

INSTITUTO FEDERAL DO ESPÍRITO SANTO  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**ÁDLER OLIVEIRA SILVA NEVES**

**SISTEMA WEB DE ARQUITETURA MODULAR PARA PROCESSAMENTO DE  
CORPORA**

Serra  
2018

**ÁDLER OLIVEIRA SILVA NEVES**

**SISTEMA WEB DE ARQUITETURA MODULAR PARA PROCESSAMENTO DE  
CORPORA**

Trabalho de Conclusão de Curso apresentado à Coordenadoria do Curso de Sistemas de Informação do Instituto Federal do Espírito Santo, Campus Serra, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Ernani Leite Ribeiro Filho

Serra  
2018

X000x

Neves, Ádler Oliveira Silva.

Sistema web de arquitetura modular para processamento de corpora /  
Ádler Oliveira Silva Neves. — Serra, 2018.

72 p.: il. (algumas color.); 30 cm.

Orientador: Prof. Me. Ernani Leite Ribeiro Filho.

Trabalho de Conclusão de Curso — Instituto Federal do Espírito  
Santo, 2018

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade  
xxx. III. Faculdade de xxx. IV. Título.

CDU 00:000:000.0

**ÁDLER OLIVEIRA SILVA NEVES**

**SISTEMA WEB DE ARQUITETURA MODULAR PARA PROCESSAMENTO DE  
CORPORA**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Sistemas de Informação do Instituto Federal do Espírito Santo, Campus Serra, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em 27 de abril de 2018.

**COMISSÃO EXAMINADORA**

---

Prof. Me. Ernani Leite Ribeiro Filho  
Instituto Federal do Espírito Santo  
Orientador

---

Prof<sup>a</sup>. Me. Alessandra Aguiar Vilarinho  
Instituto Federal do Espírito Santo

---

Membro Convidado da Banca 1  
Instituição Externa 1

---

Membro Convidado da Banca 2  
Instituição Externa 2

## **DECLARAÇÃO DO AUTOR**

Declaro, para fins de pesquisa acadêmica, didática e técnico-científica, que este Trabalho de Conclusão de Curso pode ser parcialmente utilizado, desde que se faça referência à fonte e ao autor.

Serra, 27 de abril de 2018.

Ádler Oliveira Silva Neves

## RESUMO

Profissionais de diversos ramos de atuação dependem do uso de termos específicos de sua área, alguns desses em língua estrangeira, para uma comunicação eficaz e eficiente, ~~este cujo processo de aprendizado~~ requer contato com casos de uso para formação de um repertório na língua ~~ou seu dialeto~~. No entanto, a formação de tal repertório é geralmente feita lentamente e manualmente, ou através de ferramentas não-gratuitas quando o método de ensino faz o uso de **corpora**. Neste trabalho, desenvolvemos uma aplicação de internet de código aberto para ampliar ~~as~~ funcionalidades **tendo em vista aplicações similares existentes** enquanto se mantém amigável ao usuário. O sistema tem como entradas um corpus fornecido pelo usuário. O sistema, utilizando ~~de~~ algoritmos de segmentação, etiquetagem e busca, processa o corpus, tendo como saídas a lista de sentenças, a frequência de palavras, a lista de palavras, o autômato de cada sentença, o autômato do corpus como um todo e o texto etiquetado. Opcionalmente, uma expressão de busca pode ser fornecida para a entrada e à saída é adicionada a colocação no texto dos termos encontrados. ~~Obtivemos~~ que nosso sistema é modular e extensível através de *plug-ins*, diferentemente das principais soluções disponíveis no mercado; a contrapartida é que é um sistema novo, logo, nem todas as funcionalidades presentes em outras aplicações estarão disponíveis nesta. ~~Portanto~~ o sistema desenvolvido pode ser utilizado para auxiliar o estudo da língua por linguistas ~~enquanto~~ ferramenta em futuras pesquisas, e pode ser possível usar os *plug-ins* resultantes desses trabalhos para elevar como um todo o estado da arte do Processamento Automático da Linguagem Natural.

Palavras-chave: Processamento Automático da Linguagem Natural. Sistema Web. Corpus. *Plug-in*.

## ABSTRACT

Professionals from different fields of activity depend on the use of specific terms in their area, some of them in a foreign language, for efficient and effective communication, whose learning process requires contact with use cases to form a repertoire in the language or their dialect. However, the formation of such repertoire is usually done slowly and manually, or through non-free tools when the teaching methodology uses *corpora*. In this work, we developed an open source internet application to extend the functionality having in sight similar existing applications while remaining user friendly. The system has as inputs a user-supplied corpus. The system, using algorithms of segmentation, tagging and search, processes the corpus, having as outputs the list of sentences, word frequencies, word list, the automaton of each sentence, the automaton of the corpus as a whole and the tagged text. Optionally, a search expression can be provided for input, so to the output is added the collocation. We have found that our system is modular and extensible through plug-ins, unlike the major solutions available on the market; the counterpart is that it is a new system, so not all the features present in other applications can be found in this one. Therefore, the developed system can be used to support the study of the language by linguists as a research tool in the future, and it may be possible to use the plug-ins developed as product of those works to raise the state of the art of Automatic Processing of Natural Language.

Keywords: Automatic Processing of Natural Language. Web system. Corpus. Plug-in.

## LISTA DE FIGURAS

Figura 1 – Pilha de protocolos da internet e modelo OSI . . . . .	25
Figura 2 – Exemplo de canal de comunicação entre aplicações no cliente e no servidor	26
Figura 3 – Excerto do modelo lógico gerado automaticamente a partir da implementação	32
Figura 4 – Árvore de navegação das páginas do sistema . . . . .	33
Figura 5 – Árvore de diretórios e <i>scripts</i> . . . . .	35
Figura 6 – Prevalência de etiquetas do <i>Unitex/GramLab</i> . . . . .	42
Figura 7 – Prevalência de etiquetas do <i>YAS-Tagger</i> . . . . .	43
Figura 8 – Prevalência de etiquetas do <i>YAS-Tagger</i> sobre corpus de Aires . . . . .	44
Figura 9 – Página inicial do sistema desenvolvido . . . . .	45
Figura 10 – Detalhe do seletor de idiomas . . . . .	45
Figura 11 – Detalhe dos botões de autenticação . . . . .	45
Figura 12 – Lista de corpora dum usuário . . . . .	46
Figura 13 – Lista de documentos dum corpus . . . . .	46
Figura 14 – Opções de análises para um corpus . . . . .	47
Figura 15 – Lista de sentenças dum corpus . . . . .	48
Figura 16 – Lista de palavras dum corpus . . . . .	48
Figura 17 – Erro do <i>MXPOST</i> . . . . .	49
Figura 18 – Erro do <i>Brill Tagger</i> . . . . .	49
Figura 19 – Corpus processado pelo <i>Tree Tagger</i> . . . . .	49
Figura 20 – Tela de busca do concordanciador . . . . .	50
Figura 21 – Tela de resultados do concordanciador . . . . .	51
Figura 22 – Visualização de documento a partir dum clique na “seta para trás” . . . . .	51
Figura 23 – Listagem de documentos a partir dum clique na “seta para trás” . . . . .	51
Figura 24 – Modelo lógico gerado automaticamente a partir da implementação . . . . .	60
Figura 25 – Sequência de inicialização . . . . .	61
Figura 26 – Requisição sem uso de <i>plug-ins</i> para exibir formulário . . . . .	62
Figura 27 – Requisição para exibir lista de <i>plug-ins</i> na página “Análise” . . . . .	63
Figura 28 – Requisição para exibir página gerada por <i>plug-in</i> que usa a aplicação . . . . .	64
Figura 29 – Requisição para exibir página gerada por <i>plug-in</i> que usa outros <i>plug-ins</i> . . . . .	65



## LISTA DE TABELAS

Tabela 1 – Tabela comparativa resumida de softwares de Processamento de Linguagem Natural . . . . .	14
Tabela 2 – O material usado no projeto Floresta Sintáctica . . . . .	20
Tabela 3 – Comparação da precisão entre três etiquetadores . . . . .	21
Tabela 4 – Tabela comparativa de softwares de Processamento de Linguagem Natural . . . . .	29
Tabela 5 – Conversão da etiquetas utilizadas do Floresta Sintáctica para notação intermediária . . . . .	38
Tabela 6 – Conversão da etiquetas utilizadas do Unitex/GramLab para notação intermediária . . . . .	38
Tabela 7 – Comparação da precisão entre cinco etiquetadores . . . . .	43
Tabela 8 – Tabela de contingência por etiqueta do <i>Unitex/Gramlab</i> . . . . .	67
Tabela 9 – Estatísticas por etiqueta do <i>Unitex/Gramlab</i> . . . . .	67
Tabela 10 – Tabela de contingência por etiqueta do <i>YAS-Tagger</i> . . . . .	68
Tabela 11 – Estatísticas por etiqueta do <i>YAS-Tagger</i> . . . . .	69
Tabela 12 – Tabela de contingência por etiqueta do <i>YAS-Tagger</i> sobre corpus de Aires . . . . .	70
Tabela 13 – Estatísticas por etiqueta do <i>YAS-Tagger</i> sobre corpus de Aires . . . . .	71

## LISTA DE QUADROS

Quadro 1 – Separação entre treino e teste . . . . .	37
Quadro 2 – Etiquetas gramaticais frequentes . . . . .	37

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Motivação	12
1.2	Justificativa	14
1.3	Objetivos	14
<b>1.3.1</b>	<b>Geral</b>	<b>14</b>
<b>1.3.2</b>	<b>Específicos</b>	<b>15</b>
1.4	Organização do texto	15
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>16</b>
2.1	Revisão histórica	16
2.2	Estado da arte	17
<b>2.2.1</b>	<b>Gerenciadores de corpora</b>	<b>17</b>
<b>2.2.2</b>	<b>Etiquetadores morfossintáticos</b>	<b>18</b>
2.3	Trabalhos Correlatos	18
<b>2.3.1</b>	<b>Indexação automática por meio da extração e seleção de sintagmas nominais em textos em língua portuguesa</b>	<b>19</b>
<b>2.3.2</b>	<b>Sistema identificador de sintagmas verbais do PB</b>	<b>19</b>
<b>2.3.3</b>	<b>O sistema de análise “PALAVRAS”: Análise Gramatical Automática do Português em uma framework de gramática de restrição</b>	<b>19</b>
<b>2.3.4</b>	<b>Floresta Sintáctica</b>	<b>20</b>
<b>2.3.5</b>	<b>Implementação, adaptação, combinação e avaliação de etiquetadores para o português do Brasil</b>	<b>20</b>
2.4	Conceitos relevantes	20
<b>2.4.1</b>	<b>Processamento automático da linguagem natural</b>	<b>21</b>
<b>2.4.2</b>	<b>Linguística de corpus</b>	<b>21</b>
<b>2.4.3</b>	<b>Separadores de sentenças</b>	<b>22</b>
<b>2.4.4</b>	<b>Medidas estatísticas de avaliação de desempenho de classificadores binários</b>	<b>23</b>
<b>2.4.5</b>	<b>Aplicações de internet</b>	<b>24</b>
2.4.5.1	Arquitetura cliente-servidor	25
2.4.5.2	Framework de aplicação de internet Django	26
2.4.5.3	Deveres legais a respeito da mínima retenção de dados	27
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>28</b>
3.1	Levantamento de requisitos	28
<b>3.1.1</b>	<b>Tecnologias escolhidas</b>	<b>31</b>
3.2	O gerenciador de corpora	32
3.3	Esboço da navegabilidade	32
3.4	O subsistema de eventos	34

3.5	Separação de sentenças e obtenção de lista de palavras . . . . .	36
3.6	Treino do etiquetador . . . . .	36
3.7	A proposta do concordanciador . . . . .	39
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>41</b>
4.1	Treino do etiquetador . . . . .	41
<b>4.1.1</b>	<b>Teste 1: Etiquetador do <i>Unitex/GramLab</i> . . . . .</b>	<b>41</b>
<b>4.1.2</b>	<b>Teste 2: Etiquetador <i>YAS-Tagger</i> . . . . .</b>	<b>41</b>
4.1.2.1	Teste 2.1: Etiquetador <i>YAS-Tagger</i> sobre corpus de Aires . . . . .	42
4.2	O sistema desenvolvido . . . . .	44
<b>4.2.1</b>	<b>Separador de sentenças e gerador de lista de palavras . . . . .</b>	<b>46</b>
<b>4.2.2</b>	<b>Etiquetador . . . . .</b>	<b>47</b>
<b>4.2.3</b>	<b>Concordanciador . . . . .</b>	<b>48</b>
4.3	Código fonte . . . . .	50
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>52</b>
5.1	Trabalhos futuros . . . . .	53
	<b>REFERÊNCIAS . . . . .</b>	<b>55</b>
	<b>APÊNDICE A – O MODELO LÓGICO COMPLETO DO GERENCIA-</b>	
	<b>DOR DE CORPORA . . . . .</b>	<b>60</b>
	<b>APÊNDICE B – FLUXOS DE EXECUÇÃO DO SUBSISTEMA DE <i>PLUG-</i></b>	
	<b><i>INS</i> . . . . .</b>	<b>61</b>
B.1	Sequência de inicialização . . . . .	61
B.2	Requisição sem uso de <i>plug-ins</i> para exibir formulário . . . . .	62
B.3	Requisição para exibir lista de <i>plug-ins</i> na página “Análise” . . . . .	63
B.4	Requisição para exibir página gerada por <i>plug-in</i> que usa a aplicação . . . . .	64
B.5	Requisição para exibir página gerada por <i>plug-in</i> que usa outros <i>plug-ins</i> . . . . .	65
	<b>APÊNDICE C – HARDWARE UTILIZADO DURANTE O TREINO E</b>	
	<b>TESTE DOS ETIQUETADORES . . . . .</b>	<b>66</b>
	<b>APÊNDICE D – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTI-</b>	
	<b>CAS DO ETIQUETADOR <i>UNITEX/GRAMLAB</i> . . . . .</b>	<b>67</b>
	<b>APÊNDICE E – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTI-</b>	
	<b>CAS DO ETIQUETADOR <i>YAS-TAGGER</i> . . . . .</b>	<b>68</b>
	<b>APÊNDICE F – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTI-</b>	
	<b>CAS DO ETIQUETADOR <i>YAS-TAGGER</i> SOBRE COR-</b>	
	<b>PUS DE AIRES . . . . .</b>	<b>70</b>

# 1 INTRODUÇÃO

Neste capítulo, será apresentada uma breve contextualização do tema e a motivação deste trabalho (seção 1.1), seguido da justificativa, a qual explicita a lacuna que este trabalho busca solucionar (seção 1.2), e depois pelos objetivos gerais e específicos deste trabalho (seção 1.3), que buscam preencher a lacuna explicitada, e, por fim, uma breve descrição da organização dos capítulos seguintes (seção 1.4).

## 1.1 MOTIVAÇÃO

O domínio ~~do idioma que fala~~ é competência necessária para qualquer indivíduo que queira exercer sua cidadania plena, ~~onde~~ termos específicos variam de acordo com a atividade profissional desempenhada. Por exemplo, pessoas que exerçam atividades no ramo jurídico utilizam termos específicos do direito no seu dia a dia profissional, tais ~~quais~~ *habeas corpus*, abroquelar, cártula chéquica, areópago, fulcro e estipêndo funcional; no ramo médico, os termos poderiam ser incisão, febrífugo, benigno, ejunorrafia ou abdução. Assim como “CSRF”, para um contador, é sigla para Contribuições Sociais Retidas na Fonte; para um desenvolvedor, é sigla para *Cross-Site Request Forgery*, um tipo de fraude que algumas aplicações web são vulneráveis. De modo semelhante, o ensino de uma língua estrangeira depende do aprendizado de novas palavras com novos significados. A aquisição de um repertório ~~de idioma ou do dialeto~~ a ser utilizado é necessário para o ~~correto~~ uso deste no dia-a-dia, ~~entendendo e fazendo~~ ser entendido por seus interlocutores.

Para auxiliar a aprendizagem de uma língua estrangeira (ANTHONY, 2013; OLIVEIRA, 2009)(AARTS; GRANGER, 1998 apud SARDINHA, 2000) ou ~~de idioma~~ para fins específicos (OLIVEIRA, 2009) como no cotidiano profissional, a análise de corpus de textos é uma técnica que pode ser utilizada. Segundo Sardinha (2000), um ~~desse~~ trabalhos do século XX ~~foi o~~ de Thorndike, publicado em 1921, ~~que~~ identificou as palavras mais frequentes de língua inglesa e “[...] impulsionou mudanças no ensino de língua materna e estrangeira, tanto nos Estados Unidos quanto na Europa [...]” (p. 236), onde “durante boa parte do século XX [...] muitos pesquisadores [...] se dedicaram à descrição da linguagem por meio de corpora<sup>1</sup>, [...] [que] eram coletados, mantidos e analisados manualmente [...]” (p. 235). Apesar dos benefícios, no século XX, a velocidade e os custos de propagação da informação ainda eram elevados.

Com o desenvolvimento tecnológico das telecomunicações, os custos de acesso à informação reduziram, o tempo até ~~obtenção~~ da informação reduziu a frações de segundo e possibilitou

<sup>1</sup> Copora é o plural de corpus, que é um conjunto de textos relevante para estudar um determinado grupo de pessoas ou um ramo do conhecimento (TAGNIN, 2010, p. 358).

que o tráfego da informação também aumentasse em volume. Uma dessas tecnologias de telecomunicações é a internet, que possibilita que repositórios de conhecimento, plataformas de entretenimento e aplicações de interação social coexistam e apresentem funcionalidades em tempo-real, caso programadas para tal. De acordo com uma pesquisa do CETIC.BR (2016), 51% do total de domicílios possuem acesso à internet, 86% dos alunos acessaram a internet ao menos uma vez no último trimestre à pesquisa e 95% dos professores possuíam acesso à internet em casa. Isso vem gerando uma crescente pervasividade da internet na vida das pessoas.

Não foi apenas no contexto das telecomunicações que a tecnologia se desenvolveu. A velocidade de processamento também se cresceu de forma que trabalhos manuais tais como a contagem de palavras ou busca em textos, por exemplo, puderam ser automatizados de forma que seu tempo-seja reduzido vertiginosamente. A exemplo, um computador portátil Toshiba 2250XCDS comprado em 2000 vinha equipado com um disco rígido de 6GB, 64MB de RAM e um processador de 600MHz de um único núcleo (Toshiba, 2000), e um computador portátil CCE n325 comprado em 2013 vinha equipado com um disco rígido de 500GB (83 vezes o anterior), 2GB de RAM (32 vezes o anterior) e um processador 1.8GHz de dois núcleos (6 vezes o anterior) (CCE, 2013). Logo, o poder de processamento das máquinas vendidas ao consumidor final no varejo aumentou muito.

Já quanto a processamento remoto, o *Google Cloud Platform* oferece gratuitamente o que chama de *Cloud Launcher* (Google Cloud, 2018a). Tal serviço gratuito consiste de uma máquina virtual com IP estático, 30GB de armazenamento secundário, 0.6GB de RAM, 1 CPU virtual que recebe 0.2 do tempo da CPU da máquina real e 1GB de tráfego mensal de saída para fora da América do Norte, Austrália ou China (Google Cloud, 2018b). Um pré-requisito para a criação da conta é adicionar um cartão de crédito que não seja pré-pago. Portanto, os custos de realizar processamento remoto num servidor pode ser nulo dependendo do problema a ser resolvido e das tecnologias utilizadas.

Resgatando as aplicações anteriormente mencionadas segundo Anthony (2013), Oliveira (2009), Aarts e Granger (1998 apud SARDINHA, 2000) e Sardinha (2000), hoje tanto a montagem de corpora e seu processamento é mais fácil e rápido. No início do século passado (XX), haviam os custos de adquirir periódicos, requisitar cópias de artigos a bibliotecas, organizar todo o material para enfim dedicar tempo sobre o material de estudo. Hoje em dia, existem à disposição sites de notícias, ferramentas de indexação de páginas da internet, de artigos científicos, dentre outras, que podem ser utilizadas como fonte para a construção do material de estudo. Além disso, hoje as pessoas se comunicam abertamente por meios eletrônicos de forma que é possível estudar o uso da língua pelas pessoas na informalidade do cotidiano, sendo possível até analisar o sentimento da massa de pessoas a respeito dum determinado assunto. Por isso, fica evidente que hoje é bem mais fácil obter material para estudo que antigamente.

Com essa abundância de recursos a baixo custo e de informações acessíveis e baratas, é de se supor que exista uma igual abundância de ferramentas gratuitas para tal área do conhecimento, tanto locais (executadas no próprio dispositivo) quanto de internet (executadas num servidor com interface visível através dum navegador), mas isso não acontece. Com a grande diversidade de sistemas operacionais *desktop* (como, por exemplo, *Windows*, *MacOS* e *Linux*) e de dispositivos móveis (como, por exemplo, *Android*, *iOS* e *Windows Phone*), desenvolver uma aplicação de internet significa possuir uma única base de código que atenderá simultaneamente a todas as plataformas que possuem um navegador de internet, além de atingir um grande número de pessoas em território nacional. Com isso, será abordado na próxima subseção (1.2) a contribuição almejada por este trabalho.

## 1.2 JUSTIFICATIVA

Atualmente, existem ferramentas capazes de processar corpora, dentre elas pode-se destacar o Word Smith, o Unitex/GramLab e o Sketch Engine, que são voltadas ao pesquisador linguista enquanto usuário final. Destes, apenas o segundo é distribuído gratuitamente com funcionalidade plena e apenas o terceiro possui uma interface *web*; sendo nenhuma delas gratuita com funcionalidade plena e *web* simultaneamente, como pode ser observado na tabela 1. Dentre as ferramentas que processam a língua portuguesa, não há nenhuma plataforma que seja ao mesmo tempo gratuita em sua versão completa e cuja sua interface seja *web*, o que justifica a criação de outra com as características desejáveis. A última linha (“Corpus Slayer”, destacada em fundo amarelo) traz a ferramenta proposta para suprir tal lacuna.

Software	Interface	Gratuito	Corpus fornecido pelo usuário	Tokenizador	Etiquetador	Concordanciador
WordSmith 5	Desktop	Não	Sim	Sim	Não	Sim
Unitex/GramLab	Desktop	Sim	Sim	Sim	Francês	Sim
CorpusEye	Web	Sim	Não	Sim	Sim	Sim
COCA Online Corpus	Web	Sim	Não	Sim	Sim	Sim
NoSketch Engine	Web	Sim	Sim	Sim	Não	Sim
Sketch Engine	Web	Não	Sim	Sim	Sim	Sim
Corpus Slayer	Web	Sim	Sim	Sim	Sim	Sim

Tabela 1 – Tabela comparativa resumida de softwares de Processamento de Linguagem Natural

Fonte: O autor

## 1.3 OBJETIVOS

### 1.3.1 GERAL

Desenvolver uma aplicação *web* de código aberto para marcação e busca de partes do discurso em corpora, visando ampliar as funcionalidades em relação a *softwares* similares existentes e com interface amigável ao usuário.

### 1.3.2 ESPECÍFICOS

- Analisar comparativamente os recursos das ferramentas WordSmith, CorpusEye, COCA Online Corpus, Unitex/GramLab e Sketch Engine;
- Desenvolver ou adaptar um módulo separador de sentenças;
- Desenvolver ou adaptar um módulo que obtenha a lista de palavras de um texto;
- Desenvolver ou adaptar um módulo etiquetador de partes do discurso que atue sobre sentenças;
- Desenvolver ou adaptar um módulo concordanciador<sup>2</sup> que suporte busca por etiquetas;
- Integrar os módulos desenvolvidos ou adaptados numa aplicação *web*;
- Disponibilizar uma ferramenta livre para uso educacional.

### 1.4 ORGANIZAÇÃO DO TEXTO

No capítulo 2 será apresentado um histórico, os principais conceitos e alguns trabalhos relevantes da área de processamento da linguagem natural. No capítulo 3 será apresentado os materiais e os métodos utilizados para desenvolver este trabalho, bem como as decisões relevantes ao desenvolvimento do projeto. No capítulo 4 será apresentado e discutido os resultados obtidos. No capítulo 5 será apresentada a conclusão, as contribuições deste trabalho e temas para trabalhos futuros.

---

<sup>2</sup> concordanciador é o “programa que extrai todas as ocorrências de uma palavra de busca num corpus juntamente com seu cotexto [...]” (TAGNIN, 2010, p. 358).



## 2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os conceitos relativos ao processamento de linguagem natural, numa revisão histórica da área e seu progresso, e o atual estado da arte com seus requisitos para o progresso.

### 2.1 REVISÃO HISTÓRICA

A linguística de corpus, segundo Sardinha (2000), já existe desde a Antiguidade, citando o Corpus Helenístico de Alexandre, O Grande e os corpora de citações da Bíblia produzidos durante a Antiguidade e a Idade Média, sendo seus processos de obtenção e análise tarefas que não dependem dum computador para serem executadas. Um corpus é uma “coletânea de textos [...] compilados segundo critérios específicos para o estudo a que se propõem” (TAGNIN, 2010, p. 358), e estes buscam representar a linguagem (CAMBRIA; WHITE, 2014), ou uma variante desta (SARDINHA, 2000), podendo ser usada como forma de obtenção de informações nas mensagens nela codificadas. Sardinha (2000) evidencia que “durante boa parte do século XX [...] os corpora [...] eram coletados, mantidos e analisados manualmente [...]” (p. 235), o que mudou com a invenção do computador, que se popularizou na década de 1960.

Ainda enquanto trabalho manual, é possível citar o trabalho de contagem de 4,5 milhões de palavras realizada manualmente por Thorndike (SARDINHA, 2000), cujo produto foi uma lista de frequência de palavras. Uma das possíveis aplicações da linguística é extrair palavras-chave, esta que Tagnin (2010) define como sendo “resultados da comparação entre o corpus de estudo e um corpus de referência” (p. 359), sendo um processo baseado em contagem, o qual “elimina palavras com frequência relativa similar nos dois corpora de modo que restem as palavras cuja frequência é estatisticamente significativa” (p. 359), podendo as frequências ter sido calculadas computacionalmente ou manualmente. Cambria e White (2014) ressaltam que processos de obtenção automática de palavras-chave buscam apenas na “superfície” do texto, exemplificando que um texto sobre cães pode apenas mencionar as diversas raças, não sendo recuperado caso indexado por suas palavras-chave.

Em fevereiro de 1957 foi publicada a obra *Syntactic Structures* de Noam Chomsky, que, de acordo com Sardinha (2000), mudou o paradigma de abordagem na linguística, saindo o empirismo e entrando o racionalismo; ou seja, a análise de volumosos corpora de textos, que é sujeito à inúmeras falhas, deu lugar a “[...] teorias racionalistas da linguagem [...], notadamente a linguística gerativa. [...]” (SARDINHA, 2000, pp. 236-237). Esta abordagem se popularizou principalmente no início da década de 1960, quando o computador se popularizou.

Com a popularização do computador, representar nele a linguagem foi necessária, e uma

abordagem de solução ~~dessta~~ necessidade é modela-la como uma gramática gerativa de Chomsky (CHAVES; MELLO, 2014; CAMBRIA; WHITE, 2014). Um dos desafios encontrados foi resolver ambiguidades, que foi solucionado usando autômatos gerados a partir ~~deum~~ tratamento estatístico de ocorrências para a criação ~~deum~~ modelo probabilístico para a etiquetagem de palavras (partes do discurso) (VIEIRA; LIMA, 2001; THIELE, 2015); ~~todavia~~, tal abordagem desconsidera que uma sequencia de palavras pode significar mais que a soma dos significados de cada palavra individual. Entidades Nomeadas são referências de mais de uma palavra que referenciam uma única entidade (NADEAU; SEKINE, 2007 apud BRUCKSCHEN, 2010). Segundo Aranha e Passos (2006), o reconhecimento destas no texto se dá através de perguntas como “Quem, Quando, Como, Onde” (p. 6). BRUCKSCHEN (2010) menciona datas, números e unidades monetárias como elementos do texto comumente inclusas como entidades nomeadas, sendo que Aranha e Passos (2006) também inclui siglas. A união da identificação prévia de entidades nomeadas à representação das sentenças como autômatos é uma abordagem para tentar solucionar tal problema.

## 2.2 ESTADO DA ARTE

Nesta seção ~~serão abordados~~ o estado da arte do processamento automático de corpora, indicando como os problemas do passado são resolvidos atualmente. Primeiramente será ~~abordado~~ os que os gerenciadores de corpus modernos armazenam sobre seus documentos na seção 2.2.1, em seguida será abordado uma classe de etiquetadores, os etiquetadores morfossintáticos, que podem ser relevantes se aplicados sobre os corpora para uma compreensão inicial da estrutura do idioma em estudo, na seção 2.2.2.

### 2.2.1 GERENCIADORES DE CORPORA

De acordo com Biber (1995 apud OLIVEIRA, 2009, p. 49), “um corpus linguístico de base computacional corresponde a coleções de textos que ocorrem naturalmente na língua, organizadas sistematicamente para representar áreas de uso da língua, e das quais podemos extrair novas informações”. Tal organização pode seguir critérios diversos, mas é esperado que estes estejam anotados no texto. Aluísio e Almeida (2006) menciona a ferramenta *Lácio-Ref*, a qual visa fornecer metadados para os textos, para estes serem futuramente processados. Tais metadados são: título, subtítulo, língua, fonte, editor, local de publicação, comentários, nome do(s) autor(es), sexo do(s) autor(es), gênero e tipo textual (ALUÍSIO; ALMEIDA, 2006). Outra ferramenta é o *AntCorGen*, de Anthony (2017), a qual visa manter organizados corpus de texto através de categorias, subcategorias e seus metadados, tais ~~quais~~: título, autor, seção “resumo”, seção “introdução”, seção “materiais e métodos”, seção “resultados e discussão”, seção “conclusões”, seção “referências”, afiliações do autor e legendas de figuras e tabelas. Os metadados mencionados ajudam a descrever o contexto ao qual texto está inserido.

## 2.2.2 ETIQUETADORES MORFOSSINTÁTICOS

O atual estado da arte para etiquetadores morfossintáticos da língua portuguesa, por Silva (2014), é o PALAVRAS, de Bick. É uma gramática de restrições que, carregada no software CG-3, realiza o etiquetamento automático de textos. Possui uma taxa de acerto de 94% em relação aos seus próprios resultados, e 86,5% se comparado a uma extração manual (SILVA, 2014, p. 169). O etiquetador CG-3 é mantido como software livre e distribuído pelo projeto de pesquisa e desenvolvimento *Visual Interactive Syntax Learning* (VISL) do Instituto de Línguas e Comunicação (ISK) da Universidade do Sul da Dinamarca (SDU), já a gramática é paga.

Dentre os etiquetadores morfossintáticos gratuitos, o estado da arte, de acordo com Silva (2014), é o LX-Parser. Possui uma taxa de acerto de 75% em relação aos seus próprios resultados, e 97,5% se comparado a uma extração manual (SILVA, 2014, p. 170). Apesar de gratuito, sua licença é de redação própria (UNIVERSIDADE DE LISBOA, DEPT. DE INFORMÁTICA, GRUPO DE FALA E LINGUAGEM NATURAL, 2010) e em sua cláusula 6 proíbe a distribuição e comercialização de produtos ou serviços derivados, já em sua cláusula 7, veta a disponibilização da ferramenta por outros meios e a cláusula 9 veta o acesso de outros sites à ferramenta; portanto, a licença da ferramenta permite apenas a descarga e execução local da mesma.

Outro projeto é o Unitex/GramLab, onde sua versão em português brasileiro foi primeiramente apresentada por Muniz, Nunes e Laporte em 2005a. Consiste de uma interface gráfica em Java que faz chamadas a bibliotecas e utilitários de linha de comando compilados de códigos em C++. A licença do software é a *Lesser General Public License* versão 2.1 (FREE SOFTWARE FOUNDATION, 1999), já seus recursos linguísticos estão sob a *Lesser General Public License For Linguistic Resources* (UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE, 2009), ambas licenças livres. O pacote redistribuível padrão já fornece ferramentas e dados complementares necessários para etiquetar um texto de entrada arbitrário. Outros recursos da mesma plataforma são busca por autômato, busca por expressão regular, representação do texto como autômato, inferência de entidades nomeadas e aplicação de transdutores (PAUMIER, 2016).

## 2.3 TRABALHOS CORRELATOS

Nesta seção serão apresentados alguns trabalhos considerados similares a este ou cujo produto pode ser usado neste (fazendo deste trabalho uma continuação do anterior), estando o de Silva (2014) na subseção 2.3.1, o de Chaves e Mello (2014) na subseção 2.3.2, o de Bick (2000) na subseção 2.3.3, o da Linguateca (2010b) na subseção 2.3.4 e o da Aires (2000) na subseção 2.3.5.

### 2.3.1 INDEXAÇÃO AUTOMÁTICA POR MEIO DA EXTRAÇÃO E SELEÇÃO DE SINTAGMAS NOMINAIS EM TEXTOS EM LÍNGUA PORTUGUESA

Diferentemente da língua inglesa, segundo Silva (2014) (p. 155) “pesquisas [...] em língua portuguesa ainda não atingiram um grau de amadurecimento elevado [...] pois elas não alcançam uma taxa de precisão elevada para todos os tipos de corpus.” Dada a escassez de ferramentas eficazes para processamento da língua portuguesa, o referido autor comparou três etiquetadores morfossintáticos (OGMA, PALAVRAS e LX-PARSER), e constatou que cada um possui suas peculiaridades de modo a existir situações onde alguns se destacam e outras onde apresentam dificuldades. Tais ferramentas comparadas eram as principais ferramentas existentes até 2014.

### 2.3.2 SISTEMA IDENTIFICADOR DE SINTAGMAS VERBAIS DO PB

A língua portuguesa possui sintagmas nominais, verbais, adjetivais e preposicionados (GAGNON, 2000). Os verbais podem ser analisados conforme proposto por Chaves e Mello (2014), onde propõem a utilização do “[...] formalismo da gramática livre de contexto [...], comumente utilizado na linguística computacional para a implementação de protótipos capazes de modelar a estrutura sintagmática de uma língua natural”, sendo definido pelos próprios autores como “um protótipo com resultados interessantes” que carece na quantidade de estruturas cobertas pela gramática definida e na necessidade de melhorar a taxa de acerto na etiquetagem dos textos - sem citar valores numéricos. Tal trabalho fornece uma ferramenta para obtenção automática da análise sintagmática, mas que necessita de aprimoramentos e quantificações numéricas quanto sua eficácia.

### 2.3.3 O SISTEMA DE ANÁLISE “PALAVRAS”: ANÁLISE GRAMATICAL AUTOMÁTICA DO PORTUGUÊS EM UMA FRAMEWORK DE GRAMÁTICA DE RESTRIÇÃO

Com uma taxa de acerto na ordem de 99% para morfologia e entre 96% e 97% para sintaxe para processamento de texto livre, uma das adaptações necessárias por Bick no modelo da língua inglesa para operar para a língua portuguesa foi (BICK, 2000, p. 438) a “introdução de marcadores de dependência a nível de cláusula [...] e a introdução de funções de marcação de subcláusula para subcláusulas finitas e não-finitas”, onde algumas descobertas para a língua inglesa apresentaram se manter para a língua portuguesa, tais como: milhares de regras são necessárias para cada nível adicional de análise, que as sentenças que excluem significado são mais frequentes que as que buscam e que a determinação do contexto concentra-se mais à esquerda da sentença. Tal tese explicita as principais diferenças entre o processamento sintagmático dos idiomas português e inglês, bem como suas diferenças e adaptações necessárias para se adequar ao idioma e ainda assim manter uma alta taxa de acerto.

### 2.3.4 FLORESTA SINTÁCTICA

O projeto Floresta Sintáctica se define como “[...] um “treebank” para a língua portuguesa, ou seja, um conjunto de itens sintacticamente analisados e publicamente disponível [...]”, sendo “[...] um projecto de colaboração entre a Linguateca e o projecto VISL, com textos portugueses e brasileiros, anotados automaticamente pelo PALAVRAS (BICK, 2000) e revistos por linguistas” (LINGUATECA, 2010b). A tabela 2 traz o material disponibilizado pelo projeto Floresta Sintáctica. Tal projeto é relevante para este trabalho enquanto provedor de recursos linguísticos anotados, mas sua licença de uso não é explicitada exceto pela base de dados denominada “Amazônia”, que é distribuída sob a licença “CC BY-NC-SA 3.0 BR” (LINGUATECA, 2016).

corpus	Floresta Virgem	Amazônia	Bosque	Selva Lit.	Selva Fal.	Selva Cie.
palavras	c. 1.640.000	c. 4.580.000	c. 186.000	c. 105.000	c. 170.000	c. 125.000
frases	c. 96.000	c. 275.000	9368	c. 7.900	c. 14.000	c. 6.200
revisão	não	não	integral	parcial	parcial	parcial
variantes	PT BR	BR	PT BR	PT BR	PT BR	PT BR
gênero	jornalístico	opinião	jornalístico	literário	entrevistas e debates	acadêmico e informativo
domínio	genérico	cultura brasileira	genérico	genérico	biografia e política	educação, psicolinguística, computação, economia e ciências
registro	formal	formal e informal	formal	formal	formal e informal	formal
modo	escrito	escrito	escrito	escrito	falado	escrito
origem	jornais Folha de São Paulo e Público	blog Overmundo	jornais Folha de São Paulo e Público	livros	Museu da Pessoa (PT, BR) e debates parlamentares	bibliotecas universitárias, banco centrais e Wikipedia

Tabela 2 – O material usado no projeto Floresta Sintáctica

Fonte: Linguateca (2009)

### 2.3.5 IMPLEMENTAÇÃO, ADAPTAÇÃO, COMBINAÇÃO E AVALIAÇÃO DE ETIQUETADORES PARA O PORTUGUÊS DO BRASIL

O Núcleo Interinstitucional de Linguística Computacional (2012) da Universidade de São Paulo mantém público desde novembro de 2000 um corpus anotado com 51 etiquetas (sendo 15 delas de pontuação) em 103674 palavras divididas ao longo de 4726 sentenças, além dos arquivos de treino para uso nos etiquetadores *MXPOST*, *TreeTagger* e *Brill Tagger*. Tais recursos (exceto os três últimos etiquetadores) são produtos da dissertação de Aires (2000), a qual contempla uma comparação de precisão entre os diferentes etiquetadores analisados (vide tabela 3).

## 2.4 CONCEITOS RELEVANTES

Nesta seção serão apresentados conceitos relevantes ao trabalho, sendo eles o processamento automático da linguagem natural, a linguística de corpus, o que são os separadores de sentenças,

Etiquetador	Precisão
MXPOST	89,66%
Brill Tagger	88,76%
Tree Tagger	88,47%

Tabela 3 – Comparação da precisão entre os etiquetadores *MXPOST*, *Brill Tagger* e *Tree Tagger*

Fonte: Aires (2000, p. 82)

algumas das métricas usadas para medir o desempenho de classificadores binários e o que são aplicações de internet.

#### 2.4.1 PROCESSAMENTO AUTOMÁTICO DA LINGUAGEM NATURAL

Segundo Sardinha (2000), “[...] O Processamento [automático] de Linguagem Natural é uma disciplina com laços fortes com a Ciência da Computação e, embora compartilhe vários temas com a Lingüística de Corpus, as duas mantêm-se independentes.” (p. 328). Já de acordo com Oliveira (2009, p. 55), programas de processamento de linguagem natural são “[...] programas capazes de interpretar e/ou gerar informações em linguagem natural [...]”, onde (p. 55) “[...] a Linguística Computacional utiliza os corpora para poder ter acesso ao material que necessita estudar, ou seja, grande quantidade de textos que ocorrem naturalmente na língua.” Trask et al. (2004 apud ALUÍSIO; ALMEIDA, 2006, p. 158), por sua vez, defende que “a partir de corpora, podem-se fazer observações precisas sobre o real comportamento lingüístico de falantes reais, proporcionando informações altamente confiáveis e isentas de opiniões e de julgamentos prévios sobre os fatos de uma língua”. Tais observações seriam feitas com auxílio do resultado do processamento de um programa que trata os corpora, que por sua vez está codificado em linguagem natural. De todos estes autores, é possível concluir que o processamento automático da linguagem natural é o tratamento computacional de estruturas textuais que se repetem, por meio da detecção automática dos padrões dessas estruturas na língua falada e escrita de uma área do conhecimento de interesse a partir de registros relevantes e confiáveis do uso desta.

#### 2.4.2 LINGÜÍSTICA DE CORPUS

Um corpus, na perspectiva da Linguística de Corpus, pode ser definido como:

[...] um conjunto finito de enunciados tomados como objeto de análise. Mais precisamente, conjunto finito de enunciados considerados característicos do tipo de língua a estudar, reunidos para servirem de base à descrição e, eventualmente, à elaboração de um modelo explicativo dessa língua. Trata-se, pois, de uma colecção de documentos quer orais (gravados ou transcritos) quer escritos, quer orais e escritos, de acordo com o tipo de investigação pretendido. As dimensões do corpus variam segundo os objectivos do investigador e o volume dos enunciados considerados como característicos do fenómeno a estudar. Um corpus é chamado exaustivo quando compreende todos os enunciados característicos. E

é chamado selectivo quando compreende apenas uma parte desses enunciados. (GALISSON; COSTE, 1983 apud ALUÍSIO; ALMEIDA, 2006)

Conceito este que pode ser complementado com a ideia de que:

[...] corpus não pode ser considerado como constituindo a língua, mas somente como uma amostra da língua. (...) O corpus deve ser representativo, isto é, deve ilustrar toda a gama das características estruturais. Poder-se-ia pensar que as dificuldades serão levantadas se um corpus for exaustivo (...). Na realidade, sendo indefinido o número de enunciados possíveis, não há exaustividade verdadeira e, além disso, grandes quantidades de dados inúteis só podem complicar a pesquisa, tornando-a pesada. O lingüista deve, pois, procurar obter um corpus realmente significativo. Enfim, o lingüista deve desconfiar de tudo o que pode tornar o seu corpus não-representativo (método de pesquisa escolhido, anomalia que constitui a intrusão de lingüista, preconceito sobre a língua). (DUBOIS et al., 1993 apud ALUÍSIO; ALMEIDA, 2006)

Ou, de acordo com Sinclair (2005 apud ALUÍSIO; ALMEIDA, 2006) como uma coleção de peças de um texto na forma eletrônica, selecionadas de acordo com critérios externos para representar, o quanto melhor possível, ~~um idioma~~ ou variação deste como dados fonte para pesquisa linguística. Por Biber (1995 apud OLIVEIRA, 2009, p. 31), “um corpus linguístico de base computacional corresponde a coleções de textos que ocorrem naturalmente na língua, organizadas sistematicamente para representar áreas de uso da língua, e das quais podemos extrair novas informações”. Segundo Oliveira (2009), linguística de corpus é uma ciência empírica, ~~onde~~ o pesquisador utiliza diversas vezes o corpus para estender uma descrição linguística, mas esta prática pode ~~acabar por~~ deixar a teoria aberta para mudanças. Então é possível concluir que linguística de corpus é a área do conhecimento que estuda a língua através de agrupamentos de textos ~~da mesma~~ que representam um nicho a ser estudado.

Gries (2010) ainda cita três estatísticas deriváveis de uma análise de corpus, sendo ~~eles~~ a frequência de ocorrência de um certo fenômeno no texto, dispersão e frequências ajustadas e frequências de co-ocorrência. A primeira (frequência de ocorrência de um certo fenômeno no texto) é uma contagem de quantas ocorrências ~~alço~~ foram identificadas no texto, a segunda (dispersão e frequências ajustadas) é a proporção (percentual) das ocorrências dado um corpus e sua dispersão e ~~e~~ ~~último~~ (frequências de co-ocorrência) age sobre a frequência de co-ocorrência de expressões linguísticas. Portanto estas são as principais informações que uma ferramenta para a linguística de corpus deve apresentar.

### 2.4.3 SEPARADORES DE SENTENÇAS

Separadores de sentenças e separadores de frases são dois nomes para a mesma ferramenta. São programas que recebem como entrada um texto e o separam em frases (BRUCKSCHEN, 2010;

PAUMIER, 2016). Tais programas são úteis para o processamento ~~de um~~ corpus porque “ajudam a determinar as unidades para processamento linguístico” (PAUMIER, 2016, p. 33).

Tal diversidade de nomes ocorre devido ao falso cognato “*sentence*” significar “frase” e não “sentença”. A palavra inglesa “sentence”, de acordo com o dicionário Collins escolar inglês-português, pode ser traduzida tanto como “frase” quanto como “sentença”, mas todos os exemplos de uso dados a este último são relacionados ao contexto de trâmite jurídico na vara penal (COLLINS..., 2009, p. 364). Ainda assim, existem dicionários que reconhecem “sentença” e “frase” como sinônimos, como a definição “oração, frase, período” (AMORA, 2009, p. 670) presente no minidicionário Soares Amora da língua portuguesa e a definição “construção sintática com sentido completo, composta por uma ou mais palavras; frase” (DICIO, 2018) presente no dicionário online de português Dicio. De modo semelhante, mas contrário, tanto o novo dicionário Aurélio (FERREIRA, 1975, p. 670), quanto o dicionário Houaiss (HOUAISS et al., 2001, p. 2547) não reconhecem “frase” como uma sinônimo de “sentença”.

#### 2.4.4 MEDIDAS ESTATÍSTICAS DE AVALIAÇÃO DE DESEMPENHO DE CLASSIFICADORES BINÁRIOS

Para etiquetar um corpus automaticamente, é necessário um classificador, e é desejável que este atribua etiquetas corretamente a cada palavra e marca de pontuação. Embora a perfeição seja desejável, tal processo pode apresentar imperfeições (erros), como classificar “casa” enquanto substantivo quando deveria ter sido classificada como verbo (casar, flexionado na 3ª pessoa do singular do presente do indicativo). Para medir o quanto o classificador acerta ou erra para cada etiqueta, existem algumas métricas estatísticas **serão** utilizadas na seção 4.1 do capítulo de resultados e nos apêndices D, E e F; abaixo, segue a lista das métricas utilizadas (que não é exaustiva sobre o conjunto de todas as existentes):

- Precisão (*precision*), dado pela equação 1, “denota a proporção de casos preditos como positivos que são considerados positivos reais” (POWERS, 2011, p. 38).

$$Precisão = \frac{VerdadeiroPositivo}{VerdadeiroPositivo + FalsoPositivo} \quad (1)$$

- Revocação (*recall* ou sensibilidade), dado pela equação 2, “é a proporção de casos de casos positivos preditos como positivos” (POWERS, 2011, p. 38).

$$Recall = \frac{VerdadeiroPositivo}{VerdadeiroPositivo + FalsoNegativo} \quad (2)$$



- *F-measure* ( $F_1$  score), dado pela equação 3, “é definida como a média harmônica entre precisão e revocação” (RIJSBERGEN, 1979 apud BEITZEL, 2006, pp. 51, 52), mas, segundo POWERS (2011), não leva em conta os verdadeiros negativos.

$$F_1 = \frac{2 \times P_{\text{precisão}} \times R_{\text{recall}}}{P_{\text{precisão}} + R_{\text{recall}}} \quad (3)$$

- Acurácia (*accuracy*), dado pela equação 4, ao contrário das anteriormente apresentadas, “explicitamente toma em conta a classificação de negativos” (POWERS, 2011, p. 39).

$$A_{\text{curácia}} = \frac{V_{\text{verdadeiro}}P_{\text{positivo}} + V_{\text{verdadeiro}}N_{\text{negativo}}}{V_{\text{verdadeiro}}P_{\text{positivo}} + V_{\text{verdadeiro}}N_{\text{negativo}} + F_{\text{also}}P_{\text{positivo}} + F_{\text{also}}N_{\text{negativo}}} \quad (4)$$

- Taxa de erro (*miss rate*), dado pela equação 5, “é a proporção de positivos reais que são preditos como negativos” (POWERS, 2011, p. 39).

$$E_{\text{erro}} = \frac{F_{\text{also}}N_{\text{negativo}}}{V_{\text{verdadeiro}}P_{\text{positivo}} + F_{\text{also}}N_{\text{negativo}}} \quad (5)$$

- Especificidade (*specificity*), dado pela equação 6, “é a proporção de casos negativos que são corretamente preditos como negativos” (POWERS, 2011, p. 39).

$$E_{\text{especificidade}} = \frac{V_{\text{verdadeiro}}N_{\text{negativo}}}{F_{\text{also}}P_{\text{positivo}} + V_{\text{verdadeiro}}N_{\text{negativo}}} \quad (6)$$

- Prevalência (*prevalence*), dado pela equação 7 (POWERS, 2011, p. 39), representa a proporção do universo amostral que é composta por casos positivos, sendo uma métrica que independe do classificador.

$$P_{\text{prevalência}} = \frac{V_{\text{verdadeiro}}P_{\text{positivo}} + F_{\text{also}}N_{\text{negativo}}}{V_{\text{verdadeiro}}P_{\text{positivo}} + V_{\text{verdadeiro}}N_{\text{negativo}} + F_{\text{also}}P_{\text{positivo}} + F_{\text{also}}N_{\text{negativo}}} \quad (7)$$

## 2.4.5 APLICAÇÕES DE INTERNET

Uma ferramenta para a linguística de corpus cuja interface de operação seja acessível pela internet pode ser considerada uma aplicação de internet. A lei nº 12.965 as define como “o conjunto de funcionalidades que podem ser acessadas por meio de um terminal conectado à

internet” (BRASIL, 2014). Tal definição se aproxima da de terminal burro<sup>1</sup> no ponto em que uma considerável maioria dos dados e processamentos estão ou são realizados remotamente, numa arquitetura que cliente-servidor com um processo de aplicação executando no servidor.

#### 2.4.5.1 Arquitetura cliente-servidor

De acordo com Kurose e Ross (2013, p. 62), nesta arquitetura “há um hospedeiro sempre em funcionamento, denominado servidor, que atende a requisições de muitos outros hospedeiros, denominados clientes”, mencionando aplicações *Web* como um caso clássico de uso desta. Cada cliente e cada servidor possui uma pilha de protocolos da internet, assim como ilustrado pela figura 1 (KUROSE; ROSS, 2013, p. 37). A comunicação entre os processos executados no cliente e no servidor, ilustrada pela figura 2, ocorre através de uma interface de software de troca de mensagens denominada *socket* (KUROSE; ROSS, 2013, pp. 65, 73), e possibilita que tais máquinas troquem mensagens. No contexto de aplicações de internet, tais trocas de mensagens geralmente seguem o protocolo HTTP.

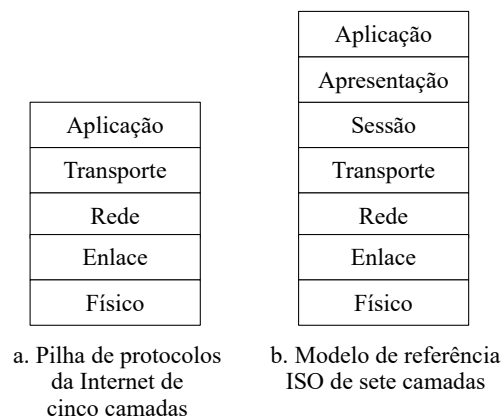


Figura 1 – A pilha de protocolos da internet (a) e o modelo de referência OSI (b)

Fonte: Kurose e Ross (2013, p. 37)

As mensagens do protocolo HTTP transportam as requisições feitas pelo cliente e as respostas dadas pelo servidor, tal como uma página *Web*, que é composta de objetos, como, por exemplo, “um documento HTML, uma imagem JPEG, um applet Java, ou um clipe de vídeo”(KUROSE; ROSS, 2013, p. 72). Kurose e Ross (2013) ainda afirma que “a maioria das páginas *Web* são constituídas de um arquivo-base HTML e diversos objetos referenciados” (p. 72), exibidas para o usuário através dum navegador, e que navegador e cliente referem-se à mesma coisa neste contexto, sendo o *Internet Explorer* e o *Mozilla Firefox* exemplos deste (KUROSE; ROSS, 2013). O servidor *Web* abriga objetos que possuem endereços (URL), sendo estes acessíveis pelo clientes através destes endereços (KUROSE; ROSS, 2013). São *softwares* populares deste tipo o

<sup>1</sup> *The Linux Information Project* (2005) define “terminal burro” como um terminal de computador que consiste apenas de um teclado e monitor, sem unidade de processamento central ou disco rígido interno, com pouco ou nenhum poder de processamento ou armazenamento.

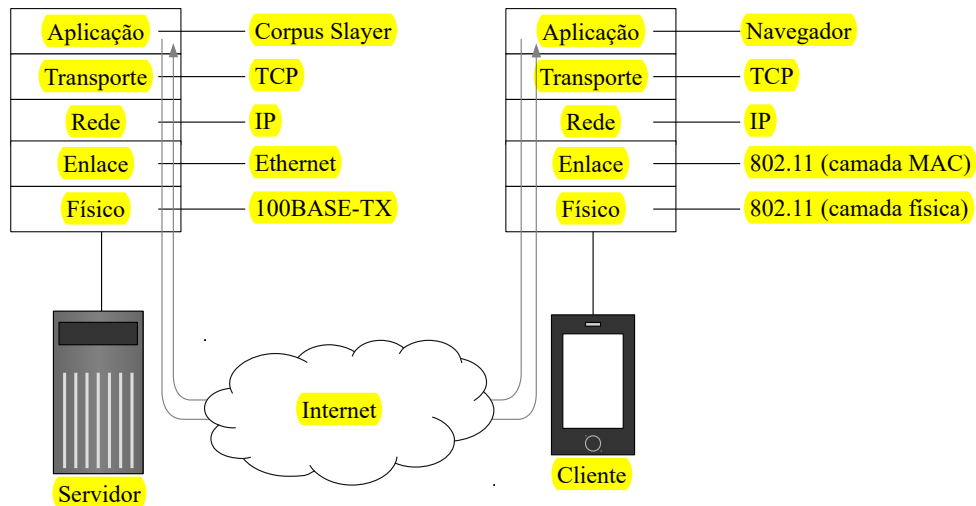


Figura 2 – Exemplo de canal de comunicação entre aplicações no cliente e no servidor

Fonte: O autor

*Apache* e o *Microsoft Internet Information Server* (KUROSE; ROSS, 2013), os quais são capazes de delegar a um algoritmo a geração da resposta a uma requisição (Apache Foundation, 2018; Microsoft, 2017). Algoritmos que geram a resposta a requisições de um cliente podem ser escritos utilizando o *framework* Django.

#### 2.4.5.2 *Framework* de aplicação de internet Django

Para facilitar o desenvolvimento de aplicações de internet, algumas *frameworks* foram criadas. Um desses *frameworks* é o Django, escrito na linguagem de programação Python, se autodefine como “*framework web* para perfeccionistas com prazos” e provê uma série de ferramentas prontas para agilizar o desenvolvimento da aplicação, como ferramentas de autenticação de usuário, um renderizador de *templates* e um mapeador objeto-relacional cujo banco de dados padrão é o SQLite (havendo suporte para outras soluções de persistência) (Django Software Foundation, 2018).

SQLite se autodefine como “uma biblioteca embutida no processo que implementa um banco de dados transacional SQL de maneira autocontida, sem servidor e sem configuração”, e também afirma ser “o banco de dados mais vastamente implementado no mundo” (The SQLite Consortium, 2018). Como solução de banco de dados, para Elmasri e Navathe (2012), é característica fundamental deste oferecer ao programador algum nível de abstração de dados. Para aumentar ainda mais a abstração além das técnicas apresentadas por Elmasri e Navathe (2012) nos capítulos onde aborda as técnicas de programação de banco de dados, Elmasri e Navathe instrui criar funções (ou métodos, dependendo da linguagem de programação) que encapsulam o dado usando consultas SQL escritas pelo programador e convertem registros em objetos e vice-versa. Contrariamente às práticas apresentadas por Elmasri e Navathe, O’Neil

(2008) afirma que “programadores veementemente preferem trabalhar com dados persistentes em objetos do programa a usar consultas SQL diretamente para acessar os dados” (p. 1351) e apresenta o mapeamento objeto-relacional como uma “metodologia e mecanismo para que sistemas orientados a objetos mantenham controle de seus dados persistidos seguramente no banco de dados” (p. 1351), sendo tal controle feito a nível transacional de forma a tratar condições de corrida, onde as “aplicações de internet dos dias atuais são particularmente adequadas a esta abordagem” (p.1351). Portanto, o *framework* Django *prover* uma abstração para toda a camada de persistência, mecanismos de migração automática, segurança contra ataques conhecidos e outras facilidades que permitem ao programador focar no problema a ser resolvido e não nos detalhes tecnológicos da implementação, cumpre melhor o papel de abstrair os dados defendido por Elmasri e Navathe (2012) que a técnica por eles apresentada.

Apesar da criação das tabelas e suas colunas poderem ser feitas completamente a critério do modelador do ponto de vista de código, a retenção de alguns dados do usuário se faz obrigatória do ponto de vista jurídico. A seguir será abordado quais dados são esses que devem ser armazenados e por quanto tempo devem ser armazenados.

#### 2.4.5.3 Deveres legais a respeito da mínima retenção de dados

Popularmente chamado de “Marco Civil da Internet”, a lei nº 12.965, “estabelece princípios, garantias, direitos e deveres para o uso da internet no Brasil” (BRASIL, 2014). Um desses deveres pode se aplicar aos provedores das aplicações de internet, quando este for uma pessoa jurídica atuando com fins econômicos:

Art. 15. O provedor de aplicações de internet constituído na forma de pessoa jurídica e que exerça essa atividade de forma organizada, profissionalmente e com fins econômicos deverá manter os respectivos registros de acesso a aplicações de internet, sob sigilo, em ambiente controlado e de segurança, pelo prazo de 6 (seis) meses, nos termos do regulamento. (BRASIL, 2014)

Onde o que constituem “registros de acesso a aplicações de internet” fora anteriormente definido no Art. 5º:

VIII - registros de acesso a aplicações de internet: o conjunto de informações referentes à data e hora de uso de uma determinada aplicação de internet a partir de um determinado endereço IP. (BRASIL, 2014)

Portanto, para uma aplicação de internet ser usável comercialmente **em território nacional** sem insegurança jurídica, deve ser auditável de forma a rastrear o endereço IP, data e hora de qualquer conteúdo gerado por qualquer usuário dentro da aplicação por **pelo menos** 6 meses.

### 3 DESENVOLVIMENTO

Este capítulo descreve as etapas do desenvolvimento da aplicação de internet proposta. A seção 3.1 descreve as etapas do levantamento de requisitos, as funcionalidades desejáveis identificadas, a 3.2, o gerenciador de corpora numa modelagem já utilizando a tecnologia escolhida, já a seção 3.3 contém um esboço da navegabilidade contemplando a estrutura de navegação da aplicação, a seção 3.4, descreve o subsistema de eventos que acaba por controlar o fluxo de informações dentro da plataforma de forma a reduzir o acoplamento entre componentes, favorecer a substituição de componentes e favorecer a interoperabilidade, a seção 3.5 traz a solução pronta a ser adaptada para fornecer ao sistema lista de sentenças e de palavras, a seção 3.6 contém a metodologia utilizada para treinar o etiquetador do Unitex/GramLab e mais um outro que foi desenvolvido durante este trabalho, e, por fim, a seção 3.7 traz a proposta do concordanciador e alguns requisitos extras pertinentes a apenas este módulo.

#### 3.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma das primeiras etapas do ciclo de vida de um projeto de software (SOMMERVILLE, 2011). Para tal, o primeiro passo foi pesquisar as ferramentas voltadas ao usuário final existentes. Excluídas as ferramentas de processamento de corpora de linha de comando, e excluídas as ferramentas que não possuíssem versão gratuita, de demonstração ou de testes, e tendo como ponto de partida os artigos do capítulo anterior, foram pesquisadas as ferramentas diretamente mencionadas nos artigos e também as ferramentas citadas nos *sites* que os artigos mencionam. As ferramentas encontradas foram: *WordSmith*, *CorpusEye*, *COCA Online Corpus*, *Unitex/GramLab* e *Sketch Engine*. Então, suas principais funcionalidades presentes na versão gratuita foram levantadas, comparadas, e então expressas na tabela 4, da qual foram extraídas, de forma não-exaustiva, características desejáveis de um sistema de processamento de corpora e listadas a seguir:

- Possuir interface web compatível com tamanhos de tela diversos (*smartphones*, *tablets*, *notebooks* e *desktops*);
- Processar corpora fornecidos pelo usuário;
- Etiquetar corpora fornecidos pelo usuário;
- Criar corpus a partir de buscas na internet;

	WordSmith 5	CorpusEye	COCA Online Corpus	Unitex/GramLab	Sketch Engine
Interface	Desktop Win32 Windows	Web Perl Ubuntu 16.04	Web Asp Windows Server 2012	Desktop Swing Java	Web Python 2.7 Fedora 25
Versão completa gratuita	não	sim	sim	sim	não
Corpus fornecido pelo usuário	sim	não	não	sim	sim
Etiqetagem do corpus fornecido pelo usuário	não	não	não	francês	sim
Criação de corpus a partir de buscas na internet	sim	não	não	não	sim
Frequência de palavras	sim	não	não	sim	sim
Frequência de lemas	sim	não	não	não	não
Contagem de palavras	sim	não	não	não	sim
Contagem de lemas	sim	não	não	não	não
Razão lemas e palavras	sim	não	não	não	não
Contagem de sentenças	sim	não	não	não	não
Contagem de parágrafos	sim	não	não	não	sim
Tamanho em caracteres das palavras	sim	não	não	não	sim
Caracteres por palavra	sim <sup>1</sup>	não	não	não	não
Palavras por sentença	sim <sup>1</sup>	não	não	não	não
Extração de palavras-chave	sim	não	não	não	não
Distribuição da palavra-chave no corpus	sim <sup>2</sup>	não	não	não	não
Palavras relacionadas à palavra-chave	sim	não	não	não	não
Busca por ocorrência	sim	sim	sim	sim	sim
Busca por expressão	apenas por radical	sim	apenas por etiqueta	sim	sim
Concordanciador	sim <sup>3</sup>	sim	sim	sim	sim
Posição da concordância no corpus	sim <sup>2,3</sup>	não	não	sim <sup>4</sup>	não
N-gramas	sim <sup>3</sup>	não	sim <sup>3</sup>	não	sim

Legenda:

- 1– com desvio padrão
- 2– com gráfico
- 3– sem etiquetas de parte do discurso
- 4– com clique que leva de volta ao corpus de entrada

Tabela 4 – Tabela comparativa de softwares de Processamento de Linguagem Natural

Fonte: O autor

- Calcular estatísticas quanto à frequência de lemas e palavras;
- Calcular estatísticas quanto à contagem de lemas e palavras;
- Calcular estatísticas quanto à razão de lemas e palavras;
- Calcular estatísticas quanto à contagem de sentenças e parágrafos;
- Calcular estatísticas quanto ao tamanho em caracteres das palavras;
- Calcular estatísticas quanto à quantidade de palavras por sentença;
- Extrair as palavras-chave de um corpus;
- Calcular estatísticas quanto à distribuição da palavra-chave no corpus;
- Calcular estatísticas quanto às palavras relacionadas às palavras-chave do corpus;
- Buscar por uma ocorrência de uma palavra ou sequência de palavras;
- Buscar por uma expressão que simbolize uma construção textual;
- Possuir um concordanciador;
- Calcular os n-gramas mais frequentes de um texto.

Como nenhum dos softwares apresenta extensibilidade através de *plug-ins*<sup>1</sup> como funcionalidade, tal requisito não está presente na lista acima. Considerando que tal lista, apesar de extensa, pode não cobrir todos os casos de uso ~~de um linguista~~ e que nem todas as funcionalidades serão implementadas neste trabalho, foi julgada como desejável facilitar a adição e substituição de componentes do **software** por outros mais atualizados, com maiores taxas de acerto, computacionalmente mais eficientes e/ou com licenças mais permissivas, a critério do administrador do servidor que executa o software. Portanto possuir extensibilidade através de *plug-ins* é um requisito do software.

<sup>1</sup> *Plug-ins* são componentes modulares de um software que modificam e/ou adicionam comportamentos, funcionalidades e/ou a aparência original do software, sendo também chamados de *add-ons* (termo utilizado no Mozilla Firefox), *add-ins* (termo utilizado no Microsoft Office Excel) e extensões (termo utilizado no Google Chrome).

Dos softwares anteriormente listados, foi observado na tabela 4 que nenhum deles possui um concordanciador gratuito com suporte a etiquetas de partes do discurso para a língua portuguesa que receba textos quaisquer fornecidos pelo usuário. Adicionando esta funcionalidade como objetivo, juntamente de suas dependências, temos:

- Possuir extensibilidade através de *plug-ins*;
- Possuir interface web compatível com tamanhos de tela diversos;
- Processar corpora fornecidos pelo usuário;
- Etiquetar corpora fornecidos pelo usuário;
- Possuir um concordanciador.

### 3.1.1 TECNOLOGIAS ESCOLHIDAS

Para gerar as páginas, foi escolhido o *framework* Django em sua versão de longo suporte 1.11. Esta escolha se deve principalmente à prévia familiaridade do autor com a ferramenta. Outros fatores que influenciaram a decisão é a quantidade de bibliotecas disponíveis para a linguagem em que o *framework* é escrito, Python, o que possibilita a inclusão destas nos *plug-ins*.

Para que as páginas geradas sejam compatíveis com tamanhos de tela diversos, foi escolhido o *framework* de folha de estilos Bootstrap em sua versão 4.0b1. Esta escolha também foi influenciada pela prévia familiaridade do autor. Em conjunto com a referida folha de estilos, também foi utilizado o pacote de ícones FontAwesome. Ele se define em sua página inicial como “um idioma pictográfico de ações relacionadas à *web*” (GANDY, 2017). Foi adicionado visando melhorar a usabilidade.

Para processar os corpora fornecidos pelo usuário, foi selecionado o utilitário de linha de comando do *software* Unitex/GramLab. É software livre licenciado sobre os termos da GPL, que não impõe restrições quanto ao seu uso. Apesar de possuir um utilitário chamado “Tagger”, somente durante a implementação foi observado que os arquivos necessários para sua utilização estavam presentes apenas para o idioma francês. Portanto, o etiquetador do Unitex/GramLab foi treinado para a língua portuguesa utilizando os corpora anotados pelo projeto Floresta Sintáctica (mais detalhes na seção 3.6); em caso de resultados insatisfatórios (menos que 75% de precisão) neste, seria utilizado os arquivos previamente treinados por Aires (2000), dando preferência aos etiquetadores com maior precisão.



### 3.2 O GERENCIADOR DE CORPORA

As classes do domínio do problema, quando representadas através da *framework* utilizada, se aproximam muito do modelo lógico. O mapeamento objeto-relacional feito pela plataforma admite herança e classes abstratas; como a solução de persistência padrão é SQLite, e esta não suporta herança entre tabelas, o mapeador objeto-relacional replica os campos definidos na classe pai na classe filho, já classes abstratas não geram tabelas. A figura 3 traz um excerto do modelo lógico gerado automaticamente a partir da implementação. A classe em magenta claro (*User*) é advinda do *framework*, já as demais classes em amarelo claro foram escritas para compor a base da plataforma, as quais os *plug-ins* utilizarão diretamente. Omitidas neste modelo resumido e presente no apêndice A, as classes com letras em cinza (*Timestampable*, *RequiresIP* e *AttributeMixin*) são abstratas e todas as classes concretas herdam de *RequiresIP*, devido às obrigações legais anteriormente mencionadas na seção 2.4.5.3.

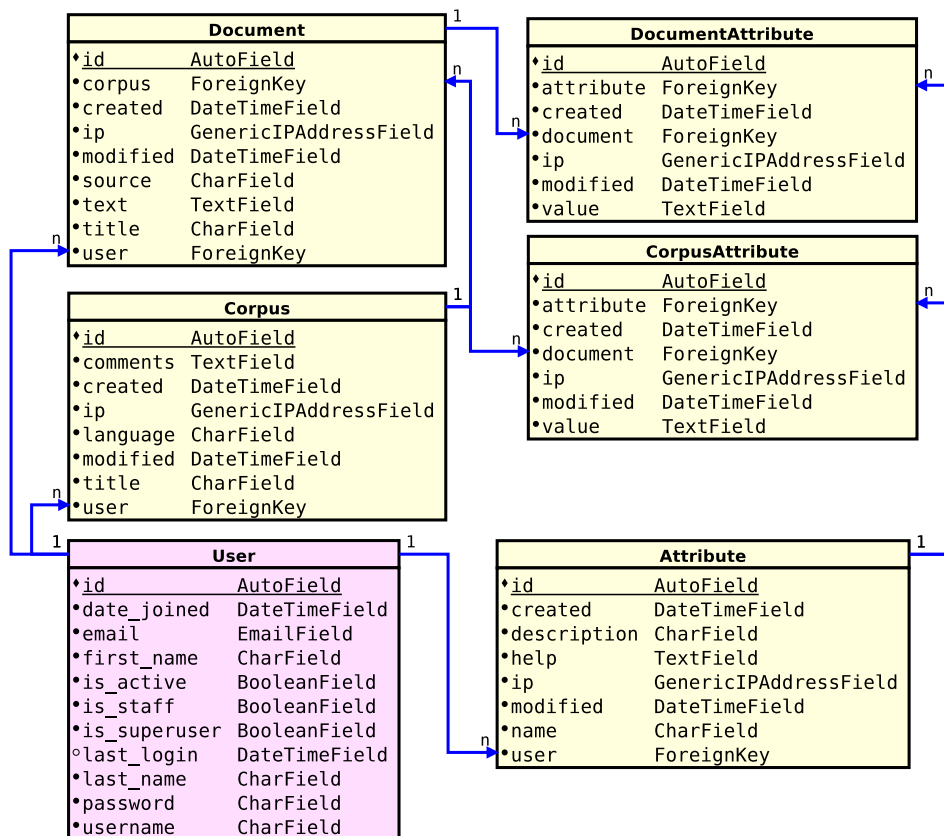


Figura 3 – Excerto do modelo lógico gerado automaticamente a partir da implementação

Fonte: O autor

### 3.3 ESBOÇO DA NAVEGABILIDADE

A navegabilidade no sistema é fundamental para a experiência de uso do sistema pelo usuário, já que trata de como o usuário irá interagir com o sistema e das opções visíveis a ele. Como os documentos de hipertexto formam um grafo, onde cada documento é um nó e cada link é

uma aresta (MEUSEL et al., 2015), páginas com informações relevantes ao usuário podem ficar muitos nós de distância de onde o usuário se encontra, mas também não deve ficar poluído de forma a deixar o usuário desorientado (KRUG, 2008). A forma que adotada para representar tal navegação é através duma árvore (vide figura 4), onde a página anterior conhece seus descendentes imediatos e sua página pai, apenas. No canto direito no rodapé de todas as páginas ficarão os *links* para as páginas de “Ajuda”, “Termos de uso” e “Política de privacidade”, cujo conteúdo será estático, definido nos arquivos de tradução. Na barra de navegação do topo de página ficará o seletor de idiomas alinhado à esquerda (caixa “Idioma”) e os botões que levam às páginas de autenticação (subcaixas da caixa “Contas”).

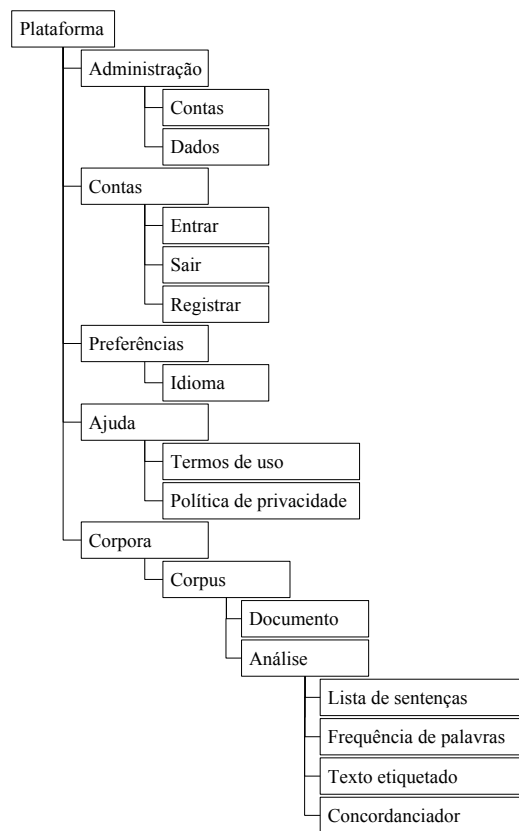


Figura 4 – Árvore de navegação das páginas do sistema

Fonte: O autor

A página inicial, representada pelas caixa “Plataforma” na figura 4, será a página inicial; para visitantes, a página deverá exibir um formulário de *login* e um botão que leva ao formulário de cadastro, enquanto, para um usuário autenticado, representa a página “Corpora” da figura 4, exibindo uma lista com os corpora por ele já criados e um botão para criar um novo corpus vazio. Ao clicar num corpus, o usuário entrará numa outra página (“Corpus”) que conterà os documentos que o compõem o corpus selecionado, juntamente a botões para editar e apagar o corpus selecionado, bem como para cada documento listado (“Documento”), além de um botão para adicionar um documento e outro para levar até a página com as operações que

podem ser feitas sobre o corpus (“Análise”). Na página de análises, todos os botões de ação são fornecidos por *plug-ins*, mas é esperado que haja ao menos um para uma lista de sentenças, um para frequência de palavras, um para o texto etiquetado e outro para o concordanciador, sendo permitido ao usuário localizar o(s) documento(s) que contém o que pode ser visto na tela num clique, no máximo dois.

### 3.4 O SUBSISTEMA DE EVENTOS

Como ser extensível através de *plug-ins* é um requisito da aplicação e este recurso não é oferecida pelo *framework* sem que seja necessário alterar o arquivo de configuração, somado à vantagem de que os *plug-ins* sejam substituíveis, foi ~~julgado preferível~~ desenvolver tal parte do sistema utilizando uma arquitetura que favoreça o desacoplamento. Para tanto, foi escolhida uma arquitetura baseada em eventos, onde cada *plug-in* forneça e/ou consuma serviços de outros *plug-ins* utilizando apenas eventos, sem consultar diretamente o outro *plug-in* ou ser informado sobre quais estão instalados ou não.

A figura 5 mostra a árvore de diretórios em amarelo e seus *scripts* em ciano. Na pasta “projeto” foram adicionados dois *scripts*: “gancho padrão” e “eventos”. O primeiro contém um gerador da estrutura que descreve um botão na página “Análise”. O segundo contém uma função “disparar” que recebe dois argumentos: uma cadeia de caracteres e um objeto qualquer; esta função dispara o evento e seu retorno depende da cadeia de caracteres de entrada, que especifica o tipo de evento: provedor, busca, filtro ou ação.

Eventos do tipo “provedor” descartam o segundo argumento de entrada e retornam uma lista. Foram idealizados para abastecer *plug-ins* que aceitam recursos linguísticos além dos embarcados nele mesmo. Um exemplo de aplicação é um etiquetador que, através da resposta a eventos por ele disparados, recebe recursos para etiquetar idiomas originalmente não suportados.

Eventos do tipo “busca” recebem o objeto que está relacionado ao desejado no segundo argumento de entrada e retornam uma lista contendo o(s) resultado(s) de cada *plug-in* que ofereceu aquele serviço de busca. Foram idealizados para buscar dados em outros *plug-ins*, sem a necessidade de conhecer todos os modelos destes e sua representação interna. Um exemplo de aplicação é no concordanciador, onde ele fornece um corpus e recebe como resposta o *cache* do corpora já etiquetado por cada etiquetador.

Eventos do tipo “filtro” recebem uma coleção de objetos no segundo argumento de entrada e retornam a uma coleção de tamanho menor ou igual, podendo ter elementos alterados, sendo a saída de um filtro a entrada do próximo. Foram idealizadas como forma de reduzir o conjunto de entrada, através de regras implementadas por outros *plug-ins*. Um exemplo de aplicação são diversos tokenizadores atuando entre a lista de sentenças e o etiquetador, para garantir que a

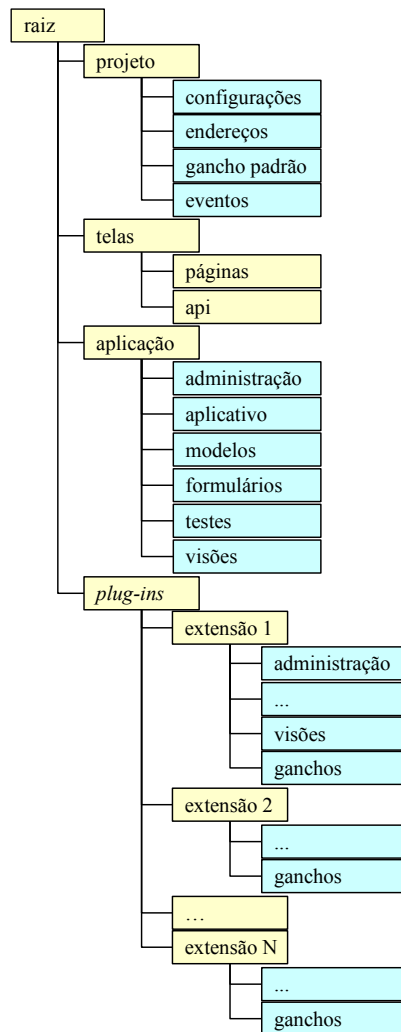


Figura 5 – Árvore de diretórios e *scripts*

Fonte: O autor

entrada do etiquetador seja consistente.

Eventos do tipo “ação” recebem um objeto qualquer e não retornam nada. Foram idealizados para notificar outros *plug-ins* que uma ação foi concluída com sucesso, ou até mesmo para comunicação interna dentro do próprio *plug-in*. Um exemplo de aplicação é num processamento conhecidamente demorado que é realizado em segundo plano e a notificação de sua conclusão se dá através de um *e-mail*.

Tais eventos ocorrem dentro do subsistema de *plug-ins*, cuja sequência de funcionamento é expressa em detalhes no apêndice B. Vale mencionar que não foi implementada nenhuma verificação para garantir que não há dependência cíclica; nesses casos, é esperado que um erro da classe “*RecursionError*” seja lançado e, caso não capturado, retorne um erro HTTP de código 500.

### 3.5 SEPARAÇÃO DE SENTENÇAS E OBTENÇÃO DE LISTA DE PALAVRAS

Os dados da lista de sentenças e a lista de palavras, sendo o primeiro a entrada do etiquetador, são ambos produtos do pré-processamento do Unitex/GramLab (PAUMIER, 2016). Com isso, o trabalho foi apenas adaptar a visualização do dado à nova interface. A representação escolhida para a lista de sentenças foi a do elemento HTML de lista ordenada, onde cada sentença é um item da lista, dispostas na ordem que aparecem no texto. A representação da lista de palavras na interface acontece em 3 listas, uma de palavras simples (“roupa”, por exemplo), outra de palavras compostas (“guarda-roupa”, por exemplo) e a última para palavras não reconhecidas, onde as duas primeiras possuem cores que identificam o lema, as etiquetas morfológicas e semânticas atribuíveis à palavra encontrada no texto; na página exibida ao usuário, cada lista com cores foi substituída por uma tabela que identifica adequadamente o significado de cada cor, ordenada do mais para o menos frequente, e a lista de palavras desconhecidas foi representada de forma semelhante à lista de sentenças, mas numa lista não-ordenada e em ordem alfabética.

### 3.6 TREINO DO ETIQUETADOR

Conforme mencionado anteriormente no final da seção 3.1, o etiquetador do Unitex/GramLab foi treinado utilizando os corpora anotados do Floresta Sintáctica, que oferece os corpora nos formatos CoNLL e Árvore Deitada para todas as bases, estando demais formatos restritos a apenas um subconjunto das bases fornecidas. O formato inicialmente escolhido foi o CoNLL, mas rapidamente ele demonstrou não possuir uma estrutura consistente (a exemplo, as *tags* que se assemelham à notação XML que determinam o início das sentenças possui um atributo ora nomeado “text”, ora nomeado “texto”). Portanto, foi utilizado os arquivos no formato de árvore deitada.

Após converter os corpora para o formato de entrada mais simples que o treinador do Unitex/GramLab aceita<sup>2</sup>, era o momento de separar em treino e teste, mas não era sabido quais corpora gerariam erros devidos a eventuais erros na conversão do formato não documentado. A solução adotada foi testar cada corpus anotado: caso o treino concluísse com sucesso era separado para o conjunto treino, caso contrário, para o de teste. Resultados no quadro 1.

O conjunto de treino (quadro 1) continha 1.286.251 palavras anotadas distribuídas ao longo de 81.438 sentenças; já o conjunto de testes continha 4.585.010 palavras distribuídas ao longo de 266.500 sentenças. Devido a diferenças no tokenizador e separador de sentenças, o teste etiquetado continha 4.726.605 palavras anotadas ao longo de 245.718 sentenças. Separando apenas as sentenças que foram *tokenizadas* da mesma forma que no conjunto de testes, tem-se

<sup>2</sup> O formato de entrada mais simples que o treinador do Unitex/GramLab aceita é uma lista de sentenças separada por duas quebras de linha seguidas, onde cada sentença é uma lista de palavras etiquetadas, que por sua vez é separada por uma quebra de linha, onde cada palavra etiquetada é um par palavra-etiqueta separados pelo caractere barra (“/”) (PAUMIER, 2016).

Treino	Teste
amaz1.ad	amaz2.ad
amaz6.ad	amaz3.ad
bosquecf.ad	amaz4.ad
selva_cien.ad	amaz5.ad
selva_fala.ad	bosquecp.ad
selva_lit.ad	fv_cf.ad
	fv_cp.ad

Quadro 1 – Separação entre treino e teste

Fonte: O autor

2.801.255 palavras em 176.034 sentenças. Como cada palavra é uma amostra para o treino do etiquetador, tem-se a proporção treino/teste de aproximadamente 31% / 69%.

Apesar de ter sido treinado para usar as etiquetas do Projeto Floresta Sintáctica, o Unitex/Gram-Lab usou do próprio conjunto de etiquetas para etiquetar o texto, estas sendo incompletamente documentadas apenas pelo conteúdo do quadro 2. Por isso, durante a etapa de avaliação de desempenho foi necessário aproximar ambos padrões de etiquetas utilizando como referência o “glossário de etiquetas florestais” da Linguateca (2010a) e a breve e incompleta tabela de Paumier (2016) (quadro 2), onde etiquetas não documentadas tiveram seu significado manualmente inferido (a exemplo, a etiqueta “X” provavelmente representa palavras estrangeiras, como “CD-ROM”). A conversão final antes da comparação e montagem das tabelas de contingência é expressa nas tabelas 5 e 6 (teste 1, resultados na seção 4.1.1).

Etiqueta	Descrição	Exemplos
A	adjetivo	<i>fabulous, broken-down</i>
ADV	advérbio	<i>actually, years ago</i>
CONJC	conjunção coordenativa	<i>but</i>
CONJS	conjunção subordinativa	<i>because</i>
DET	determinante	<i>each</i>
INTJ	interjeição	<i>eureka</i>
N	substantivo	<i>evidence, group theory</i>
PREP	preposição	<i>without</i>
PRO	pronome	<i>you</i>
V	verbo	<i>overeat, plug-and-play</i>

Quadro 2 – Etiquetas gramaticais frequentes

Fonte: Paumier (2016, p. 47)

O etiquetador do Unitex/GramLab (PAUMIER, 2016) usa “tuplas de unigramas, bigramas e trigramas” (p. 329) gerados a partir do corpus fornecido como treino para que a aplicação, durante

Etiqueta	Convertido
PROP	N
VAUX	V
PP	PREP
PRP	PREP
DET	PRON
INTJ	INTERJ

Tabela 5 – Conversão da etiquetas utilizadas do Floresta Sintáctica para notação intermediária

Fonte: O autor

Etiqueta	Convertido
A	ADJ
X	N
SIGL	N
ABREV	N
PRO	PRON
PREPXD	PREP
PREPXPRO	PREP
PONCT	???

Tabela 6 – Conversão da etiquetas utilizadas do Unitex/GramLab para notação intermediária

Fonte: O autor

a etiquetagem, “aplique o algoritmo do caminho de Viterbi sobre o corpus e produza um autômato linear” (p. 294), ou seja, o processo de etiquetagem apenas remove as etiquetas atribuídas anteriormente durante o processamento feito ao carregar o corpus na ferramenta, no qual usa por padrão os dicionários nele embutidos, até restar apenas uma etiqueta por palavra. Analogamente ao problema dos significados das etiquetas, Paumier é vago sobre quais as fontes utilizadas para construir os dicionários numa tabela cuja descrição é “algumas referências bibliográficas para os dicionários eletrônicos” (p. 74), onde é passada a ideia de que é uma lista potencialmente incompleta, e não menciona nenhum processo de revisão para adequar eventuais diferenças semânticas no conjunto de etiquetas utilizadas pelos cinco diferentes trabalhos citados. Dos cinco, apenas o de Muniz, Nunes e Laporte (2005b) possui acesso público e menciona apenas o nome completo das 13 classes de etiquetas utilizadas: abreviação, adjetivo, advérbio, determinador, conjunção, interjeição, numeral, prefixo, preposição, pronome, acrônimo, substantivo e verbo; a abreviação que o software retorna ao usuário não está presente no artigo, e o *link* mencionado na seção “contribuições para o *software* UNITEX” (<http://www.nilc.icmc.usp.br:8180/unitex-pb/>) retorna um erro informando que o objeto requisitado não foi encontrado no servidor. Portanto, diante da limitação de compreensão da semântica das etiquetas imposta pela falta de documentação do software e pela impossibilidade de acessar gratuitamente os outros quatro artigos referenciados pelo manual, então mesmo sabendo que o ideal seria converter todo o

conjunto de etiquetas do Projeto Floresta Sintáctica para a notação do Unitex/GramLab antes do treino, a conversão foi feita durante a etapa de testes; é previamente sabido que o desempenho será prejudicado.

Um outro etiquetador de *design* simples que pode ser construído é um baseado em casamento de padrões. Nomeado *YAS-Tagger* (Mais Um Etiquetador Simples de Partes do Discurso, do inglês *Yet Another Simple Part-Of-Speech Tagger*), foi escrito para possuir formatos de entrada e saída similares ao do Unitex/GramLab de forma a reaproveitar sem adaptações os *scripts* de avaliação de desempenho escritos para o teste anterior (teste 2, resultados na seção 4.1.2). Internamente, faz o casamento de padrões sem diferenciar maiúsculas de minúsculas usando uma tabela associativa que associa tuplas à sua etiqueta mais frequente, construída durante o treinamento. Durante a etiquetagem, o texto de entrada é agrupado em tuplas de tamanhos entre 1 e 3, consultando da maior para a menor, atribuindo a etiqueta armazenada ou “???” caso nenhum padrão seja encontrado.

Para responder uma pergunta que surgiu apenas após a realização do teste anterior a respeito da influência do tamanho do corpus de treinamento no desempenho do etiquetador, foi realizado mais uma avaliação, esta utilizando o corpus anotado por Aires (2000), este disponibilizado publicamente pelo Núcleo Interinstitucional de Linguística Computacional (2012), e o mesmo algoritmo etiquetador do teste anterior. A base utilizada possui 103.666 etiquetas distribuídas ao longo de 4.713 sentenças, sendo separados 41.398 etiquetas em suas 1.915 sentenças para treino e 62.268 etiquetas em suas 2.798 sentenças para teste (proporção treino/teste de 40% / 60%), mas, devido a diferenças no alinhamento das etiquetas dos conjuntos das etiquetas de teste e predição, foram comparadas 55.808 etiquetas em suas 2.581 sentenças para teste. Portanto, a proporção treino/teste desta avaliação de desempenho foi 43% / 57% (teste 2.1, resultados na seção 4.1.2.1).

### 3.7 A PROPOSTA DO CONCORDANCIADOR

O concordanciador, um dos requisitos desta aplicação, como definido por Tagnin (2010), é o “programa que extrai todas as ocorrências de uma palavra de busca num corpus juntamente com seu cotexto [...]” (p. 358), ou seja, um buscador de padrões. Um problema observado enquanto usuário das outras ferramentas para confecção da tabela 4 da seção 3.1 é que não havia nenhum *feedback* para o usuário se o que estava sendo digitado era válido ou o que aquela busca significa, portanto o módulo a ser desenvolvido deve solucionar esse problema de usabilidade. Outro problema identificado é que a página onde a busca é digitada não possuem uma referência rápida com todas as principais construções válidas de interesse do usuário, mesmo quando existe espaço livre para tal conteúdo; também é outro problema que deve ser resolvido. Foi idealizado que este seria executado após o etiquetador, atuando por sentença, podendo filtrar por palavras inteiras, segmentos destas ou simplesmente especificar uma distância ente palavras e, ao mesmo tempo,



podendo filtrar resultados por etiquetas.

Tendo como entrada o corpus previamente etiquetado, para separar a palavra da etiqueta foi reservado o símbolo de sublinhado duplo (“\_\_”), já para indicar que é um segmento (e não uma palavra inteira) foi reservado o símbolo de ponto final duplo (“..”), para indicar que é esperado um número fixo ou variável de etiquetas foi reservado a construção “{exato}” e “{inicio, fim}”, respectivamente, onde **fim** pode ser indefinido (até o final da sentença) se usado o caractere asterisco. A título de exemplo, nestas condições, a busca “..ado\_\_VERBO {0,1} terrorista {0,\*} ..mic..” reconheceria a sentença “Um atentado terrorista realizado em nome do Estado Islâmico feriu 15 pessoas esta tarde” se, e somente se “atentado” seja etiquetado como verbo, bem como “Seu Zé foi fichado como terrorista durante a ditadura pelas opiniões que propagava que disse nos microfones em rede nacional” é uma sentença que é possível de ser reconhecido por esta busca.

## 4 RESULTADOS

Neste capítulo, serão apresentados e discutidos os resultados dos testes cujos métodos foram descritos e da implementação brevemente descrita no capítulo anterior, estando o primeiro na seção 4.1 e o segundo na seção 4.2.

### 4.1 TREINO DO ETIQUETADOR

Nesta seção serão discutidos o desempenho dos etiquetadores nos testes baseado nos gráficos de cada seção e nas tabelas de resultados presentes nos apêndices D, E e F, estas, medidas através das estatísticas apresentadas na seção 2.4.4, onde os ~~os~~ espaços em branco nas tabelas são resultados de divisão por zero ou cujo resultado não seja um número real ( $\mathbb{R}$ ). Em momentos onde o desempenho da implementação é mencionado, o computador *desktop* utilizado possui as configurações expressas no apêndice C.

#### 4.1.1 TESTE 1: ETIQUETADOR DO UNITEX/GRAMLAB

Das 2.801.233 etiquetas comparadas, apenas 1.701.989 foram atribuídas corretamente (*Precisão*  $\approx$  60,76%). Na tabela 9 (apêndice D), encontram-se as estatísticas de cada etiqueta derivadas a partir da tabela de contingência (tabela 8, apêndice D), a qual mostra que apenas as etiquetas “PREP”, “PRON” e “V” demonstraram desempenho satisfatório (precisão maior que 75%), que também estão entre as quatro etiquetas mais prevalentes tanto no conjunto de treino quanto no conjunto de testes, como pode ser observado na figura 6, já que é esperado que o etiquetador etiquete mais corretamente as etiquetas ~~que forma~~ mais frequentes no treino; e a etiqueta mais frequente tanto no treino, quanto no teste, obteve uma precisão ruim (59,25%).

Outro ponto relevante é que o treino deste etiquetador levou alguns poucos segundos, mas, usando do mesmo *pipeline* que seria executado do lado do servidor caso fosse implementado, o fez levar mais de 72 horas para concluir o processamento, onde utilizava, num momento arbitrário de observação já no segundo dia de execução, 14GB da memória principal mais 30GB de *swap*. Portanto, por conta destes problemas e dos apresentados na última seção do capítulo anterior, este etiquetador com esta base de dados não apresentou características atrativas para entrar na solução desenvolvida.

#### 4.1.2 TESTE 2: ETIQUETADOR YAS-TAGGER

Das 4.571.238 etiquetas comparadas, 3.516.634 foram atribuídas corretamente (*Precisão*  $\approx$  76,93%). Na tabela 11 (apêndice E), encontram-se as estatísticas de cada etiqueta derivadas a par tabela de contingência (tabela 10, apêndice E), nesta estão destacadas em cinza algumas

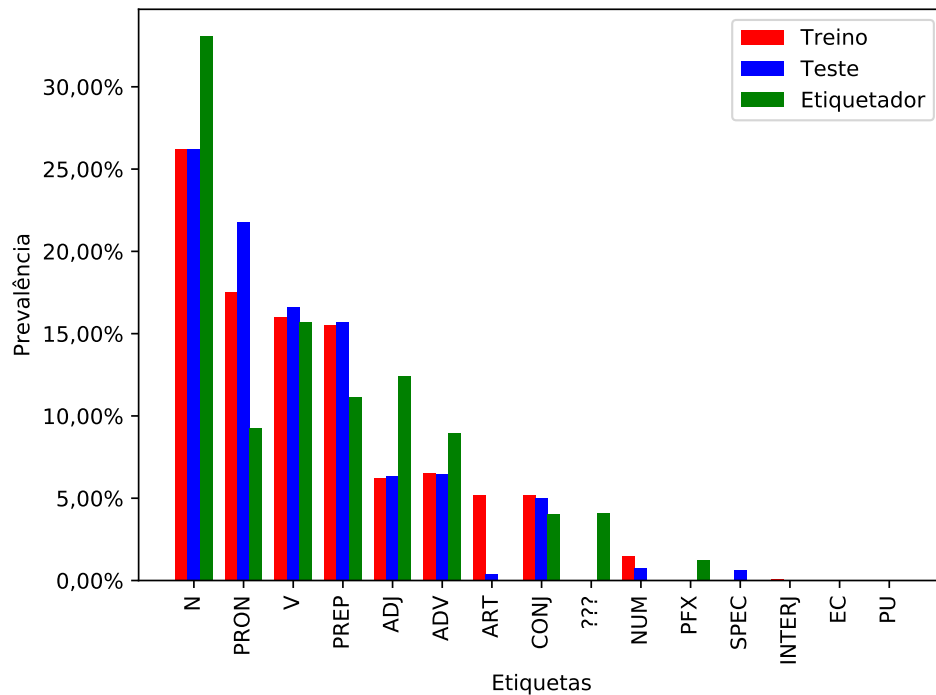


Figura 6 – Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo *Unitex/Gram-Lab*

Fonte: O autor

linhas que são erros advindos de várias conversões de formatos onde o caractere barra (“/”), usado em datas e outros, colide com o separador usado para distinguir excertos do texto de suas respectivas etiquetas; tais erros somam 52 ocorrências distribuídas ao longo de todo o corpus de teste, totalizando aproximadamente 0,0011375% da amostra e, portanto, possuem impacto desprezível sobre as métricas exibidas na precisão de casas decimais apresentadas. Também é possível observar que o etiquetador como um todo demonstrou desempenho satisfatório ( $F_1$  score maior que 75% e acurácia maior que 97,5%), mas também errou 14,13% das etiquetas atribuindo a etiqueta “???”. Tal etiqueta é reservada para significar que não foi possível atribuir nenhuma outra etiqueta, sendo consequência dum treino que não recebeu a construção encontrada no texto durante o teste; o corpus possui um número na ordem de milhão de etiquetas, mas se fosse da ordem de grandeza menor (centena de milhar), o quanto isso afetaria o desempenho negativamente?

#### 4.1.2.1 Teste 2.1: Etiketador *YAS-Tagger* sobre corpus de Aires

Das 55.808 etiquetas comparadas, apenas 29.799 foram atribuídas corretamente (*Precisão*  $\approx$  53,40%). Tabelas de contingência e suas estatísticas encontram-se no apêndice F. Observada essa queda de precisão de 23,56%, os erros advindos da etiqueta “???” agora são 33,74% (aumento de aproximadamente 138%), que a faz a etiqueta mais prevalente na saída do etiquetador (vide

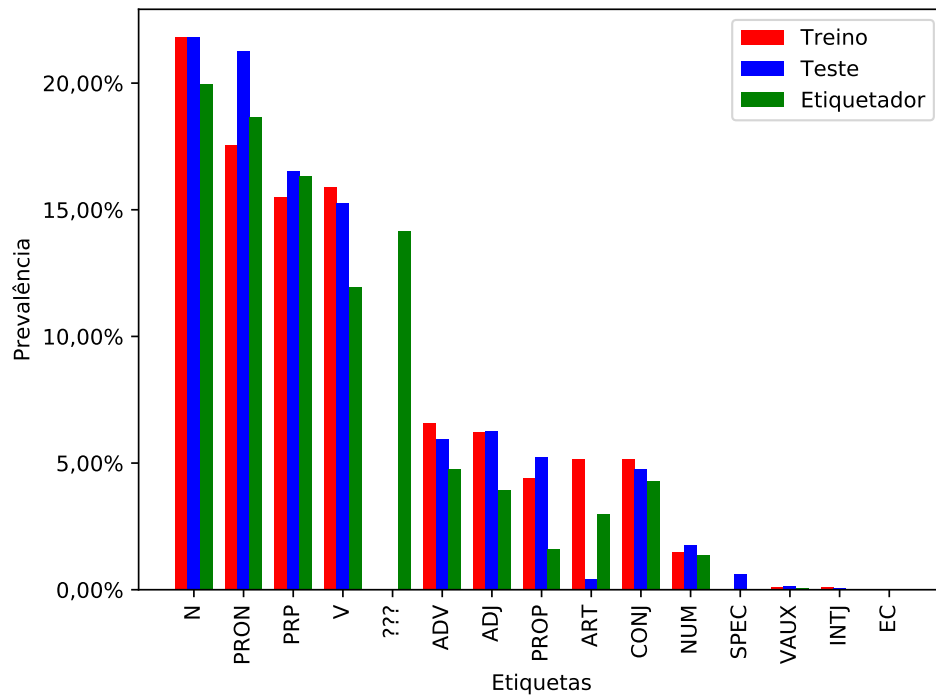


Figura 7 – Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo YAS-Tagger

Fonte: O autor

figura 8). Tais mudanças indicam que este etiquetador é altamente dependente de seu corpus de treinamento, e que este deve ser suficientemente volumoso para cobrir todos os casos de uso da língua para o contexto em que este pretende ser utilizado; como a aplicação de internet em que este etiquetador seria incluído não possui um contexto de aplicação previamente determinado, outro algoritmo terá preferência.

Concluídos os testes, com os dados da tabela 3 (da seção 2.3.5), pode-se montar a tabela 7 com os melhores resultados dos etiquetadores testados.

Etiquetador	Precisão
MXPOST	89,66%
Brill Tagger	88,76%
Tree Tagger	88,47%
YAS-Tagger	76,93%
Unitex/GramLab	60,76%

Tabela 7 – Comparação da precisão entre os etiquetadores MXPOST, Brill Tagger, Tree Tagger, YAS-Tagger e Unitex/GramLab

Fonte: O autor e Aires (2000, p. 82)

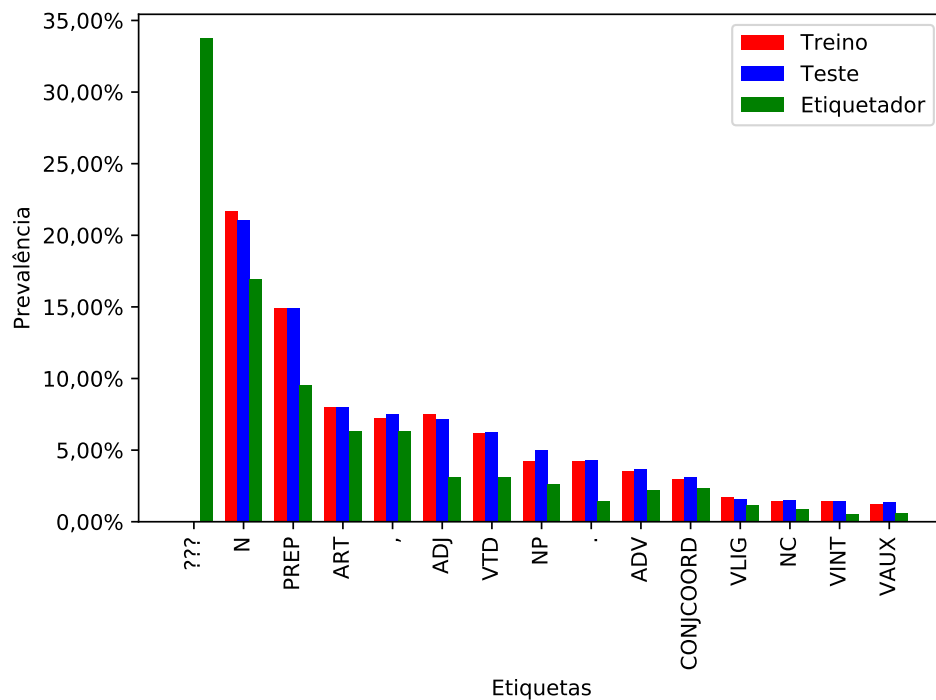


Figura 8 – Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo *YAS-Tagger* sobre o copus de Aires

Fonte: O autor

## 4.2 O SISTEMA DESENVOLVIDO

Nesta seção será apresentado o sistema enquanto aplicação implantada, bem como as decisões e soluções para os problemas apresentados no capítulo anterior. Apesar de não ter sido mencionado anteriormente, durante a codificação foi utilizado o sistema de versionamento de código GitLab, com implantação contínua automatizada usando BuildBot e servidor *web* NGINX, num servidor privado virtual alugado da empresa de hospedagem OVH, disponibilizado o acesso através do domínio “corpusslayer.com”.

Começando da página principal (figura 9), esta traz os principais elementos que se repetem em todas as páginas: na barra de navegação, o nome da aplicação e o seletor de idiomas (em detalhe na figura 10), no rodapé, um texto estático alinhado à esquerda arbitrariamente definido durante a tradução e três *links* que levam a páginas cujo ~~ambos~~ conteúdo e título são definidos na tradução. Porque o formulário de *login* é exibido nesta tela juntamente ao botão que leva ao formulário de cadastro, está ausente nesta página, mas a figura 11 traz esses botões em detalhe.

Após autenticar-se na aplicação, ao usuário é exibido uma tela como a da figura 12, a qual mostra a lista de corpora que o usuário criou e um botão para adicionar um corpus vazio sugestivamente nomeado “Adicionar corpus”. Clicando no botão “Detalhes”, o usuário é levado à lista de

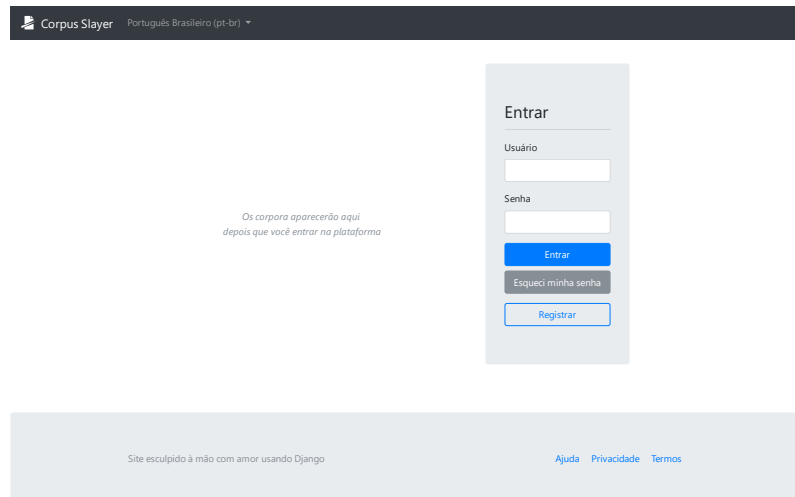


Figura 9 – Página inicial do sistema desenvolvido

Fonte: O autor

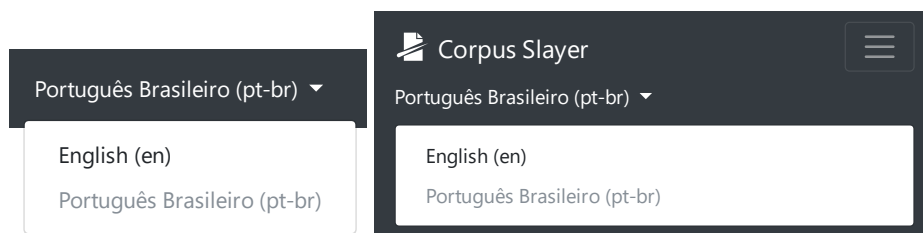


Figura 10 – Detalhe do seletor de idiomas, *desktop* à esquerda e em dispositivos móveis à direita

Fonte: O autor



Figura 11 – Detalhe dos botões de autenticação, *desktop* à esquerda, sendo encontrado alinhado à direita na mesma barra que aparece o seletor de idiomas, e em dispositivos móveis à direita, sendo encontrado imediatamente abaixo do seletor de idiomas

Fonte: O autor

documentos daquele corpus, como na figura 13, que permite editar e deletar o corpus, adicionar, editar e deletar documentos, além exibir um botão para ações extras que leva à página nomeada “Análise”.

Na página de análises (figura 14), são encontrados os serviços dos *plug-ins* desenvolvidos já aplicando o abordado na seção 3.4. Nas seções seguintes serão abordados o *Unitex/GramLab* como separador de sentenças e gerador de lista de palavras (seção 4.2.1), o *MXPOST* e o *Tree*

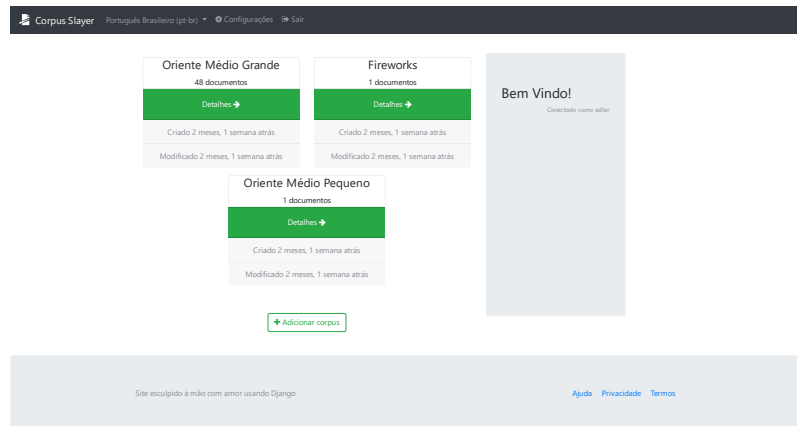


Figura 12 – Lista de corpora dum usuário

Fonte: O autor

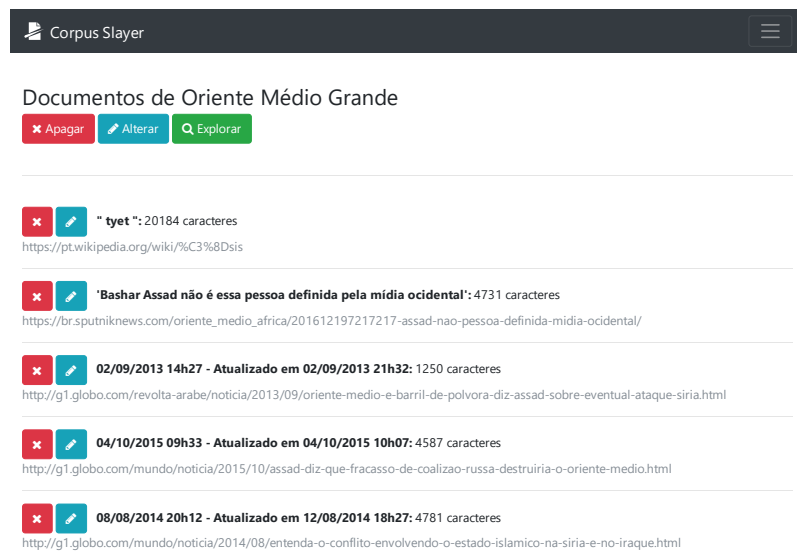


Figura 13 – Lista de documentos dum corpus

Fonte: O autor

*Tagger* enquanto etiquetadores (seção 4.2.2) e a implementação do concordanciador proposto na seção 3.7 (seção 4.2.3).

#### 4.2.1 SEPARADOR DE SENTENÇAS E GERADOR DE LISTA DE PALAVRAS

Usando o *Unitex/GramLab* como a parte que faz, de fato, o processamento dos dados, restou apenas ~~apenas~~ encapsular o dado e ~~exibi~~-los ao usuário. A primeira parte foi solucionada escrevendo ~~código~~ código a partir do manual de Paumier (2016), que serviu de base para a realização da segunda parte. ~~A segunda parte~~ teve fortes influências da interface do *Unitex/GramLab*, ~~onde~~ a lista de sentenças (figura 15) também ~~exibe~~ o número sequencial do item da lista e a lista de palavras (figura 16), que uma janela composta de três listas, tornou-se uma página com três



Figura 14 – Opções de análises para um corpus

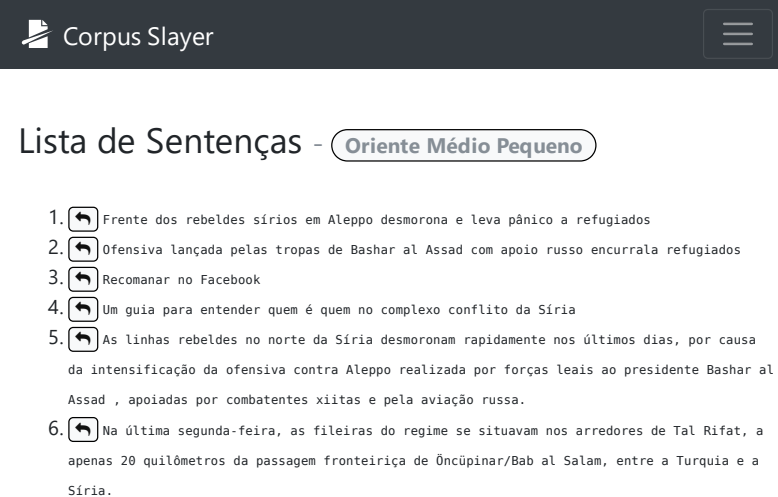
Fonte: O autor

seções (são elas: “Palavras não reconhecidas”, “Palavra Composta” e “Palavra Simples”) que abrigam as lista e cada lista teve seu conteúdo reorganizado em formato de tabela.

## 4.2.2 ETIQUETADOR

A primeira tentativa de integrar um etiquetador à ferramenta foi o *MXPOST*, que, para textos da língua portuguesa, retornava o erro da figura 17, mas foi mantido por funcionar para etiquetar com sucesso textos na língua inglesa. O segundo foi o *Brill Tagger*, que requer ~~fornece~~ permissão de execução (figura 18) a arquivos cujo nome sugere que abrigarão conteúdo fornecido pelo usuário, o que pode gerar possibilidade de execução remota de código no servidor se assumir que o usuário sempre será mal-intencionado, descartando-o como possibilidade. O terceiro foi o *Tree*






Lista de Sentenças - **Oriente Médio Pequeno**

1. Frente dos rebeldes sírios em Aleppo desmorona e leva pânico a refugiados
2. Ofensiva lançada pelas tropas de Bashar al Assad com apoio russo encorrala refugiados
3. Recomanar no Facebook
4. Um guia para entender quem é quem no complexo conflito da Síria
5. As linhas rebeldes no norte da Síria desmoronam rapidamente nos últimos dias, por causa da intensificação da ofensiva contra Aleppo realizada por forças leais ao presidente Bashar al Assad , apoiadas por combatentes xiitas e pela aviação russa.
6. Na última segunda-feira, as fileiras do regime se situavam nos arredores de Tal Rifat, a apenas 20 quilômetros da passagem fronteiriça de Ôncüpinar/Bab al Salam, entre a Turquia e a Síria.

Figura 15 – Lista de sentenças dum corpus

Fonte: O autor



Lista de Palavras - **Oriente Médio Pequeno**

**Palavras não reconhecidas**

Afrin	Ahrar	Alepo	Aleppo	Ansari	Assad	Bab	Bashar	Corps	EI	EL	Facebook	Hassan
Hawa	Hazer	Idlib	jihad	jihadistas	Kazim	Kenyo	Kerem	Killis	Kinik	Marea	Mercy	Nusra
Nyirjesy	Ôncüpinar	Qaeda	Recomanar	Reyhani	Rifat	salafista	Salam	Salih	Sham			

Palavra Composta	Lema	Gramática & Semântica	Inflexão
porta-voz	porta-voz	N VN	n s t s
queixaram-se	queixar	V PRO	s s p

Figura 16 – Lista de palavras dum corpus

Fonte: O autor

*Tagger*, cuja visualização é bastante similar à lista de sentenças, mas com pares palavra-etiqueta, como pode ser observado na figura 19.

### 4.2.3 CONCORDANCIADOR

Para resolver os problemas de compreensão por parte do usuário a respeito da sintaxe do buscador e do significado termo de busca digitado, levantados na seção 3.7, foi adicionado na

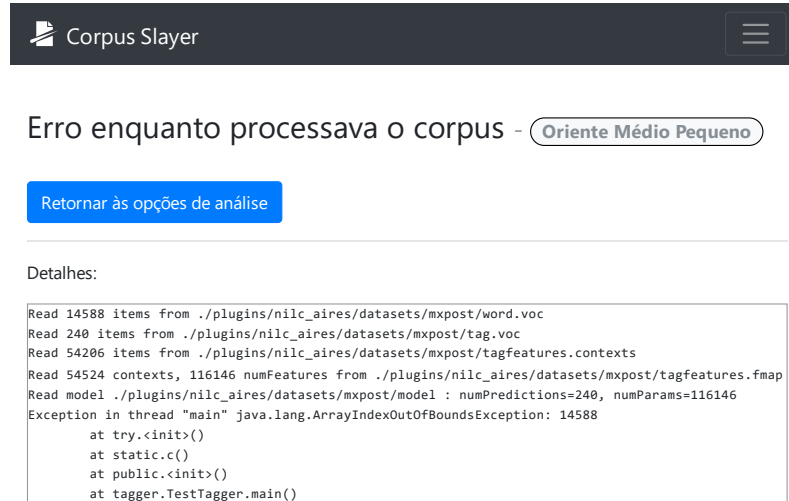


Figura 17 – Erro do *MXPOST* ao etiquetar para o Português Brasileiro utilizado arquivos treinados por Aires (2000)

Fonte: O autor

```
$ ./tagger
YOU MUST RUN THIS PROGRAM IN THE SAME DIRECTORY AS ./start-state-tagger and ./final-state-tagger
AND ./start-state-tagger and ./final-state-tagger MUST HAVE EXECUTE PERMISSION SET
$ █
```

Figura 18 – Erro do *Brill Tagger* requisitando arquivos com permissão de execução

Fonte: O autor



Figura 19 – Corpus processado pelo *Tree Tagger*

Fonte: O autor

tela de pesquisa do concordanciador desenvolvido (figura 20) uma tabela com a referência das construções aceitas e uma previsualização em tempo real de como o termo será entendido pelo buscador que está no servidor. Já nos resultados (figura 21), o destaque se dá tanto pelo par palavra-etiqueta em vermelho se destacando a cinza (pessoas com alguns tipos de deficiência visual congênita incapazes de distinguir certas cores (daltonismo) podem não ver esta diferença)

e pela cor de destaque da etiqueta com fundo colorido e letra branca se diferenciando da com fundo branco e letra preta. Também é observado um botão com uma seta apontando “para trás” na figura 21, bem como na 19, 16 e 15, onde tal seta, quando clicada, busca no corpus pelo documento que contém o que estava sendo exibido na tela, retornando o documento diretamente em caso de resultado único (figura 22) ou uma lista deles em caso haja múltiplos resultados (figura 23).

Corpus etiquetado

TreeTagger

Este campo é obrigatório.

Visibilidade da vizinhança

4 palavras de cada lado

Busca

..ado\_\_VERBO {0,1} terrorista {0,\*} .mic..

Este campo é obrigatório.

Buscar

Decompondo consulta

termina com ado VERBO

pula de 0 até 1 Palavras

é terrorista

pula de 0 até any Palavras

contém mic

Referência da notação de busca

Observe a tabela abaixo:

Busca	Significado
abc	A palavra é "abc"
\\	A palavra é "\\"
\\.\\.\\.	A palavra é "..."

Figura 20 – Tela de busca do concordanciador

Fonte: O autor

### 4.3 CÓDIGO FONTE

O código fonte da aplicação desenvolvida neste trabalho está publicamente disponível tanto em <<https://github.com/adlerosn/corpuslayer>>, quanto em <<https://git.adlerosn.com/root/corpuslayer>>; já o do etiquetador está publicamente disponível tanto em <<https://github.com/adlerosn/yas-pos-tagger>>, quanto em <<https://git.adlerosn.com/root/yas-pos-tagger>>.

## Resultados do Concordanciador - Oriente Médio Grande

1. apenas PDEN atua V conforme PREP o ART planejado N na PREP  
propaganda N terrorista ADJ gravada PCP com PREP duas NUM câmeras N

2. Estado N Islâmico NPROP é NPROP já NPROP considerado PCP o ART  
grupo N terrorista ADJ mais KC perigoso ADJ de PREP todos PROADJ

3. do ADV executivo ADJ turco N e KC considerado PCP um ART  
grupo N terrorista ADJ por PREP Ancara NPROP , , pelos N

Figura 21 – Tela de resultados do concordanciador

Fonte: O autor

Corpus Slayer

Documento #88 de Oriente Médio Grande

Território reivindicado pelo Estado Islâmico no mundo.  
[https://pt.m.wikipedia.org/wiki/Estado\\_Isl%C3%A2mico\\_do\\_Iraque\\_e\\_do\\_Levante](https://pt.m.wikipedia.org/wiki/Estado_Isl%C3%A2mico_do_Iraque_e_do_Levante)

Território reivindicado pelo Estado Islâmico no mundo.

Desde 2004, a principal meta do grupo é a fundação de um Estado islâmico. [117] [118] O FIII procurou estabelecer-se como um califado, um tipo de Estado islâmico liderado por um

Figura 22 – Visualização de documento a partir dum clique na “seta para trás”

Fonte: O autor

Corpus Slayer

Documentos encontrados em Oriente Médio Grande

08/08/2014 20h12 - Atualizado em 12/08/2014 18h27  
<http://g1.globo.com/mundo/noticia/2014/08/entenda-o-conflito-envolvendo-o-estado-islamico-na-siria-e-no-iraque.html>

14 pontos-chave sobre o Oriente Médio e o papel do Estado Islâmico  
[http://brasil.elpais.com/brasil/2015/10/11/internacional/1444563614\\_586697.html](http://brasil.elpais.com/brasil/2015/10/11/internacional/1444563614_586697.html)

Figura 23 – Listagem de documentos a partir dum clique na “seta para trás”

Fonte: O autor

## 5 CONCLUSÃO

Neste trabalho, iniciado com a observação de que havia espaço dentro dos *softwares* para estudo da língua, sucedido ~~com a~~ comparação de características e a listagem de requisitos ~~duma~~ ferramenta desejável, culminando com a implementação e implantação desta. Para tanto, foi necessário compatibilizar a entrada e saída de diversos sistemas e ferramentas com características diferentes, com e sem documentação adequada, tanto com formatos ~~cujo~~ *parser* foi mais fácil de escrever que outros.

Quanto ao aspecto documentação, etiquetadores como *MXPOST* e *Tree Tagger* não precisaram de nenhum documento além do breve e direto “leia-me” incluído em seus respectivos pacotes de distribuição. Já o *Unitex/GramLab*, além do manual do usuário para compreender o conteúdo nada trivial dos arquivos gerados pelo programa, foi necessário usar o programa para, a partir do *log* de comandos disparados ao terminal, inferir quais os comandos produzem quais arquivos e seus efeitos colaterais, ~~onde~~ em um momento, o código fonte da interface *desktop* teve de ser consultado; além do já discorrido sobre a documentação das etiquetas utilizadas na seção 3.6. Isso demonstra que o *Unitex/GramLab* ainda não possui uma documentação madura, que cubra o *software* como um todo de forma que um programador seja capaz de estender suas funcionalidades, não apenas ~~o manuseio de~~ sua interface. Independentemente do nível de maturidade da ferramenta, neste trabalho, elas foram integradas de forma transparente ao usuário que pode escolher a fonte de informação desejada para a tarefa desejada, ocultando detalhes de implementação intrínsecos a cada ferramenta (utilizando as configurações padrão) e trazendo o resultado pronto e relevante para o usuário (o linguista pesquisador); sem ocultar o fluxo de informações: os *softwares* envolvidos naquele fluxo e quais bases de dados estão sendo utilizadas em conjunto com o *software*, está explícito ~~duma~~ forma que é esperado ser possível, a partir da ferramenta desenvolvida, produzir conhecimento científico sem que seja necessário pagar pelo uso das ferramentas.

Outra contribuição ~~é sobre o~~ problema relatado na seção (3.7) que aborda o concordanciador, ~~onde~~ os softwares que realizam este papel de concordanciador (tabela 4 da seção 3.1) assumem que o usuário conhece e domina perfeitamente a sintaxe que apenas aquele software utiliza, já leu o manual e praticou; o que vai de encontro às premissas de usabilidade defendidas por Krug (2008). Com uma tabela de exemplos que abrangem todas as construções da sintaxe, a consulta ao manual foi substituída por uma rolagem na mesma página, e com a visualização gráfica em tempo real da consulta digitada, é esperado que o usuário iniciante sinta-se mais seguro e confortável quanto às buscas realizadas. Portanto, este trabalho também contribuiu para ~~com~~ a usabilidade dos concordanciadores.

## 5.1 TRABALHOS FUTUROS

- Embora a interoperabilidade de programas dentro da aplicação desenvolvida tenha sido uma preocupação durante o desenvolvimento, a interoperabilidade com aplicações externas acabou sendo negligenciada por conta do cronograma. É trabalho futuro tornar as funcionalidades acessíveis através de outras aplicações (sugestão: API RESTful<sup>1</sup> com dados em JSON<sup>2</sup>);
- Embora este trabalho tenha levado em conta a acessibilidade de daltônicos na escolha de cores, a aplicação não foi projetada para cegos que usam leitores de tela. Portanto, tornar a aplicação acessível para tais pessoas é trabalho futuro;
- Embora mencionado na introdução (seção 1.1) uma possível aplicação para o ensino da língua, são trabalhos futuros realizar um estudo de caso do uso da aplicação desenvolvida enquanto ferramenta de apoio no processo de ensino-aprendizagem num curso de letras, num curso de língua estrangeira, num curso de treinamento profissional e/ou numa turma de ensino médio;
- Embora atributos de corpora arbitrariamente fornecidos pelo usuário estejam implementados no modelo de dados (figura 24 da seção 3.2), eles não foram implementados para o usuário. É trabalho futuro implementar tais atributos, de preferência movendo o gerenciador de corpora para um *plug-in* à parte;
- Embora considerado desejável para grupos de pesquisadores atuando sobre um único objeto que pudessem compartilhar corpora e resultados uns com os outros, o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Embora um painel de controle administrativo seja considerado relevante para administradores dos sistemas de tecnologia da informação de universidades ou empresas que venham a instalar a aplicação, o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Embora a aplicação tenha sido idealizada para usar apenas softwares gratuitos de código aberto ou livre, o *MXPOST* e o *Tree Tagger* possuem licenças restritivas quanto a quais usos podem ser feitos dos etiquetadores em questão. É trabalho futuro substituir tais etiquetadores a fim de remover restrições quanto aos usos permitidos da plataforma;

<sup>1</sup> REST é a sigla para Transferência de Estado Representacional (do inglês, *Representational State Transfer*), conceito abordado na tese de Fielding (2000), a qual descreve padrões arquiteturais para a transferência do estado da aplicação sobre o protocolo HTTP (que não possui estado).

<sup>2</sup> JSON é sigla para Notação de Objetos JavaScript (do inglês, *JavaScript Object Notation*), uma notação para representação de dados como texto.

- Pela falta de documentação no *Unitex/GramLab*, é trabalho futuro inferir o significado das etiquetas por ele fornecido e documentá-las adequadamente;
- Embora considerado desejável extrair automaticamente palavras-chave a partir do texto fornecido, o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Embora considerado desejável montar corpora automaticamente a partir de buscas na internet, o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Embora considerado desejável obter a frequência dos n-gramas do texto, o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Embora considerado desejável obter as estatísticas do texto (itens da lista de requisitos em 3.1 que iniciam com “Calcular estatísticas”), o requisito foi abandonado para esta primeira versão quando colocado no cronograma. É trabalho futuro implementá-lo;
- Uma das pessoas ~~que~~ o autor conversou com numa rede social durante os estágios iniciais da execução deste trabalho dizia possuir dislexia (que o DSM-5 (American Psychiatric Association, 2013) cataloga como “315.00 (F81.0)” sob o nome “desordem específica de aprendizado com prejuízo na leitura” (p. 67)) e digitava com muitas palavras com grafia incorreta e com pontuação em locais arbitrários (que o DSM-5 (American Psychiatric Association, 2013) cataloga como “315.2 (F81.81)” sob o nome “desordem específica de aprendizado com prejuízo na escrita” (p. 67)), dificultando a compreensão da mensagem; usando o recurso de digitação por voz do teclado do telefone móvel dele, a grafia era correta mas a pontuação era inexistente. Para amenizar a dificuldade de interação social em meios *online* por pessoas ~~como as narradas anteriormente~~, é trabalho futuro implementar um módulo de correção ortográfica automática baseado em fonemas e um módulo de pontuação automática de textos não pontuados.

## REFERÊNCIAS

- AARTS, J.; GRANGER, S. Tag sequences in learner corpora: A key to interlanguage grammar and discourse. *Learner English on computer*, London: Longman, v. 132, p. 141, 1998.
- AIRES, R. V. X. *Implementação, adaptação, combinação e avaliação de etiquetadores para o português do Brasil*. 154 p. Tese (Doutorado) — Universidade de São Paulo, São Carlos, 2000. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28042016-090039/pt-br.php>>.
- ALUÍSIO, S. M.; ALMEIDA, G. M. de B. O que é e como se constrói um corpus? lições aprendidas na compilação de vários corpora para pesquisa linguística. *Calidoscópico*, v. 4, n. 3, p. 156–178, 2006.
- American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5®)*. [S.l.]: American Psychiatric Publishing, 2013. ISBN 9780890425572.
- AMORA, A. *Minidicionário Soares Amora da língua portuguesa*. 19<sup>a</sup>. ed. São Paulo: Saraiva, 2009. 818 p. ISBN 978-85-02-07980-9.
- ANTHONY, L. A critical look at software tools in corpus linguistics. *Linguistic Research*, v. 30, n. 2, p. 141–161, 2013.
- ANTHONY, L. *AntCorGen*. Tokyo: [s.n.], 2017. 1–4 p. Disponível em: <<http://www.laurenceanthony.net/software/antcorgen/releases/AntCorGen101/help.pdf>>.
- ARANHA, C.; PASSOS, E. A tecnologia de mineração de textos. *Revista Eletrônica de Sistemas de Informação ISSN 1677-3071 doi: 10.21529/RESI*, v. 5, n. 2, 2006.
- BEITZEL, S. M. On understanding and classifying web queries. *Evaluation*, n. May, p. 96, 2006.
- BIBER, D. *Dimensions of register variation: A cross-linguistic comparison*. [S.l.]: Cambridge University Press, 1995.
- BICK, E. The parsing system palavras. *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*, University of Aarhus, 2000.
- BRASIL. Princípios, garantias, direitos e deveres para o uso da internet no brasil: Lei n. 12.965, de 23 de abr. de 2014. Brasília, DF, abr 2014. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2014/lei/l12965.htm](http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm)>. Acesso em: 4 dez. 2017.
- BRUCKSCHEN, M. *RECONHECIMENTO DE ENTIDADES NOMEADAS E RELAÇÕES NO DOMÍNIO, DE PRIVACIDADE, E RESPONSABILIZAÇÃO*. Tese (Doutorado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2010.
- CAMBRIA, E.; WHITE, B. Jumping nlp curves: a review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine*, IEEE, v. 9, n. 2, p. 48–57, 2014.
- CCE. *Manual de instruções*. 2013. Disponível em: <<https://www.extra-imagens.com.br/Control/ArquivoExibir.aspx?IdArquivo=11749644>>. Acesso em: 30 mar. 2018.



CETIC.BR. Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras - tic educação 2015. 2016. Disponível em: <[http://nic.br/media/docs/publicacoes/2/TIC\\_Edu\\_2015\\_LIVRO\\_ELETRONICO.pdf](http://nic.br/media/docs/publicacoes/2/TIC_Edu_2015_LIVRO_ELETRONICO.pdf)>. Acesso em: 22 nov. 2016.

CHAVES, H.; MELLO, H. Sistema identificador de sintagmas verbais do pb. In: *Anais do Congresso Nacional de Estudos Linguísticos-CONEL*. [S.l.: s.n.], 2014. v. 1, n. 2.

COLLINS Escolar Plus Dictionary. 2<sup>a</sup>. ed. [S.l.]: Cengage Learning, 2009. (Collins COBUILD Dictionaries of English Series). 862 p. ISBN 978-1-4240-7588-1.

DICIO. *Sentença*. 2018. Disponível em: <<https://www.dicio.com.br/sentenca/>>. Acesso em: 2 abr. 2018.

Django Software Foundation. *Django em um relance*. 2018. Disponível em: <<https://docs.djangoproject.com/pt-br/2.0/intro/overview/>>. Acesso em: 25 jan. 2018.

DUBOIS, J. et al. *Dicionário de lingüística*. [S.l.]: Cultrix, 1993.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. Trad. 6 ed. São Paulo: Pearson, 2012. ISBN 978-85-7936-085-5.

FERREIRA, A. B. de H. *Novo dicionário da língua portuguesa*. 1<sup>a</sup>. ed. Rio de Janeiro: Editora Nova Fronteira, 1975. 5<sup>a</sup> reimpressão, 1500 p.

FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. 162 p. Tese (Doutorado), 2000. Disponível em: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.

FREE SOFTWARE FOUNDATION. *GNU Lesser General Public License*. 1999. Disponível em: <<https://spdx.org/licenses/LGPL-2.1.html>>.

GAGNON, M. *Processamento da Linguagem Natural*. 2000.

GALISSON, R.; COSTE, D. *Dicionário de didáctica das línguas*. [S.l.: s.n.], 1983.

GANDY, D. *Font Awesome, the iconic font and CSS toolkit*. 2017. Disponível em: <<http://fontawesome.io/>>. Acesso em: 3 dez. 2017.

Google Cloud. *Camada gratuita do GCP: Avaliações gratuitas estendidas e produtos sempre gratuitos*. 2018. Disponível em: <<https://cloud.google.com/free/?hl=pt-br>>. Acesso em: 30 mar. 2018.

Google Cloud. *Compute Engine Documentation: Preços do google compute engine*. 2018. Disponível em: <<https://cloud.google.com/compute/pricing?hl=pt-br>>. Acesso em: 30 mar. 2018.

GRIES, S. T. Useful statistics for corpus linguistics. *A mosaic of corpus linguistics: Selected approaches*, Peter Lang Frankfurt, Germany, v. 66, p. 269–291, 2010.

HOUAISS, A. et al. *Dicionário Houaiss da língua portuguesa*. 1<sup>a</sup>. ed. Rio de Janeiro: Objetiva, 2001. 2925 p. ISBN 85-7302-383-X.

KRUG, S. *Não me Faça Pensar: uma abordagem de bom senso e usabilidade*. 2<sup>a</sup>. ed. Rio de Janeiro: Alta Books, 2008. 224 p. ISBN 0-321-34475-8.

KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: Uma abordagem top-down*. Trad. 6 ed. São Paulo: Pearson, 2013. ISBN 978-85-8143-677-7.

LINGUATECA. *Floresta Sinta(c)tica: breve descrição dos corpora*. 2009. Disponível em: <<http://www.linguateca.pt/Floresta/corpus.html>>. Acesso em: 8 jan. 2018.

LINGUATECA. *Glossário de etiquetas florestais*. 2010. Disponível em: <<http://www.linguateca.pt/Floresta/BibliaFlorestal/anexo1.html>>. Acesso em: 27 jan. 2018.

LINGUATECA. *Projecto Floresta Sinta(c)tica*. 2010. Disponível em: <<http://www.linguateca.pt/Floresta/principal.html>>. Acesso em: 8 jan. 2018.

LINGUATECA. *Levantamento integral da Floresta Sintáctica*. 2016. Disponível em: <<http://www.linguateca.pt/Floresta/levantamento.html>>. Acesso em: 8 jan. 2018.

MEUSEL, R. et al. The graph structure in the web – analyzed on different aggregation levels. *The Journal of Web Science*, v. 1, n. 1, p. 33–47, 2015. Disponível em: <<http://dx.doi.org/10.1561/106.00000003>>.

Microsoft. *Using Python Scripts with IIS*. 2017. Disponível em: <<https://support.microsoft.com/en-ie/help/276494/using-python-scripts-with-iis>>. Acesso em: 25 jan. 2018.

MUNIZ, M. C.; NUNES, M. D. G. V.; LAPORTE, E. UNITEX-PB, a set of flexible language resources for Brazilian Portuguese. In: *Workshop on Technology on Information and Human Language (TIL)*. São Leopoldo, Brazil: [s.n.], 2005. p. 2059–2068. Disponível em: <<https://halshs.archives-ouvertes.fr/halshs-00190857>>.

MUNIZ, M. C.; NUNES, M. D. G. V.; LAPORTE, E. UNITEX-PB, a set of flexible language resources for Brazilian Portuguese. In: *Workshop on Technology on Information and Human Language (TIL)*. São Leopoldo, Brazil: [s.n.], 2005. p. 2059–2068. Disponível em: <<https://halshs.archives-ouvertes.fr/halshs-00190857>>.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. *Linguisticae Investigationes*, John Benjamins publishing company, v. 30, n. 1, p. 3–26, 2007.

Núcleo Interinstitucional de Linguística Computacional. *NILCTaggers*. 2012. Disponível em: <<http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html>>. Acesso em: 20 jan. 2018.

OLIVEIRA, L. P. d. *Linguística de corpus: teoria, interfaces e aplicações*. 2009.

O'NEIL, E. J. Object/relational mapping 2008: Hibernate and the entity data model (edm). In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2008. (SIGMOD '08), p. 1351–1356. ISBN 978-1-60558-102-6. Disponível em: <<http://doi.acm.org/10.1145/1376616.1376773>>.

PAUMIER, S. *Unitex 3.1 user manual*. 2016.

POWERS, D. Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011. ISSN 2229-3981. Disponível em: <[http://www.bioinfopublication.org/files/articles/2\\_1\\_1\\_JMLT.pdf](http://www.bioinfopublication.org/files/articles/2_1_1_JMLT.pdf)>.

RIJSBERGEN, C. J. van. *Information Retrieval*. 2<sup>a</sup>. ed. London: Butterworth-Heinemann, 1979.

SARDINHA, T. B. Lingüística de corpus: histórico e problemática. *Delta*, SciELO Brasil, v. 16, n. 2, p. 323–367, 2000.

SILVA, T. J. d. *Indexação automática por meio da extração e seleção de sintagmas nominais em textos em língua portuguesa 2014*. Tese (Doutorado) — Dissertação (Mestrado)—Departamento de Ciência da Informação, Universidade Federal de Pernambuco—UFPE, Recife, 2014. 219 f.

SINCLAIR, J. Corpus and text-basic principles. *Developing linguistic corpora: A guide to good practice*, Oxbow Books Oxford, p. 1–16, 2005.

SOMMERVILLE, I. *Software Engineering*. [S.l.]: Pearson, 2011. (International Computer Science Series). ISBN 9780137053469.

TAGNIN, S. E. Glossário de linguística de corpus. *São Paulo: HUB Editorial*, 2010.

The Apache Software Foundation. *Dynamic Content with CGI*. 2018. Disponível em: <<https://httpd.apache.org/docs/2.4/howto/cgi.html>>. Acesso em: 25 jan. 2018.

The Linux Information Project. *Dumb Terminal Definition*. 2005. Disponível em: <[http://www.linfo.org/dumb\\_terminal.html](http://www.linfo.org/dumb_terminal.html)>. Acesso em: 23 jan. 2018.

The SQLite Consortium. *About SQLite*. 2018. Disponível em: <<https://www.sqlite.org/about.html>>. Acesso em: 26 jan. 2018.

THIELE, P. F. O. *Desambiguação de anotações morfossintáticas feitas por MTMDD*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2015.

Toshiba. *Detailed specs for Satellite 2250XCDS*. 2000. Disponível em: <<https://support.toshiba.com/support/staticContentDetail?contentId=638109>>. Acesso em: 30 mar. 2018.

TRASK, R. L. et al. *Dicionário de linguagem e lingüística*. [S.l.]: Editora Contexto, 2004.

UNIVERSIDADE DE LISBOA, DEPT. DE INFORMÁTICA, GRUPO DE FALA E LINGUAGEM NATURAL. *LX-Parser License*. 2010. Disponível em: <[http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LX-Parser\\_License.pdf](http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LX-Parser_License.pdf)>.

UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE. *Lesser General Public License For Linguistic Resources*. 2009. Disponível em: <<https://spdx.org/licenses/LGPLLR.html>>.

VIEIRA, R.; LIMA, V. L. Lingüística computacional: princípios e aplicações. In: SN. *Anais do XXI Congresso da SBC. I Jornada de Atualização em Inteligência Artificial*. [S.l.], 2001. v. 3, p. 47–86.

## **Apêndices**

## APÊNDICE A – O MODELO LÓGICO COMPLETO DO GERENCIADOR DE CORPORA

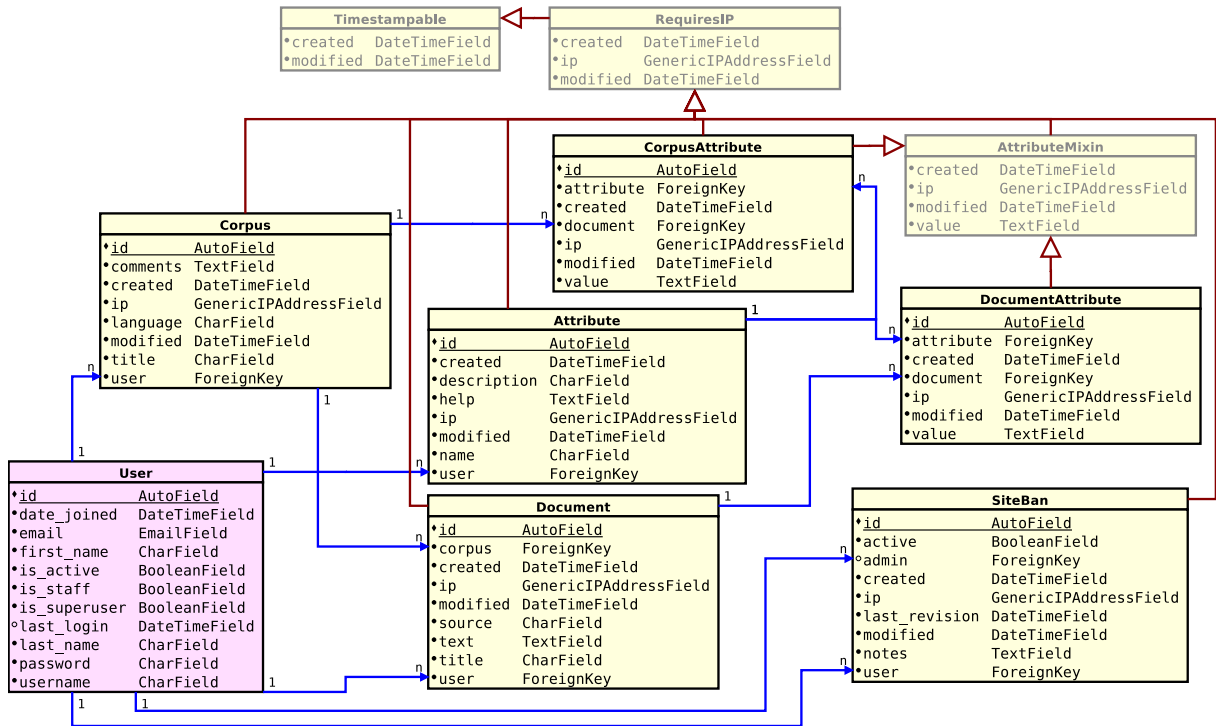


Figura 24 – Modelo lógico gerado automaticamente a partir da implementação, onde

Fonte: O autor

## APÊNDICE B – FLUXOS DE EXECUÇÃO DO SUBSISTEMA DE *PLUG-INS*

### B.1 SEQUÊNCIA DE INICIALIZAÇÃO

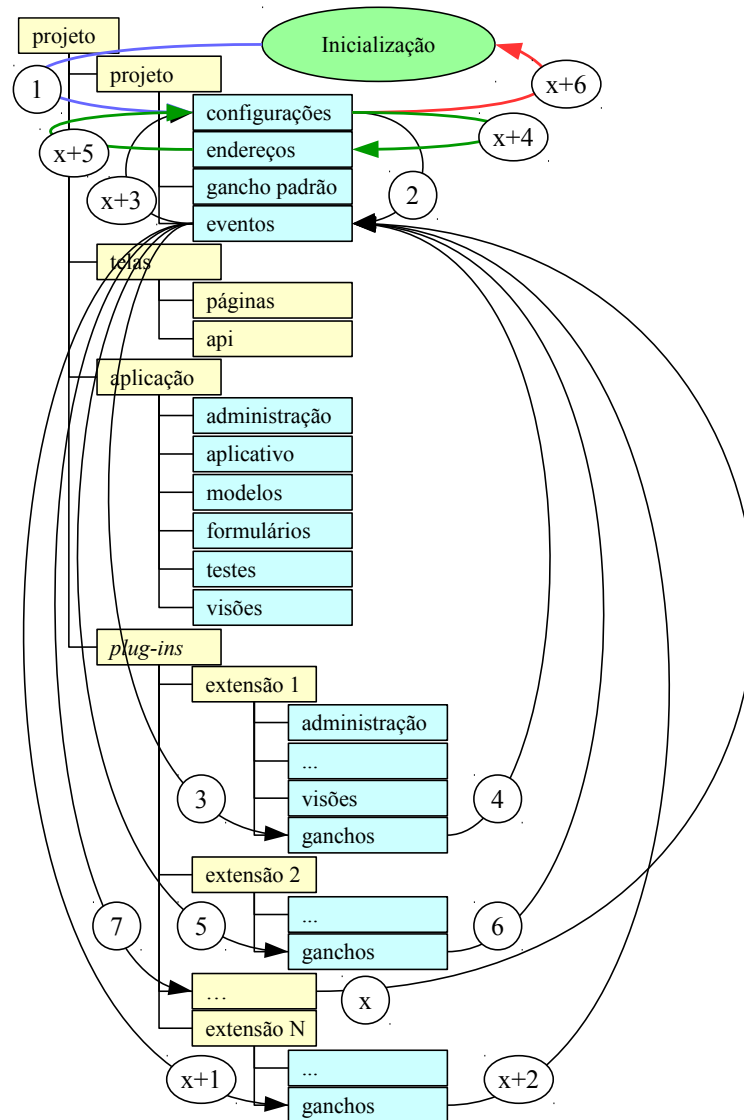


Figura 25 – Esta figura traz a sequência de inicialização do subsistema de eventos. O primeiro arquivo que o *framework* Django carrega é o de configurações (seta “1”), que foi utilizado para inicializar o subsistema de eventos (seta “2”), que, por sua vez, carrega os “ganchos” (setas “3” e “4”) do primeiro “plug-in” e todos os demais (setas “5”, “6”, “7”, “x”, “x+1” e “x+2”), termina sua configuração interna, retorna a execução ao arquivo de configurações (seta “x+3”), que atualiza os endereços disponíveis (setas “x+4” e “x+5”) e retorna à execução normal do *framework* (seta “x+6”).

Fonte: O autor

## B.2 REQUISIÇÃO SEM USO DE *PLUG-INS* PARA EXIBIR FORMULÁRIO

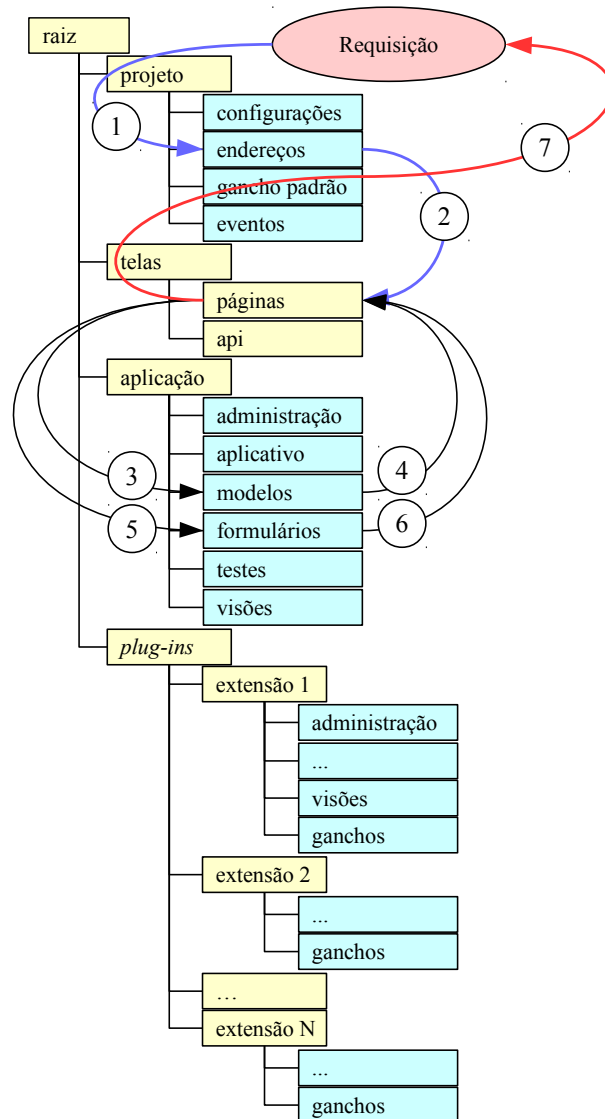


Figura 26 – Esta figura traz como exemplo o cenário de exibir um formulário para editar um documento pertencente a um corpus. O *framework* busca o endereço especificado pela requisição (seta “1”) e encontra uma entrada para uma classe de controlador no arquivo de “visões” presente no pacote “páginas” (seta “2”), que, por sua vez, carrega o dado (setas “3” e “4”), o formulário (seta “5” e “6”), monta como resposta um documento HTML usando um arquivo de *template* (omitido do modelo) e retorna a resposta (seta “7”).

Fonte: O autor

### B.3 REQUISIÇÃO PARA EXIBIR LISTA DE *PLUG-INS* NA PÁGINA “ANÁLISE”

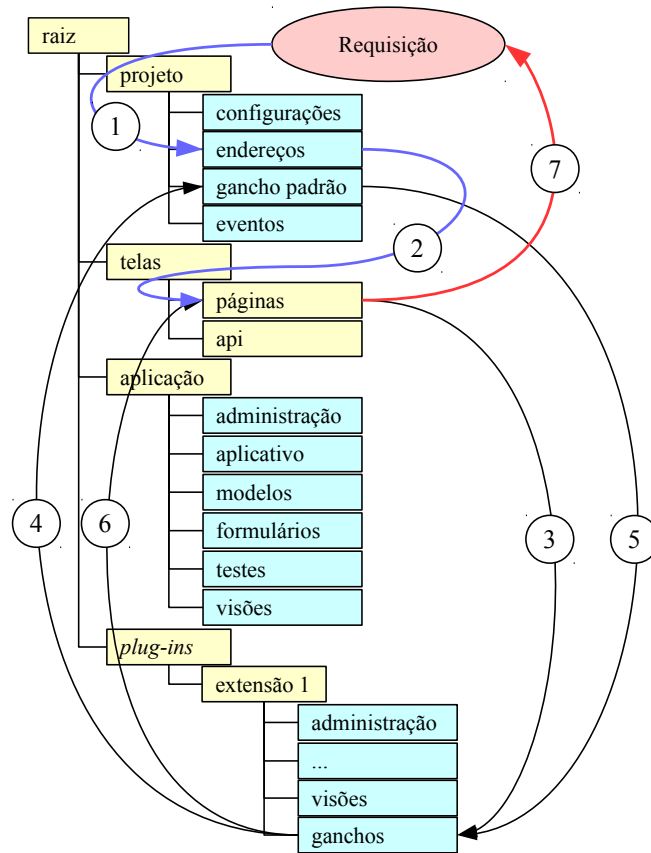


Figura 27 – Esta figura traz como exemplo uma requisição para listar os *plug-ins* na página “Análise”. O *framework* busca o endereço especificado pela requisição (seta “1”) e encontra a entrada para a classe controladora da página “Análise” no pacote “páginas” (seta “2”), onde tal controlador dispara consultas para os “ganchos” de cada *plug-in* (seta “3”) e cada “gancho” carrega o gerador de “ganchos” do *script* “gancho padrão” (setas “4” e “5”) e os altera conforme suas necessidades, o retorna ao controlador (seta “6”), que gera a página a partir de um *template* e a retorna (seta “7”).

Fonte: O autor



#### B.4 REQUISIÇÃO PARA EXIBIR PÁGINA GERADA POR *PLUG-IN* QUE USA A APLICAÇÃO

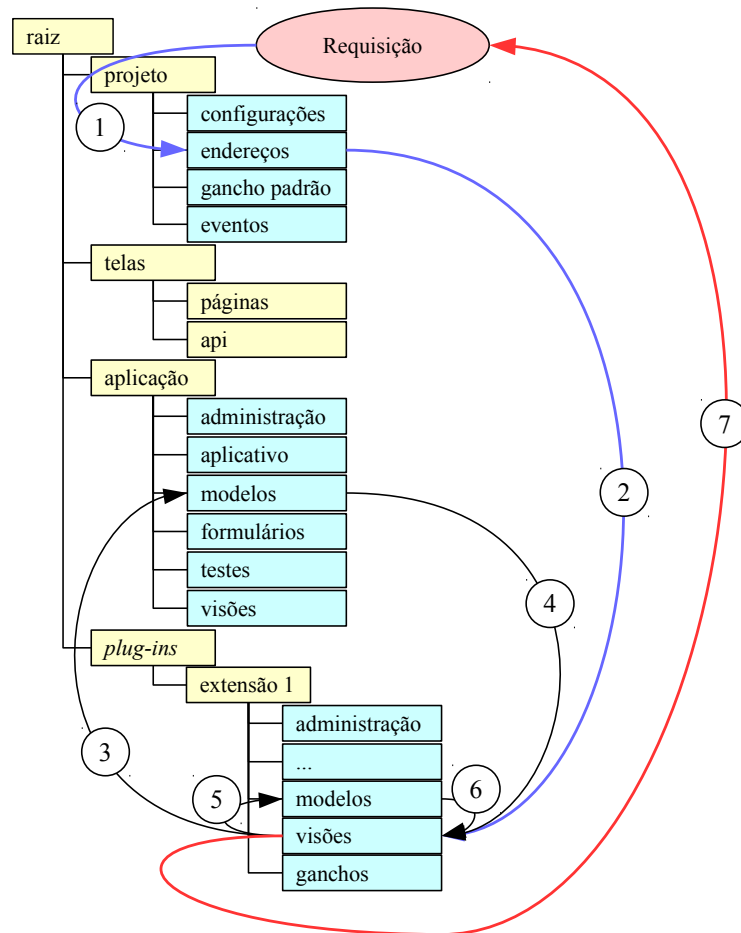


Figura 28 – Esta figura traz como exemplo uma requisição para buscar uma lista de sentenças associada a um corpus. As setas “1” e “2” seguem o mesmo algoritmo dos parágrafo anterior, mas é executada uma classe controladora no arquivo de visões do *plug-in*, que **por sua vez** carrega o corpus a partir dos modelos da aplicação (setas “3” e “4”), que, por sua vez, é utilizada para carregar a lista de sentenças em seu modelo interno para, enfim, gerar o documento de resposta e retorná-lo ao usuário (seta “7”).

Fonte: O autor

## B.5 REQUISIÇÃO PARA EXIBIR PÁGINA GERADA POR *PLUG-IN* QUE USA OUTROS *PLUG-INS*

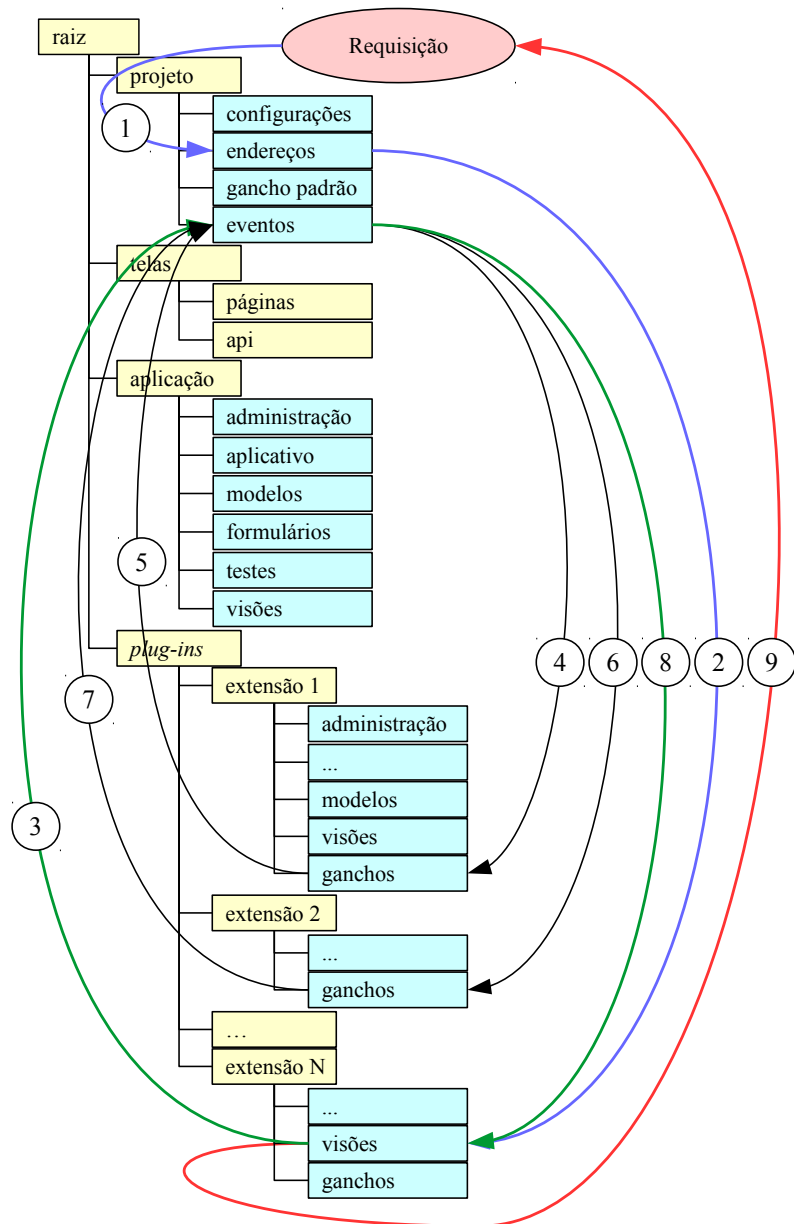


Figura 29 – Esta figura traz como exemplo uma requisição para buscar uma concordância, que depende de uma lista de palavras etiquetada fornecida por outros *plug-ins* através de eventos. Novamente, as setas “1” e “2” seguem o mesmo algoritmo dos parágrafo anterior, que, desta vez, termina por executar o controlador presente no arquivo de visões do concordanciador, que dispara um evento de busca (seta “3”), que acaba por executar código definido durante a inicialização presente nos arquivos da “extensão 1” (setas “4” e “5”) e da “extensão 2” (setas “6” e “7”), retornando ao controlador uma lista do corpus etiquetado por diferentes etiquetadores (seta “8”), que processa os dados, gera um documento e o retorna ao usuário (seta “9”).

Fonte: O autor

## APÊNDICE C – HARDWARE UTILIZADO DURANTE O TREINO E TESTE DOS ETIQUETADORES

- Sistema operacional Ubuntu Studio 16.04 LTS;
- Processador Intel Core i7 4790 (8 *threads* em 4 núcleos a 4GHz);
- 16GB de RAM DDR3 a 1600MHz;
- Partição raiz do sistema operacional (/) de 60GB num SSD SanDisk SATA III de 128GB;
- Partição de usuário (/home) de 1,5TB num Western Digital Green 5400RPM SATA III de 2TB;
- Partição de *swap* de 100GB num Western Digital Blue 7200RPM SATA III de 1TB.

**APÊNDICE D – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTICAS DO  
ETIQUETADOR UNITEX/GRAMLAB**

	VerdadeiroPositivo	VerdadeiroNegativo	FalsoPositivo	FalsoNegativo
???	0	2686948	114285	0
ADJ	133080	2407797	215249	45107
ADV	144557	2512313	107085	37278
ART	0	2789672	0	11561
CONJ	84805	2631209	29076	56143
DET	0	2647994	153239	0
EC	0	2801051	0	182
INTERJ	363	2798958	637	1275
N	548603	1689793	377382	185455
NUM	0	2780150	0	21083
PFX	0	2766566	34667	0
PREP	311423	2360014	1610	128186
PRON	98171	2183604	7501	511957
SPEC	0	2783960	0	17273
V	380987	2277989	58513	83744

Tabela 8 – Tabela de contingência por etiqueta do *Unitex/Gramlab*

Fonte: O autor

	Acurácia	F <sub>1</sub> score	Taxa de erro	Precisão	Prevalência	Revocação	Especificidade
???	95,92%			0,00%	0,00%		95,92%
ADJ	90,71%	50,55%	25,31%	38,21%	6,36%	74,69%	91,79%
ADV	94,85%	66,70%	20,50%	57,45%	6,49%	79,50%	95,91%
ART	99,59%		100,00%		0,41%	0,00%	100,00%
CONJ	96,96%	66,56%	39,83%	74,47%	5,03%	60,17%	98,91%
DET	94,53%			0,00%	0,00%		94,53%
EC	99,99%		100,00%		0,01%	0,00%	100,00%
INTERJ	99,93%	27,52%	77,84%	36,30%	0,06%	22,16%	99,98%
N	79,91%	66,10%	25,26%	59,25%	26,20%	74,74%	81,74%
NUM	99,25%		100,00%		0,75%	0,00%	100,00%
PFX	98,76%			0,00%	0,00%		98,76%
PREP	95,37%	82,75%	29,16%	99,49%	15,69%	70,84%	99,93%
PRON	81,46%	27,43%	83,91%	92,90%	21,78%	16,09%	99,66%
SPEC	99,38%		100,00%		0,62%	0,00%	100,00%
V	94,92%	84,27%	18,02%	86,69%	16,59%	81,98%	97,50%
~GERAL~	94,77%	60,76%	39,24%	60,76%	6,67%	60,76%	97,20%

Tabela 9 – Estatísticas por etiqueta do *Unitex/Gramlab*

Fonte: O autor

**APÊNDICE E – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTICAS DO  
ETIQUETADOR YAS-TAGGER**

	VerdadeiroPositivo	VerdadeiroNegativo	FalsoPositivo	FalsoNegativo
00	0	4571236	1	1
07	0	4571231	7	0
08	0	4571237	1	0
1	0	4571237	1	0
11	0	4571237	1	0
184	0	4571235	3	0
1906	0	4571234	4	0
1966	0	4571228	10	0
1969	0	4571236	2	0
1979	0	4571234	4	0
2	0	4571233	5	0
2006	0	4571235	2	1
3	0	4571237	1	0
9	0	4571237	0	1
91	0	4571237	0	1
93	0	4571237	0	1
95	0	4571237	0	1
???	0	3925119	646119	0
ADJ	161698	4266181	18538	124821
ADV	202535	4284369	14947	69387
ALT	0	4571234	0	4
ART	5655	4422383	130248	12952
CONJ	174792	4331987	21164	43295
EC	91	4570858	50	239
Elvis	0	4571237	1	0
IND	0	4571237	0	1
INTJ	468	4568713	239	1818
N	835036	3497956	77027	161219
NP	0	4571235	3	0
NUM	58851	4487778	3037	21572
PP	0	4570839	0	399
PRON	777726	3525289	74278	193945
PROP	65210	4324331	8166	173531
PRP	715887	3786512	29800	39039
PU	0	4571228	10	0
Papel	0	4571236	2	0
SPEC	0	4544058	0	27180
V	517169	3844184	28937	180948
VAUX	1516	4563479	1995	4248
http	0	4571237	1	0

Tabela 10 – Tabela de contingência por etiqueta do *YAS-Tagger*

Fonte: O autor

	Acurácia	F <sub>1</sub> score	Taxa de erro	Precisão	Prevalência	Revocação	Especificidade
00	100.00%		100.00%	0.00%	0.00%	0.00%	100.00%
07	100.00%			0.00%	0.00%		100.00%
08	100.00%			0.00%	0.00%		100.00%
1	100.00%			0.00%	0.00%		100.00%
11	100.00%			0.00%	0.00%		100.00%
184	100.00%			0.00%	0.00%		100.00%
1906	100.00%			0.00%	0.00%		100.00%
1966	100.00%			0.00%	0.00%		100.00%
1969	100.00%			0.00%	0.00%		100.00%
1979	100.00%			0.00%	0.00%		100.00%
2	100.00%			0.00%	0.00%		100.00%
2006	100.00%		100.00%	0.00%	0.00%	0.00%	100.00%
3	100.00%			0.00%	0.00%		100.00%
9	100.00%		100.00%		0.00%	0.00%	100.00%
91	100.00%		100.00%		0.00%	0.00%	100.00%
93	100.00%		100.00%		0.00%	0.00%	100.00%
95	100.00%		100.00%		0.00%	0.00%	100.00%
???	85.87%			0.00%	0.00%		85.87%
ADJ	96.86%	69.29%	43.56%	89.71%	6.27%	56.44%	99.57%
ADV	98.16%	82.77%	25.52%	93.13%	5.95%	74.48%	99.65%
ALT	100.00%		100.00%		0.00%	0.00%	100.00%
ART	96.87%	7.32%	69.61%	4.16%	0.41%	30.39%	97.14%
CONJ	98.59%	84.43%	19.85%	89.20%	4.77%	80.15%	99.51%
EC	99.99%	38.64%	72.42%	64.54%	0.01%	27.58%	100.00%
Elvis	100.00%			0.00%	0.00%		100.00%
IND	100.00%		100.00%		0.00%	0.00%	100.00%
INTJ	99.96%	31.27%	79.53%	66.20%	0.05%	20.47%	99.99%
N	94.79%	87.52%	16.18%	91.55%	21.79%	83.82%	97.85%
NP	100.00%			0.00%	0.00%		100.00%
NUM	99.46%	82.71%	26.82%	95.09%	1.76%	73.18%	99.93%
PP	99.99%		100.00%		0.01%	0.00%	100.00%
PRON	94.13%	85.29%	19.96%	91.28%	21.26%	80.04%	97.94%
PROP	96.03%	41.79%	72.69%	88.87%	5.22%	27.31%	99.81%
PRP	98.49%	95.41%	5.17%	96.00%	16.51%	94.83%	99.22%
PU	100.00%			0.00%	0.00%		100.00%
Papel	100.00%			0.00%	0.00%		100.00%
SPEC	99.41%		100.00%		0.59%	0.00%	100.00%
V	95.41%	83.13%	25.92%	94.70%	15.27%	74.08%	99.25%
VAUX	99.86%	32.69%	73.70%	43.18%	0.13%	26.30%	99.96%
http	100.00%			0.00%	0.00%		100.00%
~GERAL~	98,85%	76,93%	23,07%	76,93%	2,50%	76,93%	99,41%

Tabela 11 – Estatísticas por etiqueta do YAS-Tagger

Fonte: O autor

**APÊNDICE F – TABELA DE CONTINGÊNCIA E SUAS ESTATÍSTICAS DO  
ETIQUETADOR *YAS-TAGGER* SOBRE CORPUS DE AIRES**

	VerdadeiroPositivo	VerdadeiroNegativo	FalsoPositivo	FalsoNegativo
	0	55805	3	0
!	14	55717	0	77
"	330	55364	11	103
'	1	55801	0	6
(	50	55660	0	98
)	125	55663	0	20
,	3488	51566	54	700
-	208	55505	4	91
.	815	53411	0	1582
...	43	55724	1	40
:	0	55806	2	0
;	78	55660	0	70
?	7	55716	0	85
???	0	36977	18831	0
ADJ	1337	51428	388	2655
ADV	1076	53599	165	968
ADV+PPR	0	55807	1	0
ART	3398	51220	143	1047
AUX	0	55806	0	2
CONJCOORD	1069	53844	218	677
CONJSUB	350	54988	151	319
I	10	55784	4	10
INT	0	55807	0	1
LADV	202	55228	116	262
LCONJ	144	55426	113	125
LDEN	66	55678	47	17
LP	61	55656	42	49
LPREP	261	55127	177	243
N	8278	42890	1153	3487
NC	465	54935	18	390
NO	5	55754	9	40
NP	1193	52752	281	1582
ORD	47	55713	16	32
PAPASS	2	55774	10	22
PD	112	55508	10	178
PDEN	1	55805	1	1
PIND	99	55556	30	123
PINT	1	55769	14	24
PPOA	187	55400	53	168
PPOT	9	55787	2	10
PPR	215	55500	10	83
PPS	290	55381	12	125
PR	412	54865	260	271
PREAL	0	55793	0	15
PREP	2915	47212	264	5417
PREP+ADV	0	55804	4	0
PREP+ART	0	53751	2057	0
PREP+N	0	55806	2	0
PREP+PD	0	55755	53	0
PREP+PPOT	0	55807	1	0
PREP+PPR	0	55794	14	0
PREP+PREP	0	55807	1	0
PTRA	14	55778	8	8
RES	42	55643	49	74
VAUX	237	54955	74	542
VAUX!PPOA	0	55807	0	1
VAUX+PPOA	0	55801	7	0
VBI	43	55366	81	318
VBI+PPOA	0	55797	11	0
VINT	128	54832	171	677
VINT+PPOA	0	55803	5	0
VLIG	520	54787	127	374
VLIG+PPOA	0	55807	1	0
VTD	1235	51875	453	2245
VTD!PPOA	0	55807	0	1
VTD+PPOA	0	55771	37	0
VTI	208	54809	237	554
VTI+PPOA	0	55807	1	0
[	4	55804	0	0
]	4	55803	1	0

Tabela 12 – Tabela de contingência por etiqueta do *YAS-Tagger* sobre corpus de Aires

Fonte: O autor

	Acurácia	F <sub>1</sub> score	Taxa de erro	Precisão	Prevalência	Revocação	Especificidade
!	99,86%	26,67%	84,62%	100,00%	0,16%	15,38%	100,00%
"	99,80%	85,27%	23,79%	96,77%	0,78%	76,21%	99,98%
,	99,99%	25,00%	85,71%	100,00%	0,01%	14,29%	100,00%
(	99,82%	50,51%	66,22%	100,00%	0,27%	33,78%	100,00%
)	99,96%	92,59%	13,79%	100,00%	0,26%	86,21%	100,00%
,	98,65%	90,25%	16,71%	98,48%	7,50%	83,29%	99,90%
-	99,83%	81,41%	30,43%	98,11%	0,54%	69,57%	99,99%
.	97,17%	50,75%	66,00%	100,00%	4,30%	34,00%	100,00%
...	99,93%	67,72%	48,19%	97,73%	0,15%	51,81%	100,00%
:	100,00%			0,00%	0,00%		100,00%
;	99,87%	69,03%	47,30%	100,00%	0,27%	52,70%	100,00%
?	99,85%	14,14%	92,39%	100,00%	0,16%	7,61%	100,00%
???	66,26%			0,00%	0,00%		66,26%
ADJ	94,55%	46,77%	66,51%	77,51%	7,15%	33,49%	99,25%
ADV	97,97%	65,51%	47,36%	86,70%	3,66%	52,64%	99,69%
ADV+PPR	100,00%			0,00%	0,00%		100,00%
ART	97,87%	85,10%	23,55%	95,96%	7,96%	76,45%	99,72%
AUX	100,00%		100,00%		0,00%	0,00%	100,00%
CONJCOORD	98,40%	70,49%	38,77%	83,06%	3,13%	61,23%	99,60%
CONJSUB	99,16%	59,83%	47,68%	69,86%	1,20%	52,32%	99,73%
I	99,97%	58,82%	50,00%	71,43%	0,04%	50,00%	99,99%
INT	100,00%		100,00%		0,00%	0,00%	100,00%
LADV	99,32%	51,66%	56,47%	63,52%	0,83%	43,53%	99,79%
LCONJ	99,57%	54,75%	46,47%	56,03%	0,48%	53,53%	99,80%
LDEN	99,89%	67,35%	20,48%	58,41%	0,15%	79,52%	99,92%
LP	99,84%	57,28%	44,55%	59,22%	0,20%	55,45%	99,92%
LPREP	99,25%	55,41%	48,21%	59,59%	0,90%	51,79%	99,68%
N	91,69%	78,11%	29,64%	87,77%	21,08%	70,36%	97,38%
NC	99,27%	69,51%	45,61%	96,27%	1,53%	54,39%	99,97%
NO	99,91%	16,95%	88,89%	35,71%	0,08%	11,11%	99,98%
NP	96,66%	56,15%	57,01%	80,94%	4,97%	42,99%	99,47%
ORD	99,91%	66,20%	40,51%	74,60%	0,14%	59,49%	99,97%
PAPASS	99,94%	11,11%	91,67%	16,67%	0,04%	8,33%	99,98%
PD	99,66%	54,37%	61,38%	91,80%	0,52%	38,62%	99,98%
PDEN	100,00%	50,00%	50,00%	50,00%	0,00%	50,00%	100,00%
PIND	99,73%	56,41%	55,41%	76,74%	0,40%	44,59%	99,95%
PINT	99,93%	5,00%	96,00%	6,67%	0,04%	4,00%	99,97%
PPOA	99,60%	62,86%	47,32%	77,92%	0,64%	52,68%	99,90%
PPOT	99,98%	60,00%	52,63%	81,82%	0,03%	47,37%	100,00%
PPR	99,83%	82,22%	27,85%	95,56%	0,53%	72,15%	99,98%
PPS	99,75%	80,89%	30,12%	96,03%	0,74%	69,88%	99,98%
PR	99,05%	60,81%	39,68%	61,31%	1,22%	60,32%	99,53%
PREAL	99,97%		100,00%		0,03%	0,00%	100,00%
PREP	89,82%	50,65%	65,01%	91,70%	14,93%	34,99%	99,44%
PREP+ADV	99,99%			0,00%	0,00%		99,99%
PREP+ART	96,31%			0,00%	0,00%		96,31%
PREP+N	100,00%			0,00%	0,00%		100,00%
PREP+PD	99,91%			0,00%	0,00%		99,91%
PREP+PPOT	100,00%			0,00%	0,00%		100,00%
PREP+PPR	99,97%			0,00%	0,00%		99,97%
PREP+PREP	100,00%			0,00%	0,00%		100,00%
PTRA	99,97%	63,64%	36,36%	63,64%	0,04%	63,64%	99,99%
RES	99,78%	40,58%	63,79%	46,15%	0,21%	36,21%	99,91%
VAUX	98,90%	43,49%	69,58%	76,21%	1,40%	30,42%	99,87%
VAUX!PPOA	100,00%		100,00%		0,00%	0,00%	100,00%
VAUX+PPOA	99,99%			0,00%	0,00%		99,99%
VBI	99,29%	17,73%	88,09%	34,68%	0,65%	11,91%	99,85%
VBI+PPOA	99,98%			0,00%	0,00%		99,98%
VINT	98,48%	23,19%	84,10%	42,81%	1,44%	15,90%	99,69%
VINT+PPOA	99,99%			0,00%	0,00%		99,99%
VLIG	99,10%	67,49%	41,83%	80,37%	1,60%	58,17%	99,77%
VLIG+PPOA	100,00%			0,00%	0,00%		100,00%
VTD	95,17%	47,79%	64,51%	73,16%	6,24%	35,49%	99,13%
VTD!PPOA	100,00%		100,00%		0,00%	0,00%	100,00%
VTD+PPOA	99,93%			0,00%	0,00%		99,93%
VTI	98,58%	34,47%	72,70%	46,74%	1,37%	27,30%	99,57%
VTI+PPOA	100,00%			0,00%	0,00%		100,00%
[	100,00%	100,00%	0,00%	100,00%	0,01%	100,00%	100,00%
]	100,00%	88,89%	0,00%	80,00%	0,01%	100,00%	100,00%
~GERAL~	98,67%	53,40%	46,60%	53,40%	1,42%	53,40%	99,32%

Tabela 13 – Estatísticas por etiqueta do *YAS-Tagger* sobre corpus de Aires

Fonte: O autor



## **Anexos**