Revista RBBA

Revista Binacional Brasil Argentina

JSON UMA ALTERNATIVA PARA CORPUS LINGUÍSTICO ANOTADO EM PADRÃO XML

JSON UNA ALTERNATIVA PARA EL CORPUS DEL LENGUAJE ANOTADO EN LA NORMA XML

Aline Silva Costa
Universidade Estadual do Sudoeste da Bahia – UESB
Instituto Federal de Educação Tecnologia e Ciência da Bahia - IFBA
alinecosta@ifba.edu.br

Bruno Silvério Costa
Universidade Estadual do Sudoeste da Bahia – UESB
Instituto Federal de Educação Tecnologia e Ciência da Bahia - IFBA
brunosilverio@ifba.edu.br

Romenito Pereira Damaceno
Instituto Federal de Educação Tecnologia e Ciência da Bahia - IFBA
romen_damaceno@hotmail.com

Cristiane Namiuti
Universidade Estadual do Sudoeste da Bahia – UESB
cristianenamiuti@uesb.edu.br

Jorge Viana Santos
Universidade Estadual do Sudoeste da Bahia – UESB
viana.jorge.viana@gmail.com

Resumo

Para as investigações em Linguística nas Humanidades Digitais, sobretudo para a formulação de hipóteses sobre gramáticas nos estudos de Linguística Histórica, necessita-se de um grande volume de dados, fato que intensificou a construção e implementação de corpora anotados que crescem em tamanho exigindo maior grau de escalabilidade. Neste artigo discute-se a viabilidade técnica de uma solução computacional alternativa à linguagem XML (*eXtensible Markup Language*) para corpora linguísticos anotados. A linguagem XML tem sido utilizada em vários corpora que se baseiam no Corpus anotado do português histórico Tycho Brahe (CTB), como o Corpus de Documentos Oitocentistas de Vitória da Conquista (DOViC) e o Corpus Eletrônico de Documentos Históricos do Sertão (CE-DOHS). A linguagem XML pode apresentar problemas de performance para grande volume de dados, além de alto custo de memória. O crescimento de bancos de dados não relacionais, com

características de alta flexibilidade e performance, associado aos potenciais problemas de desempenho da anotação XML, motivou uma pesquisa de viabilidade técnica de uma solução computacional alternativa para representação e armazenamento atual dos textos do corpus DOViC em um banco de dados NoSQL no formato JSON (*JavaScript Object Notation*) (MONGODB, 2008). A pesquisa aqui apresentada verifica a viabilidade da representação,compara a performance obtida em buscas morfossintáticas feitas na anotação proposta (Banco de dados NoSQL e formato JSON) com a anotação e armazenamento atual do corpus DOViC (Sistema de arquivos e formato XML), e faz uma análise de outros aspectos da proposta. Os resultados obtidos no tocante à performance da proposta JSON indicam viabilidade técnica dessa vertente computacional.. Não obstante, para além da performance, o XML apresenta maiores vantagens de interoperabilidade por ser amplamente aceita como padrão para anotação de corpora.
Palavras-chave: Corpus anotado. JSON. XML.

Resumen

Para las investigaciones en Lingüística en Humanidades Digitales, especialmente para la formulación de hipótesis sobre las gramáticas en los estudios de Lingüística Histórica, se requiere un gran volumen de datos, un hecho que intensificó la construcción e implementación de los corpora anotados que crecen en tamaño y requieren mayor escalabilidad. Este documento analiza la viabilidad técnica de una solución computacional alternativa al XML (*eXtensible Markup Language*) para corpora lingüísticos anotados. El lenguaje XML se ha utilizado en muchos corpora que se basan en el corpus anotado del portugués histórico Tycho Brahe (CTB), como el Corpus de Documentos Oitocentistas de Vitória da Conquista (DOViC) y el Corpus Eletrônico de Documentos Históricos do Sertão (CE-DOHS). El lenguaje XML puede presentar problemas de rendimiento para grandes datos, así como un alto costo de memoria. El crecimiento de las bases de datos no relacionales, con alta flexibilidad y características de rendimiento, asociadas con posibles problemas de rendimiento de la anotación XML, motivó un estudio de viabilidad técnica de una solución computacional alternativa para la representación y el almacenamiento actual de textos de corpus DOViC. Base de datos NoSQL en formato JSON (*JavaScript Object Notation*) (MONGODB, 2008). La investigación presentada aquí verifica la viabilidad de la representación, compara el rendimiento obtenido en las búsquedas morfosintácticas realizadas en la propuesta (base de datos NoSQL y formato JSON) con la actual anotación y almacenamiento del corpus DOViC (sistema de archivos y formato XML), y hace un análisis de otros aspectos de la propuesta. Los resultados obtenidos con respecto al rendimiento de la propuesta JSON indican la viabilidad técnica de este

aspecto computacional. Además del rendimiento, XML tiene importantes ventajas de interoperabilidad, ya que es ampliamente aceptado como el estándar para la anotación de corpus.
Palabras clave: Corpus anotado. JSON. XML.

1- Introdução

Para a formulação de hipóteses sobre gramáticas nos estudos de Linguística Histórica, necessita-se de um grande volume de dados, fato que intensificou a construção e implementação de corpora anotados que crescem em tamanho exigindo maior grau de escalabilidade. Nesse sentido, no âmbito das Humanidades Digitais, a construção de corpora tem sido objeto da Linguística de Corpus[i], uma subárea da Linguística Computacional[ii].

Os recursos computacionais para exploração de *corpora* exercem um papel importante por permitir análises estatísticas e buscas por padrões em grandes volumes de dados. Assim, softwares para análises de corpora têm sido desenvolvidos e utilizados como aliados em pesquisas acerca de fenômenos linguísticos (MELLO; SOUZA, 2012).

Nessa perspectiva, o Laboratório de Pesquisa em Linguística de Corpus (Lapelinc) da Universidade Estadual do Sudoeste da Bahia (UESB), tem intensificado a investigação, aplicação e uso de novas soluções técnicas para o gerenciamento, catalogação, edição especializada e buscas automáticas em textos de *corpora* anotados. Um produto desenvolvido no Lapelinc, no âmbito dos projetos *Memória conquistense: implementação de um* corpus *digital* (NAMIUTI; SANTOS, 2013 - CNPq N485098/2013-0), *Corpora digitais para a história do português brasileiro - documentos históricos da região sudoeste da Bahia: aliança PHPB-Tycho Brahe* (SANTOS; NAMIUTI, 2010 - FAPESB PET0034/2010) e Corpora digitais de documentos históricos da imperial Vila da Victoria, atual Vitória da Conquista-Bahia: resgate e preservação do patrimônio linguístico e da memória da escravidão na Bahia (SANTOS; NAMIUTI 2016 – FAPESB APP0014/2016) é o corpus digital DOViC (Corpus de Documentos Oitocentistas de Vitória da Conquista) (SANTOS; NAMIUTI, 2016), constituído por documentos notariais manuscritos do século XVIII e XIX pertencentes ao Primeiro Cartório de notas do Fórum de Vitória da Conquista.

Os textos do corpus DOViC são editados e anotados nos mesmos moldes do Corpus Tycho Brahe (CTB), corpus digital composto de textos em português de autores nascidos

entre 1380 e 1881, desenvolvido na Universidade Estadual de Campinas (GALVES; ANDRADE; FARIA, 2017).

A linguagem XML (*eXtensible Markup Language*) é uma linguagem de marcação amplamente aceita para anotação de corpus. Essa representação tem sido empregada em muitos projetos e padrões de anotação linguística (ZELDES, 2019). Vários corpora que se baseiam na metodologia do Corpus anotado do português histórico Tycho Brahe (CTB), como o Corpus Eletrônico de Documentos Históricos do Sertão (CE-DOHS) e o corpus de Documentos Oitocentistas de Vitória da Conquista (DOViC), também utilizam a linguagem XML para anotação dos documentos.

Neste artigo[iii] discute-se a viabilidade técnica de uma solução computacional alternativa à linguagem XML para corpora linguísticos anotados. Assim, a pesquisa aqui apresentada verifica a viabilidade da representação em outro formato e compara a performance obtida em buscas morfossintáticas feitas nessa proposta (Banco de dados NoSQL e formato JSON) com a anotação e armazenamento atual do corpus DOViC (Sistema de arquivos e formato XML). Por fim, discutem-se os aspectos de interoperabilidade e persistência da proposta em JSON.

Os documentos que compõem o DOViC recebem anotações acerca da estrutura e formatação dos textos, das interferências de edição de grafia e segmentação e de informações linguísticas nos níveis morfossintático e sintático. Todas as anotações são mantidas em arquivos de texto, no formato XML. (COSTA, 2015). Uma outra vertente computacional para o gerenciamento, armazenamento e recuperação dessas anotações é a utilização de Sistemas Gerenciadores de Bancos de Dados (SGBDs). A linguagem XML pode apresentar problemas de performance com grandes volumes de dados, além de um alto custo de memória (ABINADER, 2006). Ademais, o surgimento de um número crescente de bancos de dados não relacionais, chamados de banco de dados NoSQL, motivou a investigação da viabilidade e performance da utilização destes como uma alternativa à anotação em XML. Os bancos NoSQL possuem características de flexibilidade, disponibilidade e performance maiores que os bancos tradicionais (BRITO, 2010). A característica de flexibilidade é interessante no que tange à anotação de corpora linguísticos, uma vez que os esquemas flexíveis dos bancos NoSQL permitem a representação e armazenamento de estruturas hierárquicas, o que em tese torna natural o armazenamento da estrutura sintática de constituintes de uma sentença.

2- Metodologia

Para os testes de viabilidade e performance com a representação proposta, foram utilizados os textos anotados do corpus DOViC, compilado e anotado com o formato XML. Considerou-se o acesso aos documentos que compõem o corpus e o trabalho já desenvolvido por Costa (2015) para realizar buscas morfossintáticas sobre o documento intitulado "Carta de Liberdade da cabra de nome Sofia", escrita em 1845. Assim, este trabalho se configura como um estudo de caso do Corpus DOViC, com o propósito de testar a viabilidade e performance de uma nova representação para corpora anotados nestes moldes, contrastando-a com o armazenamento e anotação já utilizados.

O SGBD NoSQL escolhido para o estudo foi o MongoDB, por ser um dos bancos de dados NoSQL mais populares e possuir código aberto (MONGODB, 2008). Ademais, a similaridade do formato JSON utilizado no MongoDB com XML também contribuiu para a escolha deste banco. As anotações morfossintática e de edições (modernização de grafia, uniformização de pontuação, expansão de abreviaturas, etc) utilizadas no DOViC foram convertidas para o formato JSON na representação proposta e inseridas no banco de dados MongoDB para o documento "Carta de Liberdade da cabra de nome Sofia".

Apresentamos os resultados obtidos comparando a anotação JSON com a anotação já utilizada no corpus DOViC, mostrando o desempenho das buscas realizadas com ambas e fazendo uma análise de outros aspectos da proposta para corpora anotados. Para comparar a performance entre as anotações, obtivemos o tempo de respostas das consultas no corpus através de aplicações desenvolvidas em linguagem de programação Java[iv]. Com a criação dos ambientes de teste, foi feita a coleta dos tempos de execução para cada consulta (ou busca morfossintática), permitindo avaliar o melhor desempenho computacional entre as anotações e buscas realizadas.

3- Corpora anotados e a linguagem XML

A anotação (ou *tagging*) é o processo de adicionar novas informações em textos fontes, seja por humanos ou por sistemas treinados para a tarefa (anotação automática). As anotações linguísticas podem ser de vários tipos e níveis, representando informações morfológicas, sintáticas ou semânticas. A anotação que marca as palavras com suas classes

gramaticais é conhecida como *Part-Of-Speech tagging* (*POS tagging*) e pode trazer informações morfológicas ou morfossintáticas. A anotação em nível sintático é realizada por meio da marcação da estrutura sintática de constituintes nas sentenças dos textos do corpus. A forma mais comum de representar esse tipo de informação é por meio de uma estrutura arbórea (MEGERDOOMIAN, 2003; DAMACENO, 2018).

Para anotações em corpora há diversos padrões para representação das informações sintáticas e morfossintáticas, que podem variar ao longo de diferentes abordagens de análise. Entre os padrões/projetos de anotação que se destacam estão: NITE XML[v], *Tiger*-XML[vi], *Text Encoding Initiative* (TEI)[vii], *Corpus Encoding Standard* (CES), *Corpus Encoding Standard* for XML (XCES) e padrão ISO TC37/SC4[viii]. Todos estes padrões/projetos empregam ou recomendam a linguagem XML, a qual consiste numa linguagem de marcação de padrão aberto, propícia ao armazenamento de dados estruturados e semiestruturados, proposta pelo W3C (*World Wide Web Consortium).* Documentos XML são textos que representam dados de maneira estruturada utilizando um conjunto de etiquetas (*tags*) ou elementos. Essas *tags* não são predefinidas, são projetadas para serem auto descritivas, possibilitando aos autores definirem suas próprias *tags*, o que torna a XML uma importante metalinguagem para representação de qualquer tipo de dado (SILVA FILHO, 2004; DEITEL et al., 2005; W3SCHOOLS, 2017). Consultas a dados em arquivos XML são possíveis através de linguagens de consulta, como XPath[ix] e XQuery[x], as quais são padrões para navegação em documentos XML reconhecidos pelo W3C (W3C, 2010) .

4- JSON (*JavaScript Object Notation*) e Bancos de Dados NoSQL (*Not Only SQL*)

A W3Shools (2017) define JSON como uma sintaxe para armazenamento e transmissão de dados escrito em uma notação de objeto JavaScript[xi]. A sintaxe JSON é derivada da forma utilizada pelo JavaScript para representar informações. Seu formato é apenas texto, considerado leve para intercâmbio de dados, fácil para os seres humanos lerem e escreverem e para máquinas analisarem. Fonseca e Simões (2007) descrevem que o JSON foi desenhado com o objetivo de ser simples, portável e textual. Assim como XML, JSON é um padrão reconhecido pelo W3C para intercâmbio de dados. Vem substituindo o uso de XML em aplicações baseadas em serviços Web (W3SCHOOLS, 2017).

A ideia utilizada pelo JSON para representar informações é intuitiva: para cada valor representado, atribui-se um nome que descreve o seu significado (GONÇALVES, 2012). Na figura 1 é apresentado um exemplo de um formato de dados em JSON, que segue uma estrutura de objeto. No exemplo, "título", "resumo", "ano" e "gênero" são classificados como chaves. Para cada uma das três primeiras chaves, é associado um valor, que são: "JSON x XML", "o duelo de dois modelos de representação de informações" e "2012". Já para a última chave, "gênero", é associado um *array*[xii] com três valores: "aventura", "ação" e "ficção".

```
{
  "titulo": "JSON x XML",
  "resumo": "o duelo de dois modelos de representação de informações",
  "ano": 2012,
  "genero": ["aventura", "ação", "ficção"]
 }
```

Figura 1- Estrutura do Documento JSON
Fonte: Gonçalves (2012)

Existem vários exemplos de banco de dados[xiii] e seus Sistemas de Gerenciamento (SGBDs), "O principal objetivo de um SGDB é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados "(SILBERSCHATZ; KORTH; SUDARSHAN, 1999, p. 1). Conforme Brito (2010), os SGBDs mais difundidos em aplicações são os dos Modelos Relacionais, por terem sido utilizados em praticamente todos os tipos de sistemas de bancos de dados nas últimas décadas. Porém, o crescimento do volume de dados e certos fatores limitantes ao relacional têm possibilitado que modelos alternativos de banco de dados sejam utilizados em tais circunstâncias. Motivados principalmente pela questão da escalabilidade do sistema, uma nova geração de bancos de dados conhecidos como NoSQL (*Not Only SQL* – não apenas SQL) vem ganhando força e espaço no mercado, sendo utilizados onde é necessária uma maior flexibilidade na estruturação do banco e, principalmente, em casos em que o modelo relacional não apresente performance adequada (BRITO, 2010). O modelo NoSQL surgiu para atender aos requisitos de gerenciamento de grandes volumes de dados, semiestruturados ou não estruturados. Isto permite que as aplicações tenham vantagens como: alta disponibilidade, escalabilidade, esquema flexível e alta performance (LÓSCIO et al., 2011).

O MongoDB é um SGBD do tipo NoSQL que armazena os dados na forma de documentos em representação JSON (MARTINS FILHO, 2015). Apresenta vantagens como acesso mais rápido aos dados, facilidade de uso e flexibilidade no esquema de dados. Com a capacidade de alterar a estrutura de registros já armazenados, o MongoDB é útil para representar relações hierárquicas, armazenar matrizes, e outras estruturas mais complexas de maneira simplificada. Como os documentos possuem formato de entrada JSON, as consultas são realizadas com base na chave e no valor (GOMES; BASSO, 2015; SOARES, 2016 MONGODB, 2019).

5- O Corpus DOViC

O corpus digital DOViC (Documentos Oitocentistas de Vitória da Conquista) é composto por textos anotados de manuscritos dos séculos XVIII e XIX, que se distribuem entre cartas de alforria, testamentos, procurações, matrículas de escravos, escrituras de imóveis e atas de eleições municipais, os quais se encontram guardados nos arquivos do Fórum de Vitória da Conquista (SANTOS; NAMIUTI, 2016).

Visando contribuir com a ampliação de pesquisas sobre a história do Português do Brasil e preservar o patrimônio histórico e linguístico da cidade de Vitória da Conquista-Bahia, o corpus foi compilado no âmbito dos projetos *Memória conquistense: implementação de um* corpus *digital* (NAMIUTI; SANTOS, 2013), *Corpora digitais para a história do português brasileiro - documentos históricos da região sudoeste da Bahia: aliança PHPB-Tycho Brahe* (SANTOS; NAMIUTI, 2010 - FAPESB PET0034/2010) e Corpora digitais de documentos históricos da imperial Vila da Victoria, atual Vitória da Conquista-Bahia: resgate e preservação do patrimônio linguístico e da memória da escravidão na Bahia (SANTOS; NAMIUTI 2016 – FAPESB APP0014/2016).

Os textos do corpus DOViC são editados e anotados nos mesmos moldes do Corpus Tycho Brahe (CTB), corpus digital composto de textos em português de autores nascidos entre 1380 e 1845, desenvolvido na Universidade Estadual de Campinas (GALVES; ANDRADE; FARIA, 2017). Os documentos que compõem o DOViC recebem anotações acerca da estrutura e formatação dos textos, das interferências de edição de grafia e segmentação e de informações linguísticas nos níveis morfossintático e sintático (COSTA, 2015).

No corpus DOViC as anotações morfossintáticas e de edições são realizadas com etiquetas da linguagem XML. A figura 1 mostra um trecho do arquivo XML correspondente ao documento "Carta de liberdade da Cabra de nome Sofia", com anotações morfossintática e de edições. A etiqueta <w> marca uma palavra do texto. A etiqueta <o> faz anotação da palavra na sua forma original. A etiqueta <m> traz a informação morfossintática, com a classe gramatical correspondente no atributo "v". Como exemplo na figura 1, <m v="ADV"/> indica que a palavra é um advérbio. A etiqueta <e> faz anotação da edição. Na figura 2, temos a anotação da modernização do termo "hoji" para "hoje".

```
- <w id="53">
  - <o>
      hoji
      <bk t="I" id="bk_6" />
    </o>
    <e t="mod">hoje</e>
    <m v="ADV" />
  </w>
```

Figura 2 - Trecho de documento do corpus DOViC anotado em XML
Fonte: Santos; Namiuti (2016)

6- Proposta de anotação no formato JSON para representação de corpus anotado no banco de dados NoSQL

Apresentamos aqui a proposta de uma anotação morfossintática e de edições para qualquer corpus anotado nos mesmos moldes do CTB no formato JSON, que é o formato de entrada para o SGBD MongoDB. Com a anotação proposta nesse formato, foi possível representar um documento do corpus DOViC no banco MongoDB, armazenando-o e recuperando informações linguísticas através de buscas, tornando possível testar a viabilidade e a performance das consultas.

Existe similaridade entre os formatos XML e JSON. Ambos são formatos de texto simples, capazes de representar, no formato tabular, informação complexa e difícil, como a estrutura hierárquica de uma anotação sintática. Enquanto a XML se baseia na sintaxe de etiquetas, atributos e valores, o JSON utiliza um formato mais conciso, com pares "chave: valor", em que, para cada valor representado, atribui-se um nome (chave) que descreve o seu significado (W3SCHOOLS, 2017).

Para elaborar a proposta, levou-se em consideração as seguintes características: (i) manter o máximo de etiquetas já estabelecidas nas estruturas XML, (ii) aproximar-se ao máximo da hierarquia estabelecida nas estruturas XML e (iii) atender às necessidades linguísticas e filológicas, mantendo a preparação de conteúdo para análises linguísticas o mais simples e eficiente possível (DAMACENO, 2018).

Em termos gerais, a codificação da estrutura JSON é muito flexível, e isso se dá pelas suas duas estruturas básicas: objetos e *arrays* (vetores). Os objetos que contém conjuntos de pares "chave: valor" são definidos entre chaves ("{ }"), enquanto que os vetores, que contém uma sequência de valores ordenados (strings, objetos, vetores, numéricos, null, etc), são expressos entre colchetes ("[ ]"). Entretanto, em razão dessa flexibilidade, a especificação da estrutura JSON deve ser feita de modo sistemático e bem refletido, para que se possa organizar e modelar adequadamente todas as informações necessárias às consultas linguísticas (DAMACENO, 2018).

Os arquivos anotados em XML dos documentos do corpus DOViC contêm informações morfossintáticas (categoria POS das palavras) e de edições. O quadro 1 apresenta as etiquetas XML e atributos usados nessa anotação.

Etiqueta XML	**Descrição**
<p>	Formam os parágrafos onde estão contidas as divisões de sentença.
<s>	Formam as sentenças onde estão contidas as palavras.
<w>	Indicam os limites de palavras
<o>	Indicam a grafia original da palavra
<e>	Indicam a grafia modernizada da palavra
<m>	Especificam uma etiqueta de *part-of-speech* (POS) responsável por marcar a categoria lexical da palavra.
<bk>	Indicam uma quebra de linha.
Atributos	**Descrição**
id	Atributo identificador de parágrafos "<p>", sentenças "<s>", limites de palavras "<w>" e quebras de linhas "<bk>".
t	Atributo associado a etiqueta "<e>" que indica o tipo de modificação na grafia da palavra.
v	Atributo associado a etiqueta "<m>" que indica o valor da categoria POS da palavra.

Quadro 1 - Etiquetas e atributos utilizados na anotação XML do corpus DOViC
Fonte: Damaceno (2018) adaptado de Paixão de Souza (2014); Costa (2015)

Para a anotação proposta em JSON, inicialmente foi estabelecida uma estrutura capaz de codificar as informações existentes no arquivo XML, as quais são apresentadas nos quadros 7 e 8. Procuramos manter os mesmos nomes das etiquetas e atributos usados na anotação XML. Os nomes das *tags* definidas no arquivo XML que indicam início de parágrafos e sentenças, que são "<p>" e "<s>", respectivamente, permaneceram com os mesmos nomes na anotação JSON. Assim, inserimos as chaves "p" e "s" na proposta. As *tags* que acompanham as palavras como a forma original "<o>", editada "<e>" e o atributo que indica o tipo de edição "t" também permanecem com os nomes inalterados na proposta em JSON. Como uma palavra poderá sofrer mais de uma alteração na sua modernização, a *tag* "<e>" da anotação XML foi mapeada para uma chave JSON, transformando-se em um vetor ou um objeto contendo essas edições. Dessa forma, a chave "e" contém dentro da sua estrutura de objetos as chaves "c" com o conteúdo da palavra modernizada e "t", que indica o tipo de edição sofrida pela palavra (DAMACENO, 2018).

Apesar de a *tag* "<m>", que indica a categoria POS, permanecer com o nome inalterado neste trabalho, houve uma pequena alteração em relação à sua sintaxe no arquivo XML. Na anotação XML havia o atributo "v" para esta *tag*, o qual foi "excluído" da nova anotação (não há chave correspondente a ele), tendo o seu valor mapeado diretamente como valor da chave "m". Essa exclusão foi necessária devido ao formato JSON não trabalhar com sintaxe "elemento/atributo/valor", mas apenas com a estrutura "chave/valor". Os identificadores de parágrafos e sentenças (atributos "id") permanecem na nova anotação com a sintaxe "_id" e funcionam como chaves obrigatórias para o sucesso da inserção do documento no MongoDB. Também foram mantidos na anotação proposta os identificadores de limites de palavras "id", que seguem a mesma estrutura de Costa (2015), modelo de anotação que possui um identificador para cada palavra. Não houve a necessidade da *tag* "<w>" definida no arquivo XML, pois a delimitação de palavras no arquivo JSON ocorre na abertura e fechamento das chaves "{ }". A anotação proposta também visa permitir as buscas morfossintáticas que Costa (2015) realizou sobre o arquivo XML do corpus DOViC. Sendo assim, o acréscimo da chave "*path*" foi necessário para permitir determinados tipos de buscas morfossintáticas, como precedência de termos. Não existe uma etiqueta correspondente a essa informação no arquivo anotado XML. A chave "*path*" deve ser inserida na implementação da conversão do arquivo anotado em JSON, contendo o valor que corresponde às categorias lexicais de todas as palavras da sentença, separadas por vírgula. O quadro 2 contém a relação

das etiquetas e atributos da anotação XML e seu correspondente mapeamento para as chaves que constituem a anotação proposta para o formato de armazenamento do MongoDB (DAMACENO, 2018).

Etiqueta/atributo XML	**Chave JSON na anotação proposta**	**Tipo da Chave JSON**	**Descrição**
Id	_id	String	Identificadores de parágrafos e sentenças.
Id	id	Númerico	Identificadores de palavras.
<p>	p	Vetor de Objetos	Vetores de objetos que formam os parágrafos onde estão contidos as divisões de sentença.
<s>	s	Vetor de Objetos	Vetores de objetos que formam as sentenças onde estão contidas as palavras.
<w>	{ }		Determina limite de palavras
<o>	o	String	Grafia original da palavra
<e>	e	Vetor/Objeto	Vetor ou objeto que conterá a grafia modificada da palavra
Não há etiqueta correspondente.	c	String	Indica o conteúdo modernizado da palavra.
t	t	String	Identifica o tipo de modificação na grafia da palavra.
<m>	m	String	Etiqueta de *part-of-speech* (POS) responsável por marcar a categoria lexical da palavra
<bk>	bk	Objeto	Indica uma quebra de linha.
Não há etiqueta correspondente.	path	String	Especifica todo o caminho ordenado das categorias lexicais de uma sentença.

Quadro 2 - Listagem do mapeamento das etiquetas e atributos da anotação XML para as chaves que constituem a anotação para formato de armazenamento do MongoDB.
Fonte: Damaceno (2018)

A figura 3 mostra um trecho de um arquivo do corpus DOViC, a "Carta de Liberdade da Cabra de nome Sofia", com anotação morfossintática na estrutura XML, e a figura 4 mostra o resultado do mesmo trecho do documento no formato JSON proposto. Na anotação JSON apresentada, uma sentença é representada como um *array* que contém vários objetos, que por sua vez são constituídos de alguns campos ("id", "o", "e", "m", entre outros).

```
<p id="p_1" t="title" f="b">
  <s id="s_1">
    <w id="2">
      <o>Carta</o>
      <m v="NPR"/>
    </w>
    <w id="3">
      <o>de</o>
      <m v="P"/>
    </w>
    <w id="4">
      <o>liberdade</o>
      <m v="N"/>
    </w>
    <w id="5">
      <o>da</o>
      <m v="P+D-F"/>
    </w>
    <w id="6">
      <o>Cabra</o>
      <m v="NPR"/>
    </w>
    <w id="7">
      <o>de</o>
      <m v="P"/>
    </w>
    <w id="8">
      <o>nome<bk t="l" id="bk_1"/></o>
      <m v="N"/>
    </w>
    <w id="9">
      <o>Sofia</o>
      <m v="NPR"/>
    </w>
```

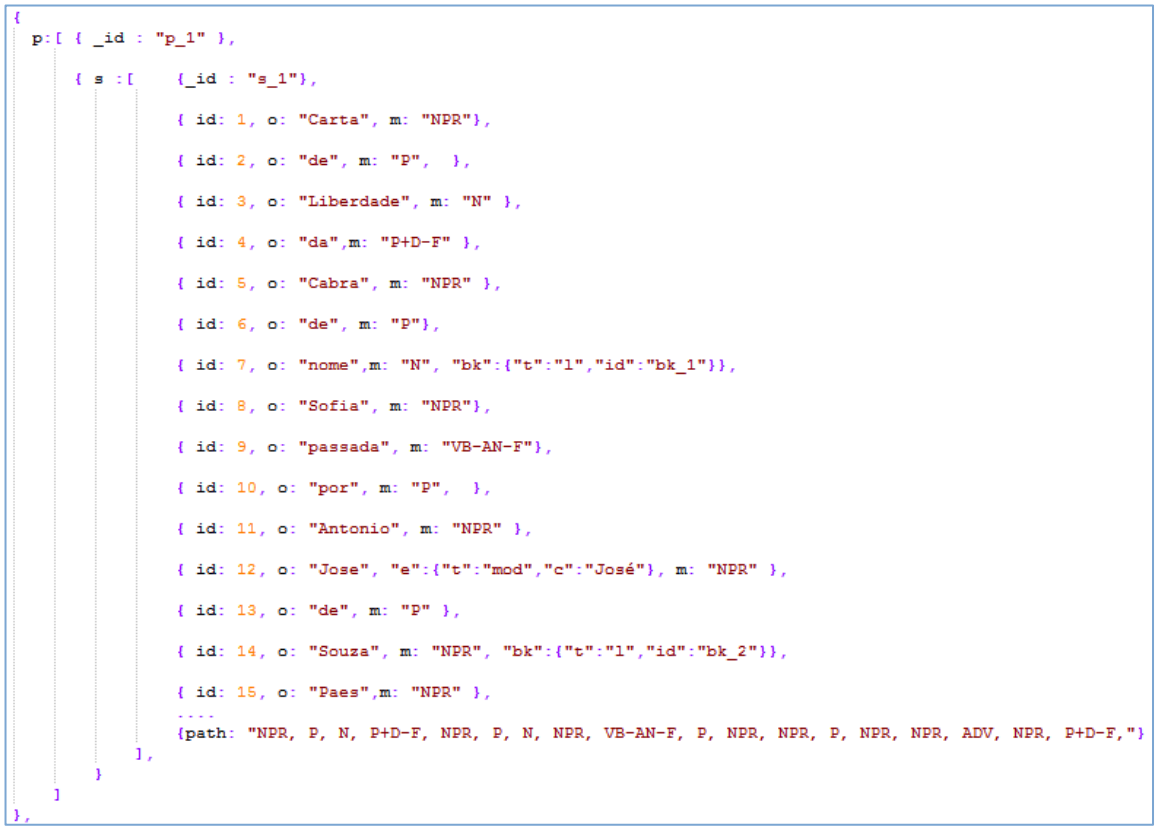
Figura 3 - Trecho de anotação morfossintática no formato XML para o corpus DOViC
Fonte: Santos; Namiuti (2016)

7- Representação no Banco NoSQL e buscas realizadas

Depois do mapeamento das informações existentes no arquivo XML para a codificação JSON e sua inserção no banco de dados, foram realizados os mesmos tipos de buscas morfossintáticas feitas por Costa (2015) sobre o documento "Carta de Liberdade da cabra de nome Sofia", do corpus DOViC.

No MongoDB, as informações podem ser recuperadas por meio da linguagem de consultas disponível no banco. As pesquisas foram feitas pelas chaves "m" contidas dentro do *array* de sentenças do documento, que indicam a anotação da categoria POS. O MongoDB fornece o resultado das buscas trazendo as sentenças na forma original do texto, possibilitando também exibi-lo na forma modernizada, caso se obtenha o resultado da busca

pela chave “e”. Costa (2015) utiliza a linguagem de consulta XQuery para fazer as buscas sobre os textos anotados em XML.



```
{
  p:[ { _id : "p_1" },
      { s :[    {_id : "s_1"},
                { id: 1, o: "Carta", m: "NPR"},
                { id: 2, o: "de", m: "P",  },
                { id: 3, o: "Liberdade", m: "N" },
                { id: 4, o: "da",m: "P+D-F" },
                { id: 5, o: "Cabra", m: "NPR" },
                { id: 6, o: "de", m: "P"},
                { id: 7, o: "nome",m: "N", "bk":{"t":"l","id":"bk_1"}},
                { id: 8, o: "Sofia", m: "NPR"},
                { id: 9, o: "passada", m: "VB-AN-F"},
                { id: 10, o: "por", m: "P",  },
                { id: 11, o: "Antonio", m: "NPR" },
                { id: 12, o: "Jose", "e":{"t":"mod","c":"José"}, m: "NPR" },
                { id: 13, o: "de", m: "P" },
                { id: 14, o: "Souza", m: "NPR", "bk":{"t":"l","id":"bk_2"}},
                { id: 15, o: "Paes",m: "NPR" },
                ....
                {path: "NPR, P, N, P+D-F, NPR, P, N, NPR, VB-AN-F, P, NPR, NPR, P, NPR, NPR, ADV, NPR, P+D-F,"}
          ],
      }
  ]
},
```


Figura 4 - Trecho do Corpus DOViC com anotação morfossintática em JSON
Fonte: Damaceno (2018)

Nos testes realizados, foram utilizados os mesmos parâmetros nas consultas de Costa (2015), as quais foram: Existência, Precedência, Precedência Imediata, Palavra na n-ésima posição das sentenças e Palavras no início ou no fim das sentenças. A linguagem de consulta específica do MongoDB e um *framework* de agregação foram utilizados para realizar as consultas. No arquivo XML, a busca pela precedência de termos é feita através de funções que pesquisam nós irmãos (adjacentes). Com o *framework* de agregação do MongoDB, esse tipo de busca não foi possível, o que requereu a informação da chave adicional com o uso de expressões regulares para a pesquisa (última chave da sentença, conforme figura 4). A flexibilidade do MongoDB com o uso de expressões regulares associadas a esta chave tornou possível esse tipo de buscas. As funções de buscas por Vizinhança não foram implementadas,

devido à ausência de operadores correspondentes aos da linguagem XQuery nas opções de buscas do MongoDB (DAMACENO, 2018).

As figuras 5 e 6 apresentam um exemplo de consulta de agregação realizada para uma das funções de buscas morfossintáticas, a função Existência. A figura 5 mostra a sintaxe de agregação utilizada neste trabalho para realizar uma busca no MongoDB por sentenças onde existem verbos no gerúndio no arquivo armazenado (elementos VB-G). A figura 6 mostra a expressão XQuery utilizada por Costa (2015) para realizar a mesma busca da figura 5 no arquivo XML. Todas as consultas e testes realizados no trabalho podem ser consultados em Damaceno (2018).

```
db.corpora.aggregate([{$match : { "texto.titulo" : "Carta de Alforia da Cabrita Sofia" } },
                      { $unwind : "$texto" },
                      { $unwind : "$texto.p" },
                      {$match : {"texto.p.s.m" : {$exists : true, $in : ["VB-G"]}}},
                      {$group:{_id : null, Sentenças : {$push : "$texto.p.s.o"}}},
                      { $project : {"_id" : 0,}},
                      ])
```

Figura 5 - Expressão de agregação para buscas por sentenças
onde existem verbos no gerúndio (VB-G)
Fonte: Damaceno (2018)

```
for $s in doc('WebContent/arquivo.xml')//document/body/text/sc/p/s
let $sentenca:= data($s/w/o)
where ($s/w/m[@v="VB-G"])
return $sentenca
```

Figura 6 - Expressão XQuery para buscar por sentenças
onde existem verbos no gerúndio
Fonte: Costa (2015)

8- Testes de performance e Discussão (ou análise??) {DISCUSSÃO}

Segundo Hirschman e Mani (2003), entre vários métodos para avaliação dos resultados produzidos pelo sistema, a saída pode ser avaliada por si mesma, sendo comparada com outras saídas, ou com o resultado esperado para determinada entrada.

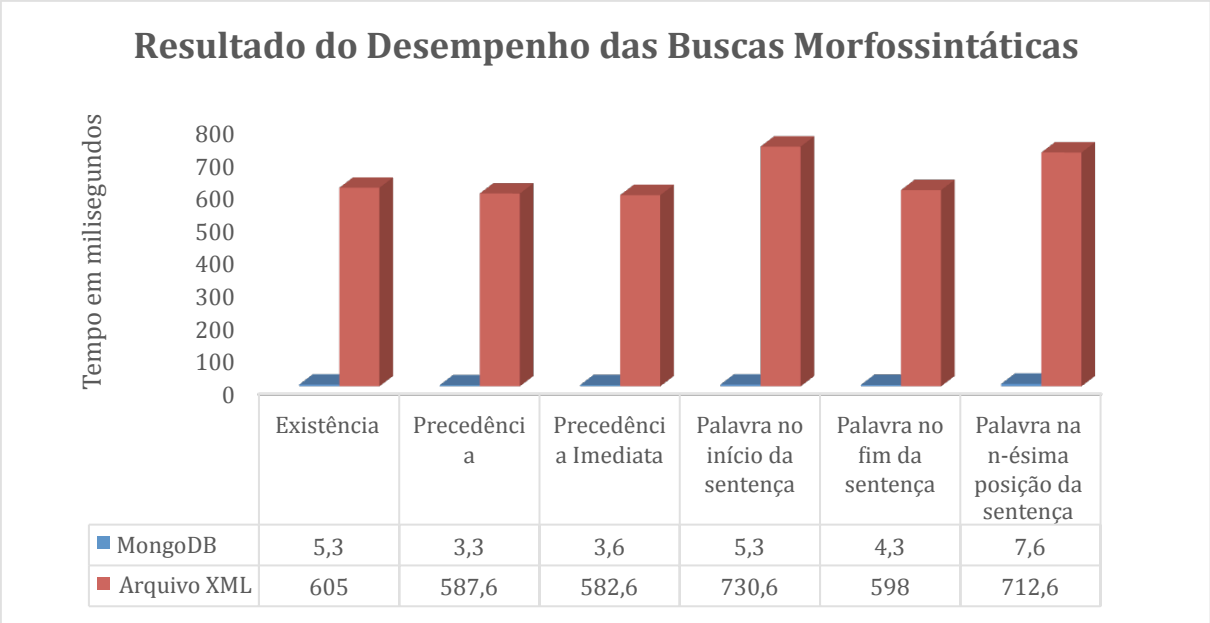
Para avaliar os resultados produzidos pelas buscas morfossintáticas, foi utilizado o método de comparação da saída entre a ferramenta desenvolvido por Costa (2015), o software WebSinc, com a saída produzida pelo MongoDB. Os resultados foram comparados verificando o número total de ocorrências para a busca em cada ferramenta e a igualdade das

sentenças retornadas. Foram testadas seis funções morfossintáticas, fazendo uso também de operadores lógicos, o que resultou em vinte e quatro consultas. Nos vinte e quatro testes, as sentenças retornadas pelas consultas feitas no WebSinc foram iguais às consultas retornadas nas consultas feitas no MongoDB (DAMACENO, 2018).

Um teste de desempenho tem como função principal apresentar a capacidade de resposta, o rendimento, a confiabilidade, e/ou a escalabilidade de um sistema sob uma determinada carga de trabalho (MICROSOFT, 2007). Os testes de desempenho desenvolvidos neste trabalho foram voltados para a obtenção de tempos de resposta e níveis de utilização dos recursos que cumpram os objetivos de desempenho das consultas. Os testes foram realizados para obtenção do tempo de resposta (em milissegundos) de buscas no arquivo com anotação morfossintática armazenado no MongoDB, a partir de uma aplicação Java. Os tempos de resposta obtidos foram comparados com o retorno de execução das mesmas buscas feitas por Costa (2015) sobre o arquivo XML com a linguagem XQuery (DAMACENO, 2018).

Com o intuito de prezar ao máximo pela correta extração dos dados, todos os testes foram realizados sob o mesmo ambiente (mesma arquitetura e dispositivo computacionais). O gráfico da figura 7 demonstra o tempo médio que foi demandado para realização de cada busca morfossintática no MongoDB, contrastando-o ao tempo médio obtido para as mesmas consultas no arquivo XML feitas por Costa (2015). Detalhes técnicos computacionais a respeito dos testes podem ser consultados em Damaceno (2018).

Com os dados do gráfico, pode-se observar um maior desempenho do SGBD MongoDB para todas as buscas morfossintáticas, o que confirma a característica de alta performance em consultas citadas por Lóscio et al. (2011). Segundo Rosa (2009), o ganho de desempenho elevado nas buscas no MongoDB comparado com a leitura dos dados em discos ocorre pelo fato do banco manter os dados mais recentes dispostos para consultas diretamente na memória RAM (*Random Access Memory*). E conforme Abinader (2006), o *parser* XML pode levar um tempo de processamento maior para ser carregado dentro da memória, o que pode torná-lo mais lento na manipulação do documento. Além do ganho na performance, o formato de codificação JSON para armazenamento no MongoDB produziu um documento com arquivo de tamanho menor do que o arquivo XML.



	Existência	Precedência	Precedência Imediata	Palavra no início da sentença	Palavra no fim da sentença	Palavra na n-ésima posição da sentença
MongoDB	5,3	3,3	3,6	5,3	4,3	7,6
Arquivo XML	605	587,6	582,6	730,6	598	712,6

Figura 7 – Resultados dos testes de desempenhos das buscas morfossintáticas
Fonte: Damaceno (2018)

Apesar do melhor desempenho apresentado pela proposta no formato JSON, outras características além da performance devem ser consideradas na decisão para escolha do formato de anotação. A representação XML usa uma marcação amplamente aceita para anotação de corpus, trazendo vantagens de interoperabilidade e legibilidade por máquinas. É importante considerar o uso de padrões bem documentados e aceitos, que facilitam o intercâmbio de dados e sua conversão para uso com novas ferramentas de anotação. Um projeto de corpus pode criar um modelo próprio de anotação e desenvolver ferramentas específicas para o formato criado no âmbito do mesmo projeto. No entanto, uma representação altamente idiossincrática torna menos provável que outros grupos de pesquisadores possam usar esse formato ou estender esse corpus (ZELDER, 2019).

Adotar um formato em conformidade com padrões é crucial também para a escalabilidade do corpus. Formatos padrão trabalham mais facilmente com ferramentas automáticas de PLN (Processamento de Linguagem Natural), como *parsers*, etiquetadores e outros. Representações idiossincráticas reduzem o potencial de compatibilidade com tais ferramentas, dificultando ainda mais a anotação tanto para humanos quanto para máquinas (ZELDER, 2019). Tanto JSON quanto XML são padrões para intercâmbio de dados, mas apenas o último tem sido empregado em padrões de anotação. Assim, uma gama de

ferramentas já implementadas e disponíveis para o padrão XML se configuram como uma grande vantagem para esse formato. Como contraponto, é importante destacar que existe uma potencialidade de utilização futura de JSON para anotação de corpora, uma vez que já é um padrão aceito para intercâmbio de dados pelo W3C, ao contrário de uma representação altamente idiossincrática que utilize um formato livre.

Considerando a persistência da proposta, o armazenamento em um banco de dados pode refletir negativamente na interoperabilidade, uma vez que requer uma ferramenta específica de interface com o usuário para recuperação da informação. Arquivos JSON ou XML são arquivos de texto simples, e ambos podem ser lidos facilmente por seres humanos. Sendo assim, a realização da persistência da anotação em arquivos de texto ao invés de banco de dados favorece uma maior disponibilidade da anotação para pesquisadores interessados no corpus, sem requerimento de softwares adicionais. A persistência em arquivos também reflete mais adequadamente as estruturas do texto original, possibilitando a leitura das informações com menor custo computacional, e em casos de falha dos elementos de software utilizados nessa recuperação, possibilita ainda a leitura do texto por humanos a partir do arquivo anotado.

Para as buscas linguísticas, é importante considerar a existência de ferramentas computacionais existentes para reuso. Em buscas como Precedência de termos, foi necessário utilizar uma chave adicional (chave "*path*", figura 4) e aplicar o recurso de expressões regulares para as consultas. Essa flexibilidade dos bancos NoSQL é uma de suas principais vantagens. No entanto, a inserção da chave adicional configura-se como uma desvantagem, uma vez que consiste em informação redundante, com fins puramente computacionais de viabilização da busca. As funções de buscas por Vizinhança não foram implementadas por ausência de operadores ou funções de propósito específico no *framework* de consulta utilizado. Em contrapartida, todas as buscas realizadas nos arquivos anotados em XML podem ser feitas na própria sintaxe da linguagem de consulta (XQuery ou XPath), sem necessidade de informações ou recursos adicionais, o que configura um ponto positivo em favor da anotação já utilizada. Soma-se a isso a existência de funções para a estrutura hierárquica arbórea na linguagem XQuery, que viabilizou as buscas em estruturas sintáticas realizadas por Costa (2015), não sendo possível contemplá-las neste trabalho.

9- Considerações finais

Objetivou-se verificar a viabilidade de uma representação do corpus DOViC para armazenamento em banco de dados NoSQL e comparar a performance das buscas na anotação atual com a anotação proposta, fazendo uma análise de aspectos como interoperabilidade e persistência desta última. Com base nos resultados obtidos para buscas morfossintáticas nesta pesquisa, utilizando um documento do corpus DOViC, entendemos que uma representação morfossintática e de edições de corpora anotados nos moldes do CTB em formato JSON e armazenamento no banco MongoDB, aproveitando as vantagens da tecnologia NoSQL, é viável. Para buscas como Precedência de termos, foi necessário utilizar-se da flexibilidade de usar o recurso de expressões regulares com uma chave adicional. Essa flexibilidade dos bancos NoSQL configura-se como uma de suas principais vantagens, mas o uso da chave adicional foi desvantajosa em relação à anotação XML. Acrescenta-se que todas as buscas realizadas nos arquivos anotados em XML podem ser feitas na própria sintaxe da linguagem de consulta (XQuery ou XPath), sem necessidade de informações ou recursos adicionais, o que configura mais um ponto positivo da anotação já utilizada.

Constatou-se ainda que os resultados obtidos para as funções de buscas implementadas são os mesmos resultados do trabalho de Costa (2015), considerando o conteúdo e quantidade das sentenças retornadas. Já com relação ao desempenho das buscas morfossintáticas, como demonstrado, a proposta de anotação JSON se sobressai, obtendo um tempo de resposta de aproximadamente 99% menor do que as buscas em XQuery nos arquivos XML. Nessa perspectiva, os resultados demonstram uma vantagem de se utilizar a tecnologia NoSQL com formato JSON nos estudos da Linguística de Corpus, por obter melhor desempenho computacional na extração de informações do corpus. No entanto, buscas por Vizinhança não foram possíveis na proposta aqui apresentada.

Por fim, JSON apresenta maior performance do que XML, mas ainda não é utilizado como padrão para anotação de corpora. Considerando aspectos além da performance, a anotação XML apresenta mais pontos positivos. Por ser utilizada em padrões de anotação, traz a vantagem de maior interoperabilidade. Usar uma anotação JSON pode reduzir o potencial de compatibilidade com ferramentas já implementadas. Na tomada de decisões para escolha do formato de anotação, o formato JSON pode ser considerado quando o desempenho e a utilização de recursos computacionais forem cruciais. No entanto, problemas relacionados à

performance não foram apresentados no corpus DOViC ou outro baseado no corpus Tycho Brahe até então.

10- Referências

ABINADER, Jorge Abílio. **Web services em Java**. Brasport: Rio de Janeiro, 2006.

BRITO, Ricardo W. **Bancos de dados NoSQL x SGBDs relacionais**: análise comparativa. Faculdade Farias Brito e Universidade de Fortaleza, 2010. Disponível em: <http://shorturl.at/dGU27 >. Acesso em 17 fev. 2019.

COSTA, Aline Silva. **WebSinC:** Uma Ferramenta Web para buscas sintáticas e morfossintáticas em *corpora* anotados - Estudo de Caso do Corpus DOViC- Bahia. Dissertação (Programa de Pós-graduação em Linguística). Universidade Estadual do Sudoeste da Bahia (UESB), Vitória da Conquista, 2015. Orientadora: Cristiane Namiuti; Coorientador: Jorge Viana Santos.

DAMACENO, Romenito Pereira. **REPRESENTAÇÃO DE UM CORPUS LINGUÍSTICO EM UM BANCO DE DADOS NoSQL: Estudo de caso do corpus DOViC**. Monografia (Curso de Bacharelado em Sistemas de Informação). Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA). Vitória da Conquista, 2018. Orientadora: Aline Silva Costa.

DATE, C.J. **Uma Introdução a Sistemas de Banco de Dados**. São Paulo: Editora Edgard Blücher, 2004.

DEITEL, H.M.; DEITEL, P.J.; NIETO, T.M.; LIN, T.M.; SHADU, P.V. **XML:** Como programar. Porto Alegre: Bookman, 2005.

DEITEL, H.M; DEITEL, P.J. **Java:** como programar. 6.ed. São Paulo: Pearson Prentice Hall, 2005.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Addison Wesley, 2011.

FONSECA, Rúben; SIMOES, Alberto. **Alternativas ao XML:** YAML e JSON. 2007. Disponível em: <https://goo.gl/9aJgm7>. Acesso em: 12 maio 2017.

GALVES, Charlotte; ANDRADE, Aroldo Leal de.; FARIA, Pablo. **Tycho brahe parsed corpus of historical portuguese**. 2017. Disponível em: <http://www.tycho.iel.unicamp.br/~tycho/corpus/en/index.html>. Acesso em: 13 mar. 2019.

GOMES. Bruna C. Kalles.; BASSO, Carla de Almeida M. Desempenho de Banco de Dados Não Relacionais com Big Data**. 12th International Conference on Information Systems & Technology Management - Contecsi**. 2015. Disponível em: <

http://www.contecsi.tecsi.org/index.php/contecsi/12CONTECSI/paper/view/3040/2348>. Acesso em: 02 set. 2017.

GONÇALVES, Eduardo Corrêa. Introdução ao Formato JSON**. Devmedia.** 2012. Disponível em:<https://www.devmedia.com.br/json-tutorial/25275>. Acesso em 06 jul. 2017.

HIRSCHMAN, Lynette.; MANI, Inderjeet. Evaluation. In: MIKTOV, R. (Editor). **The Oxford Handbook of Computational Linguistics**. New York: Oxford University Press, 2003.

LÓSCIO, Bernadette Farias.; OLIVEIRA, Hélio Rodrigues de; PONTES, Jonas César de S. NoSQL no desenvolvimento de aplicações Web colaborativas. **Anais do VIII Simpósio Brasileiro de Sistemas Colaborativos**, v. 10, p. 11, 2011. Disponível em: <https://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf>. Acesso em 20 mar. 2019.

MARTINS FILHO, Marcos André P. **SQL X NOSQL**: **Análise de desempenho do uso do MongoDB em relação ao uso do PostgreSQL**. Trabalho de Graduação (Graduação em Ciência da Computação). Universidade Federal de Pernambuco. Recife, 2015. Orientador: Fernando da Fonseca de Souza. Disponível em:< https://www.cin.ufpe.br/~tg/2014-2/mapmf.pdf>. Acesso em 20 mar. 2019.

MEGERDOOMIAN, Karine. Text mining, Corpus building, and testing. In: FARGHALY, Ali Ahmed Sabry (Ed.). **Handbook for language engineers**. Standford: CSLI, 2003. p.14.

MELLO, Heliana Ribeiro de; SOUZA, Renato Rocha. A linguagem da ciência: Prospecção de dados baseados em corpora**. Anais – Seminários Teóricos Interdisciplinares do SEMIOTEC – I STIS**. UFMG. 2012. Disponível em: <http://www.periodicos.letras.ufmg.br/index.php/stis/issue/view/177>. Acesso em 13 mai. 2017.

MICROSOFT. **Performance Testing Guidance for Web Applications:** Microsoft Developer Network. 2007. Disponível em: <https://msdn.microsoft.com/en-us/library/bb924375.aspx>. Acesso em: 18 mar. 2018.

MONGODB. **The MongoDB 3.6 Manual**. 2008. Disponível em: <https://docs.mongodb.com/manual>. Acesso em: 13 de jun. 2017.

MONGODB. **What is MongoDB?.** 2019. Disponível em: <https://www.mongodb.com/what-is-mongodb>. Acesso em 16 jun. 2019.

NAMIUTI-TEMPONI, Cristiane; SANTOS, Jorge Viana (coords.). **Memória conquistense: implementação de um corpus digital** (CNPq N485098/2013-0). Vitória da Conquista: UESB, 2013.

PAIXÃO DE SOUSA, Maria Clara. **O Corpus Tycho Brahe: contribuições para as humanidades digitais no Brasil**. São Paulo, v. 16, n. esp., p. 53-93, dez. 2014. Disponível em: <http://dx.doi.org/10.11606/issn.2176-9419.v16ispep53-93>. Acesso em 16 ago. 2017.

ROSA, Adriano Guzzo. **Análise da Estrutura do Banco de Dados Mongodb:** Testes de Desempenho Mongodb X Mysql. Clube de Autores, 2009.

SANTOS, Jorge Viana; NAMIUTI, Cristiane. **DOViC - Documentos Oitocentistas de Vitória da Conquista. Memória Conquistense**. Vitória da Conquista: UESB/LAPELINC, 2016. Disponível em: <http://memoriaconquistense.uesb.br/websinc>. Acesso em 19 nov 2017.

SANTOS, Jorge Viana; NAMIUTI, Cristiane (coord.). **Corpora digitais para a história do português brasileiro - documentos históricos da região sudoeste da Bahia: aliança PHPB-Tycho Brahe** (FAPESB PET0034/2010). Vitória da Conquista: UESB, 2010.

SANTOS, Jorge Viana; NAMIUTI, Cristiane (coord.). **Corpora digitais de documentos históricos da imperial Vila da Victoria, atual Vitória da Conquista-Bahia: resgate e preservação do patrimônio linguístico e da memória da escravidão na Bahia** (FAPESB APP0014/2016). Vitória da Conquista: UESB, 2016.

SILVA FILHO, Antônio Mendes da. **Programando com XML**. Rio de Janeiro: Elsevier, 2004.

SILBERSCHATZ, Abraham.; KORTH, Henry. F.; SUDARSHAN, S. **Sistemas de Banco de Dados.** 3. ed. São Paulo: Makron Books, 1999.

SOARES, Jhonathan. **O que é MongoDB e porque usá-lo?** 2016. Disponível em: <https://codigosimples.net/2016/03/01/o-que-e-mongodb-e-porque-usa-lo/>. Acesso em 16 jul. 2017.

W3C. **XML Technology**. 2010. Disponível em: < http://www.w3.org/standards/xml/> Acesso em: 20 jul. 2017.

W3C.**JavaScript**. 2011. Disponível em:< https://www.w3.org/wiki/Javascript>. Acesso em: 28 ago. 2017.

W3SCHOOLS. **JSON VS XML**. 2017. Disponível em: < https://www.w3schools.com/js/js_json_xml.asp>. Acesso em: 17 nov. 2017.

Sobre os autores

Aline Silva Costa é professora efetiva do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), mestre em Linguística (2015) pela Universidade Estadual do Sudoeste da Bahia (UESB), atualmente é doutoranda do Programa de Pós-graduação em Linguística da mesma instituição, sob a orientação da professora Dra Cristiane Namiuti. Possui graduação em Ciência da Computação pela UESB (2004), especialização em Administração de Sistemas de Informação pela Universidade Federal de Lavras (2005). Tem experiência na área de Ciência da Computação, com ênfase em Linguagem de Programação e Engenharia de Software.

Bruno Silvério Costa é professor efetivo do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), mestre em Desenvolvimento Regional e Meio Ambiente pela Universidade Estadual Santa Cruz - UESC (2015), atualmente é doutorando do Programa de Pós-graduação em Linguística da Universidade Estadual do Sudoeste da Bahia (UESB), sob a orientação do professor Dr. Jorge Viana Santos. Possui graduação em Ciência da Computação pela UESB (2004), especialização em Administração de Sistemas de Informação pela Universidade Federal de Lavras (2005). Tem experiência na área de Ciência da Computação, com ênfase em Redes de Computadores, Programação e Sistemas de Alto Desempenho.

Romenito Pereira Damaceno é Bacharel em Sistemas de Informação pelo Instituto Federal da Bahia – IFBA (2018). Atualmente é Analista de Suporte da Linet Serviços de Comunicação LTDA. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação.

Cristiane Namiuti é professora titular da Universidade Estadual do Sudoeste da Bahia (UESB), atuando no quadro permanente do Programa de Pós-graduação em Linguística (PPGLin). Possui doutorado em Linguística pela Universidade Estadual de Campinas (UNICAMP). Tem experiência na área de Linguística, com ênfase em Linguística Histórica e metodologias automáticas de busca de dados em textos escritos, atuando principalmente, nos seguintes temas: interpolação, clítico, mudança linguística, história do português e linguística de corpus. Possui Bacharelado em Linguística pela UNICAMP (2001), Doutorado (2008) e Pós-Doutorado (2010), em Lingüística, pela mesma instituição.

Jorge Viana Santos é professor titular da Universidade Estadual do Sudoeste da Bahia (UESB), atuando no quadro permanente do Programa de Pós-graduação em Linguística (PPGLin) e docente colaborador do Programa de Pós-Graduação em Memória: Linguagem e Sociedade (PPGMLS/UESB). Possui doutorado em Linguística pela Universidade Estadual de Campinas (UNICAMP), e mestrado em Comunicação e Semiótica pela Pontifícia Universidade Católica de São Paulo. Tem experiência na área de Linguística e Semiótica, atuando nos seguintes temas: sentido, argumentação, lugares de enunciação, processos de designação, reescritura, subjetivação, textos, Linguística de Corpus, fotografia, imagem e memória.

Notas

[i] Corpus, aqui, é tomado como uma coleção de dados linguísticos que podem ser analisados e estudados, sejam em formato de texto escrito ou transcrição de fala, dispostos de tal modo que possam ser processados por computador.

[ii] A Linguística Computacional é a área da ciência linguística que cuida de investigar o tratamento computacional da linguagem e das línguas naturais para diversos fins práticos. De acordo com Vieira e Lima (2001, p. 1), "é a área de conhecimento que explora as relações entre linguística e informática, tornando possível a construção de sistemas com capacidade de reconhecer e produzir informação apresentada em linguagem natural".

[iii] Os resultados aqui apresentados foram obtidos no âmbito dos projetos temáticos dos quais os autores participam: Fapesb APP0007/2016, Fapesb APP0014/2016, CNPq 436209/2018-7.

[iv] Java é uma linguagem de programação gratuita e bastante utilizada no desenvolvimento de aplicativos baseados na Internet (DEITEL; DEITEL, 2005).

[v]O formato de anotação NITE XML faz parte do projeto NITE (*Natural Interactivity Tools Engineering*), financiado pela UE em 2001-2003 (ZELDER, 2019).

[vi]http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html.

[vii]http://www.tei-c.org/index.xml

[viii]http://www.tc37sc4.org/

[ix]A *XPath*, recomendada pelo W3C (*World Wide Web Consortium*), é uma linguagem para navegação em partes de um documento XML (W3C, 2010; DEITEL et al., 2005).

[x] A XQuery é uma linguagem de expressão funcional que é usada para consultar ou processar dados XML e é um padrão do W3C (W3C, 2010).

[xi] Linguagem de programação usada em páginas Web, cujas instruções são compostas por valores, operadores, expressões, palavras-chave e comentários (W3C, 2011; W3SCHOOLS, 2017).

[xii] *Array* ou vetor é uma estrutura de dados linear que armazena uma coleção de elementos, os quais podem ser recuperados por meio de um índice ou uma chave (DEITEL; DEITEL, 2005).

[xiii] Para Elmasri e Navathe (2011), um banco de dados é uma coleção de dados relacionados, os quais representam fatos registrados. De acordo com Date (2004, p. 6), um sistema de banco de dados é "um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar". Em resumo, um banco de dados mantém as informações armazenadas e as disponibiliza sob demanda.