

Uma API para Gerenciamento de Objetos de Aprendizagem Digitais Baseada em Web Semântica

Rafael Silva da Silva¹, João Pablo Silva da Silva¹, Alice Fonseca Finger¹

¹Laboratory of Intelligent Software Engineering (LabISE)
Campus Alegrete – Universidade Federal do Pampa (UNIPAMPA)
Av. Tiarajú, 810 – 97546-550 – Alegrete – RS – Brasil

{rafaelsds3.aluno, joaosilva, alicefinger}@unipampa.edu.br

Abstract. *Digital Learning Objects (DLOs) support teaching in different contexts, but still face a scarcity of open repositories and access difficulties. Semantic Web emerges as an alternative to standardize and organize these repositories. In previous work, we developed the OntoObADi, an ontology aimed at describing DLOs in semantic repositories. In this work, we propose a RESTful service that encapsulates OntoObADi, promoting the interoperability and content reuse. The API was implemented in Python with the Owlready2 library, allowing for manipulation of ontologies in OWL. With it, it is possible to create and classify individuals, apply SWRL rules, and integrate reasoners for logical inference.*

Resumo. *Os Objetos de Aprendizagem Digitais (OADs) apoiam o ensino em diferentes contextos, mas ainda enfrentam escassez de repositórios abertos e dificuldades de acesso. A Web Semântica surge como alternativa para padronizar e organizar esses repositórios. Em um trabalho anterior, desenvolvemos a OntoObADi, uma ontologia voltada à descrição de OADs em repositórios semânticos. Neste trabalho, propomos um serviço RESTful que encapsula a OntoObADi, promovendo a interoperabilidade e reutilização dos conteúdos. A API foi implementada em Python com a biblioteca Owlready2, permitindo manipular ontologias em OWL. Com ela, é possível criar e classificar indivíduos, aplicar regras em SWRL e integrar raciocinadores para inferência lógica.*

1. Introdução

A crescente digitalização da educação tem impulsionado a produção de conteúdos educacionais em diferentes formatos, tais como: textos, vídeos, animações, softwares, etc. Nesse contexto, os Objetos de Aprendizagem Digitais (OADs) ganham relevância como unidades educacionais reutilizáveis, modularizadas e voltadas ao suporte de estratégias pedagógicas flexíveis [Wiley 2000]. Sua estrutura permite a fragmentação de conteúdos maiores em partes menores, facilitando sua reutilização em diferentes contextos educacionais [Braga 2015]. Os OADs atuam como facilitadores do ensino e da aprendizagem, apresentando objetivos pedagógicos claros, conteúdo autocontido e instrucional, interatividade e *feedback* durante a sua utilização [Carneiro and Silveira 2014, Aguiar and Flôres 2014]. Quando enriquecidos com metadados estruturados e descrições semânticas, eles também promovem a interoperabilidade entre diferentes sistemas e plataformas educacionais, favorecendo a integração de conteúdos em cenários de aprendizagem distribuída [de S. Santana et al. 2023].

A ausência de raciocinadores que interpretem semanticamente os dados limita o potencial de funcionalidades como a classificação automatizada de conteúdos, a recomendação adaptativa e a análise de trajetórias de aprendizagem. Isso dificulta o aproveitamento pleno dos objetos de aprendizagem como instrumentos dinâmicos de apoio ao ensino personalizado [Schrammel and Fontoura 2023]. Nesse sentido, a Web Semântica desponta como uma abordagem tecnológica promissora, aprimorando a forma de descrever o conteúdo disponível na Web, por meio de modelos que possibilitam a organização, categorização e padronização das informações [Berners-Lee et al. 2001]. Seu propósito primordial é dotar os dados disponíveis na Web de um significado explícito e compreensível por sistemas computacionais. Isso é alcançado por meio da criação de uma “Web de Dados” interligados e estruturados, onde as máquinas são capazes de realizar inferências lógicas, extrair conhecimento de forma automatizada e estabelecer relações complexas entre diferentes fontes de dados [Breitman 2005, Segundo and Coneglian 2016].

Sistemas baseados em Web Semântica necessitam de um modelo que descreva as entidades do mundo real e seus relacionamentos, observados os domínios do problema ou da solução. As ontologias, especificações formais e explícitas de conceitualizações compartilhadas [Gruber 1993, Borst 1997], atendem a essa necessidade, pois atuam como vocabulários com semântica pré-definida, permitindo que recursos sejam descritos de forma compreensível para computadores, viabilizando a organização, categorização e padronização das informações na Web [Berners-Lee et al. 2001]. Essa capacidade de processamento semântico permite a criação de aplicações mais inteligentes, sistemas de busca aprimorados e a integração de dados heterogêneos, viabilizando uma internet mais inteligente e interconectada, onde a interoperabilidade e a reutilização de dados são maximizadas [Berners-Lee et al. 2001, Breitman 2005, Segundo and Coneglian 2016]. Em um trabalho anterior, desenvolvemos a OntoObADi, uma ontologia de aplicação que formaliza os conceitos e relações necessárias para viabilizar a implementação de um repositório Web Semântico de OADs [Forrati 2024].

Neste trabalho propomos uma *Application Programming Interface* (API) Web RESTful que encapsula a OntoObADi, visando ampliar a interoperabilidade e a reutilização de conteúdos educacionais. A API foi desenvolvida em Python, utilizando a biblioteca Owlready2, que permite a manipulação de ontologias em *Web Ontology Language* (OWL). Com essa API, é possível criar e classificar indivíduos, aplicar regras semânticas escritas em *Semantic Web Rule Language* (SWRL) e integrar raciocinadores para realizar inferências lógicas. Esperamos com este trabalho promover: uma maior interoperabilidade entre repositórios; a reutilização facilitada de conteúdos; buscas mais eficientes e precisas por OADs; o desenvolvimento de aplicações educacionais inteligentes; uma validação prática da OntoObADi; e a adoção da Web Semântica na educação.

O restante deste trabalho está organizado como segue: na Seção 2 reportamos estudos que aplicam a Web Semântica no âmbito da educação digital; na Seção 3 explicamos o modelo conceitual da ontologia OntoObADi; na Seção 4 detalhamos o desenvolvimento da API Web RESTful; na Seção 5 apresentamos um exemplo de uso do serviço desenvolvido; por fim, na Seção 6 apresentamos as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Diversos estudos têm se dedicado ao gerenciamento e à recomendação de Objetos de Aprendizagem Digitais (OADs), explorando desde soluções práticas de armazenamento até arquiteturas baseadas em serviços inteligentes. Nesse cenário, alguns trabalhos concentram-se na gestão e interoperabilidade: por exemplo, [Costa 2021] e [Darmadi et al. 2018] desenvolveram APIs RESTful para repositórios educacionais, contemplando etapas essenciais como submissão, curadoria e disponibilização de OADs. Embora robustas, essas iniciativas priorizam a organização e o acesso aos recursos, deixando em segundo plano aspectos relacionados à recomendação personalizada e ao uso de inferências semânticas mais sofisticadas.

Outros estudos avançam na incorporação de semântica e serviços inteligentes. O trabalho de [Rabahallah and Ahmed-Ouamer 2015], por exemplo, propõe uma arquitetura de serviços Web semânticos em sistemas de e-learning, reutilizando funcionalidades como autenticação, gerenciamento de cursos e avaliações. A adoção de descrições ontológicas viabiliza descoberta e composição automática de serviços, favorecendo escalabilidade e flexibilidade. De forma complementar, [Júnior et al. 2022] apresenta uma abordagem híbrida para recomendação personalizada de OADs, combinando algoritmos bioinspirados e técnicas de Web Semântica. A solução integra diferentes fontes (como AVAs, YouTube e Wikipédia), representando metadados por meio de ontologias e promovendo recomendações alinhadas ao perfil do estudante. Apesar da relevância, a avaliação empírica não demonstrou diferenças estatisticamente significativas entre as estratégias de recomendação testadas.

Assim, a literatura revela duas grandes vertentes: a primeira voltada ao gerenciamento e à interoperabilidade de repositórios; a segunda, ao uso de semântica para recomendação e serviços inteligentes. Entretanto, nota-se uma lacuna na integração dessas abordagens, uma vez que muitas propostas permanecem conceituais ou limitadas a aspectos específicos. Nesse sentido, o presente trabalho busca avançar ao combinar a modelagem semântica da OntoObADI com a implementação de uma API dedicada ao gerenciamento de OADs. Além de assegurar a interoperabilidade e a reutilização, pretende-se validar na prática a eficácia da integração entre ontologias, inferências automáticas por meio de regras lógicas e serviços Web.

3. Modelo Conceitual da OntoObADI

A Ontologia de Objetos de Aprendizagem Digitais (OntoObADI) é uma ontologia de aplicação voltada à formalização de conceitos e relações para um Repositório Web Semântico de OADs. Ela foi desenvolvida aplicando o método *Ontology Development 101* [Noy et al. 2001] e especifica em OWL. Apresentamos na Figura 1 o modelo conceitual da OntoObADI.

No topo da Figura 1 temos o conceito *Coisa*, uma classe nativa da OWL, que representa o conceito mais genérico e fundamental da ontologia. Ela possui uma propriedade chamada *nome*, atribuída como obrigatória. Todas as subclasses diretamente associadas ou derivadas de *Coisa* herdam essa propriedade. No centro da Figura 1 está a classe *Objeto de Aprendizagem Digital*, contendo propriedades como *conteúdo gerado por IA*, *palavra-chave*, *requisito de hardware e software*, *Descrição*, *data de criação*, *objetivo*

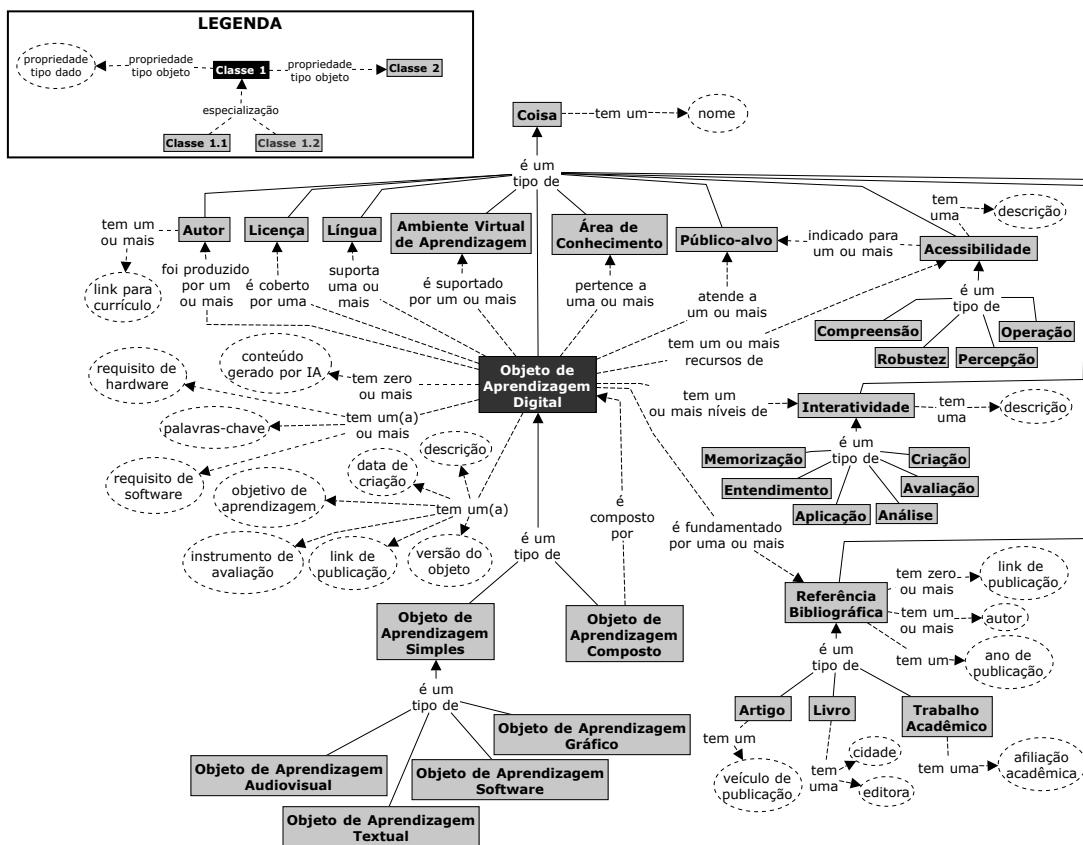


Figura 1. Modelo Conceitual da OntoObADI.

de aprendizagem, instrumento de avaliação, link de publicação e versão do objeto, todas com restrições específicas. A classe possui subclasses especializadas: Objeto de Aprendizagem Simples, subdividida em Audiovisual, Textual, Software e Gráfico, e Objeto de Aprendizagem Composto, que permite hierarquias do tipo “parte-todo”, seguindo o padrão de projeto *Composite*.

No canto superior esquerdo da Figura 1 está a classe Autor, que identifica a autoria do OAD por meio da propriedade link para currículo. Em seguida, a classe Licença, define o tipo de licença associada ao OAD. A classe Língua, representa o idioma utilizado no objeto. Além dessas, estão representadas as classes Ambiente Virtual de Aprendizagem, descrevendo plataformas suportadas; Área de Conhecimento, orientada pela tabela de áreas de conhecimento da *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES); e Público-alvo, identificando os diferentes grupos atendidos pelo OAD. No canto superior direito da Figura 1 está representada a classe Acessibilidade. Ela representa os tipos de acessibilidade oferecidos pelo OAD, visando garantir acesso e interação a um público diversificado. A classe Acessibilidade possui uma propriedade do tipo dado denominada descrição, que permite detalhar as características de acessibilidade implementadas. Além disso, ela inclui as seguintes subclasses: Compreensão, Robustez, Percepção e Operação, cada uma representando dimensões específicas das diretrizes de acessibilidade.

Logo abaixo da classe Acessibilidade, temos a classe Interatividade. Ela alinha as atividades e conteúdos interativos do OAD com os diferentes níveis de processos cognitivos envolvidos. Ela possui uma propriedade do tipo dado denominada descrição, e é especializada por subclasses que representam os níveis cognitivos: Memorização, Entendimento, Aplicação, Análise, Avaliação e Criação. Por fim, no canto inferior direito da Figura 1, está a classe Referência Bibliográfica, que descreve o embasamento utilizado pelo autor na construção do OAD. A mesma possui propriedades do tipo dado, como autor, link de publicação e ano de publicação. Além disso, Referência Bibliográfica é especializada em subclasses que abrangem diferentes tipos de referência, cada uma com propriedades do tipo dado específicas: Artigo, veículo de publicação; Livro, editora e cidade; Trabalho Acadêmico, afiliação acadêmica.

4. Desenvolvimento da API Web RESTful

A aplicação desenvolvida consiste de uma API Web RESTful voltada à manipulação dinâmica do OWL da OntoObADI, com suporte à criação, edição e classificação de indivíduos, além da execução e criação de regras semânticas SWRL. Seu desenvolvimento teve como objetivo facilitar a integração de sistemas externos com bases de conhecimento formais, promovendo a interoperabilidade e a inferência automática a partir de regras e axiomas ontológicos. O serviço foi desenvolvido em Python 3.13, utilizando a biblioteca Owlready2. [Eremeev et al. 2022] Permitindo leitura, escrita, modificação, raciocínio e também uma abordagem orientada a objetos sobre arquivos OWL. Assim facilitando o acesso às entidades de uma ontologia [Lamy 2017].

Para viabilizar a interoperabilidade com aplicações externas e permitir o consumo dinâmico dos dados ontológicos, adotamos o framework FastAPI. Sua utilização permitiu a criação de uma interface RESTful de alto desempenho, garantindo que sistemas clientes pudessem acessar, manipular e consultar a ontologia de forma padronizada.

Estruturada para os princípios da arquitetura *Representational State Transfer* (REST), um estilo arquitetural amplamente adotado em sistemas distribuídos pela sua simplicidade, escalabilidade e compatibilidade com os protocolos da Web [Xiao-Hong 2014], tal abordagem permite que operações sejam realizadas por meio dos métodos do protocolo *Hypertext Transfer Protocol* (HTTP), facilitando que aplicações externas interajam com a OntoObADI de forma intuitiva e padronizada, utilizando formatos de dados *JavaScript Object Notation* (JSON) para entrada e saída, conforme ilustrado na Figura 2.

A API foi estruturada com base em uma arquitetura semelhante à em camadas, visando garantir uma organização mais clara do código, bem como promover a modularidade e a separação de responsabilidades entre os componentes da aplicação. Essa abordagem contribui para a escalabilidade e a manutenibilidade do sistema, além de favorecer a reutilização de código e a realização de testes de forma mais eficiente.

A camada Router é responsável por definir os *endpoints* acessíveis por meio dos métodos HTTP (GET, POST, PUT e DELETE). Essa camada atua como a porta de entrada da aplicação, permitindo que sistemas externos interajam com os recursos expostos de forma padronizada, por meio de requisições REST. Cada rota é mapeada diretamente para funções que executam operações específicas sobre a ontologia, como a criação de

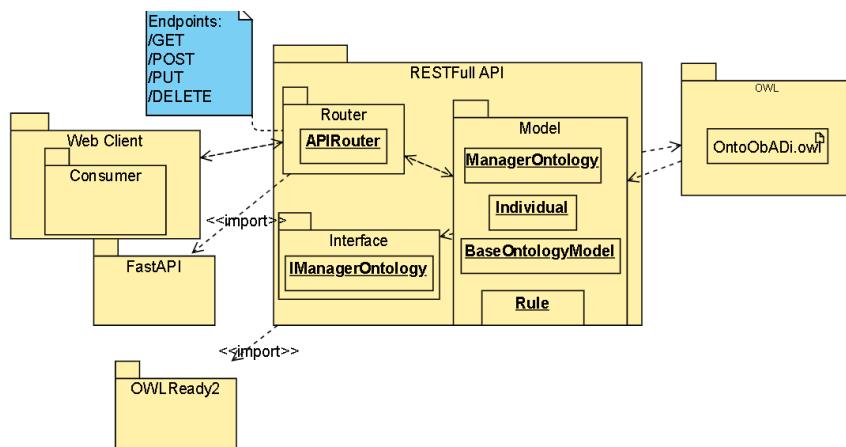


Figura 2. Diagrama de pacotes REST do serviço de manipulação da OntoObADI.

indivíduos, a consulta a classes e a execução de regras SWRL.

A camada Interface tem como principal objetivo abstrair o acesso à OntoObADI, promovendo o desacoplamento entre os componentes de alto nível da aplicação e a implementação concreta de carregamento, manipulação e persistência dos dados ontológicos. Essa camada define um contrato por meio de uma interface, IOntologyManager, que estabelece os métodos essenciais para interagir com a ontologia, como carregar e salvar arquivos OWL, recuperar classes, propriedades e regras semânticas. Esse modelo de abstração permite substituir ou estender a fonte da ontologia (por exemplo, de arquivos locais para APIs remotas) sem impactar os módulos consumidores.

Por fim, a camada Model concentra as entidades centrais do domínio do problema. Nela são definidas as classes e estruturas de dados que refletem diretamente os conceitos da ontologia, como Individual, BaseOntologyModel, Rule, entre outras. Essa organização contribui para um desenvolvimento mais limpo e modular, permitindo que cada componente da aplicação possa ser evoluído, testado e mantido de forma independente, além de facilitar a integração com outras plataformas e serviços. Essas entidades e suas respectivas responsabilidades podem ser visualizadas no diagrama de classes ilustrado na Figura 3.

A classe ManagerOntology atua como componente central da infraestrutura da aplicação, implementando a interface IOntologyManager. Sua principal responsabilidade é carregar a OntoObADI a partir de um arquivo OWL, disponibilizando acesso estruturado às classes, propriedades e regras, além de permitir a execução de inferência semântica utilizando o raciocinador Pellet. Todas as demais classes interagem com a ontologia por meio dessa abstração, garantindo flexibilidade e desacoplamento em relação à biblioteca owlready2. A classe BaseOntologyModel oferece uma abstração orientada a metaclasses OWL, permitindo a exploração dinâmica da estrutura de uma classe ontológica, incluindo suas propriedades do tipo DataProperty e ObjectProperty. Também fornece rótulos amigáveis em português para facilitar a construção de interfaces dinâmicas. Adicionalmente, permite a criação de instâncias genéricas com base em dicionários de dados, sendo particularmente útil para sistemas que geram formulários ou visualizações a partir da OntoObADI.

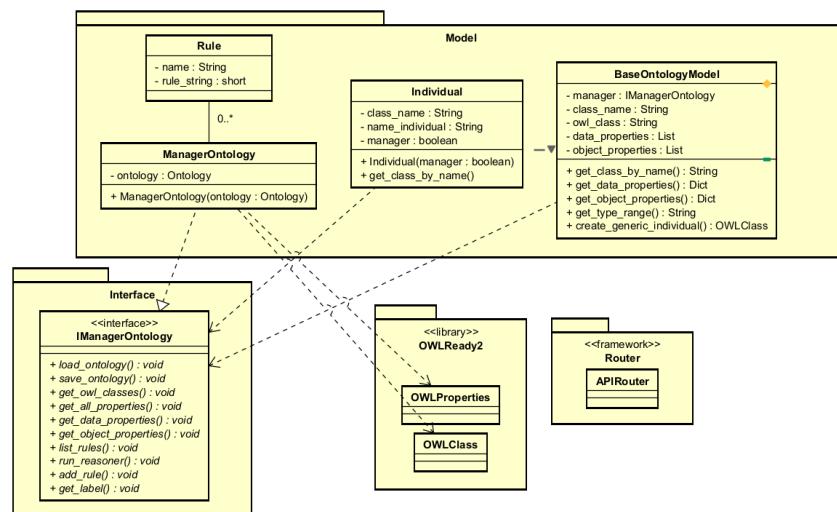


Figura 3. Diagrama de classe da API Web RESTFul.

A classe Individual é responsável pela criação e manipulação direta de instâncias da ontologia. A partir do nome da classe OWL e do identificador do indivíduo, ela instancia o objeto OWL correspondente e permite a atribuição de propriedades individualmente, mantendo uma referência direta ao objeto criado. Essa abordagem é ideal para cenários em que se requer maior controle na construção de indivíduos. A classe Rule representa uma entidade semântica do tipo regra, sendo responsável por armazenar uma estrutura lógica no formato SWRL, juntamente com um identificador textual. Essa classe encapsula os dados essenciais para a definição de regras, permitindo que elas sejam posteriormente processadas, adicionadas ou removidas da ontologia.

5. Exemplos de Uso

A abstração proporcionada pela classe BaseModelOntology permite a geração automática de interfaces dinâmicas que refletem a estrutura da ontologia subjacente. Essa funcionalidade é acessada através do endpoint GET/ontologySchema, que retorna um JSON estruturado contendo conceitos da ontologia, propriedades e relações, como mostra a Figura 4. Esse esquema dinâmico serve como base para a construção automática de formulários, permitindo que sistemas clientes criem interfaces de cadastro e edição de conteúdo educacional de forma padronizada.

```

Request URL
http://127.0.0.1:8081/ontologySchema

Server response
Code Details
200 Response body
{
    "Author": {
        "title": "Author",
        "type": "object",
        "properties": {
            "hasResumelink": {
                "type": "string",
                "title": "Um link para currículo"
            }
        },
        "required": [
            "hasResumelink"
        ],
        "CompositeLearningObject": {
            "type": "object",
            "properties": {
                "hasResumelink": {
                    "type": "string",
                    "title": "Um link para currículo"
                }
            }
        }
    }
}
  
```

Figura 4. Retorno do endpoint GET/ontologySchema.

Outro cenário relevante é a classificação automática de OADs. A API instancia dinamicamente indivíduos na OntoObADI, por meio da classe `Individual` e do endpoint `POST/createindividual`, que espera parâmetros claros: o campo `name`, que indica o nome do indivíduo a ser criado; e dois blocos de propriedades: `data_properties`, destinado a valores literais, e `object_properties`, que referencia outros indivíduos já existentes ou os cria quando não existem, conforme ilustrado na Figura 5. Nela, um objeto é criado e classificado automaticamente com base na estrutura da ontologia e na execução do raciocinador, sendo assim uma forma de aprimorar a organização e permitir uma personalização pedagógica mais eficiente.



The screenshot shows a REST API interface. The 'Request URL' field contains `http://127.0.0.1:8081/createIndividual`. The 'Server response' section has tabs for 'Code' (selected) and 'Details'. Under 'Code', it shows '201 Undocumented'. Under 'Details', the 'Response body' is a JSON object:

```
{  
  "message": "Indiv\u00edduo 'teste_de_individuo_classificado' criado com sucesso!",  
  "classe": "Thing",  
  "propriedades": [  
    "hasResumelink"  
  ],  
  "inferencias": [  
    " Inferido: 'string' \u00e9 uma inst\u00e1ncia da classe 'Author'.",  
    " Inferido: 'teste_de_individuo_classificado' \u00e9 uma inst\u00e1ncia da classe 'Author'.",  
    " Inferido: 'teste_de_individuo_classificado' possui atributo 'hasResumelink' com valor 'http://lattes.cnpq.br/2691501072064698'  
  ]  
}
```

Figura 5. Retorno do endpoint `POST/createIndividual`.

Além disso, a adoção de uma arquitetura REST e o uso de dados em JSON facilitam a integração com outros sistemas, como repositórios educacionais, mecanismos de busca e ambientes de ensino. Essa interoperabilidade promove o compartilhamento, a reutilização e a descoberta de OADs com base em critérios semânticos, ampliando sua acessibilidade e aplicabilidade.

6. Conclusão

Apresentamos neste trabalho o desenvolvimento inicial e resultados preliminares de uma API Web RESTful baseada na OntoObADI, com o intuito de promover a interoperabilidade, reutilização e inferência semântica sobre OADs. Por meio do encapsulamento da OntoObADI, a API permite a criação e classificação automatizada de indivíduos, a aplicação de regras SWRL e a integração com raciocinadores semânticos. Os exemplos de uso indicam que a API é capaz de apoiar desde a construção dinâmica de interfaces educacionais até a recomendação personalizada de conteúdos, evidenciando seu potencial em contribuir com o avanço de soluções pedagógicas mais inteligentes e centradas no aluno.

Como trabalho futuro, propomos a criação de um repositório educacional baseado em Web Semântica, voltado à classificação e recomendação de objetos de aprendizagem digitais. Acredita-se que essa iniciativa contribuirá significativamente para a organização e o compartilhamento de conteúdos educacionais, potencializando a reutilização e a interoperabilidade entre diferentes plataformas. Com este trabalho, espera-se promover buscas mais eficientes e precisas por OADs, apoiar o desenvolvimento de aplicações educacionais inteligentes, validar a aplicação prática da OntoObADI e incentivar a adoção de tecnologias semânticas no contexto educacional.

Agradecimentos

Os autores agradecem à FAPERGS pelo apoio financeiro (Projeto ARD/ARC - processo 23/2551-0000761-4) e ao CNPq pela concessão de bolsa de Iniciação Científica.

Referências

- Aguiar, E. V. B. and Flôres, M. L. P. (2014). *Objetos de Aprendizagem: Conceitos Básicos*, chapter 1. Evangraf. Recurso online. Disponível em: <https://www.lume.ufrrgs.br/bitstream/handle/10183/102993/000937201.pdf>. Acesso em: 12 set. 2024.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):35–43. Disponível em: <https://www.lassila.org/publications/2001/SciAm.html>. Acesso em: 01 set. 2024.
- Borst, W. N. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. PhD thesis, University of Twente, Enschede, Netherlands. Recurso online. Disponível em: <https://research.utwente.nl/files/6036649/t0000004.pdf>. Acesso em: 14 maio 2023.
- Braga, J. (2015). Objetos de aprendizagem volume 1: introdução e fundamentos. Recurso online.
- Breitman, K. K. (2005). *Web semântica*. LTC, Rio de Janeiro. recurso online. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/978-85-216-1958-1>. Acesso em: 25 maio 2023.
- Carneiro, M. L. F. and Silveira, M. S. (2014). Objetos de aprendizagem como elementos facilitadores na educação a distância. *Educar em Revista*, pages 235–260.
- Costa, A. C. (2021). Re-mar: Repository of marine learning objects. In *Anais do XII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais (WCAMA 2021)*.
- Darmadi, H., Liawatimena, S., Abbas, B. S., and Trisetyarso, A. (2018). Hypermedia driven application programming interface for learning object management. *Procedia Computer Science*, 135:120–127. The 3rd International Conference on Computer Science and Computational Intelligence (ICCS 2018) : Empowering Smart Technology in Digital Era for a Better Life.
- de S. Santana, J., de A. Gomes, E., do N. Salvador, L., and da R. Seixas, L. (2023). Lomc: Uma extensão ao lom para a descrição de recursos educacionais baseada no desenvolvimento de competências. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 345–356. Sociedade Brasileira de Computação.
- Eremeev, A. P., Paniavin, N. A., and Marenkov, M. A. (2022). An object-oriented approach to ontology modelling in specialists education of methods and technologies of artificial intelligence. In *2022 VI International Conference on Information Technologies in Engineering Education (Inforino)*, pages 1–4.
- Forrati, S. M. (2024). Ontoobadi: Uma ontologia para repositórios de objetos de aprendizagem digitais. Master's thesis, Universidade Federal do Pampa (Unipampa).

- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.
- Júnior, C. P., Araújo, R., and Dorça, F. (2022). Uma abordagem híbrida apoiada por algoritmo bioinspirado e tecnologias de web semântica para recomendação personalizada de objetos de aprendizagem. In *Anais Estendidos do XI Congresso Brasileiro de Informática na Educação*, pages 35–46, Porto Alegre, RS, Brasil. SBC.
- Lamy, J.-B. (2017). Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, 80:11–28.
- Noy, N. F., McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology.
- Rabahallah, K. and Ahmed-Ouamer, R. (2015). Creating e-learning web services towards reusability of functionalities in creating e-learning systems. In *2015 Global Summit on Computer Information Technology (GSCIT)*, pages 1–6.
- Schrammel, L. and Fontoura, L. (2023). Recomendação híbrida de objetos de aprendizagem a partir da predição de necessidades personalizadas de estudantes. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 139–151, Porto Alegre, RS, Brasil. SBC.
- Segundo, J. E. S. and Conegiani, C. S. (2016). Web semântica e ontologias: um estudo sobre construção de axiomas e uso de inferências. *Informação & Informação*, 21(2):217–244.
- Wiley, D. A. (2000). *Learning object design and sequencing theory*. PhD thesis, Brigham Young University.
- Xiao-Hong, L. (2014). Research and development of web of things system based on rest architecture. In *2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, pages 744–747.