

XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017

GT-2 – Organização e Representação do Conhecimento

ESTUDO DE REQUISITOS PARA ESCOLHA DE *SOFTWARES* DE TESAURO

Linair Maria Campos - Universidade Federal Fluminense (UFF)

Maria Luiza de Almeida Campos - Universidade Federal Fluminense (UFF)

Joice Cleide Cardoso Ennes de Souza - Universidade Federal Fluminense (UFF)

STUDY OF REQUIREMENTS FOR THESAURUS SOFTWARE SELECTION

Modalidade da Apresentação: Comunicação Oral

Resumo: Para a construção e gestão de tesouros é indispensável o uso de *softwares* que promovam o controle terminológico e gestão efetiva de seu aperfeiçoamento. Devido a isso, a escolha da ferramenta adequada, levando em conta as necessidades peculiares ao seu uso nas corporações, é tarefa de suma importância. Contudo, as ferramentas existentes no mercado diferem em suas características e há uma variedade de opções, sendo por isso necessária a adoção de uma metodologia para que a escolha seja mais precisa e menos sujeita a julgamentos subjetivos. O presente trabalho objetiva propor uma abordagem visando orientar a escolha de um determinado software de tesouro, voltado para a finalidade e a complexidade de um tesouro específico a ser construído. Metodologicamente, fundamenta-se na análise de requisitos básicos para *softwares* de tesouro, 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 tesouro, agrupados de acordo com categorias pré-definidas. Como conclusão, observa-se que a adoção do quadro permite minimizar a subjetividade na escolha de um *software* de tesouro, permitindo uma tomada de decisão de forma mais precisa.

Palavras-Chave: *Softwares* de Tesouros; Avaliação de *Software*; Requisitos de *Software*.

Abstract: For the construction and management of thesauri it is indispensable to use software that promotes terminological control and effective management of its improvement. Because of this, choosing the right tool, taking into account the peculiar needs of its use in corporations, is a task of paramount importance. However, the existing tools in the market differ in their characteristics and there are a variety of options, hence it is necessary to adopt a methodology so that the choice is more precise and less subject to subjective judgments. The present work aims to propose an approach aimed at guiding the choice of a particular thesaurus software, focused on the purpose and complexity of a specific thesaurus to be constructed. Methodologically, it is based on the analysis of basic requirements for thesaurus software, in order to analyze requirements as a proposal 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 table 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 tesouros apresentam 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 tesouro pode contribuir para a precisão da recuperação da informação, minimizando ambiguidades, respeitando o uso da linguagem variada ao tratar o uso de sinonímias (GOMES; CAMPOS, 2004), apontando possibilidades alternativas para expansão de pesquisas, através de termos associados. Além disso, o uso do tesouro apoia no entendimento do conceito a ser utilizado, na indexação e na recuperação, ao permitir o uso de notas de indexação e de definição. Nesse contexto, é importante destacar que para cada função do tesouro existe uma necessidade que ele pode atender. Assim, o indexador utiliza o tesouro 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 tesouro como um instrumento que evidencia as temáticas existentes no domínio de dados a serem recuperados. O tesouro 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.

No escopo desse trabalho, nosso foco está centrado no profissional que elabora ou necessita escolher um *software* de tesouro no mercado. Para isso, nos reportaremos, quando for o caso, 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.

O perfil do profissional da informação atuante na gestão terminológica pressupõe competência no que diz respeito à criação de termos, sua associação com outros termos, definição de sinônimos e notas de explicação voltadas para os usuários que, efetivamente, usarão o tesouro. Para esse profissional, a elaboração de um tesouro 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 tesouro de forma a poder facilmente entender e alterar sua estrutura e controlar as mudanças realizadas. Dessa forma, considerando que a construção de um tesouro 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 tesouros que adotam o conceito como base de formação terminológica (CAMPOS et al., 2006).

Softwares para elaboração de tesouros 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 tesouro?

Defendemos a ideia que para responder a essas perguntas é premente a adoção de uma abordagem metodológica para que a escolha do *software* de tesouro 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 (CAMPOS et al., 2006; GANZMANN, 1990), e, além disso, boa parte 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 tesouro. Naturalmente, é possível derivar um conjunto de características desejáveis para um *software* de tesouro 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 tesouro, destacam-se Ganzmann (1990), Soergel (2002), Campos (2001), *International Federation of Library Associations and Institutions* (2009), Campos et al. (2006), e Gomes, 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). 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: Synptica (2017), Multites (2017) e Thesaurus Builder (2017). Além disso, há que se considerarem aspectos que não dizem respeito a um tesouro 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.

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

O trabalho se insere no âmbito de um grupo de pesquisa da área da Ciência da Informação e, como tal, apresenta resultados que estão em conformidade com projeto de pesquisa inserido nesse contexto. Está estruturado da seguinte forma: aborda-se o tesouro como instrumento de representação e recuperação da informação; apresentam-se os *softwares* de tesouro analisados; abordam-se requisitos e sua caracterização; apresentam-se os procedimentos metodológicos, seguidos pelos resultados e pelas conclusões.

2 TESAURO COMO INSTRUMENTO DE REPRESENTAÇÃO E RECUPERAÇÃO

Linguagem artificial disseminada na área de representação e recuperação da informação, o tesouro remonta historicamente a 1852, ano em que Peter Mark Roget publicou a obra "*Thesaurus of English Words and Phrases*". Ao contrário dos dicionários publicados até então, onde se partia de uma palavra para chegar ao seu significado, a obra de Roget foi organizada de forma a localizar as palavras de acordo com as ideias que expressavam. "Parte-se de um significado, de uma idéia, para se chegar às palavras que a melhor representem" (ROGET apud CAMPOS, 2001, p. 88). O Thesaurus de Roget foi organizado em duas partes: estrutura classificatória ou sistema de classificação de ideias, segundo o autor, e um índice alfabético dos cabeçalhos. As palavras foram organizadas em seis categorias: relações abstratas, espaço,

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

matéria, intelecto, volição e afeições. As categorias eram subdivididas sucessivamente em classes, seções, subseções e conceitos isolados.

O uso contemporâneo da palavra tesouro, conforme aponta Campos (2001), deve-se a Hans Peter Luhn, do *Research Center* da IBM, em 1950. Ele foi o primeiro a utilizar o termo tesouro para designar uma lista de palavras autorizadas com relações estabelecidas entre elas com fins documentários.

Em 1973, através do programa UNISIST, a UNESCO divulga sua definição de tesouro focando dois aspectos: estrutura e função. Para a instituição, de acordo com a estrutura, tesouro "é um vocabulário controlado e dinâmico de termos relacionados semântica e genericamente cobrindo um domínio específico do conhecimento". Já no tocante à função, considera o tesouro como "dispositivo de controle terminológico usado na tradução da linguagem natural dos documentos, dos indexadores ou dos usuários numa linguagem do sistema (linguagem de documentação, linguagem de informação) mais restrita" (apud CAMPOS, 2001, p. 91). Tal definição indica a importância de se elaborar o instrumento dentro da realidade de um sistema de informação, atendendo às necessidades de seus diferentes participantes e a uma realidade específica. Para Campos (2001), a definição da UNESCO atende à elaboração e à organização e recuperação da informação, respectivamente.

Maniez (1993) destaca que a grande vantagem do tesouro é a integração da ferramenta com a lógica dos sistemas especialistas, que permite uma busca mais consistente e ampla para o usuário. No que diz respeito ao modelo de linguagem por facetas "postula que todo assunto pode e deve ser decomposto numa sequência ordenada de categorias semânticas fundamentais" (MANIEZ, 1993), acumulando vantagens do tesouro com as da classificação hierárquica.

Em 1969, Jean Aitchinson apresentou o Thesaurofacet, organizado pelo CRG (*Classification Research Group*), que adota a Teoria da Classificação Facetada na organização dos conceitos combinada ao tesouro convencional, de tradição norte-americana. Como esclarece Lancaster (1986, p. 33) é "uma tentativa de criar um instrumento que compreenda o melhor de ambos os mundos". Segundo Rivier,

este novo conceito integra, em um só sistema, as vantagens da apresentação sistemática dos termos, para análise de uma área em facetas e as dos tesouros alfabéticos. [...] As duas tradições de indexação muito diferentes (americana e indo-européia) assim se encontram (RIVIER, 1992, p. 71).

O Thesaurofacet apresenta a Tabela de Classificação Facetada e o Tesouro Alfabético que são complementares (CAMPOS, 2001). As relações entre os termos são feitas através do UF (*used for*) e Use para relações de equivalência como sinônimos e entrada de termos específicos não usados na indexação e recuperação. Na Tabela de Classificação Facetada observa-se a hierarquia completa composta pelos termos genéricos e específicos, facilitando a pesquisa genérica. Já na listagem alfabética, cada descritor é acompanhado do seu número de classe, facilitando o acesso imediato ao termo específico. Segundo Lancaster (1986), a vantagem do Thesaurofacet sobre o tesouro tradicional consiste no fato do primeiro exibir logo à primeira vista as relações entre os termos e as associações corretas.

Analisando o percurso histórico do tesouro, observamos duas vertentes distintas: a europeia, influenciada pela Teoria da Classificação Facetada de Ranganathan (abordagem sistemática) e a norte-americana, com raízes no Unitermo (abordagem alfabética) (LANCASTER, 1986).

Aqui defendemos a vertente europeia para a elaboração de tesouros, por possuir uma parte sistemática que consideramos fundamental para garantir a consistência semântica entre os termos. Neste sentido, os requisitos que serão apresentados posteriormente pretendem atender a construção teórica e metodológica desta tipologia de tesouro.

A vertente que surge na Europa, segundo Campos (2001, p. 97), “oferece princípios para melhor posicionar o conceito no sistema de conceitos”, fornecendo as bases para a ordenação das classes e propondo a apresentação sistemática do tesouro, além da ordenação alfabética dos termos. Campos (2001) relata ainda que, apesar dos avanços na construção deste novo modelo de linguagem documentária pós-coordenada, observa-se que as vertentes apresentam base teórica explícita e instruções para seleção da unidade de trabalho com base linguística.

Segundo Campos (2001), na década de 70, Ingetraut Dahlberg introduz a Teoria do Conceito aliada à defesa do uso de Categorias, apresentado anteriormente por Ranganathan, implementando uma nova metodologia para a construção de tesouros, que dispõe de uma solução para a organização dos conceitos em um Sistema de Conceitos, independentemente da finalidade de aplicação, onde as categorias têm um papel fundamental. Em 1977, tem uma oportunidade de aplicar a metodologia em um estudo-piloto para a *Deutsche Bibliothek*. O estudo culmina na publicação, em 1989, de um piloto na área de Esporte. Esse piloto possui uma parte sistemática, onde ficam evidenciadas as relações de gênero-espécie, e também possui uma indicação para os termos não preferidos.

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

Em 1984, ainda em conformidade com a linha europeia de tesouros, Gomes publica uma obra com diretrizes para elaboração de tesouros monolíngues. Segue os princípios da *British Standard Institution* “que, por seu turno, fundamentam-se na Publicação 2788 da ISO (“*Guidelines for the establishment and development of monolingual thesauri*”)” (GOMES, 1984, p.1). Em sua obra, Gomes destaca que

Um autêntico tesouro é construído a partir dos conceitos, apresentados num arranjo sistemático. Esse arranjo é determinado pelos elementos classificatórios que se façam presentes nas áreas específicas do conhecimento - o que torna licito afirmar que os tesouros são suscetíveis de converter-se, a qualquer momento, em verdadeiros sistemas de classificação (GOMES, 1984, p. 2).

Partindo do pressuposto que o tesouro é construído a partir de conceitos, Gomes (1984) propõe um conjunto de diretrizes, que continuam válidas na atualidade, e são voltadas para o tesouro enquanto instrumento de recuperação da informação. Essas diretrizes levam em conta as finalidades principais de um tesouro, que, de acordo com Gomes (1984) são: (i) controle terminológico na indexação “mediante um instrumento que traduza a língua natural dos autores, indexadores e pesquisadores numa linguagem mais controlada, usada na indexação e recuperação” (GOMES, 1984, p.1); (ii) assegurar a consistência no uso dos termos na indexação, de modo que diferentes indexadores usem a mesma terminologia; (iii) limitar o número de termos usados para indexar os documentos, de modo que os termos associados a um documento representem “tão especificamente quanto possível, os conceitos expostos pelo autor, sem que haja necessidade de incluir termos de conotação mais ampla e demais termos associados, se tais termos estiverem implicados nos quadros normais de referência” (GOMES, 1984, p.1-2); (iv) servir como instrumento de apoio para recuperação de informação.

Dentre as diretrizes de Gomes (1984), destacam-se, para efeitos de elaboração de um *software* de tesouro com base no conceito, as seguintes orientações gerais:

- i. uso de descritor, que é “palavra ou expressão que representa um conceito, ou seja, o termo preferido para a indexação” (GOMES, 1984, p.5);
- ii. uso de termo indicador de faceta, que é um “termo que não é atribuído aos documentos na indexação, mas que se insere na parte sistemática do tesouro para indicar a base lógica adotada na subdivisão de uma classe” (GOMES, 1984, p.5);

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

- iii. uso de relações “vigentes entre os termos ou descritores - sinonimicas, hierárquicas e outras - que, no conjunto, constitui a linguagem da indexação” (GOMES, 1984, p.5);
- iv. uso de abreviaturas para relações e notas (como, por exemplo, TG para termo genérico, TE para termo específico, TO para termo oposto, NA para nota de aplicação, dentre outras;
- v. a importância do uso de notas de aplicação e definições, pois “podem ser acrescentadas aos termos para limitarem o sentido que o descritor deve ter na indexação, excluindo assim possíveis significados alternativos” (GOMES, 1984, p.15);
- vi. a adoção de informações complementares que são úteis para entender o contexto de criação do termo, a saber: “a data em que um termo foi adotado, ou em que seu âmbito foi redefinido dentro do serviço de indexação” (GOMES, 1984, p.16), e, ainda a fonte do termo e o nome das autoridades consultadas;
- vii. quanto à apresentação, gráfica, destaca-se a importância de “apresentação alfabética, com notas de aplicação e relacionamentos para cada termo; apresentação sistemática com índice alfabético; apresentação planigráfica com índice alfabético” (GOMES, 1984, p.36), sendo que a autora esclarece que uma apresentação planigráfica é uma “representação gráfica que explicita as diferentes facetas da terminologia de um tesouro, suas relações hierárquicas, partitivas e outras, mediante os variados recursos do desenho geométrico” (GOMES, 1984, p.36);
- viii. controle de exclusão de termos, de modo que “Se um termo for retirado do tesouro, mas tiver figurado anteriormente na indexação, seus registros devem receber a anotação "Apenas para fins de recuperação", e a data de retirada do termo deve ser também registrada” (GOMES, 1984, p.55);
- ix. uso de recursos de automação para “Seleção automática de termos candidatos extraídos de títulos e resumos; Determinação automática (...) da frequência com que os termos são usados na indexação e nas perguntas de busca” (GOMES, 1984, p.56), de modo a entender a relevância do uso do termo e, eventualmente, sugerir a sua exclusão, se o uso for baixo;

- x. preocupação com as limitações que podem existir, tais como: “número de caracteres permitido num termo; conjunto de caracteres, inclusive sinais de pontuação, disponíveis na impressora; número de níveis de subordinação permitido num tesouro sistemático” (GOMES, 1984, p.56).

Outra iniciativa alinhada com a linha europeia de tesouros é o método relacional, proposto por Motta (1987), baseado na abordagem relacional de Dahlberg (1976). Esta se apoia em estudos sobre a teoria do conceito (DAHLBERG, 1978) para propor o método relacional na construção de sistemas de classificação. De acordo com Motta (1987, p. 39), o método relacional é um "Método analítico que consiste na análise das definições dos conceitos que integrarão o sistema, de forma a identificar suas características, e na estruturação desses conceitos, tendo em vista as relações existentes entre eles". Motta, como um dos resultados da aplicação de seu método, aponta para um conjunto de relações úteis em tesouros conceituais, a saber: TG (termo hierárquico superior), TE (termo hierárquico inferior), TGP (termo partitivo superior), TEP (termo partitivo inferior), TA (termo associado). Além dessas relações, usa o símbolo up para indicar a relação com o termo não preferido, e o símbolo NA para notas de aplicação, para "particularizar o uso de determinados termos" (MOTTA, 1987, p. 50).

Mais recentemente, em 2010, reafirmando um documento de 2005, a *National Information Standards Organization* (NISO) propõe um conjunto de diretrizes para a construção, formatação e gestão de vocabulários controlados monolíngues (NISO, 2010), das quais destacamos, a título de complementação das iniciativas mencionadas anteriormente, os aspectos relatados a seguir.

Em relação às notas associadas aos termos, a NISO sugere a adoção de notas de escopo, para designar o significado escolhido para o termo, e notas históricas para registrar e acompanhar a criação e as mudanças ocorridas nos termos ao longo do tempo.

No que tange ao uso de relações, observa a importância do uso de (i) variações léxicas, que são diferentes nomenclaturas para a mesma expressão, o que é diferente de sinônimos, que são termos diferentes para o mesmo conceito; (ii) o uso de quase sinônimos, que são termos cujo significado pode diferir, mas que são tratados como equivalentes, como, por exemplo, água do mar e água salgada; (iii) relações de instância, como, por exemplo: os Alpes como uma instância de região montanhosa.

Em relação à apresentação, deve-se levar em conta o perfil de cada usuário, a saber: (i) **gestores do vocabulário**: devem ter acesso e navegar por todas as visões do vocabulário, todos

os dados dos termos e suas relações, podendo criar, alterar e excluir dados; (ii) **indexadores e especialistas em buscas**: devem poder buscar e visualizar os termos, suas notas, relações e navegar em sua hierarquia; (iii) **usuários finais**: devem poder visualizar os termos do tesouro e suas relações, e especialmente, ter acesso a ajuda *online*.

Além da questão do perfil, várias funcionalidades são sugeridas para facilitar a apresentação do tesouro, e suas buscas a saber:

- i. indentação de termos, de modo a fornecer ajuda na visualização de relações hierárquicas e associativas;
- ii. recursos para ordenação dos termos e suporte ao uso de caracteres não alfanuméricos;
- iii. exibição rotacionada (*permuted or rotated*) dos termos;
- iv. uso de *hyperlinks* para o termo, apontando para detalhes adicionais, tais como notas e histórico;
- v. exibição da estrutura do tesouro por cada termo raiz (*top term*);
- vi. exibição facetada, de acordo com a categoria mais ampla a que o termo pertence;
- vii. exibição do tesouro de forma gráfica, de modo a facilitar a visualização das relações entre os conceitos do tesouro;
- viii. busca por texto livre para todos os termos, incluindo parte do nome;
- ix. busca por termos não preferidos, com a indicação do termo preferido associado;
- x. busca sem ser sensível a maiúsculas;
- xi. visualização de dados gerais do termo, tais como, histórico, notas, relações;
- xii. visualização da hierarquia em seus diversos níveis de profundidade;
- xiii. exibição da profundidade da navegação (também denominado como trilha de migalhas). Por exemplo: livros> assunto> história > Europa. Isso facilita a navegação, em especial quando o usuário precisa voltar para a tela anterior;
- xiv. uso de ajuda *online*.

No que tange à gestão terminológica, a NISO sugere algumas diretrizes, justificando a necessidade de apoio a decisões de natureza intelectual por parte dos gestores do tesouro. Nesse sentido, destaca a necessidade de tratamento de termos candidatos, tanto no que se refere à sua importação (a partir, por exemplo, de textos legíveis por máquina), quanto à sua proposição pelos próprios usuários ou pelo *software*, de acordo com o perfil de uso do tesouro

(por exemplo, a partir de buscas frequentes por um termo que não existe no tesouro). Sugere ainda que os termos candidatos possam ser identificados como tal e aprovados ou não. Outra questão para a gestão é a validação do tesouro, de modo a mantê-lo íntegro e consistente. Por exemplo, quando um termo é deletado, o *software* deve pedir confirmação e garantir que todos os links para o termo deletado sejam também excluídos, informando quando da criação de termos órfãos (sem um termo genérico ou associado). Além disso, o *software* deve evitar erros do tipo: termos duplicados e potencialmente duplicados (diferindo, por exemplo, apenas em relação a letras maiúsculas), e, ainda, inconsistências estruturais tais como relações recíprocas faltando.

Outro aspecto destacado pela NISO é a interoperabilidade. Nesse quesito, destaca necessidades tais como indexar conteúdo de um domínio com vocabulários de outros domínios e, ainda, indexação, busca e recuperação multilíngue.

Detalhes adicionais de implementação também são observados, tais como, preferir o uso de *software* não proprietário, multiusuário, independente de hardware e sistema operacional específico, facilidade de uso, documentação detalhada disponível.

3 OS SOFTWARES DE TESAURO: SYNAPTICA, MULTITES PRO E THESAURUS BUILDER

Apresentamos a seguir uma visão geral dos *softwares* de tesouro avaliados, conforme documentação presente no sítio web de cada ferramenta. Os requisitos mais detalhados de cada *software*, fruto de uma análise minuciosa, encontra-se no Quadro 1.

O *software* **Synaptica** (www.sinaptyca.com) foi lançado em 1995 e continua sendo evoluído até a presente data, de acordo com o sítio *web* da ferramenta. O *software* é licenciado pela empresa americana Synaptica, LLC. É voltado para gestão de taxonomias e ontologias, porém se aplica também a tesouros. Sua arquitetura é moderna, voltada para *web*, com uso de APIs (interfaces de programação) abertas e serviços *web*. Pode ser configurado para manipular taxonomias, vocabulários controlados, tesouros, arquivos de autoridade, dentre outros vocabulários. O *software* **Multites Pro** (www.multites.com) foi lançado em 1983 e, até onde pudemos perceber, a última versão data de 31 de dezembro de 2016. O *software* é desenvolvido pela empresa americana Multisystems. É um *software desktop* voltado para ambiente Windows, o que sugere uma arquitetura computacional pouco flexível. Permite criar e gerenciar tesouros, taxonomias e outros vocabulários controlados. O *software* **Thesaurus Builder** (www.thesaurusbuilder.com) é fabricado pela empresa CodeSells, sediada em Amsterdã. Até onde pudemos perceber, a última versão data de 13 de junho de 2012. Da mesma forma que o

Multites Pro, é um software desktop para ambiente Windows, sugerindo pouca flexibilidade em sua arquitetura computacional. A empresa tem experiência de mais de 8 anos em construção de *softwares* de tesouro sendo o *software* adotado em instituições de diversos países.

4 REQUISITOS 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 tesouros, 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 tesouro do ponto de vista de sua concepção intelectual, como, por exemplo, visualização do tesouro 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 a impressão do tesouro, como, por exemplo, relatórios sistemáticos, relatórios definidos pelo usuário; tratamento de dados - funcionalidades ligadas aos dados contidos no tesouro 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 tesouro, 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 tesouro.

Já os requisitos não funcionais descrevem restrições sobre as funcionalidades do *software*, tais como restrições de espaço, 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 tesouro 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 tesouro, 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 um conjunto de categorias, 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.

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 tesouro, 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 et al., 2014), estando fora do escopo do presente trabalho detalhá-las.

5 PROCEDIMENTOS METODOLÓGICOS

A abordagem é baseada metodologicamente na análise de requisitos básicos para *softwares* de tesouro, 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 requisitos funcionais e não funcionais. Os requisitos não funcionais são agrupados de acordo com as categorias propostas por Sommerville (2007). 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 tesouro obtidos.

Os requisitos são obtidos a partir de levantamento na literatura sobre as características de um tesouro e da análise de um conjunto de ferramentas atuais. A coleta de requisitos a partir da análise de softwares já existentes é prática adotada na área de Engenharia de Software em especial no caso reformulação de sistemas legados (FAHMI; CHOI, 2007).

Deste modo, procura-se confrontar o que é necessário para que um tesouro 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 tesouro, com as necessidades que são inerentes ao instrumento em si, levando-se em conta os diferentes perfis de uso do tesouro (gestão terminológica, indexação, recuperação de informação), de modo a garantir que sejam atendidos os requisitos mínimos que a automatização de um *software* de tesouro deve ter. Foram avaliados os *softwares*: Synaptica, Multites e Thesaurus Builder, analisados com base na última versão disponível. 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 do *software*. Esses pressupostos visam garantir minimamente que o *software* possua maturidade, ou seja, que possua um grau pequeno de falhas e, além disso, que possua alguma receptividade no mercado, o que é um indício de sua adequação. Os critérios adotados foram adaptados de Ganzmann (1990).

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 tesouro 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 tesouro para indexação e recuperação de conteúdo em bases de dados associadas.

6 RESULTADOS

A análise das características de um tesouro conceitual foi feita à luz de sua concepção como instrumento de recuperação da informação, tomando como base estudos da área da Ciência da Informação que definem o tesouro como instrumento para indexação e busca de conteúdos documentais. Por outro lado, foi feita também uma análise de como os fornecedores de *software* têm levado em conta essas características em seus produtos. Os recursos de *software* existentes na atualidade permitem implementar soluções flexíveis e sofisticadas, tanto para contemplar os metadados que são necessários para um tesouro, como, por exemplo, data de criação de termos, relacionamentos, notas de aplicação, etc., até mecanismos de visualização e relatórios diversos que permitem ao usuário entender de forma mais precisa o contexto em que os termos se inserem e suas associações.

Como resultado, apresenta-se no Quadro 1 um conjunto básico de requisitos para *softwares* de tesouro, agrupados de acordo com categorias de Sommerville (2007) 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 tesouro. Por exemplo, no requisito “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 Quadro 1 não é exaustiva, porém é representativa das necessidades de automação de um tesouro, considerando o que foi exposto na seção de metodologia.

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

Quadro 1: Requisitos para *softwares* de tesouro.

| CATEGORIA | REQUISITO | ORIGEM | TIPO |
|-------------------------------|--|---|-------------|
| Apoio metodológico | Usa modelos ou templates para apoiar a formação de nomes | Synaptica | F |
| | Visualiza o tesouro de forma gráfica | Synaptica | F |
| | 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 <i>softwares</i> 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, Synaptica | F |
| 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 |

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

| | | | |
|--|---|---|---|
| | 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 <i>software</i> , 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, T.Builder, Multites | N |
| | Suporta configuração de <i>software</i> (em especial banco de dados) compatível com o que a empresa adota / permite | No caso não se aplica especificar o <i>software</i> , pois vai depender do que cada empresa adota/permite, não sendo possível avaliar de antemão | N |
| Interoperabilidade | Integração com outros <i>softwares</i> | Synaptica | N |
| | Integração com outros <i>softwares</i> (ex: Alfresco, MS Sharepoint) de modo a poder usar os termos do tesouro nos <i>softwares</i> | 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 <i>softwares</i> | 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 <i>softwares</i> 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 |
| | Termos relacionados são ligados por hiperlinks para 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 <i>softwares</i> que não são livres | Synaptica, T.Builder, Multites | N |
| Características gerais de limitação | Quantidade máxima de hierarquias (termos raiz) por tesouro (informar) | No caso não se aplica especificar o <i>software</i> , pois cada um possui (ou não) uma limitação específica | N |

| | | | |
|--|--|---|---|
| | Quantidade máxima de níveis de hierarquias (informar) | No caso não se aplica especificar o <i>software</i> , pois cada um possui (ou não) uma limitação específica | N |
| | Tamanho máximo do termo (informar) | No caso não se aplica especificar o <i>software</i> , pois cada um possui (ou não) uma limitação específica | N |
| | Quantidade máxima de caracteres para os campos de metadados do tesouro | No caso não se aplica especificar o <i>software</i> , pois cada um possui (ou não) uma limitação específica | N |
| | Quantidade máxima de relações para cada termo (informar) | No caso não se aplica especificar o <i>software</i> , pois cada um possui (ou não) uma limitação específica | N |

Fonte: Elaboração própria – 2017.

7 CONSIDERAÇÕES FINAIS

Os tesouros são usados como instrumento para indexação e recuperação de informação em meio digital e outros. Por se constituir em um instrumento de controle terminológico, pode contribuir para a precisão da recuperação da informação. Além disso, o uso do tesouro apoia o usuário no entendimento do conceito a ser utilizado, na indexação e na recuperação. Nesse cenário, é importante que os *softwares* de tesouro estejam alinhados para atender o papel que os tesouros desempenham como instrumento não só de recuperação da informação, mas ainda como ferramenta para apoiar metodologicamente os usuários que vão criar o vocabulário e os que vão usá-lo para indexar conteúdos. Esses diversos papéis do tesouro nem sempre são evidentes para os que constroem esses *softwares*, como podemos evidenciar em algumas das ferramentas existentes no mercado. Como se pode observar, *softwares* maduros de tesouro contemplam de forma adequada, de modo geral, os requisitos básicos para automação de um tesouro. Entretanto, não encontramos tais *softwares* com interface em português, tampouco livres e sem custo. Destaca-se também que poucos *softwares* específicos para tesouros 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 Synptica, que atende não só a elaboração de tesouros 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 tesouro, contribuindo para que ela seja feita de forma mais precisa, uma vez que apresenta uma compilação de requisitos baseadas no que é um tesouro 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 tesouro. 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.

Como limitações do trabalho, observa-se o número de softwares analisados. Trabalhos futuros podem expandir a análise para um número maior de softwares. Acrescenta-se que esta investigação é fruto das pesquisas realizadas em grupo de pesquisa "Estudos Ônticos e Ontológicos em Contextos Informacionais: representação, recuperação e métricas", coordenado pela Profa. Dra. Maria Luiza de Almeida Campos.

REFERÊNCIAS

- ACHIMUGU, P. et al. A systematic literature review of software requirements prioritization research. **Information and Software Technology**, v.56, p. 568–585, 2014.
- BERANDER, P.; ANDREWS, A. Requirements Prioritization. In: AURUM A.; WOHLIN, C. (Eds.). **Engineering and Managing Software Requirements**. Alemanha: Springer, 2005, p. 69-94.
- CAMPOS, M. L. A. **Linguagem documentária: teorias que fundamentam sua elaboração**. Niterói, RJ: EdUFF, 2001.
- CAMPOS, M. L. A. et al. Estudo Comparativo de Softwares de Construção de Tesouros. **Perspectivas em Ciência da Informação**, v.11, n.1, p. 68-81, jan./abr. 2005.
- DAHLBERG, I. Teoria do conceito. **Ciência da Informação**, Rio de Janeiro, v.7, n. 2, p. 101-107, 1978.
- DAHLBERG, I. Classification theory, yesterday and today. **International Classification**, v. 3, n.2, p. 85-90, 1976.
- FAHMI, S. A.; CHOI, H. Software Reverse Engineering to Requirements. **International Conference on Convergence Information Technology**. IEEE-Computer Society, 2007.
- FALBO, R. A. **Engenharia de Software**. Notas de Aula. Universidade Federal do Espírito Santo - Departamento de Informática, 2005. Disponível em: <<http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf>>. Acesso em: 10 ago. 2017.
- GANZMANN, J. Criteria for the evaluation of thesaurus software. **International Classification**, v.17, n. 3/4, p. 148-157, 1990.
- GOMES, H. E. **Diretrizes para elaboração de tesouros monolíngues**. Brasília: IBICT, 1984.
- GOMES, H. E.; CAMPOS, M. L. A. Tesauro e normalização terminológica: o termo como base para intercâmbio de informações. **DataGramZero**, v.5, n.6, 2004.
- GOMES, H. E.; CAMPOS, M. L. A.; MOTTA, D. F. **Elaboração do tesauro documentário: tutorial**, 2004. Disponível em: <<http://conexaorio.com/bit/tesauro>>. Acesso em: 24 fev. 2017.
- INTERNATIONAL FEDERATION OF LIBRARY ASSOCIATIONS AND INSTITUTIONS. **Guidelines for multilingual thesauri: working group on guidelines for multilingual thesauri IFLA classification and indexing section**, 2009.
- LANCASTER, F.W. **Vocabulary Control for Information Retrieval**. Virginia: Information Resources Press, 1986.

**XVIII ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO – ENANCIB 2017
23 a 27 de outubro de 2017 – Marília – SP**

MANIEZ, J. L'évolution des langages documentaires. Tradução de Hagar Espanha Gomes. **Documentaliste - Scienses de l'information**, v. 30, n. 4-5, p. 254-259, 1993.

MOTTA, D. F. **Método relacional como nova abordagem para a construção de tesouros**. Rio de Janeiro: SENAI / DN, 1987.

Multites.com - Thesaurus and Taxonomy Authoring Solutions. Multites.com, 2017. Disponível em: <<http://www.multites.com/>>. Acesso em: 23 fev. 2017

NISO. Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies, National Information Standards Organization. Baltimore, Maryland, U.S.A., 2010. Disponível em: <http://www.niso.org/apps/group_public/download.php/12591/z39-19-2005r2010.pdf>. Acesso em 2 ago. 2017.

SOERGEL, D. Thesauri and ontologies in digital libraries: tutorial. In: European Conference on Digital Libraries (ECDL 2002). Rome, Italy, 2002. Disponível em: <http://www.dsoergel.com/cv/B63_rome.pdf>. Acesso em: 23 fev. 2017.

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. Brasil: Pearson Education, 2007.

Synaptica | Synaptica Central. Synapticacentral.com, 2017. Disponível em: <<http://www.synapticacentral.com/content/synaptica>>. Acesso em: 23 fev. 2017.

Thesaurus Builder. Thesaurus Builder, 2017. Disponível em: <<http://thesaurusbuilder.com/>>. Acesso em: 23 fev. 2017.