em relação ao conteúdo.	ISO 5963-1985
Identificar informação numa entidade de conhecimento (que seja texto ou não) e organizá-la para que esteja disponível num sistema de recuperação.	Cleveland, D.B.; Cleveland, A.D. (2001, p. 97)
Determinar o assunto temático dos documentos e expressar em índices (por exemplo, descritores, cabeçalhos de assunto, números de chamada, códigos de classificação ou índice) para tornar possível a recuperação temática.	Mai (2005, p. 599)

Figura 2.1: Conceitos de Indexação [14].

Existem duas formas de indexação: a manual e a automática. A manual é executada por uma pessoa, enquanto a automática é feita por meio de um programa. Se na manual o autor precisa se preocupar com falhas na indexação, na automática - de acordo com Leiva (2012) [14], não ocorrem fatores externos que podem afetar o próprio sistema. "Um programa de computador indexará sempre igual, bem ou mal, um documento sem que intervenha no contexto. A indexação variará somente quando forem feitas alterações nos parâmetros de análise do sistema".

2.1.1 Sistema de Recuperação de Informação

O estudo do armazenamento e da recuperação automática de documentos é feito por uma subárea da Ciência da Computação. Segundo Cardoso (2000) [8], esses documentos são objetos de dados - geralmente textos. A autora simplificou a estruturação do sistema de Recuperação de Informação (SRI) conforme Figura 2.2

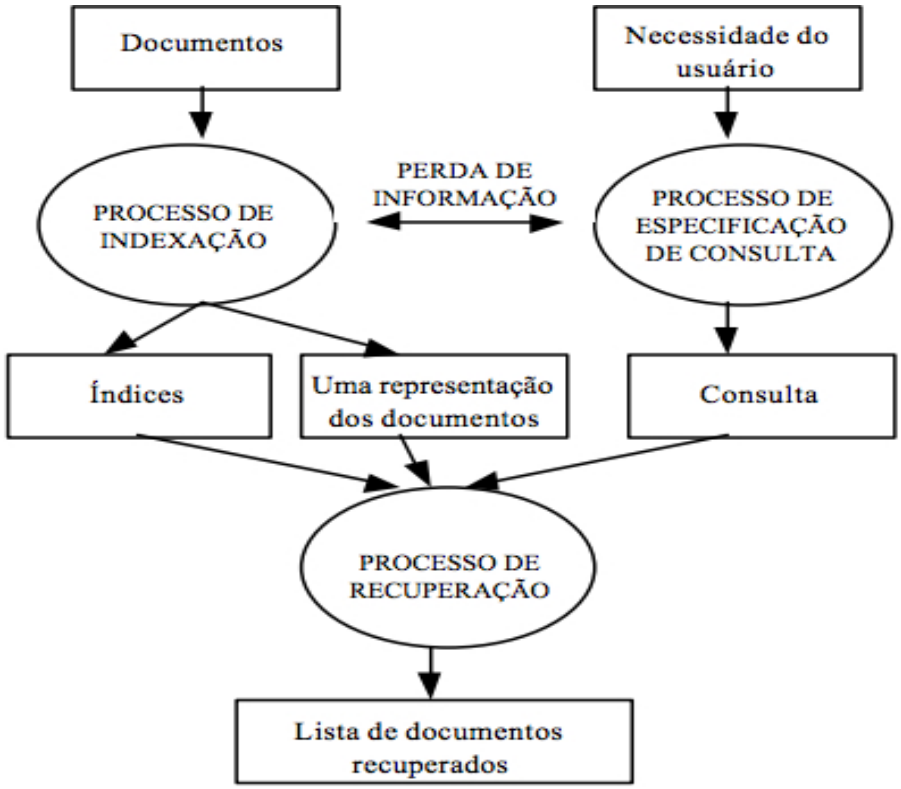


Figura 2.2: Componentes de um sistema de recuperação de informação [8].

Conforme podemos ver na Figura 2.2, para que haja uma recuperação de informação, precisamos de dois elementos básicos: o documento que deve ser recuperado e a necessidade do usuário. A junção dessas duas combinações fazem com que o processo de indexação seja iniciada e o processo de especificação de consulta ao documento seja iniciado. O fato de uma representação dos documentos, dos índices e da consulta do usuário resulta no processo de recuperação e na lista de documentos recuperados.

Cardoso afirma que "o processo de indexação envolve a criação de estruturas de dados associados à parte textual dos documentos, por exemplo, as estruturas de arranjos de

sufixos (PAT arrays) e arquivos invertidos". Ela explica que estas estruturas podem conter dados sobre características dos termos na coleção de documentos, tais como a frequência de cada termo em um documento.

2.1.2 Sistema de Recuperação de Informação x Sistema de Gerenciadores de Bancos de Dados

A grande diferença entre os dois sistemas é que, em um deles, no caso o de recuperação de informações, o sistema busca resgatar uma informação sobre algo, não dados, nem documentos.

Segundo Ferneda (2003) [13], "o sistema gerenciador de banco de dados têm por objetivo a recuperação de todos os objetos ou itens que satisfazem precisamente às condições formuladas através de uma expressão de busca. Já no sistema de recuperação de informação a precisão não é tão estrita". O autor afirma que os sistemas de recuperação de informação lidam com objetos linguísticos - textos - e herdam toda a problemática inerente ao tratamento da linguagem natural. Já o sistema de bancos de dados organiza itens de "informação- dados - que têm uma estrutura e uma semântica bem definidas.

2.1.3 Processo de Indexação

Para que a indexação automática ocorra de forma eficaz, deve ser feita uma pesquisa eficiente com as áreas ideais para cada documento. Por isso a análise conceitual é uma das etapas mais importantes da indexação. Segundo Gomes (2012) [32], "é na análise que consiste a definição do assunto de um documento, para o atendimento às necessidades de recuperação de informação por determinado perfil de usuário". Por exemplo, se há um documento específico da área da arquivologia, é importante que as palavras indexadas ao documento que representam este documento sejam específicas da área, para que as pessoas consigam entender e visualizar melhor o significado das palavras.

Apesar de não haver um consenso extremamente definido sobre o processo de indexação em conteúdos de texto, no geral é determinado pela relação com a representação da informação e o assunto que consta no documento.

De acordo com Chaumier (1988) [18], existem quatro etapas no processo da indexação:

- Conhecimento do conteúdo do documento: com uma leitura rápida nas informações mais importantes como o título, a introdução, as legendas de ilustrações, tabelas e gráficos, etc.
- Escolha dos conceitos: "exige uma verdadeira análise conceitual do documento (...) Ela poderá também guiar-se por um procedimento lógico adaptado à área de assuntos a que se refere: definição dos fenômenos estudados, as teses apresentadas, dos argumentos utilizados, dos resultados obtidos, etc".
- Tradução dos conceitos escolhidos: Tradução feita nos termos da linguagem documentária utilizada pelo serviço de documentação. "Na maioria das vezes, a linguagem documentária será um "thesaurus"próprio ao centro de documentação ou setorial".

- Incorporação dos elementos sintáticos: A importância e o peso dos conceitos em relação ao conjunto do documento a indexar também devem ser importantes e finalizam a última operação do processo de indexação.

2.1.4 Modelos de SRI

A análise conceitual e os diferentes tópicos que devem ser analisado na indexação e no sistema de recuperação de informações mostram que há diversos modelos a serem seguidos para determinar como um documento deve ou não ser indexado.

Estes modelos, também conhecidos como "Quantitativos", facilitam o processo. Desta forma, Leiva (2012)[14] apud Peña, Baeza-Yates e Rodriguez Muñoz apresenta os seguintes modelos.

Busca Por Texto Livre

Leiva (2012) [14] explica que, "o modelo de busca por texto livre, que não exige a indexação dos documentos portanto, não se mantém nos índices. A informação está diretamente acessível tal como foi apresentada e as questões na base de dados são feitas através de combinações de cadeias de caracteres".

Booleano

É conhecido como um dos modelos de sistema de recuperação de informações mais usados. "O sistema combina os termos utilizados pelo usuário na pergunta através dos operadores Y (e) O (ou) e NO (não) aos termos (não vazios) presentes na base de dados. Não dispões de mecanismos para especificar em que grau um termo é mais representativo para um documento do que outro".

Booleano Extendido

A diferença entre os dois é que, neste, pode-se atribuir um peso aos termos de indexação durante as etapas do processo de definição. "Pode ser representado da seguinte maneira: Ap1 OB p2; Ap1 YB p2; Ap1... P2 B onde A e B representam os conjuntos indexados pelos respectivos termos, enquanto que P1 e P2 são os pesos aplicados a cada conjunto".

Modelo de Espaço Vetorial

O Modelo de Espaço Vetorial é a soma entre a expressão de busca e a associação de pesos em relação aos termos de indexação. "É representado no sistema por duas listas ordenadas numericamente. Estas listas correspondem, por um lado, aos pesos atribuídos ao documento (em função da soma dos valores atribuídos aos termos de indexação que representam o documento), e por outro lado, aos valores da pergunta do usuário (em função dos pesos atribuídos aos termos de indexação da pergunta)".

Modelo de Lógica

Segundo Leiva (2012) [14], "o modelo de lógica difusa se baseia na inexistência de um limite definido que permita agrupar algo a um único grupo, mas sempre existem

características que podem incluí-lo em algum lado. O conjunto difuso D é definido como: D= $\{x$ÎU, $f(x)/(x) > 0\}$ em que x são os elementos do conjunto, $\in$ é o símbolo que indica pertinência, U é o conjunto universal e f(x) a função de pertinência que define grau de pertinência do indivíduo ao grupo".

2.2 Etapas de Pré Processamento

Para fazer o processamento de texto é preciso de alguns conceitos que podem ajudar na hora abri-lo e acessá-lo de maneira correta. Um texto, quando aberto, pode trazer consigo vários caracteres e informações indesejáveis, tais como *tags* e metadados vindos de outros programas, que "sujam"o texto original e podem ser confundidas com as informações que são realmente relevantes ao contexto em questão. Por isso, antes de se processar um texto, é preciso fazer o que chamamos de pré processamento. Conforme explanado na seções seguintes.

2.2.1 *Tokenization*

Tokenization é a tarefa de dividir uma *string*, ou palavra, em unidades linguísticas identificáveis, que constituem parte dos dados da linguagem [5]. Quando abrimos um texto cru, texto em sua forma original, ele é armazenado em uma grande *string* que contém todas as palavras, espaços em branco, pontuação, etc.

Para facilitar a manipulação dos dados, existe a necessidade de dividir o texto em unidades menores, que chamamos de *tokens*. Os *tokens* são frequentemente denominados como "termos"ou "palavras", porém existe uma diferença entre essas as denominações. Segundo Manning et. al. (2009) [11], é preciso fazer uma distinção entre os conceitos de *token*, tipos e termos:

- "Token"é uma instância de uma sequência de caracteres em um determinado documento, que são agrupados de acordo com uma unidade semântica significante para o processamento, ou seja, o escopo do *token* pode variar de acordo com o objetivo do processamento;
- "Tipo"é a classe de todos os *tokens* contendo a mesma sequência de caracteres. Na frase, "nas casas e nas ruas", por exemplo, existem cinco *tokens*, porém somente 4 tipos, pois existem dois tipos duplicados ("nas");
- "Termo"é o mesmo que "tipo", porém, para ser considerado como tal, precisa ser significante para o contexto em questão, ou seja, precisa estar contido dentro do léxico utilizado.

Na Figura 2.3, podemos ver um exemplo de como ocorre o processo de *tokenization* no Python, utilizando expressões regulares. Neste exemplo, temos uma *string* que contém um trecho de um determinado texto e queremos separá-lo em *tokens*, utilizando o espaço em branco como um delimitador. O método *split*() da biblioteca "re"importada no Python, recebe uma expressão regular como primeiro parâmetro, que no nosso caso é um espaço em branco . O segundo parâmetro da função recebe a *string* que irá passar pelo processo de *tokenization*, nesse caso identificada como "texto".

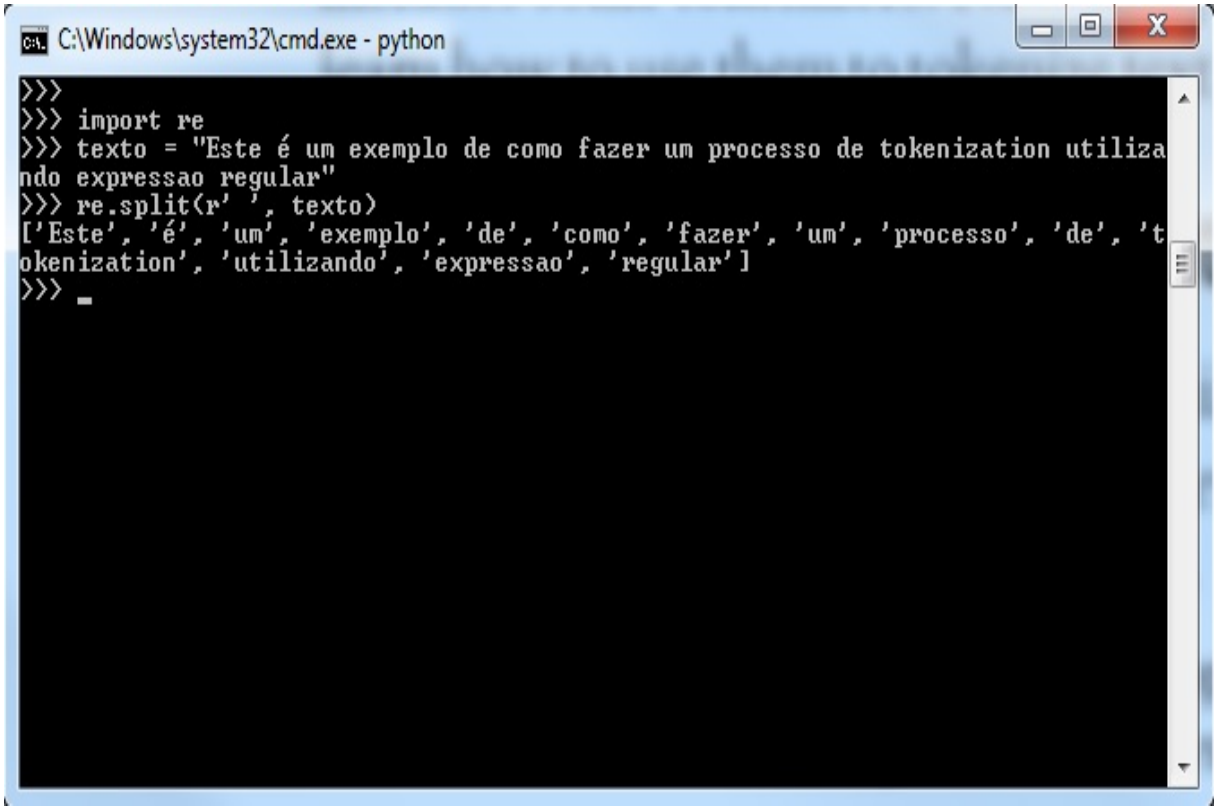


Figura 2.3: Exemplo de *tokenization* utilizando expressão regular.

2.2.2 Expressões Regulares

Muitas tarefas de processamento linguísticos envolvem o casamento de padrões. Expressões regulares são um método poderoso e flexível para descrever o padrão que estamos interessados em uma cadeia de caracteres [5]. Segundo Sipser (1997) [28], as expressões regulares tem um papel fundamental em certas aplicações da ciência da computação como alguns utilitários tais como AWK e GREP no UNIX, linguagens de programação como PERL e editores de texto. O Python também possui suporte a expressões regulares de maneira bem poderosa.

Para usar expressões regulares no Python, precisamos importar a biblioteca "re", usando o comando "import re". Na Figura 2.4, podemos ver um exemplo de seu uso dentro do ambiente. É possível observar que dado uma lista de palavras podemos buscar por determinados padrões, de acordo com nossa necessidade.

No exemplo da Figura 2.4, dado uma lista de palavras identificada pelo nome de "tarefas", procura-se por palavras que terminam com a letra "r", geralmente contidas no final de verbos da língua portuguesa. Após o processo da busca pela expressão regular "r$"(Veja Tabela 2.1), obtemos uma lista de termos do que provavelmente irá indicar todos os verbos, usados na forma infinitiva, no texto dado.

2.2.3 *Stopwords*

Em tradução livre, Stopwords são "Palavras de Parada", que podem ser consideradas irrelevantes para o conjunto de resultados a ser exibido em uma busca.

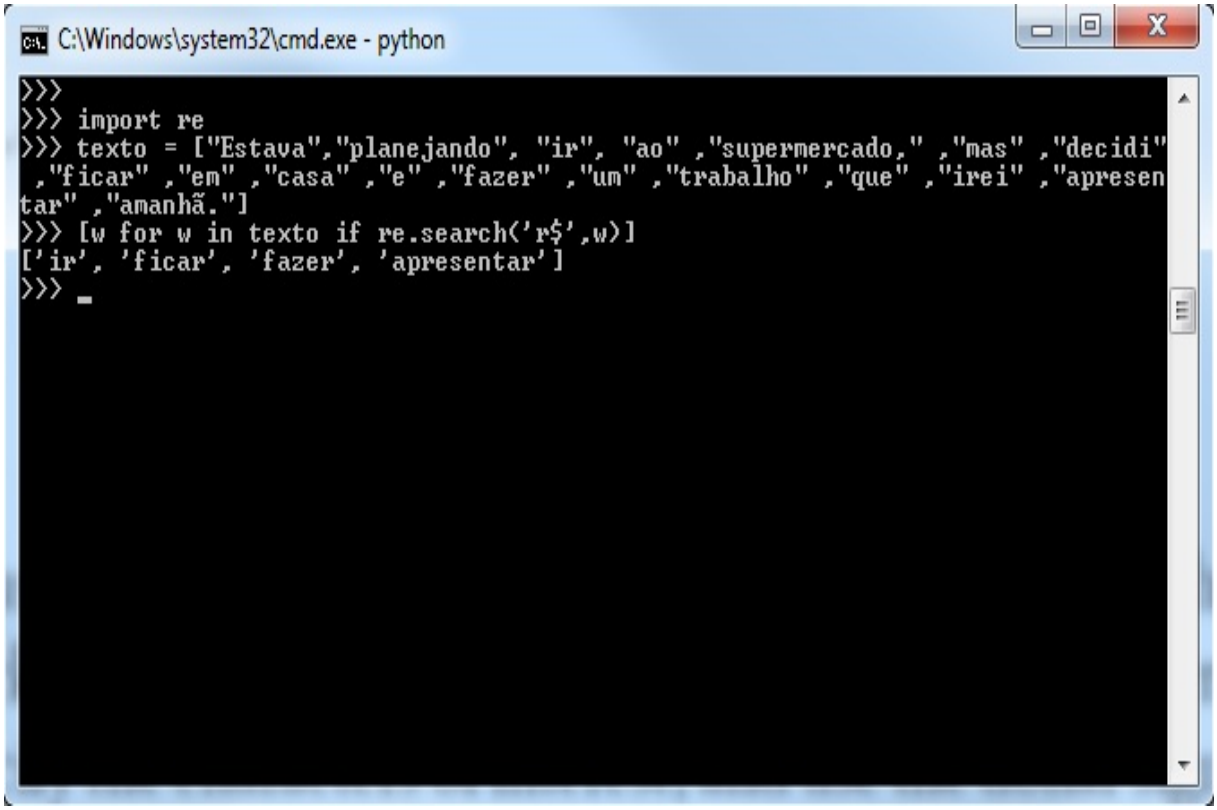


Figura 2.4: Exemplo do uso de expressão regular. Busca por verbos no infinitivo, caracterizados por conterem "r"como última letra.

Estas palavras são, basicamente, artigos e preposições que não têm valor semântico dentro do contexto. Ou seja, elas podem ser retiradas do processamento de texto. Estas palavras são muito utilizadas por alguns profissionais em otimização de sites.

Segundo o Neto (2013) [6], estas palavras, que não carregam significado na linguagem natural, podem ser ignoradas - já que se tornam não pesquisáveis.

Algumas das principais *stopwords* utilizadas são artigos, preposições, advérbios, como: a, à, agora, ainda, alguém, algum, ampla, antes, ao, após, aquelas, cada, com, como, contudo, da, daquele, de, dela, dele, depois, dessa, dever, disso, diz, do, e, é, ela, ele, enquanto, essas, esses, estamos, eu, fazendo, feito, foram, fosse, grande, há, isso, já, lá, lhe, lo, mesma, mesmo, meu, minhas, muitas, não, nenhum, nessa, nós, nossas, o, outra, outro, para, pela, pelos, perante, pode, porque, primeiro, própria, quais, qual, quando, quanto, quem, são, sejam, sempre, será, seus, talvez, também, tendo, tinham, toda, tu, tudo, uma, vez, e vós.

No caso do Indexer, todas essas palavras ficam armazenadas em uma tabela dentro do banco de dados, que será consultada sempre que necessário pelo programa. Os critérios para a consulta dessa tabela de *stopwords* será explicada melhor na proposta do trabalho na seção 4.2.5.

Tabela 2.1: Metacaracteres básicos de expressões regulares [5].

Operador	Comportamento
.	*Wildcard*, casa com qualquer caracter
ˆabc	Casa com padrão abc no começo de uma *string*
abc$	Casa com padrão abc no final de uma *string*
[abc]	Casa com um elemento do conjunto de caracteres
[A-Z0-9]	Casa com um elemento do intervalo do conjunto de caracters
ed|ing|s	Casa com uma das strings (disjunção)
*	Zero ou mais itens anteriores (*Kleene Closure*)
+	Um ou mais itens anteriores
?	Zero ou um dos itens anteriores
{n}	Exatamente n repetições, onde n é inteiro-não-negativo
{n,}	No mínimo n repetições
{,n}	Não mais do que n repetições
{m,n}	No mínimo m e no máximo n repetições
a(b|c)+	Parênteses que indicam o escopo dos operadores

2.3 O Léxico

Segundo Barros (2003) [15], o léxico consiste em dicionários com os termos utilizados pelo sistema no processamento de textos. Cada termo (ou palavra) no léxico pode estar associado às suas características.

A expressão "base de dados lexical"também é usada para se referir a uma coleção de informações lexicais, organizada de maneira estruturada e acessível a sistemas de recuperação de informação [25].

O propósito dos léxicos (ou dicionários) é prover informações sobre as palavras, tais como etimologia, morfologia, sintaxe, entre outras. Eles fornecem as definições de seus sentidos, trazendo conhecimento sobre a linguagem usada [22].

Segundo Wertheimer (1995) [7], podemos classificar os dicionários em cinco categorias:

- Convencionais, com verbetes em ordem alfabética;
- Analógicos, que organizam os itens lexicais de acordo com seu significado;
- Etimológicos, que se ocupam exclusivamente da origem das palavras;
- Morfológicos, que apresetam as formas flexionais dos lexemas;
- Sinônimos e antônimos, quer armazenam palavras semelhantes ou contrários em seu sentido.

Quanto ao objetivo, os dicionários também podem ser classificados em 5 categorias, explica Wilks (1996) [1]:

- Dicionários padrão, que explicam os significados das palavras;
- Thesauri, que apontam relacionamentos entre os itens lexicais;

- Dicionários bilíngües, que buscam relacionar dois idiomas em nível de equivalência de sentidos das palavras;
- dicionários de estilo, que dão orientações sobre o bom uso das regras gramaticais;
- Dicionários de concordância, que são essencialmente ferramentas escolares.

Existem ainda os léxicos com capacidade de serem legíveis e tratáveis por máquina [1]. Nos "dicionários legíveis por máquina"(machine-readable dictionarie — MRD), informações lexicais de larga escala podem ser extraídas automaticamente, melhorando em muito a uniformidade e a consistência da informação. Já no tratamento de dicionários, os MRDs não conseguem atender aos requisitos, por isso, surge o conceito dos "dicionários tratáveis por máquina" (machine-tractable dictionaries — MTDs). Os MTDs contém uma grande quantidade de informações linguísticas, o que torna viável a conversão de um dicionário existente em uma forma apropriada [25].

Capítulo 3

Referencial Tecnológico

Neste capítulo será apresentado o referencial teórico usado para o desenvolvimento deste trabalho. As tecnologias e conceitos englobados serão detalhados nas seções seguintes. As tecnologias foram estruturados, como mostra a Figura 3.1, em *back end* e *front end*, que são respectivamente a etapa inicial do trabalho que executa todo o processamento de linguagem natural e a etapa final que é responsável pela apresentação dos dados processados.

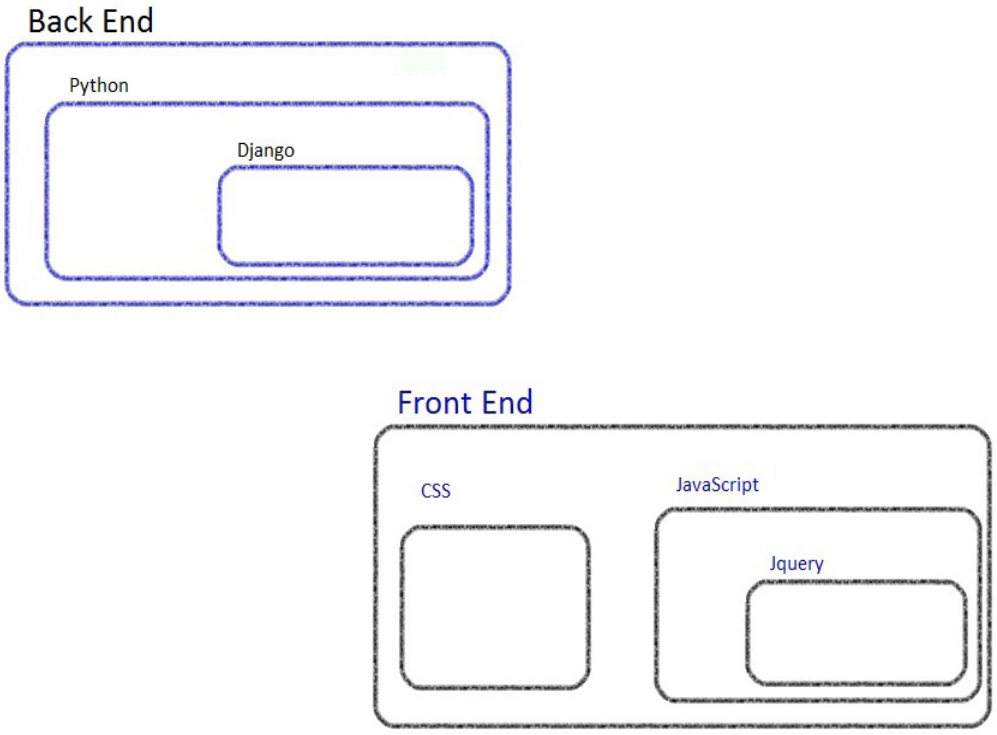


Figura 3.1: Estrutura do Projeto.

3.1 Linguagem Python

Python, representado na Figura 3.2, é uma linguagem de altíssimo nível que segundo Neto e Galesi (2010) [30], seu desenvolvimento começou em 1989 pelo holandês Guido Van Rossum durante as festas de final de ano. A série humorística britânica Monty Python's Flying Circus do grupo humorístico Monty Python foi usada como inspiração para o nome da linguagem, que foi criada sobre o ideal de "Programação de Computadores para todos" [20]. Guido queria criar uma linguagem gratuita, de código aberto e que agregasse características importantes de várias outras linguagens de programação. Uma de suas principais características é sua sintaxe simples e clara, onde obriga o programador a indentar seu código de acordo com o escopo de cada estrutura de programação. Sua licença, chamada de CPython foi mudando ao longo do tempo e hoje utiliza uma licença considerada compatível como a GPL, muito utilizada em diversos softwares livres. Serão apresentadas a seguir algumas características da linguagem.

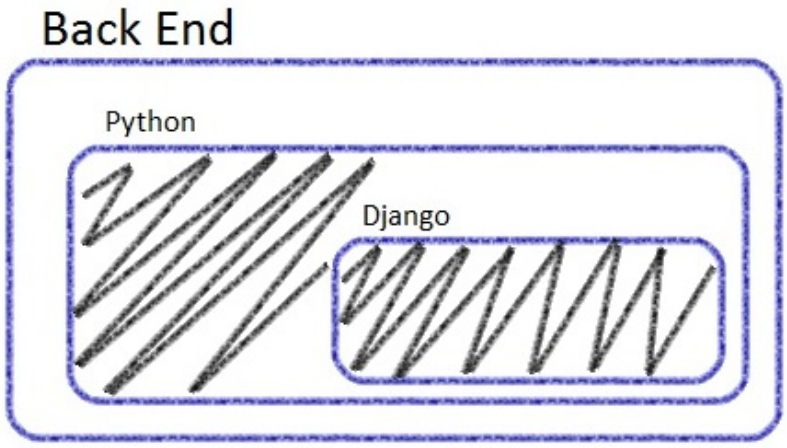


Figura 3.2: Python representado pela área hachurada.

Existem duas maneiras de uma linguagem alto nível ser processada em linguagem de máquina: por um interpretador ou por um compilador. Um interpretador lê a linguagem de alto nível e a executa ao mesmo tempo, ou seja, processa o programa lendo as linhas de código e computando suas instruções (Veja Figura 3.3).

Um compilador lê o programa inteiro e o traduz completamente antes de ele ser executado. Nesse caso o programa de alto nível é escrito em, o que é chamado de código fonte e o programa traduzido é chamado de código objeto ou executável. Uma vez que o programa é compilado, é possível executá-lo repetidamente sem que seja necessário

Figura 3.3: Interpretador [19].

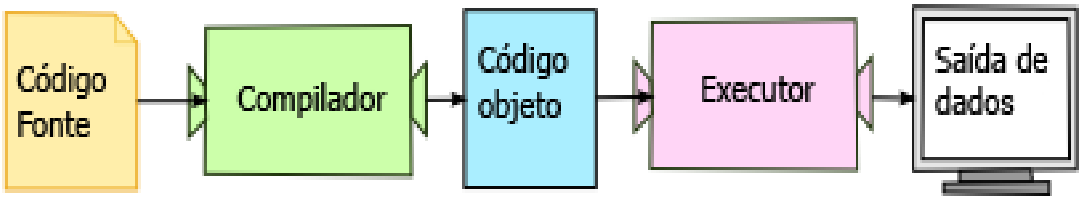


Figura 3.4: Compilador [19].

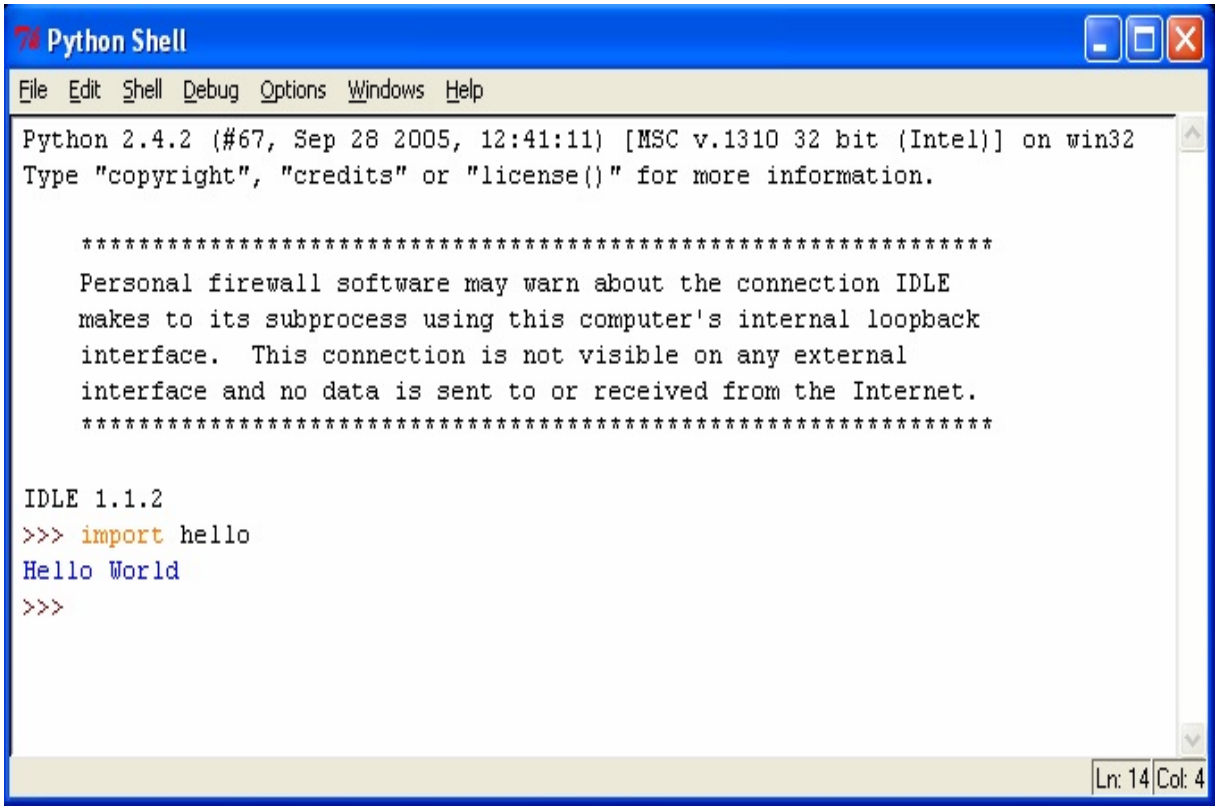


Figura 3.5: Interpretador interativo IDLE [20].

traduzi-lo novamente (Veja Figura 3.4). Muitas linguagens modernas usam ambos os processos. Elas são primeiramente compiladas em uma linguagem de baixo nível, chamada de byte code, e então interpretadas por um programa chamado máquina virtual [19]. Python usa ambos os processos, mas como os programadores interagem com a linguagem, ela é frequentemente chamada de interpretada.

3.1.1 Python como Linguagem Interpretada

É possível executar os programas escritos em Python através do interpretador interativo IDLE (Python's Integrated Development Environment), que é disponibilizado junto com sua distribuição oficial. Para executar um programa escrito em Python, basta chama-lo pelo interpretador utilizando o comando *"import"*, como mostrado na Figura 3.5. Todos os arquivos da linguagem Python possuem a extensão do tipo ".py".

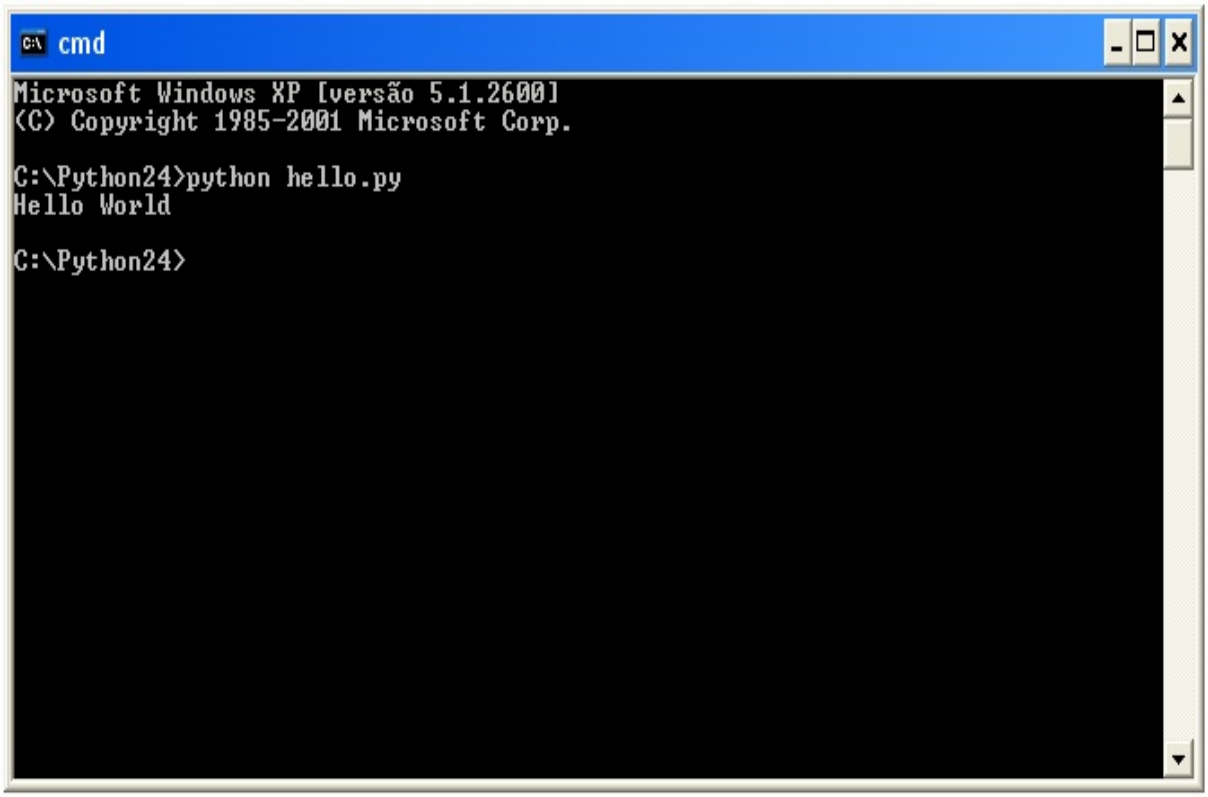


Figura 3.6: Prompt de comando do Windows [20].

O programa invocado pelo interpretador na Figura 3.5 simplesmente imprime uma mensagem "Hello world!"na tela, um exemplo de um programa básico do Python escrito em apenas uma linha. Para chamarmos o programa pelo interpretador não é necessário explicitar a extensão do arquivo (.py), basta, somente, chamar seu nome principal.

Também é possível invocar um programa Python chamando-o pelo prompt de comando do Windows. Para isto, é necessário chamar o Python e passar o arquivo .py como parâmetro, usando o comando Python nomeDoArquivo.py, como mostrado na Figura 3.6.

Note que o programa que executamos diretamente é o interpretador Python (o prompt), não havendo necessidade de um compilador. Passamos como parâmetro o próprio nome do arquivo com código-fonte hello.py. Não há o passo de geração de executável; o interpretador transforma o programa especificado à medida que é executado. O nome do programa é sempre considerado um primeiro parâmetro. Existe a possibilidade de passar mais parâmetros, desde que sempre depois do nome do programa a ser interpretado.

3.1.2 Interpretador Interativo

Python está disponível para diversas plataformas. O interpretador interativo do Python está disponível em todas elas. Ele recebe o adjetivo de interativo devido à facilidade que o programador tem para testar trechos de código, antes mesmo de colocar o programa para executar e evitando a necessidade de se criar outro arquivo somente para teste. Como podemos ver na Figura 3.7, é possível, por exemplo, testar o retorno de um

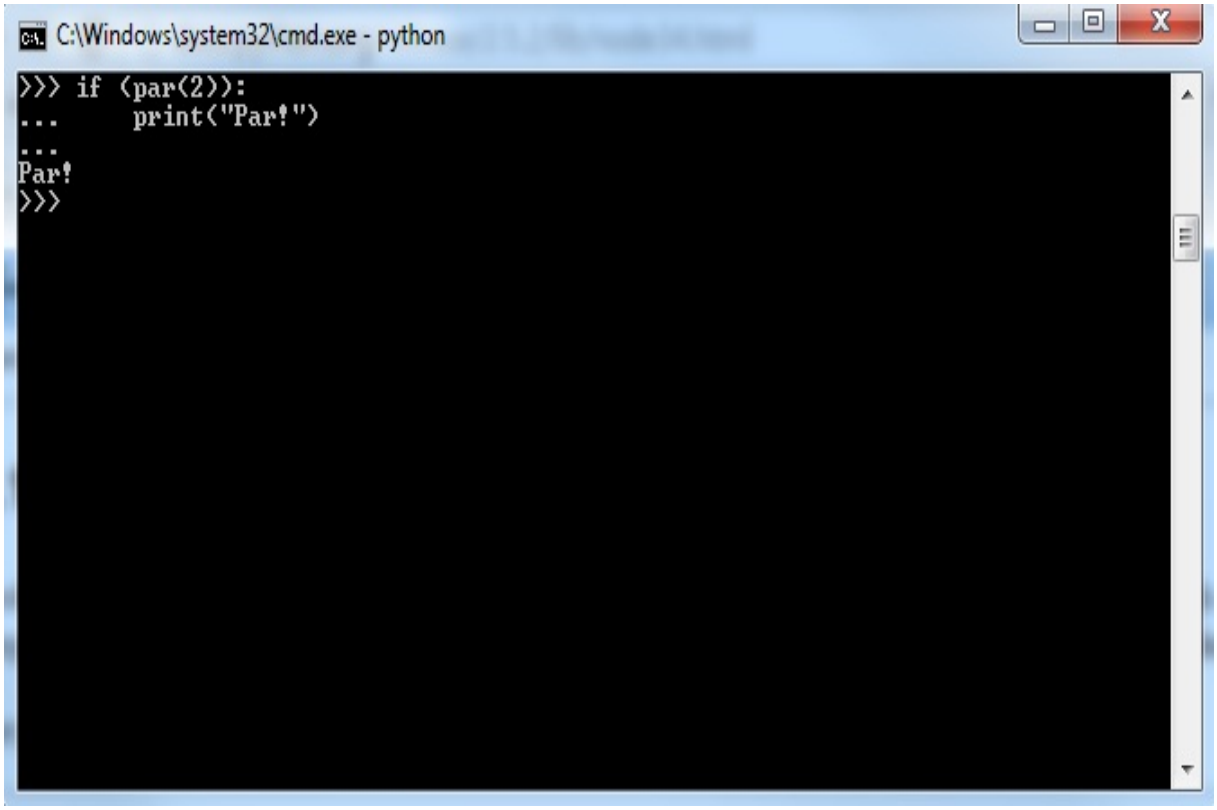


Figura 3.7: Testando retorno do método par().

método de maneira instantânea. O símbolo » indica que o interpretador está aguardando um comando. O interpretador permite que o programador teste seu código e observe linha por linha como a execução de seu programa está ocorrendo já que sua interpretação e execução ocorrem de maneira imediata. Isso o torna muito conveniente para testes, no entanto, ele não armazena os comandos digitados, por isso, a utilização do modo interativo se torna viável apenas para testes e procedimentos simples. Geralmente, para programas extensos, o código fonte é armazenado em arquivos separados de extensões do tipo .py, chamados de módulos, onde podem ser referenciados por outros módulos, por meio do comando "import arquivo.py".

3.1.3 Tipagem dinâmica

O Python utiliza um conceito um pouco diferente de linguagens procedurais em relação à declaração de variáveis. Normalmente, a primeira coisa que devemos fazer para declarar uma variável é definir que tipo de dado essa variável consegue armazenar. Para isso, escreve-se primeiro o tipo e depois um identificador para o espaço de memória alocado, assim definimos a variável. Em Python essa declaração se dá de maneira dinâmica, ou seja, o espaço de memória alocado só tem seu tipo definido em tempo de execução. É necessário, apenas, que se crie um identificador para essa variável e, a partir daí, assim que ocorre uma atribuição, a máquina virtual do Python se encarrega de definir o tipo de dado a ser armazenado. Apesar das variáveis serem denominadas de tipagem dinâmicas,

elas sempre assumem um único tipo em um determinado momento, ou seja, elas só são definidas em tempo de execução, mas isso não significa que elas não possuem tipo (a chamada tipagem fraca). Com a tipagem dinâmica, ganha-se a possibilidade de se criar métodos com parâmetros de tipo indefinido, o que evita a restrição do que será passado como argumento do método.

3.1.4 Controle de bloco por indentação

Outra notável caraterística do Python é a forma como os blocos de código são definidos. Não existem quaisquer marcações explícitas para delimitar o início e o fim de funções, estruturas condicionais, estruturas de repetição, etc. As estruturas de programação são delimitadas apenas pelo sinal de (':') dois pontos e pela própria indentação do código, ou seja, não existem marcadores do tipo 'begin' ou 'end'. Para se definir o escopo de uma função, por exemplo, usamos os níveis de indentação como delimitadores, aumentando em um nível à direita para indicar o começo da função e diminuindo um nível à esquerda novamente para indicar que o bloco de código terminou. Essa regra obriga que o programador mantenha sempre seu código legível e bem organizado, pois, nesse caso, cada espaço em branco é levado em conta para a interpretação e execução do código, o que por um lado ajuda na indentação perfeita, mas por outro exige costume e controle mais formal por parte do programador.

3.1.5 Orientação a Objetos

Segundo Lutz (2009) [26], Python é uma linguagem orientada a objeto da cabeça aos pés. Seu modelo de classes suporta avançadas noções de orientação a objetos como polimorfismo, sobrecarga, herança múltipla, etc. No contexto da sintaxe e escrita do Python, a orientação a objetos é muito simples de ser aplicada, pois sua sintaxe ajuda muito a leitura e entendimento do código, mesmo para quem não tem muita prática com outras linguagens de programação.

Apesar da Orientação a Objetos no Python ser intuitiva e de grande utilidade para a estruturação de um projeto, principalmente quando estamos falando de proporções grandes , ela não faz parte da essência da Linguagem. Assim como em C++, a Orientação a Objetos é adicional. É possível escrever código Python também utilizando o paradigma procedural, isso vai depender, é claro, da necessidade do programador.

3.2 Django

O Django é um framework de rápido desenvolvimento, fácil e objetivo. Com ferramentas para desenvolvimento ágil na web e escrito em Python (Veja Figura 3.8), ele adere ao princípio DRY - "Don't Repeat Yourself" - ou seja, evita que o código seja feito diversas vezes, aproveitando o que já foi apresentado [12].

Mauro Rocco [27] descreve o framework como um suporte para tradutor de textos, formatações de dados, números, horários e esclarece que tudo isso faz com que seja possível criar um projeto web multilingual de forma clara e simples.

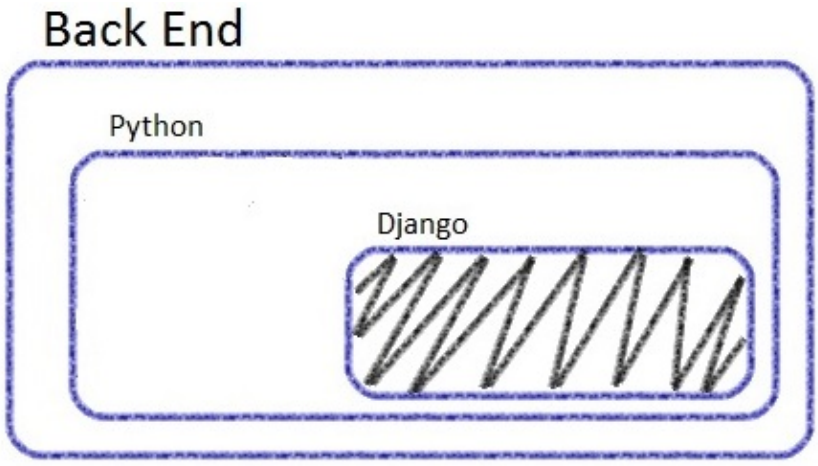


Figura 3.8: Django representado pela área hachurada.

A estrutura do Django é uma variação do padrão de arquitetura Model - View - Controller (MVC) e pode ser definida como Model - Template - View (MTV). O modelo é a aplicação de dados, a visão significa quais dados estão presentes na aplicação e o template é a forma que esses dados são apresentados [27].

3.2.1 Os componentes do Django

Este framework possui diversos componentes que aprimoram ainda mais o desenvolvimento do programa. Segundo o site do Django [12], os principais itens com as respectivas descrições que podem ser encontrados nele são:

- Mapeador Objeto-Relacional: Define todo o modelo de dados em Python. Além disso, este componente faz com que o desenvolvedor tenha um API livre, que permite a "abstração de banco de dados e permite a criação, o recebimento a atualização e a exclusão de objetos";
- Administração automática da Interface: Esse componente desempenha os metadados do model para fornecer uma interface pronta para a produção que os redatores podem usar para adicionar conteúdos ao site;
- Sistema de Template: Essa função tem variedade de tags e filtros embutidos projetados para direcionar a lógica da aplicação;
- Sistema de Cache: Esse componente do Django permite que sejam guardadas páginas dinâmicas para que elas não tenham que ser calculadas a cada requisição. O framwork oferece diversos níveis de granularidade de cache: De saída de views específicas, das partes difíceis de produzir ou do site inteiro;
- Internacionalização: Permite que uma aplicação web ofereça os conteúdos e funcionalidades em múltiplas línguas adicionando hooks - conhecidos como translation strings - ao código Python e aos templates.

3.3 JavaScript

JavaScript, representado no diagrama da Figura 3.9, é uma linguagem interpretada originalmente desenvolvida por Brendan Eich da Netscape sob o nome de Mocha, mas em setembro de 1995 teve seu nome mudado em um anúncio conjunto com a Sun Microsystems para o que conhecemos hoje como JavaScript. O nome escolhido foi resultado do suporte que a NetScape adicionou ao Java em seu navegador e por uma questão de marketing [9].

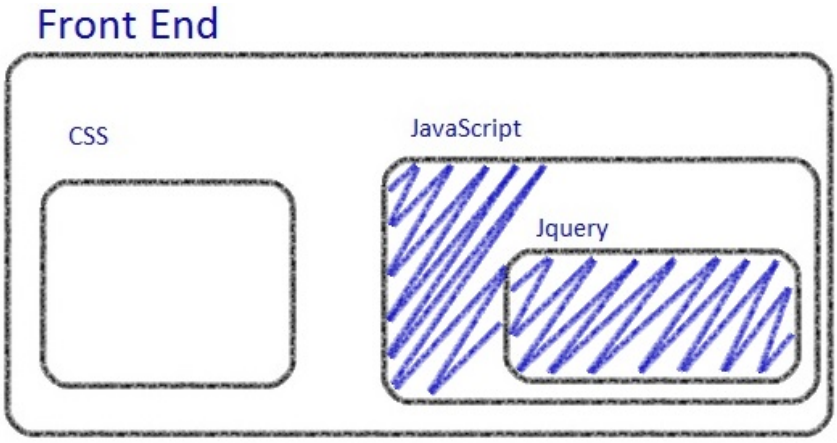


Figura 3.9: JavaScript representado pela área hachurada.

De acordo com David Flanagan [9], "é uma linguagem de programação leve, interpretada com recursos de orientação a objeto (...) Sintaticamente, o núcleo da linguagem JavaScript assemelha-se às linguagens C, C++ e Java, contendo construções de programação como a instrução if, o laço, while e o operador &&". Ainda de acordo com o autor, as variáveis de JavaScript não precisam ter um tipo especificado.

Sendo uma linguagem interpretada pela web - e, assim, processada para o cliente, - o JavaScrip é muito utilizada para incluir, na web, todo o tipo de interação com imagens e animações. O código pode ser feito no corpo da página HTML, como uma função ou como um arquivo separado com o sufixo ".js".

Com grande aceitação, o JavaScript tem funções incluídas nas páginas HTML que vão de acordo com o Modelo de Objeto de Documentos da página, como uma nova janela com controle programático sobre a sua posição, atributos e tamanho; como a validação de valores de um formulário para garantir que são corretos antes de serem enviados ao servidor e como a mudança de imagens à medida em que o mouse se movimenta sob elas [31].

O script gerado é capaz de interagir com os objetos e tags do HTML, sendo por isso considerada uma ferramenta poderosa pelos desenvolvedores web [23].

3.3.1 Jquery

Jquery é uma biblioteca JavaScript, representada no diagrama da Figura 3.10, desenvolvida para simplificar os scripts client side que interagem com o HTML. Ela foi lançada em 2006 por John Resig e é a biblioteca javascript mais utilizada nos dias de hoje [21].

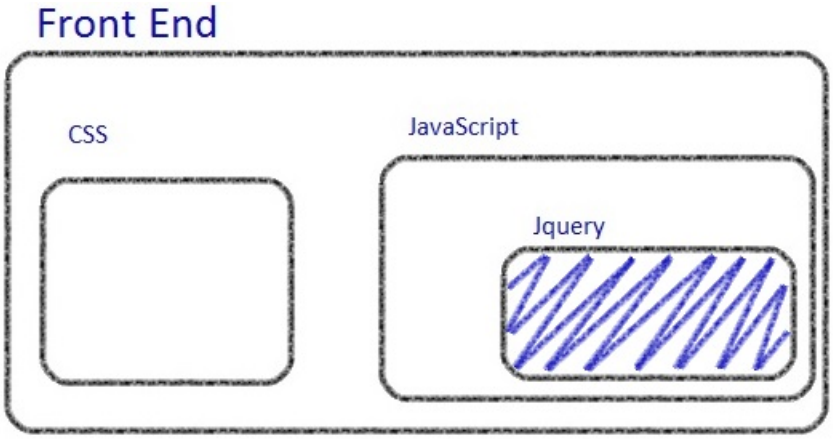


Figura 3.10: Jquery contido dentro de JavaScript e representado pela área hachurada.

De acordo com David Flanagan [10], ela faz com que seja fácil achar elementos em um documento e manipula-los adicionando conteúdo, editando os atributos do HTML e as propriedades do CSS, definindo eventos manipuladores e de performances.

A biblioteca jQuery tem diversas funcionalidades e, por isso, é vista como uma das melhores opções por desenvolvedores web e por designers. Segundo o autor Jake Rutter [21], jQuery é incrível por não ser necessário o conhecimento de programação avançado para executar a manipulação do DOM. Ele afirma que, algumas áreas avançadas que exigem um conhecimento prévio de JavaScript, como a utilização dos métodos Ajax para a obtenção e postagem de conteúdo, a criação de plug-ins jQuery personalizados e o uso de sites móveis [21].

Rutter descreve os principais recursos da biblioteca como:

- Eventos que incluem interações de mouse, teclado, formulário e do usuário;
- Efeitos que incluem exibir/ocultar, deslizar, transição gradual e animações personalizadas;
- Animações que permitem a movimentação de objetos com CSS e efeitos nativos;
- Métodos Ajax para a interface com o processamento de formulários no servidor com XML e JSON;
- Extensibilidade para a criação de plug-ins pessoas que ampliam a base da API da jQuery;
- Manipulação do DOM;
- Manipulação da CSS;
- Utilitários que fornecem a detecção de navegadores e interfaces mais fáceis para funções comuns do JavaScript.

3.4 CSS

O Cascading Style Sheet, mais conhecido como CSS, é uma folha de estilo em cascatas. É um mecanismo inserido no HTML que estabelece a estética da página web ou do programa, definindo, por exemplo, as fontes, as cores e os espaçamentos [17], o que o torna uma linguagem front end (Veja Figura 3.11).

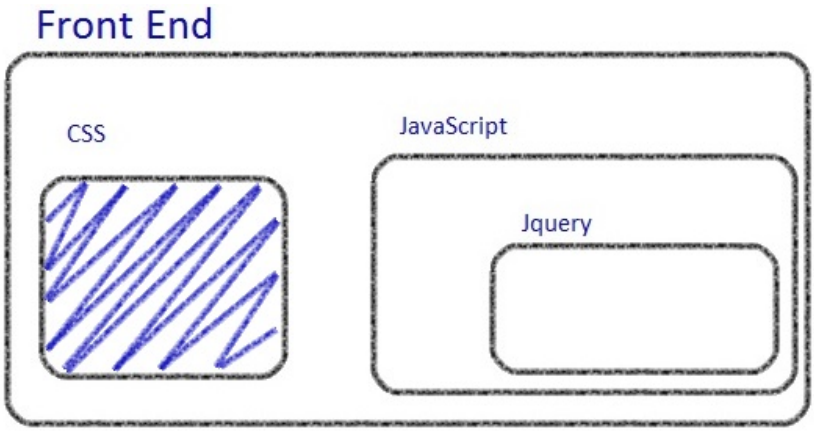


Figura 3.11: CSS representado pela área hachurada.

De acordo com Harvey M. Deitel [24], os objetivos do CSS são:

- Assumir o controle da aparência de um site Web através da criação de folhas de estilo
- Usar uma folha de estilo para dar todas as páginas de um site Web a mesma aparência e comportamento
- Usar o atributo class para aplicar estilos
- Especificar a fonte, o tamanho e a cor precisos e outras propriedades do texto exibido
- Especificar elementos de fundo e cores
- Compreender o modelo de caixa capaz de controlar as margens, bordas e os enchimentos
- Usar folhas de estilo para separar a apresentação do conteúdo.

De acordo com Hugo Rossetti Savoia [17], há duas maneiras de anexar o CSS em uma página: uma delas é referenciando dentro do head o arquivo .css e a outra é incluindo dentro do head a tag style e colocando todo o código CSS dentro dela.

O CSS é um dos responsáveis por diferenciar o layout de um site ou de um programa com o conteúdo de texto do mesmo [17]. Por exemplo, o desenvolvedor irá utilizar o CSS para ajustar o espaçamento e as margens e vai usar a linguagem HTML para a estrutura de um documento, como o corpo texto.

Segundo Deitel [24], essa separação da estrutura da apresentação permite uma maior facilidade de gerenciamento e torna mais fácil efetuar mudança no estilo de seus documentos.

A sintaxe do CSS é dividida basicamente em duas partes: a primeira é o seletor, que identifica que elemento dentro do programa vai ser estilizado, e a segunda a declaração - que é composta por propriedade e valor, definindo a forma de estilização. A propriedade, por sua vez, estebelece o alvo que vai ser estilizado como, por exemplo, a fonte. E o valor define "qualificação da propriedade" [17], no caso - a fonte a ser usada, como a Times.

Desta forma, a estrutura do CSS é escrita do seguinte jeito: seletor {propriedade:valor}.

Capítulo 4

Proposta

Neste capítulo encontram-se informações detalhadas sobre a proposta do projeto em questão, que foi responsável pelo desenvolvimento do software batizado de "Indexer". O nome foi inspirado exatamente na sua função principal, que é a indexação de referência de palavras. No caso do Indexer a idéia é que a partir de uma palavra chave, seja ela um termo simples ou um termo composto, se consiga recuperar informação de uma determinada fonte léxica que seja significativa para a descrição dessa palavra chave dentro do seu contexto.

O Indexer desempenha dois papéis principais, o primeiro é a identificação, dentro de um documento escrito em linguagem natural portuguesa, das palavras que fazem parte de um contexto específico, a partir de uma fonte léxica, neste caso, no documento do PNAE [2] e no contexto da área espacial utilizando o Dicionário Enciclopédico de Astronomia e Astronáutica [29]; o segundo é recuperar informações que descrevem as palavras identificadas entre o documento e o léxico, a fim de agregar valor descritivo às palavras de cunho técnico e específico.

Devido a grande importância para soberania do Brasil, o PNAE e o relatório que descreve todas essas diretrizes governamentais em relação à área espacial brasileira se tornou um documento de extrema notoriedade para a sociedade e a todos que estejam interessados nas ações tomadas pelo governo nesse campo [2].

Por se tratar de um documento público, ou seja, aberto à toda a sociedade, porém com grande uso de terminologias específicas, surgiu a necessidade de usarmos o domínio espacial para fazer a indexação das palavras chaves deste documento com o intuito de identificar as palavras do domínio em questão e enriquecer o valor descritivo delas, mostrando seus respectivos significados ao usuário leitor.

Neste projeto, estamos usando o Dicionário Enciclopédico de Astronomia e Astronáutica [29] como fonte de informação para a identificação das palavras do domínio espacial. O dicionário está sendo usado como um léxico para o nosso sistema. Ele foi transformado de um simples documento em formato .doc para um léxico estruturado dentro de um banco de dados disponível para consultas.

4.1 O Léxico no Indexer

Segundo Barros 2003 [15], léxicos consistem em dicionários com os termos utilizados pelo sistema no processamento dos textos. Cada termo (ou palavra) no léxico pode estar associado às suas características. O léxico usado para o desenvolvimento do Indexer

é o Dicionário Enciclopédico de Astronomia e Astronáutica [29], que possui termos e significados do campo espacial. O dicionário é disponibilizado em formato .doc e, por isso, a consulta às palavras chave pelo Indexer se torna uma tarefa complicada, já que palavras-chave e significados estão em um mesmo documento não estruturado, ou seja, não é possível fazer uma busca somente pelas palavras-chave, pois a busca seria feita em todo o documento, inclusive nos respectivos significados. Além disso, o formato .doc é um formato proprietário, o que dificulta a leitura por outros programas que não sejam o Microsoft Word. Por esse motivo, optou-se por realizar a extração das palavras-chave do dicionário e separá-las dos significados de maneira estruturada, conforme explanado na subseção 4.1.1.

4.1.1 Extração das palavras-chave

Textos em ASCII e HTML são textos em formatos legíveis por seres-humanos. Existem textos, porém, que são disponibilizados em formatos binários, como PDF e Microsoft Word, que só podem ser acessados por softwares especiais. Acessar documentos multicolunas no Python, por exemplo, pode ser um grande desafio, especialmente quando se trata de documentos em formatos binários. Para a conversão desses tipos de documentos pode ser mais simples abri-los em seu próprio programa e copiá-los como texto comum em um arquivo local, descreve Steven (2009) [5]. Levando em conta essa dificuldade de leitura do PNAE, que é disponibilizado em formato binário (.doc), e por sua enorme quantidade de formatações, decidiu-se por dividir a extração das palavras-chave e dos significados em duas etapas:

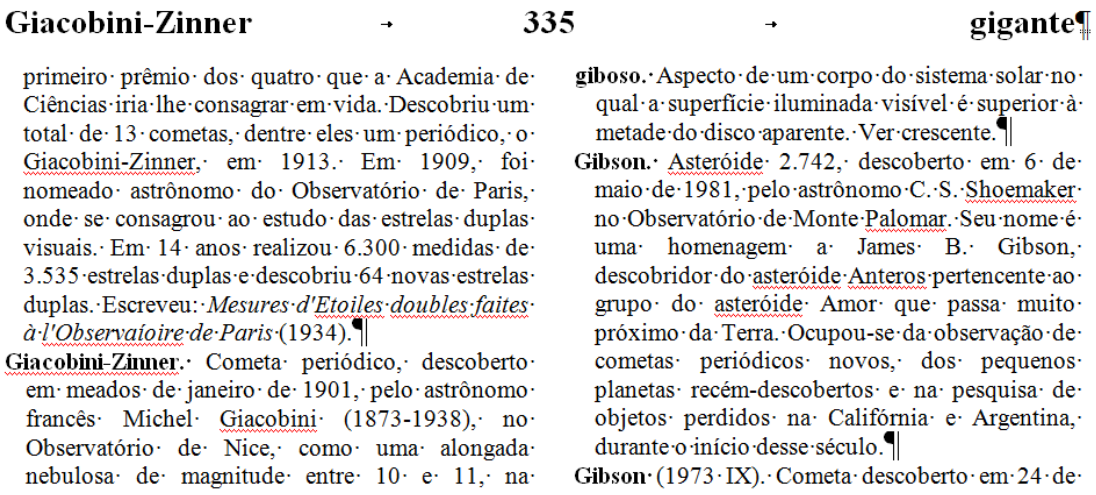
- (1) Pré processamento do texto para retirar tabulações, quebras de coluna e metadados utilizando o localizador do Microsoft Word;
- (2) Identificação e extração das palavras-chave e significados por meio de expressões regulares no ambiente do Python.

4.1.1.1 1ª Etapa

Para realizar a primeira etapa da extração, utilizamos as ferramentas de localizar e substituir do Microsoft Word, que possui a funcionalidade de pesquisar palavras pelo tamanho de sua fonte, formatação e tabulações específicas de seu próprio programa, que não podem ser lidas pelo Python. Como mostrado na Figura 4.1, é possível identificar alguns problemas que podem dificultar o processamento pelas expressões regulares no ambiente do Python.

- Texto formatado em duas colunas;
- Título no começo de cada página;
- Quebra de seção;

Primeiramente, para evitar a quebra de colunas, o documento foi transformado em um texto corrente, com coluna única. Desta maneira fica mais fácil reconhecê-lo na hora de fazer a leitura pelo ambiente do Python. Com a ajuda da ferramenta de localizar e substituir do Microsoft Word foi possível identificar e selecionar os caracteres de quebra



Giacobini-Zinner → 335 → gigante¶

primeiro· prêmio· dos· quatro· que· a· Academia· de· Ciências·iria·lhe·consagrar·em·vida.·Descobriu·um· total· de· 13· cometas,· dentre· eles· um· periódico,· o· Giacobini-Zinner,· em· 1913.· Em· 1909,· foi· nomeado· astrônomo· do· Observatório· de· Paris,· onde· se· consagrou· ao· estudo· das· estrelas· duplas· visuais.· Em· 14· anos· realizou· 6.300· medidas· de· 3.535·estrelas·duplas·e·descobriu·64·novas·estrelas· duplas.·Escreveu:·*Mesures·d'Etoiles·doubles·faites· à·l'Observaíoire·de·Paris*·(1934).¶

Giacobini-Zinner.· Cometa· periódico,· descoberto· em· meados· de· janeiro· de· 1901,· pelo· astrônomo· francês· Michel· Giacobini· (1873-1938),· no· Observatório· de· Nice,· como· uma· alongada· nebulosa· de· magnitude· entre· 10· e· 11,· na·

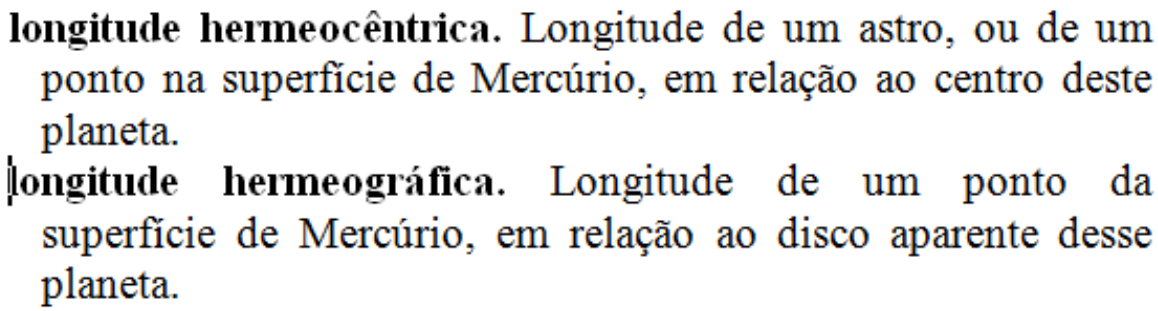
giboso.·Aspecto·de·um·corpo·do·sistema·solar·no· qual·a·superfície·iluminada·visível·é·superior·à· metade·do·disco·aparente.·Ver·crescente.¶

Gibson.· Asteróide· 2.742,· descoberto· em· 6· de· maio·de·1981,·pelo·astrônomo·C.·S.·Shoemaker· no·Observatório·de·Monte·Palomar.·Seu·nome·é· uma· homenagem· a· James· B.· Gibson,· descobridor·do·asteróide·Anteros·pertencente·ao· grupo· do· asteróide· Amor· que· passa· muito· próximo·da·Terra.·Ocupou-se·da·observação·de· cometas· periódicos· novos,· dos· pequenos· planetas· recém-descobertos· e· na· pesquisa· de· objetos· perdidos· na· Califórnia· e· Argentina,· durante·o·início·desse·século.¶

Gibson·(1973·IX).·Cometa·descoberto·em·24·de·

Figura 4.1: Dicionário antes da filtragem.

de seção, o que possibilitou sua exclusão. As ferramentas de localizar e substituir do Microsoft Word permitem a busca por palavras com filtro de tipo e tamanho da fonte. Dessa maneira, observou-se que os títulos no início de cada página possuíam o tamanho 11 e tipo negrito como características particulares. Com isso, o uso das ferramentas foi imprescindível para identificar, selecionar e excluir os títulos, tornando assim o texto o mais livre de formatações e "lixos"que pudessem atrapalhar o uso das expressões regulares. Depois da exclusão dos três problemas principais, foi possível observar melhor o padrão que as palavras-chave e os significados se encontravam dentro do texto. Observou-se que toda palavra-chave começava uma linha nova, estava em negrito e era finalizada com um ponto final e consequentemente era seguida de seu significado, que era finalizado por um caractere de quebra de linha, veja Figura 4.2.



longitude hermeocêntrica. Longitude de um astro, ou de um ponto na superfície de Mercúrio, em relação ao centro deste planeta.

longitude hermeográfica. Longitude de um ponto da superfície de Mercúrio, em relação ao disco aparente desse planeta.

Figura 4.2: Padrão de palavra-chave e significado.

O único problema é que como o texto seria importado sem nenhum tipo de fonte, não seria possível diferenciar uma palavra que estava em negrito ou não, fora do ambiente do Microsoft Word. Para solucionar o problema, foi utilizado mais uma vez as ferramentas de localizar e substituir para identificar, selecionar e adicionar ao começo de cada palavra em negrito, o símbolo identificador único "#", que tornaria possível o reconhecimento de uma palavra em negrito pelo ambiente do Python de um texto sem qualquer formatação.

Dessa forma, a versão final do documento, que seria copiada e gravada em um arquivo de texto sem formatação, ficou o mais livre de formatações possíveo, como pode ser vista na Figura 4.3.

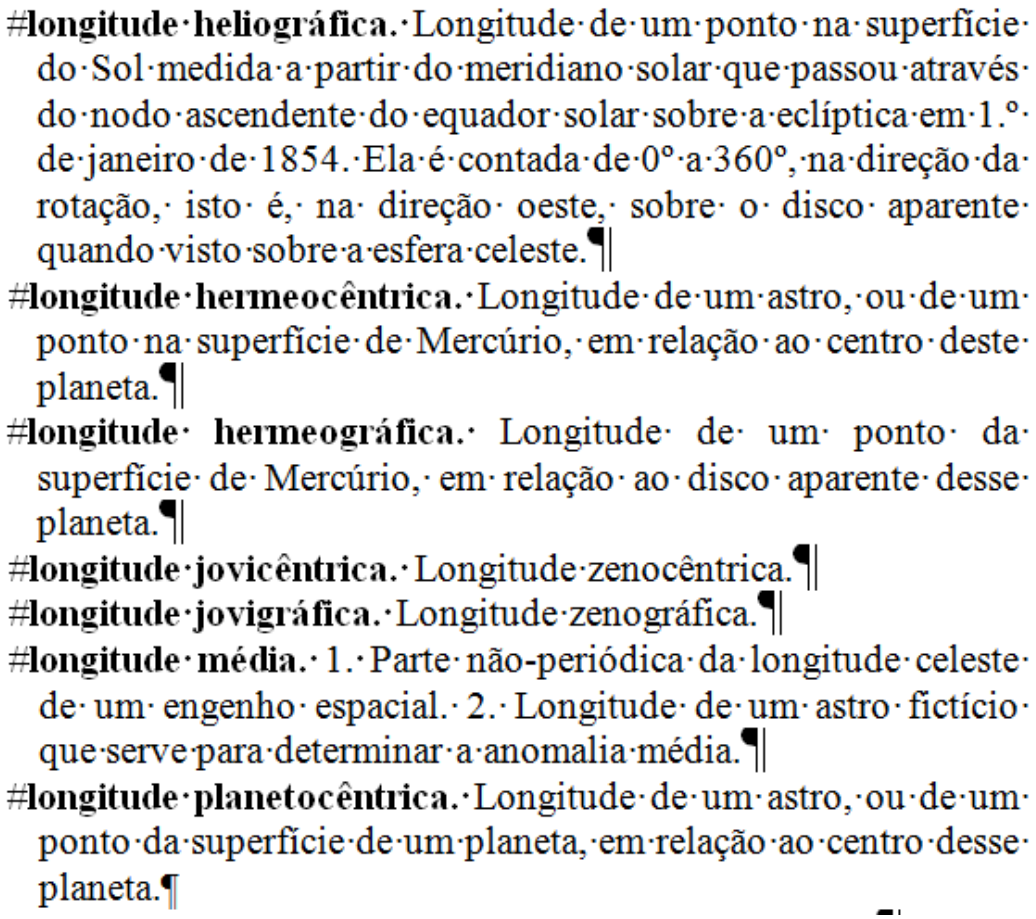

#**longitude heliográfica.** Longitude de um ponto na superfície do Sol medida a partir do meridiano solar que passou através do nodo ascendente do equador solar sobre a eclíptica em 1.º de janeiro de 1854. Ela é contada de 0º a 360º, na direção da rotação, isto é, na direção oeste, sobre o disco aparente quando visto sobre a esfera celeste.¶
#**longitude hermeocêntrica.** Longitude de um astro, ou de um ponto na superfície de Mercúrio, em relação ao centro deste planeta.¶
#**longitude hermeográfica.** Longitude de um ponto da superfície de Mercúrio, em relação ao disco aparente desse planeta.¶
#**longitude jovicêntrica.** Longitude zenocêntrica.¶
#**longitude jovigráfica.** Longitude zenográfica.¶
#**longitude média.** 1. Parte não-periódica da longitude celeste de um engenho espacial. 2. Longitude de um astro fictício que serve para determinar a anomalia média.¶
#**longitude planetocêntrica.** Longitude de um astro, ou de um ponto da superfície de um planeta, em relação ao centro desse planeta.¶

Figura 4.3: Dicionário depois de passar pela primeira etapa da extração.

4.1.1.2 2ª Etapa

Na segunda etapa da extração das palavras, já com o texto pré processado e com a exclusão e adição de alguns metadados no Microsoft Word, é possível fazer o processamento do texto para a estruturação do dicionário, que será separado em palavras-chave e significados e incluídos dentro de um banco de dados.

Como é possível ver no fluxograma do algoritmo de extração da Figura 4.4, o primeiro passo é a leitura do dicionário, que é feita por linhas, ou seja, a cada caractere de quebra de linha dentro do texto é armazenado um item dentro de uma lista chamada "dicionário". A partir dessa lista, que foi dividida em trechos separados por caracteres de quebra de linha, testou-se item por item até que fosse encontrado o padrão da expressão regular das palavras-chave e dos significados.

Conforme dito na subseção anterior, foi observado um padrão de existência das palavras-chave e dos significados dentro do texto. Para cada uma das extrações foi usado um padrão de expressão regular, conforme a Figura 4.5.

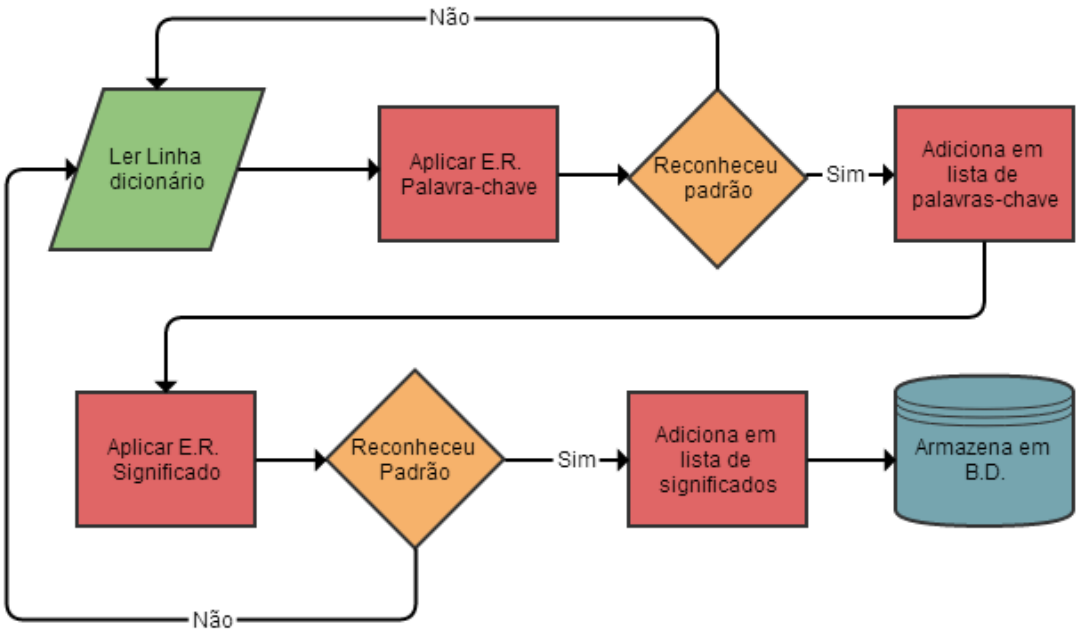


Figura 4.4: Fluxograma do algoritmo de extração.

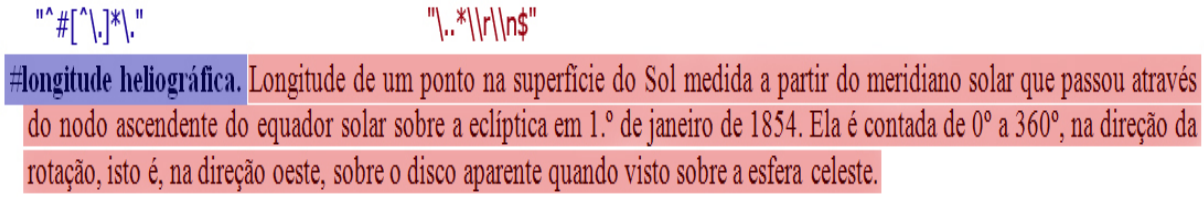


Figura 4.5: Expressão regular das palavras-chave em azul e dos significados em vermelho.

No caso das palavras-chave foi usado a expressão regular "^#[^\.]*\.", que pode ser traduzida em todo o item que necessariamente comece com o caractere identificador "#", que logo em seguida possua quantos caracteres possíveis que não sejam ponto final "."e que seja seguido por um ponto final ".". Toda substring dos itens correntes que são reconhecidos por esse padrão são armazenadas em uma nova lista chamada "palavras_chave".

O mesmo acontece para o reconhecimento do padrão dos significados, porém com outra expressão regular que é representada por "\..*\\r\\n$". Ela pode ser traduzida como todo item que possua uma substring que comece com um ponto final ".", que logo em seguida possua qualquer caractere zero ou mais vezes e que seja finalizado pelos caracteres "\r\n", que indicam quebra de linha. Toda substring que satisfaça esse padrão é adicionada a uma lista chamada "significados". Depois que toda lista do dicionário é percorrida e todas as palavras-chave e significados reconhecidos estão dentro de suas respectivas listas, o próximo passo agora é armazenar as listas dentro de uma tabela do banco de dados. Cada elemento da lista de palavras-chave está associado aos elementos da lista de significados, ou seja, o elemento de posição x da lista "palavras_chave"é o termo correspondente ao significado do elemento de posição x da lista "significados".

Para armazenar os dados colhidos, foi criada uma tabela no banco de dados chamada "TERMOS", que armazena as palavras-chave e significados do léxico. Ela será vista com

mais detalhes na subseção 4.2.3

Agora com a tabela criada e populada, o léxico do Dicionário Enciclopédico de Astronomia e Astronáutica está pronto para ser consultado. O código fonte completo do algoritmo de extração está disponível no Anexo I

Na Seção 4.2, será apresentado uma visão geral do Indexer; como seu algoritmo faz uso dos léxicos para fazer a indexação das referências; como os significados são mostrados no documento indexado e outras informações como levantamento de requisitos, banco de dados, etc.

4.2 Indexer

4.2.1 Visão Geral do Indexer

A tela inicial do programa é dividida em duas partes: um quadro de inclusão de documentos e outro de inclusão/seleção de dicionários (veja Figura 4.6). Na parte esquerda da tela encontra-se o quadro de documentos onde é possível fazer o upload dos documentos, que serão posteriormente submetidos à indexação, de seu computador para o servidor do Indexer. Desta maneira, seus arquivos ficam armazenados e disponíveis sempre que quiser.

Os documentos são colocados em uma lista a medida em que o usuário os adiciona e podem ser visualizados imediatamente após a inclusão do documento. O Indexer só aceita documentos .txt e .html que estejam usando a codificação utf-8.

O lado direito da tela inicial encontra-se o quadro da escolha dos dicionários usados como fonte para a indexação dos documentos. Neste projeto, o único dicionário disponível é o Dicionário Enciclopédico de Astronomia e Astronáutica [29], utilizado para fazer a indexação de documentos relacionados à área espacial. É possível também adicionar novos dicionários de acordo com o tipo de documento, isso vai depender do que o usuário deseja.

A inclusão de novos dicionários, porém, não está disponível para essa versão do Indexer, pois, para isso, é preciso haver um padrão de estrutura de palavras chaves e significados para que o Indexer seja compatível com qualquer dicionário, mas essa funcionalidade fica como trabalhos futuros.

Depois que o documento é adicionado e o dicionário é escolhido pelo usuário, basta apertar o botão "indexar"na lista de documentos, ao lado do respectivo documento desejado. Essa ação irá chamar o algoritmo de indexação do Indexer, que irá executá-lo com base nas opções escolhidas pelo usuário. Depois de executado o Indexer irá disponibilizar o novo documento indexado para download em seu *browser*. Veremos mais detalhadamente como esse algoritmo funciona na Seção 4.2.5.

4.2.2 Trabalhos Correlatos

Foram encontrados alguns softwares semelhantes ao Indexer, a famosa enciclopédia online Wikipedia e a ferramenta Dictionary da Apple. Na Figura 4.7 é possível observar o comparativo entre as funções que o Indexer, a Wikipedia e o Dictionary possuem e suas principais diferenças. Foram analisados alguns requisitos para o comparativo, que foram baseados nos requisitos levantado no desenvolvimento do Indexer.

Figura 4.6: Tela inicial do Indexer.

A grande diferença entre o Indexer e os demais softwares é a fase de identificação das palavras-chave em comum com o léxico.

O Dictionary da Apple é um software muito completo, porém, não possui a funcionalidade de identificar os termos presentes no léxico automaticamente. Para que o usuário veja o significado de uma palavra, ele mesmo precisa selecionar o termo desejado para que o programa verifique se ela existe no dicionário [4].

A Wikipedia também possui a funcionalidade de identificar termos dentro de um artigo, a partir de um link que pode referenciar uma página externa ou até mesmo outro artigo, porém tudo isso é feito manualmente, pelo próprio autor do artigo ou por outros leitores/autores [33].

4.2.3 Banco de dados

Os SGDBs usados para a execução do projeto foram o Mysql e o Sqlite. O Sqlite foi usado no desenvolvimento do Indexer para armazenar alguns metadados que o framework web (Django) utiliza para rodar a aplicação, mas apenas uma tabela foi criada para armazenar os caminhos dos documentos adicionados pelo usuário.

Para armazenar as palavras-chave e os significados foi usado o SGDB MySql, que possui apenas duas tabelas conforme diagrama da Figura 4.8. A tabela "TERMOS"possui três colunas: idEnquete, nome e significado. A coluna "idEnquete"armazena os ids de cada termo; a coluna nome o nome de cada termo e a coluna significado armazena os significados dos termos do banco. A tabela "TERMOS"serve basicamente para armazenar o léxico que será usado na hora do processamento do documento dado como input. Dessa maneira, o léxico suporta qualquer tipo de consulta que possa ser feita usando a linguagem Sql.

A outra tabela existente no banco de dados é a "*STOPWORDS*". Ela possui apenas duas colunas: "idStopword"e "nome". Elas significam, respectivamente, o id e o nome

Nome	Suporte ao domínio Espacial	Suporte para novos dicionários	Identificação automática dos termos	Apresentação instantânea dos significados	Distribuição
Dictionary Apple	não	sim	não	sim	proprietário
Wikipedia	não	Não	não	links inseridos manualmente	gratuito
Indexer	sim	Em desenvolvimento	sim	sim	gatuito

Figura 4.7: Comparativo de softwares correlatos.

de cada *stopword*. Esta tabela é consultada no momento da execução do algoritmo de indexação do Indexer. Antes de um termo ser verificado na tabela "TERMOS", ele passa primeiramente por uma consulta à tabela "*STOPWORDS*", caso o termo esteja contido na tabela ele é descartado, porém levando em conta algumas regras descritas com mais detalhes na seção 4.2.5.

4.2.4 Levantamento de Requisitos

O levantamento de requisitos foi feito levando em consideração algumas necessidades da Agência Espacial Brasileira. O Programa Nacional de Atividades Espaciais teve um papel de extrema importância na motivação do projeto e a partir dele foram levantados os requisitos necessários para a execução de um programa que pudesse ajudar na identificação de termos contextualizados da área espacial e prover ao leitor informações que agregasse valor descritivo aos termos encontrados.

4.2.4.1 Requisitos Funcionais

Juntamente com o chefe da divisão de informática da Agência Espacial Brasileira, foram levantados os seguintes requisitos para o desenvolvimento do Indexer:

- Armazenar léxico da área espacial.
- Identificar termos técnicos do domínio espacial dentro do Programa Nacional de Atividades Espaciais e de outros documentos relacionados à área;
- Diferenciar termos simples de termos compostos dentro do texto;
- Gerar documento em formato web .html indexado, mantendo o aspecto visual do documento original.
- Apresentar ao leitor, de forma instantânea, os significados dos termos identificados;

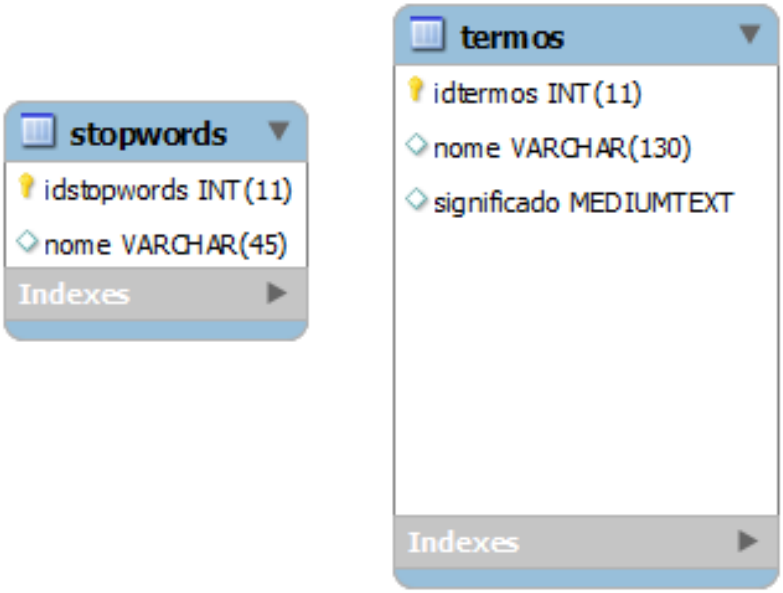


Figura 4.8: Tabelas "STOPWORDS"e "TERMOS".

4.2.4.2 Requisitos Não Funcionais

Na etapa do levantamento de requisitos, observamos alguns pontos que poderiam melhorar o desempenho do processamento, assim como a usabilidade, a facilidade de acesso e a portabilidade. Por isso, foram levantados alguns requisitos não funcionais que ajudasse na otimização desses pontos.

- Utilização de banco de dados para melhorar a performance do processamento do algoritmo de indexação;
- Tornar o Indexer uma aplicação Web, facilitando o acesso a qualquer tipo de dipositivo que tiver acesso a internet;
- Interface amigável para o usuário;
- Armazenamento de documentos do usuário.

4.2.5 Algoritmo

O algoritmo tratado nesta seção é usado no processo de indexação das referências dos termos do documento junto ao léxico selecionado. Como visto na Seção 4.2.1, para se executar o algoritmo de indexação proposto por esse projeto é preciso, primeiramente,

escolher o documento que irá passar pelo processamento e depois o tipo de léxico a ser submetido ao processo de indexação das referências.

No diagrama da Figura 4.9 é possível ver como o processamento é executado, desde o input do documento selecionado pelo usuário, passando pelo processamento de indexação das referências dos termos, até chegar ao output do documento processado.

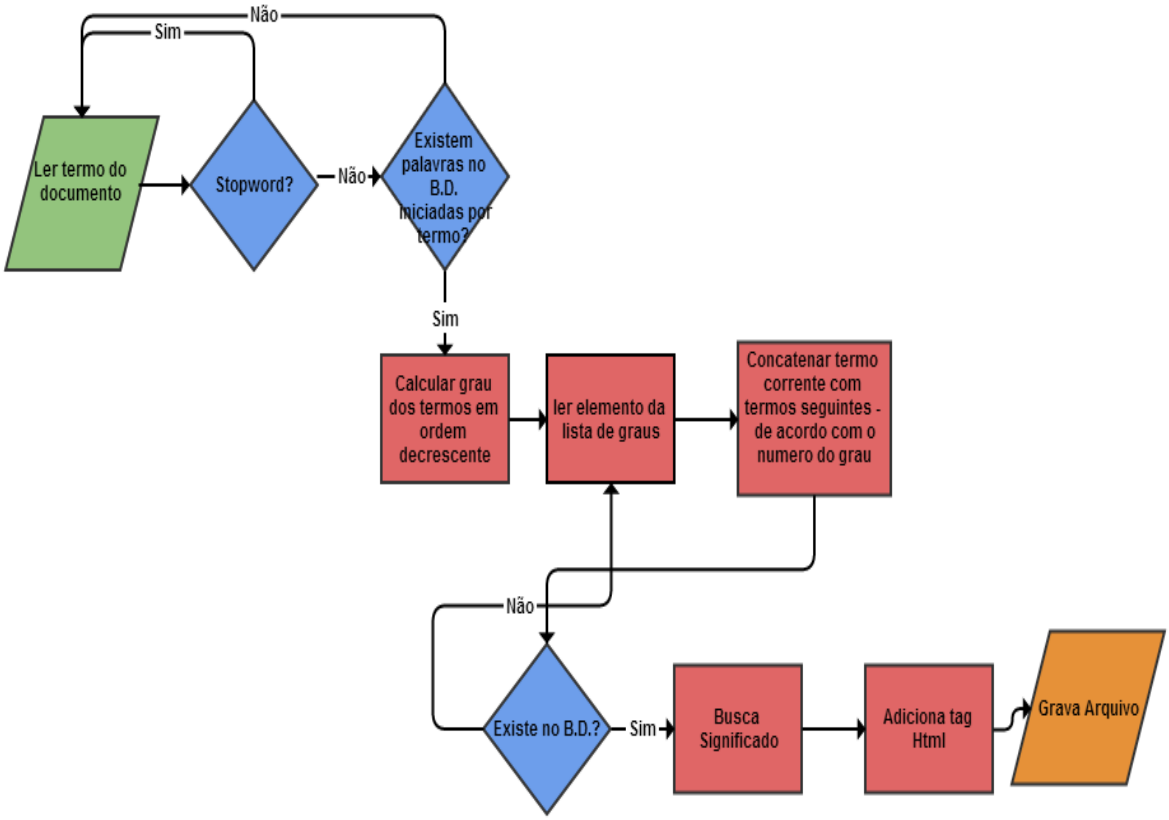


Figura 4.9: Fluxograma do algoritmo do Indexer.

Na primeira fase do algoritmo, é possível observar a leitura do documento recebido pelo Indexer. O documento é lido em codificação utf-8 e armazenado por completo dentro de uma única string. É preciso, porém, realizar o processo de tokenization, para transformar todo o texto, que está armazenado na string, em uma lista de strings. Para isso é usado uma função chamada "split()"que separa a string, a cada caractere de espaço encontrado, em uma lista de substrings. Dessa maneira, cada termo dentro do texto é representado por um elemento da lista, facilitando a manipulação dos dados.

Depois que a lista de termos do documento é criada, é preciso percorrê-la para fazer a identificação dos termos em comum com o léxico selecionado pelo usuário. Para fazer essa diferenciação são usados os conceitos de dois tipos de termos: os simples e os compostos. Essa distinção é feita para aumentar a contextualização dos termos em comum entre o léxico e o documento. Por exemplo, no trecho do documento do PNAE "...tendo em vista o hemisfério sul...", os termos simples "hemisfério"e "sul"possuem significados únicos dentro do léxico espacial, porém, é possível encontrar o termo composto "hemisfério sul",

que dentro do contexto seria o mais adequado e contextualizado com a semântica do documento do que os termos simples analisados separadamente.

Outro fator importante é a presença de termos que não tem valor significativo dentro do contexto do documento, como preposições, artigos, pronomes, etc. Esses termos são chamados de stopwords e, por isso, também devem ser desconsiderados na hora da processamento dos termos em comum. É preciso, porém, notar que apesar de não terem valor semântico dentro do texto, são de extrema importância para a identificação dos termos compostos. O termo composto "sistema de controle", por exemplo, possui três termos simples, sendo que um deles a preposição "de". Neste caso, apesar do valor semântico da preposição não ser relevante para o significado do termo composto, é de extrema importância para identificação do termo no léxico, já que a palavra está registrada como "sistema de controle"e não "sistema controle". Portanto, as stopwords não podem ser desconsideradas quando se tratam de termos compostos como vimos anteriormente, porém, elas são totalmente dispensáveis em caso de termos simples, pois uma preposição não tem valor algum se não estiver acompanhada de outra palavra significativa.

Usando estrutura de repetição, a lista de termos do documento é percorrida uma a uma. Para cada palavra do documento verifica-se, primeiramente, se ela existe na lista de stopwords armazenada dentro de um banco de dados. Se a palavra for uma stopword ela é desconsiderada automaticamente e o algoritmo passa para o próximo termo do documento. O processo se repete até que um termo não esteja contido dentro da lista de stopwords, neste caso, o termo é pesquisado dentro da lista de palavras-chave que também está no banco de dados. Executa-se a query "SELECT nome, significado FROM termos WHERE nome LIKE 'termo %'", que pode ser traduzida como a seleção de todas as palavras-chave, sejam elas simples ou compostas, que comecem com o termo corrente. Depois de recuperar todos os termos simples e compostos pela query, ou seja, iniciados pelo termo corrente do documento, é necessário saber o grau das palavras selecionadas, isto é, quantos termos compõem cada palavra. Essa informação é necessária para identificar quantos termos devemos considerar na procura dentro da coluna de palavras-chave. O algoritmo então calcula os graus de todas as palavras encontradas e os armazena em uma lista ordenada decrescentemente. A partir daí, é feita uma busca no banco com o termo corrente concatenado com x termos seguintes no documento (entenda-se x como um elemento da lista de graus.).

A lista é ordenada decrescentemente para que o algoritmo consiga identificar o termo composto de maior grau dentro do documento, ou seja, o algoritmo primeiro procura pelo maior termo possível contido no banco para depois procurar os menores. Dessa maneira evita-se que o termo presente no documento "sistema de controle", fosse reconhecido apenas como "sistema"ou, por exemplo, como duas palavras distintas "sistema"e "controle", trazendo dois significados corretos, porém descontextualizados.

O intuito do algoritmo do Indexer é que ele possa reconhecer termos compostos tais como, "hemisfério sul", "sistema de controle", "espectro eletromagnético"que, neste caso, estão contidos dentro do documento do PNAE e dentro do Dicionário Enciclopédico de Astronomia e Astronáutica. Com isso, os significados buscados no léxico se tornam mais contextualizados e condizentes com o que o autor deseja passar ao leitor e não, simplesmente, significados soltos e sem contexto como os de "hemisfério", "sistema"e "espectro".

Depois que o termo é identificado, seja ele simples ou composto, o algoritmo injeta uma marcação HTML no documento para adicionar o significado e explicitá-lo mudando

a cor de sua fonte. A apresentação do documento depois de totalmente indexado será detalhada na próxima seção 4.2.6

O código fonte do algoritmo de indexação está disponivel no Anexo II.

4.2.6 Apresentação do Documento Indexado

Depois que o documento original adicionado pelo usuário passa por todo o processamento descrito na seção 4.2.5, gera-se então um novo documento já processado com todos os termos em comum com o dicionário.

4.2.6.1 Termos Identificados

Cada termo identificado dentro do documento, recebe no momento de sua identificação uma injeção de uma tag html. Esta tag, descrita pela marcação "<font>"possui um atributo chamado "color", que juntos definem a cor da fonte do texto envolvido pela tag.

A cor da fonte do termo identificado, vai variar de acordo com sua classificação, ou seja, se é um termo simples ou composto. Dessa forma, fica mais claro identificar o que o Indexer está considerando como significado dentro do texto.

As cores definidas para representar os dois tipos de termos foram escolhidas aleatoriamente, mas de maneira que ficassem agradáveis ao leitor ao mesmo tempo em que deixassem claro quais termos foram resultantes do algoritmo de indexação. Para identificar os termos simples, foi usada a cor azul e para os termos compostos foi usada a cor vermelha, conforme a Figura 4.10.

4.2.6.2 Significados

Depois que um termo é identificado e devidamente marcado pelo atributo "color", ele recebe outra injeção de marcação html. Desta vez ele é envolto por uma tag "<a>", que representa um link, mas que ao invés de referenciar outra página web, recebe o significado como atributo. Para fazer esta referência, utilizamos o atributo "title", que é atribuído de uma string contendo o significado já recuperado do banco de dados.

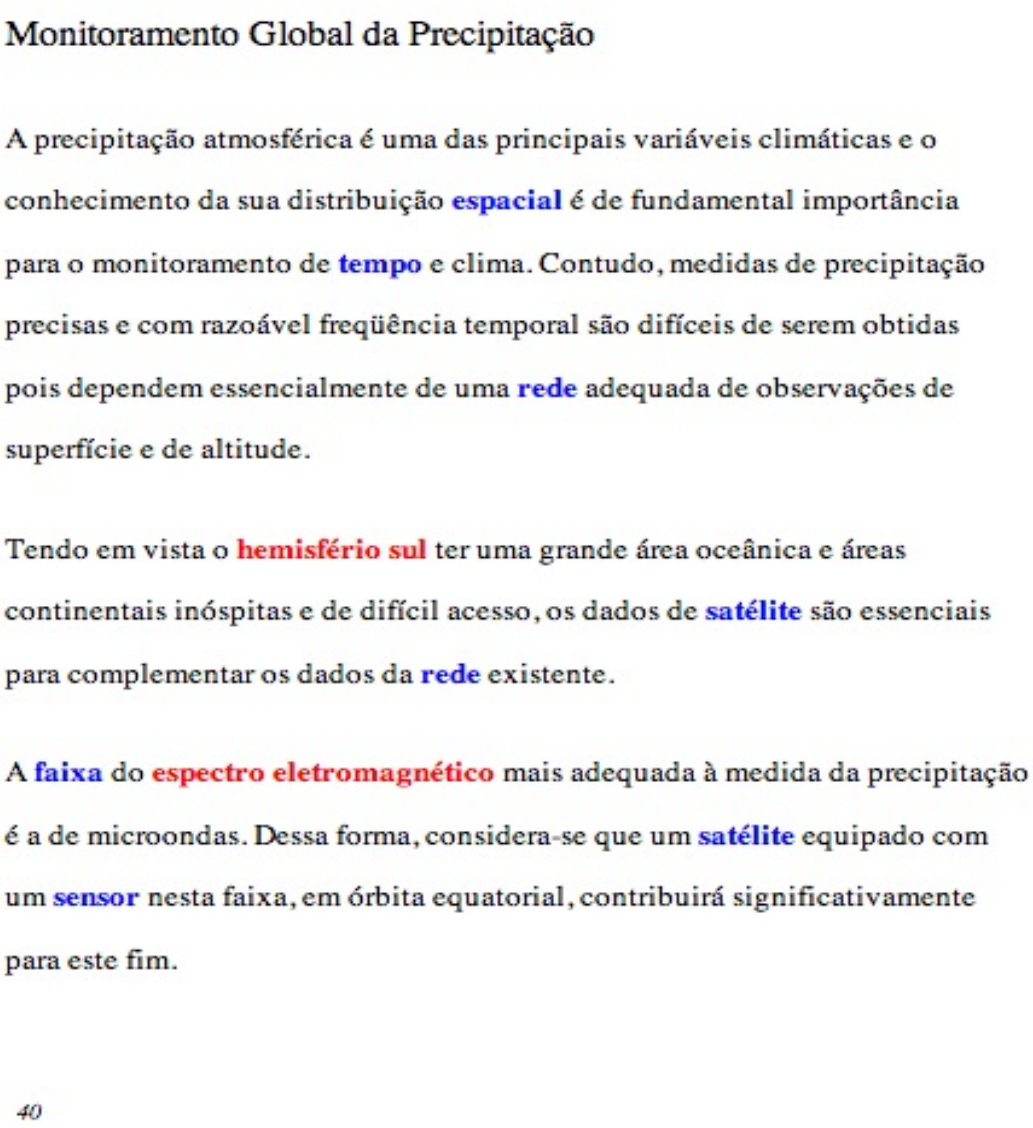
Caso o usuário queira ver o significado dos termos encontrados, basta sobrepor o ponteiro do mouse sobre a palavra desejada, seja ela um termo simples ou composto. Ao sobrepor o ponteiro do mouse, uma legenda contendo o respectivo significado deverá abrir enquanto o ponteiro estiver sobreposto à palavra. Se o usuário mover o ponteiro para fora da área da palavra, o significado deverá desaparecer. A Figura 4.11 mostra um exemplo da palavra marcada com sua respectiva cor e significado aberto, enquanto o usuário posiciona o ponteiro do mouse sobre a palavra desejada.

Essa maneira de apresentação torna a visualização do significado agradável ao usuário, facilitando a usabilidade do aplicativo e tornando-o mais intuitivo.

4.2.6.3 Características do programa

- O dicionário utilizado pelo Indexer possui 5.6GB e 18.039 termos específicos da área espacial;
- O documento do PNAE possui 3.7MB;

- O programa demora 15 minutos para concluir a indexação do documento do PNAE ao dicionário;
- Os arquivos aceitos são em formatos .HTML e .TXT



Monitoramento Global da Precipitação

A precipitação atmosférica é uma das principais variáveis climáticas e o conhecimento da sua distribuição **espacial** é de fundamental importância para o monitoramento de **tempo** e clima. Contudo, medidas de precipitação precisas e com razoável freqüência temporal são difíceis de serem obtidas pois dependem essencialmente de uma **rede** adequada de observações de superfície e de altitude.

Tendo em vista o **hemisfério sul** ter uma grande área oceânica e áreas continentais inóspitas e de difícil acesso, os dados de **satélite** são essenciais para complementar os dados da **rede** existente.

A **faixa** do **espectro eletromagnético** mais adequada à medida da precipitação é a de microondas. Dessa forma, considera-se que um **satélite** equipado com um **sensor** nesta faixa, em órbita equatorial, contribuirá significativamente para este fim.

40

Figura 4.10: Termos identificados.

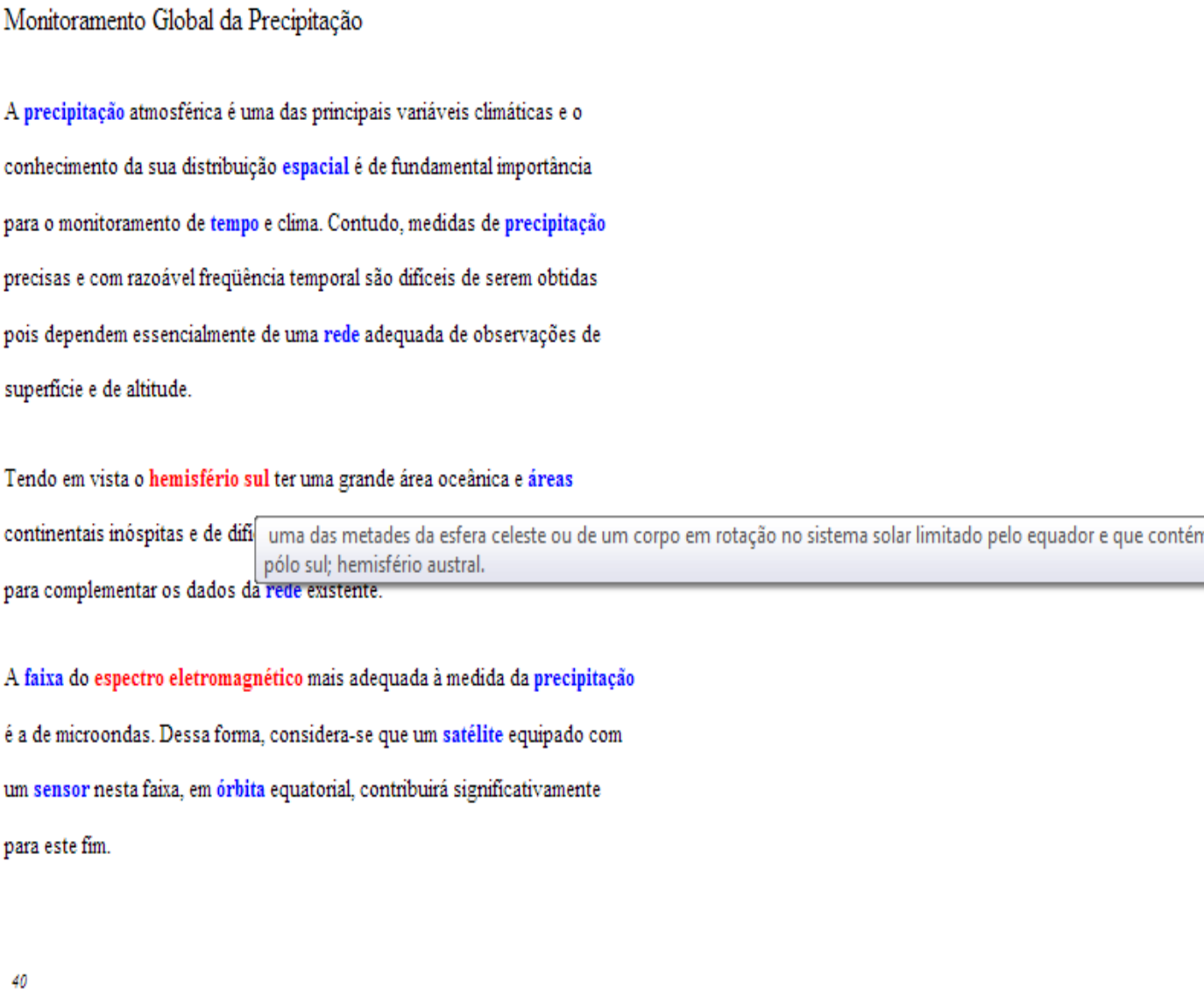


Figura 4.11: Significado do termo composto "hemisfério sul".

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho foi proposto e desenvolvido o sistema Indexer,voltado tanto para gestores da área espacial quanto para leitores leigos da área que estejam interessados na leitura do documento do Programa Nacional de Atividades Espaciais do Brasil e outros texto relacionados ao contexto da área espacial.

O programa propõe ajudar o leitor na identificação de termos simples e compostos da área espacial e prover significados ou informações adicionais que possam ajudar na leitura do documento.

Os objetivos específicos deste trabalho foram a elaboração de um Léxico da área espacial, o desenvolvimento de um algoritmo de indexação das referências de palavras-chave do léxico ao documento desejado e a explicitação dos significados dos termos encontrados de maneira instântanea ao usuário.

A estruturação do Dicionário Enciclopédico de Astronomia e Astronáutica em palavras-chave e significados dentro de um banco de dados, nos proporcionou a criação de um léxico que pudesse abranger aproximadamente 18 mil termos técnicos da astronomia, astrofísica, física, etc. Isto agrega um grande poder de descrição de palavras-chave em diversos textos relacionados a essas áreas afins.

O Indexer roda em um servidor web, o que permite ao usuário acessar a aplicação de qualquer computador ou dispositivo que estiver conectado a internet, facilitando a utilização em diferentes plataformas, bastando apenas um browser como requisito básico. Além disso, ele disponibiliza uma espaço para armazenar até 10 documentos de 5.6MB em seu servidor, evitando que o usuário precise disponibilizar o documento toda vez que for submetê-lo ao algoritmo de indexação.

A parte notável do trabalho é a diferenciação que Indexer faz entre os termos simples e compostos, melhorando ainda mais o tipo de informação mostrada ao leitor. O significado dos termos se torna mais contextualizado, evitando que termos compostos sejam fragmentados em significados soltos e sem contexto.

Pode-se concluir deste trabalho que os objetivos almejados foram alcançados, ou seja, a criação de um léxico que abrangesse a área espacial, o desenvolvimento de um algoritmo de identificação e indexação de referências dos termos em questão e a apresentação dos significados dos termos foram concluídos com eficácia. O usuário leitor do documento do Programa Nacional de Atividades Espaciais detém de uma ferramenta que o auxilia na identificação e busca de informações adicionais que possam enriquecer o valor descritivo

de termos técnicos dentro do texto, provendo ao leitor mais conhecimento teórico sobre o assunto tratado.

Como trabalhos futuros, propôem-se melhoras no algoritmo de identificação das palavras-chave, utilizando o conceito de stemming, que seria a identificação dos termos com base em seus radicais, extendendo a indexação aos termos que possuem o sentido básico igual, mas que estão escritos de formas diferentes, aumentando ainda mais o alcance do número de termos identificados.

Outra proposta a ser feita é a padronização de um formato para dicionários, facilitando o suporte do Indexer a qualquer dicionário que esteja dentro deste padrão. Desta maneira, fica mais fácil a adição de novos léxicos de diferentes domínios, podendo ajudar a enriquecer o valor descritivo de termos de variados contextos, como a área do direito, medicina, arquivologia, etc.

A maneira como os significados são disponibilizados ao usuário pode ficar mais rica, podendo haver diferentes formas de apresentação dessas informações de acordo com usuário. Significados de diferentes léxicos podem ser apresentados conjuntamente para que seja possível a comparação de informações entre tipos diferentes de contexto, ou até mesmo a possibilidade de se mostrar esses significados no próprio documento, sem que seja necessário a criação de um novo documento, como é feito hoje.

Referências

[1] Wiks Y. A., Slator B. M., and Guthrie L. M. *Eletric Words: Dictionaries, Computers and meanings*. The MIT Press, Cambridge, 2009. 14, 15

[2] Ministério da Ciência e Tecnologia Agência Espacial Brasileira. *Programa Nacional de Atividades Espaciais*. 2005. 1, 2, 27

[3] Ministério da Ciência e Tecnologia Agência Espacial Brasileira. *Programa Nacional de Atividades Espaciais*. 2012. 1

[4] Apple. http://support.apple.com/kb/ht2496, acessado em 24-janeiro-2014. 33

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, Gravenstein Highway North, 2009. xi, 11, 12, 14, 28

[6] Neto B.R. and Yates R.B. *Recuperação de Informação*. 2013. 13

[7] Wertheimer A. M. C. O dicionário remissivo comparado aos outros dicionário existentes. *Encontro do CELSUL*, 1:393–429, 1995. 14

[8] O.N.P. Cardoso. Recuperação de informação. *INFOCOMP Journal of Computer Science*, 2(1):33–38, 2000. x, 8

[9] Flanagan D. *JavaScript O Guia Definitivo*. Artmed Editora S.A, 2004. 23

[10] Flanagan D. *jQuery Pocket Reference*. O'Reilly, 2011. 24

[11] Manning D.C., Raghavan P., and Schutze H. *An Introduction to Information Retrieval*. Cambridge Univ. Press, Cambridge, 2009. 11

[12] Django. https://www.djangoproject.com, acessado em 05-janeiro-2014. 21, 22

[13] Ferneda E. *Recuperação de Informação: Análisa da contribuição da Ciência da Computação para a Ciência da Informação*. USP, 2003. 9

[14] Leiva I.S e Fujita M.S.L. *Política de Indexação*. Cultura Acadêmica, 2012. x, 7, 8, 10

[15] Barros F.A. and Robin J. Processamento de linguagem natural. *Universidade Federal de Pernambuco*, 2003. 14, 27

[16] Borges G.S.B. *Indexação automática de documentos textuais: Proposta de critérios essenciais*. Escola de Ciência da Informação - UFMG, 2009. 7

[17] Savoia H.R. *XHTML e CSS + PHP e MySQL Primeiros Passos.* IELD Editora, 2013. 25, 26

[18] Chaumier J. Indexação: conceito, etapas e instrumentos. *Revista Brasileira de Biblioteconomia e Documentação, São Paulo*, 21(1/2):63–79, 1988. 9

[19] Elkner J., Downe A. B., and Meyers C. *Learning with Python.* O'Reilly, 2012. x, 17, 18

[20] Labaki J. *Introdução ao Python.* Universidade Estadual Paulista, 2009. x, 17, 18, 19

[21] Rutter J. *Smasching jQuery.* Wiley, 2011. 23, 24

[22] Guthrie L., Pustejovsky J., Wilks Y., and Slator B.M. The role of lexicons in natural language processing. *Communications of the ACM*, 39:63–72, 1996. 14

[23] Zacharias G.K. Lalli F.M., Bueno F.F. *Evolução da Programação Web.* Faculdade Comunitária de Campinas, 2008. 23

[24] Deitel H. M. *Xml Como Programar.* Pearson Education. Inc, 2001. 25

[25] Gonzalez M. and Vera L.S. Recuperação de informação e processamento de linguagem natural. *PUCRS - Faculdade de Informática*, 2009. 14, 15

[26] Lutz M. *Learning Python 4th Edition.* O'Reilly, 2009. 21

[27] Rocco M. *Instant Django 1.5 Application Development Starter.* Packt Publishing. 21, 22

[28] Sipser M. *Introduction to the Theory of Computation.* PWS Publishing Company, Boston, 1997. 12

[29] Mourão R.R.F. *Dicionário Enciclopédico de Astronomia e Astronáutica.* Editora Nova Fronteira, 1987. x, 1, 5, 6, 27, 28, 32

[30] Neto O.S; Galesi T. *Python e Django. Desenvolvimento ágil de aplicações web.* Novatec, 2010. 17

[31] TecMundo. `http://www.tecmundo.com.br/programacao/2710-o-que-e-java-.htm`, acessado em 31-dezembro-2014. 23

[32] Gomes T.P.D. Aspectos críticos na análise conceitual de charges. *Múltiplos Olhares em Ciência da Informação, Belo Horizonte*, 2(2), 2012. 9

[33] Wikipedia. `http://pt.wikipedia.org/wiki/Ajuda:Guia_de_ediç~ao/formataç~ao`, acessado em 24-janeiro-2014. 33

Anexo I

Código do Programa de Extração de Termos e Significados

```
'''
@author: igor
'''
#abrir dicionario
import codecs

import MySQLdb

fp_dicionario = codecs.open(r"dicionario_filtrado.txt",encoding
    ="utf8")

dicionario = fp_dicionario.readlines()


fp_dicionario.close()

import re

palavras_chave = []
for x in dicionario:
    if(re.findall("^#[^\.]*\.", x)):
        palavras_chave.append(re.findall("^#[^\.]*\.", x)[0])
    else:
        palavras_chave.append("")

# print(palavras_chave)

significados = []
for y in dicionario:
    if(re.findall("\..*\r\n$", y)):
```

```
        significados.append(re.findall("\..*\r\n$", y)[0])
    else:
        significados.append("")

# print(significados)




#!/usr/bin/python

db = MySQLdb.connect(host="127.0.0.1",
                     port=3306,
                     user="usuario",
                      passwd="senha",
                      db="processer",
                      )


cur = db.cursor()



db.set_character_set('utf8')
cur.execute('SET NAMES utf8;')
cur.execute('SET character_set_connection=utf8;')


for x in range(len(palavras_chave)):
    if(len(palavras_chave[x]) > 0 and len(palavras_chave[x]) <
       130):
        current_palavra_chave = palavras_chave[x][1:-1].lower().
           encode("utf-8")
        current_significado = significados[x][1:-2].lower().
           encode("utf-8")
        cur.execute("INSERT INTO termos (nome, significado)
           VALUES (%s,%s)",(current_palavra_chave,
           current_significado))


db.commit()
cur.close()
db.close()

print(cur._executed)
```

```
print("persistido com sucesso!!")
exit()
```

Anexo II

Código do Programa de Indexação do Indexer

```
horaInicio = datetime.datetime.now()
    document = Document.objects.get(id=document_id)
    document_path = document.docfile.path
    fp_pnae_html = codecs.open(document_path,encoding="utf8")
    pnae = fp_pnae_html.read()
    pnae = pnae.split()
    fp_pnae_html.close()




    #Recebe uma lista de palavras e retorna um cojunto dos graus
        de todos as palavras contidas na lista
    def words_order_set(lista_palavras):
        lista_graus = [len(lista_palavras[x][0].split(" ")) for
            x in range(len(lista_palavras))]
        lista_graus_sorted = sorted(lista_graus, reverse=True)
        conjunto = set(lista_graus_sorted)
        lista_graus = [conjunto.pop() for x in range(len(
            conjunto))]
        return lista_graus

    #Conexao com o banco de dados
    db = MySQLdb.connect(host="127.0.0.1",
```

```
                        port=3306,
                        user="usuario",
                         passwd="senha",
                         db="processer",
                         )


cur = db.cursor()


db.set_character_set('utf8')
cur.execute('SET NAMES utf8;')
cur.execute('SET character_set_connection=utf8;')


sql = "Select nome from termos where nome LIKE '%s\%'"



w = 0
while w < (len(pnae)):
    encontrouComposto = False
    #Somente para termo simples. Se o termo simples for
       encontrado na lista de stopwords entao ele eh
       desconsiderado
    #e o algoritmo passa para a proxima palavra.
    isStopword = False
    cur.execute(r'Select nome from stopwords where nome LIKE
        %s', pnae[w].lower().encode("utf-8") )
    isStopword = cur.fetchall()
    if(isStopword):
        #Incrementa 1 no contador dos termos do pnae,
           passando para a proxima palavra.
        #Nao executa o algoritmo para esse termo
        w = w+1
        continue

    #Caso contrario procura termo simples e seus derivados
       no dicionario de termos.
    termo_simples = pnae[w].encode("utf-8").lower()+"%"
    cur.execute(r'Select nome from termos where nome LIKE %s
       ',termo_simples)
    data = cur.fetchall()
```

```
#Se encontra algum termo simples, ou seja, data conter
   alguma informacao, entao entre no if.
if(data):
    #Laco que percorre uma lista de graus de cada
       expressao que contem o termo simples encontrado.
    #A lista estah em ordem decresente
    for x in reversed(words_order_set(data)):
        #Se o indice corrente eh menor que o tamanho da
           lista entao checar palavra,
        #caso contrario, dar um continue e pular para o
           proximo grau
        if((w+x) < len(pnae)):
            termo_composto = " ".join(pnae[w:(w+x)]).
               encode("utf-8").lower()
        else:continue

        cur.execute(r'Select nome, significado from
           termos where nome LIKE %s',termo_composto)
        composto_encontrado = cur.fetchall()

        #Se um termo composto eh encontrado
        if(composto_encontrado):

            significado = composto_encontrado[0][1].
               decode("utf-8")
            if(x == 1):
                palavra_indexada = '<a title="'+
                   significado+'"><b><Font color="blue
                   ">'+pnae[w]+'</font></b></a>'
                pnae[w] = palavra_indexada
            elif(x > 1):
                composto_indexado_inicio = '<a title="'+
                   significado+'"><b><Font color="red
                   ">'+pnae[w]
                pnae[w] =  composto_indexado_inicio
                composto_indexado_final = pnae[w+x
                   -1]+'</font></b></a>'
                pnae[w+x-1] = composto_indexado_final

            #Jah que o termo composto foi encontrado,
               desconsidera-se as palavras do termo e
               pula-se as devidas posicoes.
            w += x
            encontrouComposto = True
            break
```

```
        #Se um termo composto foi encontrado pula para o proximo
            termo, de acordo com o grau do termo encontrado
        #Caso contrario passa para o proximo termo
        if(encontrouComposto):
            continue
        else:
            print
            w = w + 1


horaFim = datetime.datetime.now()
cur.close()
db.close()

# exit()

#saida de dados
root_path = "C:/IGOR_TRABALHO/AEB/PESQUISA_PLN"
pnae = " ".encode("utf-8").join(pnae)
pnae_utf8 = pnae.encode("utf-8")
fp_saida = open(root_path + r"/TESTES/Teste_Expressoes16.
    html","w")
fp_saida.write(pnae_utf8.__str__())
fp_saida.close()
print("arquivo criado com sucesso!")
```