# Capítulo 1

# Introdução

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

Dominar o próprio idioma é um requisito essencial a qualquer indivíduo que viva em sociedade. Tal domínio é necessário tanto para exercer a cidadania plena, quanto para atuar profissionalmente. Neste último caso, existem termos específicos que variam de acordo com a profissão e colegas de ofício esperam que seus pares conheçam. Por exemplo, profissionais atuantes no ramo jurídico utilizam termos específicos à área, tais como *habeas corpus*, abroquelar, cártula chéquica, areópago, fulcro e estipêndio funcional; no ramo médico, os termos poderiam ser incisão, febrífugo, benigno, ejunorrafia ou abdução. Isso pode ser um problema na comunicação entre profissionais de diferentes áreas que desconhecem completamente a área do outro: "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. Aprender uma nova área de atuação profissional não envolve apenas o "saber fazer", mas também o "saber comunicar-se", já que muitas vezes é necessário compreender as ordens dos superiores, redigir relatórios e emitir declarações. A aquisição de um repertório a ser utilizado é necessária para o uso do idioma cotidianamente para que exista uma comunicação entre o indivíduo e seus diversos interlocutores. De modo semelhante ao âmbito profissional, o ensino de uma língua estrangeira também depende do aprendizado de novas palavras e seus respectivos significados.

Para auxiliar a aprendizagem de uma língua estrangeira (**??**, **??**; **??**, **??**; **??**, **??** apud **??**, **??**) ou para fins específicos (**??**) como no cotidiano profissional, a análise de *corpus* de textos é uma técnica que pode ser utilizada. Segundo **??**), Thorndike publicou em 1921 um trabalho que fez uso desta técnica, o qual 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*[1], [...] [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 a obtenção da informação reduziu a frações de segundo e possibilitou 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 **??**), 51% do total de domicílios possuem acesso à internet, 86% dos alunos acessaram a internet ao menos uma vez no último trimestre a 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 cresceu de tal forma que trabalhos manuais tais como a contagem de palavras ou busca em textos, por exemplo, puderam ser automatizados de forma que seu tempo ser reduzido vertiginosamente. A exemplo, um computador portátil Toshiba 2250XCDS, de 2000, vinha equipado com um disco rígido de 6GB, 64MB de RAM e um processador de 600MHz de um único núcleo (**??**), e um computador portátil CCE n325, de 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) (**??**). Logo, o poder de processamento das máquinas vendidas ao consumidor final, no varejo, aumentou muito, o que permite aplicações mais complexas.

Quanto ao poder de processamento remoto, o *Google Cloud Platform* oferece gratuitamente o que chama de *Cloud Launcher* (**??**). Tal serviço gratuito consiste de uma máquina virtual com IP estático, 30GB de armazenamento secundário, 0.6GB de RAM, 1 CPU virtual que recebe 20% do tempo da CPU da máquina real e 1GB de tráfego mensal de saída para fora da América do Norte, Austrália ou China (**??**). 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.

[1] *Corpora* é o plural de *corpus*, que é um conjunto de textos relevante para estudar um determinado grupo de pessoas ou um ramo do conhecimento (**??**, p. 358).

Resgatando as aplicações anteriormente mencionadas, segundo **??**), **??**), **??** (**??** apud **??**, **??**) e **??**), hoje, tanto a montagem de *corpora* quanto seu processamento são, ambos, tarefas menos trabalhosas que no passado. No início do século 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 pesquisar tal área do conhecimento, tanto locais (executadas no próprio dispositivo) quanto de internet (executadas em um servidor com interface visível através de um navegador), mas isso não acontece. Na próxima subseção (1.1) apresentaremos a contribuição almejada por este trabalho.

## 1.1 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*; inexistindo uma gratuita e com funcionalidade simultaneamente. Tal lacuna justifica a criação de outra ferramenta que apresente tais características desejáveis.

A criação de uma nova ferramenta baseada em computador requer a delimitação de sua plataforma-alvo, que pode se tornar um fator limitador a seu público alvo. 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*), e uma aplicação comum nestes é um navegador de internet. Portanto, desenvolver uma aplicação de internet significa possuir uma única base de código que atenda simultaneamente a todas as plataformas que possuem um navegador de internet, além de atingir um grande número de pessoas em território nacional nas quantidades mencionadas anteriormente.

Apresentada a ideia central do sistema, esta pode ser condensada visualmente numa tabela ao mesmo tempo que mostra os principais atributos de seus simila-

Tabela 1.1: Tabela comparativa resumida de *softwares* de Processamento de Linguagem Natural

| 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 |

**Fonte:** O autor

res. A tabela 1.1 traz a ferramenta proposta em sua última linha (*Corpus Slayer*, destacada em fundo amarelo). Com isso, os objetivos deste trabalho podem ser formalizados na seção a seguir (1.2).

## 1.2 Objetivos

### 1.2.1 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.

### 1.2.2 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[2] que suporte busca por etiquetas.

[2]Concordanciador é o "programa que extrai todas as ocorrências de uma palavra de busca num *corpus* juntamente com seu cotexto [...]" (**??**, p. 358).

- Integrar os módulos desenvolvidos ou adptados numa aplicação *web*.
- Disponibilizar uma ferramenta livre para uso educacional.

## 1.3 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, serão apresentados os materiais e os métodos utilizados para desenvolver este trabalho, bem como as decisões relevantes ao desenvolvimento do projeto. No capítulo 4, serão apresentados e discutidos os resultados obtidos. No capítulo 5, serão apresentados a conclusão, as contribuições deste trabalho e temas para trabalhos futuros.

# Capítulo 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 **??**), 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" (**??**, p. 358), e estes buscam representar a linguagem (**??**), ou uma variante desta (**??**), podendo ser usada como forma de obtenção de informações nas mensagens nelas codificadas. **??**) 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 (**??**), cujo produto foi uma lista de frequência de palavras. Uma das possíveis aplicações da linguística é extrair palavras-chave, esta que **??**) 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 terem sido calculadas computacionalmente ou manualmente. **??**) 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 **??**), mudou o paradigma de abordagem na linguística, saindo o empirismo e entrando o racionalismo; ou seja, a análise de volumosos *corpora* de textos, que era sujeito a inúmeras falhas, deu lugar a "[...] teorias racionalistas da linguagem [...], notadamente a linguística gerativa. [...]"(**??**, 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 para esta necessidade foi modelá-la como uma gramática gerativa de Chomsky (**????**). Um dos desafios encontrados foi resolver ambiguidades, que foi solucionado usando autômatos gerados a partir de um tratamento estatístico de ocorrências para a criação dum modelo probabilístico para a etiquetagem de palavras (partes do discurso) (**????**). Todavia, tal abordagem desconsidera que uma sequência 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 (**??**, **??** apud **??**, **??**). Segundo **??**), o reconhecimento destas no texto se dá através de perguntas como "Quem, Quando, Como, Onde" (p. 6). **??**) menciona datas, números e unidades monetárias como elementos do texto comumente inclusas como entidades nomeadas, sendo que **??**) também incluem 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 o foco é o estado da arte do processamento automático de *corpora*, indicando como os problemas do passado são resolvidos atualmente. Primeiramente será apresentado o 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 **??** (**??** apud **??**, **??**, 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. **??**) mencionam 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 (**??**). Outra ferramenta é o *AntCorGen*, de **??**), a qual visa manter organizados *corpus* de texto através de categorias, subcategorias e seus metadados, tais como: 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

Segundo **??**), o atual estado da arte para etiquetadores morfossintáticos da língua portuguesa é o *PALAVRAS*, de **??**. 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 (**??**, 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, **??**) destaca o *LX-Parser*. Este possui uma taxa de acerto de 75% em relação aos seus próprios resultados, e 97,5% se comparado a uma extração manual (**??**, p. 170). Apesar de gratuito, sua licença é de redação própria (**??**) 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 sua descarga e execução local.

Outro projeto é o *Unitex/GramLab*, que teve sua versão em português brasileiro foi primeiramente apresentada por **??** em **??**. 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 (**??**), já seus recursos linguísticos estão sob a *Lesser General Public License For Linguistic Resources* (**??**), 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 (**??**).

## 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 **??**) na subseção 2.3.1, o de **??**) na subseção 2.3.2, o de **??**) na subseção 2.3.3, o da **??**) na subseção 2.3.4 e o da **??**) 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 **??**) (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 nas quais alguns se destacam e outras em que apresentam dificuldades. Tais ferramentas comparadas eram as principais ferramentas existentes até 2014, que seriam utilizadas caso suas respectivas licenças assim permitissem e executassem no servidor.

### 2.3.2 Sistema identificador de sintagmas verbais do PB

A língua portuguesa possui sintagmas nominais, verbais, adjetivais e preposicionados (**??**). Os verbais podem ser analisados conforme proposto por **??**), que 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" (p. 2), sendo definido pelos próprios autores como "um protótipo com resultados interessantes" (p. 3) que precisa ser melhorado. Os autores apontam sem mencionar valores numéricos que o que precisa ser melhorado é a quantidade de estruturas reconhecidas pela gramática definida e a taxa de acerto na etiquetagem dos textos. Tal trabalho fornece uma ferramenta para obtenção automática da análise sintagmática, mas que necessita de aprimoramentos e quantificações numéricas quanto sua eficácia.

É uma abordagem diferente da utilizada neste trabalho pois trabalha sobre sintagmas (comumente modelado computacionalmente como uma árvore de palavras) ao invés de sentenças (comumente modelado computacionalmente como uma lista de palavras). Com o resultado declarado, o trabalho contribuiu para evitar a adoção de uma metodologia que possuísse o reconhecimento de uma complexa estrutura do texto como uma preocupação.

### 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 feitas por Bick no modelo da língua inglesa para operar para a língua portuguesa foi 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"(**??**, p. 438), em que algumas descobertas para a língua inglesa parecem se manter para a língua portuguesa, tais como: milhares de regras são necessárias para cada nível adicional de análise, as sentenças que excluem significado são mais frequentes que as que buscam e a determinação do contexto concentra-se mais à esquerda da sentença. Tal tese explicita as principais diferenças entre o processamento sintagmático das línguas inglesa e portuguesa, bem como suas diferenças e adaptações necessárias para se adequar à língua 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* (**??**) e revistos por linguistas" (**??**). A tabela 2.1 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" (**??**).

### 2.3.5 Implementação, adaptação, combinação e avaliação de etiquetadores para o português do Brasil

O **??**) 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 **??**), a qual contempla uma comparação de precisão entre os diferentes etiquetadores analisados (vide tabela 2.2).

Tabela 2.1: O material usado no projeto Floresta Sintáctica

| **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 |

**Fonte: ??**)

Tabela 2.2: Comparação da precisão entre os etiquetadores *MXPOST*, *Brill Tagger* e *Tree Tagger*

| Etiquetador | Precisão |
|---|---|
| MXPOST | $89,66\%$ |
| Brill Tagger | $88.76\%$ |
| Tree Tagger | $88.47\%$ |

**Fonte: ??**, p. 82)

## 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, 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 **??**), "[...] O Processamento [automático] de Linguagem Natural é uma disciplina com laços fortes com a Ciência da Computação e, embora compartilhe vários temas com a Linguística de *Corpus*, as duas mantêm-se independentes." (p. 328). Já de acordo com **??**, 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." **??** (**??** apud **??**, **??**, p. 158), por sua vez, defende que "a partir de *corpora*, podem-se fazer observações precisas sobre o real comportamento linguístico 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 estrita de uma área do conhecimento de interesse a partir de registros relevantes e confiáveis do uso desta.

### 2.4.2 Linguística 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 coleção 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 objetivos do investigador e o volume dos enunciados considerados como característicos do fenômeno a estudar. Um *corpus* é chamado exaustivo quando compreende todos os enunciados característicos. E é chamado seletivo quando compreende apenas uma parte desses enunciados (**??**, **??** apud **??**, **??**).

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 linguista deve, pois, procurar obter um *corpus* realmente significativo. Enfim, o linguista deve desconfiar de tudo o que pode tornar o seu *corpus* não-representativo (método de pesquisa escolhido, anomalia que constitui a intrusão de linguista, preconceito sobre a língua) (**??**, **??** apud **??**, **??**).

Ou, de acordo com **??** (**??** apud **??**, **??**) como uma coleção de peças de

um texto na forma eletrônica, selecionadas de acordo com critérios externos para representar, o quanto melhor possível, uma língua ou variação desta como dados fonte para pesquisa linguística. Para **??** (**??** apud **??**, **??**, 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 **??**), linguística de *corpus* é uma ciência empírica, na qual o pesquisador utiliza diversas vezes o *corpus* para estender uma descrição linguística, mas esta prática pode 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 que representam um nicho a ser estudado.

**??**) ainda cita três estatísticas deriváveis de uma análise de *corpus*, sendo elas 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 foram identificadas no texto, a segunda (dispersão e frequências ajustadas) é a proporção (percentual) das ocorrências dado um *corpus* e sua dispersão e a última (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 (**????**). Tais programas são úteis para o processamento de um *corpus* porque "ajudam a determinar as unidades para processamento linguístico" (**??**, 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 (**??**, p. 364). Ainda assim, existem dicionários que reconhecem "sentença" e "frase" como sinônimos, como a definição "oração, frase, período" (**??**, 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" (**??**) presente no dicionário online de português Dicio. De modo semelhante, mas contrário, tanto o novo dicionário Aurélio (**??**, p. 670), quanto o dicionário Houaiss (**??**, 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 que serão utilizadas na seção 4.1 do capítulo de resultados e nos apêndices 9, 10 e 11; abaixo, segue a lista das métricas utilizadas (que não é exaustiva sobre o conjunto de todas as existentes):

- Precisão (*precision*), dado pela equação 2.1, "denota a proporção de casos preditos como positivos que são considerados positivos reais" (**??**, p. 38).

$$P_{recisão} = \frac{V_{erdadeiro}P_{ositivo}}{V_{erdadeiro}P_{ositivo} + F_{also}P_{ositivo}} \tag{2.1}$$

- Revocação (*recall* ou sensibilidade), dado pela equação 2.2, "é a proporção de casos de casos positivos preditos como positivos" (**??**, p. 38).

$$R_{ecall} = \frac{V_{erdadeiro}P_{ositivo}}{V_{erdadeiro}P_{ositivo} + F_{also}N_{egativo}} \tag{2.2}$$

- *F-measure* ($F_1$ *score*), dado pela equação 2.3, "é definida como a média harmônica entre precisão e revocação" (**??**, **??** apud **??**, **??**, pp. 51, 52), mas, segundo **??**), não leva em conta os verdadeiros negativos.

$$F_1 = \frac{2 \times P_{recisão} \times R_{ecall}}{P_{recisão} + R_{ecall}} \tag{2.3}$$

- Acurácia (*accuracy*), dado pela equação 2.4, ao contrário das anteriormente apresentadas, "explicitamente toma em conta a classificação de negativos" (**??**, p. 39).

$$A_{curácia} = \frac{V_{erdadeiro}P_{ositivo} + V_{erdadeiro}N_{egativo}}{V_{erdadeiro}P_{ositivo} + V_{erdadeiro}N_{egativo} + F_{also}P_{ositivo} + F_{also}N_{egativo}} \tag{2.4}$$

- Taxa de erro (*miss rate*), dado pela equação 2.5, "é a proporção de positivos reais que são preditos como negativos" (**??**, p. 39).

$$E_{rro} = \frac{F_{also}N_{egativo}}{V_{erdadeiro}P_{ositivo} + F_{also}N_{egativo}} \tag{2.5}$$

- Especificidade (*specificity*), dado pela equação 2.6, "é a proporção de casos negativos que são corretamente preditos como negativos" (**??**, p. 39).

$$E_{specificidade} = \frac{V_{erdadeiro}N_{egativo}}{F_{also}P_{ositivo} + V_{erdadeiro}N_{egativo}} \tag{2.6}$$

- Prevalência (*prevalence*), dado pela equação 2.7 (**??**, p. 39), representa a proporção do universo amostral que é composta por casos positivos, sendo uma métrica que independe do classificador.

$$P_{revalência} = \frac{V_{erdadeiro}P_{ositivo} + F_{also}N_{egativo}}{V_{erdadeiro}P_{ositivo} + V_{erdadeiro}N_{egativo} + F_{also}P_{ositivo} + F_{also}N_{egativo}} \tag{2.7}$$

### 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" (**??**). Tal definição se aproxima da de terminal burro[1] 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.

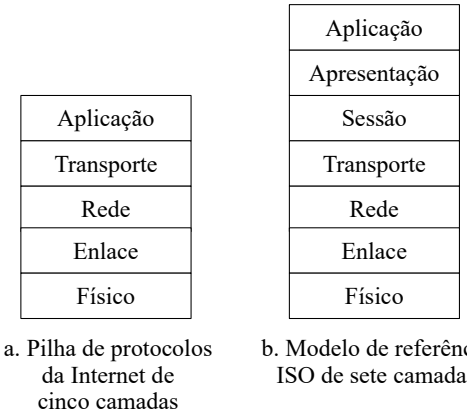
#### Arquitetura cliente-servidor

De acordo com **??**, 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 seu uso. Cada cliente e cada servidor possui uma pilha de protocolos da internet, assim como ilustrado pela figura 2.1 (**??**, p. 37). A comunicação entre os processos executados no cliente e no servidor, ilustrada pela figura 2.2, ocorre

[1] ***??*** (**??**) 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.

através de uma interface de *software* de troca de mensagens denominada *socket* (**??**, 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 2.1: A pilha de protocolos da internet (a) e o modelo de referência OSI (b)




a. Pilha de protocolos da Internet de cinco camadas

b. Modelo de referência ISO de sete camadas

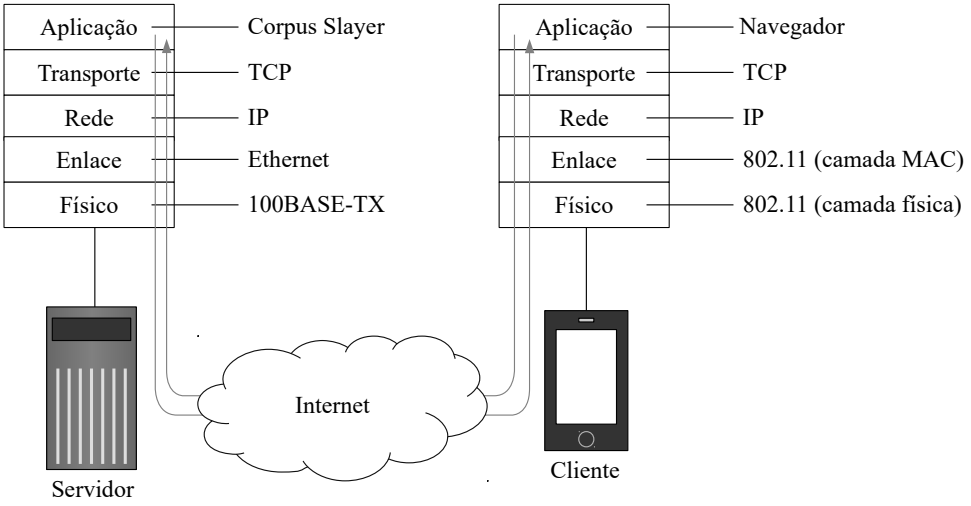
**Fonte: ??**, p. 37)

As menssagens 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"(**??**, p. 72). **??**) ainda afirmam 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 (**??**). O servidor *Web* abriga objetos que possuem endereços (URL), sendo estes acessíveis aos clientes através destes endereços (**??**). São *softwares* populares deste tipo o *Apache* e o *Microsoft Internet Information Server*(**??**), os quais são capazes de delegar a um algoritmo a geração da resposta a uma requisição (Apache Foundation, **??**; **??**, **??**). Algoritmos que geram a resposta a requisições de um cliente podem ser escritos utilizando o *framework Django*.

### *Framework* de aplicação de internet Django

Para facilitar o desenvolvimento de aplicações de internet, algumas *frameworks* foram criadas. Uma dessas *frameworks* é o *Django*, escrito na linguagem de programação *Python*, se autodefine como "a *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

Figura 2.2: Exemplo de canal de comunicação entre aplicações no cliente e no servidor




**Fonte:** O autor

*templates* e um mapeador objeto-relacional cujo banco de dados padrão é o *SQLite* (havendo suporte para outras soluções de persistência) (**??**).

*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" (**??**). Como solução de banco de dados, para **??**), é característica fundamental ofereça ao programador algum nível de abstração de dados. Para aumentar ainda mais a abstração, além das técnicas apresentadas por **??**), nos capítulos em que aborda as técnicas de programação de banco de dados, **??** instruem a 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 **??**, **??**) 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, em que as "aplicações de internet dos dias atuais são particularmente adequadas a esta abordagem" (p.1351). Portanto, a *framework Django* provê 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 **??**) de 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.

### 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" (**??**). Um desses deveres pode se aplicar aos provedores das aplicações de internet, quando este for uma pessoa jurídica atuando com fins econômicos:

rt. 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 (**??**).

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

III - 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 (**??**).

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.

# Capítulo 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 e as funcionalidades desejáveis identificadas; a seção 3.2, o gerenciador de *corpora* numa modelagem já utilizando a tecnologia escolhida; 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* (**??**). 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. A 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 3.1, 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;
- 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*[1] 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 para

[1] *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*).

Tabela 3.1: Tabela comparativa de *softwares* de Processamento de Linguagem Natural

| | WordSmith 5 | CorpusEye | COCA Online Corpus | Unitex/GramLab | Sketch Engine |
|---|---|---|---|---|---|
| Interface | Desktop<br>Win32<br>Windows | Web<br>Perl<br>Ubuntu 16.04 | Web<br>Asp<br>Windows Server 2012 | Desktop<br>Swing<br>Java | Web<br>Python 2.7<br>Fedora 25 |
| Versão completa gratuita | não | sim | sim | sim | não |
| Corpus fornecido pelo usuário | sim | não | não | sim | sim |
| Etiquetagem 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 | sim |
| Contagem de parágrafos | sim | não | não | não | sim |
| Tamanho em caracteres das palavras | sim | não | não | não | não |
| Caracteres por palavra | sim[1] | não | não | não | não |
| Palavras por sentença | sim[1] | não | não | não | não |
| Extração de palavras-chave | sim | não | não | não | não |
| Distribuição da palavra-chave no corpus | sim[2] | 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[3] | sim | sim | sim | sim |
| Posição da concordância no corpus | sim[2,3] | não | não | sim[4] | não |
| N-gramas | sim[3] | não | sim[3] | 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

**Fonte:** O autor

uma análise linguística 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*.

Dos *softwares* anteriormente listados, foi observado na tabela 3.1 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 a *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 a *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*" (**??**). 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. Trata-se de um *software* livre licenciado sob os termos da *GPL*, que não impõem 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 **??**), dando preferência aos etiquetadores com maior precisão.

## 3.2 O gerenciador de *corpora*

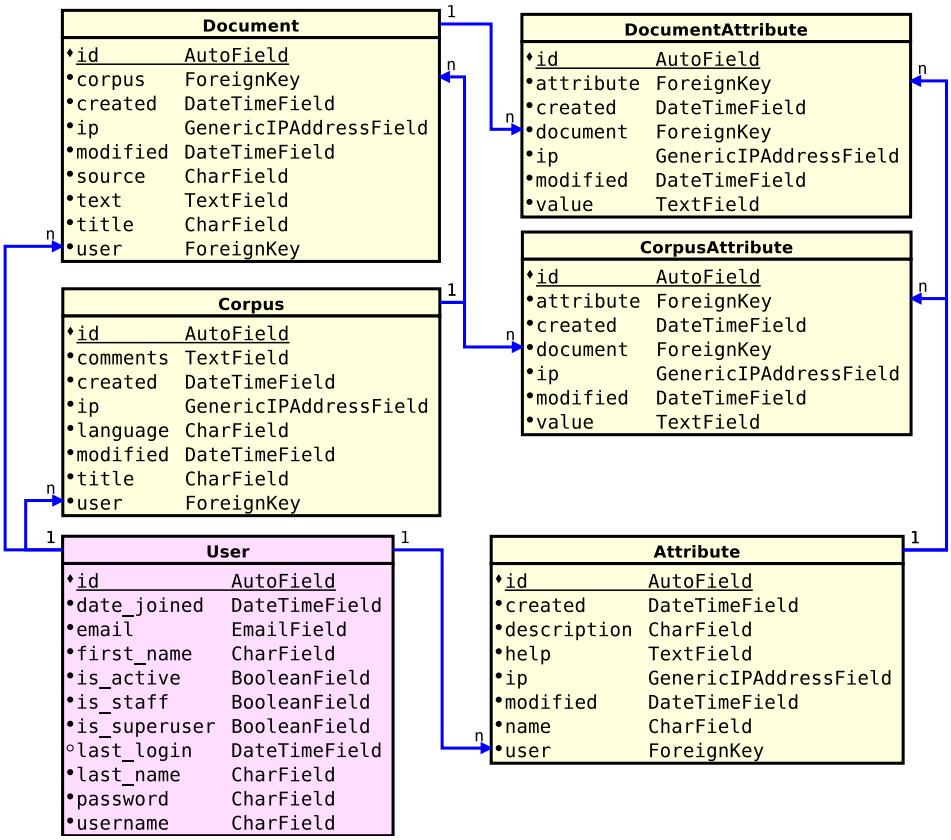
As classes do domínio do problema, quando representadas através da *framework* utilizada, se aproximam muito do modelo lógico. O mapeamento objeto-relacional feito pela plataforma admite herança e classes abstratas; como a solução de persistência padrão é *SQLite*, e esta não suporta herança entre tabelas, o mapeador objeto-relacional replica os campos definidos na classe pai na classe filho, já classes abstratas não geram tabelas. A figura 3.1 traz um excerto do modelo lógico gerado automaticamente a partir da implementação. A classe em magenta claro (*User*) é advinda da *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 6, 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.

## 3.3 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 (**??**), 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 (**??**). A forma que é adotada para representar tal navegação é através de uma árvore (vide figura 3.2), 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 (caixas hierarquicamente dependentes da caixa "Contas").

A página inicial, representada pelas caixa "Plataforma", na figura 3.2, 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,

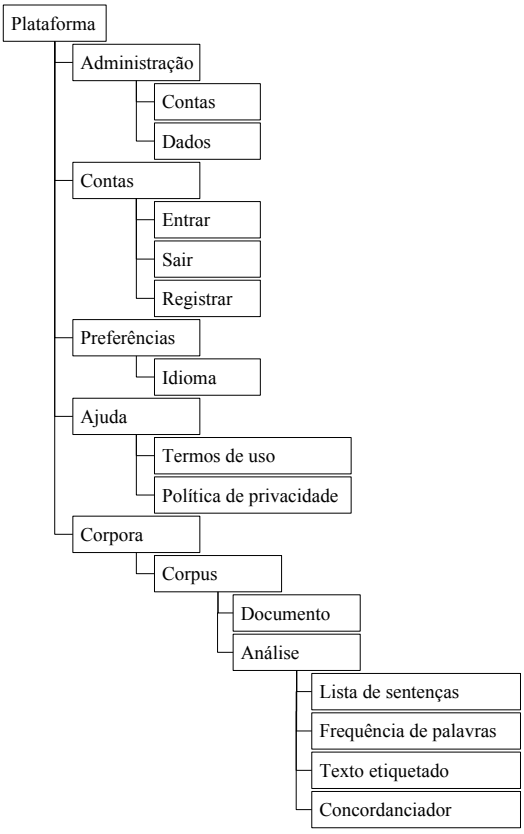
Figura 3.1: Excerto do modelo lógico gerado automaticamente a partir da implementação




**Fonte:** O autor

representa a página "*Corpora*" da figura 3.2, 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 conterá 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.

Figura 3.2: Árvore de navegação das páginas do sistema



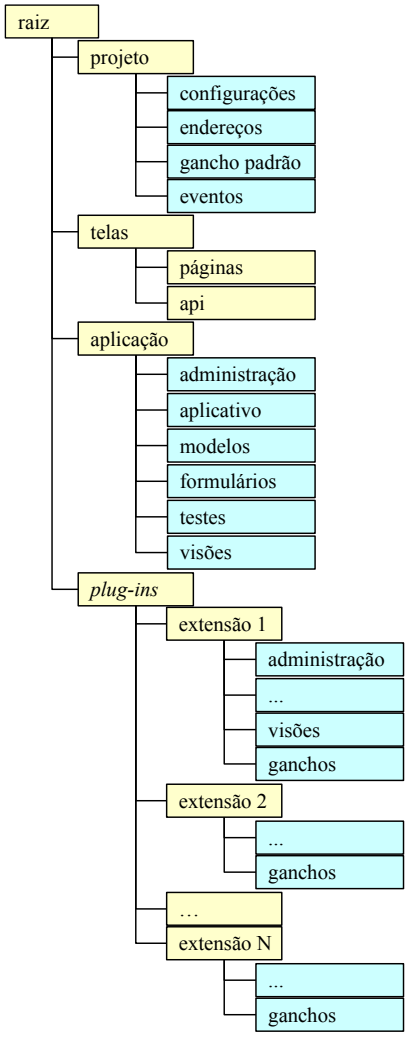

**Fonte:** O autor

## 3.4 O subsistema de eventos

Como ser extensível através de *plug-ins* é um requisito da aplicação e este recurso não é oferecido pela *framework* sem que seja necessário alterar o arquivo de configuração, somado à vantagem de que os *plug-ins* sejam substituíveis, foi considerado melhor 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 3.3 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

Figura 3.3: Árvore de diretórios e *scripts*




**Fonte:** O autor

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 entrada do etiquetador seja consistente.

Eventos do tipo "ação" recebem um objeto qualquer e não retornam nada. Foram idealizados para notificar outros *plug-ins* de 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 7. 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 (**??**). Com isso, o trabalho foi apenas adaptar a visualização do dado à nova

interface. A 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 em 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, em que 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 os 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[2], 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 3.2.

O conjunto de treino (quadro 3.2) 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 2.801.255 palavras em 176.034 sentenças. Como cada palavra é uma amostra para o treino

[2]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 (“/”) (**??**).

Tabela 3.2: Separação entre treino e teste

| 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 |

**Fonte:** O autor

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/GramLab* usou do próprio conjunto de etiquetas para etiquetar o texto, estas sendo incompletamente documentadas apenas pelo conteúdo do quadro 3.3. 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 **??**) e a breve e incompleta tabela de **??**) (quadro 3.3), 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 3.4 e 3.5 (teste 1, resultados na seção 4.1.1).

O etiquetador do *Unitex/GramLab* (**??**) usa "tuplas de unigramas, bigramas e trigramas" (p. 329) gerados a partir do *corpus* fornecido como treino para que a aplicação, durante 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, **??** 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 **??**) 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, subs-

Tabela 3.3: Etiquetas gramaticais frequentes

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

**Fonte: ??**, p. 47)

tantivo 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 a 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 realizada mais uma avaliação, esta utilizando o *corpus*

Tabela 3.4: Conversão da etiquetas utilizadas do Floresta Sintáctica para notação intermediária

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

**Fonte:** O autor

Tabela 3.5: Conversão da etiquetas utilizadas do Unitex/GramLab para notação intermediária

| Etiqueta | Convertido |
|---|---|
| A | ADJ |
| X | N |
| SIGL | N |
| ABREV | N |
| PRO | PRON |
| PREPXDET | PREP |
| PREPXPRO | PREP |
| PONCT | ??? |

**Fonte:** O autor

anotado por **??**), este disponibilizado publicamente pelo **??**), 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 sem 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).

## 3.7 A proposta do concordanciador

O concordanciador, um dos requisitos desta aplicação, como definido por **??**), 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 3.1 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 foram reservadas as construções "{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 o que disse nos microfones em rede nacional" é uma sentença que é possível de ser reconhecida por esta busca.

# Capítulo 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 9, 10 e 11, estas, medidas através das estatísticas apresentadas na seção 2.4.4, onde os espaços em branco nas tabelas são resultados de divisão por zero ou cujo resultado não seja um número real ($\mathbb{R}$); para conveniência do leitor, a tabela 4.1 traz as estatísticas do desempenho de cada etiquetador como um todo, que é a transcrição da última linha da última tabela de cada apêndice anteriormente mencionado. Nos momentos em que o desempenho da implementação é mencionado, o computador *desktop* utilizado possui as configurações expressas no apêndice 8.

### 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 9.2 (apêndice 9), encontram-se as estatísticas de cada etiqueta derivadas a partir da tabela de contingência (tabela 9.1, apêndice 9), a qual mostra que apenas as etiquetas "PREP", "PRON" e "V" demonstraram desempenho satisfatório (precisão maior que 75%), que representam preposições, pronomes e verbos.

Uma característica deste etiquetador é que este possui um estágio anterior de enumeração de quais etiquetas são possíveis que aquela palavra represente e seu processo de "etiquetar" é escolher qual a etiqueta mais provável que cada palavra

Tabela 4.1: Estatísticas gerais dos etiquetadores treinados discutidos nas subseções da seção 4.1

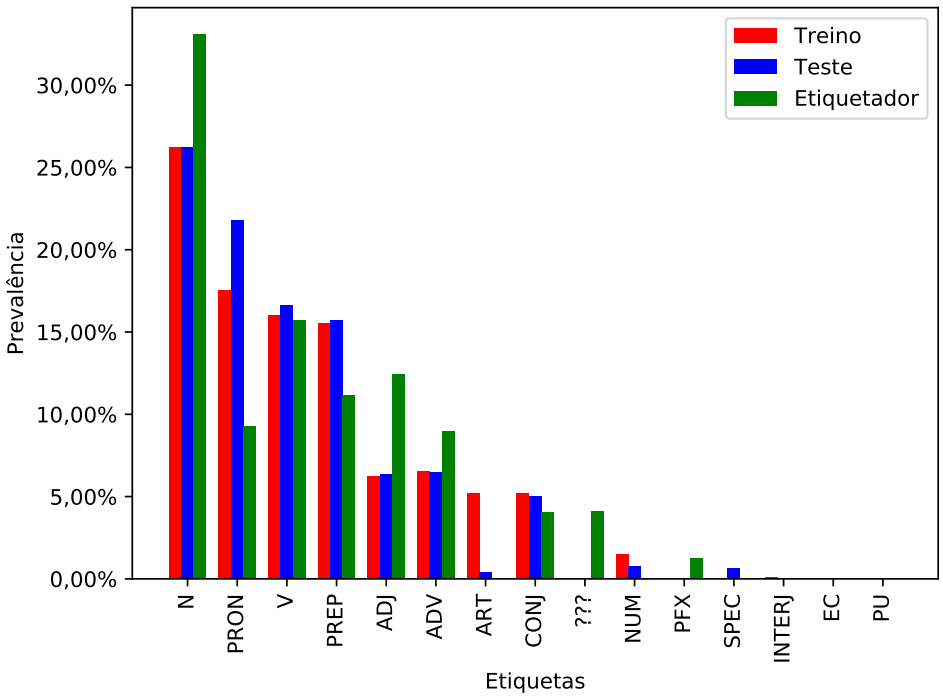
| Teste | Acurácia | $F_1$ score, Precisão e Revocação | Taxa de erro | Prevalência | Especificidade |
|---|---|---|---|---|---|
| Seção 4.1.1 | 94,77% | 60,76% | 39,24% | 6,67% | 97,20% |
| Seção 4.1.2 | 98,85% | 76,93% | 23,07% | 2,50% | 99,41% |
| Seção 4.1.2 | 98,67% | 53,40% | 46,60% | 1,42% | 99,32% |

**Fonte:** O autor

receba. Disso, pode-se destacar que preposições (“PREP”; exemplos: em, no, na, de, do, da, a, etc.) possuem número finito e conhecido; pronomes (“PRON”; exemplos: eu, tu, ele, nós, vós, eles, me, mim, te, ti, etc.), também; e verbos (“V”) possuem flexões que, na maioria das vezes, são indistinguíveis de outras classes morfológicas, o que significa que o etiquetador acertou em alguns lugares onde era esperado que ele não errasse. Também vale ressaltar que tais etiquetas estão entre as quatro etiquetas mais prevalentes tanto no conjunto de treino quanto no conjunto de testes, como pode ser observado na figura 4.1. Como é esperado que o etiquetador etiquete com maior precisão aquelas etiquetas com mais amostras no treino, foi inesperado observar que uma precisão ruim (59,25%) veio da etiqueta mais frequente em ambos treino e teste, a qual representa substantivos (“N”). Por isso, o reconhecimento de padrões linguísticos do *Unitex/GramLab* para o português brasileiro é mais limitada que o esperado para ser esta ferramenta seja incluída na ferramenta deste trabalho.

Outro ponto relevante é quanto a performance do etiquetador. O treino deste etiquetador levou alguns poucos segundos, o que é bom, mas ao usar o *pipeline* desenvolvido que seria executado no servidor caso fosse considerado um etiquetador viável, este levou mais de 72 horas para concluir o processamento, e utilizava, num momento arbitrário de observação já no segundo dia de execução, 14 GB da memória principal mais 30 GB de *swap* (paginação em disco), o que excede a capacidade do servidor. 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.

Figura 4.1: Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo *Unitex/GramLab*




**Fonte:** O autor

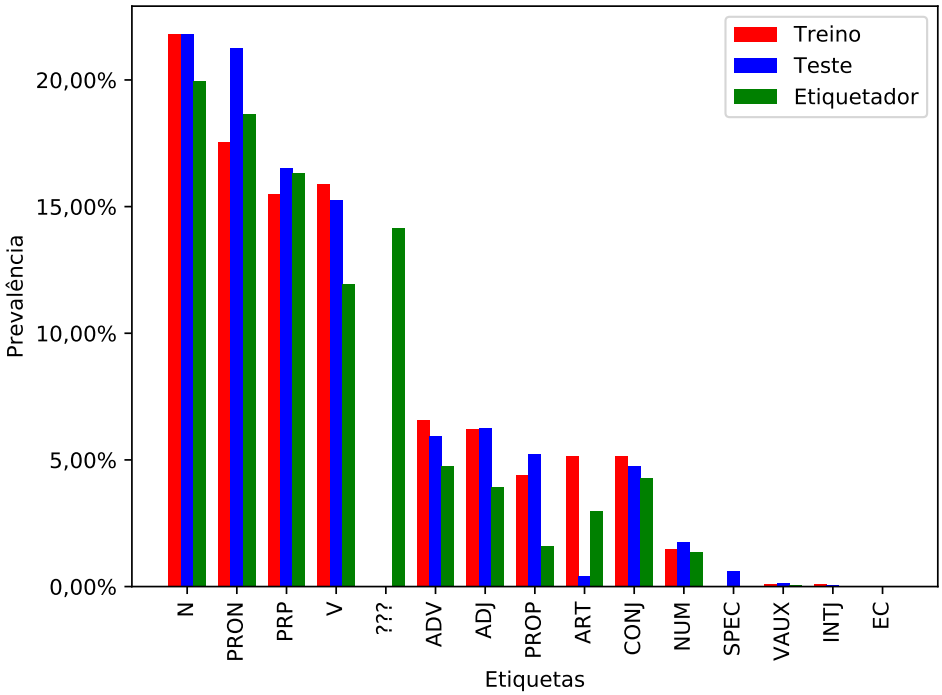
### 4.1.2 Teste 2: Etiquetador *YAS-Tagger*

Das 4.571.238 etiquetas comparadas, 3.516.634 foram atribuídas corretamente ($Precisão \approx 76,93\%$). Na tabela 10.2 (apêndice 10), encontram-se as estatísticas de cada etiqueta derivadas a par tabela de contingência (tabela 10.1, apêndice 10), nesta estão destacadas em cinza algumas 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.

O etiquetador, como um todo, demonstrou desempenho satisfatório (precisão maior que 75%), mas também errou 14,13% das etiquetas atribuindo a etiqueta “???”. Tal etiqueta foi reservada para o etiquetador demonstrar que não foi possível atribuir nenhuma outra etiqueta, sendo consequência de um treino que não recebeu

nenhuma amostra da construção encontrada no texto durante o teste. Isso pode indicar que o etiquetador não possui boa capacidade de identificação dos padrões da língua, hipótese esta que pode ser testada com um *corpus* menor, o que gera a dúvida a ser respondida: o quanto isso afetaria o desempenho negativamente?

Figura 4.2: Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo *YAS-Tagger*




**Fonte:** O autor

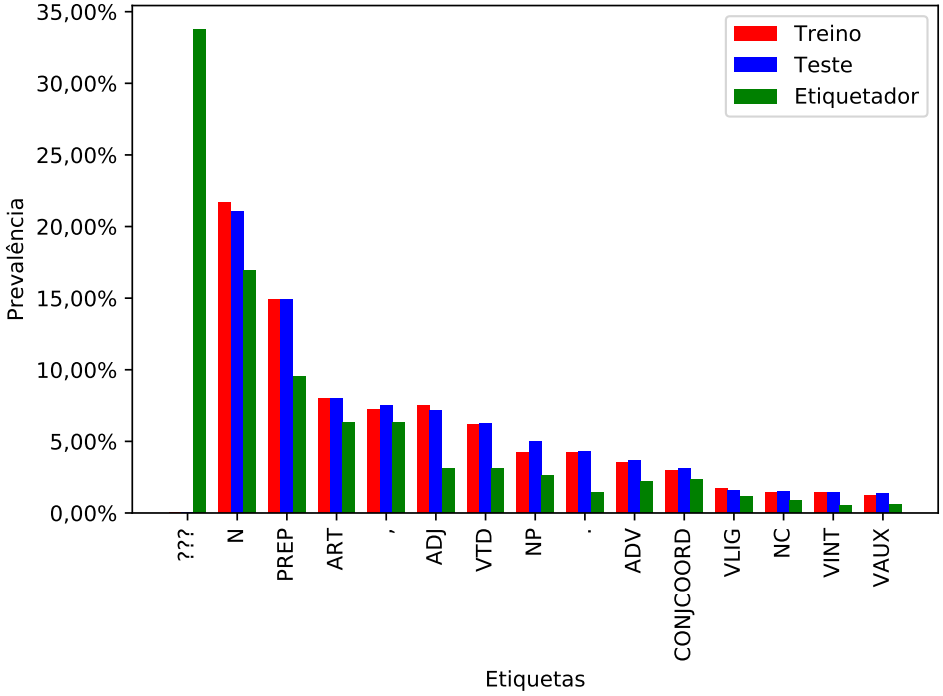
**Teste 2.1: Etiquetador *YAS-Tagger* sobre *corpus* de ??**

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 11. Observada essa vertiginosa 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 4.3).

Tais mudanças indicam que este etiquetador não possui boa capacidade de identificação dos padrões da língua e é altamente dependente de seu *corpus* de treinamento e, por isso, seu treino deve ser suficientemente volumoso apresentar bons

resultados no teste. Como a intenção é que o etiquetador seja capaz de generalizar a língua de forma a não depender de um grande conjunto de treino para obter bons resultados, isso conta negativamente para a escolha deste, principalmente enquanto parte de uma aplicação de internet, cuja entrada é toda e qualquer que usuário forneça, e este espera que as etiquetas sejam atribuídas corretamente em todos os casos. Por isso outro algoritmo etiquetador terá a preferência.

Figura 4.3: Prevalência das 15 etiquetas mais frequentes do conjunto de treino, teste e etiquetado pelo *YAS-Tagger* sobre o copus de **??**




**Fonte:** O autor

Concluídos os testes, com os dados da tabela 2.2 (da seção 2.3.5), pode-se montar a tabela 4.2 com os melhores resultados dos etiquetadores testados. Em tal tabela, fica evidente que os etiquetadores desenvolvidos e/ou treinados neste trabalho possuem uma precisão ruim, fazendo-se necessário utilizar um etiquetador de terceiros.

Tabela 4.2: Comparação da precisão entre os etiquetadores *MXPOST*, *Brill Tagger*, *Tree Tagger*, *YAS-Tagger* e *Unitex/GramLab*

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

**Fonte:** O autor e **??**, p. 82)

## 4.2 O sistema desenvolvido

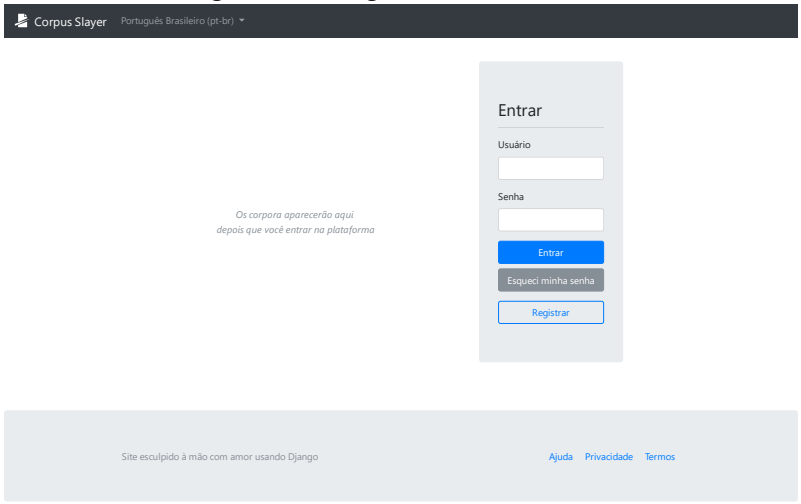
Nesta seção, será apresentado o sistema enquanto aplicação implantada, bem como as decisões e soluções para os problemas apresentados no capítulo anterior. Apesar de não ter sido mencionado anteriormente, durante a codificação foi utilizado o sistema de versionamento de código *GitLab*, com implantação 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 4.4), 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 4.5), no rodapé, um texto estático alinhado à esquerda arbitrariamente definido durante a tradução e três *links* que levam a páginas nas quais conteúdo e título são definidos na tradução. Porque o formulário de *login* é exibido nesta tela juntamente ao botão que leva ao fomulário de cadastro, está ausente nesta página, mas a figura 4.6 traz esses botões em detalhe.

Após autenticar-se na aplicação, ao usuário é exibido uma tela como a da figura 4.7, 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 documentos daquele *corpus*, como na figura 4.8, 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".

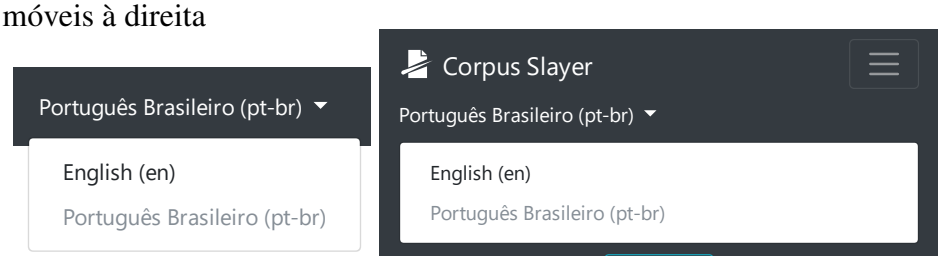
Na página de análises (figura 4.9), são encontrados os serviços dos *plug-ins* desenvolvidos já aplicando o abordado na seção 3.4. Nas seções seguintes, serão abordados o *Unitex/GramLab* como separador de sentenças e gerador de lista de palavras (seção 4.2.1), o *MXPOST* e o *Tree Tagger* enquanto etiquetadores (seção 4.2.2) e a implementação do concordanciador proposto na seção 3.7 (seção 4.2.3).

Figura 4.4: Página inicial do sistema desenvolvido




**Fonte:** O autor

Figura 4.5: Detalhe do seletor de idiomas, *desktop* à esquerda e em dispositivos móveis à direita

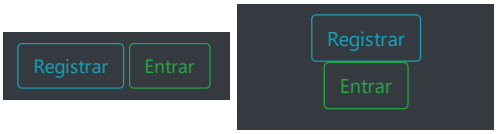



**Fonte:** O autor

### 4.2.1 Separador de sentenças e gerador de lista de palavras

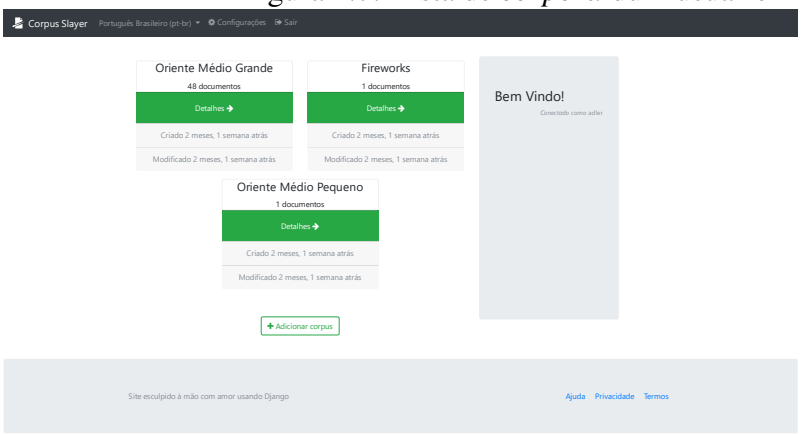
Usando o *Unitex/GramLab* como a parte que faz, de fato, o processamento dos dados, restou apenas encapsular o dado e exibi-los ao usuário. A primeira parte foi solucionada escrevendo o código a partir do manual de **??**), que serviu de base para a realização da segunda parte. Esta teve fortes influências da interface do Unitex/GramLab, no qual a lista de sentenças (figura 4.10) também exibe o número sequencial do item da lista e a lista de palavras (figura 4.11), que uma janela composta de três listas, tornou-se uma página com três 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.

Figura 4.6: Detalhe dos botões de autenticação, *desktop* à esquerda, sendo encontrado alinhado à direita na mesma barra em que aparece o seletor de idiomas, e em dispositivos móveis à direita, sendo encontrado imediatamente abaixo do seletor de idiomas




**Fonte:** O autor

Figura 4.7: Lista de *corpora* dum usuário




**Fonte:** O autor

### 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 4.12, mas foi mantido por funcionar para etiquetar com sucesso textos na língua inglesa. O segundo foi o *Brill Tagger*, que requer permissão de execução (figura 4.13) 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 Tagger*, cuja visualização é bastante similar à lista de sentenças, mas com pares palavra-etiqueta, como pode ser observado na figura 4.14.

### 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 tela de pesquisa do concordanciador desenvolvido (figura

Figura 4.8: Lista de documentos dum *corpus*


**Fonte:** O autor

4.15) uma tabela com a referência das construções aceitas e uma pré-visualização em tempo real de como o termo será entendido pelo buscador que está no servidor. Já nos resultados (figura 4.16), 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 4.16, bem como na 4.14, 4.11 e 4.10, e 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 4.17) ou uma lista deles em caso haja múltiplos resultados (figura 4.18).

### 4.2.4 A ferramenta desenvolvida

O último objetivo deste trabalho é "disponibilizar uma ferramenta livre para uso educacional". Para tanto, a tabela 4.3 traz os *URL*s dos repositórios que hospedam o código-fonte. Além disso, a aplicação desenvolvida foi implantada e disponibilizada no endereço eletrônico <https://corpusslayer.com>.

Tabela 4.3: *URL* para o código fonte das ferramentas desenvolvidas neste trabalho

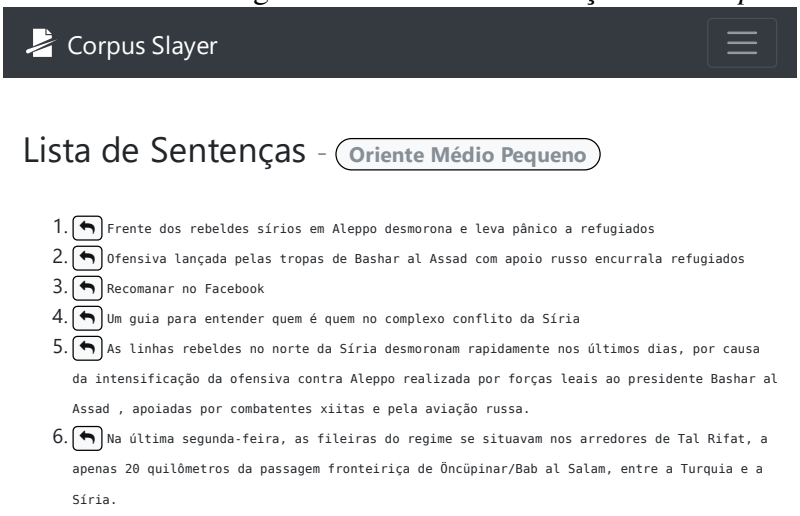
| *Software* | *URL* para o repositório do código fonte |
|---|---|
| *Corpus Slayer* | • <https://git.adlerosn.com/adler/corpusslayer><br>• <https://github.com/adlerosn/corpusslayer> |
| *YAS-Tagger* | • <https://git.adlerosn.com/adler/yas-pos-tagger><br>• <https://github.com/adlerosn/yas-pos-tagger> |

**Fonte:** O autor

Figura 4.9: Opções de análises para um *corpus*


**Fonte:** O autor

Figura 4.10: Lista de sentenças dum *corpus*

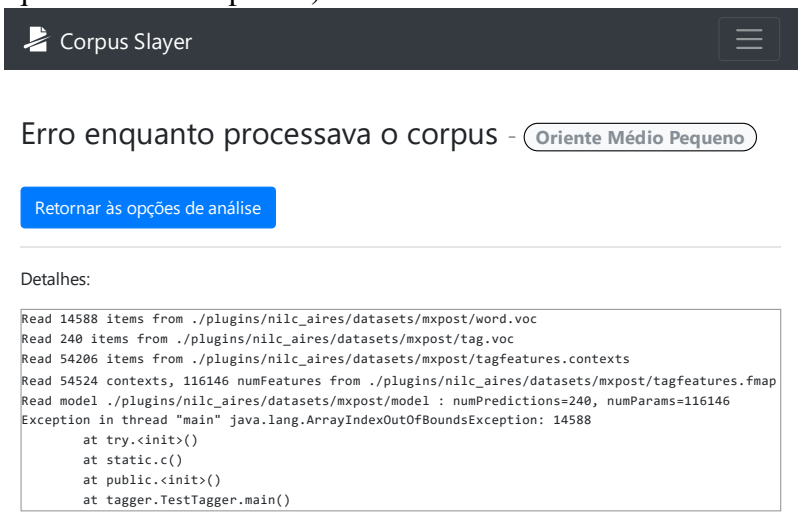



**Fonte:** O autor

Figura 4.11: Lista de palavras dum *corpus*


**Fonte:** O autor

Figura 4.12: Erro do *MXPOST* ao etiquetar para o Português Brasileiro utilizado arquivos treinados por **??**)



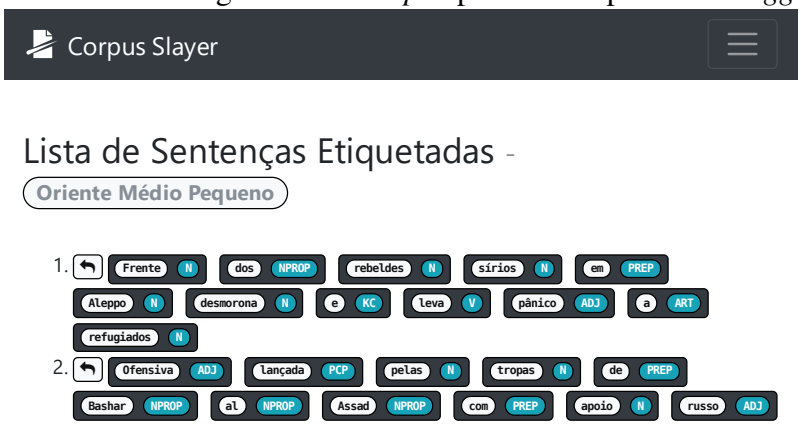

**Fonte:** O autor

Figura 4.13: Erro do *Brill Tagger* requisitando arquivos com permissão de execução

```
$ ./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
$
```

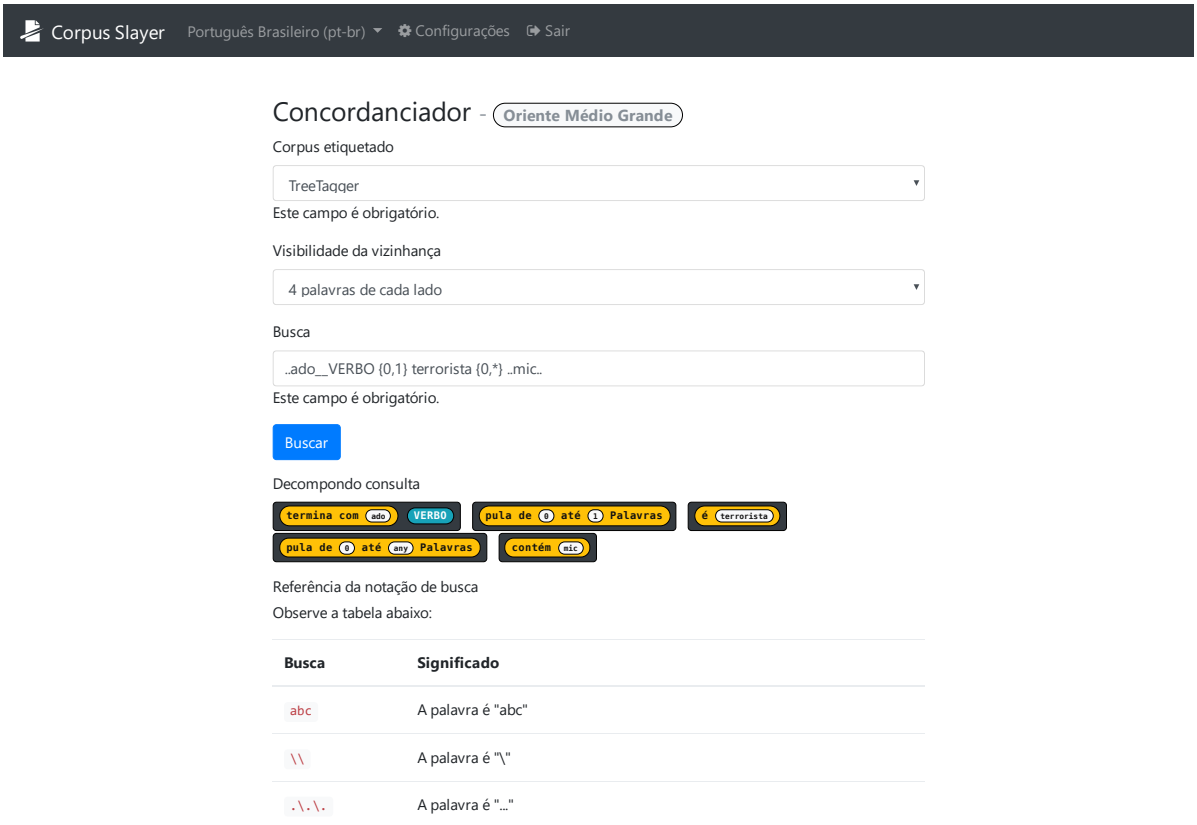
**Fonte:** O autor

Figura 4.14: *Corpus* processado pelo *Tree Tagger*




**Fonte:** O autor

Figura 4.15: Tela de busca do concordanciador




**Fonte:** O autor

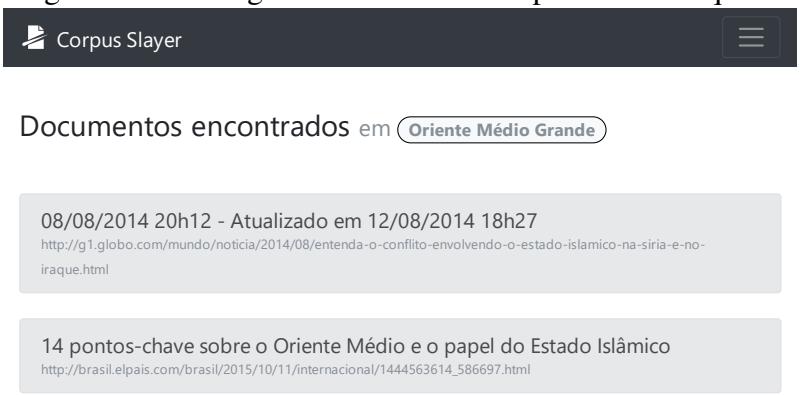
Figura 4.16: Tela de resultados do concordanciador


**Fonte:** O autor

Figura 4.17: Visualização de documento a partir dum clique na “seta para trás”


**Fonte:** O autor

Figura 4.18: Listagem de documentos a partir dum clique na “seta para trás”




**Fonte:** O autor

# Capítulo 5

# Conclusão

O objetivo deste trabalho, plenamente atingido, foi desenvolver uma aplicação *web* de código aberto para marcação e busca de partes do discurso em *corpora*.

Para concluir tal desenvolvimento, identificamos que havia espaço dentro dos *softwares* para estudo da língua para uma nova ferramenta gratuita, comparamos as existentes, levantamos requisitos, comparamos etiquetadores, apresentamos a ferramenta e a implantamos.

O maior benefício deste trabalho pode ser observado ao analisar custos com ferramentas para linguistas. O concorrente pago, de acordo com a tabela 1.1 (seção 1.1), é o SketchEngine, que custa 4,83 € mensais por conta de estudante (**??**), o que, para uma faculdade que oferta um único curso de letras de 8 semestres preenchidos com uma média de 36 alunos por período, custaria o montante de 16.692,48 €. Este, por ser em moeda estrangeira, varia de acordo com o câmbio e é sujeito a IOF (6,38%); supondo uma taxa de conversão de 1 € para R$ 4,50, aquele valor em reais, após IOF, seria R$ 79.908,57. Esta quantia pode ser um impeditivo, enquanto implantar uma instância do *Corpus Slayer* possui apenas o custo do servidor que o executará, que, dependendo da infraestrutura de TI da instituição, pode ser o mesmo servidor que já hospeda o site principal desta, reduzindo o custo de implantação a zero.

Apesar de tal comparação ser feita, o *Corpus Slayer* ainda carece de um tesauro, de um gerador de *corpora* a partir de buscas na *web*, de ferramentas para trabalhar com *corpus* paralelos, de geração de N-gramas, de identificação automática de palavras-chave, de ferramentas para análise de tendências, e das funcionalidades listadas na seção 3.1 que não foram implementadas neste trabalho, mas que podem ser adicionadas através de *plug-ins* desenvolvidos em trabalhos futuros.

Sendo uma das preocupações deste trabalho, mencionado como detalhe no objetivo, o problema de usabilidade do concordanciador relatado na seção (3.7) recebeu uma proposta de melhora desta. Entretanto, esta melhora na usabilidade não foi testada; apenas as premissas de usabilidade defendidas por **??**) foram

seguidas, ficando como trabalho futuro analisar comparativamente a usabilidade dos concordanciadores existentes de forma a propor melhoras na usabilidade destes. Também vale ressaltar que a proposta não suporta repetições ou o operador lógico "ou" como algumas sintaxes de outras ferramentas, o que pode ser adicionado em versões futuras.

Identificamos que tanto o *Unitex/GramLab*, quanto o *YAS-Tagger* possuem a limitação de que etiquetam apenas as palavras que já encontraram anteriormente, sem fazer atribuição de etiquetas por inferência a partir de prefixos e sufixos. Um trabalho futuro pode implementar tal comportamento em ambos com a finalidade de aumentar os acertos e diminuir os erros de ambos os etiquetadores, com ênfase nas capacidade de generalização para palavras que não faziam parte do conjunto de treino.

Na introdução (capítulo 1), comentamos brevemente o impacto que a linguística de corpus teve na educação do século XX e expressamos o desejo de que tais benefícios continuassem. No entanto, não submetemos a ferramenta desenvolvida a nenhuma situação a qual fosse utilizada como instrumento de aprendizagem. Por isso, é trabalho futuro fazer um estudo de caso que inicie na elaboração ou adaptação de um plano de ensino para fazer uso de ferramentas computacionais para o aprendizado da língua, de preferência com o *Corpus Slayer* na lista de ferramentas utilizadas, enquanto analisa comparativamente os resultados obtidos por ambas metodologias de ensino: a nova em contraste com a antiga.

A ferramenta desenvolvida pode ser utilizada, mesmo em face de todas essas limitações, para compreender melhor como algumas estruturas da língua são usadas por seus falantes. Um exemplo é o estudo da variação linguística típica de cada região do Brasil, que possui dimensões continentais e, por consequência, muitas variações linguísticas a serem estudadas.

# Bibliografia


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. **Implementaçao, adaptaçao, 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. **Calidoscópio**, v. 4, n. 3, p. 156–178, 2006.

AMORA, A. **Minidicionário Soares Amora da língua portuguesa**. 19ª. 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 mineraçao 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 Arhus, 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>. 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>. 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. 2ª. 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**. 1ª. ed. Rio de Janeiro: Editora Nova Fronteira, 1975. 5ª reimpressão, 1500 p.

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**. 1ª. 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**. 2ª. 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.

Lexical Computing CZ s.r.o. **Price List | Sketch Engine**. 2018. Disponível em: <https://www.sketchengine.eu/price-list/>. Acesso em: 29 jun. 2018.

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. **Lingvisticae Investigationes**, John Benjamins publishing company, v. 30, n. 1, p. 3–26, 2007.

Núcleo Interinstitucional de Lingúistica 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>.

RIJSBERGEN, C. J. van. **Information Retrieval**. 2ª. 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>. 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>.

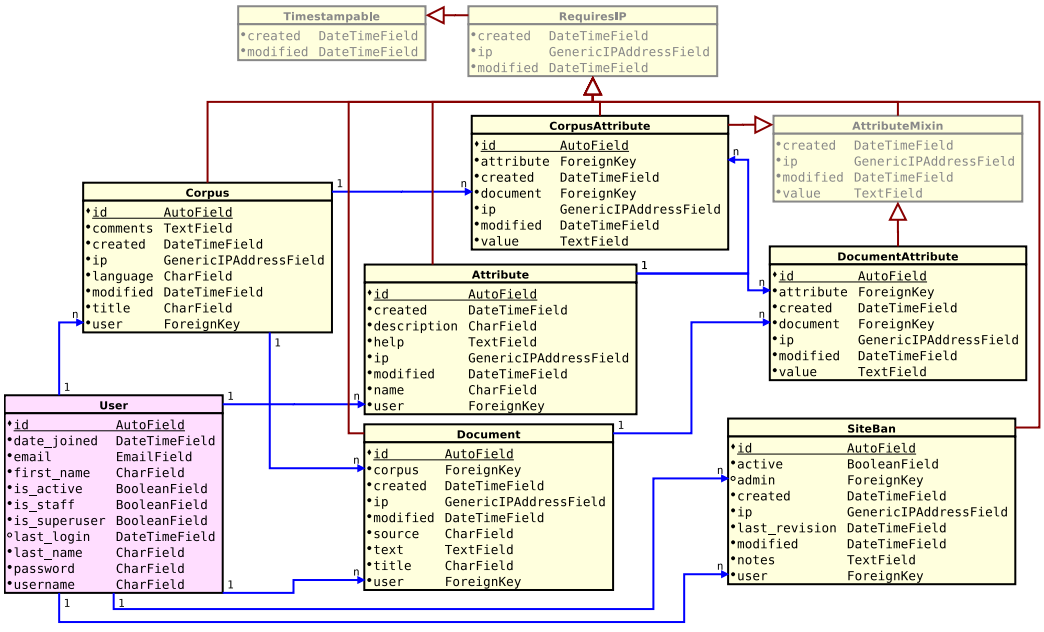
UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE. **Lesser General Public License For Linguistic Resources**. 2009. Disponível em: <https://spdx.org/licenses/LGPLLR.html>.

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.

# Capítulo 6

# O modelo lógico completo do gerenciador de corpora

Figura 6.1: Modelo lógico gerado automaticamente a partir da implementação, onde



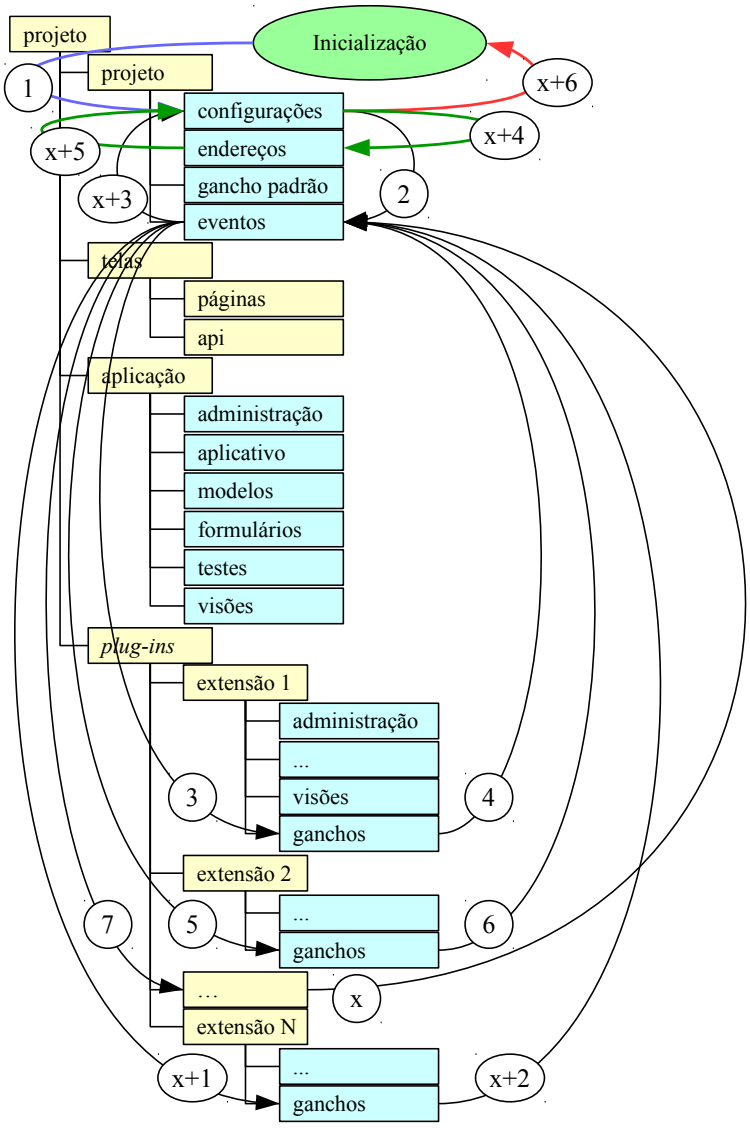

**Fonte:** O autor

# Capítulo 7

# Fluxos de execução do subsistema de *plug-ins*

## 7.1 Sequência de inicialização

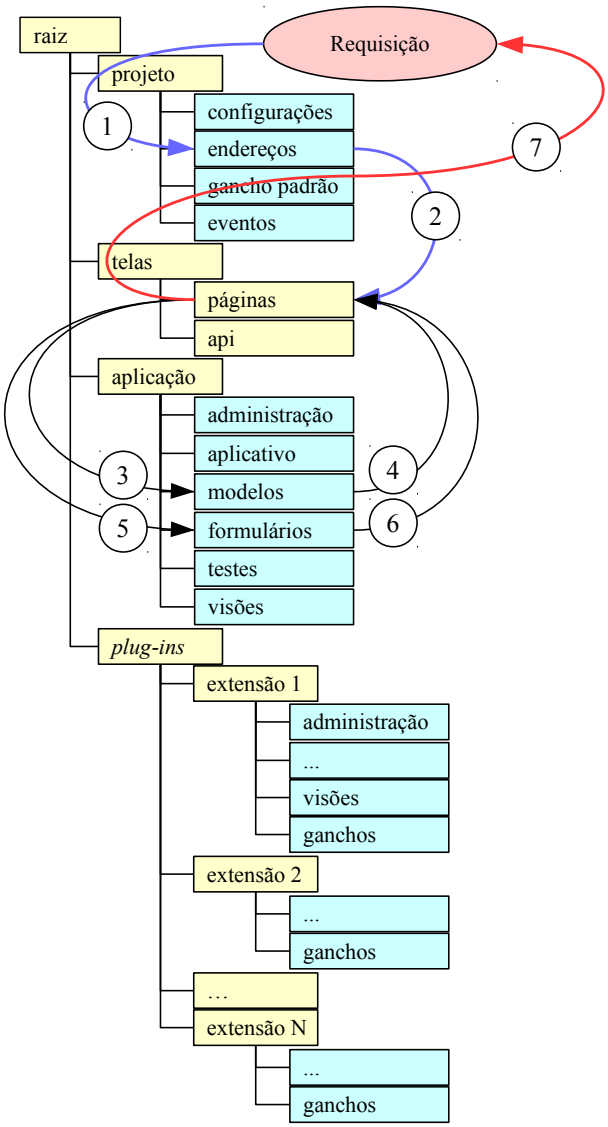
Figura 7.1: Esta figura traz a sequência de inicialização do subsistema de eventos. O primeiro arquivo que o *framework* Django carrega é o de configurações (seta "1"), que foi utilizado para inicializar o subsistema de eventos (seta "2"), que, por sua vez, carrega os "ganchos" (setas "3" e "4") do primeiro "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

## 7.2 Requisição sem uso de *plug-ins* para exibir formulário

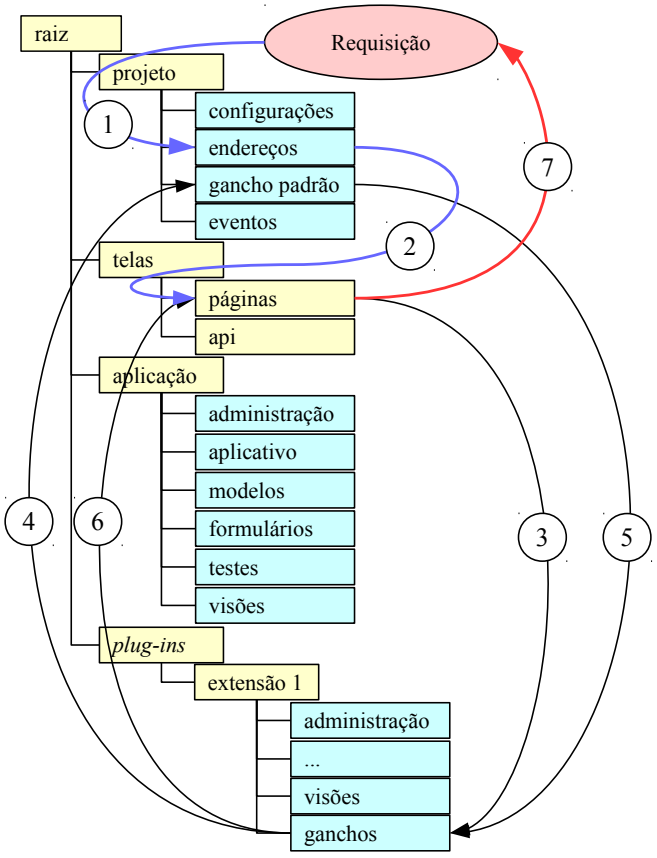
Figura 7.2: Esta figura traz como exemplo o cenário que exibe um formulário para editar um documento pertencente a um corpus. O *framework* busca o endereço especificado pela requisição (seta "1") e encontra uma entrada para uma classe de controlador no arquivo de "visões" presente no pacote "páginas" (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

## 7.3 Requisição para exibir lista de *plug-ins* na página “Análise”

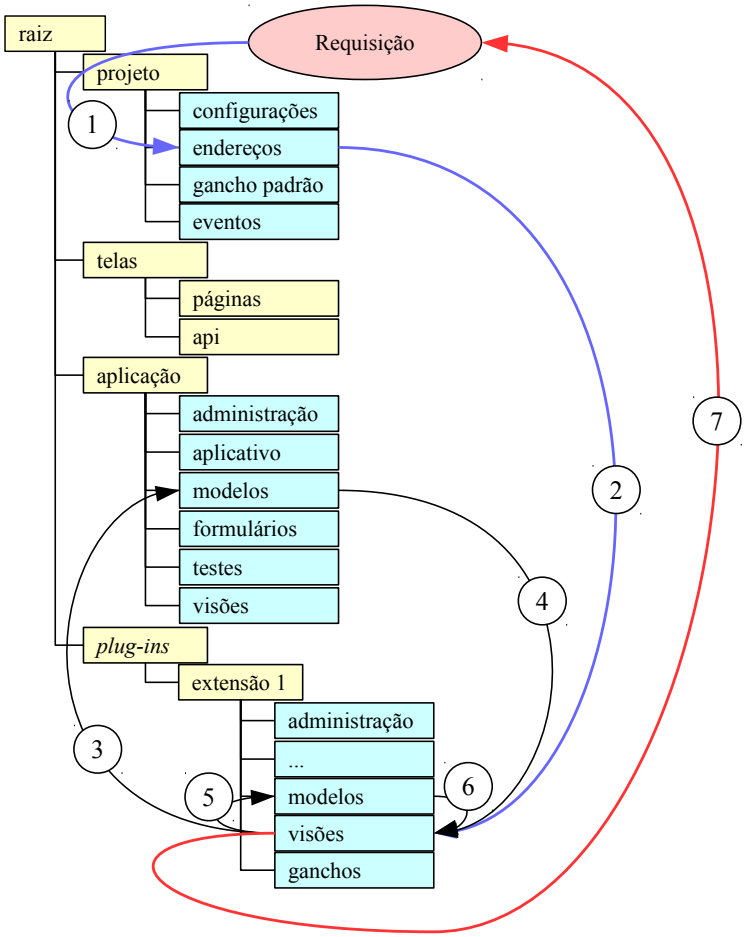
Figura 7.3: Esta figura traz como exemplo uma requisição para listar os *plug-ins* na página “Análise”. O *framework* busca o endereço especificado pela requisição (seta “1”) e encontra a entrada para a classe controladora da página “Análise” no pacote “páginas” (seta “2”), no qual tal controlador dispara 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

## 7.4 Requisição para exibir página gerada por *plug-in* que usa a aplicação

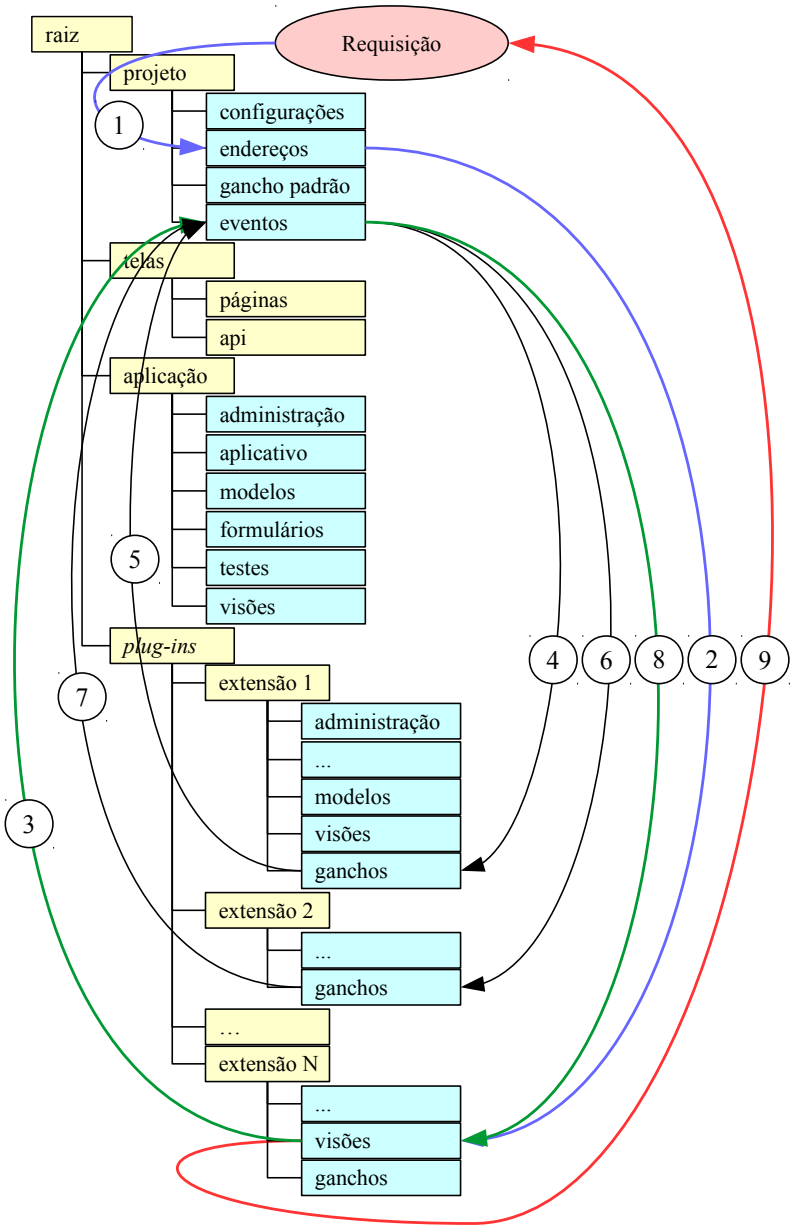
Figura 7.4: Esta figura traz como exemplo uma requisição para buscar uma lista de sentenças associada a um corpus. As setas "1" e "2" seguem o mesmo algoritmo dos parágrafo anterior, mas é executada uma classe controladora no arquivo de visões do *plug-in*, que, por sua vez, carrega o corpus a partir 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

## 7.5 Requisição para exibir página gerada por *plug-in* que usa outros *plug-ins*

Figura 7.5: 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

# Capítulo 8

# Hardware utilizado durante o treino e teste dos etiquetadores

- Sitema 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.

# Capítulo 9

# Tabela de contingência e suas estatísticas do etiquetador *Unitex/Gramlab*

Tabela 9.1: Tabela de contingência por etiqueta do *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 |

**Fonte:** O autor

Tabela 9.2: Estatísticas por etiqueta do *Unitex/Gramlab*

| | Acurácia | $F_1$ 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% |

**Fonte:** O autor

# Capítulo 10

# Tabela de contingência e suas estatísticas do etiquetador *YAS-Tagger*

Tabela 10.1: Tabela de contingência por etiqueta do *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 |

**Fonte:** O autor

Tabela 10.2: Estatísticas por etiqueta do *YAS-Tagger*

| | Acurácia | $F_1$ 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% |

**Fonte:** O autor

# Capítulo 11

# Tabela de contingência e suas estatísticas do etiquetador *YAS-Tagger* sobre corpus de Aires

Tabela 11.1: Tabela de contingência por etiqueta do *YAS-Tagger* sobre corpus de **??**

| | 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 |

**Fonte:** O autor

Tabela 11.2: Estatísticas por etiqueta do *YAS-Tagger* sobre corpus de **??**

| | Acurácia | $F_1$ 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% |

**Fonte:** O autor