**UNIVERSIDADE FEDERAL DO CEARÁ**
**CENTRO DE HUMANIDADES**
**DEPARTAMENTO DE LETRAS VERNÁCULAS**
**PROGRAMA DE PÓS-GRADUAÇÃO EM LINGUÍSTICA**

**KATIUSCIA DE MORAES ANDRADE**

# ASTROLÁBIO: UM CORPUS DE REDAÇÕES ESCOLARES DO CEARÁ ANOTADO MULTIDIMENSIONALMENTE CONFORME A TEI P5

**FORTALEZA**
**2013**

**KATIUSCIA DE MORAES ANDRADE**

# ASTROLÁBIO: UM CORPUS DE REDAÇÕES ESCOLARES DO CEARÁ ANOTADO MULTIDIMENSIONALMENTE CONFORME A TEI P5

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Linguística, do Departamento de Letras Vernáculas da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Linguística.

Área de concentração: Descrição e Análise Linguística.

Orientador: Leonel Figueiredo de Alencar Araripe.

**FORTALEZA**

**2013**

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências Humanas

A567a Andrade, Katiuscia de Moraes.
Astrolábio : um corpus de redações escolares do Ceará anotado multidimensionalmente conforme a TEI P5 / Katiuscia de Moraes Andrade. – 2013.
133 f. : il. color., enc. ; 30 cm.

Dissertação(mestrado) – Universidade Federal do Ceará, Centro de Humanidades, Departamento de Letras Vernáculas, Programa de Pós-Graduação em Linguística, Fortaleza, 2013.
Área de Concentração: Descrição e análise linguística.
Orientação: Prof. Dr. Leonel Figueiredo de Alencar Araripe

1.Linguística – Processamento de dados. 2.Linguística de corpus. 3.Língua portuguesa – Correção de textos. 4.Prosa escolar brasileira – Ceará. 5.Língua portuguesa – Composição e exercícios. I.Título.

CDD 410.285

**KATIUSCIA DE MORAES ANDRADE**

# ASTROLÁBIO: UM CORPUS DE REDAÇÕES ESCOLARES DO CEARÁ ANOTADO MULTIDIMENSIONALMENTE CONFORME A TEI P5

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Linguística, do Departamento de Letras Vernáculas da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Linguística.

Área de concentração: Descrição e Análise Linguística.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

_____

Prof. Dr. Leonel Figueiredo de Alencar Araripe (Orientador)

Universidade Federal do Ceará (UFC)

_____

Prof[a]. Dr[a]. Vládia Célia Monteiro Pinheiro

Universidade de Fortaleza (UNIFOR)

_____

Prof[a]. Dr[a]. Rosemeire Selma Monteiro Plantin

Universidade Federal do Ceará (UFC)

Para Mardônio França – meu amor, meu cúmplice.

## AGRADECIMENTOS

A natureza transdisciplinar da Linguística Computacional encantou-me por ter-me possibilitado o contato com pessoas de diversas áreas de conhecimento. Desta forma, o trabalho em conjunto, mais que uma opção, foi uma necessidade. Assim, devo os meus profundos agradecimentos a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho. A lista de nomes que menciono aqui, certamente, não é exaustiva. Contudo, não poderia deixar de agradecer explicitamente a algumas pessoas e entidades, devido à relevância que tiveram para a conclusão desta dissertação.

Em primeiro lugar, queria agradecer à Espiritualidade Maior, que me renovou quando todas as forças e esperanças já se tinham esvaído. Em segundo lugar, ao meu esposo, Mardônio França, que se debruçou incansavelmente sobre o meu projeto, trazendo seus valiosos conhecimentos da Computação, além de todo o apoio emocional e o encorajamento que me foi dado.

Do mesmo modo, devo graças ao meu orientador, Professor Dr. Leonel F. de Alencar, por sua coragem e ousadia de aceitar-me como sua orientanda e acreditar na minha capacidade, mesmo sabendo da minha inexperiência, até aquele momento, com a Linguística de *Corpus* e Linguística Computacional. Sinto-me honrada por tê-lo como orientador, tanto por sua peculiar competência como pela máxima dedicação com que ele conduziu o meu projeto.

Sinto-me igualmente grata aos colegas de COMPLIN: Gezenira, Karine, Andréa, Tiago, Ednardo, Hélio, Cid Ivan e Davis, por terem compartilhado seus conhecimentos e experiências, além da amizade.

Gostaria de agradecer às Professoras Dra. Áurea Zavam e Dra. Rosemeire Selma Monteiro-Plantin, que me acompanham desde a qualificação do meu projeto, e à Professora Dra. Vládia Célia Monteiro Pinheiro e Dra. Vera Lúcia Santiago Araújo, por terem aceitado o desafio de participar da minha banca de defesa.

Também não posso esquecer-me da Professora Dra. Sandra Maia F. de Vasconcelos, que apesar de ter-se afastado enquanto orientadora, não desligou os vínculos de amizade, atenção e ensinamentos. No mesmo sentido, agradeço aos colegas e ex-colegas do grupo GELDA.

Agradeço ainda aos demais professores, colegas e funcionários do PPGL, em especial ao Eduardo Xavier, sempre muito gentil diante de minhas solicitações.

Em tempo, agradeço aos amigos e parceiros que integraram o projeto Rota das

Especiarias, cujas oficinas deram origem ao *corpus* Astrolábio, além de outros amigos e familiares que tanto me encorajaram nesta atividade.

Agradeço, finalmente, à CAPES, que me deu o apoio financeiro, imprescindível para que eu pudesse dedicar-me a esta pesquisa.

“Os computadores são incrivelmente rápidos, precisos e burros; os homens são incrivelmente lentos, imprecisos e brilhantes; juntos, seus poderes ultrapassam os limites da imaginação.” (Albert Einstein)

## RESUMO

Astrolábio é um *corpus* compilado, anotado multidimensionalmente e disponibilizado eletronicamente sob a licença *Creative Commons Attribution-NonCommercial 3.0 Unported.* Trata-se de um *corpus,* em Português brasileiro, que emprega avançadas tecnologias para o processamento de texto e anotação de *corpora*. Astrolábio possui anotação multidimensional baseada na codificação TEI P5, que prescreve o uso metalinguagem XML. Com o uso dessa codificação, preservaram-se características essenciais da estrutura e do conteúdo dos documentos anotados, tornando a transcrição o mais fiel possível ao original. Por meio do emprego da *tag <choice>,* foi possível reunir, em um mesmo arquivo, fenômenos de variação linguística, erros ortográficos e de pontuação, bem como as respectivas formas corrigidas e normalizadas, além de possibilitar a visualização de termos que foram acrescidos ou suprimidos. Para a integração automática dos vários níveis de anotação, utilizou-se o Astro, um *software* que utiliza diversos módulos em Python para o Processamento da Linguagem Natural (PLN), como o Aelius e o Enchant. Na etiquetagem morfossintática, utilizou-se o pacote Aelius, que, por sua vez, recorre à biblioteca *Natural Language Toolkit* (NLTK). O etiquetador escolhido, dentro do Aelius, foi o AeliusHunposMacMorpho, criado a partir do etiquetador Hunpos, treinado no *corpus* de textos jornalísticos MAC-Morpho. Efetivou-se a correção ortográfica com o Enchant, uma vasta biblioteca com API (*Application Programming Interface*) em linguagem C e C++. Os textos que compõem esse *corpus* foram produzidos durante as oficinas de produção textual da segunda edição do projeto Rota das Especiarias, realizadas no primeiro semestre de 2012, com alunos de escolas públicas das cidades cearenses de Camocim, Barroquinha e Jijoca de Jericoacoara. Até o presente momento da construção do Astrolábio, encontram-se concluídas as etapas de seleção, escanerização, compilação e a primeira fase de anotação automática dos textos por meio do Astro. O *corpus* Astrolábio já se encontra parcialmente disponível no sítio eletrônico Rota das Especiarias (www.rotadasespeciarias.art.br). Em breve, será submetido ao repositório eletrônico *University of Oxford Text Archive* (OTA). Pelo que se observou do panorama de *corpora* do Português, inexiste um *corpus,* em Português Brasileiro, com esse nível de anotação.



**Palavras-chave:** Linguística Computacional. Linguística de Corpus. TEI P5. NLTK. Correção automática de textos. Etiquetagem morfossintática.

**ABSTRACT**

Astrolábio is a compiled *corpus*, with multidimensional annotation, and shared under Creative Commons Attribution-NonCommercial 3.0 Unported licence. It is a *corpus*, in Brazilian Portuguese, that uses advanced technologies to text processing and *corpora* annotation. Astrolábio has multidimensional annotation based on TEI P5 guidelines, that prescribes XML metalanguage. Through these guidelines, essential structures from the annotated documents were preserved, keeping the transcription as reliable as possible to the original. By using tag <*choice*>, it enabled keep, in the same archive, linguistic variation phenomena, orthographic and punctuation errors, as the respectives corrected and normalized forms, and also makes possible the visualization of added and deleted terms. To automatize the integration of many levels of annotation, Astro was used, it is a software that works with several Python modules to Natural Language Processing (NLP), including Aelius and Enchant. To POS tagging, Aelius, a package that uses Natural Language Toolkit (NLTK) libraries, was utilized. From Aelius, AeliusHunPosMacMorpho was chosen, it is a tagger based on HunPos and trained by MAC-Morpho, a corpus composed of journalistic texts. The 9spell checking was made by Enchant, a large library with API (Application Programming Interface) in C and C++ languages. The tagger chosen from inside training *corpus* MacMorpho,. Astrolábio's texts were produced during text production workshops from the second edition of Rota das Especiarias project, realized on first semester of 2012, with public school students from Camocim, Barroquinha e Jijoca de Jericoacoara, cities located in Ceará. Until this moment of Astrolábio's creation, concluded stages are texts selection, compilation and the first step of automatic annotation by Astro. Astrolábio *corpus* is already partially avaiable at Rota das Especiarias' website (www.rotadasespeciarias.art.br). Soon, the corpus will be submitted to *University of Oxford Text Archive* (OTA). As we observed from *corpora* scene of Portuguese, there's no *corpus*, in Brazilian Portuguese, with this level of annotation.

**Keywords:** Computacional linguistics. Corpus linguistics. TEI P5. NLTK. Spell checking. POS tagging.

## LISTA DE ESQUEMAS

## LISTA DE FIGURAS

## LISTA DE QUADROS

**LISTA DE TABELAS**

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| CLUL | Centro de Linguística da Universidade de Lisboa |
| DiK | Dolmetschen im Krankenhaus |
| DTD | Document Type Definition |
| EEM | Escola de Ensino Médio |
| FFLCH | Faculdade de Filosofia, Letras e Ciências Humanas |
| HTML | HyperText Markup Language |
| IEL-UNICAMP | Instituto de estudos da Linguagem da Universidade Estadual de Campinas |
| IME | Instituto de Matemática e Estatística |
| MBT | Memory-Based Tagger-Generator and Tagger |
| NLP | Natural Language Processing |
| NLTK | Natural Language Processing Toolkit |
| NILC | Núcleo Interinstitucional de Lingüística Computacional (USP) |
| OCR | Optical Character Recognition |
| OTA | University of Oxford Text Archive |
| PDF | Portable Document Format |
| POS | Part-of-speech |
| SGML | Standard Generalized Markup Language |
| TBCHP | Tycho Brahe Parsed Corpus of Historical Portuguese |
| TEI | Text Encoding Initiative |
| XCES | Corpus Encoding Standard for XML |
| XML | Extensible Markup Language |

**SUMÁRIO**

# 1 INTRODUÇÃO

A compilação e anotação de corpus constitui o tema dessa pesquisa. Assim, o objetivo foi compilar, anotar multidimensionalmente e disponibilizar um corpus, em Português brasileiro, utilizando a Text Encoding Iniciative (TEI P5), um sistema de codificação que se vale da metalinguagem Extensible Markup Language (XML) para realçar, explicar ou delimitar certos aspectos de um texto. Também utilizamos o software Astro, criado especialmente para atender às necessidades deste projeto.

O corpus produzido recebeu o nome de 'Astrolábio', instrumento de orientação que foi muito utilizado no período das Grandes Navegações, pois está no mesmo campo semântico de 'Rota das Especiarias', projeto que lhe deu origem. Os textos que integram esse corpus[1] foram produzidos durante as oficinas de produção textual da segunda edição do projeto Rota das Especiarias[2], realizadas no primeiro semestre de 2012, com alunos de escolas públicas das cidades cearenses de Camocim, Barroquinha e Jijoca de Jericoacoara.

Acreditamos que as experimentações em torno da anotação multidimensional[3], bem como a integração entre os diferentes tipos de anotações trouxeram consideráveis contribuições para o desenvolvimento de ferramentas e estudos em Linguística Computacional. Assim, acreditamos trazer significativas contribuições para o desenvolvimento de ferramentas de tecnologia da linguagem natural, como, por exemplo, etiquetadores morfossintáticos mais robustos, capazes de lidar com a linguagem não padrão, e corretores ortográficos. Além disso, pelo que se observou do panorama de *corpora* do Português, inexiste um *corpus*, em Português Brasileiro, com o nível de anotação do Astrolábio.

Até o momento, na construção do *corpus* Astrolábio, encontram-se concluídas as etapas de seleção, escanerização, compilação e a primeira fase de anotação automática dos textos por meio do Astro. Entretanto, parte do corpus já se encontra parcialmente disponível, sob a licença *Creative Commons Attribution-NonCommercial 3.0 Unported*[4], *no* sítio

---

[1]São textos manuscritos que foram transcritos por uma equipe de quatro pessoas. Maiores informações, ver capítulo 6.

[2]www.rotadasespeciarias.art.br . Maiores informações ver capítulo 6.

[3]O termo 'multidimensional' foi uma sugestão do orientador, Prof. Leonel F. de Alencar, pelas razões que apresentamos no capítulo 2.

[4]Essa licença permite aos usuários distribuir, copiar, retransmitir o conteúdo, bem como criar obras derivadas, desde que citada a fonte e não seja usado para fins comerciais.

eletrônico do projeto Rota das Especiarias[5]. Ao ser finalizado, também será submetido ao repositório eletrônico *University of Oxford Text Archive*[6] (OTA).

Por outro lado, não tivemos o intuito de apresentar um estudo crítico acerca do sistema de anotação da TEI P5 nem do *tagset* (conjunto de etiquetas) do *corpus* de treino MAC-Morpho, que utilizamos no *corpus* Astrolábio. Limitamo-nos a empregar as etiquetas, adequando os nossos casos concretos de fenômenos linguísticos aos referidos sistemas e explicando-lhes o funcionamento, conforme necessário.

Apesar de automatizarmos algumas etapas, a atividade de construção de um *corpus* está distante de ser uma tarefa simples. Isto porque, além do conhecimento das ferramentas computacionais empregadas, requer uma árdua investigação linguística, uma vez que a atividade de inserção das *tags* XML é um trabalho de categorização, não arbitrário, e requer o domínio de conceitos linguísticos. A título de exemplo, podemos citar um fenômeno encontrado em um texto de *F.I.S*, de 16 anos, aluna do Liceu de Camocim[7]:

Esquema 1 – Exemplo de anotação de sentença com *tags* <*choice*>

```
<p>
<s> Foi aqui o lugar da minha
    <choice>
        <sic> umilhação </sic>
        <corr> humilhação </corr>
    </choice>
    <choice>
        <add> . </sic>
    </choice>
</s>
</p>
```

Fonte: Elaborado pela autora.

O esquema acima traz um exemplo de anotação de um pequeno excerto do texto, com anotações XML, conforme a TEI P5. As *tags* <*p*> e </*p*> demarcam o parágrafo e as *tags* <*s*> e </*s*>, por sua vez, delimita a sentença.

[5]www.rotadasespeciarias.art.br
[6]http://ota.ahds.ac.uk/
[7]Aqui, temos um exemplo onde existem apenas etiquetas XML, conforme a TEI P5, anotadas manualmente.

As *tags* <*choice*> servem para apontar uma escolha dentro do texto: ou se escolhe a forma original ou a forma modificada. O erro de ortografia *umilhação* foi inserido dentro das *tags* <*sic*></*sic*>, que, por sua vez, estão dentro das *tags* <*choice*></*choice*>. A forma correta *humilhação*, por sua vez, foi inserida dentro da *tag* <*corr*>, dentro da mesma *tag* <*choice*>. O sinal de pontuação, inexistente no texto original, teve que ser inserido por nós, por meio da tag <*add*>, tornando possível a delimitação da sentença.

A tomada de decisões tem um papel crucial na construção do *corpus*, posto que, muitas vezes, o resultado de uma ação pressupõe o resultado de uma ação anterior. Assim, um erro em uma fase inicial levará a uma cadeia sucessiva de erros. Por exemplo, a etiquetagem morfossintática não é resultado de uma ação isolada, mas deve obedecer a uma *pipeline*, ou seja, uma espécie de "cano condutor", responsável por guiar todo o processo. Assim, a primeira ação a ser tomada pelo etiquetador é a quebra do texto em sentenças, em seguida, ocorre a toquenização, que é a quebra das sentenças em *tokens* (palavras). A etiquetagem morfossintática, propriamente dita, é a terceira etapa dessa cadeia e leva em consideração o resultado das duas etapas anteriores.

Como se pode perceber, para a anotação de um pequeno texto, inúmeras questões linguísticas já foram suscitadas. A língua em uso leva-nos a situações não previstas em regras, o que torna as investigações do Processamento da Linguagem Natural (PLN) extremamente desafiadoras.

Nosso trabalho de construção do *corpus* também não se limitou ao uso de ferramentas já consagradas. Buscamos as tecnologias de alto padrão existentes para o processamento de *corpora*. Além disso, realizamos experimentações, a fim de otimizar a etiquetagem automática, bem como a integração de diferentes níveis de anotação e a biblioteca de verificação ortográfica PyEnchant[8], baseada na biblioteca Enchant. Estas experimentações levaram ao desenvolvimento do *software* Astro que, além de integrar a etiquetagem morfossintática, por meio do Aelius, e as anotações conforme a TEI P5, também utiliza para corrigir erros ortográficos e insere *tags* <*choice*> quando encontra erros detectados automaticamente.

A motivação inicial para essa pesquisa surgiu a partir de nossa vivência no universo editorial. Quando recebemos os originais de um autor, para a criação de um livro, na maioria dos casos, o arquivo foi salvo em algum formato existente nos editores de texto (*.txt*,

[8]PyEnchat está disponível para download em http://pythonhosted.org//pyenchant/ .

*.doc*, *.docx* etc). Isso permite-nos, facilmente, copiar e colar o conteúdo em um novo arquivo, aberto em um programa de diagramação (geralmente, usamos o *Indesign*, da *Adobe*, mas existem muitos outros como o *Corel* ou o *PageMaker*).

Porém, há situações em que o autor já não mais dispõe do arquivo salvo em um editor de texto; são os casos, por exemplo, em que o livro foi editado há muito tempo. Nestas circunstâncias, geralmente, só dispomos do livro impresso, dos manuscritos ou de arquivos salvos em formatos não-editáveis, como os famosos arquivos do tipo *Portable Document Format* (PDF). Então, o que necessitamos fazer para utilizar o texto?

Talvez, a primeira solução que nos apareça seja aplicar a técnica da escanerização (digitalização das imagens). De fato, é uma excelente saída para o armazenamento de informações. Contudo, recaímos no mesmo problema anterior: os arquivos salvos em formatos não-editáveis. Isso ocorre porque, na realidade, o *scanner* "fotografa" o material e converte o arquivo gerado em uma imagem e não no formato de texto editável.

Contudo, os *scanners* dispõem de um mecanismo denominado de *Optical Caracter Recognition* (OCR) ou Reconhecimento Ótico de Caracteres, capaz de reconhecer os caracteres do texto e convertê-lo em um texto editável. O que ocorre, no entanto, é que essa ferramenta, até onde sabemos, não possui um bom desempenho para a língua portuguesa, uma vez que alguns caracteres, próprios da língua, como o 'ç', não são reconhecidos corretamente e o trabalho de correção torna-se tão dispendioso que, muitas vezes, pode tornar ineficaz o procedimento. Nesse sentido, encontramos um recente estudo que verificou a acurácia do OCR em textos portugueses do século XVII:

> "[...] mostramos que um software de OCR de ponta apresenta um índice de acertos na faixa de apenas 57% a 77% em impressos portugueses do século XVII, por exemplo. Já para os documentos manuscritos, simplesmente não há tecnologia de reconhecimento disponível. Assim, as pesquisas linguísticas precisam lançar mão do trabalho de transcrição dos documentos a serem estudados" (PAIXÃO;KEPLER;FARIAS, 2009, online, grifo nosso).

Acerca do tema, discorrem os editores do *Tycho Brahe Parsed Corpus of Historical Portuguese*[9] (TBCHP):

> A forma mais fiel de se reproduzir um texto antigo no meio digital é sem dúvida o *fac-simile*. Entretanto, para pesquisas linguísticas é necessário trabalhar o texto como sequências de caracteres (não como imagens).
> A solução de transposição automática da imagem em texto via programas de OCR não é uma opção satisfatória por enquanto, uma vez que as características

[9]Para maiores informações sobre o *corpus* Tycho-Brahe, ver a seção '4.2.' deste trabalho.

> tipográficas dos textos mais antigos são desafiantes para os programas de OCR disponíveis.
> No trabalho de preparação de textos para o Corpus Tycho Brahe, a saída de curto prazo escolhida foi a transcrição dos originais, enquanto se pesquisam formas de adequação do reconhecimento automático (tanto via OCRs aprimorados como via sistemas de correção posterior). (GALVES; FARIA, 2010, *online*).

De acordo com o que foi destacado acima, inexiste tecnologia para o reconhecimento de caracteres em documentos manuscritos, em Português brasileiro, que são os documentos originais integrantes do Astrolábio. Não nos cabe, todavia, investigar aqui o estado da arte do OCR para a língua portuguesa, mas simplesmente ilustrar os caminhos que nos trouxeram a este projeto de pesquisa.

Questões como essas levaram-nos a procurar, cada vez mais, soluções alternativas para facilitar o nosso trabalho de edição. Essa busca culminou com uma imediata empatia pela Linguística de Corpus e Linguística Computacional, devido à vastidão de possibilidades para criação e aperfeiçoamento de ferramentas para manipulação de texto. Assim, chegamos ao universo da *Text Enconding Initiative* (TEI).

Outro importante aspecto desse *corpus* não é de ordem linguística, e nem por isso é menos relevante, mas é de ordem social e política. A microrregião do Litoral Oeste de Camocim, onde foram realizadas as oficinas, engloba a cidade com o menor IDH do Ceará: Barroquinha[10].

Durante as oficinas, tivemos o cuidado de colher dados dos informantes (vide *Esquema 2*), permitindo-nos explorar as dimensões diatópicas, diastráticas e diafásicas, podendo embasar pesquisas na área de Sociolinguística. Variações diatópicas dizem respeito às divergências existentes entre os falares locais, regionais e, até, intercontinentais. Por sua vez, variações diastráticas correspondem às diferenças entre os níveis socioculturais e, finalmente, variações diafásicas significam oposições entre os tipos de modalidade expressiva: linguagem dos homens, linguagem das mulheres, língua falada, língua escrita etc. (CUNHA, CINTRA, 2008, p.03). A *tag* <*person*>, dentro do *corpus* Astrolábio, agrega informações acerca do sujeito (autor), que nos auxiliam na identificação de possíveis variações:

[10]Fonte: Ceará em Mapas - http://www2.ipece.ce.gov.br/atlas/capitulo3/31.htm

Esquema 2 – Representação de dados referentes aos informantes, utilizando a *tag* <*person*>

```
<person sex=" " age=" ">
        <birth when=" ">
                <date></date>
                <name type="place"></name>
        </birth>
        <residence></residence>
        <education>High School Student</education>
        <occupation></occupation>
 </person>
```

Fonte: Elaborado pela autora.

# 2 PANORAMA DE *CORPORA* DO PORTUGUÊS BRASILEIRO

Neste capítulo, selecionamos alguns corpora eletrônicos, segundo nosso critério de relevância. Assim, consideramos um ou mais dos seguintes requisitos: ter sido utilizado ou mencionado nos trabalhos de PLN que consultamos; ter anotação XML; ser destaque nos projetos de corpora que citamos aqui.

## 2.1 O que é um *corpus*

O propósito de construir um *corpus (*no sentido de conjunto de documentos), é bastante remoto, a exemplo do *corpus* Helenístico, definido por Alexandre, o Grande, e os *corpora* de citações da Bíblia, produzidos na Idade Média (BERBER SARDINHA, 2004). Os *corpora* eletrônicos, todavia, elaborados com critérios precisos e com a finalidade de embasarem pesquisas linguísticas, são bem mais recentes.

O primeiro *corpus* eletrônico de linguagem escrita de que temos notícia é o *Brown University Standard Corpus of Present-day American English*, de 1964, contendo um milhão de palavras (BERBER SARDINHA, 2004). Considerando os recursos computacionais da época, para elaborar um *corpus* composto por essa quantidade de palavras, foi necessário um trabalho colossal. Atualmente, já dispomos de *corpora* com uma quantidade significativamente maior que o *Brown Corpus:* o *Corpus de Referência do Português Contemporâneo (CRPC)*, por exemplo, contém aproximadamente 300 milhões de palavras[11].

Os corpora eletrônicos, além de servir de base para pesquisas linguísticas, também são de grande utilidade na construção e aprimoramento de ferramentas de PLN que utilizam aprendizagem de máquina como, por exemplo, *parsers* e analisadores morfossintáticos.

[11]http://www.clul.ul.pt/pt/recursos/183-reference-corpus-of-contemporary-portuguese-crpc

## 2.2 Critérios para a construção de um *corpus*

Para que um conjunto de textos seja considerado um *corpus*, é necessário que a escolha dos textos observe uma determinada metodologia. Berber Sardinha (2004) faz uma didática distinção entre conjuntos de dados que, apesar das semelhanças, não são sinônimos de *corpus*, são eles: arquivo, um conjunto de textos sem qualquer critério de organização e biblioteca eletrônica, uma coleção que obedece a alguns critérios de organização. De outro modo, o corpus, construído a partir de uma biblioteca eletrônica, deve seguir critérios explícitos e ter finalidades específicas. Ele ainda enumera os pré-requisitos para a construção de um corpus eletrônico: origem, propósito, composição, formatação, representatividade e tamanho.

Em relação à origem, temos que levar em consideração a autenticidade dos textos. Podemos dizer que uma coleção de textos é dotada de autenticidade quando foi produzida em linguagem natural, produzida por falantes nativos e não foi criada com o intuito de servir de base para uma pesquisa linguística.

A formatação, por sua vez, deve observar aspectos que tornem os textos legíveis pelo computador e, mais especificamente, precisam estar no formato correto de *input* para as ferramentas eletrônicas utilizadas.

Através da representatividade busca-se reproduzir, na totalidade, a língua de uma determinada comunidade linguística. Como essa situação é hipotética, procura-se incluir, ao máximo, os aspectos que possam representá-la. No caso de um corpus formado por obras de um determinado autor, por exemplo, para que ele seja representativo, é interessante incluir toda a obra do autor.

Segundo Bieber *et al.* (1998 *apud* ALUÍSIO; ALMEIDA, 2006), um *corpus* de língua deve ser organizado demograficamente. Como já dissemos, o *corpus* a ser anotado foi elaborado a partir de oficinas para alunos de escolas públicas, de três cidades de uma mesma microrregião do Ceará, com faixa etária e socioeconômica semelhantes. Assim, podemos dizer que o *corpus* em análise obedece aos critérios da amostragem e diversidade.

Um *corpus* balanceado, por sua vez, é um *corpus* equilibrado. Este é um importante aspecto a ser observado para justificar um aspecto da língua que se pretende analisar.

Acerca do tamanho, convém dizer que este critério não é relevante para que o corpus seja representativo. Berbet Sardinha (2004) considera três abordagens para a definição

do tamanho do corpus: impressionística, histórica e estatística.

Na primeira, a extensão é auferida a partir de conclusões de especialistas, a partir de suas atividade de construção e exploração de corpora. A nosso ver, esse critério não é muito interessante porque é muito subjetivo e há muita divergência entre opiniões desses profissionais. Na abordagem histórica, é feito um estudo comparativo dos corpora existentes até então, cujos resultados podem ser observados a seguir:

Tabela 1 – Classificação do tamanho de um *corpus* – abordagem histórica

| **Tamanho em palavras** | **Classificação** |
|---|---|
| < 80 mil | Pequeno |
| 80 a 250 mil | Pequeno-médio |
| 250 mil | Médio |
| 1 milhão a 10 milhões | Médio-grande |
| 10 milhões ou mais | Grande |

Fonte: SARDINHA, 2004.

Por último, a abordagem estatística, como o próprio nome sugere, propõe que sejam feitos cálculos matemáticos a fim de se ter uma noção do tamanho ideal de um corpus para que ele seja representativo.

## 2.3 Projeto floresta sintáti(c)tica

Floresta Sintá(c)tica consiste em um projeto para a construção de um conjunto de árvores com anotações sintáticas, *treebanks*, abrangendo as variantes do Português Europeu e Português Brasileiro. Dentre as utilidades de um treebank, podemos citar a avaliação e o treino de *parsers* e analisadores morfossintáticos, além de servir de suporte para pesquisas linguísticas, tais como estudos de sintaxe e investigações baseadas em corpora. A Floresta Sintá(c)tica é compreendida pelos seguintes subcorpora: Bosque, Selva Literária, Selva Falada, Selva Científica, Floresta Virgem, Amazônia e Floresta Virgem.

Bosque é um corpus revisado por linguistas e é composto por 9.368 frases extraídas de dois corpora de textos jornalísticos: CETEMPúblico (composto por textos retirados do Público, representando o Português Europeu) e o CETEMFolha (formado por textos da Folha de São Paulo, representando o Português Brasileiro). Bosque possui

anotações morfossintáticas e sintáticas, no formato de árvores deitadas, e foi anotado automaticamente pelo parser PALAVRAS, de autoria do dinamarquês Eckhard Bick (AFONSO, 2008). Bosque tem sido largamente utilizado no desenvolvimento de ferramentas de PLN e bastante citado na literatura sobre o assunto.

## 2.4 Projetos de *corpora* desenvolvidos pelo CLUL (Centro de Linguística da Universidade de Lisboa)

O CLUL é uma unidade, diretamente ligada à Faculdade de Letras da Universidade de Lisboa, que desenvolve investigações teóricas e experimentais em torno da gramática do Português[12]. Busca estabelecer um diálogo entre as mais diversas áreas do estudo da linguagem, como a filologia, a psicolinguística e a linguística computacional, e disponibilizar materiais e ferramentas. Atualmente, é constituído pelos seguintes grupos de investigação: Dialectologia e Diacronia; Laboratório de Psicolinguística; Anagrama (Análise Gramatical e *Corpora*); Labfon (Laboratório de Fonética); CLG (Computação do Conhecimento Léxico-Gramatical) e Filologia. A seguir, enumeramos e fazemos uma breve descrição dos projetos de *corpora* desenvolvidos pelo CLUL.

O CLUL é uma unidade, diretamente ligada à Faculdade de Letras da Universidade de Lisboa, que desenvolve investigações teóricas e experimentais em torno da gramática do Português[13]. Busca estabelecer um diálogo entre as mais diversas áreas do estudo da linguagem, como a filologia, a psicolinguística e a linguística computacional, e disponibilizar materiais e ferramentas. Atualmente, é constituído pelos seguintes grupos de investigação: Dialectologia e Diacronia; Laboratório de Psicolinguística; Anagrama (Análise Gramatical e *Corpora*); Labfon (Laboratório de Fonética); CLG (Computação do Conhecimento Léxico-Gramatical) e Filologia. A seguir, enumeramos e fazemos uma breve descrição dos projetos de *corpora* desenvolvidos pelo CLUL.

[12]De acordo com os objetivos apresentados no sítio eletrônico do CLUL, as diferentes formas do Português são apresentadas como "variantes", sendo todas elas objetos de seu interesse, bem como "as línguas crioulas de base lexical portuguesa".

[13]De acordo com os objetivos apresentados no sítio eletrônico do CLUL, as diferentes formas do Português são apresentadas como "variantes", sendo todas elas objetos de seu interesse, bem como "as línguas crioulas de base lexical portuguesa".

### *2.4.1 CRPC* – Corpus *de Referência*[14] *do Português Contemporâneo*

O CRPC é um *corpus* do Português, eminentemente Europeu, mas que também comporta variantes do Brasil, Angola, Cabo Verde, Guiné-Bissau, Moçambique, São Tomé e Príncipe, Goa, Macau e Timor-Leste. É composto por 311,4 milhões de palavras, extraídas de textos escritos, em sua grande maioria, e transcições de registros orais. Os textos que o constituem são originários do período compreendido entre a segunda metade do século XIX e 2006 e são de gêneros diversificados: jornalísticos, literários, técnicos, científicos etc. Também possui um *subcorpus* oral, formado por diálogos, conversas, telefonemas etc., extraídos do *Corpus Português Fundamental* (PF)[15].

Os textos escritos passaram por um processo de tokenização automática por meio do LX-Tagger. Em seguida, receberam etiquetas morfossintáticas por meio do etiquetador MBT (*Memory-Based Tagger-Generator and Tagger*), treinado em uma versão adaptada do *corpus* CINTIL.

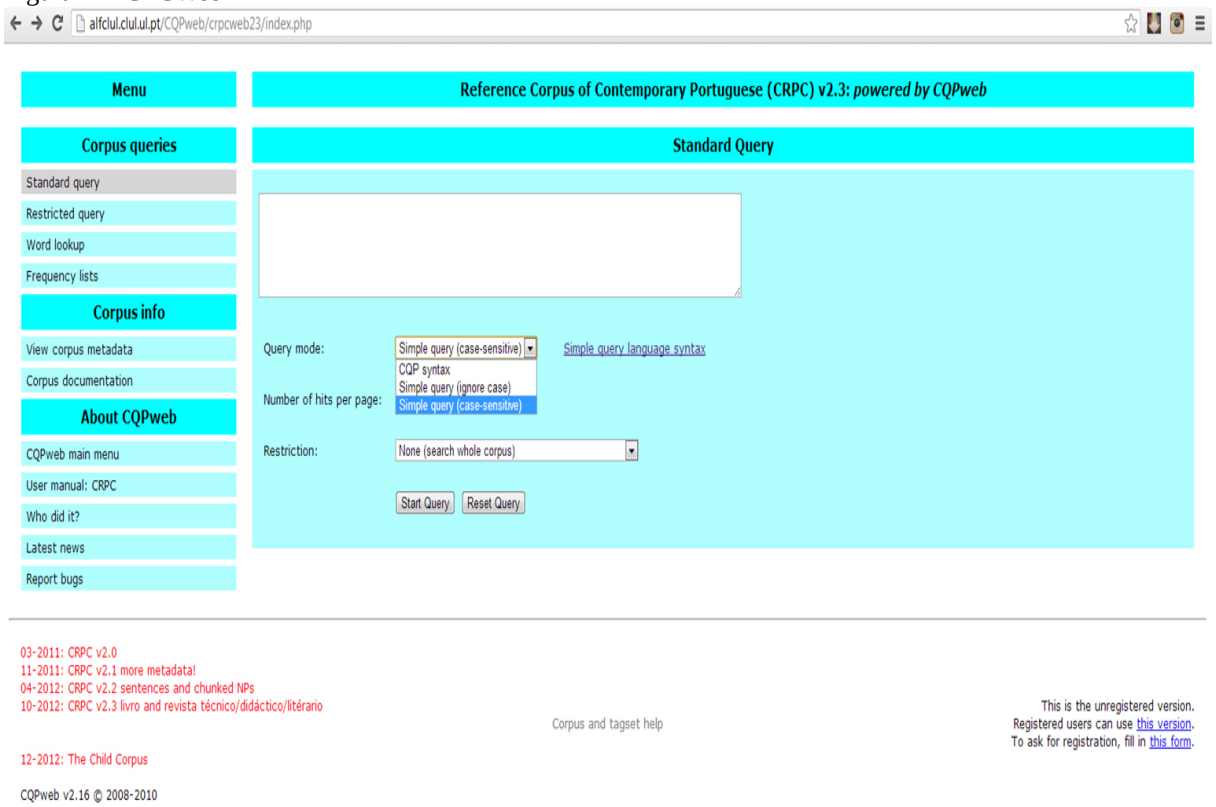
O *corpus* está disponível para pesquisas *online* e, para ter acesso, não é necessário registrar-se. Porém, o registro, além de ser gratuito, permite ao usuário usufruir de algumas exclusividades, como a criação de *subcorpora*. A ferramenta utilizada para a pesquisa *online*, o *CPCWeb – Corpus Query Processor,* permite realizar consultas avançadas, tais como: realizar a busca em todo o *corpus* ou restringir a pesquisa a um determinado gênero e/ou país; verificar as diferentes ocorrências de palavras etc. A seguir, algumas das funcionalidades do *CPCWeb*:

---

[14]Segundo Berber Sardinha (2004), o *corpus* de referência também recebe a denominação de *corpus* de controle e tem o propósito de servir de contraste com o *corpus* de estudo. A partir daí, podemos obter *keywords* (palavras-chave), através da comparação da frequência de palavras. Os autores do CRPC, por sua vez, consideram-no um *corpus* de referência porque os textos escritos, antes de integrarem o *corpus*, foram submetidos a um processo de amostragem (sítio eletrônico).

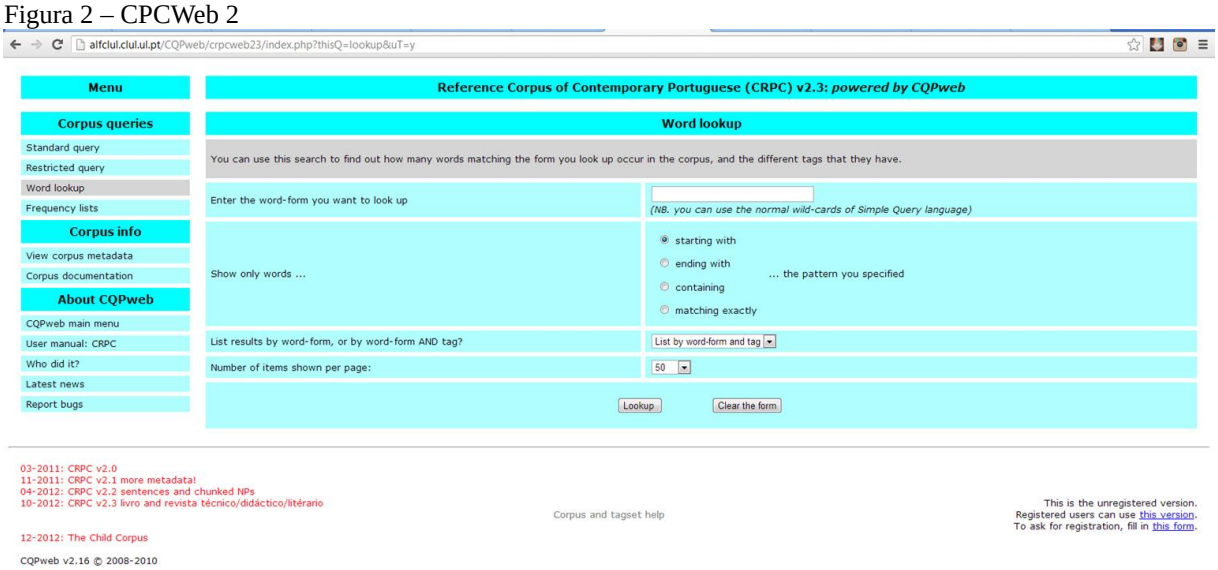
[15]Para maiores informações, consultar a sessão *4.1.3* deste capítulo.

Figura 1 – CPCWeb 1




Fonte: CPCWeb.

Figura 2 – CPCWeb 2




Fonte: CPCWeb.

### *2.4.2 CORDIAL-SIN* – Corpus *Dialectal para o Estudo da Sintaxe*

O CORDIAL-SIN é composto por transcrições de registros sonoros[16], tendo por objetivo o estudo da sintaxe dialetal do Português europeu. De acordo com a última atualização do sítio eletrônico[17], o *corpus* possui 600.000 palavras. Os arquivos da versão normalizada estão codificados no formato ASCII (consultar seção sobre Unicode).

As transcrições, bem como a anotação morfossintática (com algumas revisões/ampliações), seguiram o mesmo manual de transcrições desenvolvido para o *corpus* Tycho-Brahe. Além de etiquetas morfossintáticas, o CORDIAL-SIN também possui anotação sintática, seguindo os parâmetros adotados pelo *Penn-Helsinki Parsed Corpus of Middle English*. O formato deste último sistema de anotação permite ao CORDIAL-SIN a compatibilidade com a ferramenta para pesquisa linguística: *CorpusSearch2*.

O *corpus* está disponível em quatro versões: transcrição conservadora; transcrição ortográfica normalizada; texto com anotação morfossintáctica (anotação por palavra); texto com anotação sintáctica (anotação por frase), conforme exemplificado a seguir:

Figura 3 – Exemplo de transcrição conservadora – CORDIAL-SIN

INF Nós dantes, nestas redes, era raro o dia que {PHlnũ=não} se pegava um, dois lavagantes ou três.

Agora {PHlnũ=não} há. {PHlnũ=Não} há. Vai acontecer {CTlku'ma=como à} lagosta, aqui ao norte.

*INQ1 Desaparece tudo…*

INF Aqui era o mar da lagosta… Desapareceu (…). {PHlvi'ɛrũ=Vieram} os franceses,

{PHlvi'ɛrũ=vieram} os espanhóis {pp}, quando {PHlpu'diɐ̃w̃=podiam} aqui trabalhar… Desapareceu.

E o{fp} lavagante já {PHlnũ=não} há, também. É raro se pilhar um. Quando é um, é uma festa.

*INQ2 É por causa disso …*

Fonte: http://www.clul.ul.pt/pt/recursos/212-cordial-sin-syntax-oriented-corpus-of-portuguese-dialects.

[16]As transcrições são compostas por trechos de discursos livres e semi-dirigidos, obtidos a partir de mais de 4.500 horas de gravações, realizadas em mais de 200 locais de Portugal. As gravações foram coletadas a partir de outros projetos desenvolvidos pelo CLUL.

[17]Última atualização em Seg, 02 de Julho de 2012 12:18

Figura 4 – Exemplo de transcrição normalizada – CORDIAL-SIN

INF Nós dantes, nestas redes, era raro o dia que não se pegava um, dois lavagantes ou três. Agora não há. Não há. Vai acontecer como à lagosta, aqui ao norte.

*INQ1 Desaparece tudo...*

INF Aqui era o mar da lagosta… Desapareceu (…). Vieram os franceses, vieram os espanhóis, quando podiam aqui trabalhar… Desapareceu. E o lavagante já não há, também. É raro se pilhar um. Quando é um, é uma festa.

*INQ2 É por causa disso...*

Fonte: http://www.clul.ul.pt/pt/recursos/212-cordial-sin-syntax-oriented-corpus-of-portuguese-dialects.

Em ambas as transcrições, cada turno inicia-se com um parágrafo e é precedido pela indicação do enunciador: *INF, INQ1, INQ2.* Porém, somente na transcrição conservadora podemos visualizar onde há sobreposição na fala dos enunciadores, observando o traço sublinhado dos enunciados. Também nos é permitido visualizar variantes fonéticas e morfofonológicas (*{PH|nɐ̃=não}, {PH|vi'ɛrũ=vieram}*), contrações (*{CT|ku'ma=como à}*) etc. Na transcrição normalizada, como o próprio nome sugere, só temos acesso à fala normalizada de acordo com a ortografia oficial portuguesa[18]. A seguir, vemos um exemplo de texto com anotação por palavras:

<header> VPA01 </header>

<inf> INF </inf> Nós/PRO dantes/ADV ,/, nestas/P+D-F-P redes/N-P ,/, era/SR-D-3S raro/ADJ o/D dia/N que/WPRO não/NEG se/SE pegava/VB-D-3S um/NUM ,/, dois/NUM lavagantes/N-P ou/CONJ três/NUM ./.
Agora/ADV não/NEG há/HV-P-3S ./.
Não/NEG há/HV-P-3S ./.
Vai/VB-P-3S acontecer/VB como/CONJS à/P+D-F lagosta/N ,/, aqui/ADV ao/P+D norte/N ./.
<inq> INQ1 Desaparece tudo... </inq>
<inf> INF </inf> Aqui/ADV era/SR-D-3S o/D mar/N da/P+D-F lagosta/N .../.
Desapareceu/VB-D-3S <break> (...) </break> ./.
Vieram/VB-D-3P os/D-P franceses/N-P ,/, vieram/VB-D-3P os/D-P espanhóis/N-P ,/, quando/WADV podiam/VB-D-3P aqui/ADV trabalhar/VB .../.
Desapareceu/VB-D-3S ./.
E/CONJ o/D lavagante/N já/FP não/NEG há/HV-P-3S ,/, também/ADV ./.
É/SR-P-3S raro/ADJ se/SE pilhar/VB um/D-UM ./.
Quando/WADV é/SR-P-3S um/D-UM ,/, é/SR-P-3S uma/D-UM-F festa/N ./.
<inq> INQ2 É por causa disso... </inq>

Essa forma de anotação recorre a algumas etiquetas XML para descrever

[18]Para maiores informações, consultar o manual de transcrição do CORDIAL-SIN: http://www.clul.ul.pt/sectores/variacao/cordialsin/manual_normas.pdf

elementos da estrutura geral do documento, como o cabeçalho (<*header*>), e elementos relacionados à estrutura do discurso, como informante (<*inf*>), inquiridor (<*inq*>) e pausa (<*break*>). No exemplo acima, também podemos visualizar as etiquetas morfossintáticas[19], em letras maiúsculas, ao lado direito de cada palavra: *Nós/PRO, dantes/ADV* etc.

A seguir, um exemplo de anotação sintática do corpus CORDIAL-SIN:

```
(IP-MAT (NP-SBJ (D-F-P as)
            (N-P portas))
        (VB-P-3P caem)
        (PP (P para)
            (NP (D-F a)
                (N água)))
        (, ,))  [VPA05]

(IP-MAT (CONJ e)
        (NP-SBJ (D o)
                (N barco))
        (VB-P-3S vai)
        (IP-GER (VB-G botando)
        (NP-ACC (D-F a)
                (N rede)))
        (. .)) [VPA05]
```

### *2.4.3 PF* – Corpus *Português Fundamental*

O PF é um *corpus* de, aproximadamente, um milhão de palavras e também funciona como *subcorpora* do CRPC. É composto por outros dois *subcorpora*: o *Corpus de Frequência* e o *Corpus de Disponibilidade*, com o objetivo de embasar o conhecimento acerca do Português[20] mais frequente em situações cotidianas. O primeiro *subcorpora* foi elaborado a partir de uma seleção de excertos, transcritos a partir gravações de situações comunicativas espontâneas[21]. Já o segundo, constituido por um vocabulário mais específico, selecionado a partir de inquéritos direcionados, envolvendo temas menos recorrentes.

A amostra do *corpus* disponível para *download*[22] não possui anotação e dispõe apenas da versão normalizada da transcrição. Através do índice geral podemos acessar as

---

[19]Manual das etiquetas morfossintáticas do CORDIAL-SIN: http://www.clul.ul.pt/sectores/variacao/cordialsin/manual_anotacao_morfologica.pdf

[20]Português europeu.

[21]O grupo de falantes é bastante diversificado, sendo composto por pessoas de diversas faixas etárias, níveis de escolaridade etc.

[22]http://www.clul.ul.pt/english/sectores/linguistica_de_corpus/corpus_oral_pf_publicado.zip

entrevistas com os metadados dos falantes e consultar as tabelas com os códigos utilizados:

Figura 5 – RCPC

**CRPC sub-corpus oral espontâneo**

Tabela - Códigos dos Grupos Profissionais
Tabela - Códigos dos Inquiridores do PF
Tabela - Códigos dos Níveis de Instrução

| | | | |
|---|---|---|---|
| Entrevista nº 0022 | Entrevista nº 0328 | Entrevista nº 0770 | Entrevista nº 1072 |
| Entrevista nº 0029 | Entrevista nº 0340 | Entrevista nº 0776 | Entrevista nº 1082 |
| Entrevista nº 0031 | Entrevista nº 0356 | Entrevista nº 0784 | Entrevista nº 1093 |
| Entrevista nº 0041 | Entrevista nº 0376 | Entrevista nº 0785 | Entrevista nº 1098 |
| Entrevista nº 0053 | Entrevista nº 0377 | Entrevista nº 0793 | Entrevista nº 1146 |
| Entrevista nº 0067 | Entrevista nº 0426 | Entrevista nº 0795 | Entrevista nº 1166 |
| Entrevista nº 0075 | Entrevista nº 0455 | Entrevista nº 0796 | Entrevista nº 1201 |
| Entrevista nº 0079 | Entrevista nº 0457 | Entrevista nº 0816 | Entrevista nº 1202 |
| Entrevista nº 0090 | Entrevista nº 0467 | Entrevista nº 0832 | Entrevista nº 1212 |
| Entrevista nº 0091 | Entrevista nº 0476 | Entrevista nº 0836 | Entrevista nº 1230 |
| Entrevista nº 0093 | Entrevista nº 0479 | Entrevista nº 0837 | Entrevista nº 1232 |
| Entrevista nº 0106 | Entrevista nº 0482 | Entrevista nº 0839 | Entrevista nº 1238 |
| Entrevista nº 0108 | Entrevista nº 0485 | Entrevista nº 0854 | Entrevista nº 1242 |
| Entrevista nº 0109 | Entrevista nº 0502 | Entrevista nº 0863 | Entrevista nº 1248 |
| Entrevista nº 0111 | Entrevista nº 0523 | Entrevista nº 0864 | Entrevista nº 1250 |
| Entrevista nº 0122 | Entrevista nº 0528 | Entrevista nº 0883 | Entrevista nº 1253 |
| Entrevista nº 0129 | Entrevista nº 0529 | Entrevista nº 0885 | Entrevista nº 1261 |
| Entrevista nº 0134 | Entrevista nº 0555 | Entrevista nº 0886 | Entrevista nº 1264 |
| Entrevista nº 0135 | Entrevista nº 0560 | Entrevista nº 0894 | Entrevista nº 1292 |
| Entrevista nº 0147 | Entrevista nº 0564 | Entrevista nº 0913 | Entrevista nº 1293 |
| Entrevista nº 0149 | Entrevista nº 0598 | Entrevista nº 0920 | Entrevista nº 1296 |
| Entrevista nº 0164 | Entrevista nº 0618 | Entrevista nº 0956 | Entrevista nº 1308 |
| Entrevista nº 0170 | Entrevista nº 0622 | Entrevista nº 0962 | Entrevista nº 1315 |
| Entrevista nº 0173 | Entrevista nº 0633 | Entrevista nº 0964 | Entrevista nº 1325 |
| Entrevista nº 0184 | Entrevista nº 0653 | Entrevista nº 0965 | Entrevista nº 1333 |
| Entrevista nº 0187 | Entrevista nº 0657 | Entrevista nº 0977 | Entrevista nº 1336 |
| Entrevista nº 0194 | Entrevista nº 0673 | Entrevista nº 0985 | Entrevista nº 1338 |
| Entrevista nº 0218 | Entrevista nº 0682 | Entrevista nº 0990 | Entrevista nº 1358 |
| Entrevista nº 0221 | Entrevista nº 0710 | Entrevista nº 0994 | Entrevista nº 1367 |
| Entrevista nº 0232 | Entrevista nº 0725 | Entrevista nº 1009 | Entrevista nº 1377 |
| Entrevista nº 0236 | Entrevista nº 0757 | Entrevista nº 1016 | Entrevista nº 1378 |
| Entrevista nº 0248 | Entrevista nº 0763 | Entrevista nº 1020 | Entrevista nº 1383 |
| Entrevista nº 0262 | Entrevista nº 0764 | Entrevista nº 1042 | Entrevista nº 1392 |
| Entrevista nº 0290 | Entrevista nº 0765 | Entrevista nº 1055 | Entrevista nº 1394 |
| Entrevista nº 0308 | Entrevista nº 0769 | Entrevista nº 1071 | Entrevista nº 1396 |

Fonte: http://www.clul.ul.pt/en/research-teams/183-reference-corpus-of-contemporary-portuguese-crpc.

### *2.4.4* Corpus *do Português Falado – Variedades geográficas e sociais*

Esse *corpus* é composto de excertos de fala (situações comunicativas espontâneas) do Português de diversos países (dentre eles, Brasil, Angola e Timor-Leste), além de Portugal. O material produzido foi reunido em um CD-ROM que, além de servir para pesquisas linguísticas, tem como principal objetivo auxiliar o ensino do Português como língua estrangeira. Ao total, são 8h e 44min de gravação, além da transcrição normalizada das falas, sincronizadas com o áudio, contabilizando 91.966 palavras.

## 2.5 Projetos desenvolvidos pelo Lácio-Web

O Lácio-Web é um projeto desenvolvido na USP, através da parceria entre NILC (Núcleo Interinstitucional de Lingüística Computacional), IME (Instituto de Matemática e Estatística) e FFLCH (Faculdade de Filosofia, Letras e Ciências Humanas). O projeto tem por objetivo a divulgação e disponibilização, na *web*, de corpora do português brasileiro contemporâneo e ferramentas linguístico-computacionais.

O MAC-Morpho, desenvolvido pelo projeto, é um *corpus* de 1.167.183 *tokens* fechado, revisado manualmente e anotado pelo *parser* PALAVRAS, criado por Eckhard Bick. Apesar de ter sido anotado pelo PALAVRAS, o MAC-Morpho não manteve as etiquetas oriundas desse etiquetador e um novo tagset (vide ANEXO J) foi desenvolvido para o projeto Lácio-Web.

O córpus foi construído a partir de artigos da Folha de São Paulo do ano de 1994 possui e já serviu de corpus de treino para etiquetadores de alta relevância, como o MXPOST e o TreeTagger. Além do MAC-Morpho, existem mais cinco *corpora* desenvolvidos pelo Lácio-Web:

Quadro 1 – Projetos desenvolvidos pelo Lácio-Web

| **Nome** | **Tipo** | **Gênero** |
|---|---|---|
| Lácio-Ref | Aberto/sem anotação | Textos em português que respeitam a norma culta |
| Lácio-Dev | Aberto/ainda não disponível | Textos em português com redações de vestibular (não revisados em relação à norma culta). |
| Par-C | Aberto | Textos em português/inglês de 01 ano da revista Pesquisa Fapesp. |
| Comp-C | Aberto | Textos originais de conteúdo comparável em português/inglês. |
| Lácio-Sint | Fechado/etiquetado/ainda não disponível | Textos de diversos gêneros |

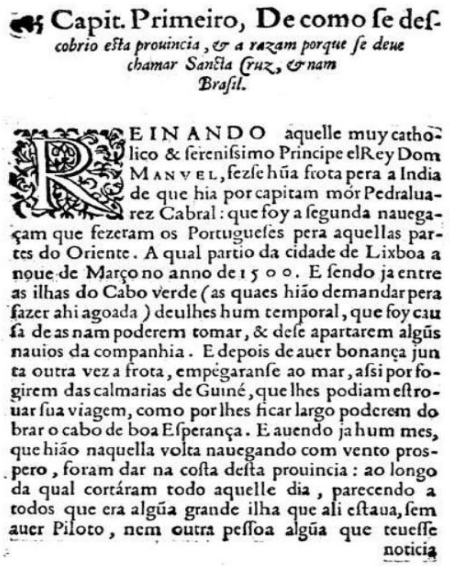
Fonte: http://www.nilc.icmc.usp.br/lacioweb/corpora.htm.

## 2.6 *Tycho-Brahe*

*Tycho Brahe Parsed Corpus of Historical Portuguese (*TBCHP*)*, é um corpus eletrônico anotado, desenvolvido pelo Instituto de Estudos da Linguagem da Universidade Estadual de Campinas (IEL-UNICAMP), formado por textos em português, de autores nascidos entre 1380 e 1845.

Por se tratarem de documentos históricos, cuja fonte primária são manuscritos, a fase de transcrição do TBCHP requereu uma acurada metodologia e valeu-se de um sistema de anotação em XML, especialmente elaborado para a situação. Vejamos a seguir exemplos dessa transcrição:

Figura 6 – *Cópia* digitalizada do documento original do TBCHP



Capit. Primeiro, De como ſe deſ-
cobrio eſta prouincia, & a razam porque ſe deue
chamar Santa Cruz, & nam
Brasil.

REINANDO aquelle muy catho-
lico & ſereniſsimo Principe elRey Dom
MANVEL, fezſe hũa frota pera a India
de que hia por capitam mór Pedralua-
rez Cabral: que foy a ſegunda nauega-
çam que fezeram os Portugueſes pera aquellas par-
tes do Oriente. A qual partio da cidade de Lixboa a
noue de Março no anno de 1500. E ſendo ja entre
as ilhas do Cabo verde (as quaes hião demandar pera
fazer ahi agoada) deulhes hum temporal, que foy cau
ſa de as nam poderem tomar, & deſe apartarem algũs
nauios da companhia. E depois de auer bonança jun
ta outra vez a frota, empẽgaranſe ao mar, aſsi por fo-
girem das calmarias de Guiné, que lhes podiam eſtro-
uar ſua viagem, como por lhes ficar largo poderem do
brar o cabo de boa Eſperança. E auendo ja hum mes,
que hião naquella volta nauegando com vento pros-
pero, foram dar na coſta deſta prouincia: ao longo
da qual cortáram todo aquelle dia, parecendo a
todos que era algũa grande ilha que ali eſtaua, ſem
auer Piloto, nem outra peſſoa algũa que teueſſe
noticia

Fonte: http://www.tycho.iel.unicamp.br/corpus/manual/prep/manual_frameset.html.

Figura 7 – Exemplo de transcrição diplomática de original TBCHP

*Capit. Primeiro, De como ſe deſ-*
*cobrio esta prouincia, & a razam porque ſe deue*
*chamar Sancta Cruz, e nam*
*Braſil.*

REINANDO aquelle muy catho-
lico & ſereniſsimo Principe elRey Dom
MANVEL, fezſe hũa frota pera a India
de que hia por capitam mór Pedralua-
rez Cabral: que foy a ſegunda nauega-
çam que fezeram os Portugueſes pera aquellas par-
tes do Oriente. A qual partio da cidade de Lixboa a
noue de Março no anno de 1500. E ſendo ja entre
as ilhas do Cabo verde (as quaes hião demandar pera
fazer ahi agoada) deulhes hum temporal, que foy cau
ſa de as nam poderem tomar, & deſe apartarem algũs
nauios da companhia. E depois de auer bonança jun
ta outra vez a frota, empegaranſe ao mar, aſsi por fo-
girem das calmarias de Guiné, que lhes podiam eſtro-
uar sua viagem, como por lhes ficar largo poderem do
brar o cabo de boa Eſperança. E auendo jahum mes,
quehião naquella volta nauegando com vento pros-
pero, foram dar na coſta deſta prouincia: ao longo
da qual cortáram todo aquelle dia, parecendo a
todos que era algũa grande ilha que ali eſtaua, ſem
auer Piloto, nem outra peſſoa algũa que teueſſe
noticia

Fonte:
http://www.tycho.iel.unicamp.br/corpus/manual/prep/manual_frameset.html.

Figura 8 – Exemplo de edição normalizada de original TBCHP

**Capítulo Primeiro, De como se descobriu esta província, e a razão porque se deve chamar Santa Cruz, e não Brasil.**

REINANDO aquele muito católico
e sereníssimo Príncipe el-Rei Dom
Manuel, fez-se uma frota para a Índia
de que ia por capitão mór Pedro Álvares
Cabral: que foi a segunda navegação
que fizeram os Portugueses para aquelas partes
do Oriente. A qual partiu da cidade de Lisboa a
nove de Março no ano de 1500. E sendo já entre
as ilhas do Cabo Verde (as quais iam demandar para
fazer aí aguada) deu-lhes um temporal, que foi causa
de as não poderem tomar, e de se apartarem alguns
navios da companhia. E depois de haver bonança junta
outra vez a frota, empegaram-se ao mar, assim por fugirem
das calmarias de Guiné, que lhes podiam estorvar
sua viagem, como por lhes ficar largo poderem dobrar
o cabo de Boa Esperança. E havendo já um mês,
que iam naquela volta navegando com vento próspero,
foram dar na costa desta província: ao longo
da qual cortaram todo aquele dia, parecendo a
todos que era alguma grande ilha que ali estava, sem
haver Piloto, nem outra pessoa alguma que tivesse
[notícia]

Fonte:
http://www.tycho.iel.unicamp.br/corpus/manual/prep/manual_frameset.html.

Figura 9 – Exemplo de texto do TBCHP com anotações em XML

```
<section_title>
 <ed_mark id="ed_mark_373">
    <ed id="e_373">Capítulo</ed>
    <or id="o_373">Capit.</or></ed_mark> Primeiro, De como
 <ed_mark id="ed_mark_374">
    <ed id="e_374">se</ed>
    <or id="o_374">ſe</or></ed_mark>
 <ed_mark id="v_375">
  <ed id="e_375">descobriu<nl/></ed>
  <or id="o_375">deſ-<nl/>cobrio</or></ed_mark> esta
 <ed_mark id="ed_mark_376">
  <ed id="e_376">província</ed>
  <or id="o_376">prouincia</or></ed_mark>,
 <ed_mark id="ed_mark_377">
  <ed id="e_377">e</ed>
  <or id="o_377">&</or></ed_mark> a
 <ed_mark id="ed_mark_378">
  <ed id="e_378">razão</ed>
  <or id="o_378">razam</or></ed_mark> porque
 <ed_mark id="ed_mark_379">
  <ed id="e_379">se</ed>
  <or id="o_379">ſe</or></ed_mark>
 <ed_mark id="ed_mark_380">
  <ed id="e_380">deve</ed>
  <or id="o_380">deue</or></ed_mark><nl/> chamar
 <ed_mark id="ed_mark_381">
  <ed id="e_381">Santa</ed>
  <or id="o_381">Sancta</or></ed_mark> Cruz, e
 <ed_mark id="ed_mark_382">
  <ed id="e_382">não</ed>
  <or id="o_382">nam</or></ed_mark><nl/>
 <ed_mark id="ed_mark_383">
  <ed id="e_383">Brasil</ed>
  <or id="o_383">Braſil</or></ed_mark><nl/>
</section_title>
```

Fonte: http://www.tycho.iel.unicamp.br/corpus/manual/prep/manual_frameset.html.

Figura 10 – Exemplo comparativos da transcrição, edição e anotação morfossintática do TBCHP

| transcrição | edição | anotação POS |
|---|---|---|
| **R**EINANDO aquelle muy catho- lico & ſereniſsimo Principe elRey Dom MANVEL , fezſe hũa frota pera a India de que hia por capitam mór Pedralua- rez Cabral | Reinando aquele mui católico e sereníssimo Príncipe el-Rei Dom Manuel, fez-se uma frota para a Índia de que ia por capitão mór Pedro Álvares Cabral | Reinando/VB-AN aquele/D mui/ADV católico/ADJ e/CONJ sereníssimo/ADJ-S Príncipe/NPR el-Rei/NPR Dom/NPR Manuel/NPR ,/, fez/VB-D se/SE uma/D-UM-F frota/N para/P a/D-F Índia/NPR de/P que/WPRO ia/VB-D por/P capitão/N mór/ADJ-R Pedro/NPR Álvares/NPR Cabral/NPR |

Fonte: http://www.tycho.iel.unicamp.br/corpus/manual/prep/manual_frameset.html.

Na figura 06, temos um texto original digitalizado, ou seja, um arquivo do tipo imagem. Este tipo de arquivo apresenta a vantagem de armazenar um grande número de informações com grande fidelidade. Por outro lado, o texto não é manipulável. Assim, é necessário fazer uma transcrição do texto, de modo a torná-lo manipulável.

No projeto do TBCHP, em alguns casos é usada a ferramenta de OCR, que reconhece os caracteres constantes na imagem e faz a conversão em texto editável. Apesar de auxiliar o trabalho de transcrição, a acurácia do OCR para o Português, como já dissemos, não

é muito alta, sobretudo em textos antigos. Este fato faz com que a atividade de revisão após o OCR torne-se imprescindível. Também há casos, como os de manuscritos, por exemplos, onde o uso do OCR é dispensado. Assim, toda a atividade de transcrição é feita de forma manual.

Dentre os *corpora* em Português que analisamos, o TBCHP é o que mais guarda semelhanças com o Astrolábio, no que diz respeito à anotação.

Não obstante os esforços empregados na elaboração de uma precisa metodologia para a transcrição e anotação de textos do TBCHP, a anotação do Astrolábio conta com o respaldo de um sofisticadíssimo sistema de codificação para a anotação de textos em XML: a TEI P5. Conforme já dissemos, a TEI P5 dispõe de uma *tag*, a <*choice*>, que permite armazenar, ao mesmo tempo, tanto a versão corrigida/normalizada, quanto a original.

Como se observa na figura acima, o sistema do TBCHP não dispõe desse recurso, permitindo o armazenamento, em um único arquivo, de apenas um tipo de transcrição.

Outra semelhança com o Astrolábio é que o TBCHP também possui anotação morfossintática. A diferença é que, no Astrolábio, temos tanto a versão XML quanto a anotação morfossintática em um único arquivo. Acreditamos que essa característica seja uma das maiores contribuições de nosso *corpus*.

## 2.7 *Dolmetschen im Krankenhaus* (DiK)

O DiK é um *corpus* elaborado pelo Collaborative Research Centre Multilingalism, fundado pelo *German Research Foundation (Deutsche Forschungsgemeinschaft, DFG)*. É composto por transcrições de áudios de conversações médico-paciente em hospitais, em alemão, turco e português, e foi compilado entre os anos de 1999 e 2005. O interessante desse corpus é que ele traz anotações conforme a TEI P5, porém, não possui anotação morfossintática.

# 3 *NATURAL LANGUAGE PROCESSING* (NLP)

Neste capítulo, fazemos uma breve apresentação de algumas etapas do *Natural Language Processing* (NLP) ou Processamento da Linguagem Natuaral (PLN), em português, e o pacote de ferramentas *Natural Language Toolkit* (NLTK), com ênfase para as funcionalidades do Aelius, o etiquetador usado no Astrolábio.

## 3.1 *Natural Language Toolkit* (NLTK) e as etapas do Processamento da Linguagem Natural

NLTK consiste em um conjunto de ferramentas *open source* criadas pelo *Department of Computer and Information Science at the University of Pennsylvania.* Segundo os autores, não se trata de uma enciclopédia, mas sim de um amplo leque de funções especialmente desenvolvidas para o PLN, dentre as quais estão módulos na linguagem *Python*, bancos de dados e diversos tutoriais. O *nltk.tokenize* (usado na toquenização de palavras e sentenças), *nltk.stem* (útil à lematização) e *nltk.tag* (utilizado na etiquetagem) são apenas alguns exemplos dos módulos constantes em NLTK. O pacote NLTK está disponível para *download* no seguinte endereço: http://nltk.org/.

Os criadores do NLTK argumentam que a linguagem Python foi escolhida em razão da amplitude de funcionalidades que apresenta para o processamento de textos, além de facilitar uma exploração interativa e possuir uma riquíssima biblioteca. Pode-se carregar o Python por meio de um interpretator bastante amigável: *Interactive DeveLopment Environment* (IDLE). Então, dispondo do pacote NLTK e Python instalados em nossa máquina, já podemos iniciar diversas operações em PLN.

O Processamento da Linguagem Natural é um encadeamento sucessivo de atividades executadas computacionalmente e não ocorre de forma aleatória, mas como uma cadeia sucessiva de atividades conhecida como *pipeline*. Conforme Bird, Klein e Loper (2009), os passos são os seguintes: s*entenciador, itemizador, etiquetador morfossintático , chunker (segmentador sintático), shallow parser, deep parser e analisador semântico*.

## 3.2 Toquenizador de sentenças

Quando lemos um texto, podemos reconhecer as suas divisões por meio de regras decorrentes de nosso sistema linguístico. Por exemplo, podemos identificar os parágrafos por meio da indentação (recuo) e do espaçamento entre um parágrafo e outro. As sentenças, por sua vez, podem ser delimitadas por um ponto final.

O computador, entretanto, não interpreta um texto da mesma forma que nós. Então, precisamos informá-lo sobre as segmentações do texto ou ele irá interpretá-lo como sendo apenas uma longa cadeia de caracteres ou *string*. Então, para que possamos iniciar a anotação, necessitamos, inicialmente, dividir o texto em porções menores, ou seja, toquenizar. Na primeira etapa da cadeia, temos a toquenização do texto em sentenças, "tokenization is the process of splitting a string into a list of pieces, or tokens." (PERKINS, 2010, p. 8). Vejamos como esse processo ocorre, utilizando as ferramentas da *Natural Language Processing Toolkit* (NLTK):

Esquema 3 – Exemplo do uso da função *sent_toquenize*

```
>>> import nltk
>>> from nltk.tokenize import sent_tokenize
>>> para= "Sou aluno do Liceu de Camocim. Tenho muitos amigos na minha escola."
>>> sent_tokenize(para)
['Sou aluno do Liceu de Camocim.', 'Tenho muitos amigos na minha escola.']
>>>
```

Fonte: Elaborado pela autora.

No exemplo acima, vemos um pequeno parágrafo composto por duas sentenças: 'Sou aluno do Liceu de Camocim.' e 'Tenho muitos amigos na minha escola.' Nesse caso, as sentenças foram reconhecidas por meio do ponto final.

No pacote de ferramentas NLTK, esse procedimento é realizado por meio do módulo *Punkt Sentence Tokenizer*, que utiliza um algoritmo com abordagem não-supervisionada[23] para detecção dos limites da sentença, proposto por Kiss e Strunke (2006). Esta atividade, porém, não é algo simples, uma vez que um mesmo grafema pode servir para diferentes propósitos:

---

[23]"Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. By contrast with SUPERVISED LEARNING or REINFORCEMENT LEARNING, there are no explicit target outputs or environmenta evaluations associated with each input; rather the unsupervised learner brings to bear prior biases as to what aspects of the structure of the input should be captured in the output." (DAYAN, 1999, p. 1)

> The successful determination of these boundaries is thus a prerequisite for proper sentence processing. Sentence boundary detection is not a trivial task, though. Graphemes often serve more than one purpose in writing systems. The period, which is employed as sentence boundary marker, is no exception. It is also used to mark abbreviations, initials, ordinal numbers, and ellipses. Moreover, a period can be used to mark an abbreviation and a sentence boundary at the same time. In such cases, the second period is haplologically omitted and only one period is used as end-of-sentence and abbreviation marker.1 Sentence boundary detection thus has to be considered as an instance of ambiguity resolution.
> (KISS, STRUNKE, 2006, p. 1)

Há dois tipos principais de desambiguadores dos limites da sentença: baseado em regras e estatístico. Os modelos baseados em regras elaboram-nas com base em gramáticas de expressões regulares, acrescidas de algumas listas de nomes próprios, abreviações etc. Estes modelos geralmente são interessantes para um tipo de *corpus* específico, mas não para usos mais gerais (MIKHEEV, 2004).

O modelo estatístico requer um treino automático de *software, o que é* interessante para o treino rápido e eficiente de um mesmo sistema em diferentes *corpora* e em diferentes línguas. A desambiguação dos limites da sentença é considerada um problema de classificação pela abordagem da aprendizagem de máquina supervisionada. Assim, busca-se o limite da sentença onde são altas as possibilidades de encontrarmos pontos finais de sentença, tais como após letras maiúsculas. Entretanto, esse tipo de abordagem requer um *corpus* previamente anotado (com etiquetas). Em oposição, a aprendizagem com abordagem não-supervisionada pode ser aplicada a textos não-etiquetados, partindo do pressuposto de que as ambiguidades são minoria e que se pode aprender através de usos não-ambíguos (MIKHEEV, 2004).

O módulo *Punkt* oferece toquenizadores para diversas línguas, incluindo o toquenizador para o Português brasileiro, que foi treinado no *corpus* CETENFolha. Pelos exemplos que pudemos testar, o *Punkt* toqueniza corretamente as sentenças que contém abreviações, distinguindo-as o ponto final de sentença, vide exemplo a seguir:

Esquema 4 – Módulo *Punkt*

```
>>> tokenizer = nltk.data.load('tokenizers/punkt/portuguese.pickle')
>>> s= 'Hoje, apresentaremos um artigo de Linguística Computacional, escrito pelo Prof.
Dr. Leonel F. de Alencar. Em seguida, faremos alguns comentários sobre o tema.'
>>> tokenizer.tokenize(s)
['Hoje, apresentaremos um artigo de Lingu\xc3\xadstica Computacional, escrito pelo Prof.
Dr. Leonel F. de Alencar.', 'Em seguida, faremos alguns coment\xc3\xalrios sobre o tema.
']
>>>
```

Fonte: Elaborado pela autora.

Vejamos agora um caso de toquenização de sentenças com reticências delimitando o final da sentença:

Esquema 5 – Reticências delimitando o final da sentença

```
>>>
>>> tokenizer=nltk.data.load('tokenizers/punkt/portuguese.pickle')
>>> s="Estou meio confuso...sem saber o que fazer... Talvez seja hora de parar..
."
>>> tokenizer.tokenize(s)
['Estou meio confuso...sem saber o que fazer...', 'Talvez seja hora de parar...'
]
>>>
>>>
>>> j = "Estou meio confuso...sem saber o que fazer...Talvez seja hora de parar."
>>> sent_tokenize(j)
['Estou meio confuso...sem saber o que fazer...Talvez seja hora de parar.']
>>>
```

Fonte: Elaborado pela autora.

Conforme observamos, a função identificou corretamente as reticências e quebrou o parágrafo em duas sentenças. Porém, quando não inserimos um espaço entre a primeira sentença e a segunda, o computador não reconhece as reticências como delimitadora do final da sentença, interpretando o texto como uma única sentença, ao invés de duas:

Esquema 6 – Reticências delimitando o final da sentença 2

```
>>> s="Estou meio confuso...sem saber o que fazer...Talvez seja hora de parar...
"
>>> tokenizer.tokenize(s)
['Estou meio confuso...sem saber o que fazer...Talvez seja hora de parar...']
>>>
```

Fonte: Elaborado pela autora.

A partir dos exemplos apresentados, também concluímos que a segmentação entre as sentenças também serve como parâmetro para que a função possa reconhecer os limites da sentença.

## 3.3 Toquenizador de palavras (*Word tokenizer*)

O processo seguinte é o *word tokenizer*, ou seja, a quebra das sentenças em palavras:

Esquema 7 – Função *word_tokenizer*

```
>>> from nltk.tokenize import word_tokenize
>>> word_tokenize(para)
['Sou', 'aluno', 'do', 'Liceu', 'de', 'Camocim.', 'Tenho', 'muitos', 'amigos', 'na', 'min
ha', 'escola', '.']
>>>
```

Fonte: Elaborado pela autora.

A toquenização de palavras nas sentenças acima não apresentou grandes dificuldades. Porém, assim como na toquenização de sentenças, a toquenização de palavras está longe de ser algo trivial. Na língua falada, por exemplo, o ouvinte segmenta o discurso em palavras. Para um aprendiz da língua, com pouco conhecimento vocabular, quando as pausas entre as palavras ou são breves ou inexistentes, o processo de segmentação é mais difícil (BIRD,KLEIN,LOPPER,2010). Questões como essa são de extrema relevância para o reconhecimento e síntese da fala, por exemplo.

No mesmo sentido, o discurso presente em *chats* e redes sociais também apresenta algumas particularidades na toquenização, pois são comuns as situações onde os usuários propositadamente desconsideram a segmentação, *#amomuitotudoisso*, *#prontofalei*, *#postenaomamae* etc.

Caso semelhante também ocorre com línguas não-segmentadas, como algumas línguas orientais, e também em segmentações não-convencionais, muito comuns na aquisição da escrita, tanto de crianças como de adultos, como os casos de hipersegmentação apontados no quadro abaixo:

Quadro 2 – Alguns casos de hipersegmentação da escrita

| palavra gramatical + palavra fonológica | palavra fonológica + palavra gramatical | palavra gramatical + palavra gramatical | palavra fonológica + palavra fonológica |
|---|---|---|---|
| em bora (embora) | tu do (tudo) | por que (porque) | verda deiro (verdadeiro) |
| na mora (namora) | correm do (correndo) | | ar partamento (apartamento) |
| a onde (aonde) | gritan do (gritando) | | ter mina (termina) |

Fonte: CUNHA, 2004 apud GONÇALVES; Veçossi, 2011, p. 3.

A hifenização também é um ponto relevante, pois é um elemento que serve para muitos propósitos no texto. Serve tanto para separar palavras compostas, como em '*recém-nascido'* e '*pé-de-meia'*, como para o uso dos clíticos, como em '*diga-me'* e '*dar-te-ei'*. Também se usa hífen para separar as sílabas e para separar palavras no final de linhas. Neste último caso, o hífen deve ser suprimido e a palavra recomposta. Para esta ação, a acurácia[24] do toquenizador pode ter um considerável ganho se, uma vez suprimido o hífen, seja aplicada

[24]A acurácia, ou precisão, de um processador de texto pode ser calculada através da razão entre as respostas corretas e as respostas produzidas (MIKHEEV, 2004).

uma abordagem lexical à palavra concatenada: primeiro, verifica-se em uma lista se ela pertence a um determinado léxico e se é uma palavra legítima. Caso contrário, recoloca-se o hífen (MIKHEEV, 2004).

Outros aspectos também devem ser levados em conta, como o uso de contrações, abreviações, numerais, datas, moedas etc. Veremos a seguir como algumas destas questões são tratadas pelo Aelius.

## 3.4 Aelius e o módulo Toqueniza

O módulo *Toqueniza* do Aelius possui vários modos de toquenização, permitindo ao usuário executá-los de acordo com a sua conveniência. Nesta subsecção, iremos demonstrar alguns estilos de toquenização do módulo sem, contudo, esgotá-los.

A função *TOK_PORT.tokenize* considera hífens e apóstrofos de forma independente:

Esquema 8 – Aelius e a função *TOK_PORT.tokenize*

```
>>>
>>> s1 = "Deixem-me sair, mas antes diga-lhe para que me tragam um copo d'água."
>>> from Toqueniza import TOK_PORT
>>> TOK_PORT.tokenize(s1)
['Deixem', '-', 'me', 'sair', ',', 'mas', 'antes', 'diga', '-', 'lhe', 'para', 'qu
e', 'me', 'tragam', 'um', 'copo', 'd', "'", '\xc3', '\xa1', 'gua', '.']
>>>
```

Fonte: Elaborado pela autora.

No exemplo, percebemos que os caracteres *'\xc3'* e *'x\a1'* foram toquenizados separadamente. Quando definimos o *output* da variável *s1* para a codificação UTF-8, obtivemos o seguinte:

Esquema 9 – Exemplo de sentença codificada em UTF-8

```
>>> s1 = "Deixem-me sair, mas antes diga-lhe para que me tragam um copo d'água.".d
ecode("utf-8")
>>> TOK_PORT.tokenize(s1)
[u'Deixem', u'-', u'me', u'sair', u',', u'mas', u'antes', u'diga', u'-', u'lhe', u
'para', u'que', u'me', u'tragam', u'um', u'copo', u'd', u"'", u'\xe1gua', u'.']
```

Fonte: Elaborado pela autora.

Vejamos agora um exemplo extraído do próprio Aelius:

>>> from Aelius import Toqueniza
>>> print Toqueniza.TEXTO
O Dr. José P. Fernandes disse-lhe que a pistola .45 custa R$ 3,5 mil, 35.08% de Cz$

3.800,98, às 18h30min da segunda-feira (22/10/2010).
No passado.
Dir-se-ia que ele deu com os burros n'água...
>>> for sent in Toqueniza.SENTENCAS:
    for t in Toqueniza.TOK_PORT.tokenize(sent):
        print t

O
Dr.
José
P.
Fernandes
disse
-
lhe
que
a
pistola
.45
custa
R$
3,5
mil
,
35.08
%
de
Cz$
3.800,98
,
às
18h30min
da
segunda
-
feira
(
22
/
10
/
2010
)
.
No
passado
.
Dir
-
se
-
ia
que
ele
deu
com
os
burros
n

```
'
água
...
```

Aqui, cada linha corresponde a um *token*. Assim, em '*dir-se-ia'*, ao invés de uma única palavra, ou *token*, obtivemos cinco *tokens* diferentes (*'dir', '-', 'se', '-', 'ia*'). A função *TOK_PORT_LX*, ao contrário da *TOK_PORT*, não faz essa separação, senão vejamos:

```
>>>for sent in Toqueniza.SENTENCAS:
        for t in Toqueniza.TOK_PORT_LX.tokenize(sent):
                print t


O
Dr.
José
P.
Fernandes
disse-lhe
que
a
pistola
.45
custa
R$
3,5
mil
,
35.08
%
de
Cz$
3.800,98
,
às
18h30min
da
segunda-feira
(
22
/
10
/
2010
)
.
No
passado
.
Dir-se-ia
que
ele
deu
com
os
burros
```

n'água

Verificamos que, em casos de reticências no meio da frase, a função considerou o primeiro ponto como sendo integrante da palavra e os outros dois foram considerados um único *token*. No final da frase, porém, as reticências foram corretamente reconhecidas:

Esquema 10 – Reticências delimitadoras de sentença

```
>>> ret= "Estou pensando...hmmm...Vou continuar pensando..."
>>> TOK_PORT_LX.tokenize(ret)
['Estou', 'pensando.', '..', 'hmmm.', '..', 'Vou', 'continuar', 'pensando', '...
']
>>>
```

Fonte: Elaborado pela autora.

O mesmo não ocorre quando inserimos um espaço entre as palavras e as reticências:

Esquema 11 – Reticências delimitadoras de sentença (2)

```
>>> ret2= "Estou pensando ... hmmm ... Vou continuar pensando..."
>>> TOK_PORT_LX.tokenize(ret2)
['Estou', 'pensando', '...', 'hmmm', '...', 'Vou', 'continuar', 'pensando', '...
']
>>>
```

Fonte: Elaborado pela autora.

A função *Toqueniza.toquenizaPontuacao*, por sua vez, toqueniza cada ponto das reticências, considerando-os *tokens* individuais:

Esquema 12 – Função *Toqueniza.toquenizaPontuacao*

```
>>> Toqueniza.toquenizaPontuacao(ret2)
['Estou', 'pensando', '.', '.', '.', 'hmmm', '.', '.', '.', 'Vou', 'continuar',
'pensando', '.', '.', '.']
>>>
```

Fonte: Elaborado pela autora.

Conforme advertimos no início desta subsecção, apresentamos apenas algumas formas de toquenização permitidas através do módulo *Toqueniza* do Aelius, contudo, muitas outras formas são possíveis.

## 3.5 *Part-of-speech tagging*

A anotação morfossintática, *Part-of-Speech tagging*, *POS tagging* ou simplesmente *tagging*, consiste em atribuir etiquetas (*tags*) às palavras, de acordo com as suas

classificações morfossintáticas (geralmente as etiquetas são inseridas logo após a palavra, *v.g*., *Escola/N*). Essas etiquetas servem como identificadores das classes gramaticais das palavras e são inseridas automaticamente pelo etiquetador. O primeiro etiquetador morfossintático de que temos notícia é o TAGGIT, usado na construção do *corpus* Brown (BERBER SARDINHA, 2004).

O conjunto de etiquetas é denominado de *tagset* e, geralmente, é bem divergente de um etiquetador para outro (vide exemplos dos *tagsets* do LX-Tagger e do MAC-Morpho em anexo), pois são concebidas a partir dos objetivos de cada etiquetador.

Uma vez conhecidas as classes das palavras, torna-se mais fácil a desambiguação lexical e a descrição de padrões léxico-gramaticais. Deste modo, a *POS tagging* serve a diferentes propósitos. Como vimos acima, um *corpus* anotado pode servir de treino para um sistema com abordagem supervisionada. Além disso, também serve para tecnologias em processamento da fala e a indexação de termos na tecnologia da informação.

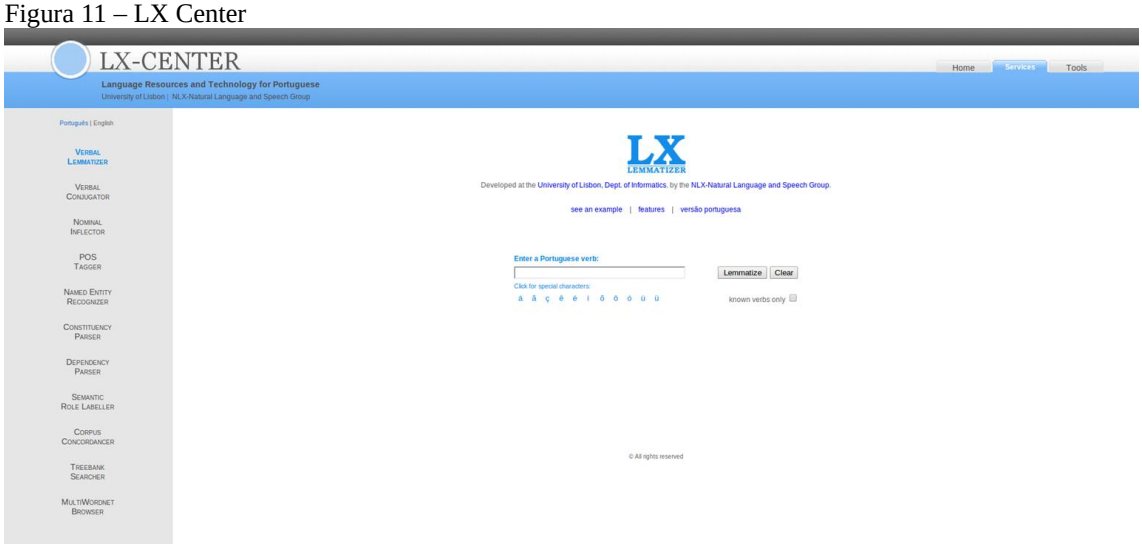
Existem três formas de abordagens utilizadas pelos etiquetadores morfossintáticos: linguística (baseada em regras), probabilística (baseada em *n-gramas*) e híbrida (baseada em regras e em *n-gramas*). Esta última é a adotada pelo Aelius (ALENCAR, 2010).

### *3.5.1 LX-Tagger*

LX-Tagger é um etiquetador morfossintático criado pelo LX-Center (*Language Resources and Technology for Portuguese),* vinculado ao NLX - Grupo de Fala e Linguagem Natural, do Departamento de Informática da Universidade de Lisboa. O termo *LX* é uma referência ao codinome atribuído aos cidadãos lisbonenses.

Além do LX-Tagger, o LX-Center também desenvolveu outras ferramentas para o processamento do Português, como o LX-Tokenizer (toquenizador), LX-Gram (uma gramática computacional do Português) e o LX-Lemmatizer (lematizador automático para verbos em Português). Todas essas estão disponíveis *online* no sítio eletrônico do LX-Center:

Figura 11 – LX Center




Fonte: http://lxcenter.di.fc.ul.pt/.

Também podemos utilizar o LX-Tagger através de uma ferramenta *online*. Por meio dela, podemos anotar um texto bruto. Basta inserirmos um texto na caixa, tomando o cuidado de inserir uma linha em branco, a fim de que o etiquetador possa fazer o reconhecimento dos parágrafos. O *output* desta operação é um texto com etiquetas morfossintáticas (*DEM, PREP, REL* etc. Para maiores informações, consultar o *tagset* em anexo) e XML, indicativas de parágrafos e sentenças (<*p*> e <*s*>), acrescidos do lema e das etiquetas indicativas de gênero (***m**asculine* ou ***f**eminine*), número (***s**ingular* ou ***p**lural*), pessoa (***1**st, **2**nd, **3**rd*) e grau (***d**iminutive*, ***s**uperlative* e ***c**omparative*). As etiquetas ***g*** e ***n*** são utilizadas para identificar palavras com gênero e número não específicos, respectivamente (*v.g., Vi/V pianistas/CN#gp*). Vejamos o exemplo a seguir:

Figura 12 – Ferramenta do LX-Tagger para anotação *online*


Fonte: http://lxcenter.di.fc.ul.pt/services/en/LXServicesSuite.html.

O LX-Tagger também está disponível para download[25] gratuito. Porém, diferentemente dos *softwares* de código aberto, a licença[26] determina que não é lícito aos usuários distribuir ou comercializar qualquer produto ou serviço derivado do todo ou quaisquer de suas partes. De acordo com nosso entendimento, um *corpus* anotado pelo LX-Tagger pode ser considerado um produto derivado. Portanto, ficamos receosos em utilizá-lo em um *corpus* que será compartilhado pela licença *Creative Commons*, razão pela qual refutamos o seu uso no *corpus* Astrolábio.

A etiquetagem morfossintática pelo LX-Tagger compreende três etapas: *chunking*, toquenização e etiquetagem propriamente dita. No *chunking*, realizado pelo LX-Chunker, há o reconhecimento das fronteiras das sentenças e parágrafos e, em sequência, a atribuição de etiquetas XML indicativas de parágrafos (<*p*>) e sentenças (<*s*>).

A etapa seguinte, prevendo situações de ambiguidade, como a toquenização de *deste*: *de* | *este* (preposição + pronome) ou *deste* (verbo), é realizada em duas fases. No referido exemplo, temos um problema circular: para que *deste* seja etiquetado como verbo e não como preposição + pronome, é mister que ele tenha sido previamente toquenizado como

[25]http://lxcenter.di.fc.ul.pt/services/pt/LXServicesSuitePT.html
[26]http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LX-Tagger_License.pdf

um único *token,* ou seja, *deste* e não *de* | *este.* Em razão disto, a etiquetagem é realizada entre duas fases.

Na primeira, as palavras são quebradas em *tokens*, exceto aquelas consideradas ambíguas, que permanecem agrupadas como um único *token*. Em seguida, os *tokens* ambíguos recebem etiquetas duplas, para a forma contraída, e uma única etiqueta para a não-contraída: *deste_V* (verbo) ou *deste_PREPDEM* (preposição + pronome demonstrativo). Finalmente, os *tokens* identificados com duas etiquetas são quebrados em dois *tokens*: *de_PREP* e *este_DEM* (BRANCO; SILVA, 2004).

O referido esquema foi construído a partir do sistema MXPOST, desenvolvido por Adwait Ratnaparkhi (1996). O MXPOST é um modelo estatístico, considerado de Máxima Entropia, cuja meta é maximizar a entropia[27] de uma distribuição sujeita a certas restrições. Originalmente, MXPOST foi treinado no *Wall Street Journal corpus*, do Penn Treebank project[28], mas pode ser treinado a partir de outros *corpora* com etiquetas morfossintáticas. A vantagem de um modelo de Máxima Entropia, diante de outros etiquetadores morfossintáticos baseados em *corpora*, é que ele combina técnicas de outros modelos, utilizando um rico esquema de representação, baseado em regras, e gerando uma distribuição de possíveis etiquetas para cada palavra (RATNAPARKHI, 1996). O modelo MXPOST permitiu que o etiquetador alcançasse a precisão de 97,08%. No *corpus* utilizado para o treino (LX-Corpus), foram encontrados 2% de *tokens* ambíguos, dentre os quais foram resolvidos 99,04% dos casos (BRANCO; SILVA, 2004).

Além do MXPOST, o LX-Corpus foi utilizado para testar a acurácia de outros etiquetadores, com diferentes algoritmos: TBL (*Transformation Based Learning*[29]), TnT e Qtag (utilizam modelo HMM – *Hidden Markov Model*). A seguir, vemos uma tabela com os resultados obtidos:

---

[27]O conceito de entropia é oriundo da Termodinâmica, mas tem utilização em diferentes áreas de estudo. Assim, "para especificar o calor devemos utilizar pelo menos dois números: um para medir a quantidade de energia, o outro para medir a quantidade de desordem. A quantidade de energia é medida em termos de uma unidade prática chamada caloria... A quantidade de desordem é medida em termos do conceito matemático chamado entropia... (HALLIDAY; RESNICK, 1971, P. 699)". Portanto, compreendemos a entropia como a unidade de medida da desordem de um um sistema.

[28]http://www.cis.upenn.edu/~treebank/

[29]Ao contrário de modelos estatísticos como o HMM e o de Máxima Entropia, o TBL é m modelo baseado em regras.

Tabela 2 – Acurácia de etiquetadores

| **Sistema** | TBL | TnT | MXPOST | QTag |
|---|---|---|---|---|
| **Acurária** | 97,09% | 96, 87% | 97, 08% | 89,97% |

Fonte: BRANCO; SILVA, 2004.

A partir dos resultados apontados, podemos concluir que o etiquetador que apresentou melhor performance com o LX-Corpus foi o etiquetador TBL, que utiliza um modelo baseado em regras, seguido do MXPOST.

Retomamos o exemplo da *esquema 10*, onde fizemos a toquenização de sentenças, contendo reticências, por meio da função *sent_tokenizer* do NLTK, e refizemos o teste por meio da ferramenta online do LX-Tagger:

Figura 13 – Etiquetagem pela ferramenta *online* do LX-Tagger


Fonte: http://lxcenter.di.fc.ul.pt/services/en/LXServicesSuite.html.

Assim como na função *sent_tokenize*, o LX-Tagger reconheceu corretamente os limites das sentenças que continham reticências como delimitador. Porém, do mesmo modo que a *sent_tokenize*, necessita da segmentação entre as sentenças para os limites das mesmas sejam identificados corretamente.

### *3.5.2 HunPos*

HunPos (HALÁCSY; KORNAI; ORAVECZ, 2007) é um etiquetador *open source* e com licença *freeware,* que se apresenta como alternativa ao Trigrams'n'Tags, TnT (BRANTS, 2000), um etiquetador *freeware* de alta performance, que utiliza o modelo estatístico HMM (*Hidden Markov Model*), mas que traz a inconveniência de ser *closed source*. Acerca dos modelos HMM ou Modelo Oculto de Markov, valemo-nos das esclarecedoras palavras de Espíndola (2009):

> Nas palavras de Rabiner, na maioria dos processos Markovianos, cada estado corresponde a um observável do sistema. Para esclarecer a ideia, consideremos o exemplo sobre modelagem do tempo, introduzido no capítulo 2, sobre Cadeias de Markov. Ao verificar a condição do tempo em um determinado dia, o observador obterá diretamente um dos estados da Markov como resposta, {S1 = chuvoso, S2 = nublado, S3 = ensolarado}.
> Por outro lado, Modelos Ocultos de Markov são usados na modelagem de processos Markovianos que geram observáveis de forma indireta, em função das transições entre os estados da cadeia de Markov que governa o processo, mas que não pode ser diretamente observada. Em outras palavras, a evolução da cadeia de Markov está escondida do observador. Em comparação à proposta anterior de modelagem do tempo por Cadeias de Markov, uma possível modelagem em HMM poderia tratar da observação do comportamento de um trabalhador em sua forma de transporte ao trabalho. Esse trabalhador se locomove de bicicleta ou táxi em função do tempo ou de sua previsão. Geralmente vai ao trabalho de bicicleta, mas costuma pegar táxi em dias chuvosos. Assim, se esse trabalhador foi trabalhar de bicicleta em um determinado dia, há uma probabilidade maior de que o dia esteja ensolarado do que chuvoso, mas ainda assim pode se tratar de um dia de chuva.
> Assim, a diferença fundamental entre HMM e o resto dos formalismos Markovianos está na forma de se observar o sistema. Enquanto que na maioria dos processos Markovianos a observação é direta, pois os observáveis são os próprios estados, em HMM a observação é indireta, feita por inferência, pois os observáveis são funções probabilísticas dos estados da Markov ou das transições entre esses estados.
> (ESPÍNDOLA, 2009, p. 9)

O HunPos foi construído em Ocaml, uma linguagem de programação de alto nível, e é um modelo baseado em *n*-gramas:

> Um modo de agrupar todas as sequências de tamanho *n* que começam pelas mesmas *n-1* palavras em uma classe de equivalência é supor que o contexto local prévio afeta a palavra seguinte, e construir o modelo de Markov de ordem *(n-1)* ou modelo de *n*-Gramas (sendo a última palavra do *n*-grama a que está sendo prevista). Os casos de *n*-gramas mais utilizados são com n = 2, 3 e 4, particularmente denominados bigramas, trigramas e tetragramas.
> Quanto maior o valor de *n*, isto é, maior o número de classes que dividem os dados, maior a confiabilidade da inferência. No entanto, o número de parâmetros a serem estimados cresce exponencialmente em relação a n. Por isso, geralmente são utilizados bigramas ou trigramas em sistemas dessa natureza.
> (GASPERIN; LIMA, 2001, p.18)

Este modelo é interessante para a resolução de problemas que necessitam da inferência estatística, como a previsibilidade da palavra seguinte em uma frase, levando em consideração as palavras anteriores (GASPERIN; LIMA, 2001). Assim, no HunPos, são estimadas as probabilidades de ocorrência de uma palavra do léxico, tomando por parâmetros a etiqueta atual e a etiqueta anterior, utilizando trigramas.

### *3.5.3 Aelius Brazilian Portuguese Pos-Tagger*

O Aelius[30] é um pacote *open source*, construído na linguagem Python, com base na biblioteca NLTK. O pacote foi desenvolvido com o intuito de treinar, avaliar e anotar *corpora* em Português brasileiro e suas variedades. Além das ferramentas oferecidas pelo NLTK, Aelius traz outras funcionalidades como um algoritmo próprio para lidar com palavras com inicial maiúscula. Com isso, presta-se à execução de pré-processamento de texto; avaliação de etiquetadores; comparação entre tipos distintos de anotação; construção de *language models*[31] e e etiquetadores com base em corpora etiquetados (ALENCAR, 2010).

Os etiquetadores morfossintáticos AeliusBRUBT, AeliusRUBT e AeliusHunPos foram treinados a partir do TBCHP. O AeliusRUBT, apesar de ser mais rápido que o AeliusBRUBT, apresentou menos precisão no procedimento de usado como parte do procedimento de expansão de contrações (ALENCAR, 2010).

O modelo AeliusHunPosMacMorpho, por sua vez, foi treinado com o etiquetador HunPos[32], a partir do *corpus* de treino MAC-Morpho[33]. O Aelius também usa o etiquetador LX-Tagger, que por sua vez foi desenvolvido a partir do etiquetador MXPOST[34]. A seguir, podemos ver alguns exemplos de trechos do Astrolábio etiquetados por diferentes etiquetadores dentro do Aelius:

---

[30]http://aelius.sourceforge.net/
[31]Language models são utilizados, em NLP, para estimar a probabilidade de sequências de palavras.(MITKOV, 2004).
[32]http://code.google.com/p/hunpos/
[33]http://www.nilc.icmc.usp.br/lacioweb/corpora.htm
[34]http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

Esquema 13 – Trecho do Astrolábio para *input* de etiquetadores Aelius

```
>>> print texto
tt.txt
>>> print open(texto).read()
Escola, um lugar de ensino e aprendizagem,Onde, todos contribuem para mais capac
idade, E assim vou descobrindo e me dando com a realidade .

Liceu, foi a escola que escolhi para estudar, E para cada dia esta nesse ambient
e escola, tudo é bem legal e bem interessante, E assim vou seguindo confiante.
```

Fonte: Elaborado pela autora.

Esquema 14 – Etiquetagem de um trecho do Astrolábio pelo LX-Tagger dentro do Aelius

```
>>> import Extras, Toqueniza, AnotaCorpus
>>> lx=m=Extras.carrega("lxtagger")
>>> from AnotaCorpus import anota_texto
>>> from Toqueniza import TOK_PORT_LX2
>>> AnotaCorpus.anota_texto (texto,lx,"mxpost",Toqueniza.TOK_PORT_LX2,separacao_
contracoes=True)
Arquivo anotado:
tt.mxpost.txt
>>> ttlx="tt.mxpost.txt"
>>> print ttlx
tt.mxpost.txt
>>> print open(ttlx).read()
Escola/PNM ,/PNT um/UM lugar/CN de/PREP ensino/CN e/CJ aprendizagem/CN ,/PNT Ond
e/ADV ,/PNT todos/QNT contribuem/V para/PREP mais/ADV capacidade/CN ,/PNT E/CJ a
ssim/ADV vou/V descobrindo/GER e/CJ me/CL dando/GER com/PREP a/DA realidade/CN .
/PNT


Liceu/PNM ,/PNT foi/V a/DA escola/CN que/REL escolhi/V para/PREP estudar/INF ,/P
NT E/CJ para/PREP cada/QNT dia/CN esta/V em/PREP esse/DEM ambiente/ADJ escola/CN
 ,/PNT tudo/IND é/V bem/ADV legal/ADJ e/CJ bem/ADV interessante/ADJ ,/PNT E/CJ a
ssim/ADV vou/V seguindo/GER confiante/ADJ ./PNT
```

Fonte: Elaborado pela autora.

Esquema 15 – Etiquetagem de um trecho do Astrolábio pelo AeliusHunPos

```
>>> h=Extras.carrega("AeliusHunPos")
>>> AnotaCorpus.anota_texto(texto,h,"hunpos",Toqueniza.TOK_PORT)
>>> h=Extras.carrega("AeliusHunPos")
>>> h=Extras.carrega("AeliusHunPos")
>>> AnotaCorpus.anota_texto(texto,h,"hunpos",Toqueniza.TOK_PORT)
>>>
>>> AnotaCorpus.anota_texto(texto,h,"hunpos",Toqueniza.TOK_PORT)
Arquivo anotado:
tt.hunpos.txt
>>> tth="tt.hunpos.txt"
>>> print open(tth).read()
Escola/N ,/, um/D-UM lugar/N de/P ensino/N e/CONJ aprendizagem/N ,/, Onde/WADV ,
/, todos/Q-P contribuem/VB-P para/P mais/ADV-R capacidade/N ,/, E/CONJ assim/ADV
 vou/VB-P descobrindo/VB-G e/CONJ me/CL dando/VB-G com/P a/D-F realidade/N ./.


Liceu/VB-D ,/, foi/SR-D a/D-F escola/N que/WPRO escolhi/VB-D para/P estudar/VB ,
/, E/CONJ para/P cada/Q-G dia/N esta/ET-P nesse/P+D ambiente/N escola/N ,/, tudo
/Q é/SR-P bem/N legal/ADJ-G e/CONJ bem/ADV interessante/ADJ-G ,/, E/CONJ assim/A
DV vou/VB-P seguindo/VB-G confiante/ADV ./.
```

Fonte: Elaborado pela autora.

Esquema 16 – Etiquetagem de um trecho do Astrolábio pelo AeliusHunPosMacMorpho

```
>>> m=Extras.carrega("AeliusHunPosMM")
>>> AnotaCorpus.INFIXO="hunpos_macmorpho"
>>> AnotaCorpus.anota_texto(texto,m,"hunpos",Toqueniza.TOK_PORT_MM,separacao_con
tracoes=True)
Arquivo anotado:
tt.hunpos_macmorpho.txt
>>> tthm="tt.hunpos_macmorpho.txt"
>>> print open(tthm).read()
Escola/N ,/, um/ART lugar/N de/PREP ensino/N e/KC aprendizagem/N ,/, Onde/ADV ,/
, todos/PROSUB contribuem/V para/PREP mais/PROADJ capacidade/N ,/, E/KC assim/AD
V vou/VAUX descobrindo/V e/KC me/PROPESS dando/V com/PREP a/ART realidade/N ./.


Liceu/NPROP ,/, foi/V a/ART escola/N que/PRO-KS-REL escolhi/V para/PREP estudar/
V ,/, E/KC para/PREP cada/PROADJ dia/N esta/PROADJ em/PREP|+ esse/PROADJ ambient
e/N escola/N ,/, tudo/PROSUB é/V bem/ADV legal/ADJ e/KC bem/ADV interessante/ADJ
 ,/, E/KC assim/ADV vou/VAUX seguindo/V confiante/ADJ ./.


>>>
```

Fonte: Elaborado pela autora.

## 4 *TEXT ENCONDING INITIATIVE* (TEI) P5

A TEI surgiu em 1987, como um consórcio das seguintes instituições: *Association for Computers and the Humanities*, *Association for Computational Linguistics* e *Association for Literary and Linguistic Computing* e é financiada por recursos de diversas agências. Atualmente, o consórcio TEI é uma corporação sem fins lucrativos, adotada por: *The Research Technologies Service na University of Oxford*; *The Scholarly Technology Group (Brown University)*; Grupo francófono ATILF, INIST e LORIA; *The Institute for Advanced Technology in the Humanities (University of Virginia)*.

A TEI é uma codificação cujas diretrizes são dirigidas a todos que desejam trocar informações armazenadas em formato eletrônico, permitindo a manipulação do texto, o que não seria possível através de um texto em formato de imagem. As diretrizes da TEI definem meios para tornar explícitas certas características de um texto, a fim de permitir o seu processamento em diferentes máquinas (TEI, 2012).

Convém destacar um trecho do relato sobre os impactos da TEI, encontrado no sítio eletrônico do projeto:

> O impacto da TEI no conhecimento digital tem sido enorme. Hoje, a TEI é internacionalmente reconhecida como sendo uma ferramenta essencial, tanto para a preservação de dados eletrônicos por um longo período, como para a efetiva contribuição no uso de tais dados em muitas áreas. É o esquema de codificação escolhido para a produção de edições críticas e rebuscadas de textos literários, para trabalhos de referência e grandes *corpora* linguísticos e para o gerenciamento e produção de metadados detalhados associados com texto eletrônico e coleções do patrimônio cultural de muitos tipos [...].
> As recomendações da TEI têm sido endossadas por muitas organizações, incluindo US National Endowment for the Humanities, UK's Arts and Humanities Research Board, Modern Language Association, European Union's Expert Advisory Group for Language Engineering Standards e muitas outras agências ao redor do mundo que financiam ou promovem bibliotecas digitais e projetos de texto eletrônico. Reconhecendo a sua importância na emergente comunidade de biblioteca digital, a Biblioteca do Congresso tem produzido diretrizes para a melhor prática na aplicação das recomendações dos metadados da TEI para interoperabilidade com outros padrões. (TEI, 2012, online, tradução nossa).

Os textos anotados segundo essa codificação também são de interesse do relevante depositório eletrônico *University of Oxford Text Archive* (OTA[35]). Assim,

[35]"If you have created a valuable academic resource that conforms to the TEI Guidelines and you want to ensure that a copy will be available in the future, you may wish to investigate the possibility of depositing your texts with an electronic text archive, quite independently of any plans you have for distribution of it yourself. The

submeteremos o *corpus* Astrolábio a um possível depósito no OTA. Até o presente momento, não consta, no referido depositório, *corpus* em português.

Outro importante ponto é que as diretrizes da TEI estão disponíveis para *download* gratuito no sítio eletrônico[36], sob a licença *Creative Commons Attribution 3.0 Unported License*[37], onde é permitido a qualquer usuário compartilhar e, inclusive, alterar o conteúdo, desde que cite a fonte. Além disso, lá existem vários tutoriais e a participação da comunidade é bastante incentivada[38]. Também devemos acrescentar que muitos editores gratuitos, inclusive o EditiX, e o etiquetador Aelius, possuem a *template*[39] da TEI P5.

## 4.1 A metalinguagem *Extensible Markup Language* (XML) e a TEI

As diretrizes prescrevem o uso da linguagem *Extensible Markup Language* (XML), uma metalinguagem de programação derivada da *Standard Generalized Markup Language* (SGML). Ainda segundo as diretrizes, o *design* dos documentos precisa obedecer a alguns critérios, como: apresentar as características textuais necessárias para a pesquisa; ser simples, claro e concreto; ser usado facilmente por pesquisadores, sem a necessidade de uso de *softwares* especiais; permitir definições rigorosas e eficientes para o processamento de textos; permitir extensões definidas pelo usuário e estar em conformidade com as normas existentes (TEI, 2012).

Em suma, para que um linguista anote um texto segundo a codificação da TEI, não é necessário que ele disponha de conhecimentos avançados de programação ou de *softwares* específicos. Então, se o pesquisador tiver uma noção básica de linguagem *HyperText Markup Language*[40] HTML, muito utilizada para a criação de páginas de hipertexto na *web* e, por isso, bastante difundida, já dispõe de boa parte das habilidades computacionais necessárias para anotar a estrutura de um texto.

Isso ocorre porque, apesar de que a TEI prescreva a XML como linguagem, tanto

---

Oxford Text Archive is one of the oldest such archives and is always on the look-out for scholarly digital resources." (TEI, 2012)

[36] http://www.tei-c.org/Guidelines/

[37] http://www.tei-c.org/Guidelines/access.xml

[38] http://www.tei-c.org/Guidelines/participation.xml

[39] Template é um modelo pré-formatado, que segue as diretrizes de uma determinada linguagem.

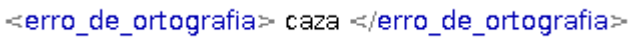
[40] O *tagset* da HTML pode ser encontrado aqui: http://www.w3schools.com/tags/default.asp .

ela como a HTML são descendentes da linguagem *Standard Generalized Markup Language* (SGML), sendo inevitáveis as semelhanças. Essa característica é importante pois facilita a receptividade do projeto no meio acadêmico, sobretudo entre estudantes e pesquisadores fora das ciências exatas.

A HTML, todavia, apresenta algumas limitações, como a impossibilidade de criação de novas *tags*. Em outras palavras, o conjunto de etiquetas é pré-definido, sendo defeso ao usuário criar as suas próprias (etiquetas). Isto talvez seja porque a linguagem HTML é mais voltada para o *design* de páginas na *web* e suas funcionalidades estejam mais voltadas para a disposição gráfica dos elementos na página do que para o arquivamento das informações do texto em si.

Por outro turno, a XML, uma metalinguagem, possui *tags* extensíveis. Isso permite que o pesquisador possa marcar o texto em análise, conforme a sua conveniência. Vejamos o exemplo a seguir, anotado em XML:

Esquema 17 – Exemplo de *tag* extensível em XML




Fonte: Elaborado pela autora.

Dentro da sintaxe XML, o exemplo acima está correto. O pesquisador poderia, inclusive, valer-se das *tags* <*erro_de_ortografia*> e fazer uma pequena programação, em *Python*[41], por exemplo, que lhe permitisse calcular a quantidade de erros ortográficos *caza* encontrados no texto, extraindo e contanto os elementos <*erro_ de_ ortografia*>.

Um texto pode ser anotado em qualquer editor de texto, sem complicações, porém, é recomendável usar um editor de XML, para a validação da sintaxe. Mas, isso não é empecilho, pois existe uma vasta gama de editores com licença *Freeware* (como o EditiX[42], que usamos para esta pesquisa) ou com licenças do tipo *Shareware*[43] (como é o caso do Oxygen).

Como vimos, é possível valer-se da metalinguagem para anotar um texto. Imaginemos, porém, como seria difícil compreender as marcações elaboradas por diferentes pesquisadores. Então, a TEI surgiu com esse intuito: criar uma codificação abrangente o

---

[41]"Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs." (PYTHON, 2012, online).

[42]Endereço disponível para *download*: http://free.editix.com/download/editix-free-2010.tar.gz

[43]Na licença *Shareware*, o uso é restrito ou limitado a um certo período. No caso do *Oxygen*, o período da versão *trial* é de 30 dias.

suficiente para padronizar a anotação de todo e qualquer texto. A TEI P5 prescreve o uso da linguagem XML para a anotação. É importante salientar que a TEI permite a anotação dos metadados, tais como indicações de autoria, data e informações editoriais e também possibilita a utilização de ferramentas de validação a partir do *Document Type Definition* (DTD)[44] da TEI. Tomando por base o exemplo do esquema 2, utilizando a *tag* <*choice*>, prescrita pela TEI P5, teríamos isto:

Esquema 18 – Exemplo de uso da *tag* <*choice*>

```
<choice>
        <sic>caza</sic>
        <corr> casa </corr>
</choice>
```

Fonte: Elaborado pela autora.

Obviamente, a empreitada requereu empenho e perseverança de seus criadores. A quantidade de versões (a mais recente é a TEI P5), que sucederam à original, ressaltam as melhorias e dificuldades de encontrar um meio próprio de descrever algo tão complexo e dinâmico como o texto. Por outro lado, devemos considerar o respaldo do consórcio que mantém o projeto, formado pelas seguintes instituições: Universidade da Virgínia (EUA), Universidade de Bergen (Noruega), Oxford (Inglaterra) e Universidade de Brown (EUA)[45].

A XML é uma linguagem do tipo descritiva, ou seja, tem por finalidade descrever os elementos do texto. Opõe-se às linguagens do tipo procedimental, que definem qual o procedimento a ser realizado em um determinado ponto do documento (RAMALHO, 2000). Por exemplo, na linguagem *Python*, que é procedimental, se desejamos saber o resultado da soma '1 + 2', devemos "dizer" ao computador o que ele deve fazer, da seguinte forma:

Esquema 19 – Exemplo de execução de comando em Python

```
>>>
>>> print 1 + 1
2
>>>
```

Fonte: Elaborado pela autora.

[44]A DTD pode ser compreendida como um conjunto de regras que define quais tags podem existir dentro de um documento e a sua sintaxe.

[45]Para que tenhamos noção da importância dessa instituição, foi lá que se originou o primeiro *corpus* eletrônico de que temos notícia: o *Brown Corpus*, no início da década de 60, contendo um milhão de palavras da inglês americano (OLIVEIRA, 2009; SARDINHA, 2000).

Após digitar os comandos e pressionarmos a tecla *Enter*, temos o resultado fornecido pela máquina: *3*. A XML, ao contrário, não é prescritiva, ou seja, não "diz" ao computador que ação ele deve executar, mas vale-se da metalinguagem para descrever o conteúdo de um texto.

As *tags* XML, diferentemente das tags HTML, são extensíveis, ou seja, não existem em um número finito e os usuários podem criar as suas próprias etiquetas, de modo que atenda melhor às suas necessidades. De outro modo, não podem contruir um texto com etiquetas XML de modo aleatório, é necessário observar a sua sintaxe[46]. Por exemplo, se desejamos marcar determinado texto como parágrafo, usando a *tag* de parágrafo para HTML, devemos inserir o texto entre as *tags* <*p*></*p*>. A primeira *tag* indica o início do texto e a segunda delimita o seu encerramento. Algumas vezes, entretanto, uma única *tag* inicia e encerra o conteúdo, como é o caso da *tag* <*title/*>, dentro da *tag* <*teiHeader*>, no esquema 6.

## 4.2 Anotação da estrutura do Astrolábio conforme a TEI P5

As primeiras edições das TEI usavam a linguagem SGML. A partir de 2002, foi substituída pela XML. Um texto anotado conforme a TEI é dividido, basicamente, em duas estruturas: cabeçalho, <*teiHeader*>, onde são anotados os metadados (autoria e demais informações sobre o documento) e texto, <*text*>, contendo o texto em si. A seguir, temos uma visão geral dessa estrutura:

Esquema 20 – Exemplo de esquema geral de um documento anotado conforme a TEI P5

```
<TEI>
        <teiHeader/>
        <text>
        <front/>
        <body/>
        <back/>
        </text>
</TEI>
```

Fonte: Elaborado pela autora.

[46]Por essa razão, recomendamos o uso de um editor de XML, ainda que esse tipo de documento possa ser construído em qualquer editor de texto (*Word*, *OppenOffice* etc).

A *tag* <*front*> serve para marcar informações introdutórias ao texto, como dedicatória, epígrafe etc. A *tag* <*body*> indica o corpo do texto e a *tag* <*back*>, por sua vez, delimita os apêndices.

Obedecendo a esse esquema, o Astrolábio tem a seguinte estrutura geral:

Esquema 21 – Estrutura geral do Astrolábio conforme a TEI P5

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
        <teiHeader>
            <fileDesc>
                    <titleStmt>
                     <title></title>
                            <genre> </genre>
                     <person sex=" " age=" ">
                            <birth when=" ">
                                    <date></date>
                                    <name type="place"></name>
                            </birth>
                            <residence></residence>
                            <education></education>
                            <occupation></occupation>
                     </person>
                <respStmt xml:id="km">
                     <resp>Corpus Astrolabio organizer</resp>
                        <name>Katiuscia de Moraes Andrade</name>
            </respStmt>
                    <respStmt xml:id="ap">
                     <resp>Automatic tagger</resp>
                            <name>Astro.py</name>
                    </respStmt>
                    </titleStmt>
              </fileDesc>
              <publicationStmt>
                    <availability></availability>
                    </publicationStmt>
                    <editorialDecl>
                    <correction>
                    <p>Errors in th original .txt file were corrected by using the interface Astro.py for Enchant </p>
                    <p>Another erros that couldn't be identified by Enchant (as its own errors) were submitted to a human revision</p>
                    </correction>
                    <normalization></normalization>
                    </editorialDecl>
        </teiHeader>
<facsimile>
        <surfaceGrp n=" ">
                <surface>
                        ulx=" "
                        uly=" "
                        <graphic url=" "/>
                        <zone ulx=" " uly=" "/>
                        <!-- contains a black tag to cover the autor's signature -->
              </surface>
</surfaceGrp>
</facsimile>
        <text>
<body>
        <pb facs=" "/>

 </body>
        </text>
</TEI>
```

Fonte: Elaborado pela autora.

Na primeira linha, temos a indicação da versão da XML, bem como a codificação em que o documento foi elaborado[47]. A segunda, dá informações a respeito da data em que foi produzido e sobre o programa onde foi construído: EditiX. Na terceira linha, temos a

[47]Vide sessão 2.3 deste capítulo.

indicação de que o documento está conforme os padrões da TEI e, em seguida, dentro do cabeçalho, temos a *tag* <*titleStmt/*>, que contém outras *tags* que nos fornecem informações gerais a respeito da obra: <*title*> indica o título propriamente dito; <*genre*> aponta o gênero; <*person*> traz as informações sobre o autor (idade, sexo, data de nascimento, local de nascimento, local de residência, escolaridade e ocupação).

A <*respStmt*> informa quem são os responsáveis (editores, revisores, organizadores etc.) pela obra. A tag possui um atributo para identificar esses responsáveis: *@resp*. Para o nosso *corpus*, esse atributo possui grande importância, pois nos ajuda a identificar quem foi o responsável por determinada correção. No exemplo a seguir, podemos identificar, através do atributo *#at* que a correção foi feita automaticamente (*Automatic tagger*).

Esquema 22 – Exemplo de atributo identificador de responsável

```
<choice>
 <corr resp="#at">
  <w xml:id="15">imenso</w>
 </corr>
 <sic>
  <w xml:id="15sic">imenço</w>
 </sic>
</choice>
```

Fonte: Elaborado pela autora.

O atributo também nos possibilita a identificação de erros na correção automática:

Esquema 23 – Atributo para identificar erros na correção automática

```
<choice resp="km">
 <corr>
  <w xml:id="39" at="vim to">vento</w>
 </corr>
 <sic>
  <w xml:id="39sic">vinto</w>
 </sic>
</choice>
```

Fonte: Elaborado pela autora.

No caso acima, *at=”vim to”* quer dizer que o corretor automático corrigiu a palavra *vinto* para *vim to* e *resp=”km”*, posteriormente, corrigiu para *vento*.

Já a <*publicationStmt*> serve para fornecer dados acerca da publicação (licença, editorial etc.). por sua vez, <*facsimile*> significa que o texto é uma transcrição a partir de um

documento digitalizado; <*surfaceGrp*> representa a folha ao qual o texto se refere e traz informações sobre a resolução da imagem (ulx e uly), enquanto a aparição da <*pb facs*> significa que, a partir daquele momento, o texto é uma representação de uma determinada imagem página da folha indicada na <*surfaceGrp*>. Finalmente, *a tag* <*text*> indica a transcrição do texto propriamente dito.

# 5 QUESTÕES E METODOLOGIA

Neste capítulo, procuramos descrever as etapas para a construção do Astrolábio, abordando os problemas e as soluções encontradas.

## 5.1 Problemas

a) Como preservar as características, tornando a transcrição o mais fiel possível ao original?
b) Como “limpar” os arquivos para a anotação automática pelo Aelius?
c) Como integrar anotação estrutural e morfossintática, agregando *tags* em um nível de análise mais detalhado, como a *tag* <*choice*>, por exemplo?
d) Que etiquetador morfossintático do Aelius é mais preciso na etiquetagem de textos do Astrolábio?
e) Como preservar, no mesmo arquivo, fenômenos de variação linguística, erros ortográficos e a forma corrigida?
f) De que maneira o *corpus* Astrolábio pode embasar o desenvolvimento de ferramentas de tecnologia da linguagem natural?

## 5.2 Hipóteses

a) A preservação das características originais é possível por meio da anotação estrutural, utilizando a metalinguagem XML, conforme a TEI P5.
b) A anotação automática dos textos com anotação estrutural, por meio do Aelius, pode ser realizada por meio do isolamento das *tags* XML e utilizando-se apenas os textos inseridos dentro das mesmas.
c) A integração de *tags* mais específicas, como por exemplo, a *tag* <*choice*>, é feita através de inserção manual, complementando a etiquetagem morfossintática e estrutural, previamente realizadas automaticamente pelo

Aelius.

d) O AeliusHunPosMacMorpho é mais preciso que o AeliusHunPos e o LX-Tagger.

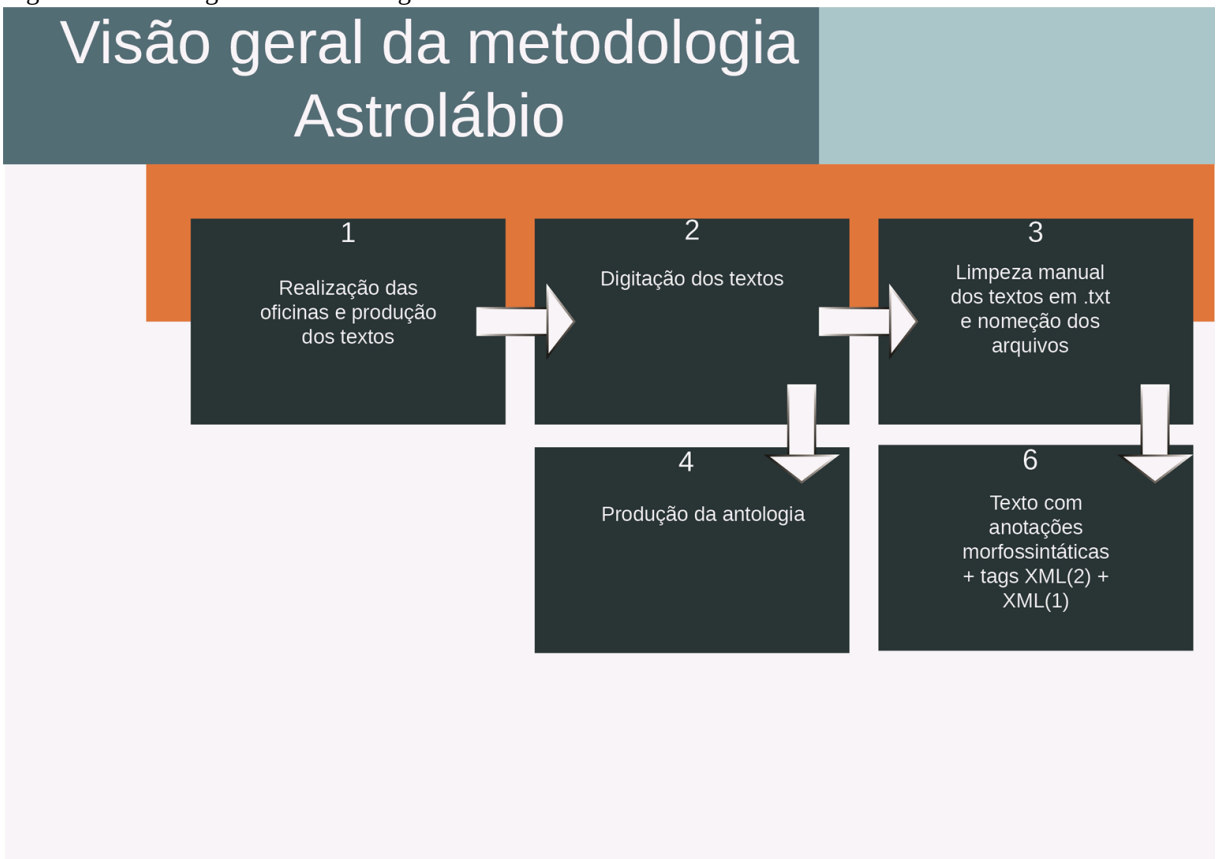
e) A *tag* <*choice*> permite reunir, em um único arquivo, fenômenos de variação linguística, erros ortográficos e a forma corrigida.

f) O *corpus* Astrolábio será um *corpus* de treino para etiquetadores morfossintáticos e corretores ortográficos, tornando-os mais robustos e capazes de lidar com a linguagem não-padrão.

## 5.3 Metodologia utilizada

A seguir, temos uma figura que nos apresenta uma visão geral dessas etapas:

Figura 14 – Visão geral da metodologia




Fonte: Elaborada pela autora.

### *5.3.1 A escolha do* corpus*: oficinas do projeto Rota das Especiarias – Temperos Literários e a compilação dos arquivos*

No primeiro semestre de 2012, recebemos o desafio de coordenar a segunda edição do Rota das Especiarias – Temperos Literários[48], que contemplou as cidades de Barroquinha, Camocim e Jijoca de Jericoacoara. Trata-se de um projeto itinerante de literatura, cujos principais objetivos são popularizar a cultura do livro e da leitura, incentivar a produção escrita e encontrar soluções alternativas para a distribuição de obras de autores independentes e pequenas editoras. A edição teve produção geral e executiva da Fotossíntese:.Arte:.Comunicação, juntamente com a Editora Corsário, e patrocínio do Banco do Nordeste, através do edital do Programa BNB de Cultura 2011, em parceria com o BNDES.

O projeto dividiu-se em três etapas principais: oficinas, capacitação e feiras. As oficinas de leitura e produção textual foram realizadas no Liceu de Camocim, EEM Jaime Laurindo (Barroquinha) e EEM José Teixeira de Albuquerque (Jijoca), durante os meses de março e abril de 2012.

Os textos originados durante as oficinas, inicialmente, tinham a única finalidade de constituir a antologia *Temperos Literários 2* (AZIGON *et al*., 2012), publicada durante as feiras. De outro modo, percebemos que o conteúdo produzido, redações sob a forma de manuscritos, consistia um rico material para análise linguística. Com isso, decidimos organizá-los no *corpus* que denominamos Astrolábio.

Por último, a abordagem estatística, como o próprio nome sugere, propõe que sejam feitos cálculos matemáticos a fim de se ter uma noção do tamanho ideal de um corpus para que ele seja representativo.

Abaixo, apresentamos a quantidade de autores e textos produzidos no Astrolábio, distribuídos por cidade:

[48]www.rotadasespeciarias.art.br

Tabela 3 – Número de textos e autores do *Corpus* Astrolábio

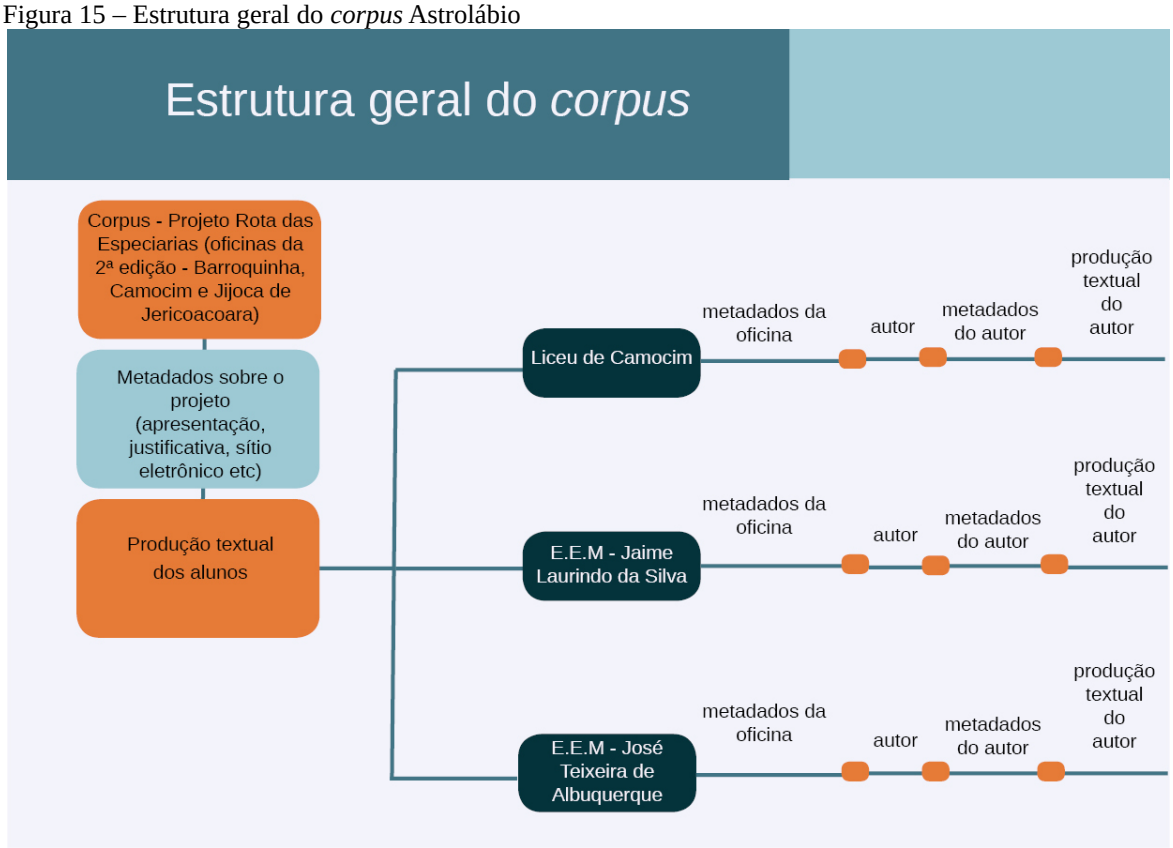
| **Número de textos do Corpus Astrolábio** | | |
|---|---|---|
| **Cidade** | **Número de autores** | **Número de textos produzidos durante as oficinas** |
| Barroquinha | 35 | 122 |
| Camocim | 31 | 103 |
| Jijoca | 21 | 61 |
| **Total** | **87** | **286** |

Fonte: Elaborada pela autora.

Portanto, o *corpus* é formado por um total de 286 textos, produzidos por 87 autores. Como o Astrolábio ainda está em fase de conclusão, não temos como medir com exatidão o número de *tokens*, embora tenhamos a estimativa de que sejam 500 *tokens* por texto, logo, teríamos algo em torno de 143.000 *tokens*. Assim, de acordo com a abordagem histórica, Astrolábio seria considerado um *corpus* de tamanho pequeno-médio.

Na figura a seguir, podemos ter uma visão geral da organização do *corpus*:

Figura 15 – Estrutura geral do *corpus* Astrolábio




Fonte: Elaborada pela autora.

Quanto à tipologia do *corpus*, enquadramos o Astrolábio nas categorias (BERBER SARDINHA, 2004) apresentados no quadro a seguir:

Quadro 3 – Tipologia do *corpus* Astrolábio segundo categorias enumeradas por Berber Sardinha (2004)

| **Tipologia do *Corpus* Astrolábio** | |
|---|---|
| Modo | Escrito – composto por redações produzidas por alunos, em forma de manuscritos. |
| Tempo | Sincrônico – a língua foi analisada em um determinado período, relativo ao período . |
| Seleção | Amostragem – composto por uma amostra finita da linguagem dos alunos de ensino médio das escolas públicas da Microrregião do Litoral de Camocim (CE).<br><br>Estático – os corpus não pode ser atualizado, uma vez que ele foi criado para representar a língua em um momento específico. |
| Conteúdo | Regional ou dialetal – produzidos por alunos de escolas públicas da Microrregião do Litoral de Camocim (CE). |
| Autoria | Falantes de língua nativa |
| Disposição interna | Alinhado – ao lado de cada palavra corrigida ou anotada, pode-se visualizar a forma original, por meio da *tag* <*choice*>. |
| Finalidade | Estudo – serve para pesquisas linguísticas.<br>De treino – serve para o treino e desenvolvimento de ferramentas computacionais. |

Fonte: BERBER SARDINHA, 2004.

*5.3.1.1 Metodologia das oficinas*

A oficina da cidade de Camocim aconteceu entre os dias 15 e 17 de março, no Liceu de Camocim, com carga-horária de 12 horas-aula, e contou com 31 alunos participantes ao total. Eles tiveram a oportunidade de conhecer um pouco sobre a obra do escritor cearense Milton Dias e sobre o gênero crônica. A partir de então, foram incentivados a produzir textos de cunho autobiográfico (embora o tema e o gênero fossem livres).

No mesmo período e com igual carga-horária e com o número de 35 participantes,

foi realizada a oficina de Barroquinha, na EEM. Jaime Laurindo da Silva. Os estudantes conheceram um pouco do universo do poeta Horácio Dídimo e incentivados a produzir textos do gênero poesia (embora o tema e o gênero também fossem livres).

Entre os dias 19 e 20 de abril de 2012, aconteceu  a oficina de Jijoca na EEM José Teixeira de Albuquerque. Os gêneros conto e crônica foram abordados segundo a obra dos autores Moreira Campos e Milton Dias. Também falou-se sobre poesia e diversos poetas. A carga horária, foi igual às demais e o número de participantes foi 21. Do mesmo modo, os alunos tiveram liberdade de escolha do tema e gênero de suas redações.

*5.3.1.2 Limpeza dos arquivos*

Um arquivo de imagem, dependendo do modo como foi capturado (tipo de equipamento, se utilizou a fonte de luz adequada etc.) e de sua resolução, permite a visualização de todas as informações constantes em um documento.

Todavia, uma imagem não possui dados manipuláveis. Assim, se desejarmos extraí-los, necessitamos utilizar um equipamento de OCR (o que pode ser mais trabalhoso, por razões já apontadas neste trabalho) ou digitá-los. De outro modo, uma transcrição que atenda aos nossos interesses não é tarefa simples. Vejamos adiante uma opinião acerca do problema, igualmente enfrentado na construção do Corvo, um *corpus* formado por redações do ENEM:

> Veja-se um fato que ilustra esse tipo de decisão. Compreender que na transposição de registro não se interpreta somente a grafia, mas uma diversidade de informações possíveis (e mesmo comuns) dentro do manuscrito não nos afasta do problema de que não há absoluta correspondência entre os recursos do manuscrito e os do registro eletrônico escrito. Uma flecha, por exemplo, que no manuscrito indica segmentos textuais que devem ser lidos na direção apontada, é um tipo de informação que não se reproduz sem desafios num editor de texto. Num caso desses, o leitor tem de tomar decisões e, como resultado teremos um texto (eletrônico) que será, sempre, uma interpretação (do manuscrito). Ora, de tudo que pode ser "lido" do TF, o digitador do Corvo esteve preparado para apreender apenas uma parte. Tomemos o exemplo da flecha, novamente. A informação lida com esse indicador não foi reproduzida no texto do corpus. Isso porque buscamos um texto corrido, linear, efetivamente "escrito". Temos, aqui, um desses critérios de observação com decorrente rejeição de transposição (PINHEIRO, 2007, p. 87).

Assim, ao final das oficinas, recolhemos o material produzido (manuscritos, com exceção de um único texto, *cam_TCS_f_18_01.jpg*, que já havia sido digitado pela autora). Após essa etapa, todos os textos foram digitados por uma equipe de 04 pessoas, organizadores

da antologia *Temperos Literários 2* (AZIGON et. al., 2012). Conforme já dissemos, o objetivo desta fase, a princípio, era unicamente a seleção e edição da referida antologia. Por causa disso, não fizemos a anotação das etiquetas XML nessa fase inicial, além disso, não haveria tempo hábil para capacitar a equipe para a etiquetagem com XML, seguindo as diretrizes da TEI P5.

Então, os arquivos foram gerados em um editor de texto e salvos em um formato *.txt*., a partir da transcrição conservadora, ou seja, reproduzindo-os da forma mais fiel possível ao original, mesmo sem as referências ao estilo e outros aspectos que só são possíveis de descrever através de uma metalinguagem. Também procuramos manter a mesma disposição gráfica das palavras dentro da página por meio de recuos e espaçamentos, conforme trecho ilustrado a seguir :

Eu desisto - Dos conselhos de pessoas que querem que
eu seja do jeito delas.
----------verso da página----------
Eu documentaria para a humanidade - A criação do mundo.

Como se vê, não foi utilizada uma metalinguagem XML para descrever que, após a palavra *delas*, a sentença seguinte foi escrita no verso da folha. Esse tipo de anotação, além de não ser uma informação padrão, "suja" o texto, ou seja, atrapalha o resultado final do processamento quando o arquivo serve de *input* para o etiquetador. Portanto, nosso primeiro passo foi formatar os textos a fim de prepará-los para as primeiras anotações. Em outras palavras, tivemos que "limpar" informações e espaçamentos, inserindo um espaço em branco entre os parágrafos, pois este é o parâmetro utilizado pelo etiquetador para fazer o reconhecimento dos mesmos. Retomando o exemplo anterior, obtivemos o seguinte:

Eu desisto - Dos conselhos de pessoas que querem que eu seja do jeito delas.

Eu documentaria para a humanidade - A criação do mundo.

Como os textos originais não dispunham de etiquetas XML, não tivemos como limpar os cabeçalhos automaticamente.

*5.3.1.3 Procedimentos para a nomeação*

A nomeação dos arquivos de um corpus é de máxima importância. Isso porque, além de facilitar a criação de *subcorpus*, permite recuperar as informações de forma mais

rápida e eficaz. Assim sendo, não pode ser feita de modo aleatório.

Para este projeto, decidimos nomear os arquivos de modo que algumas variáveis sociolinguísticas ficassem explícitas no nome, conforme orientação a seguir:

Quadro 4 – Esquema para nomeação de arquivos

<table>
<tr><th>Cidade</th><th>Autor</th><th>Sexo</th><th>Idade</th><th>Produção</th></tr>
<tr><td>bar (Barroquinha)</td><td rowspan="3">Usar as 03 primeiras iniciais em letras maiúsculas.</td><td>m (masculino)</td><td rowspan="3">Indicar idade em algarismos arábicos.</td><td rowspan="3">Colocar o número do texto, conforme disposto dentro da pasta do autor</td></tr>
<tr><td>cam (Camocim)</td><td>f (feminino)</td></tr>
<tr><td>jij (Jijoca)</td><td></td></tr>
<tr><td colspan="5">Modelo</td></tr>
<tr><td colspan="5">cidade_ autor_sexo_idade_nº do texto (nome da pasta)<br>cidade_ autor_sexo_idade_nº do texto. txt (arquivos de texto)<br>cidade_ autor_sexo_idade_nº do texto. jpg (arquivos de imagem)<br>cidade_ autor_sexo_idade_nº do texto.01.xml (arquivo XML gerado na 1ª fase de anotação – Astro)<br>cidade_ autor_sexo_idade_nº do texto.02.xml (arquivo XML gerado na 2ª fase de anotação – Manual)</td></tr>
</table>

Fonte: Elaborado pela autora.

O quadro acima é autoexplicativo, mas devemos ficar atentos para alguns casos que fogem à regra. Por exemplo, em algumas situações, verificamos que o nome do autor é composto apenas por duas palavras, portanto, devemos utilizar apenas as duas letras iniciais. Exemplo:

*cam_LS_f_16_01.txt*

No caso dos arquivos de imagem, o caso também é um pouco diferente. Quando fazemos a leitura de um documento por meio do *scanner*, representamos as duas dimensões da folha (frente e verso) em arquivos distintos. Nos exemplo a seguir, verificamos que, após o número correspondente ao texto, vemos mais um dígito, que indica a página correspondente. Como o texto foi escrito em 02 folhas, então temos 04 páginas:

*cam_ABS_m_18_01_1.jpg*
*cam_ABS_m_18_01_2.jpg*
*cam_ABS_m_18_01_3.jpg*

*cam_ABS_m_18_01_4.jpg*

Também houve casos onde apareceram mais de um texto em um único arquivo:

*cam_FAA_f_18_07_e_08.jpg*

E um único caso de um texto escrito por mais de uma autora, mas, coincidentemente, todas tinham a mesma idade:

*jij_HFS_LAN_RFS_f_17_.txt*

Em alguns casos, a idade também não foi informada[49]. Nestes casos, usamos a letra x:

*bar_AFM_f_x_02.jpg*

### *5.3.2 Preservação da identidade dos envolvidos na pesquisa*

A identidade dos autores foi resguardada, porém, foram explicitados os dados que permitem identificar as variações diatópicas, diafásicas e diastráticas. Nos arquivos do tipo imagem, também tivemos o cuidado de ocultar assinaturas e menções ao próprio nome do autor, utilizando tarja preta, semelhante a que se segue:

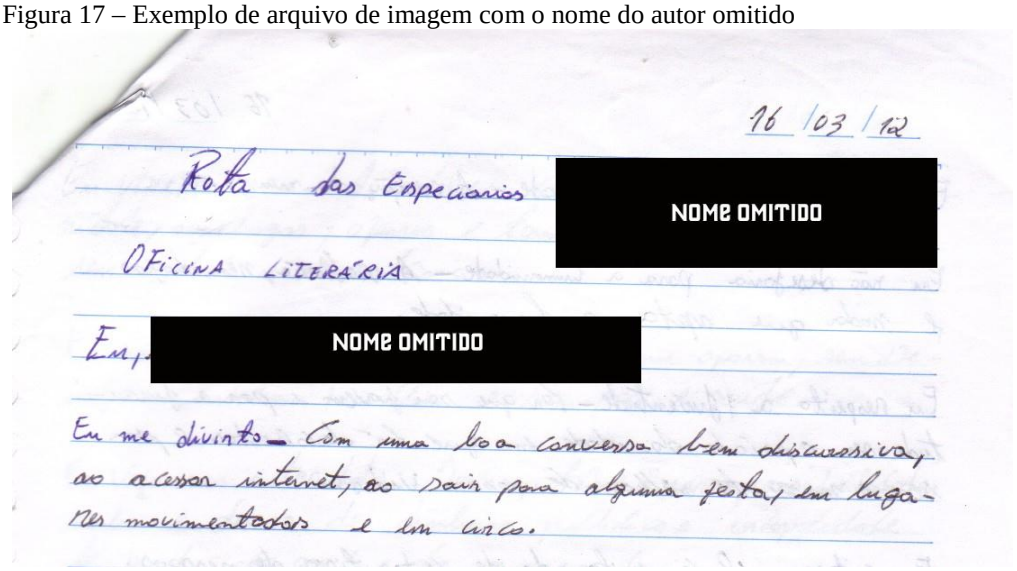
Figura 16 – Nome omitido


Fonte: Elaborado pela autora.

Para que essa ação tivesse realmente o objetivo de preservar a identidade dos envolvidos, além do arquivo de imagem (manuscritos originais), também foi necessário alterarmos o texto anotado em XML, bem como .txt original. Vejamos o exemplo do autor "cam_ABS_m_18_01.jpg":

[49]Sabemos, contudo, que os autores encontram-se na faixa etária compreendida entre 15 e 21 anos.

Figura 17 – Exemplo de arquivo de imagem com o nome do autor omitido




Fonte: Elaborado pela autora.

A seguir, vemos como as informações acima foram dispostas no "cam_ABS_m_18_01.txt":

Quadro 5 – Exemplo de arquivo de texto com o nome do autor omitido

16/03/12

Rota das Especiarias Nome omitido

OFICINA LITERÁRIA

Eu, Nome omitido

Eu me divirto- Com uma boa conversa bem discursiva,
ao acessar internet, ao sair para alguma festa, em lugares
movimentados e em circo.

Fonte: Elaborado pela autora.

Convém ainda mencionar que este trabalho será submetido à apreciação da Comissão de Ética da UFC.

## 5.4 A escolha do etiquetador

Conforme demonstramos no capítulo sobre etiquetagem, o LX-Tagger, construído a partir da arquitetura do MXPOST, tem superado o estado da arte para as etiquetadores em

Português. Contudo, não achamos que a sua escolha seja a mais adequada ao nosso projeto, pois a licença do LX-Tagger impede a criação de produtos derivados. Em razão disso, decidimos não utilizá-lo na criação do *corpus*.

Em recente trabalho (ALENCAR;SANTIAGO;SANTOS, 2011), foram verificados os desempenhos dos etiquetadores AeliusBRUBT, construído utilizando as ferramentas do NLTK, e AeliusHunPos, criado a partir do HunPos. Ambos foram treinados no TBCHP. Para a análise, foram utilizados 1994 *tokens*, extraídos de textos de divulgação científica. Dentre os resultados obtidos, verificou-se que o AeliusHunPos teve uma melhor acurácia: 92,78%, enquanto o AeliusBRUBT obteve 91,83%.

Portanto, escolhemos o etiquetador Aelius criado a partir do HunPos. Porém, Pensamos que o AeliusHunPosMacMorpho seja o mais adequado ao nosso projeto por duas importantes razões. Em primeiro lugar, porque o MAC-Morpho é formado por textos contemporâneos, ao contrário do *corpus* TBCHP. Em segundo lugar, porque a licença do etiquetador HunPos permite que a criação de produtos derivados, inclusive para fins comerciais.

## 5.5 Preservação de informações por meio da TEI P5

A TEI P5 possui uma vasta codificação que nos permitiu codificar com detalhes os textos que foram analisados.

### *5.5.1 Anotação estrutural*

A anotação estrutural permite a marcação dos dados internos e externos do texto. Os dados externos referem-se aos metadados textuais (dados bibliográficos, catalogação, autoria etc). Os dados internos, por sua vez, compreendem a marcação da estrutura geral e da estrutura de subparágrafos (capítulos, título, parágrafo, sentenças, palavras etc) (ALUÍSIO; ALMEIDA, 2006). Uma das vantagens da anotação estrutural é viabilizar a recuperação do texto e facilitar a criação de *subcorpus* (ALUÍSIO; ALMEIDA, 2006).

Como dissemos anteriormente, a anotação estrutural pode ser realizada por

algumas linguagens de marcação, como a HTML e a XML. Decidimos usar esta última, pois é a prescrita pela codificação que adotamos, a TEI P5.

Entretanto, o esquema da TEI não é o único. O TBCHP, por exemplo, tem um esquema próprio de codificação das etiquetas XML. Há, inclusive, sistemas de codificação derivados da TEI, como é o caso do XCES[50]. Porém, decidimos pela TEI P5, conforme razões já apresentadas.

### *5.5.2 Anotação da correção e da variação*

A TEI P5, para os manuscritos, prevê a *tag* <*choice*>, conforme exemplo acima. Com isso, pudemos visualisar as duas formas, a original <*sic*> *caza* <*/sic*> e a corrigida <*corr*> *casa* <*/corr*>.

Também podemos marcar os regionalismos e algumas palavras típicas da linguagem dos *chats*, como "vc", "kkkkkkk" e "fika", bastante recorrente nos textos em análise, através das tags <*orig*> e <*reg*>, dentro da *tag* <*choice*>. Esta última também nos permite agrupar as *tags* <*unclear*>, <*add*> e <*del*>, com a função de marcar algum termo ilegível, para acrescentar ou deletar termos necessários à correção, respectivamente.

A *tag* <*choice*> permite que reunamos, no *corpus* Astrolábio, tanto a forma original quanto a forma corrigida.

## 5.6 O sistema de codificação Unicode UTF-8

Para que tenhamos uma melhor compreensão do sistema de codificação Unicode UTF-8, precisamos entender um pouco sobre lógica computacional e o sistema binário.

Quando acionamos um teclado, na verdade, estamos emitindo uma mensagem codificada para o computador, que será interpretada a partir do sistema binário. Este sistema difere daquele com o qual estamos mais habituados: o sistema indo-arábico, composto por algarismos de 0 a 9. Neste último sistema, podemos contar os números usando as unidades,

[50]http://www.xces.org/

dezenas, centenas e daí por diante. Por exemplo, o número 472 pode ser igualmente representado da seguinte forma:

| Centenas | Dezenas | Unidades |
|---|---|---|
| 4 | 7 | 2 |

O exemplo acima também pode ser escrito assim:

| $100\ (10^2)$ | $10\ (10^1)$ | $1\ (10^o)$ |
|---|---|---|
| 4 | 7 | 2 |

$472= (4x10^2) + (7 \times 10^1) + (2 \times 10^o)$

Porém, o sistema binário, como o próprio nome sugere, trabalha apenas com dois algarismos: 0 e 1. Então, ao invés de dividirmos os números pela unidade decimal, 10, dividiremos por 2. Tomando o exemplo acima, teremos a sua representação no sistema binário procedendo da seguinte forma:

| | |
|---|---|
| 472 / 2 = 236 | 0 |
| 236 / 2 = 118 | 0 |
| 118 / 2 = 59 | 0 |
| 59 / 2 = 29 | 1 |
| 29 / 2 = 14 | 1 |
| 14 / 2 = 7 | 0 |
| 7 / 2 = 3 | 1 |
| 3 / 2 = 1 | 1 |

Dividimos o número 472 por 2, pegamos o resultado da divisão e dividimos novamente por 2, sucessivas vezes, até obtermos o quociente 1. Ao lado direito da tabela, temos o resto da divisão, que representa o algarismo em binário. Agora, para obtermos o

número binário, basta escrevermos o algarismo 1, seguido da sequência de 0 e 1, a partir do resultado da última divisão, seguida da penúltima e assim por diante, até chegar à primeira. Em outras palavras, o número 472, no sistema binário, equivale a 111011000.

E se desejarmos o inverso: temos um número em representação binária e quisermos vê-lo em sistema decimal? É bem simples: contamos a posição ocupada pelos algarismos, da direita para a esquerda. Em seguida, devemos multiplicá-lo por 2 elevado à potência que será dada de acordo com a posição do algarismo (a primeira posição equivale a 0, segunda a 1, a terceira a 2 e assim por diante) e depois somamos todos os resultados. Então, teremos o seguinte:

| |
|---|
| 1 x 2^8 = 256 |
| 1 x 2^7 = 128 |
| 1 x 2^6 = 64 |
| 0 x 2^5 = 0 |
| 1 x 2^4 = 16 |
| 1 x $2^3$ = 8 |
| 0 x $2^2$ = 0 |
| 0 x $2^1$ = 0 |
| 0 x $2^0$ = 0 |

256 + 128 + 64 + 0 + 16 + 8 + 0 + 0 + 0 + 0 = 472.

O processamento dos dados binários, por sua vez, é feito através de circuitos eletrônicos que utilizam em seu funcionamento a lógica booleana (toma os parâmetros verdadeiro/falso, positivo/negativo). Esse sistema possibilita que o computador interprete não apenas números, mas sistemas mais complexos, como textos e imagens.

Cada algarismo no sistema binário corresponde a 1 *bit* (*binary digit*) e um conjunto de 8 *bites* corresponde a 1 *byte*. Tomando por base o que expusemos anteriormente, temos o seguinte: cada *bit* pode representar dois valores: 0 ou 1. Então, em 1 único *byte*, podemos representar até 256 caracteres (2^8).

A partir desse raciocínio, compreendamos que os caracteres dos mais variados tipos de escrita também necessitam de um sistema binário para serem interpretados. Assim, surgiu a necessidade de padronizar esses códigos para que não houvesse conflitos em sua leitura por computadores distintos e, por isso, vários sistemas de codificação foram criados.

Talvez o mais difundido dentre eles seja o *American Standard Code for Information Interchange*, ASCII, criado na década de 60.

O ASCII trabalha com a capacidade de armazenamento de 1 *byte*, sendo que apenas 7 *bits* são utilizados para o armazenamento de caracteres. Portanto, temos aí a possibilidade de representação de 128 caracteres (2^7 = 128), dentre eles, 33 são caracteres de controle: representam comandos para o computador, restando apenas 95 caracteres para o armazenamento de caracteres alfanuméricos. Consequentemente, o espaço é muito limitado e priorizou-se o armazenamento de caracteres pertencentes ao alfabeto inglês. Em razão disso, alguns caracteres especiais, como os diacríticos, podem não ser corretamente decodificados. A seguir, podemos visualizar alguns caracteres do ASCII e suas correspondentes representações binária, decimal e hexadecimal:

Figura 18 – Comparação entre sistema binário, decimal e hexadecimal

| Binário | Decimal | Hexa | Glifo |
|---|---|---|---|
| 0010 0000 | 32 | 20 | |
| 0010 0001 | 33 | 21 | ! |
| 0010 0010 | 34 | 22 | " |
| 0010 0011 | 35 | 23 | # |
| 0010 0100 | 36 | 24 | $ |
| 0010 0101 | 37 | 25 | % |
| 0010 0110 | 38 | 26 | & |
| 0010 0111 | 39 | 27 | ' |
| 0010 1000 | 40 | 28 | ( |
| 0010 1001 | 41 | 29 | ) |
| 0010 1010 | 42 | 2A | * |
| 0010 1011 | 43 | 2B | + |
| 0010 1100 | 44 | 2C | , |
| 0010 1101 | 45 | 2D | - |
| 0010 1110 | 46 | 2E | . |
| 0010 1111 | 47 | 2F | / |
| 0011 0000 | 48 | 30 | 0 |
| 0011 0001 | 49 | 31 | 1 |
| 0011 0010 | 50 | 32 | 2 |
| 0011 0011 | 51 | 33 | 3 |
| 0011 0100 | 52 | 34 | 4 |
| 0011 0101 | 53 | 35 | 5 |
| 0011 0110 | 54 | 36 | 6 |
| 0011 0111 | 55 | 37 | 7 |
| 0011 1000 | 56 | 38 | 8 |
| 0011 1001 | 57 | 39 | 9 |
| 0011 1010 | 58 | 3A | : |
| 0011 1011 | 59 | 3B | ; |
| 0011 1100 | 60 | 3C | < |
| 0011 1101 | 61 | 3D | = |
| 0011 1110 | 62 | 3E | > |
| 0011 1111 | 63 | 3F | ? |

| Binário | Decimal | Hexa | Glifo |
|---|---|---|---|
| 0100 0000 | 64 | 40 | @ |
| 0100 0001 | 65 | 41 | A |
| 0100 0010 | 66 | 42 | B |
| 0100 0011 | 67 | 43 | C |
| 0100 0100 | 68 | 44 | D |
| 0100 0101 | 69 | 45 | E |
| 0100 0110 | 70 | 46 | F |
| 0100 0111 | 71 | 47 | G |
| 0100 1000 | 72 | 48 | H |
| 0100 1001 | 73 | 49 | I |
| 0100 1010 | 74 | 4A | J |
| 0100 1011 | 75 | 4B | K |
| 0100 1100 | 76 | 4C | L |
| 0100 1101 | 77 | 4D | M |
| 0100 1110 | 78 | 4E | N |
| 0100 1111 | 79 | 4F | O |
| 0101 0000 | 80 | 50 | P |
| 0101 0001 | 81 | 51 | Q |
| 0101 0010 | 82 | 52 | R |
| 0101 0011 | 83 | 53 | S |
| 0101 0100 | 84 | 54 | T |
| 0101 0101 | 85 | 55 | U |
| 0101 0110 | 86 | 56 | V |
| 0101 0111 | 87 | 57 | W |
| 0101 1000 | 88 | 58 | X |
| 0101 1001 | 89 | 59 | Y |
| 0101 1010 | 90 | 5A | Z |
| 0101 1011 | 91 | 5B | [ |
| 0101 1100 | 92 | 5C | \ |
| 0101 1101 | 93 | 5D | ] |
| 0101 1110 | 94 | 5E | ^ |
| 0101 1111 | 95 | 5F | _ |

| Binário | Decimal | Hexa | Glifo |
|---|---|---|---|
| 0110 0000 | 96 | 60 | ` |
| 0110 0001 | 97 | 61 | a |
| 0110 0010 | 98 | 62 | b |
| 0110 0011 | 99 | 63 | c |
| 0110 0100 | 100 | 64 | d |
| 0110 0101 | 101 | 65 | e |
| 0110 0110 | 102 | 66 | f |
| 0110 0111 | 103 | 67 | g |
| 0110 1000 | 104 | 68 | h |
| 0110 1001 | 105 | 69 | i |
| 0110 1010 | 106 | 6A | j |
| 0110 1011 | 107 | 6B | k |
| 0110 1100 | 108 | 6C | l |
| 0110 1101 | 109 | 6D | m |
| 0110 1110 | 110 | 6E | n |
| 0110 1111 | 111 | 6F | o |
| 0111 0000 | 112 | 70 | p |
| 0111 0001 | 113 | 71 | q |
| 0111 0010 | 114 | 72 | r |
| 0111 0011 | 115 | 73 | s |
| 0111 0100 | 116 | 74 | t |
| 0111 0101 | 117 | 75 | u |
| 0111 0110 | 118 | 76 | v |
| 0111 0111 | 119 | 77 | w |
| 0111 1000 | 120 | 78 | x |
| 0111 1001 | 121 | 79 | y |
| 0111 1010 | 122 | 7A | z |
| 0111 1011 | 123 | 7B | { |
| 0111 1100 | 124 | 7C | \| |
| 0111 1101 | 125 | 7D | } |
| 0111 1110 | 126 | 7E | ~ |

Fonte: http://pt.wikipedia.org/wiki/ASCII.

Visando atender à demanda por mais caracteres, surgiu o ASCII estendido que, além dos 128 caracteres do ASCII, trouxe mais 128 caracteres, perfazendo 256 no total. Entretanto, este número ainda é insuficiente para dar conta dos mais variados sistemas de escrita do mundo.

Para resolver a questão, o “Unicode fornece um número único para cada caractere,

não importa a plataforma, não importa o programa, não importa a língua.”[51]. Assim, este sistema usa como ponto de partida a codificação ASCII e busca suprir as suas lacunas, como a possibilidade de codificar apenas caracteres minúsculos de A a Z, enquanto o Unicode promete a codificação de todas os sistemas de escrita do mundo, com a impressionante possibilidade de codificação de mais de um milhão de caracteres. Em razão disso, vem sendo adotado por empresas de referência, como a Apple, Microsoft e Oracle, além dos padrões modernos como a XML e Java.

Para que tenhamos noção da amplitude do sistema de codificação do Unicode, diante do sistema ASCII, vide figura a seguir:

Figura 19 – Comparação entre Unicode e ASCII

ASCII/8859-1 Text

| | |
|---|---|
| A | 0100 0001 |
| S | 0101 0011 |
| C | 0100 0011 |
| I | 0100 1001 |
| I | 0100 1001 |
| / | 0010 1111 |
| 8 | 0011 1000 |
| 8 | 0011 1000 |
| 5 | 0011 0101 |
| 9 | 0011 1001 |
| - | 0010 1101 |
| 1 | 0011 0001 |
| | 0010 0000 |
| t | 0111 0100 |
| e | 0110 0101 |
| x | 0111 1000 |
| t | 0111 0100 |

Unicode Text

| | |
|---|---|
| A | 0000 0000 0100 0001 |
| S | 0000 0000 0101 0011 |
| C | 0000 0000 0100 0011 |
| I | 0000 0000 0100 1001 |
| I | 0000 0000 0100 1001 |
| | 0000 0000 0010 0000 |
| 天 | 0101 1001 0010 1001 |
| 地 | 0101 0111 0011 0000 |
| | 0000 0000 0010 0000 |
| س | 0000 0110 0011 0011 |
| ل | 0000 0110 0100 0100 |
| ا | 0000 0110 0010 0111 |
| م | 0000 0110 0100 0101 |
| | 0000 0000 0010 0000 |
| α | 0000 0011 1011 0001 |
| ≰ | 0010 0010 0111 0000 |
| γ | 0000 0011 1011 0011 |

Fonte: http://www.unicode.org/versions/Unicode6.2.0/ch01.pdf.

A versão *Unicode Standard 6.2.* contém 110.117[52] caracteres dos mais variados tipos de escrita do mundo, dentre as quais: o alfabeto europeu, escritas do Oriente Médio, da Ásia e da África. Ainda possui 74. 616 caracteres para representação de ideogramas.

Outro importante aspecto é que a arquitetura do Unicode considera não apenas o caractere, mas também o processamento do texto, utilizado pela grande maioria dos

[51]http://www.unicode.org/standard/translations/portuguese.html
[52]http://www.unicode.org/versions/Unicode6.2.0/ch01.pdf p. 2

computadores, como: renderização[53] de caracteres (levando em conta as ligaduras, dentre outros), quebra de linha (sem entrar em conflito com as questões de hifenização), impasse de desenhar um conjunto universal de caracteres, partindo do ponto de que não há um padrão universal para unidades fundamentais do texto, uma vez que processamento do texto e as suas particularidades dentro de cada língua:

> For example, in traditional German orthography, the letter combination “ck” is a text element for the process of hyphenation (where it appears as “k-k”), but not for the process of sorting. In Spanish, the combination “ll” may be a text element for the traditional process of sorting (where it is sorted between “l” and “m”), but not for the process of rendering. In English, the letters “A” and “a” are usually distinct text elements for the process of rendering, but generally not distinct for the process of searching text. The text elements in a given language depend upon the specific text process; a text element for spell-checking may have different boundaries from a text element for sorting purposes. For example, in the phrase “the quick brown fox,” the sequence “fox” is a text element for the purpose of spell-checking.
> (http://www.unicode.org/versions/Unicode6.2.0/ch02.pdf, p. 8)

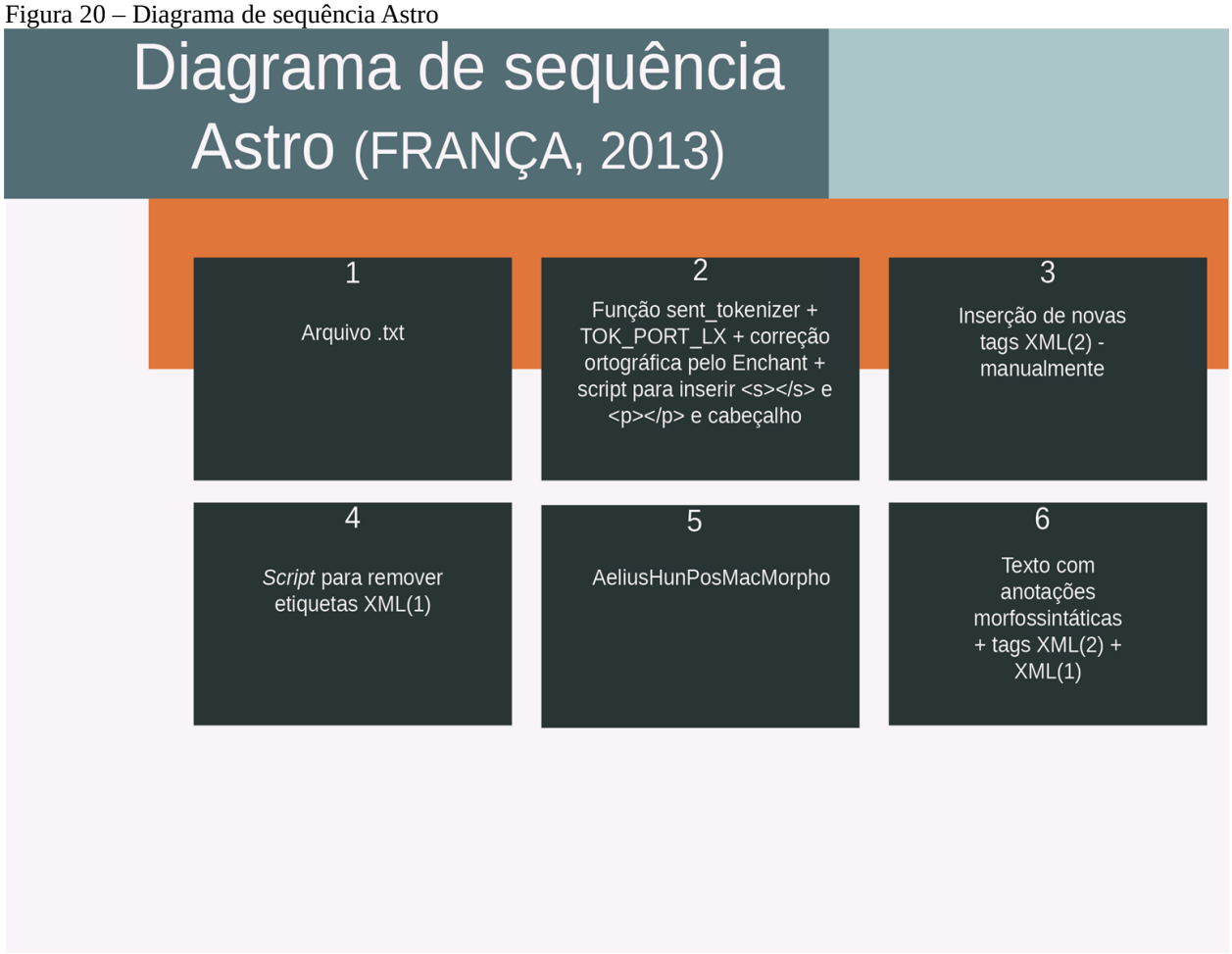
## 5.7 Integração automática entre os diversos tipos de anotação por meio do Astro

Uma das nossas dificuldades foi integrar, em um único arquivo, anotação morfossintática e tags XML. Isso por que o arquivo que serve de input para o etiquetador não pode conter etiquetas XML. Em razão disso, o software Astro (FRANÇA, 2013) foi especialmente desenvolvido para este trabalho.

A figura a seguir descreve as etapas realizadas a partir do Astro. Assim, conseguimos um *corpus* com etiquetas XML e morfossintáticas integradas.

[53]Renderização é o resultado final do processamento, ou seja, é a forma como o computador interpreta os dados que lhe foram dados como *input*. Por exemplo, o resultado final da interpretação do número binário 0110 0001, de acordo com o padrão ASCII, dever ser o caractere “a”. Em suma, o computador recebe o número 0110 0001 e interpreta para o usuário (renderiza) como “a”.

Figura 20 – Diagrama de sequência Astro




Fonte: Elaborada pela autora.

Na primeira etapa, fazemos a limpeza dos textos conforme os procedimentos descritos neste capítulo. Em seguida, o arquivo serve de *input* para o Astro. A partir do texto em .txt, Astro gera um novo arquivo XML, contendo uma estrutura de documento e um cabeçalho de acordo com a TEI P5. Aqui, ele utiliza a função sent_tokenizer para o reconhecimento das sentenças e parágrafos e insere as respectivas etiquetas. A função TOK_PORT_LX é usada para toquenizar as palavras, a fim de que sejam verificadas pela biblioteca de correção Enchant. Onde são encontrados erros ortográficos, automaticamente são inseridas *tags* <*choice*> para indicá-los e é feita a correção.

Em seguida, novos detalhes são inseridos manualmente, bem como os metadados e a revisão da correção automática feita pelo Enchant. Na próxima etapa, esse arquivo XML com novas anotações manuais passa novamente pelo Astro, que extrai as sentenças a partir das *tags* <*s*>. Estas sentenças, por sua vez, servem de input para a etiquetagem automática pelo etiquetador AeliusHunPosMacMorpho. O resultado final é um arquivo XML contendo tanto etiquetas morfossintáticas quanto XML.

Todas essas funções podem ser executadas a partir de uma interface amigável, construída por linha de comando em Python:

Figura 21 – Interface do Astro

```
Python 2.7.3 (default, Sep 26 2012, 21:53:58)
[GCC 4.7.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
------------------------------------------------------------
Astro 1.0
------------------------------------------------------------
      menu de opcoes

(1) - anotacao inicial
(2) - anotacao final
(0) - sair

digite a opcao: 1
------------------------------------------------------------
Astro 1.0
------------------------------------------------------------
      menu de opcoes

(1) - anotacao inicial
(2) - anotacao final
(0) - sair
------------------------------------------------------------

(x)  - anotacao inicial
( )  - opcoes
(1)  - continuar selecao
(2)  - modificar pasta
(3)  - modificar todos

opcao: 3
informe a pasta selecionada para a analise

pasta: /home/katiusha/Dropbox/Astro/bar_LQS_f_x/
pasta selecionada: /home/katiusha/Dropbox/Astro/bar_LQS_f_x/

arquivo: bar_LQS_f_x_01.txt
arquivo selecionado: bar_LQS_f_x_01.txt
```

Fonte: Elaborado pela autora.

Apesar das vantagens apresentadas pelo Astro, durante os primeiros testes, verificamos alguns problemas que são enumerados no quadro abaixo. Alguns melhoramentos já foram feitos na versão 1.1. do Astro. Porém, ainda não tivemos tempo hábil de testar essa ferramenta.

Quadro 6 – Problemas encontrados na versão 1.0 do Astro

| **Problemas encontrados na versão 1.0 do Astro** | | |
|---|---|---|
| **Problema** | **Observações** | **Grau de relevância do problema** |
| Embora já possamos inserir automaticamente as *tags* <*p*> e <*s*>, não há numeração automática de parágrafos e sentenças, apenas de palavras (*tokens*). | Ao anotar no formato XML, o Aelius insere a numeração de parágrafos e sentenças. | Médio. |
| Não há inserção automática de <*lg*> (estrofe) e <l> (verso) para o gênero poesia. | | Leve. |
| Quando usamos um texto todo em caixa alta como *input*, obtivemos a maioria das palavras identificadas incorretamente como Nomes Próprios (vide anexo). | O Aelius já dispõe de um algorítimo para identificação de Nomes Próprios. Porém, o problema apontado pode ser útil para incrementá-lo. Por hora, a solução é evitar esse tipo de *input*. | Leve. |
| Não podemos inserir a anotação XML de uma quebra de linha provocada por uma palavra com falso hífen, pois isso implicaria em segmentar uma palavra em dois *tokens*, dificultando a anotação morfossintática. | Já existe tecnologia para solucionar esse problema. Como dissemos neste trabalho, para resolver problemas da toquenização de falsos hífens, o LX-Tagger recompõe o *token*, eliminando o hífen, e verifica se a palavra composta existe em uma lista. Caso positivo, trata-se de um falso hífen e o mesmo é eliminado, caso contrário, o hífen deve permanecer. | Médio. |
| Na fase de anotação final do Astro, as *tags* <*sic*> e <*corr*> não estão corretamente inseridas dentro de <*choice*>. | | Grave. |
| Fora <*sic*> e <*corr*>, ainda não foram previstas outras *tags* filhas de <*choice*>: <*reg*>, <*orig*>, <*del*>, <*add*> *e* <*distinct*>. | | Grave. |

Fonte: Elaborado pela autora.

## 5.8 Distribuição do *corpus*

O *corpus* Astrolábio já se encontra parcialmente disponível na *web*. Ele pode ser acessado através do sítio eletrônico do projeto Rota das Especiarias.

Quadro 7 – Astrolábio no sítio eletrônico Rota das Especiarias


Fonte: www.rotadasespeciarias.art.br.

Também submeteremos o *corpus* Astrolábio para depósito no OTA.

# 6 CONSIDERAÇÕES FINAIS

Ao longo deste trabalho, percebemos que são inúmeras as possibilidades de aplicação da TEI, tanto no universo editorial como na Tecnologia da Informação e na Linguística. Metaforicamente falando, conhecer o universo da TEI P5 foi como ganhar de presente um canivete suíço para a pesquisa linguística.

Porém, ao mesmo tempo em que o presente nos trouxe encantamentos, também trouxe alguns problemas relacionados ao foco de nossa pesquisa. Como já foi dito exaustivamente aqui, a TEI P5 permite a anotação de, praticamente, qualquer tipo de texto em um nível muito profundo de detalhamento. Então, tivemos dificuldade em encontrar um nível que pudesse superar os *corpora* já existentes em Português, mas que também exequível em um curto período de tempo.

Com isso, optamos por fazer uma anotação mais simplificada, a partir da transcrição do texto sem a preocupação com a sua disposição gráfica mais complexa de alguns poemas que foram escritos utilizando muitos recursos da poesia concreta. Lamentamos muito essa decisão, pois esse tipo de estudo pode auxiliar no desenvolvimento de leitura de imagens por meio de programas *text-to-speech* (sintetizadores de fala).

Todavia, acreditamos que a nossa decisão não compromete o resultado final, uma vez que, juntamente aos textos em *.txt*, também disponibilizamos os arquivos em formato de imagem. Entretanto, tal opção metodológica não nos impede de, oportunamente, retomarmos a empreitada para enriquecer os textos com novas anotações XML.

Os testes para a integração entre os diferentes tipos de anotação, por sua vez, comprometeram a grande maioria do tempo dedicado a esta pesquisa. Em razão disso, ao contrário do que planejamos, o Astrolábio ainda não está totalmente concluído. Por outro lado, durante esse tempo, tivemos grandes reflexões acerca da Linguística de Corpus e da Linguística Computacional. Também convém dizer que essas reflexões levaram-nos ao desenvolvimento do Astro, que apesar de ainda estar em fase de testes, superou as nossas expectativas em termos de funcionalidades.

Portanto, concluímos que a atividade de construção de um *corpu*s é tão enriquecedora para o desenvolvimento das pesquisas em Linguística Computacional, que acaba por tornar-se tão ou mais importante que o próprio *corpus*.

# REFERÊNCIAS

AFONSO, Susana; FREITAS, Cláudia. Bíblia Florestal: Um manual lingüístico da Floresta Sintá(c)tica. Versão 8.0. Disponível em: <http://www.linguateca.pt/Floresta/BibliaFlorestal/completa.html>. Acesso em: 10 set. 2013.

ALENCAR. Aelius: uma ferramenta para anotação automática de corpora usando o NLTK. **ELC 2010 – IX Encontro de Linguística de Corpus**, PUCRS, Porto Alegre, 8 e 9 de outubro de 2010. Disponível em: <http://corpuslg.org/gelc/elc2010.php>.

_____; SANTIAGO, André Chaves; SANTOS, Andréa Feitosa. **Etiquetagem automática de textos de divulgação científica: comparação entre dois etiquetadores**. XXX Encontro de iniciação científica. UFC, 2011. Disponível em: <http://www.prppg.ufc.br/eu2011.ufc.br/Resumos/wrappers/MostrarResumo.php?cpf=30808545353&cod=001>. Acesso em: 02 dez. 2012.

_____; OTHERO, Gabriel de Ávila. **Abordagens computacionais da teoria da gramática**. Campinas, SP: Mercado das Lestras, 2011.

ALUÍSIO, Sandra Maria; ALMEIDA, Gladis Maria de Barcellos. O que é e como se constrói um corpus? Lições aprendidas na compilação de vários corpora para pesquisa linguística. **Caleidoscópio**. São Leopoldo. Vol. 4, n. 3, p. 155-177, set/dez, 2006.

AZIGON, Talles et al. (Orgs.). **Temperos literários 2:** antologia. Fortaleza: Corsário, 2012.

FRANÇA, Mardônio. **Astro**, 2013.

BÜHRIG, KRISTIN / KLICHE, ORTRUN / PAWLACK, BIRTE & MEYER, BERND (2012): The corpus „Interpreting in hospitals“ – possible applications for research and communication trainings. Submitted to: Schmidt, Thomas & Wörner, Kai (eds.): Multilingual Corpora and Multilingual Corpus Analysis. Hamburg Studies in Multilingualism (14). Amsterdam: John Benjamins.

BERBER SARDINHA, T. Linguística DE CORPUS: HISTÓRICO E PROBLEMÁTICA. **D.E.L.T.A.**, Vol. 16, N.º 2, p. 323-367, 2000.

_____. Linguística de corpus. Barueri, São Paulo: Manole, 2004.

BRANCO, António; SILVA, João, 2004. **Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese**. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa and Raquel Silva (orgs.), Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004), Paris, ELRA, ISBN 2-9517408-1-6, pp.507-510.

BIBER, D.; CONRAD, S. e REPPEN, R. 1998. **Corpus linguistics: Investigating language structure and use**. Cambridge University Press, Cambridge.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Toolkit**. [s.l]: [s.n.], 2010. Disponível

em: <http://www.nltk.org> . Acesso em: 30 sep. 2010.

BUHRIG, Kristin; KLICHE, Ortrun; PAWLACK, Birte; MEYER, Bernd (2012). **The corpus Interpreting in hospitals – possible applications for research and communication trainings.** Submitted to: Schmidt, Thomas & Worner, Kai (eds.): Multilingual Corpora and Multilingual Corpus Analysis. Hamburg Studies in Multilingualism (14). Amsterdam: John Benjamins.

CUNHA, Celso; CINTRA, Luís F. Lindley T. **Nova gramática do português contemporâneo**. Rio de Janeiro: Lexikon, 2008. 5.ed.

DAYAN, Peter. **Unsupervised Learning**. In Wilson, RA & Keil, F, editors. The MIT Encyclopedia of the Cognitive Sciences, 1999b. Disponível em: http://www.gatsby.ucl.ac.uk/~dayan/papers/dun99b.html . Acesso em: 28 jan. 2013.

DOMINGUES, M; FAVERO, E; MEDEIROS, I. O Desenvolvimento de um Etiquetador com Alta Acurácia para o Português. **VI Encontro de Lingüística de Corpus**, 06-07/setembro, São Paulo-SP, 2007.

ESPÍNDOLA, Luciana da Silvera Espíndola. **Um Estudo sobre Modelos Ocultos de Markov – Hidden Markov Model.** 2009. Disponível em: <www.inf.pucrs.br/peg/pub/tr/TI1_Luciana.pdf.>
Acesso em: 28 jan. 2013.

GALVES, Charlotte; FARIA, Pablo. 2010. **Tycho Brahe Parsed Corpus of Historical Portuguese**. URL: http://www.tycho.iel.unicamp.br/~tycho/corpus/en/index.html. 2010.

GASPERIN, Varaschin Caroline; LIMA, Vera Lúcia Strube de. **Fundamentos do Processamento Estatístico da Linguagem Natural. T**echnical Report Series, n.021, 2001. Disponível em: <https://docs.google.com/viewera=v&q=cache:LXXPRNJJOtcJ:www3.pucrs.br/pucrs/files/uni/poa/facin/pos/relatoriostec/tr021.pdf+modelos+estat%C3%ADsticos+baseados+em+n-gramas&hl=en&gl=br&pid=bl&srcid=ADGEESgZqanm1IKlZ2yZeLSTw7otHcigSTOlXDNzQZAj_BBg7D2WJpeva8CFlvSvrkdf6tUyRkaLLCAtHhP_CWrr5yyR8y9yUTAQ6—HQrLVto-WMEb1Ty8XW7AzeylLogTcuCUE7UHO&sig=AHIEtbR_ALLPQlVntUfU-aX0PRdvUOnn5Q>. Acesso em: Acesso em: 28 jan. 2013.

KENNEDY, G. 1998. **An Introduction to Corpus Linguistics**. London; New York, Longman.

KISS, Tibor; STRUNK. **Unsupervised Multilingual Sentence Boundary Detection**. Computational Linguistics 32: 485-525, 2006. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.5017&rep=rep1&type=pdf. Acesso em: 28 jan. 2013.

LINGUATECA (2007). Linguateca, centro de recursos -- distribuído -- para o processamento computacional da língua portuguesa, Disponível em: <http://www.linguateca.pt/>. Acesso em: 2 out. 2012.

MIKHEEV, Andrei. **Text Segmentation.** In: MITKOV, R. (Ed.), The Oxford handbook of

computational linguistics. Oxford, Oxford University Press, 2004. p. 219- 232.

MITKOV, R. (Ed.), **The Oxford handbook of computational linguistics**. Oxford, Oxford University Press, 2004. p. 219- 232.

OLIVEIRA, Lúcia Pacheco de. LINGUÍSTICA DE CORPUS: TEORIA, INTERFACES E APLICAÇÕES. **Matraga**, Rio de Janeiro, v.16, n.24, p. 48-76, jan./jun. 2009.

PAIXÃO DE SOUSA, M. C.; KEPLER, F. N.; FARIA, P. **E-dictor: Novas perspectivas na codificação e edição de corpora de textos históricos**. In: VIII Encontro de Linguística de Corpus, 2009, Rio de Janeiro. Resumos, 2009. Disponível em: <http://www.tycho.iel.unicamp.br/~tycho/pesquisa/artigos/PAIXAO-DE-SOUZA_etal-ECL2009.pdf>. Acesso em: 3. nov. 2012.

PERKINS, J. **Python Text Processing with NLTK 2.0 Cookbook**. Birmingham, UK: Packt, 2010. 256 p.

PYTHON Programming Language. Official Website. Disponível em: <http://www.python.org/>. Acesso em: 5. nov. 2012.

HALÁCSY, Péter; KORNAI, András; ORAVECZ, Csaba. 2007. HunPos - an open source trigram tagger In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions. **Association for Computational Linguistics**, Prague, Czech Republic, pages 209—212.

RATNAPARKHI, A. **A Maximum Entropy Part-of-Speech Tagger**. Proc. of the Empirical Methods on Natural Language Processing, ACL, 133-142. Disponível em: <http://www.google.com.br/url?sa=t&rct=j&q=a%20maximum%20entropy%20part-of-speech%20tagger&source=web&cd=1&cad=rja&ved=0CDgQFjAA&url=http%3A%2F%2Facl.ldc.upenn.edu%2FW%2FW96%2FW96-0213.pdf&ei=saUTUbWsDZKy0AGb2IGwCw&usg=AFQjCNFSxT0xKbLOatnCNP9nQ-uvcrohiQ&bvm=bv.42080656,d.dmQ>. Acesso em: 5. dez. 2012.

RAMALHO, José Carlos. **Anotação estrutural de documentos e sua semântica : especificação da sintaxe, semântica e estilo para documentos**. Universidade do Minho, 2000. Disponível em: <http://www3.di.uminho.pt/~jcr/XML/publicacoes/teses/phd-jcr/tese-doutoramento.pdf>. Acesso em: 02 out. 2012.

RENOUF, A. (ed.). 1998. **Explorations in Corpus Linguistics**. Amsterdam, Rodopi.

SINCLAIR, J. 2005. Corpus and Text - Basic Principles. In: M.WYNNE (ed.), **Developing Linguistic Corpora: a Guide to Good Practice**. Oxford, Oxbow Books, p. 1-16. Disponível em: http://ahds.ac.uk/linguistic-corpora/. Acesso em: 30/10/2006.

TEI Consortium, eds. **TEI P5: Guidelines for Electronic Text Encoding and Interchange**. [2.1.0.]. [Last updated on 17th June 2012]. TEI Consortium. Disponível em: <http://www.tei-c.org/Guidelines/P5/>. Acesso em: 02 out. 2012.

# ANEXO A – TEXTO ORIGINAL

VENTO

VENTO AGENTE NAÔ PODE VER
MAIS PODEMOS SENTIR ELE NOS
ALIVIANDO OO CALOR IMENÇO
NO NOSSO IMENÇO PAIS,

O VENTO NOS TRÁZ VENTO POIS
COM ELE PODEMOS RESPIRA NACA
NESSE MUNDO VIVE SEM ESSE MARAVILHOSO
VINTO QUE E DO NOSSO CEARÁ.

O VENTO NO CEARÁ NOS TRAZ
MUITA ALEGRIA E VONTADE DE RESPIRA
POIS COM ESSE MARAVILHOSO VENTO
NÓS PODEMO SE ORGULAR,

OVENTO QUANDO VEM TRAZ
UMA COISA MUITA MARAVILHOSA
COMO A BRISA DAS ARVORES QUE
VEM NÓS ENCONTRAR,

SENHOR SABE O QUE FAZ
FEZ O VENTO QUE TRAZ A PAZ
ALEGRIA QUE NÓS FAZ TORNA
UMAS PESSOAS MARAVILHOSA.

NOME OMITIDO

# ANEXO B – TEXTO ORIGINAL EM FORMATO .TXT

VENTO

VENTO AGENTE NÃO PODE VER
MAIS PODEMOS SENTIR ELE NOS
ALIVIANDO DO CALOR IMENÇO
NO NOSSO IMENÇO PAIS,

O VENTO NOS TRÁZ VENTO POIS
COM ELE PODEMOS RESPIRA NADA
NESSE MUNDO VIVE SEM ESSE MARAVILHOSO
VINTO QUE E DO NOSSO CEARÁ

O VENTO NO CEARÁ NOS TRAZ
MUITA ALEGRIA E VONTADE DE RESPIRA
POIS COM ESSE MARAVILHOSO VENTO
NÓS PODEMOS SE ORGULHAR,

O VENTO QUANDO VEM TRAZ
UMA COISA MUITA MATAVILHOSA
COMO A BRISA DAS ARVORES QUE
VEM NÓS ENCONTRAR,

SENHOR SABE O QUE FAZ
FEZ O VENTO QUE TRAZ A PAZ
ALEGRIA QUE NÓS FAZ TORNA
UMAS PESSOAS MARAVILHOSA.

(Nome omitido)

## ANEXO C – *INPUT* DO TEXTO ORIGINAL PARA A FASE 1 DO ASTRO

VENTO AGENTE NÃO PODE VER MAIS PODEMOS SENTIR ELE NOS ALIVIANDO DO CALOR IMENÇO NO NOSSO IMENÇO PAIS.

O VENTO NOS TRÁZ VENTO POIS COM ELE PODEMOS RESPIRA NADA NESSE MUNDO VIVE SEM ESSE MARAVILHOSO VINTO QUE E DO NOSSO CEARÁ.

O VENTO NO CEARÁ NOS TRAZ MUITA ALEGRIA E VONTADE DE RESPIRA POIS COM ESSE MARAVILHOSO VENTO NÓS PODEMOS SE ORGULHAR.

O VENTO QUANDO VEM TRAZ UMA COISA MUITA MARAVILHOSA COMO A BRISA DAS ARVORES QUE VEM NÓS ENCONTRAR.

SENHOR SABE O QUE FAZ FEZ O VENTO QUE TRAZ A PAZ ALEGRIA QUE NÓS FAZ TORNA UMAS PESSOAS MARAVILHOSA.

## ANEXO D – RESULTADO PRIMEIRA FASE DO ASTRO (XML 1) + ANOTAÇÕES MANUAIS

```
<?xml version="1.0" ?>
<xml encoding="UTF-8" version="1.0">
  <TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
      <fileDesc>
        <titleStmt>
          <title> VENTO </title>
          <genre/> Poetry
          <person age="Between 15 and 21" sex="Female"> LQS
            <birth when="">Unknown
              <date/>
              <name type=""/>
            </birth>
            <residence>Barroquinha (CE) </residence>
            <education> High School Student (1º Ensino Médio - E.E. M. Jaime L. da Silva)
</education>
            <occupation>Unknown</occupation>
          </person>
        </titleStmt>
        <tokens n="110"/>
      </fileDesc>
      <respStmt xml:id="km">
        <resp>Corpus Astrolabio organizer</resp>
        <name>Katiuscia de Moraes Andrade</name>
      </respStmt>
      <respStmt xml:id="at">
        <resp>Automatic tagger</resp>
        <name>Astro</name>
      </respStmt>
      <publicationStmt>
        <availability>Creative Commons Attribution-NonCommercial 3.0
Unported</availability>
      </publicationStmt>
      <editorialDecl>
        <correction/>
        <p>Spelling corrector was made by using the interface Astro for Enchant</p>
        <p>Another erros that couldn't be identified by Enchant (as its own errors) were
submitted to a human revision</p>
        <normalization/>
      </editorialDecl>
    </teiHeader>
    <facsimile>
          <surfaceGrp n="leaf1">
                <surface>
                      ulx=" 1275"
```

```
                        uly=" 1755"
                        <graphic url=" "/>
                        <zone  ulx=" "uly=" "/>
                        <!-- contains a black tag to cover the autor's signature -->
                    </surface>
</surfaceGrp>
</facsimile>
   <text>
     <body id="body">
         <pb facs=" bar_LQS_f_x_01.jpg"/>
       <p>
        <s>
         <w xml:id="1">Vento</w>
          <choice>
          <corr resp="#km">
            <w xml:id="2">a gente</w>
          </corr>
          <sic>
            <w xml:id="2sic">agente</w>
          </sic>
         </choice>
         <w xml:id="3">não</w>
         <w xml:id="4">pode</w>
         <w xml:id="5">ver</w>
          <choice>
         <add resp="#km">
                  <w xml:id="5add">,</w>
         </add>
         </choice>
         <choice>
         <corr resp="#km">
                  <w xml:id="6" resp="#km">mas</w>
         </corr>
         <sic>
                  <w xml:id="6sic">mais</w>
         </sic>
         </choice>
         <w xml:id="7">podemos</w>
         <w xml:id="8">sentir</w>
         <w xml:id="9">ele</w>
         <w xml:id="10">nos</w>
         <w xml:id="11">aliviando</w>
         <w xml:id="12">de</w>
         <w xml:id="13">o</w>
         <w xml:id="14">calor</w>
         <choice>
          <corr resp="#at">
            <w xml:id="15">imenso</w>
          </corr>
          <sic>
```

```
      <w xml:id="15sic">imenço</w>
     </sic>
    </choice>
    <w xml:id="16">em</w>
    <w xml:id="17">o</w>
    <w xml:id="18">nosso</w>
    <choice>
     <corr resp="#at">
      <w xml:id="19">imenso</w>
     </corr>
     <sic>
      <w xml:id="19sic">imenso</w>
     </sic>
    </choice>
     <choice>
     <corr resp="#km">
       <w xml:id="20">país</w>
     </corr>
     <sic>
      <w xml:id="20sic">pais</w>
     </sic>
    </choice>
    <w xml:id="21">.</w>
  </s>
</p>
<p>
  <s>
    <w xml:id="22">O</w>
    <w xml:id="23">vento</w>
    <w xml:id="24">nos</w>
    <choice>
     <corr  resp="#km">
      <w xml:id="25" at="trás">traz</w>
     </corr>
     <sic>
      <w xml:id="25sic">tráz</w>
     </sic>
    </choice>
    <w xml:id="26">vento</w>
    <choice>
    <add resp="#km">
              <w xml:id="26add">,</w>
    </add>
    </choice>
    <w xml:id="27">pois</w>
    <w xml:id="28">com</w>
    <w xml:id="29">ele</w>
    <w xml:id="30">podemos</w>
    <choice>
     <corr resp="#km">
```

```
            <w xml:id="31">respirar</w>
          </corr>
          <sic>
          <w xml:id="31sic">respira</w>
          </sic>
        </choice>
        <choice>
        <add resp="#km">
                <w xml:id="31add">.</w>
        </add>
        </choice>
        </s>
<s>
        <w xml:id="32">nada</w>
        <w xml:id="33">nesse</w>
        <w xml:id="34">mundo</w>
        <w xml:id="35">vive</w>
        <w xml:id="36">sem</w>
        <w xml:id="37">esse</w>
        <w xml:id="38">maravilhoso</w>
        <choice resp="km">
          <corr>
            <w xml:id="39" at="vim to">vento</w>
          </corr>
          <sic>
            <w xml:id="39sic">vinto</w>
          </sic>
        </choice>
        <w xml:id="40">que</w>
        <choice>
                <corr resp="#km">
                <w xml:id="41">é</w>
        </corr>
        <sic>
                <w xml:id="41sic">e</w>
        </sic>
        </choice>
        <w xml:id="42">de</w>
        <w xml:id="43">o</w>
        <w xml:id="44">nosso</w>
        <w xml:id="45">Ceará</w>
        <w xml:id="46">.</w>
      </s>
    </p>
    <p>
      <s>
        <w xml:id="47">O</w>
        <w xml:id="48">vento</w>
        <w xml:id="49">em</w>
        <w xml:id="50">o</w>
```

```
    <w xml:id="51">Ceaá</w>
    <w xml:id="52">nos</w>
    <w xml:id="53">traz</w>
    <w xml:id="54">muita</w>
    <w xml:id="55">alegria</w>
    <w xml:id="56">e</w>
    <w xml:id="57">vontade</w>
    <w xml:id="58">de</w>
    <choice>
     <corr resp="#km">
       <w xml:id="59">respirar</w>
      </corr>
      <sic>
      <w xml:id="59sic">respira</w>
      </sic>
      </choice>
      <choice>
    <add resp="#km">
              <w xml:id="59add">,</w>
    </add>
    </choice>
    <w xml:id="60">pois</w>
    <w xml:id="61">com</w>
    <w xml:id="62">esse</w>
    <w xml:id="63">maravilhoso</w>
    <w xml:id="64">vento</w>
    <w xml:id="65">nós</w>
    <w xml:id="66">podemos</w>
    <choice>
      <reg resp="#km">
        <w xml:id="67">nos</w>
      </reg>
      <orig>
        <w xml:id="67orig">se</w>
      </orig>
    </choice>
    <w xml:id="68">orgulhar</w>
    <w xml:id="69">.</w>
  </s>
</p>
<p>
  <s>
    <w xml:id="70">O</w>
    <w xml:id="71">vento</w>
    <w xml:id="72">quando</w>
    <w xml:id="73">vem</w>
    <w xml:id="74">traz</w>
    <w xml:id="75">uma</w>
    <w xml:id="76">coisa</w>
    <w xml:id="77">muita</w>
```

```
    <w xml:id="78">maravilhosa</w>
    <w xml:id="79">como</w>
    <w xml:id="80">a</w>
    <w xml:id="81">brisa</w>
    <w xml:id="82">de</w>
    <w xml:id="83">as</w>
    <choice>
     <corr resp="#km">
       <w xml:id="84">árvores</w>
     </corr>
     <sic>
     <w xml:id="84sic">arvores</w>
     </sic>
     </choice>
    <w xml:id="85">que</w>
    <w xml:id="86">vem</w>
    <choice>
     <corr resp="#km">
       <w xml:id="87">nos</w>
     </corr>
     <sic>
     <w xml:id="87sic">nós</w>
     </sic>
     </choice>
    <w xml:id="88">encontrar</w>
    <w xml:id="89">.</w>
  </s>
</p>
<p>
  <s>
    <w xml:id="90">Senhor</w>
    <w xml:id="91">sabe</w>
    <w xml:id="92">o</w>
    <w xml:id="93">que</w>
    <w xml:id="94">faz</w>
     <add resp="#km">
            <w xml:id="94add">,</w>
    </add>
    <w xml:id="95">fez</w>
    <w xml:id="96">o</w>
    <w xml:id="97">vento</w>
    <w xml:id="98">que</w>
    <w xml:id="99">traz</w>
    <w xml:id="100">a</w>
    <w xml:id="101">paz</w>
     <add resp="#km">
            <w xml:id="101add">,</w>
    </add>
    <w xml:id="102">alegria</w>
    <w xml:id="103">que</w>
```

```
          <choice>
           <corr resp="#km">
             <w xml:id="104">nos</w>
           </corr>
           <sic>
           <w xml:id="104sic">nós</w>
           </sic>
           </choice>
           <choice>
            <del resp="#km">
                  <w xml:id="105">faz</w>
          </del>
          <note resp="#km">O termo foi deletado com o intuito de tornar o texto mais claro.
          </note>
          </choice>
          <w xml:id="106">torna</w>
          <w xml:id="107">umas</w>
          <w xml:id="108">pessoas</w>
          <choice>
                  <corr resp="#km">
                  <w xml:id="109">maravilhosas</w>
                  </corr>
                  <sic>
                  <w xml:id="109sic">maravilhosa</w>
                  </sic>
          </choice>
          <w xml:id="110">.</w>
         </s>
        </p>
        <p>
         <s/>
        </p>
      </body>
    </text>
  </TEI>
</xml>
```

# ANEXO E – FASE FINAL DA ANOTAÇÃO POR MEIO DO ASTRO 1.0 (ANOTAÇÃO XML + ANOTAÇÃO MORFOSSINTÁTICA) - SEM REVISÃO HUMANA


```
<?xml version="1.0" ?>
<xml encoding="UTF-8" version="1.0">
  <TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>


      <fileDesc>


        <titleStmt>


          <title> VENTO </title>


          <genre/>
           Poetry

          <person age="Between 15 and 21" sex="Female">
            LQS

           <birth when="">
            Unknown

            <date/>


            <name type=""/>


          </birth>


          <residence>Barroquinha (CE) </residence>
```

<education> High School Student (1º Ensino Médio - E.E. M. Jaime L. da Silva) </education>

<occupation>Unknown</occupation>

</person>

</titleStmt>

<tokens n="110"/>

</fileDesc>

<respStmt xml:id="km">

<resp>Corpus Astrolabio organizer</resp>

<name>Katiuscia de Moraes Andrade</name>

</respStmt>

<respStmt xml:id="at">

<resp>Automatic tagger</resp>

<name>Astro</name>

</respStmt>

```
    <publicationStmt>

      <availability>Creative Commons Attribution-NonCommercial 3.0 Unported</availability>

    </publicationStmt>

    <editorialDecl>

      <correction/>

      <p>Spelling corrector was made by using the interface Astro for Enchant</p>

      <p>Another erros that couldn't be identified by Enchant (as its own errors) were submitted to a human revision</p>

      <normalization/>

    </editorialDecl>

  </teiHeader>
  <text>
    <body id="body">
      <p>
        <s>
          <w type="NPROP" xml:id="1">Vento</w>
          <choice>
            <corr>
              <w type="ART" xml:id="2">a</w>
            </corr>
```

```
  <sic>
    <w xml:id="2sic">agente</w>
  </sic>
</choice>
<w type="N" xml:id="3">gente</w>
<w type="ADV" xml:id="4">não</w>
<w type="VAUX" xml:id="5">pode</w>
<choice>
  <corr>
    <w type="V" xml:id="6">ver</w>
  </corr>
  <sic>
    <w xml:id="6sic">mais</w>
  </sic>
</choice>
<w type="," xml:id="7">,</w>
<w type="KC" xml:id="8">mas</w>
<w type="VAUX" xml:id="9">podemos</w>
<w type="V" xml:id="10">sentir</w>
<w type="PROPESS" xml:id="11">ele</w>
<w type="PROPESS" xml:id="12">nos</w>
<w type="V" xml:id="13">aliviando</w>
<w type="PREP|+" xml:id="14">de</w>
<choice>
  <corr>
    <w type="ART" xml:id="15">o</w>
  </corr>
  <sic>
    <w xml:id="15sic">imenço</w>
  </sic>
</choice>
<w type="N" xml:id="16">calor</w>
<w type="ADJ" xml:id="17">imenso</w>
<w type="PREP|+" xml:id="18">em</w>
<choice>
  <corr>
    <w type="ART" xml:id="19">o</w>
  </corr>
  <sic>
    <w xml:id="19sic">imenso</w>
```

```
    </sic>
   </choice>
   <choice>
    <corr>
     <w type="PROADJ" xml:id="20">nosso</w>
    </corr>
    <sic>
     <w xml:id="20sic">pais</w>
    </sic>
   </choice>
   <w type="ADJ" xml:id="21">imenso</w>
  </s>
</p>
<p>
  <s>
   <w type="N" xml:id="22">país</w>
   <w type="." xml:id="23">.</w>
   <w type="ART" xml:id="24">O</w>
   <choice>
    <corr>
     <w type="N" xml:id="25">vento</w>
    </corr>
    <sic>
     <w xml:id="25sic">tráz</w>
    </sic>
   </choice>
   <w type="PROPESS" xml:id="26">nos</w>
   <w type="V" xml:id="27">traz</w>
   <w type="N" xml:id="28">vento</w>
   <w type="," xml:id="29">,</w>
   <w type="KC" xml:id="30">pois</w>
   <choice>
    <corr>
     <w type="PREP" xml:id="31">com</w>
    </corr>
    <sic>
     <w xml:id="31sic">respira</w>
    </sic>
   </choice>
  </s>
```

```
  <s>
    <w type="PROPESS" xml:id="32">ele</w>
    <w type="VAUX" xml:id="33">podemos</w>
    <w type="V" xml:id="34">respirar</w>
    <w type="." xml:id="35">.</w>
    <w type="PROSUB" xml:id="36">Nada</w>
    <w type="PREP|+" xml:id="37">em</w>
    <w type="PROADJ" xml:id="38">esse</w>
    <choice>
      <corr>
        <w type="N" xml:id="39">mundo</w>
      </corr>
      <sic>
        <w xml:id="39sic">vinto</w>
      </sic>
    </choice>
    <w type="V" xml:id="40">vive</w>
    <choice>
      <corr>
        <w type="PREP" xml:id="41">sem</w>
      </corr>
      <sic>
        <w xml:id="41sic">e</w>
      </sic>
    </choice>
    <w type="PROADJ" xml:id="42">esse</w>
    <w type="ADJ" xml:id="43">maravilhoso</w>
    <w type="N" xml:id="44">vento</w>
    <w type="PRO-KS-REL" xml:id="45">que</w>
    <w type="V" xml:id="46">é</w>
  </s>
</p>
<p>
  <s>
    <w type="PREP|+" xml:id="47">de</w>
    <w type="ART" xml:id="48">o</w>
    <w type="PROADJ" xml:id="49">nosso</w>
    <w type="NPROP" xml:id="50">Ceará</w>
    <w type="." xml:id="51">.</w>
    <w type="ART" xml:id="52">O</w>
```

```
    <w type="N" xml:id="53">vento</w>
    <w type="PREP|+" xml:id="54">em</w>
    <w type="ART" xml:id="55">o</w>
    <w type="NPROP" xml:id="56">Ceaá</w>
    <w type="PROPESS" xml:id="57">nos</w>
    <w type="V" xml:id="58">traz</w>
    <choice>
     <corr>
      <w type="PROADJ" xml:id="59">muita</w>
     </corr>
     <sic>
      <w xml:id="59sic">respira</w>
     </sic>
    </choice>
    <w type="N" xml:id="60">alegria</w>
    <w type="KC" xml:id="61">e</w>
    <w type="N" xml:id="62">vontade</w>
    <w type="PREP" xml:id="63">de</w>
    <w type="V" xml:id="64">respirar</w>
    <w type="," xml:id="65">,</w>
    <w type="KC" xml:id="66">pois</w>
    <w type="PROADJ" xml:id="68">esse</w>
    <w type="ADJ" xml:id="69">maravilhoso</w>
   </s>
  </p>
  <p>
   <s>
    <w type="N" xml:id="70">vento</w>
    <w type="PROPESS" xml:id="71">nós</w>
    <w type="VAUX" xml:id="72">podemos</w>
    <w type="PREP|+" xml:id="73">em</w>
    <w type="ART" xml:id="74">os</w>
    <w type="PROPESS" xml:id="75">se</w>
    <w type="V" xml:id="76">orgulhar</w>
    <w type="." xml:id="77">.</w>
    <w type="ART" xml:id="78">O</w>
    <w type="N" xml:id="79">vento</w>
    <w type="KS" xml:id="80">quando</w>
    <w type="V" xml:id="81">vem</w>
    <w type="V" xml:id="82">traz</w>
```

```
    <w type="ART" xml:id="83">uma</w>
    <choice>
     <corr>
      <w type="N" xml:id="84">coisa</w>
     </corr>
     <sic>
      <w xml:id="84sic">arvores</w>
     </sic>
    </choice>
    <w type="PROADJ" xml:id="85">muita</w>
    <w type="ADJ" xml:id="86">maravilhosa</w>
    <choice>
     <corr>
      <w type="PREP" xml:id="87">como</w>
     </corr>
     <sic>
      <w xml:id="87sic">nós</w>
     </sic>
    </choice>
    <w type="ART" xml:id="88">a</w>
    <w type="N" xml:id="89">brisa</w>
   </s>
  </p>
  <p>
   <s>
    <w type="PREP|+" xml:id="90">de</w>
    <w type="ART" xml:id="91">as</w>
    <w type="N" xml:id="92">árvores</w>
    <w type="PRO-KS-REL" xml:id="93">que</w>
    <w type="V" xml:id="94">vem</w>
    <w type="PREP|+" xml:id="95">em</w>
    <w type="ART" xml:id="96">os</w>
    <w type="V" xml:id="97">encontrar</w>
    <w type="." xml:id="98">.</w>
    <w type="N" xml:id="99">Senhor</w>
    <w type="V" xml:id="100">sabe</w>
    <w type="PRO-KS" xml:id="101">o</w>
    <w type="PRO-KS" xml:id="102">que</w>
    <w type="V" xml:id="103">faz</w>
    <choice>
```

```
            <corr>
              <w type="," xml:id="104">,</w>
            </corr>
            <sic>
              <w xml:id="104sic">nós</w>
            </sic>
          </choice>
          <w type="ART" xml:id="106">o</w>
          <w type="N" xml:id="107">vento</w>
          <w type="PRO-KS-REL" xml:id="108">que</w>
          <choice>
            <corr>
              <w type="V" xml:id="109">traz</w>
            </corr>
            <sic>
              <w xml:id="109sic">maravilhosa</w>
            </sic>
          </choice>
          <w type="ART" xml:id="110">a</w>
        </s>
      </p>
      <p>
        <s/>
      </p>
    </body>
  </text>
 </TEI>
</xml>
```

# ANEXO F – FUNÇÃO *ATINITC.PY* – ASTRO 1.0 (FASE 1)

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Astro Corpus Annotation Tool
#
# 2013 - Author-  Mardonio Franca <mardfranca@gmail.com>
#
# URL: <http://http://www.rotadasespeciarias.art.br/astrolabio/astrolabio/fonte.zip>
# This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License
#
# $Id: ATinitC.py $

import nltk
import os
import sys
import codecs
import xml.dom.minidom
from   xml.dom.minidom import *

caminho = '/home/katiusha/'
sys.path.append(caminho)

import   Aelius
from     Aelius.Toqueniza import TOK_PORT_LX
from     Aelius.AnotaCorpus import toqueniza_contracoes

import   Astro
from     Astro import Corretor


class ATinitC:

      arqNome = None


      def __init__(self, arqNome):
         self.arqNome = arqNome
```

```
def executa(self):


    doc = Document()
    xml = doc.createElement("xml")
    xml.setAttribute("version", "1.0")
    xml.setAttribute("encoding", "UTF-8")
    doc.appendChild(xml)

    TEI         = doc.createElement("TEI")
    TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
    xml.appendChild(TEI)

    teiHeader = doc.createElement("teiHeader")
    TEI.appendChild(teiHeader)

    fileDesc     = doc.createElement("fileDesc")
    teiHeader.appendChild(fileDesc)

    titleStmt   = doc.createElement("titleStmt")
    fileDesc.appendChild(titleStmt)

    title         = doc.createElement("title")
    titleStmt.appendChild(title)
    genre      = doc.createElement("genre")
    titleStmt.appendChild(genre)

    person    = doc.createElement("person")
    person.setAttribute("sex", "")
    person.setAttribute("age", "")
    titleStmt.appendChild(person)

    birth       = doc.createElement("birth")
    birth.setAttribute("when", "")
    person.appendChild(birth)

    date      = doc.createElement("date")
    birth.appendChild(date)
    name    = doc.createElement("name")
```

```
name.setAttribute("type", "")
birth.appendChild(name)

residence   = doc.createElement("residence")
person.appendChild(residence)
education   = doc.createElement("education")
person.appendChild(education)
occupation = doc.createElement("occupation")
person.appendChild(occupation)

respStmt    = doc.createElement("respStmt")
respStmt.setAttribute("xml:id", "km")
teiHeader.appendChild(respStmt)
resp          = doc.createElement("resp")
ptext         = doc.createTextNode("Corpus Astrolabio organizer")
resp.appendChild(ptext)
respStmt.appendChild(resp)

name  = doc.createElement("name")
ptext   = doc.createTextNode("Katiuscia de Moraes Andrade")
name.appendChild(ptext)
respStmt.appendChild(name)


respStmt = doc.createElement("respStmt")
respStmt.setAttribute("xml:id", "ap")
teiHeader.appendChild(respStmt)

resp  = doc.createElement("resp")
ptext = doc.createTextNode("Automatic tagger")
resp.appendChild(ptext)
respStmt.appendChild(resp)
name = doc.createElement("name")
ptext  = doc.createTextNode("Astro.py")
name.appendChild(ptext)
respStmt.appendChild(name)

publicationStmt = doc.createElement("publicationStmt")
teiHeader.appendChild(publicationStmt)
availability        = doc.createElement("availability")
```

```
        publicationStmt.appendChild(availability)

        editorialDecl     = doc.createElement("editorialDecl")
        teiHeader.appendChild(editorialDecl)

        correction       = doc.createElement("correction")
        editorialDecl.appendChild(correction)

        p       = doc.createElement("p")
        ptext = doc.createTextNode("Errors in th original .txt file were corrected by using the interface Astro.py for Enchant")
        p.appendChild(ptext)
        editorialDecl.appendChild(p)

        p       = doc.createElement("p")
        ptext = doc.createTextNode("Another erros that couldn't be identified by Enchant (as its own errors) were submitted to a human revision")
        p.appendChild(ptext)
        editorialDecl.appendChild(p)

        normalization  = doc.createElement("normalization")
        editorialDecl.appendChild(normalization)

        facsimile        = doc.createElement("facsimile")
        TEI.appendChild(teiHeader)
        surfaceGrp= doc.createElement("surfaceGrp")
        surfaceGrp.setAttribute("n", " leaf1")

        facsimile.appendChild(surfaceGrp)
        surface= doc.createElement("surface")
        surface.setAttribute("ulx", " ")
        surface.setAttribute("uly", " ")
        surfaceGrp.appendChild(surface)

        graphic= doc.createElement("graphic")
        graphic.setAttribute("ulx", " ")
        graphic.setAttribute("uly", " ")
        surface.appendChild(graphic)

        zone= doc.createElement("zone")
```

```
zone.setAttribute("ulx", " ")
zone.setAttribute("uly", " ")
surface.appendChild(zone)


texto = doc.createElement("text")
TEI.appendChild(texto)


body = doc.createElement("body")
body.setAttribute("id", "body")
texto.appendChild(body)



arq      = open(self.arqNome )
linhas   = arq.readlines()
print "em processamento"
n  = 1
vetorLinha = []
strLinha = ""

for linha in linhas:
   valorLinha = linha
   strLinha    = strLinha + linha
   if (len(valorLinha)==1) and (valorLinha=="\n") :
      vetorLinha.append(strLinha)
      vetorLinha.append("")
      strLinha=""
vetorLinha.append(strLinha)

parag       = nltk.data.load('tokenizers/punkt/portuguese.pickle')
nTokens = 0




for vetor in vetorLinha:
      #print vetor
      if len(vetor)!=0 :
```

```
                    nTokens = nTokens + len(vetor)
                    p = doc.createElement("p")
                    body.appendChild(p)
                    sentencas = parag.tokenize(vetor.decode("utf-8"))
                    for sentenca in sentencas:
                       s   = doc.createElement("s")
                       p.appendChild(s)
                       vetorTok = TOK_PORT_LX.tokenize(sentenca)
                       vetorTok = toqueniza_contracoes([vetorTok])

                       for palavra in  vetorTok[0] :
                          palavraSic = palavra
                          ctr = Corretor.SpellingReplacer()
                          palavraCor = ctr.replace(palavraSic)
                          if (palavraSic == palavraCor) or palavraSic == ',' or palavraSic == '!' or palavraSic ==';' or
palavraSic=='-' or palavraSic=='...' or palavraSic=="":
                             w    = doc.createElement("w")
                             w.setAttribute("xml:id", str(n))
                             n= n+ 1
                             s.appendChild(w)
                             ptext = doc.createTextNode(palavra)
                             w.appendChild(ptext)
                          else:
                             choice    = doc.createElement("choice")
                             s.appendChild(choice)
                             corr    = doc.createElement("corr")
                             choice.appendChild(corr)
                             w    = doc.createElement("w")
                             w.setAttribute("xml:id", str(n))
                             corr.appendChild(w)
                             ptext = doc.createTextNode(palavraCor)
                             w.appendChild(ptext)
                             sic    = doc.createElement("sic")
                             choice.appendChild(sic)
                             w    = doc.createElement("w")
                             w.setAttribute("xml:id", str(n) + "sic")
                             sic.appendChild(w)
                             ptext = doc.createTextNode(palavraSic)
                             w.appendChild(ptext)
                             n= n+ 1
```

```
numberTokens= doc.createElement("tokens")
numberTokens.setAttribute("n",  str(nTokens))
fileDesc.appendChild(numberTokens)

nomeArqG = self.arqNome  + ".01.xml"
f  = codecs.open(nomeArqG, "w", "utf-8")
f.write(doc.toprettyxml(indent="  "))
f.close()

print "\n"
print "arquivo processado com sucesso " + "\n"
print "nome do arquivo: "  + nomeArqG   + "\n"
print "\n"
print "\n"
print "\n"
print "------------------------------------------------------------\n"
```

# ANEXO G – FUNÇÃO *ATINITC.PY* – ASTRO 1.1

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Astro Corpus Annotation Tool
#
# 2013 - Author-  Mardonio Franca <mardfranca@gmail.com>
#
# URL: <http://http://www.rotadasespeciarias.art.br/astrolabio/astrolabio/fonte.zip>
# This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License
#
# $Id: ATinitC.py $

import nltk
import os
import sys
import codecs
import xml.dom.minidom
from   xml.dom.minidom import *

caminho = '/home/franca/'
sys.path.append(caminho)

import   Aelius
from     Aelius.Toqueniza import TOK_PORT_LX
from     Aelius.AnotaCorpus import toqueniza_contracoes

import   Astro
from     Astro import Corretor


class ATinitC:

      arqNome = None


      def __init__(self, arqNome):
         self.arqNome = arqNome
```

```
def executa(self):


    doc = Document()
    xml = doc.createElement("xml")
    xml.setAttribute("version", "1.0")
    xml.setAttribute("encoding", "UTF-8")
    doc.appendChild(xml)

    TEI         = doc.createElement("TEI")
    TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
    xml.appendChild(TEI)

    teiHeader = doc.createElement("teiHeader")
    TEI.appendChild(teiHeader)

    fileDesc     = doc.createElement("fileDesc")
    teiHeader.appendChild(fileDesc)

    titleStmt   = doc.createElement("titleStmt")
    fileDesc.appendChild(titleStmt)

    title         = doc.createElement("title")
    titleStmt.appendChild(title)
    genre      = doc.createElement("genre")
    titleStmt.appendChild(genre)

    person    = doc.createElement("person")
    person.setAttribute("sex", "")
    person.setAttribute("age", "")
    titleStmt.appendChild(person)

    birth       = doc.createElement("birth")
    birth.setAttribute("when", "")
    person.appendChild(birth)

    date      = doc.createElement("date")
    birth.appendChild(date)
    name    = doc.createElement("name")
```

```
name.setAttribute("type", "")
birth.appendChild(name)

residence   = doc.createElement("residence")
person.appendChild(residence)
education   = doc.createElement("education")
person.appendChild(education)
occupation = doc.createElement("occupation")
person.appendChild(occupation)

respStmt    = doc.createElement("respStmt")
respStmt.setAttribute("xml:id", "km")
teiHeader.appendChild(respStmt)
resp          = doc.createElement("resp")
ptext         = doc.createTextNode("Corpus Astrolabio organizer")
resp.appendChild(ptext)
respStmt.appendChild(resp)

name  = doc.createElement("name")
ptext   = doc.createTextNode("Katiuscia de Moraes Andrade")
name.appendChild(ptext)
respStmt.appendChild(name)


respStmt = doc.createElement("respStmt")
respStmt.setAttribute("xml:id", "ap")
teiHeader.appendChild(respStmt)

resp  = doc.createElement("resp")
ptext = doc.createTextNode("Automatic tagger")
resp.appendChild(ptext)
respStmt.appendChild(resp)
name = doc.createElement("name")
ptext  = doc.createTextNode("Astro.py")
name.appendChild(ptext)
respStmt.appendChild(name)

publicationStmt = doc.createElement("publicationStmt")
teiHeader.appendChild(publicationStmt)
availability        = doc.createElement("availability")
```

```
        publicationStmt.appendChild(availability)

        editorialDecl     = doc.createElement("editorialDecl")
        teiHeader.appendChild(editorialDecl)

        correction       = doc.createElement("correction")
        editorialDecl.appendChild(correction)

        p       = doc.createElement("p")
        ptext = doc.createTextNode("Errors in th original .txt file were corrected by using the interface Astro.py for Enchant")
        p.appendChild(ptext)
        editorialDecl.appendChild(p)

        p       = doc.createElement("p")
        ptext = doc.createTextNode("Another erros that couldn't be identified by Enchant (as its own errors) were submitted to a human revision")
        p.appendChild(ptext)
        editorialDecl.appendChild(p)

        normalization  = doc.createElement("normalization")
        editorialDecl.appendChild(normalization)

        facsimile        = doc.createElement("facsimile")
        TEI.appendChild(teiHeader)
        surfaceGrp= doc.createElement("surfaceGrp")
        surfaceGrp.setAttribute("n", " leaf1")

        facsimile.appendChild(surfaceGrp)
        surface= doc.createElement("surface")
        surface.setAttribute("ulx", " ")
        surface.setAttribute("uly", " ")
        surfaceGrp.appendChild(surface)

        graphic= doc.createElement("graphic")
        graphic.setAttribute("ulx", " ")
        graphic.setAttribute("uly", " ")
        surface.appendChild(graphic)

        zone= doc.createElement("zone")
```

```
zone.setAttribute("ulx", " ")
zone.setAttribute("uly", " ")
surface.appendChild(zone)


texto = doc.createElement("text")
TEI.appendChild(texto)


body = doc.createElement("body")
body.setAttribute("id", "body")
texto.appendChild(body)



arq      = open(self.arqNome )
linhas   = arq.readlines()
print "em processamento"
n  = 1
vetorLinha = []
strLinha = ""

for linha in linhas:
   valorLinha = linha
   strLinha    = strLinha + linha
   if (len(valorLinha)==1) and (valorLinha=="\n") :
      vetorLinha.append(strLinha)
      vetorLinha.append("")
      strLinha=""
vetorLinha.append(strLinha)

parag       = nltk.data.load('tokenizers/punkt/portuguese.pickle')

np=0;
ns=0;
nch=0;
for vetor in vetorLinha:
      if len(vetor)!=0 :
            np = np +1
            p = doc.createElement("p")
```

```
                p.setAttribute("xml:id", "p" + str(np))
                body.appendChild(p)
                sentencas = parag.tokenize(vetor.decode("utf-8"))
                for sentenca in sentencas:
                   ns = ns + 1
                   s   = doc.createElement("s")
                   s.setAttribute("xml:id", "s" + str(ns))
                   p.appendChild(s)
                   vetorTok = TOK_PORT_LX.tokenize(sentenca)
                   vetorTok = toqueniza_contracoes([vetorTok])

                   for palavra in  vetorTok[0] :
                      palavraSic = palavra
                      ctr = Corretor.SpellingReplacer()
                      palavraCor = ctr.replace(palavraSic)
                      if (palavraSic == palavraCor) or palavraSic == ',' or palavraSic == '!' or palavraSic ==';' or
palavraSic=='-' or palavraSic=='...' or palavraSic=="":
                         w    = doc.createElement("w")
                         w.setAttribute("xml:id", str(n))
                         n= n+ 1
                         s.appendChild(w)
                         ptext = doc.createTextNode(palavra)
                         w.appendChild(ptext)
                      else:
                         nch = nch + 1
                         choice    = doc.createElement("choice")
                         choice.setAttribute("xml:id", "ch" + str(nch))
                         s.appendChild(choice)
                         corr    = doc.createElement("corr")
                         choice.appendChild(corr)
                         w    = doc.createElement("w")
                         w.setAttribute("xml:id", str(n))
                         corr.appendChild(w)
                         palavraCor = palavraCor.replace(" ", "")
                         ptext = doc.createTextNode(palavraCor)

                         w.appendChild(ptext)
                         sic    = doc.createElement("sic")
                         choice.appendChild(sic)
                         w    = doc.createElement("w")
```

```
                    w.setAttribute("xml:id", str(n) + "sic")
                    w.setAttribute("resp", "at")
                    sic.appendChild(w)
                    ptext = doc.createTextNode(palavraSic)
                    w.appendChild(ptext)
                    n= n+ 1

nomeArqG = self.arqNome  + ".01.xml"
f  = codecs.open(nomeArqG, "w", "utf-8")
f.write(doc.toprettyxml(indent="  "))
f.close()

print "\n"
print "arquivo processado com sucesso " + "\n"
print "nome do arquivo: "  + nomeArqG   + "\n"
print "\n"
print "\n"
print "\n"
print "------------------------------------------------------------\n"
```

# ANEXO H – FUNÇÃO *ATANOTAC.PY* ASTRO 1.0

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Astro Corpus Annotation Tool
#
# 2013 - Author-  Mardonio Franca <mardfranca@gmail.com>
#
# URL: <http://http://www.rotadasespeciarias.art.br/astrolabio/astrolabio/fonte.zip>
# This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License
#
# $Id: ATanotaC.py $

import nltk
import os
import sys
import codecs
import xml.dom.minidom
from   xml.dom.minidom import *

caminho = '/home/katiusha/'
sys.path.append(caminho)

import  Aelius
from    Aelius.Toqueniza import TOK_PORT_LX
from    Aelius import Extras,Toqueniza, AnotaCorpus
from    Aelius.Extras import carrega

from    AnotaCorpus import *

import  Astro
from    Astro import Corretor



class ATanotaC:

      arqNome = None
```

```
def __init__(self, arqNome):
    self.arqNome = arqNome


def executa(self):

    n  = 1
    nomeArqM =  self.arqNome
    pagina       = parse(nomeArqM)

    print "em processamento"
    doc = Document()
    xml = doc.createElement("xml")
    xml.setAttribute("version", "1.0")
    xml.setAttribute("encoding", "UTF-8")
    doc.appendChild(xml)
    TEI          = doc.createElement("TEI")
    TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
    xml.appendChild(TEI)
    texto = doc.createElement("text")
    TEI.appendChild(texto)
    body = doc.createElement("body")
    body.setAttribute("id", "body")
    texto.appendChild(body)

    nTokens    = 0

    paragrafos  = pagina.getElementsByTagName("p")

    for paragrafo in paragrafos:
        p = doc.createElement("p")
        body.appendChild(p)
        sentencas = paragrafo.getElementsByTagName("s")
        for sentenca in sentencas:
            s   = doc.createElement("s")
            p.appendChild(s)
            frase = ""
            palavras = sentenca.getElementsByTagName("w")
            for palavra in palavras:
```

```
        if palavra.hasAttribute("xml:id"):
            ch = palavra.parentNode.nodeName
            if (ch != 'sic') and (ch!='del') :
                frase = frase + " " + palavra.childNodes[0].data


    tk     =  Toqueniza.TOK_PORT_LX.tokenize(frase)
    mac =   Extras.carrega("AeliusHunPosMacMorpho")
    f      =   AnotaCorpus.anota_sentencas([tk],mac,'hunpos', separacao_contracoes=True)
    for lPalavraAnotada in f[0]:
            w    = doc.createElement("w")
            w.setAttribute("xml:id", str(n))
            n= n+ 1
            w.setAttribute("type", lPalavraAnotada[1])
            s.appendChild(w)
            ptext = doc.createTextNode(lPalavraAnotada[0])
            w.appendChild(ptext)


doc2 = Document()
xml2 = doc2.createElement("xml")
xml2.setAttribute("version", "1.0")
xml2.setAttribute("encoding", "UTF-8")
doc2.appendChild(xml2)

TEI         = doc2.createElement("TEI")
TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
xml2.appendChild(TEI)

teiHeader = doc2.createElement("teiHeader")

teiHeader = pagina.getElementsByTagName("teiHeader")[0] .....

TEI.appendChild(teiHeader)

texto = doc2.createElement("text")
TEI.appendChild(texto)

body = doc2.createElement("body")
body.setAttribute("id", "body")
```

```
texto.appendChild(body)

i = 0
paragrafos     = pagina.getElementsByTagName("p")
for paragrafo in paragrafos:

      p2 = doc2.createElement("p")
      body.appendChild(p2)
      sentencas = paragrafo.getElementsByTagName("s")

      for sentenca in sentencas:

            s2 =  doc2.createElement("s")
            p2.appendChild(s2)
            palavrasPs     = sentenca.getElementsByTagName("w") # palavra anotada

            for palavrasP in palavrasPs:
                  nodePaiName   = palavrasP.parentNode.nodeName
                  nodePaiPai = palavrasP.parentNode.parentNode
                  if  (nodePaiName=='s') :
                     palavrasAs     = doc.getElementsByTagName("w") # palavra anotada
                     for palavrasA in palavrasAs:
                         if palavrasP.getAttribute("xml:id")==  palavrasA.getAttribute("xml:id"):
                               s2.appendChild(palavrasA)
                  elif (nodePaiName=='corr') :
                        choice  =  doc2.createElement("choice")
                        corr     =   doc2.createElement("corr")
                        choice.appendChild(corr)
                        for palavrasA in palavrasAs:
                              if palavrasP.getAttribute("xml:id")==  palavrasA.getAttribute("xml:id"):
                                    corr.appendChild(palavrasA)
                        pws = nodePaiPai.getElementsByTagName("w")
                        for pw in pws:
                              nodePaiName   = pw.parentNode.nodeName
                              if (nodePaiName=='sic'):
                                 sic     = doc2.createElement("sic")
                                 sic.appendChild(pw)
                                 choice.appendChild(sic)

                        s2.appendChild(choice)
```

```
            p2.appendChild(s2)

        body.appendChild(p2)



nomeArqG = nomeArqM + ".03.xml"
f                = codecs.open(nomeArqG, "w", "utf-8")
f.write(doc2.toprettyxml(indent="  "))
f.close()
print "\n"
print "arquivo processado com sucesso " + "\n"
print "nome do arquivo: "  + nomeArqG   + "\n"
print "\n"
print "\n"
print "------------------------------------------------------------\n"
```

# ANEXO I – FUNÇÃO *ATANOTAC.PY* ASTRO 1.1

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Astro Corpus Annotation Tool
#
# 2013 - Author-  Mardonio Franca <mardfranca@gmail.com>
#
# URL: <http://http://www.rotadasespeciarias.art.br/astrolabio/astrolabio/fonte.zip>
# This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 Unported License
#
# $Id: ATanotaC.py $

import nltk
import os
import sys
import codecs
import xml.dom.minidom
from   xml.dom.minidom import *

caminho = '/home/franca/'
sys.path.append(caminho)

import  Aelius
from    Aelius.Toqueniza import TOK_PORT_LX
from    Aelius import Extras,Toqueniza, AnotaCorpus
from    Aelius.Extras import carrega

from    AnotaCorpus import *

import  Astro
from    Astro import Corretor



class ATanotaC:

      arqNome = None
```

```
def __init__(self, arqNome):
    self.arqNome = arqNome


def executa(self):

    n  = 1
    ntk =0
    nomeArqM =  self.arqNome
    pagina       = parse(nomeArqM)

    print "em processamento"
    doc = Document()
    xml = doc.createElement("xml")
    xml.setAttribute("version", "1.0")
    xml.setAttribute("encoding", "UTF-8")
    doc.appendChild(xml)
    TEI          = doc.createElement("TEI")
    TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
    xml.appendChild(TEI)
    texto = doc.createElement("text")
    TEI.appendChild(texto)
    body = doc.createElement("body")
    body.setAttribute("id", "body")
    texto.appendChild(body)

    paragrafos  = pagina.getElementsByTagName("p")
    np=0;
    ns=0;
    nch=0;
    for paragrafo in paragrafos:
        np = np +1
        p = doc.createElement("p")
        p.setAttribute("xml:id", "p" + str(np))
        body.appendChild(p)
        sentencas = paragrafo.getElementsByTagName("s")
        for sentenca in sentencas:
            ns = ns + 1
            s   = doc.createElement("s")
```

```
        s.setAttribute("xml:id", "s" + str(ns))
        p.appendChild(s)
        frase = ""
        palavras = sentenca.getElementsByTagName("w")
        for palavra in palavras:
           if palavra.hasAttribute("xml:id"):
              ch = palavra.parentNode.nodeName
              if (ch != 'sic') and (ch!='del') :
                 frase = frase + " " + palavra.childNodes[0].data
        tk     =  Toqueniza.TOK_PORT_LX.tokenize(frase)
        mac =   Extras.carrega("AeliusHunPosMacMorpho")
        f      =   AnotaCorpus.anota_sentencas([tk],mac,'hunpos', separacao_contracoes=True)
        for lPalavraAnotada in f[0]:
              w    = doc.createElement("w")
              w.setAttribute("xml:id", str(n))
              n= n+ 1
              w.setAttribute("type", lPalavraAnotada[1])
              s.appendChild(w)
              ptext = doc.createTextNode(lPalavraAnotada[0])
              w.appendChild(ptext)
doc2 = Document()
xml2 = doc2.createElement("xml")
xml2.setAttribute("version", "1.0")
xml2.setAttribute("encoding", "UTF-8")
doc2.appendChild(xml2)

TEI         = doc2.createElement("TEI")
TEI.setAttribute("xmlns", "http://www.tei-c.org/ns/1.0")
xml2.appendChild(TEI)

teiHeader = doc2.createElement("teiHeader")

teiHeader = pagina.getElementsByTagName("teiHeader")[0] .....

TEI.appendChild(teiHeader)

texto = doc2.createElement("text")
TEI.appendChild(texto)

body = doc2.createElement("body")
```

```
        body.setAttribute("id", "body")
        texto.appendChild(body)
        i = 0
        paragrafos     = pagina.getElementsByTagName("p")
        for paragrafo in paragrafos:
            p2 = doc2.createElement("p")
            body.appendChild(p2)
            sentencas = paragrafo.getElementsByTagName("s")
            for sentenca in sentencas:
                s2 =  doc2.createElement("s")
                p2.appendChild(s2)
                palavrasPs     = sentenca.getElementsByTagName("w") # palavra anotada
                for palavrasP in palavrasPs:
                    nodePaiName   = palavrasP.parentNode.nodeName
                    nodePaiPai = palavrasP.parentNode.parentNode
                    if  (nodePaiName=='s') :
                       palavrasAs     = doc.getElementsByTagName("w") # palavra anotada
                       for palavrasA in palavrasAs:
                           if palavrasP.getAttribute("xml:id")==  palavrasA.getAttribute("xml:id"):
                               s2.appendChild(palavrasA)
                               ntk = ntk + 1
                    elif (nodePaiName=='corr') :
                        choice  =  doc2.createElement("choice")
                        corr     =   doc2.createElement("corr")
                        palavrasAs     = doc.getElementsByTagName("w") # palavra anotada
                        for palavrasA in palavrasAs:
                            if palavrasP.getAttribute("xml:id")==  palavrasA.getAttribute("xml:id"):
                                corr.appendChild(palavrasA)
                                ntk = ntk + 1
                                nodePaiPaiP = palavrasP.parentNode.parentNode #pesquisa o nó avô -
choice
                                palavrasSics     =   nodePaiPaiP.getElementsByTagName("w")   #   palavra
anotada
                                for palavrasSic in palavrasSics:
                                    if (palavrasSic.parentNode.nodeName!='corr'):
                                        nodeName = palavrasSic.parentNode.nodeName
                                        nodeSic     =   doc2.createElement(nodeName)
                                        nodeSic.appendChild(palavrasSic)
                                        ntk = ntk + 1
                                        choice.appendChild(nodeSic)
```

```
                    choice.appendChild(corr)
                    s2.appendChild(choice)
                    palavrasTs = pagina.getElementsByTagName("w")
        p2.appendChild(s2)
        body.appendChild(p2)

nomeArqG = nomeArqM + ".03.xml"
f               = codecs.open(nomeArqG, "w", "utf-8")
f.write(doc2.toprettyxml(indent="  "))
f.close()
print "\n"
print "arquivo processado com sucesso " + "\n"
print "nome do arquivo: "  + nomeArqG   + "\n"
print "\n"
print "\n"
print "------------------------------------------------------------\n"
```

## ANEXO J – *TAGSET* DO MAC-MORPHO

| CLASSE GRAMATICAL | ETIQUETA |
|---|---|
| ADJETIVO | **ADJ** |
| ADVÉRBIO | **ADV** |
| ADVÉRBIO CONECTIVO SUBORDINATIVO | **ADV-KS** |
| ADVÉRBIO RELATIVO SUBORDINATIVO | **ADV-KS-REL** |
| ARTIGO (def. ou indef.) | **ART** |
| CONJUNÇÃO SUBORDINATIVA | **KS** |
| INTERJEIÇÃO | **IN** |
| NOME | **N** |
| NOME PRÓPRIO | **NPROP** |
| NUMERAL | **NUM** |
| PARTICÍPIO | **PCP** |
| PALAVRA DENOTATIVA | **PDEN** |
| PREPOSIÇÃO | **PREP** |
| PRONOME ADJETIVO | **PROADJ** |
| PRONOME CONECTIVO SUBORDINATIVO | **PRO-KS** |
| PRONOME PESSOAL | **PROPESS** |
| PRONOME RELATIVO CONECTIVO SUBORDINATIVO | **PRO-KS-REL** |
| PRONOME SUBSTANTIVO | **PROSUB** |
| VERBO | **V** |
| VERBO AUXILIAR | **VAUX** |
| SÍMBOLO DE MOEDA CORRENTE | **CUR** |
| ETIQUETAS COMPLEMENTARES (Estrangeirismos; Apostos; Dados; Números de Telefone; Datas; Horas; e Disjunção) | **EST**<br>**AP**<br>**DAD**<br>**TEL**<br>**DAT**<br>**HOR**<br>**[ ]** |
| CONTRAÇÕES e ÊNCLISES | **+** |
| MESÓCLISES | **!** |

Fonte: http://www.nilc.icmc.usp.br/lacioweb/downloads/Manual_Mac_Morpho_v10.zip.

# ANEXO K – *TAGSET* DO LX-TAGGER

| Tag | Categoria | Examplos |
|---|---|---|
| ADJ | Adjectivo | bom, brilhante, eficaz, … |
| ADV | Advérbio | hoje, já, sim, felizmente, … |
| CARD | Cardinal | zero, dez, cem, mil, … |
| CJ | Conjunção | e, ou, tal como, … |
| CL | Cliticos | o, lhe, se, … |
| CN | Nome | computador, cidade, ideia, … |
| DA | Artigo Definido | o, os, … |
| DEM | Demonstrativo | este, esses, aquele, … |
| DFR | Fracções | meio, terço, décimo, %, … |
| DGTR | Números Romanos | VI, LX, MMIII, MCMXCIX, … |
| DGT | Dígitos | 0, 1, 42, 12345, 67890, … |
| DM | Marcadores de Discurso | olá, … |
| EADR | Endereço Electrónico | http://www.di.fc.ul.pt, … |
| EOE | Final de Enumeração | etc |
| EXC | Exclamativa | ah, ei, etc. |
| GER | Gerúndio | sendo, afirmando, vivendo, … |
| GERAUX | Gerúndio "ter"/"haver" em termos compostos | tendo, havendo … |
| IA | Artigo Indefinido | uns, umas, … |
| IND | Indefinidos | tudo, alguém, ninguém, … |
| INF | Infinitivo | ser, afirmar, viver, … |
| INFAUX | Infinitivo "ter"/"haver" em termos compostos | ter, haver … |
| INT | Interrogativos | quem, como, quando, … |
| ITJ | Interjeição | bolas, caramba, … |
| LTR | Letras | a, b, c, … |
| MGT | Classe de Magnitude | unidade, dezena, dúzia, resma, … |
| MTH | Meses | Janeiro, Dezembro, … |
| NP | Sintagma Nominal | idem, … |
| ORD | Ordinal | primeiro, centésimo, penúltimo, … |
| PADR | Parte de Endereço | Rua, av., rot., … |
| PNM | Parte de Nome Próprio | Lisboa, António, João, … |
| PNT | Pontuação | ., ?, (, … |

| | | |
|---|---|---|
| POSS | Possessivos | meu, teu, seu, … |
| PPA | Particípio Passado que não ocorre em termos compostos | afirmados, vivida, … |
| PP | Sintagma Preposicional | algures, … |
| PPT | Particípio Passado em termos compostos | sido, afirmado, vivido, … |
| PREP | Preposição | de, para, em redor de, … |
| PRS | Pronome Pessoal | eu, tu, ele, … |
| QNT | Quantificadores | todos, muitos, nenhum, … |
| REL | Pronome Relativo | que, cujo, tal que, … |
| STT | Títulos Sociais | Presidente, drª., prof., … |
| SYB | Símbolos | @, #, &, … |
| TERMN | Terminações opcionais | (s), (as), … |
| UM | "um" ou "uma" | um, uma |
| UNIT | Abreviatura de unidade de medida | kg., km., … |
| VAUX | Verbos finitos "ter" ou "haver" em forma verbal composta | temos, haveriam, … |
| V | Verbos | falou, falaria, … |
| WD | Dias da Semana | segunda, terça-feira, sábado, … |
| Expressões Multi-palavra | | |
| LADV1…LADVn | Advérbios multi-palavra | de facto, em suma, um pouco, … |
| LCJ1…LCJn | Conjunções multi-palavra | assim como, já que, … |
| LDEM1…LDEMn | Demonstrativos multi-palavra | o mesmo, … |
| LDFR1…LDFRn | Fracções multi-palavra | por cento |
| LDM1…LDMn | Marcadores Discursivos multi-palavra | pois não, até logo, … |
| LITJ1…LITJn | Interjeições multi-palavra | meu Deus |
| LPRS1…LPRSn | Pronomes Pessoais multi-palavra | a gente, si mesmo, V. Exa., … |
| LPREP1…LPREPn | Preposições multi-palavra | através de, a partir de, … |
| LQD1…LQDn | Quantificadores multi-palavra | uns quantos, … |
| LREL1…LRELn | Relativos multi-palavra | tal como, … |

Fonte: http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LXTagger.html.