

Implantação de uma Interface Web para Publicação de Tesouros: Um Estudo de Caso para o Compartilhamento do Tesouro Brasileiro de Ciência da Computação

Yorranan Neves de Almeida Costa¹, Gisane Aparecida Michelin¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (Unicentro)

Rua Simeão Varela de Sá, 03 – Vila Carli – CEP 85040-080 Guarapuava – PR – Brasil

5702174007@unicentro.edu.br, gisane@unicentro.br

Abstract. *The internet and its technologies provide a vast platform for data exploration, making it a valuable resource for information acquisition. An effective approach to enhance accessibility for both machines and humans is through the utilization of the Semantic Web. The LAsED has been actively engaged in various controlled vocabulary construction endeavors, with focus on the development of the Brazilian Computer Science Thesaurus. However, the research group encountered a significant challenge due to the absence of a publishing interface commensurate with the thesaurus's significance. To address this issue, a web interface was created to enable the publication of multilingual controlled vocabularies. This interface was established using the Skosmos publishing tool, the Apache Jena-Fuseki RDF server, and two HTTP servers implemented in reverse proxy on the research group's server. Consequently, a functional platform was successfully established for the publication of thesauri, aligned with the specified requirements. Additionally, the implementation process was documented, and a concise evaluation of the system's performance was conducted. The results were achieved through the application of a thesaurus development methodology combined with a development model based on software reuse. The Brazilian Computer Science Thesaurus is freely accessible on a digital, open-access platform at conclusion of implementation.*

Keywords: *HTTP Server, Skosmos, Thesaurus, Publishing interface.*

Resumo. *A internet e suas tecnologias oferecem uma plataforma ampla para a exploração de dados, sendo, portanto, um excelente meio para a obtenção de informações. É possível utilizar a Web Semântica para facilitar o acesso a esses dados tanto por computadores quanto por pessoas. O LAsED tem trabalhado em diversos projetos de construção de vocabulários controlados, entre os quais se destaca o Tesouro Brasileiro de Ciência da Computação. Contudo, um dos desafios enfrentados pelo grupo de pesquisa era a inexistência de uma interface de publicação que estivesse à altura da importância do tesouro. Este trabalho teve como objetivo amenizar esse problema, implantando uma interface web para a publicação de vocabulários controlados multilíngues. A implantação da interface foi realizada utilizando a ferramenta de publicação Skosmos, o servidor RDF Apache Jena-Fuseki, e dois servidores HTTP em proxy reverso no servidor do grupo de pesquisa. Como resultado, obteve-se uma plataforma funcional para a publicação de tesouros, atendendo aos requisitos levantados. Além*

disso, foi documentado o procedimento de implantação e realizado um breve levantamento sobre o desempenho do sistema. Os resultados foram alcançados por meio da aplicação de uma metodologia de desenvolvimento de tesouros unida ao modelo de desenvolvimento baseado em reuso de software. O Tesouro Brasileiro de Ciência da Computação foi disponibilizado disponível gratuitamente em uma plataforma digital e de acesso livre na conclusão da implantação. **Palavras-Chaves:** Servidor HTTP, Skosmos, Tesouro, Interface de Publicação.

1. Introdução

Desde o seu surgimento, a área de informática vem passando por rápidas e grandes transformações. A Internet e suas tecnologias continuam sendo uma grande plataforma de exploração de dados, busca de informação e conhecimento. O acesso aos dados, obtenção de conhecimento e informação a partir deles ainda pode ser facilitado. Esses dados podem receber significado, sendo uma dessas maneiras através do uso de Vocabulários Controlados [Rautenberg et al. 2018, Tanenbaum et al. 2021].

Este trabalho se encontra no contexto da Web Semântica, uma vez que a sua realização é a concretização de um esforço para disponibilizar dados confiáveis ao público sobre a Ciência da Computação. Nele será apresentado um projeto que se conecta aos outros desenvolvidos no LAsEd (Laboratório de Aplicações Semânticas e Distribuídas) da Unicentro. Este Trabalho de Conclusão de Curso é uma demanda que surgiu após a pesquisa realizada por [Costa et al. 2023] com título de “Tesouros e Interfaces de Desenvolvimento para Web”. Esta pesquisa tinha por objetivo o estudo da viabilidade e da implementação de uma interface do Tesouro Brasileiro de Ciência da Computação por meio de software de publicação de vocabulários. O Tesouro Brasileiro de Ciência da Computação foi idealizado e começou a ser desenvolvido por [Rautenberg et al. 2021]; foi publicado na web por [Kelnar e Rautenberg 2020]; e evoluiu no trabalho de [Silva et al. 2023].

Um dos desafios deste projeto foi a seleção das ferramentas a serem utilizadas no processo. Já existem ferramentas que oferecem uma interface para publicação de Sistemas de Organização do Conhecimento (KOS - *Knowledge Organization Systems*), termo que representa o conjunto no qual está incluso os tesouros. É possível realizar a publicação desse tipo de vocabulário usando o Thesa¹ ou TemaTres², entretanto estas ferramentas não atendiam as necessidades do grupo de pesquisa.

Desde a especificação do *Simple Knowledge Organization System* (SKOS) e da *SKOS eXtension for Labels* (SKOS-XL) pela W3C em 2009, um número significativo de KOS convencionais se juntaram ao *mainstream* da Web Semântica. No estudo “*Knowledge Organization Systems (KOS) in the Semantic Web: a multi-dimensional review*”, publicado por [Zeng e Mayr 2019], é realizada uma revisão sistemática dos artigos que usavam o termo “LOD KOS”. Durante esse estudo, notou-se um crescimento expressivo do uso do termo em artigos publicados e, conseqüentemente aumentou-se a presença de KOS.

Um projeto considerado motivação para o desenvolvimento deste é o AGROVOC, pertencente à Organização das Nações Unidas para Agricultura e Alimentação (FAO).

¹<https://www.ufrgs.br/tesouros/index.php/thesa>

²<https://vocabularyserver.com/web/>

Nascido no começo dos anos 80, esse vasto sistema passou por muitas transformações, sendo a última grande mudança a transferência para o modelo SKOS. O projeto foi desenvolvido utilizando a ferramenta Skosmos e só foi possível após a extensão do modelo SKOS (SKOS-XL) [Subirats-Coll et al. 2022, Rajbhandari e Keizer 2012, Soergel et al. 2004].

A justificativa do desenvolvimento deste trabalho se baseia na necessidade de implementar para o LASeD uma plataforma de publicação de vocabulários controlados que utilizasse o padrão de modelagem SKOS (*Simple Knowledge Organization System*).

Nesse contexto, o objetivo desse trabalho é implantar uma interface web para publicação de vocabulários controlados multilíngues. Os objetivos específicos que levam a execução do objetivo principal são: Realizar levantamento de requisitos para a implantação da interface; Averiguar a literatura disponível sobre as funcionalidades do Skosmos como ambiente de publicação de vocabulários controlados; Hospedar um ambiente de publicação de vocabulários controlados; Realizar um estudo de caso para publicação de um tesouro multilíngue no ambiente implantado; Publicar utilizando a interface de publicação o Tesouro Brasileiro de Ciência da Computação; Documentar os procedimentos de implantação da interface de publicação de vocabulários controlados.

Na Seção 2 são apresentados os conceitos utilizados para a realização do trabalho. A Seção 3 apresenta a metodologia, as principais ferramentas e materiais utilizados na execução do projeto. Já na Seção 4 é exposto o processo de desenvolvimento do projeto e implantação do sistema. Nas Seções 5, 6 e ??, são discutidos os resultados obtidos, conclusões e apontados trabalhos vindouros.

2. Referencial Teórico

Para contextualização do trabalho são definidos alguns conceitos importantes utilizados ao longo do trabalho. Na Subseção 2.1 é introduzido o conceito de Web Semântica. Na Subseção 2.2 é abordado o tema de Dados Abertos Conectados. Na Subseção 2.3 é abordado o conceito de Sistemas de Organização do Conhecimento e Tesouros. Por fim na Subseção 2.4 é abordado o modelo Cliente-Servidor.

2.1. Web Semântica

O crescimento da web é um processo contínuo que vem ocorrendo gradualmente, contudo, [Berners-Lee 2006] sugere a organização semântica como uma das maneiras de fazer a Internet crescer de modo saudável. Para isso, ele indica quatro atividades práticas:

- Usar nomes do tipo de Identificadores Uniformes de Recursos (URI)³ para coisas;
- Usar URIs HTTP (*HyperText Markup Language*) para que as pessoas possam procurar esses nomes;
- Quando alguém procurar um URI, prover informação útil, usando os padrões *Protocol and Query Language* (SPARQL);
- Incluir *links* para outras URIs.

Essas ações estão dentro do conceito da Web Semântica. A Web Semântica é então uma extensão da Web atual, na qual é dada à informação um significado bem definido, permitindo que pessoas e computadores trabalhem em cooperação [Berners-Lee et al. 2001].

³Identificador Uniforme de Recurso (*Uniform Resource Identifier* - URI) é uma sequência compacta de caracteres que identifica um recurso abstrato ou físico. [Berners-Lee et al. 2005]

2.2. *Linked Open Data (LOD)*

Na Web Semântica existe uma atenção especial aos dados, pois eles que são os responsáveis por dar significado. Os dados podem ser classificados em não estruturados e estruturados. Dados não estruturados não têm definição ou formatação específica, dificultando o processamento, como imagens, áudios, vídeos e textos. Dados estruturados seguem um formato específico, como tabelas com linhas e colunas, facilitando seu processamento [Rautenberg et al. 2018].

Os dados que podem ser usados, modificados e compartilhados livremente por qualquer pessoa, sob uma licença que permite seu uso livre, são considerados dados abertos [KNOWLEDGE 2024]. Eles quando conectados podem ser usados para gerar contexto e significado, isso é possível através da padronização com a linguagem, sendo um dos meios o RDF. Com isso temos Dados Abertos e Conectados (*Linked Open Data - LOD*) [Rautenberg et al. 2018].

2.3. *Knowledge Organization System (KOS) e Vocabulários Controlados*

Knowledge Organization System é um termo genérico usado para referenciar uma grande quantidade de itens (e.g., Cabeçalhos de Assuntos, Tesouros, Esquemas de Classificação e Ontologias), que foram concebidos para diferentes propósitos e em momentos históricos distintos [Mazzocchi 2019].

Um termo mais específico de KOS são os vocabulários controlados, eles são KOS que utilizam sintaxes padronizadas e uma lista predeterminada de termos que abordam um assunto. Essas listas normalmente identificam uma palavra ou frase preferida para um determinado conceito e, às vezes, fornecem mapeamentos de termos. Eles também definem frequentemente relações entre termos. [Mazzocchi 2019, Standard 2010].

2.3.1. Tesouros

De acordo com [NISO 2005], os tesouros são um tipo de KOS, organizado em uma ordem conhecida e estruturado de modo que as diversas relações entre os termos sejam exibidas claramente e identificadas por indicadores de relacionamento padronizados. Os indicadores de relacionamento devem ser empregados reciprocamente.

Dentro dos tesouros, temos ainda a presença de termos que são uma ou mais palavras que designam um conceito. Os termos dos tesouros no modelo SKOS utilizam recursos mais específicos para a relação hierárquica entre os termos, ou seja, a relação de abrangência, especialidade ou igualdade simétrica quando existe conexão entre os termos [NISO 2005, W3C 2012].

2.4. Modelo Cliente-Servidor

O modelo cliente-servidor é amplamente utilizado e constitui a base de grande parte do uso da rede. Sua realização mais popular é a de aplicação web, em que o servidor fornece páginas web com base em seu banco de dados, em resposta às solicitações do cliente. Em termos de arquitetura, servidores são semelhantes aos computadores tradicionais; entretanto, devem suportar uma quantidade numerosa de requisitos de acesso. Para isso, seu hardware e software devem estar configurados para alta disponibilidade [Tanenbaum e Bos 2024].

Mediante uma análise detalhada do modelo cliente-servidor, é possível observar que dois processos estão envolvidos: um na máquina cliente e outro na máquina do servidor. A comunicação ocorre quando o processo cliente envia uma mensagem pela rede ao processo servidor. Em seguida, o processo cliente opera baseado na mensagem recebida em resposta. Quando o processo servidor recebe a solicitação, ele executa o trabalho solicitado ou busca os dados requisitados e envia uma resposta de volta [Tanenbaum et al. 2021].

3. Materiais e Métodos

Nesta Seção é apresentada a Metodologia de Desenvolvimento na Subseção 3.1, Materiais e Ferramentas na Subseção 3.2.

3.1. Metodologia

A metodologia deste trabalho segue o que foi descrito por [Rautenberg et al. 2021] e pode ser observada na Figura 1, essa metodologia é voltada para o desenvolvimento de ontologias⁴, conjunto do qual também fazem parte. Mais precisamente, este trabalho está contido na fase de Verificação. Este trabalho se limita a cumprir o item *a* da Atividade de Verificação, definida por [Rautenberg et al. 2021] como: publicar o tesouro na Web de Dados - para privilegiar a execução da prova de conceito pelos especialistas, o tesouro pode ser compartilhado em um *endpoint* na Web de Dados.

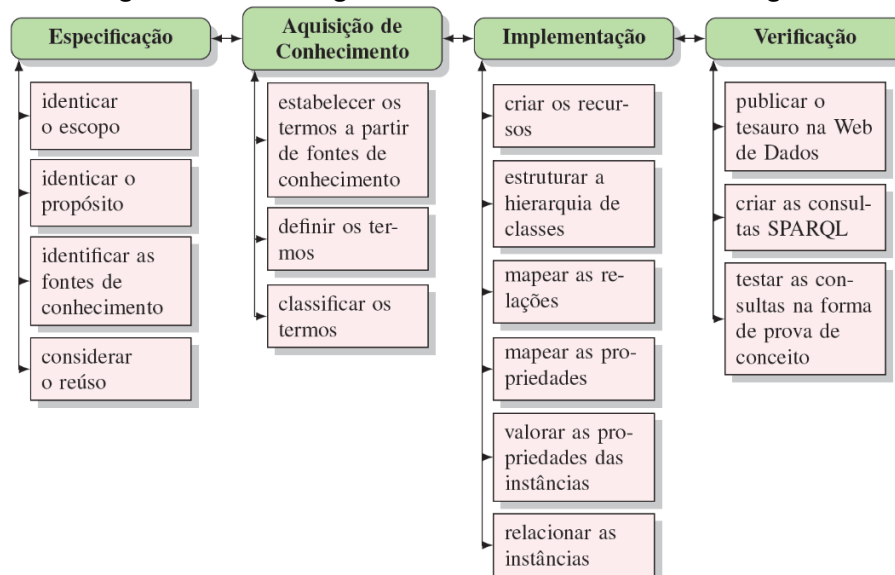
A metodologia de [Rautenberg et al. 2021] foi complementada com os passos de desenvolvimento de software baseado em reuso definido por [Sommerville 2019]. Esse processo de desenvolvimento adéqua-se a proposta deste trabalho devido ao reuso do Skosmos.

1. **Especificação de requisitos:** Os requisitos iniciais são propostos;
2. **Descoberta e avaliação do *software*:** Com base na descrição dos requisitos de *software*, é feita uma busca pelos componentes do sistemas que fornecem a funcionalidade necessárias.
3. **Refinamento de requisitos:** Nesse estágio, os requisitos são definidos com base nas informações dos componentes reusáveis e das aplicações que foram descobertas. Os requisitos são modificados para refletir os componentes disponíveis, e a especificação do sistema é redefinida.
4. **Configuração da aplicação:** Se estiver disponível um aplicação de prateleira que satisfaça os requisitos, ela pode ser configurada para utilização a fim de criar um novo sistema.
5. **Adaptação e integração dos componentes:** Se não houver uma aplicação de prateleira, componentes reusáveis podem ser modificados ou novos componentes podem ser desenvolvidos, visando a integração posterior ao sistema.

Para auxiliar o desenvolvimento algumas ferramentas foram essenciais, elas são descritas na Subseção 3.2.

⁴Ontologia é uma especificação de uma conceitualização, ou seja, uma descrição de conceitos e relacionamentos que existem entre estes conceitos, seu propósito é permitir o compartilhamento e reutilização de conhecimento [Gruber 1993]

Figura 1. Metodologia de Desenvolvimento de Ontologias



Fonte: [Rautenberg et al. 2021]

3.2. Ferramentas e Materiais

As ferramentas e materiais utilizados no desenvolvimento deste trabalho são apresentados nesta Seção. Sendo todos eles partes integrantes fundamentais para o projeto.

3.2.1. *Resource Description Framework (RDF) e SPARQL Protocol and RDF Query Language (SPARQL)*

O RDF é uma especificação do World Wide Web Consortium (W3C) que tem como objetivo a descrição de metadados da web. O RDF fornece um modelo para dados que se refere a objetos (“recurso”) e entre esses objetos relacionamentos (“propriedades”). A estrutura do RDF é baseada em triplas, uma construção que liga um recurso a um valor atribuído por intermédio de uma propriedade específica [Gandon e Schreiber 2016].

O RDF é fundamental para a realização da visão da Web Semântica, onde informações podem ser processadas por máquinas de maneira a facilitar a organização e o compartilhamento de conhecimento. Por meio do uso de URIs, o RDF consegue identificar unicamente os recursos na web, permitindo que as informações sejam integradas e reutilizadas em diferentes contextos [GROUP 2014].

Para realizar consultas em arquivos RDF o Consórcio W3C recomenda o uso do SPARQL. O SPARQL tem como principais funcionalidades:

- Realizar consultas em dados armazenados nativamente como no RDF ou através de *softwares* intermediários;
- Consultar padrões de grafos obrigatórios e opcionais, juntamente as suas conjunções e disjunções;
- Suporte a agregação, sub-consultas, negação, criação de valores por expressões, testes de valor extensíveis e restrição de consultas por gráfico RDF de origem;

Em resumo, o RDF e SPARQL são tecnologias chaves para a interoperabilidade de dados na web. Com a capacidade do RDF sua capacidade de descrever relações complexas entre objetos e de integrar dados de diversas fontes, unido ao seu padrão de consulta, servem como uma base sólida para a construção de uma web mais conectada e semântica.

Neste trabalho em específico o RDF do tesouro é armazenado no Apache Jena-Fuseki que por sua vez utiliza consultas SPARQL integradas, sendo descrito na Subseção 3.2.3.

3.2.2. Turtle

O Turtle é um termo derivado de *Terse RDF Triple Language*. A sintaxe textual é a que permite um grafo RDF seja completamente escrito em um formato de texto compacto e natural, com abreviações para padrões de uso e tipos de dados comuns. Essa sintaxe é definida pelo Consórcio W3C⁵. Uma de suas características é o uso de abreviações para facilitar a escrita de grafos RDF [W3C 2014].

3.2.3. Apache Jena-Fuseki

Apache Jena Fuseki⁶ é um servidor SPARQL. Ele pode rodar como um serviço de sistema operacional, com uma aplicação web Java (arquivo WAR), e como um servidor *standalone*. Na Seção 4 ele foi hospedado como serviço. O Fuseki fornece os protocolos SPARQL 1.1 para consulta e atualização, assim como o protocolo SPARQL *Graph Store* que permite o gerenciamento de coleções RDF [APACHE 2024, Ogbuji 2013].

3.2.4. Skosmos

Durante o projeto de Iniciação Científica na qual o autor participa, a ferramenta Skosmos⁷ foi identificada como a mais adequada para publicação dos tesouros desenvolvidos em projetos anteriores. Essa ferramenta de publicação utiliza como banco de dados um servidor RDF, através da captura dos dados de um *endpoint* SPARQL. Ela conta com algumas funcionalidades importantes para este projeto, sendo as principais [Suominen et al. 2015]:

- Busca de termos;
- Exibição de índice;
- Representação RDF;
- Acesso por meio de API;
- Ser de código aberto.

3.2.5. Proxy Reverso HTTP

O protocolo *Hypertext Transfer Protocol* (HTTP) permite acessar serviços de um endereço mediante o uso de portas. Portas são número antecedido de dois pontos após

⁵<https://www.w3.org/>

⁶<https://jena.apache.org/documentation/fuseki2/>

⁷Disponível em: <https://www.skosmos.org/>

o endereço, como no exemplo: `http://127.0.0.1:80`. Cada porta aponta para um serviço daquele servidor, alguns protocolos contam com portas padrões associadas, por exemplo: (80) HTTP, (443) HTTP *Over* TLS (HTTPS), (22) *Secure Shell* (SSH), etc. [Nielsen et al. 1999, Rescorla 2000].

Um *proxy* HTTP é um programa ou computador intermediário entre diferentes redes. Eles interceptam requisições e respostas, decidindo conforme suas instruções o que fazer com elas. Um *proxy* pode ser dividido em dois tipos [MDN 2023, Tanenbaum et al. 2021]:

- ***Forward proxy (proxy direto)***: Atuam no lado do cliente, armazenando ou encaminhando serviços de Internet, podendo ocultar a identidade do cliente.
- ***Reverse proxy (proxy reverso)***: Atuam como um intermediário obrigatório no lado servidor, podendo inclusive ocultar a identidade do servidor. Entre seus serviços ofertados estão: balanceamento de carga, armazenamento de conteúdo estático em cache. Além disso pode ser redirecionar requisições de clientes para os serviços apropriados com base nas regras de roteamento definidas, carregando as páginas HTTP requisitadas pelo cliente

3.2.6. Nginx

Nginx⁸ (*engine x*) é um servidor HTTP e *proxy* reverso, um servidor *proxy* de correio e um servidor *proxy* TCP/UDP genérico, originalmente escrito por Igor Sysoev. Entre as funcionalidades disponíveis estão [NGINX 2024]:

- Servir arquivos estáticos e de índice;
- Acelerar o serviço de *proxy* reverso com *cache*.
- Suporte ao protocolo HTTP/2 e HTTP/3.
- Servidores virtuais baseados em nomes e Ip.
- Limitar o número de conexões ou solicitações simultâneas provenientes de um endereço;

3.2.7. Apache HTTP Server

O servidor HTTP Apache⁹ é um projeto colaborativo de código aberto para sistemas operacionais modernos como Windows e UNIX. Esse projeto faz parte da Apache Software Foundation. Entre as funcionalidades disponíveis estão [APACHE 2019]:

- Servir arquivos estáticos e de índice;
- Customização de respostas para erros e problemas;
- Capacidade de atender automaticamente clientes de diversas sofisticações e níveis de conformidade com HTML, com documentos que oferecem a melhor representação das informações que o cliente é capaz de aceitar.
- *Hosts* Virtuais – Permite ao servidor distinguir entre solicitações feitas para diferentes endereços IP ou nomes (mapeados para a mesma máquina).
- Permite *proxy* reverso;

⁸<https://nginx.org/en/>

⁹<https://httpd.apache.org/>

4. Desenvolvimento

Seguindo a metodologia foi iniciado processo de estudo sobre a implantação da ferramenta de publicação de vocabulários controlados Skosmos e suas ferramentas auxiliares, esse processo é descrito nesta Seção.

4.1. Levantamento de Requisitos

É a primeira etapa da especificação de requisitos contida no processo de desenvolvimento de *software*, realiza estabelecimento de serviços, restrições e metas do sistema por meio da consulta aos usuários. Tendo isso como base, a implantação da interface deveria cumprir algumas características definidas no levantamento de requisitos, seguindo a metodologia proposta.

O levantamento de requisitos foi realizado a partir de entrevista com membros do grupo LASED em momentos anteriores ao início da escrita do artigo final. Seguindo os conceitos de [Sommerville 2019] sobre a etapa de especificação, os requisitos foram separados em requisitos de usuário e requisitos de sistema.

Requisitos de Usuário:

1. O sistema será utilizado por um tipo de usuário;
2. A primeira tela que será exibida é a de vocabulários disponíveis;
3. O usuário pode acessar qualquer vocabulário livremente;
4. O usuário pode navegar através dos termos;
5. O usuário pode acessar a plataforma por intermédio de um navegador;

Requisitos de Sistema:

1. O sistema deve ser implantado em um servidor local do LASED;
2. O sistema deve utilizar o modelo SKOS;
3. O sistema deve aceitar RDF como forma de armazenamento de dados;
4. O sistema deve mostrar os termos dos tesouros em ordem alfabética;

A partir dos requisitos levantados foi possível definir as ferramentas ideais para a realização da implantação, sendo escolhida a ferramenta *open-source* Skosmos entre as demais de distribuição livre pesquisadas.

4.2. Hospedagem no Servidor

O resultado final deste projeto deveria ser hospedado em servidor local. Além do servidor local, foi realizada a hospedagem no serviço de computação em nuvem Azure¹⁰ com um domínio pessoal do autor¹¹.

Todo o processo de hospedagem foi realizado com o auxílio do protocolo *Secure Shell* (SSH). O SSH é um protocolo de rede utilizado para login remoto e outros serviços de segurança em rede, substituindo protocolos como *File Transfer Protocol* (FTP), *Remote Login* (rlogin) e *Remote Shell* (rsh). O SSH opera sobre um transporte limpo e transparente de 8 bits e, quando utilizado sobre TCP/IP, o servidor geralmente escuta na porta 22, atribuída oficialmente pela *Internet Assigned Numbers Authority* (IANA). Durante a

¹⁰<https://azure.microsoft.com/pt-br/>

¹¹skosmos.yorranan.com

conexão, um algoritmo de criptografia e uma chave são negociados, e os dados criptografados são tratados como um fluxo contínuo, com vetores de inicialização passados entre pacotes. Cada direção de comunicação usa cifradores independentes, recomendando-se o uso do mesmo algoritmo em ambas as direções[Lonvick e Ylonen 2006, OpenSSH 2024].

O servidor no qual foi hospedado a plataforma pertence ao grupo de pesquisa LASeD. As configurações do servidor são:

- **Sistema Operacional:** Debian GNU/Linux 11 (bullseye) x86_64;
- **Kernel:** 5.10.0-30-amd64;
- **CPU:** Intel Xeon E5335 1.995GHz, *octa-core*;
- **GPU:** AMD ATI;
- **Quantidade de memória RAM:** 6 GiB.

4.2.1. Instalação do Fuseki e Skosmos

O processo de instalação utilizado foi baseado na documentação da ferramenta Skosmos¹² com adaptações pontuais para o contexto do projeto.

A hospedagem realizada no servidor começou com a instalação do *Java Runtime Environment* (JRE) versão 11, pré-requisito do servidor do Apache Jena-Fuseki. O motivo de escolher a essa versão é a compatibilidade com a versão 4.9 do Fuseki. A versão 5.0 do Fuseki não tem compatibilidade com a JRE versão 11.

Além da instalação do JRE, também era necessário criar usuários e pastas para a melhor organização da ferramenta. Os diretórios e arquivos de configuração foram criados seguindo a documentação oficial do projeto de banco de dados e da interface de publicação.

Para que o serviço fosse iniciado com a máquina, era necessário criar o arquivo de configuração: `/etc/systemd/system/fuseki.service`, nesse arquivo deveria estar definido alguns parâmetros do ambiente, as configurações desse arquivo estão disponíveis no Apêndice B. Nesse arquivo um aspecto importante é quantidade máxima de memória RAM que a *Java virtual machine* pode consumir, devido a limitação de hardware do servidor esse parâmetro foi definido em 2GiB. Por fim o serviço Fuseki foi instalado na porta 3030 do *localhost*, essa porta não está exposta na rede, sendo seu banco de dados acessível por linha de comando dentro do servidor.

Como pré requisito do Skosmos, foi necessário instalar a linguagem de programação PHP 7.4 e as bibliotecas requeridas pela ferramenta. Na documentação oficial¹³ é mencionado que a ferramenta de publicação foi testada nas versões 7.3, 7.4 e 8.0 do PHP. Essa informação é importante porque uma mudança nas bibliotecas pode ser fatal para a aplicação. Dentre os pacotes instalados estão o pacote *Apache Server 2* e o módulo PHP para integração com o Apache. O Gerenciador de depências de PHP usado foi o Composer 2.7. A versão do Skosmos instalada foi a 2.16.

¹²<https://github.com/NatLibFi/Skosmos/wiki/InstallTutorial>

¹³<https://github.com/NatLibFi/Skosmos>

4.2.2. Carregando Dados

Com o serviço do Fuseki instalado e configurado é possível realizar o *upload* do arquivo RDF que armazena o Tesouro Brasileiro de Ciência da Computação via terminal utilizando o formato Turtle. Esse tesouro continua em construção pelo LAsED.

Durante o *upload* de arquivos RDF para o servidor é importante que seja informado a URI principal, será a partir dessa URI que a ferramenta Skosmos pode localizar os dados de um vocabulário controlado armazenado no serviço de banco de dados.

Houve uma incompatibilidade na visualização de linguagem entre a modelagem do tesouro e a ferramenta Skosmos. Na modelagem as nomenclaturas em espanhol e inglês estão como opcionais, devido a este fato a interface não apresenta os termos em língua inglesa nem em língua espanhola no índice alfabético e hierárquico.

4.2.3. Configurando *Proxy Reverso*

Na máquina onde a interface de publicação foi hospedada já havia outro serviço sendo executado na porta 80 e 443, o o OpenLink Virtuoso¹⁴, esse serviço é responsável por um *endpoint* SPARQL do LAsED, mas sem relação direta com este projeto. Por esse motivo era necessário encontrar uma solução para o conflito entre o Apache e Nginx.

Para resolver esse conflito o meio encontrado foi configurar o `virtual host` do Apache para executar na porta 8080. Na configuração de hospedagem, o Apache busca a partir do diretório `/var/www/html/` (diretório raiz) páginas executáveis, assim possibilitando criar um novo caminho denominado `/Skosmos`. No diretório raiz foi inserido um link simbólico para o diretório em que estava instalada a ferramenta Skosmos.

Nesse momento existiam dois serviços de servidor HTTP sendo executados ao mesmo tempo, mas sem conexão entre eles, isso fazia com que fosse impossível acessar por rede externa o serviço do Skosmos, pois a rede interna da Unicentro só permite acessar externamente páginas HTTPS. Para contornar esse outro empecilho foi configurado um *proxy reverso*.

A primeira opção plausível seria criar um outro subdomínio no endereço que apontasse para o serviço do servidor Apache, entretanto isso era inviável. Então a solução era fazer com que o Nginx em seus arquivos de configuração apontasse para o serviço Apache na porta 8080 quando um caminho específico fosse requisitado.

Como já citado na Seção 3.2.5 o procedimento para desviar rotas em servidores é realizada através de servidores como Nginx e Apache. A ferramenta que ficou responsável por realizar o desvio é a Nginx, nesse problema em específico o convidado deve requisitar o endereço `lodlased.unicentro.br/Skosmos/pt` e ser redirecionado pelo Nginx para o serviço no *back-end* Apache de forma transparente, ou seja, sem que o usuário perceba que houve redirecionamento no servidor. Para realizar essa operação o arquivo de configuração de serviço já existente foi modificado. Para o servidor Nginx foi instruído o redirecionamento de toda requisição com caminho `/Skosmos` para o `localhost:8080/Skosmos`. Em mais detalhes, a instrução `proxy_pass` desvia

¹⁴<https://virtuoso.openlinksw.com/>

a requisição para o servidor Apache, enquanto a instrução `proxy_redirect` mascara o caminho para `lodlased.unicentro/Skosmos`.

Além do `proxy` reverso responsável pelo redirecionamento é sugerido como forma de melhorar o desempenho da ferramenta o uso do Varnish ou a ferramenta *Alternative PHP Cache*¹⁵ (APC). O Varnish *cache* é um *software* mantido pela *Varnish Software* e comunidade por ser um projeto de código aberto, ele funciona como *proxy* reverso focado em reduzir a latência e carga de servidores. Ele faz isso atendendo solicitações de clientes com conteúdo armazenado em *cache* na memória principal (RAM), eliminando a necessidade de enviar solicitações diretamente para os servidores web. A extensão do PHP APC é um método mais simples que armazena na memória principal páginas recentemente acessadas [GROUP 2024]. Na Seção 5 é discutido o teste realizado entre elas.

4.2.4. Configurações Finais

A fase de configuração da ferramenta se baseia em modificar o arquivo `config.ttl` presente nos arquivos da ferramenta, nele é necessário inserir dados sobre cada vocabulário que será utilizado na ferramenta, além de configuração globais, incluindo: nome do serviço, idiomas disponíveis, nomenclaturas utilizadas, tempo de atualização, etc. O arquivo de configuração da ferramenta usa o formato Turtle.

Na tripla de configuração, presente na Quadro 1 do tesouro era necessário indicar o mesmo identificador do grafo (`skosmos:sparqlGraph`) repassado durante o *upload* do arquivo RDF no Fuseki.

```
1 :tbcc a skosmos:Vocabulary, void:dataset ;
2     dc:title "Tesouro Brasileiro de Cincia da Computao"@pt;
3     skosmos:shortName "TBCC" ;
4     dc:subject :cat_general ;
5     dc:type mdrtype:THESAURUS ;
6     void:uriSpace "http://lodlased.unicentro.br/sparql" ;
7     skosmos:language "pt" ;
8     skosmos:defaultLanguage "pt" ;
9     skosmos:showTopConcepts true ;
10    skosmos:fullAlphabeticalIndex false ;
11    void:sparqlEndpoint <http://localhost:3030/skosmos/sparql> ;
12    skosmos:sparqlGraph <http://lod.unicentro.br/2017/DiretrizesSBC
13    > .
14
```

Quadro 1. Arquivo de configuração do Skosmos

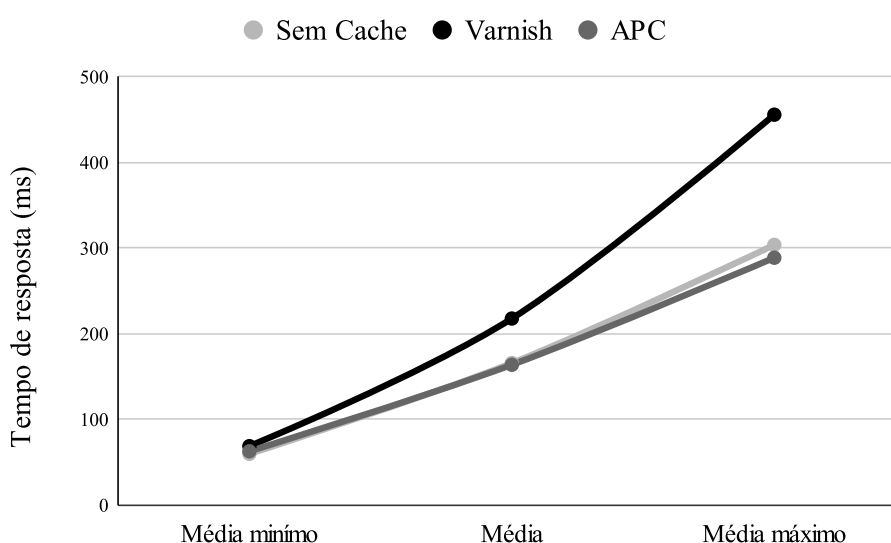
Por fim, no arquivo de configuração da ferramenta Skosmos (`config.ttl`), era necessário modificar a propriedade `skosmos:baseHref` adicionando o domínio, ficando da seguinte forma: `skosmos:baseHref` `''https://lodlased.unicentro.br/Skosmos''` ..

¹⁵https://www.php.net/manual/pt_BR/book.apcu.php

5. Resultados e Discussões

Nesta Seção são abordados os resultados obtidos ao final do processo de implantação da ferramenta. Como meio de torna mais palpável os resultado, foram realizados testes de *benchmark*¹⁶ utilizando a ferramenta *Apache Bench*. As rodadas de testes consistiam em executar mil requisições, sendo dez requisições concorrentes na página inicial do tesouro. Foram executadas dez rodadas antes da utilização de quaisquer serviços de *cache*, dez rodadas somente com o serviço APC, e dez rodadas com o serviço Varnish. Após a análise dos resultados disponíveis na Figura 2 e detalhados no Apêndice B.

Figura 2. Comparativo do tempo médio de resposta das ferramentas.



Fonte: O Autor

A partir dos resultados obtidos constatou-se que:

- O uso do serviço Varnish é prejudicial no tamanho atual do sistema;
- O uso da ferramenta APC gerou ganhos em torno de 5% na média máxima global¹⁷, sendo o melhor resultado em comparação com o sistema sem *cache*;
- O serviço de *cache* APC precisar passar por mais testes.

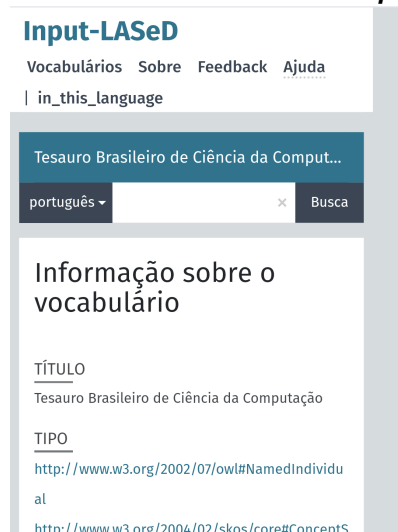
A hipótese levantada a partir dos teste é que devido ao tamanho do tesouro e a quantidade de dados armazenados, não existe a necessidade de utilizar essa ferramenta de *cache*.

Embora não tenha sido um objetivo do trabalho, observou-se que a interface de publicação não é adequada para *smartphones*, devido a falta da interface de navegação por índice alfabético, esse problema pode ser observado na Figura 3.

¹⁶*Benchmark* é uma abordagem para mensurar performance em sistemas de tempo real [Weiderman e Kamenoff 1992].

¹⁷Cálculo realizado: $\frac{\mu_{APC}}{\mu_{Semcache}} = \frac{289_{(ms)}}{304_{(ms)}} \approx 4,934\%$

Figura 3. Tela do tesauro em *smartphones*.



Fonte: O Autor

Além desses aspectos, durante a fase de testes foi identificado alguns pontos de melhoria importantes para a ferramenta, entre eles:

- Modificação do tesauro para mostrar os termos preferenciais em inglês;
- Adequação da interface para ser responsiva, facilitando o uso em telas de *smartphones*.

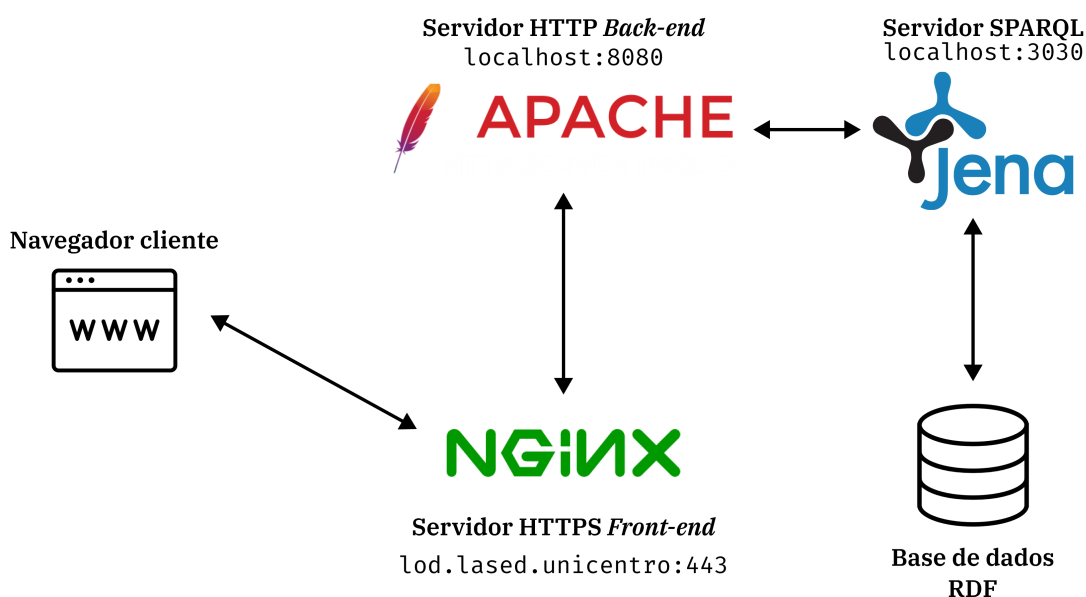
A arquitetura final do sistema implementado é observado na Figura 4. Nela as conexões entre os serviços são representadas através de setas bidirecionais, abstraindo o recebimento e resposta de requisições. Temos o navegador do cliente, que tem sua requisição atendida pelo servidor de *front-end* Nginx; O Nginx por sua vez desvia o fluxo da Requisição para o Apache que trabalha como servidor interno no *back-end*; O apache faz a requisição dos dados para o Jena-Fuseki, que por sua vez realiza uma consulta SPARQL nos dados em RDF armazenados.

6. Considerações Finais

Retomando os objetivos específicos que levam ao cumprimento do objetivo geral levantados na Seção 1, pode-se concluir que todos foram concluídos com sucesso. A realização do levantamento de requisitos foi descrito na Seção 4.1; A averiguação da literatura disponível sobre as funcionalidades do Skosmos como ambiente de publicação de vocabulários controlados foi abordado na Seção 3.2.4; O processo de hospedagem e estudo de caso e publicação estão descritos na Seção 4; Por fim a Documentação dos procedimentos de implantação da interface Skosmos está disponível no Apêndice A.

O Resultado final é a plataforma nomeada Input-LASeD, acrônimo de Interface de Publicação de Tesouros do LASeD está disponível no endereço: <https://lodlased.unicentro.br/Skosmos/pt>. Sua disponibilização, além de permitir a navegação do(s) tesauro(s) do LASeD, permite aos projetistas do Tesauro de Ciência da Computação melhor visualizarem as evoluções do projeto.

Figura 4. Arquitetura do sistema implantado.



Fonte: O Autor

Para a conclusão da hospedagem, foi necessário adequar o processo de instalação para a situação encontrada no ambiente de implantação, e também foi preciso realizar etapas adicionais ou que não estavam contempladas no tutorial de instalação da ferramenta Skosmos. Entre as dificuldades encontradas estão a falta de informações sobre configuração em servidores da ferramenta Skosmos, assim como a ausência de informações detalhadas sobre personalização da ferramenta para conexões com outros serviços.

A interface da plataforma está disponível em Inglês Britânico e Português Brasileiro. Apesar da disponibilidade de língua inglesa, a modelagem do Tesouro Brasileiro de Ciência da Computação impossibilitou a visualização em índice dos termos em inglês, estando disponível apenas em português. Nas Figuras 8 e 9, encontradas no Apêndice C pelo seus tamanhos, é possível ver a tela inicial da plataforma e tela inicial do tesouro.

Com o possível lançamento da versão 3.0 da ferramenta Skosmos, atualmente em desenvolvimento pelos colaboradores, é uma meta interessante para manter o sistema moderno. Outro ponto crucial é a adequação da interface para dispositivos móveis, para garantir a facilidade de acesso e uma melhor experiência do usuário.

A atualização constante do tesouro é outra área de foco. À medida que o grupo de desenvolvimento do LASeD e possíveis colaboradores externos ampliam o tesouro, é essencial que ele seja mantido atualizado para refletir os novos termos e conceitos que surgem. Isso assegurará que o tesouro permaneça relevante e útil para os usuários.

Finalmente, é necessário remodelar o tesouro para que se adeque a uma interface em inglês ou espanhol. Para realizar essa adaptação para outras línguas, os projetistas

devem transformar os rótulos (*labels*) em inglês ou espanhol dos termos, em rótulos preferenciais, em vez de mantê-los como rótulos alternativos, como estão atualmente definidos. A internacionalização do projeto não só ampliará seu alcance, mas também facilitará a colaboração com pesquisadores e desenvolvedores de outros países.

Em suma, os trabalhos futuros se concentram na melhoria contínua do projeto, desde atualização de versões, até a adaptação da interface e atualização do tesouro. Essas ações garantirão que o projeto permaneça relevante a comunidade.

Agradecimentos

A Deus, pela minha vida, por aqueles que estão ao meu redor, e por ter me dado condições para superar todos os desafios encontrados.

À Prof.^a Dra. Gisane Aparecida Michelon, pela sua confiança e seu trabalho como orientadora neste projeto.

Aos meus amigos e familiares que estiveram comigo nos bons e maus momentos.

Ao meu pai, José, e à minha mãe, Sirlei, dedico este trabalho de conclusão de curso. Sem o incentivo e apoio incondicional deles, não seria possível chegar até aqui.

Referências

- [APACHE 2019] APACHE (2019). Faq. Disponível em: <https://cwiki.apache.org/confluence/display/HTTPD/FAQ>. Acesso em: 09 jul. 2024.
- [APACHE 2024] APACHE (2024). Apache jena fuseki. Disponível em: <https://jena.apache.org/documentation/fuseki2/>. Acesso em: 21 jun. 2024.
- [Berners-Lee 2006] Berners-Lee, T. (2006). Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. Acesso em: 05 mai. 2024.
- [Berners-Lee et al. 2005] Berners-Lee, T., Fielding, R. T., and Masinter, L. M. (2005). Uniform Resource Identifier (URI): Generic Syntax. Disponível em: <https://www.rfc-editor.org/info/rfc3986>. Acesso em: 25 mai. 2024.
- [Berners-Lee et al. 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. Disponível em: <https://www.scientificamerican.com/article/the-semantic-web/>. Acesso em: 05 mai. 2024.
- [Costa et al. 2023] Costa, Y. N. A., Michelon, G. A., Dall'agnol, J. M. H., and Souza, L. (2023). Tesouros e interfaces de desenvolvimento para web. Disponível em: <https://evento.unicentro.br/anais/jai>. Acesso em: 13 mai. 2024.
- [Gandon e Schreiber 2016] Gandon, F. and Schreiber, G. (2016). Rdf 1.1 xml syntax. *Gandon, F. and Schreiber, G. (2016) 'RDF 1.1 XML Syntax', W3C Recommendation 25 February 2014, (February), pp. 1–35. W3C Recommendation 25 February 2014.* Disponível em: <http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/>. Acesso em: 17 jun. 2024.
- [GROUP 2024] GROUP, P. D. (2024). Apc user cache. Disponível em: <https://www.php.net/manual/en/book.apcu.php>. Acesso em: 12 jul. 2024.
- [GROUP 2014] GROUP, R. W. (2014). Resource description framework (rdf). Disponível em: <https://www.w3.org/RDF/>. Acesso em: 17 jun. 2024.

- [Gruber 1993] Gruber, T. (1993). What is an ontology. Disponível em: <https://tomgruber.org/writing/definition-of-ontology.pdf>. Acesso em: 17 jul. 2024.
- [Kelniar e Rautenberg 2020] Kelniar, J. P. and Rautenberg, S. (2020). Desenvolvendo um tesouro: um estudo aplicado. Disponível em: https://evento.unicentro.br/files/Submissaoarquivos/car_submissao/20_08_2020_car_submissao_1422328150.pdf. Acesso em: 9 mai. 2024.
- [KNOWLEDGE 2024] KNOWLEDGE, O. (2024). Open definition. Disponível em: <https://opendefinition.org/od/2.1/en/>. Acesso em: 17 jun. 2024.
- [Lonvick e Ylonen 2006] Lonvick, C. M. and Ylonen, T. (2006). The Secure Shell (SSH) Transport Layer Protocol. Disponível em: <https://www.rfc-editor.org/info/rfc4253>. Acesso em: 03 de jul. 2024.
- [Mazzocchi 2019] Mazzocchi, F. (2019). Knowledge organization system (kos). Disponível em: <https://www.isko.org/cyclo/kos>. Acesso em: 12 jun. 2024.
- [MDN 2023] MDN (2023). Proxy servers and tunneling. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTTP/Proxy_servers_and_tunneling. Acesso em: 08 jul. 2024.
- [NGINX 2024] NGINX (2024). About. Disponível em: <https://nginx.org/en/>. Acesso em: 09 jul. 2024.
- [Nielsen et al. 1999] Nielsen, H., Mogul, J., Masinter, L. M., Fielding, R. T., Gettys, J., Leach, P. J., and Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1. Disponível em: <https://www.rfc-editor.org/info/rfc2616>. Acesso em: 21 jun. 2024.
- [NISO 2005] NISO (2005). Ansi/niso z39.19-2005: Guidelines for the construction , format , and management of monolingual controlled vocabularies. Disponível em: <https://www.niso.org/publications/ansiniso-z3919-2005-r2010>. Acesso em: 29 jun. 2024.
- [Ogbuji 2013] Ogbuji, C. (2013). *SPARQL 1.1 Graph Store HTTP Protocol*. W3C. Disponível em: <https://www.w3.org/TR/sparql11-http-rdf-update/>. Acesso em: 17 jul. 2024.
- [OpenSSH 2024] OpenSSH (2024). Project history. Disponível em: <https://www.openssh.com/history.html>. Acesso em: 06 jul. 2024.
- [Rajbhandari e Keizer 2012] Rajbhandari, S. and Keizer, J. (2012). The agrovoc concept scheme – a walkthrough. *Journal of Integrative Agriculture*, 11:694–699. Disponível em: <https://www.fao.org/agrovoc/publications/agrovoc-concept-scheme-walkthrough>. Acesso em: 10 jul. 2024.
- [Rautenberg et al. 2021] Rautenberg, S., de Souza, L., and Kelniar, J. P. (2021). Uma proposta de tesouro da ciência da computação: organização de elementos de conhecimento com skos. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, pages 1–26. Disponível em: <https://periodicos.ufsc.br/index.php/eb/article/view/80386>. Acesso em: 5 mai. 2024.

- [Rautenberg et al. 2018] Rautenberg, S., Souza, L., Dall’Agnol, J. M. H., and Michelon, G. A. (2018). *Guia Prático para Publicação de Dados Abertos Conectados na Web*. Appris, 1 edition.
- [Rescorla 2000] Rescorla, E. (2000). HTTP Over TLS. Disponível em: <https://www.rfc-editor.org/info/rfc2818>. Acesso em: 21 jun. 2024.
- [Silva et al. 2023] Silva, J. V. P., Michelon, G. A., and Souza, L. (2023). A evolução do tesouro de ciência da computação. Disponível em: <https://drive.google.com/file/d/1FiMUAXHDv6ZC4UYTYMeB6H-Wuu1AiiES/view>. Acesso em: 9 mai. 2024.
- [Soergel et al. 2004] Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., and Katz, S. (2004). Reengineering thesauri for new applications: The agrovoc example. *Journal of Digital Information*, 4. Disponível em: <https://www.fao.org/agrovoc/publications/reengineering-thesauri-new-applications-agrovoc-example>. Acesso em: 18 jul. 2024.
- [Sommerville 2019] Sommerville, I. (2019). *Engenharia De Software*. PEARSON BRASIL.
- [Standard 2010] Standard, A. N. (2010). Ansi/niso z39.19-2005 (r2010) guidelines for the construction, format, and management of monolingual controlled vocabularies. Disponível em: <https://www.niso.org/publications/ansiniso-z3919-2005-r2010>. Acesso em: 18 jun. 2024.
- [Subirats-Coll et al. 2022] Subirats-Coll, I., Kolshus, K., Turbati, A., Stellato, A., Mitzsch, E., Martini, D., and Zeng, M. (2022). Agrovoc: The linked data concept hub for food and agriculture. *Computers and Electronics in Agriculture*, 196:105965. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0168169920331707>. Acesso em: 13 jul. 2024.
- [Suominen et al. 2015] Suominen, O., Ylikotila, H., Pessala, S., Lappalainen, M., Frosterus, M., Tuominen, J., Baker, T., Caracciolo, C., and Retterath, A. (2015). Publishing skos vocabularies with skosmos. *Manuscript submitted for review*, pages 1–24. Disponível em: <https://skosmos.org/publishing-skos-vocabularies-with-skosmos.pdf>. Acesso em: 28 jun. 2024.
- [Tanenbaum e Bos 2024] Tanenbaum, A. and Bos, H. (2024). *Sistemas operacionais modernos*. Bookman, 5 edition.
- [Tanenbaum et al. 2021] Tanenbaum, A., Feamster, N., and Wetherall, D. J. (2021). *Redes de computadores*. Editora Bookman.
- [W3C 2012] W3C (2012). Introduction to skos. Disponível em: <https://www.w3.org/2004/02/skos/intro>. Acesso em: 02 jul. 2024.
- [W3C 2014] W3C (2014). Rdf 1.1 turtle. Disponível em: <https://www.w3.org/TR/turtle/>. Acesso em: 06 jul. 2024.
- [Weiderman e Kamenoff 1992] Weiderman, N. H. and Kamenoff, N. I. (1992). Hartsone uniprocessor benchmark: Definitions and experiments for real-time systems.

Real-Time Systems, 4:353–382. Disponível em: <https://doi.org/10.1007/BF00355299>. Acesso em: 16 jul. 2024.

[Zeng e Mayr 2019] Zeng, M. L. and Mayr, P. (2019). Knowledge organization systems (kos) in the semantic web: a multi-dimensional review. *International Journal on Digital Libraries*, 20. Disponível em: <https://doi.org/10.1007/s00799-018-0241-2> Acesso em: 04 jul. 2024.

Apêndice A

A documentação do processo de instalação está disponível em: https://yorranan.com/Documentacao_INPUT_LASeD.html.

Apêndice B

Os resultados dos teste realizados usando a ferramenta Apache Bench são descritos em tabelas. Ao todo foram realizadas 30 rodadas de testes, 10 rodadas para cada ferramenta e 10 sem nenhuma ferramenta de *cache*. Os resultados sem *cache* estão descritos na Figura 5. Com a ferramenta APC na Figura 6. Com a ferramenta Varnish na Figura 7.

Figura 5. Resultados das rodadas de benchmark sem cache

Rodada	Sem ferramenta de <i>cache</i>				
	Tempo de conexão (ms)				
	Minímimo	Média	Desvio padrão	Mediana	Máximo
1	60	166	29.1	164	350
2	61	166	27.60	162	285
3	59	165	28.9	163	281
4	57	166	28.0	163	288
5	65	166	27.2	164	296
6	58	166	27.1	164	268
7	56	167	28.7	166	270
8	65	166	28.7	162	288
9	60	165	27.9	162	268
10	58	167	33.3	164	448

Fonte: O Autor

Figura 6. Resultados das rodadas de benchmark com APC cache

Rodada	Usando APC				
	Tempo de conexão (ms)				
	Minímimo	Média	Desvio padrão	Mediana	Máximo
1	83	164	30.2	161	463
2	62	163	27.8	160	267
3	66	162	28.3	159	270
4	61	165	28.4	162	272
5	57	164	27.8	161	310
6	64	164	26.6	162	255
7	61	164	28.2	160	277
8	56	164	26.0	161	250
9	56	163	28.4	161	267
10	62	164	27.6	163	260

Fonte: O Autor

Figura 7. Resultados das rodadas de *benchmark* com Varnish cache

Usando Varnish					
Rodada	Tempo de conexão (ms)				
	Minímimo	Média	Desvio padrão	Mediana	Máximo
1	99	221	56.4	210	437
2	65	217	56.8	204	420
3	68	217	57.8	207	436
4	61	217	59.2	204	457
5	67	219	58.4	208	487
6	81	218	58.4	205	410
7	58	219	64.0	205	516
8	61	219	61.8	205	513
9	67	217	56.7	203	428
10	67	217	58.0	202	431

Fonte: O Autor

Apêndice C

Nas Figuras 8 e 9 é apresentada a tela inicial do Tesouro Brasileiro de Ciência da Computação e da Interface Implantada respectivamente.

Figura 8. Página do Tesouro Brasileiro de Ciência da Computação

Input-LASeD Vocabulários Sobre Feedback Ajuda | in_this_language

Tesouro Brasileiro de Ciência da Computação Idioma do conteúdo português Busca

Alfabético Hierarquia Grupos

A B C D E É F G H I L M

P R S T 0-9

A teoria da computação é um subcampo da ciência da computação e matemática que busca determinar quais problemas podem ser computados em um dado modelo de computação. (Wikipédia). → Teoria da Computação

Algoritmos
Arquitetura e Organização de Computadores
Arquiteturas Paralelas de Computadores
Automação
Autorregulação da Aprendizagem
Avaliação de Desempenho

Informação sobre o vocabulário

TÍTULO Tesouro Brasileiro de Ciência da Computação

TIPO <http://www.w3.org/2002/07/owl#NamedIndividual>
<http://www.w3.org/2004/02/skos/core#ConceptScheme>

TERMOS ALTERNATIVOS Computer Science
Ciencia de la computación

CONCEITO MAIS AMPLO <http://lod.unicentro.br/2017/DiretrizesSBC#ce4>

DEFINIÇÃO Ciência da computação é a ciência que estuda as técnicas, metodologias e instrumentos computacionais, que automatiza processos e desenvolve soluções baseadas no uso do processamento de dados. (Wikipédia)

ESQUEMA DO CONCEITO Ciência da Computação

CONCEITOS MAIS RESTRITOS http://lod.unicentro.br/2017/DiretrizesSBC#C.7.5-Compreender_os_fatos_essenciais_os_conceitos_os_principios_e_as_teorias_relacionadas_a_ciencia_da_computacao_para_o_desenvolvimento_de_software_e_hardware_e_suas_aplicacoesii

URI <http://lod.unicentro.br/2017/DiretrizesSBC#CC>

Número de recursos por tipo

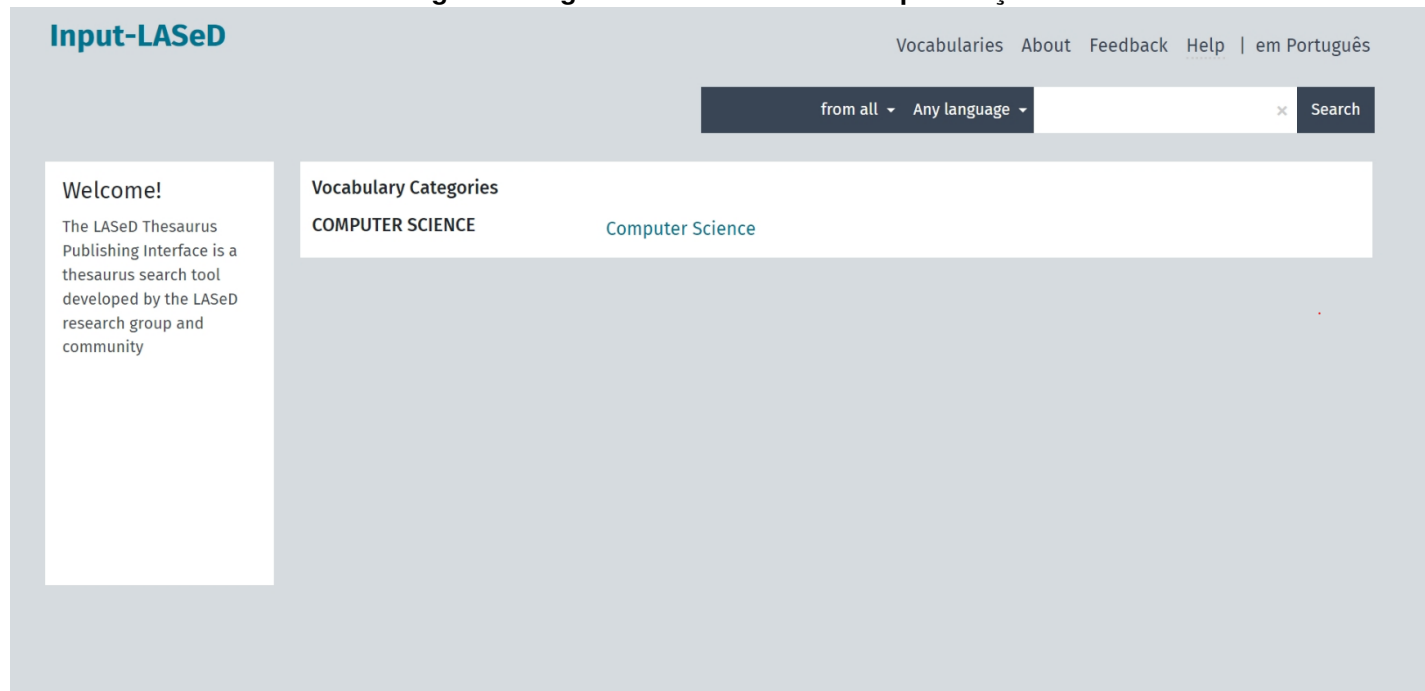
Tipo	Quantidade
Conceito	185
• Conceito obsoleto	0

Número de termos por idioma

Idioma	Termos preferidos	Termos alternativos	Termos ocultos
português	184	2	0

Fonte: O Autor

Figura 9. Página inicial da interface de publicação.



Fonte: O Autor

Apêndice D

Nos Quadros 2 e 3 é apresentado os arquivos de configurações dos servidores HTTP.

```
1     <VirtualHost *:8080>
2
3     <Directory /var/www/html>
4         Options Indexes FollowSymLinks MultiViews
5         AllowOverride All
6         Order allow,deny
7         allow from all
8     </Directory>
9     ServerAdmin webmaster@localhost
10    DocumentRoot /var/www/html/
11
```

Quadro 2. Arquivo de configuração do servidor Apache

```
1 server_name lodlased.unicentro.br;
2
3     location /Skosmos {
4         proxy_pass http://localhost:8080/;
5         [...]
6         proxy_redirect http://localhost:8080/Skosmos/ /Skosmos
7     /;
8     }
```

Quadro 3. Arquivo de configuração de redirecionamento do Nginx