Brazilian Journal of Development

Requisitos para softwares de tesauro

Requirements for tesauro software

Recebimento dos originais: 25/12/2018
Aceitação para publicação: 23/01/2019

Linair Maria Campos
Doutora em Ciência da Informaçãopelo Instituto Brasileiro em Informação Científica e Tecnológica - IBICT/UFF
Instituição: Universidade Federal Fluminense/PPGCI
Endereço: Rua Tiradentes 148, Ingá - Niterói - RJ - Brasil
E-mail: lmcampos@id.uff.br

Maria Luiza de Almeida Campos
Doutora em Ciência da Informação pelo Instituto Brasileiro em Informação Científica e Tecnológica - IBICT/UFRJ
Instituição: Universidade Federal Fluminense/PPGCI
Endereço: Rua Tiradentes 148, Ingá - Niterói - RJ - Brasil
E-mail: maria.almeida@pq.cnpq.br

Jóice Cleide Cardoso Ennes de Souza
Doutora em Ciência da Informação pelo Instituto Brasileiro em Informação Científica e Tecnológica - IBICT/UFRJ
Instituição: Universidade Federal Fluminense/PPGCI
Endereço: Rua Tiradentes 148, Ingá - Niterói - RJ - Brasil
E-mail: joicecardoso@id.uff.br

RESUMO

Softwares para elaboração de tesauros são ferramentas indispensáveis para a construção e gestão de tais vocabulários, de modo a promover o controle terminológico e gestão efetiva de seu aperfeiçoamento. Nesse cenário, a escolha da ferramenta adequada, levando em conta as necessidades peculiares ao seu uso nas corporações, é tarefa de suma importância. Entretanto, as ferramentas existentes no mercado diferem bastante em suas características e existe uma ampla variedade de opções, sendo por isso premente a adoção de uma abordagem metodológica para que a escolha seja mais precisa e menos sujeita a julgamentos subjetivos. O objetivo do presente trabalho é propor uma abordagem visando orientar a escolha de um determinado software de tesauro, voltado para a finalidade e a complexidade de um tesauro específico a ser construído. A abordagem é baseada metodologicamente na análise de requisitos básicos para softwares de tesauro, tendo em vista a análise de requisitos como proposta na área da Engenharia de Software. Os requisitos são obtidos a partir de levantamento na literatura e da análise de um conjunto de ferramentas atuais. Como resultado, apresenta-se um quadro com um conjunto básico de requisitos para softwares de tesauro, agrupados de acordo com categoriaspré-definidas. Como conclusão, observa-se que a adoção do quadro permite minimizar a subjetividade na escolha de um software de tesauro, permitindo uma tomada de decisão de forma mais precisa.

Palavras-chave: Softwares de tesauros, Avaliação de software, Requisitos de software.

ABSTRACT

Brazilian Journal of Development

Thesaurus software are indispensable tools for the construction and management of such vocabularies, in order to promote terminological control and effective management of their improvement. In this scenario, choosing the right tool, taking into account the needs peculiar to its use in corporations, is a task of paramount importance. However, the existing tools in the market differ greatly in their characteristics and there is a wide variety of options, so it is urgent to adopt a methodological approach so that the choice is more precise and less prone to subjective judgments. The purpose of the present work is to propose an approach aimed at guiding the choice of a particular thesaurus software, focused on the objective and complexity of a specific thesaurus to be constructed. The approach is methodologically based on the analysis of basic requirements for thesaurus software, according to requirements definition in the area of Software Engineering. The requirements are obtained from a survey in the literature and from the analysis of a set of current tools. As a result, a framework with a basic set of requirements for thesaurus software, grouped according to pre-defined categories is presented. As a conclusion, it is observed that the adoption of the framework allows to minimize the subjectivity in the choice of a thesaurus software, allowing a more precise decision making.

Keywords: Thesaurus software, Software evaluation, Software requirements.

1 INTRODUÇÃO

Os tesauros têm tido ampla utilização no mercado, como instrumento para indexação e recuperação de informação em meio digital. Por se constituir em um instrumento de controle terminológico, o uso do tesauro pode contribuir para a precisão da recuperação da informação, minimizando ambiguidades, respeitandoo uso da linguagemvariada ao tratar o uso de sinonímias (Gomes & Campos, 2004), apontando possibilidades alternativas para expansão de pesquisas, atravésde termos associados. Além disso, o uso do tesauro apoia no entendimento do conceito a ser utilizado, na indexação e na recuperação, ao permitir o usode notas de indexação e de definição. Nesse contexto, é importante destacar que para cada função do tesauro existe uma necessidade que ele pode atender. Assim, o indexador utiliza o tesauro como um instrumento para a tradução e adequação da temática do documento à necessidade de um usuário que está vinculado a um contexto específico. O usuário, por sua vez, quando realiza uma pesquisa, pode utilizar o tesauro como um instrumento que evidencia as temáticas existentes no domínio de dados a serem recuperados. O tesauro que possui uma parte sistemática funciona como uma taxonomia, explicitando o conhecimento existente naquele domínio e, se este estiver em meio digital, espelha também, por conseguinte, os documentos existentes. Além disso, ao consultar a sua parte alfabética o usuário terá informações sobre conceitos associados que poderão melhorar a sua estratégia de busca.

Neste sentido, consideramos fundamental que tesauros apresentem uma parte sistemática e uma alfabética, o que configura tesauros de vertente européia. Os tesauros produzidos nesta vertente fornecem bases para a ordenação das classes e preconizam a apresentação sistemática do tesauro, além da tradicional parte alfabética (Campos, 2001).

Brazilian Journal of Development

A vertente europeia é influenciada pela Teoria da Classificação Facetada de Ranganathan (abordagem sistemática) e a norte-americana, com raízes no Unitermo (abordagem alfabética) (Lancaster, 1986; Campos, 2001). A vertente que surge na Europa, segundo Campos (2001, p. 97), "oferece princípios para melhor posicionar o conceito no sistema de conceitos".

No escopo desse trabalho, nosso foco é o da gestão terminológica, ou seja, está voltada para o profissional que elabora ou necessita escolher um software de tesauro no mercado. Destacamos ainda que apesar do foco estar na gestão de tesauros, nos reportaremos, quando for o caso, sempre aos aspectos que venham atender tanto a necessidade do indexador, quanto a do usuário que busca a informação, como, por exemplo, a questão do uso de notas explicativas e de indexação.

A questão da gestão terminológica envolve um perfil de profissional da informaçãovoltado para a criação de termos, sua associação com outros termos, definição de sinônimos e notas de explicação voltadas para os usuários que, efetivamente, usarão o tesauro. Para esse profissional, a elaboração de um tesauro é uma tarefa de natureza intelectual, que abrange a identificação e delimitação de um corpus terminológico e a compreensão da natureza dos conceitos envolvidos. Para isso, é importante visualizar de que forma os conceitos se organizam dentro de hierarquias e de que forma se relacionam uns com os outros. Esse profissional precisa ter apoio para manipular o tesauro de forma a poder facilmente entender e alterar sua estrutura e controlar as mudanças realizadas.Dessa forma, considerando que a construção de um tesauro não é uma tarefa trivial, o uso de uma ferramenta de software é importante para apoiar não só as questões da construção em si, mas também da metodologia usada para a sua elaboração, o seu pensar, em especial nos tesauros que adotam o conceito como base de formação terminológica(Camposet al., 2006).

Softwares para elaboração de tesauros são ferramentas indispensáveis para a construção e gestão de tais vocabulários, de modo a promover o controle terminológico e sua gestão. Nesse cenário, a escolha da ferramenta adequada, levando em conta as necessidades peculiares ao seu uso nas corporações, é tarefa de suma importância. Entretanto, as ferramentas existentes no mercado diferem bastante em suas características, que por si são variadas,e há uma grande diversidade de softwares, o que pode levar a decisões equivocadas, se estas forem tomadas com base em uma análise subjetiva dos softwares. Como decidir de forma mais precisa? Como mensurar se um software é mais adequado que outro semelhante? Quais aspectos são os mais relevantes para uma determinada empresa, com uma missão específica para o uso do tesauro?

Defendemos a ideia que para responder a essas perguntas é premente a adoção de uma abordagem metodológica para que a escolha do software de tesauro seja mais precisa e menos sujeita a julgamentos subjetivos. Ao se efetuar uma revisão de literatura voltada para a questão que se coloca, percebe-se que há pouca discussão sobre a temática específica (Camposet al., 2006),

Ganzmann (1990), e, além disso, boaparte são trabalhos com mais de dez anos, merecendo o tema uma revisão atualizada, em especial levando-se em conta a forma pela qual os softwares atuais implementam a automação do tesauro. Naturalmente, é possível derivar um conjunto de características desejáveis para um software de tesauro a partir do entendimento de suas características e perfis de uso. No sentido de possibilitar a identificação de elementos necessários para a constituição de um tesauro, destacam-seGanzmann (1990),Soergel (2002), Campos (2001), InternationalFederationof Library AssociationsandInstitutions (2009), Campos et al. (2006), Gomes (1984) eGomes, Campos e Motta (2004). Além destes estudos, nos apoiamos nos princípios determinados na Norma ISO 25964-1 – Information and Documentation – Thesauri and interoperability with other vocabularies (2011).

Conforme as diretrizes propostas por Gomes (1984) para elaboração de tesauros monolíngues, observamos aspectos voltados especificamente para a gestão terminológica tais como: (i) controle de exclusão de termos; (ii) uso de termo indicador de faceta, que é usado para indicar a base lógica adotada na subdivisão de uma classe; (iii) a adoção de informações complementares, que são úteis para entender o contexto de criação do termo, tais como a data de criação, a fonte e notas históricas; (iv) mecanismos de validação para evitar termos duplicados e relações recíprocas faltantes, em especial quando da deleção de termos. Observamos ainda, de acordo com Gomes (1984),aspectos voltados para a apresentação e uso do tesauro, como: (i) relatórios alfabético e sistemático; (ii) representação gráfica que permita explicitar as facetas do tesauro e suas diversas relações; e, ainda, os aspectos ligados ao uso de relações que devem estar presentes no tesauro, como as relações partitivas, de equivalência e de sinonímia.

De maneira complementar, é possível levantar um conjunto de características desejáveis a partir da análise de características existentes em softwares comercializados no mercado, dos quais, se destacam pela sua maturidade, até onde pudemos perceber, os seguintes: Synaptica (2017), Multites (2017) e Thesaurus Builder (2017).

A avaliação dos softwares de tesauro permite-nos observar determinadas funcionalidades específicas para a gestão terminológica que a tecnologia pode facilitar, como, por exemplo: (i) workflow para aprovação de termos;(ii) uso de modelos ou *templates* para apoiar a formação de nomes; (iv) registro de mudanças com opção de reversão (trilhas de auditoria); (v) criação/deleção automática de relações recíprocas.

Além disso, há que se considerarem aspectos que não dizem respeito a um tesauro em si, mas à forma como o software é implementado, seu custo, confiabilidade, dentre outros. Para isso, é importante recorrer à área da Engenharia de Software (Sommerville, 2007), onde são detalhados os tipos de requisitos de software e sua categorização.

Brazilian Journal of Development

Esses aspectos ligados à implementação do software também podem ser observados nas ferramentas analisadas, como, por exemplo: (i) compatibilidade com padrão de codificação Unicode / UTF; (ii) suporte a múltiplos acessos concorrentes sem prejuízo de desempenho; (iii) configuração de software (em especial banco de dados) compatível com o que a empresa adota / permite; (iv) controle de acesso por usuário; (v) integração com outros softwares; (vi) menus sensitivos ao contexto para acesso rápido a funções comumente usadas (ex: editar, deletar, adicionar relação, etc.); (vii) interface web; (viii) funcionamento em diversos navegadores (browsers); (ix) exportação de dados em formatos diversos; (x) termos relacionados ligados por hiperlinks para facilitar a navegação; (xi) software roda em Unix e Windows.

De modo a orientar a escolha de um determinado software de tesauro voltado para a finalidade e complexidade de um tesauro específico a ser construído, propomos partir de um quadro de referência, contendo um conjunto de requisitos para software de tesauro levantados na literatura, e tendo como princípio, a análise de características de softwares de tesauro existentes, sem perder de vista os diferentes perfis de uso de um tesauro e as suas características próprias, enquanto instrumento terminológico. Espera-se assim contribuir para minimizar o problema da subjetividade na escolha de softwares de tesauro.

2 REQUISITOS DE SOFTWARE E SUA CARACTERIZAÇÃO

Requisitos são características que um software possui, de que forma ele se comporta para atender necessidades de seus usuários, podendo ser divididos em funcionais e não funcionais.

Os requisitos funcionais dizem respeito às funções que o sistema deve realizar e como o sistema deve se comportar em determinadas situações (Falbo, 2005). Geralmente estão ligados às funcionalidades específicas que um usuário realiza em seu dia a dia, no que tange à finalidade do software. Por exemplo, se a finalidade do software é criar tesauros, então uma atividade típica seria permitir criar um termo, descrever suas associações, informar uma nota de escopo, e assim por diante.

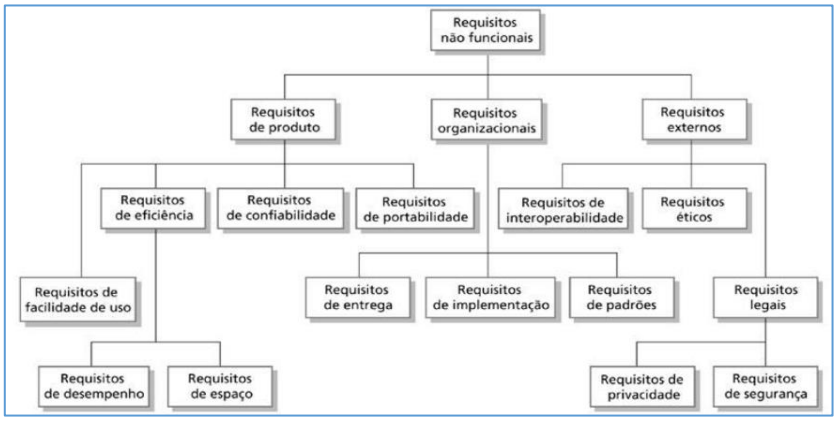
Como os requisitos funcionais são muito variados, as categorias utilizadas para agrupá-los foram definidas a partir de aspectos comuns observados nesses requisitos, e são as seguintes: apoio metodológico - aspectos que ajudam o usuário a elaborar o tesauro do ponto de vista de sua concepção intelectual, como, por exemplo, visualização do tesauro em forma gráfica, uso de templates para formação de nomes; atributos do termo - metadados associados ao termo, como, por exemplo, notas de escopo, notas definidas pelo usuário; gestão terminológica - aspectos que permitem controlar a criação, alteração e exclusão de termos, como, por exemplo, workflow para aprovação de termos, existência de trilhas de auditoria; relatórios - funcionalidades específicas para

Brazilian Journal of Development

a impressão do tesauro, como, por exemplo, relatórios sistemáticos, relatórios definidos pelo usuário; tratamento de dados - funcionalidades ligadas aos dados contidos no tesauro como um todo, sua entrada e saída, como, por exemplo, exportação e importação de dados; tratamento de relações - funcionalidades voltadas especificamente para o uso de relações no tesauro, como, por exemplo, se é possível o uso de relações criadas pelo usuário, se permite relações partitivas; validação / consistência - funcionalidades ligadas à integridade dos dados e sua verificação, como, por exemplo, impedir que termos duplicados sejam inseridos no tesauro.

Já os requisitos não funcionais descrevem restrições sobre asfuncionalidades do software, tais como restrições deespaço, de número de usuários simultâneos, etc. Geralmente, não dizem respeito à atividade específica do usuário que o software está automatizando, mas sim,de modo geral, ao software como um todo. Por exemplo, não faz parte da atividade específica de construção de um tesauro a preocupação com o espaço ocupado em disco ou o número de usuários simultâneos. Isso diz respeito à forma pela qual o software foi concebido, sem levar em conta a tarefa específica da construção de um tesauro, se tomarmos esse tipo de software como exemplo. Embora não haja um consenso na literatura sobre como categorizar requisitos não funcionais, é comum referência à Sommerville (2007), conceituado autor da área de Engenharia de Software, que propõe as categorias exibidas na Figura 1, que divide os requisitos não funcionais em três grandes grupos, a saber: requisitos de produto, requisitos organizacionais e requisitos externos.Os requisitos de produto estão ligados ao comportamento do software, como, por exemplo, a rapidez de execução, o espaço de memória requerido, dentre outros. Os requisitos organizacionais, por sua vez, estão ligados às políticas e procedimentos que devem ser observados nas organizações do cliente e do desenvolvedor, no que tange ao desenvolvimento do software. Como exemplos, podemos citar a escolha da linguagem de programação, métodos de desenvolvimento adotados, dentre outros. Por fim, os requisitos externos dizem respeito a fatores externos ao software e ao seu desenvolvimento, como os aspectos legais e éticos que o software deve estar em conformidade, dentre outros.

Figura 1. Categorias de requisitos não funcionais de Sommerville



Fonte: Sommerville (2007)

Brazilian Journal of Development

Além da identificação e categorização dos requisitos, é importante ainda a sua priorização, uma vez que determinados requisitos são mais importantes que outros. Por exemplo, para um software de tesauro, o controle de sinonímia é requisito de fundamental importância. Por outro lado, o software ter ajuda online pode ser útil, mas não é indispensável. Alguns autores como Sommerville (2007) propõem que se adote uma priorização em três níveis: requisitos essenciais (o software tem de possuir, do contrário não é adequado); úteis (o software teria sua eficácia reduzida se não estivessem contemplados, mas a sua ausência não é um fator que impediria o uso do software); e desejáveis (sua presença tornaria o software mais atrativo, mas sua ausência é de importância menor). Após a definição da prioridade, pode-se associar um valor numérico a cada prioridade, de modo a poder efetuar uma aferição numérica objetiva para um determinado conjunto de requisitos. Existem várias técnicas para determinar esse valor numérico (Berander& Andrews, 2005; Achimugu, Selamat, Ibrahim, &Mahrin, 2014), estando fora do escopo do presente trabalho detalhá-las.

3 METODOLOGIA

A abordagem é baseada metodologicamente na análise de requisitos básicos para softwares de tesauro, tendo em vista a análise de requisitos como proposta na área da Engenharia de Software, uma vez que o estudo de requisitos de software é originário dessa área. Dessa forma, os requisitos são agrupados em requistos funcionais e não funcionais. Os requisitos não funcionais são agrupados de acordo com as categorias propostas por Sommerville. Quanto aos requisitos funcionais, devido à sua ampla natureza, a sua categorização foi feita com base nos atributos comuns que foram observados a partir dos diversos requisitos funcionais para tesauro obtidos.

Os requisitos são obtidos a partir de levantamento na literatura sobre as características de um tesauroe da análise de um conjunto de ferramentas atuais.

Deste modo, procura-se confrontar o que é necessário para que um tesauro possa ser automatizado, ao mesmo tempo em que se levanta na prática de que forma esses requisitos têm sido contemplados pelos fabricantes de software. Esse procedimento garante que seja feito um contraste entre o estado da arte na implementação dos softwares de tesauro, com as necessidades que são inerentes ao instrumento em si, levando-se em conta os diferentes perfis de uso do tesauro (gestão terminológica, indexação, recuperação de informação), de modo a garantir que sejam atendidos os requisitos mínimos que a automatização de um software de tesauro deve ter.Foram avaliados os softwares: Synaptica, Multites eThesaurusBuilder. A escolha desses softwares pressupôs os seguintes aspectos: tempo de mercado (no mínimo três anos), existência de documentação, existência de suporte para o produto, existência de uma base de clientes apontada pelo fornecedor

Brazilian Journal of Development

do software. Esses pressupostos visam garantir minimamente que o software possua maturidade, ou seja, que possua um grau pequeno de falhas e, além da maturidade, que possuaalguma receptividade no mercado, o que é um indício de sua adequação.

Sugere-se ainda que seja adotada uma priorização dos requisitos e uma associação de valor a essa priorização, de modo que possam ser usados para avaliar um conjunto de softwares de forma menos subjetiva, com base nos mesmos critérios.Essa priorização, entretanto, não é contemplada no presente trabalho, pois depende dos objetivos de uso do tesauro e da percepção de importância dos requisitos por parte do grupo de usuários que decidirá pela adoção da ferramenta. Da mesma forma, alguns requisitos básicos podem ser acrescentados, dependendo da utilização específica do software de tesauro para indexação e recuperação de conteúdo em bases de dados associadas.

4 RESULTADOS

Como resultado, apresenta-se um trecho do quadro com um conjunto básico de requisitos para softwares de tesauro, agrupados de acordo com categorias de Sommerville e as categorias de requisitos funcionais propostas. Sendo que a coluna “Tipo” indica se o requisito é funcional (F) ou não funcional (N). A coluna “Origem” indica em que software o requisito foi encontrado (até onde foi possível avaliar pela documentação).Cabe destacar que em alguns dos requisitos não foi indicado o(s) software(s) específico(s), pois se trata de avaliar situações específicas que envolvem aspectos quantitativos a serem definidos pelos responsáveis da empresa que vai adotar o tesauro. Por exemplo, norequisito “Múltiplos acessos concorrentes sem prejuízo de desempenho”, não se aplica especificar o software, pois vai depender do número mínimo de acessos concorrentes que cada empresa vai definir como sendo razoável em termos de desempenho. Uma determinada empresa pode achar que o software deve proporcionar um desempenho razoável para pelo menos 10 acessos concorrentes, enquanto outra pode achar que o mínimo razoável seja 15 acessos concorrentes.

A lista do Quadro1 não é exaustiva, porém é representativa das necessidades de automação de um tesauro, considerando o que foi exposto na seção de metodologia.

Quadro 1. Requisitos para softwares de tesauro

CATEGORIA	REQUISITO	ORIGEM	TIPO
Apoio metodológico	Usa modelos ou templates para apoiar a formação de nomes	Synaptica	F
	Visualiza o tesauro de forma	Synaptica	F

Brazilian Journal of Development

	gráfica		
	Visualiza termos e suas relações em ordem hierárquica	Synaptica, T.Builder, Multites	F
Atributos do termo	Categoria	Synaptica, T.Builder, Multites	F
	Data da criação	Synaptica, T.Builder, Multites	F
	Data da última atualização	Synaptica, T.Builder, Multites	F
	Nota de escopo	Synaptica, T.Builder, Multites	F
	Nota definida pelo usuário	T.Builder, Multites	F
	Situação (candidato, aprovado, obsoleto)	Synaptica, T.Builder, Multites	F
Gestão terminológica	Permite a gestão terminológica (aprovar termos candidatos, invalidar termos, torná-los obsoletos, etc.) individualmente ou por hierarquias de termos	Synaptica	F
	Registra mudanças com opção de reverter (trilhas de auditoria)	Não foi encontrada informação nesse sentido em nenhum dos softwares analisados (a não ser reverter exclusão de termos)	F
	Workflow para aprovação de termos	Synaptica	F
Relatórios	Relatório alfabético	Synaptica, T.Builder, Multites	F
	Relatório em ordem sistemática e alfabética que indique as relações dos termos	Synaptica, T.Builder, Multites	F
	Relatório Sistemático	Synaptica, T.Builder, Multites	F

	Relatório definido pelo usuário	Synaptica, T.Builder, Multites	F
Tratamento de dados	Exporta dados em algum formato (especificar o formato e quais dados)	Synaptica, T.Builder, Multites	F
	Importa dados (especificar como e quais dados)	Synaptica, T.Builder, Multites	F
	Multilíngue	Synaptica, T.Builder, Multites	F
Tratamento de relações	Diferencia entre relações lógicas (genéricas) e ônticas (partitivas, associativas/funcionais)	T.Builder, Multites	F
	Relações padrão (TG/TE (BT/NT), USE/UP (USE/UF), TA (RT))	Synaptica, Multites	F
	Relações padrão (TGP/TEP)	Multites	F
	Permite relações criadas pelo usuário	T.Builder, Multites	F

Brazilian Journal of Development

Validação/consistência	Cria/deleta automaticamente relações recíprocas	Synaptica, T.Builder, Multites	F
	Impede a ocorrência de termos duplicados	Synaptica, T.Builder, Multites	F
	Alerta ao excluir termo que possua termo subordinado	Synaptica, T.Builder, Multites	F
	Impede ou alerta sobre poli-hierarquias (múltiplos termos genéricos), exceto no interior de uma mesma classe	T.Builder	F
	Impede que um termo seja ao mesmo tempo preferido e não preferido	Synaptica, T.Builder, Multites	F
	Impede referências circulares (ex: termo que é específico dele mesmo)	Synaptica, T.Builder, Multites	F
	Permite apenas relações de sinonímia entre um descritor (termo preferido) e um não descritor	T.Builder, Multites	F
Eficiência	Suporta armazenar volume de dados limitado apenas pelo hardware onde é instalado	Synaptica, T.Builder, Multites	N
	Suporta múltiplos acessos concorrentes sem prejuízo de desempenho (especificar)	No caso não se aplica especificar o software, pois vai depender do que cada empresa vai definir como sendo razoável em termos de desempenho	N
Implementação	Interface web	Synaptica, T.Builder	N
	Multiusuário	Synaptica,	N

Brazilian Journal of Development

		T.Builder, Multites	
	Suporta configuração de software (em especial banco de dados) compatível com o que a empresa adota / permite	No caso não se aplica especificar o software, pois vai depender do que cada empresa adota/permite, não sendo possível avaliar de antemão	N
Interoperabilidade	Integração com outros softwares	Synaptica	N
	Integração com outros softwares (ex: Alfresco, MS Sharepoint, softwares de DAM) de modo a poder usar os termos do tesauro nestes sofwares	Synaptica	N
Segurança /Legais	Controla o acesso por usuário	Synaptica, T.Builder, Multites	N
	Mecanismos para auditoria de segurança	Não foi encontrada informação nesse sentido em nenhum dos softwares analisados	N
	Permissão de acesso baseada em perfis de usuário	Synaptica, T.Builder	N
Usabilidade	Interface em português	Não foi encontrada informação nesse sentido em nenhum dos softwares analisados	N
	Interface gráfica de uso intuitivo	Synaptica, T.Builder, Multites	N

	Menus sensitivos ao contexto para acesso rápido a funções comumente usadas (ex: editar, deletar, adicionar relação, etc.)	Synaptica, T.Builder, Multites	N
	Suporte ao usuário	Synaptica, T.Builder, Multites	N

Brazilian Journal of Development

Usabilidade (continuação)	Termos relacionados são ligados por hiperlinkspara facilitar a navegação	Synaptica, T.Builder	N
	Possui documentação	Synaptica, T.Builder, Multites	N
Uso de padrões	Compatibilidade com Padrão de codificação Unicode / UTF	Synaptica, Multites	N
	Compatibilidade com padrão ISO 2788	Synaptica	N
	Compatibilidade com padrão ISO 5964	Synaptica, T.Builder	N
Portabilidade	Roda em Unix e Windows	Synaptica	N
	Funciona em diversos navegadores (browsers) (especificar)	Synaptica	N
	Depende de softwares que não são livres	Synaptica, T.Builder, Multites	N
Características gerais de limitação	Quantidade máxima de hierarquias (termos raiz) por tesauro (informar)	No caso não se aplica especificar o software, pois cada um possui (ou não) uma limitação específica	N
	Quantidade máxima de níveis de hierarquias (informar)	No caso não se aplica especificar o software, pois cada um possui (ou não) uma limitação específica	N
	Tamanho máximo do termo (informar)	No caso não se aplica especificar o software, pois cada um possui (ou não) uma limitação específica	N
	Quantidade máxima de caracteres para os campos de	No caso não se aplica especificar o software,	N

	metadados do tesauro	pois cada um possui (ou não) uma limitação específica	
	Quantidade máxima de relações para cada termo (informar)	No caso não se aplica especificar o software, pois cada um possui (ou não) uma limitação específica	N

Fonte: Autoria própria.

5 CONCLUSÕES

Como se pode observar, softwares maduros de tesauro contemplam de forma adequada, de modo geral, os requisitos básicos para automação de um tesauro. Entretanto, não encontramos tais softwares com interface em português, tampouco livres e sem custo. Destaca-se também que poucos softwares específicos para tesauros maduros foram encontrados no mercado, sendo que, até onde pudemos perceber, parece haver um maior direcionamento para a construção de softwares voltados para taxonomias e ontologias, como inclusive é o caso do Synaptica, que atende não só a elaboração de tesauros como também de taxonomias e ontologias.

Por fim, observa-se que a adoção do quadro permite minimizar a subjetividade na escolha de um software de tesauro, contribuindo para que ela seja feita de forma mais precisa, uma vez que apresenta uma compilação de requisitos baseadas no que é um tesauro e o seu uso, bem como no que se espera encontrar no mercado. Além disso, o quadro fornece uma síntese de atributos funcionais e não funcionais para o desenvolvimento de um software de tesauro. A lista de requisitos apresentada não é exaustiva, por questões de espaço. Entretanto, é um ponto de partida importante, uma vez que contempla requisitos que são fundamentais para automação de um software de tesauro.

Acrescenta-se que esta investigação é fruto das pesquisas realizadas emgrupo de pesquisa de Instituiçao Federal de Ensino Superior brasileira onde seus componentes possuem perfis interdisciplinares.

REFERÊNCIAS

Achimugu, P., Selamat, A., Ibrahim, R.,& Mahrin, M. N. (2014). A systematic literature review of software requirements prioritizationresearch.*Information and Software Technology*, 56, 568–585.

Berander, P., & Andrews, A. (2005).Requirements Prioritization. In A. Aurum&C. Wohlin (Eds.), *Engineering and Managing Software Requirements* (pp. 69-94). Alemanha: Springer.

Campos, M. L. A. (2001). *Linguagem documentária: teorias que fundamentam sua elaboração*. Niterói, RJ: EdUFF.

Campos, M. L. A., Gomes, Hagar E.,Campos, M. L. M.,Campos, L. M.,Sales, L. F., Martins, A. E., & Souza, S. R. (2006, Janeiro/Abril). Estudo Comparativo de Softwares de Construção de Tesauros. *Perspectivas em Ciência da Informação*, 11(1), 68-81.

Falbo, R. A. (2005).*Engenharia de Software*. Notas de Aula, Universidade Federal do Espírito Santo - Departamentode Informática, Espírito Santo. Recuperado de http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf

Ganzmann, J. (1990). Criteria for the evaluation of thesaurus software.*International Classification*, 17(3/4), 148-157.

Gomes, H. E. (1984). *Diretrizes para elaboração de tesauros monolíngues*. Brasília:IBICT.

Gomes, H. E.,& Campos, M. L. A. (2004). Tesauro e normalização terminológica: o termo como base para intercâmbio de informações.*DataGramaZero*, 5(6).

Gomes, H. E., Campos, M. L. A, & Motta, D. F. (2004). *Elaboração do tesauro documentário: tutorial*. Recuperadoem 24fevereiro, 2017de http://conexaorio.com/biti/tesauro

International Federation of Library Associations and Institutions.(2009).*Guidelines for multilingual thesauri: working group on guidelines for multilingual thesauri IFLA classification and indexing section* (IFLA Professional Reports, No. 115). The Hague: International Federation of Library Associations and Institutions.

Lancaster, F.W. (1986). *Vocabulary Control for Information Retrieval*. Virginia: Information Resources Press.

Multites.com - Thesaurus and Taxonomy Authoring Solutions.(2017). *Multites.com*. Recuperado em 23 fevereiro, 2017 de http://www.multites.com/

Soergel,D. (2002). Thesauri and ontologies in digital libraries: tutorial. In *European Conference on Digital Libraries (ECDL 2002*). Rome, Italy. Recuperado dehttp://www.dsoergel.com/cv/B63_rome.pdf

Sommerville , I. (2007).*Engenharia de Software* (8a. ed.). Brasil: Pearson Education.

Synaptica | Synaptica Central. (2017). *Synapticacentral.com*. Recuperado em 23 fevereiro, 2017 de http://www.synapticacentral.com/content/synaptica

Thesaurus Builder. (2017). *Thesaurus Builder*. Recuperado em 23 fevereiro, 2017 de http://thesaurusbuilder.com/