

## XXV ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO - XXV ENANCIB

### GT 8 – Dados, Informação e Tecnologia

#### ARQUITETURA ORIENTADA POR ONTOLOGIAS PARA INTERFACES ADAPTATIVAS

##### *AN ONTOLOGY-DRIVEN ARCHITECTURE FOR ADAPTIVE INTERFACES*

**Fernanda Farinelli** – Universidade de Brasília (UnB)

**João Pedro Sousa Nunes** – Universidade Federal de Uberlândia (UFU)/Instituto  
Brasileiro de Informação em Ciência e Tecnologia (IBICT)

#### **Modalidade: Trabalho Completo**

**Resumo:** este trabalho investiga como a representação formal do conhecimento pode contribuir para o desenvolvimento de interfaces adaptativas mais precisas e semanticamente coerentes. O objetivo é propor e validar uma arquitetura de interface adaptativa orientada por ontologias, visando melhorar a inserção de dados em sistemas de informação. A metodologia inclui pesquisa bibliográfica, aplicação do método Design Science Research e desenvolvimento de uma prova de conceito utilizando a linguagem OWL, a biblioteca RDFLib e o framework Flask. Os resultados demonstram que a geração automatizada de formulários com base em ontologias melhora a consistência dos dados inseridos, evita ambiguidade e reduz o esforço de desenvolvimento. Conclui-se que a utilização de ontologias na geração de interfaces promove maior precisão na coleta de dados, facilita sua reutilização e contribui para a integridade e usabilidade em sistemas de informação.

**Palavras-chave:** interfaces adaptativas; ontologias; sistemas orientados por ontologias; representação do conhecimento; arquitetura de sistemas.

**Abstract:** this paper investigates how the formal representation of knowledge can contribute to the development of more precise and semantically coherent adaptive interfaces. The objective is to propose and validate an ontology-driven adaptive interface architecture, aiming to improve data insertion in information systems. The methodology includes bibliographic research, application of the Design Science Research method, and development of proof of concept using the OWL language, RDFLib library, and Flask framework. The results show that the automatic generation of forms based on ontologies improves data consistency, avoids ambiguity, and reduces development effort. It is concluded that the use of ontologies in interface generation promotes greater accuracy in data collection, facilitates its reuse, and contributes to data integrity and usability in information systems.

**Keywords:** adaptive interfaces; ontologies; ontology-driven systems; knowledge representation; systems architecture.

## 1 INTRODUÇÃO

É fundamental evitar inconsistências semânticas na inserção de dados, pois a presença de contradições nos registros pode comprometer a qualidade da informação e levar à tomada

de decisões incorretas (Batini; Scannapieco, 2009). A ausência de padronização conceitual e a ambiguidade nos formulários de entrada dificultam a compreensão dos dados, comprometem sua integridade e impactam negativamente a recuperação e o reuso das informações.

Nesse contexto, cresce o interesse por soluções que promovam maior alinhamento semântico entre os dados coletados e os conceitos representados nos sistemas. As ontologias, enquanto sistemas formais de organização do conhecimento, oferecem mecanismos para estruturar entidades, relações e restrições de forma lógica e explícita (Almeida, 2013, 2021). Ontologias, quando utilizadas desde as etapas iniciais do projeto conceitual de sistemas, contribuem para a modelagem semântica da informação, apoiando a definição do modelo de dados, a lógica de funcionamento e, posteriormente, a geração de interfaces. Nesse contexto, permitem a criação de formulários mais precisos, adaptativos e semanticamente coerentes, como parte de um sistema concebido de forma integrada.

O objetivo deste artigo é apresentar uma proposta de arquitetura orientada por ontologias para geração automatizada de interfaces adaptativas e validá-la. A abordagem visa aprimorar a entrada de dados em sistemas de informação, garantindo maior precisão conceitual, redução de ambiguidade e padronização na coleta de informações.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Com base na revisão de literatura realizada, apresentam-se a seguir os fundamentos conceituais e técnicos que sustentam a proposta deste trabalho.

### **2.1 Fundamentos em Ontologias**

Originalmente vinculada à Filosofia como o ramo da metafísica que estuda as categorias do ser, a ontologia passou a ser compreendida, na Ciência da Informação (CI) e na Computação, como um artefato conceitual que representa formalmente algum conhecimento, conforme o nível de detalhe necessário (Almeida, 2013). Arp, Smith e Spear (2015) definem ontologias como estruturas destinadas a descrever universais, categorias e suas inter-relações.

Na CI, as ontologias são classificadas como Sistemas de Organização do Conhecimento (SOC), juntamente com taxonomias e tesouros, distinguindo-se pelo alto grau de formalização e pela capacidade de representar logicamente os conceitos, permitindo inferência automática (Sørgel, 1999; Souza; Tudhope; Almeida, 2010; Vickery, 1997).

Neste trabalho, ontologias são entendidas como SOC que descrevem aspectos da realidade por meio de entidades, propriedades e relacionamentos. Segundo Arp, Smith e Spear (2015), entidades são tudo aquilo que pode ser representado em uma ontologia formal, incluindo objetos materiais, qualidades, papéis, processos e outras categorias ontológicas organizadas em taxonomias fundamentadas. Contribuem para reduzir ambiguidades semânticas e integrar fontes heterogêneas de informação, ao articular linguagens, documentos e sistemas (Farinelli; Almeida, 2014; Gruber, 1993). Sua estrutura lógica favorece a interoperabilidade, o reuso e a inferência computacional.

A classificação das ontologias orienta seu uso e reuso em sistemas informacionais. Guarino (1998) propõe dois eixos: o nível de abstração (ontologias de alto nível, de domínio, de tarefa e de aplicação) e o grau de compartilhamento (ontologias de referência ou compartilháveis). Arp, Smith e Spear (2015) complementam com o escopo conceitual e o propósito funcional das ontologias.

Ontologias de alto nível, como a BFO<sup>1</sup> e a DOLCE<sup>2</sup>, representam categorias genéricas e domínio-independentes. As ontologias de referência, como a IAO<sup>3</sup>, organizam conceitos amplos de macrodomínios, promovendo padronização terminológica. Ontologias de domínio, como a OntONeo<sup>4</sup> e a CIDO<sup>5</sup>, detalham áreas específicas do conhecimento. Já as ontologias de tarefa descrevem processos e ações operacionais, enquanto as de aplicação combinam elementos de outras categorias para resolver problemas concretos em contextos específicos.

## 2.2 A linguagem OWL e os elementos formais da ontologia

As ontologias são formadas por elementos que estruturam a representação do conhecimento, como entidades, atributos, relacionamentos, instâncias, restrições, anotações e axiomas. Ressalta-se que tanto na linguagem OWL quanto na ferramenta Protégé, as entidades são chamadas de classes (Farinelli, 2017).

Cada elemento desempenha um papel específico: **classes** representam as coisas sejam elas abstratas ou não; **atributos** qualificam as classes com valores; **relacionamentos** conectam classes; **instâncias** que são exemplos de classes; **restrições** delimitam ocorrências; **anotações**

---

<sup>1</sup> Mais detalhes em: <https://basic-formal-ontology.org>

<sup>2</sup> Mais detalhes em: <https://www.loa.istc.cnr.it/dolce/overview.html>

<sup>3</sup> Mais detalhes em: <https://obofoundry.org/ontology/iao.html>

<sup>4</sup> Mais detalhes em: <https://obofoundry.org/ontology/ontoneo.html>

<sup>5</sup> Mais detalhes em: <https://obofoundry.org/ontology/cido.html>

adicionam metadados; e **axiomas** definem regras formais. Essa estrutura é independente da linguagem utilizada, sendo a base da modelagem ontológica em diferentes domínios (Farinelli, 2017).

Para formalizar e compartilhar essas representações, utiliza-se a *Web Ontology Language* (OWL), recomendada pelo W3C (*World Wide Web Consortium*). Baseada em lógica descritiva, a OWL permite estruturar o conhecimento com diferentes níveis de expressividade e suporte a inferência automática. A linguagem descreve **classes**, **propriedades** e **indivíduos**, possibilitando raciocínio sobre os dados modelados e suporta diversas formas de serialização, como RDF/XML, OWL/XML (Hitzler *et al.*, 2012).

Na modelagem realizada neste trabalho, alguns elementos utilizados da OWL são: *owl:Class* para representar os conceitos do domínio (entidades e classes); *owl:ObjectProperty* para relações entre classes; *owl:DatatypeProperty* para atributos literais aos conceitos, como texto, número ou data; *rdfs:label* para inclusão do nome legível da propriedade (termo); *obo:IAO\_0000115* para descrições conceituais; e *owl:Restriction* para expressar restrições sobre propriedades, como cardinalidade e participação em relações. Também foram aplicadas restrições baseadas em XML Schema, como *xsd:string*, *xsd:date*, *xsd:maxLength* e *xsd:minLength*, que definem o formato e os limites de entrada de dados.

A Figura 1 apresenta um trecho da ontologia ONTAE (descrita na seção 4) em OWL, referente à classe “*minicurso*”, que exemplifica a estrutura formal utilizada pelo parser.

Figura 1 – Exemplo de formalização da classe Minicurso em OWL

```
<!-- http://www.semanticweb.org/ontologias/ONTAE/ONTAE_0000023 -->
<owl:Class rdf:about="http://www.semanticweb.org/ontologias/ONTAE/ONTAE_0000023">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologias/ONTAE/ONTAE_0000021"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://purl.obolibrary.org/obo/BFO_0000050"/>
      <owl:minQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">0</owl:minQualifiedCardinality>
      <owl:onClass>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <rdf:Description rdf:about="http://www.semanticweb.org/ontologias/ONTAE/ONTAE_0000016"/>
            <rdf:Description rdf:about="http://www.semanticweb.org/ontologias/ONTAE/ONTAE_0000017"/>
          </owl:unionOf>
        </owl:Class>
      </owl:onClass>
    </owl:Restriction>
  </rdfs:subClassOf>
  <obo:IAO_0000115 xml:lang="pt-br">Curso destinado a fornecer conhecimento e habilidades específicas através de um programa estruturado de ensino, focado em atividades práticas, com duração curta.</obo:IAO_0000115>
  <rdfs:label xml:lang="pt">minicurso</rdfs:label>
</owl:Class>
```

Fonte: elaborado pelos autores (2025)

O fragmento evidencia a representação estruturada e semântica dos conceitos do domínio. A classe “*minicurso*” é definida como subclasse de outra mais geral “*Curso*” (*ONTAE\_0000021*) e inclui uma restrição de cardinalidade mínima (*owl:minQualifiedCardinality = 0*) sobre uma propriedade do tipo *owl:ObjectProperty*. Essa

propriedade está vinculada a uma união de classes (*owl:unionOf*), indicando que a instância pode relacionar-se, opcionalmente, a qualquer uma das classes especificadas.

Além disso, observa-se o uso da propriedade *rdfs:label* com o atributo *xml:lang="pt"*, que fornece o rótulo legível “minicurso” no idioma português, utilizado diretamente como rótulo dos campos da interface. A definição conceitual da classe é especificada por meio da anotação *obo:IAO\_0000115*, também em português, e insere a definição de minicurso como um programa estruturado, voltado à prática, com curta duração.

Esses elementos são fundamentais para o parser, pois permitem a geração automática de formulários com campos nomeados e descritos de forma semanticamente coerente com o domínio representado na ontologia.

### **2.3 Interface Adaptativa do Usuário**

A Interação Humano-Computador (IHC) investiga como usuários interagem com sistemas computacionais, tendo a interface como ponto central. Com os avanços em ontologias, inteligência artificial e computação em nuvem, surgiram interfaces mais inteligentes e adaptativas (Zouhaier *et al.*, 2022). As Interfaces Gráficas do Usuário (GUI) organizam elementos visuais para facilitar a navegação, exigindo equilíbrio entre clareza visual e requisitos técnicos, como restrições de formato e tipo de dado. As Interfaces Adaptativas do Usuário (IAU) ampliam essa abordagem ao ajustar dinamicamente sua estrutura conforme o perfil, preferências ou contexto do usuário (Deuschel, 2018).

As Interfaces do Usuário Orientadas por Ontologias (ODUI) avançam ainda mais ao utilizar ontologias de domínio para gerar dinamicamente os campos da interface, interpretando conceitos, propriedades e restrições para construir formulários semanticamente coerentes e padronizados (Shahzad, 2011).

## **3 METODOLOGIA**

Esta é uma pesquisa aplicada, de abordagem qualitativa, com caráter descritivo e exploratório (Gil, 2008). A revisão bibliográfica foi conduzida por meio de buscas nas bases de dados SpringerLink, IEEE Xplore, BDTD e Portal de Periódicos da CAPES, com foco em trabalhos publicados a partir de 2018. A seleção considerou publicações que utilizassem a linguagem OWL para representação ontológica e que propusessem a geração automática de interfaces. Para conduzir a concepção da arquitetura e sua comprovação empírica por meio de uma prova de conceito (PoC), adotou-se o método *Design Science Research* (DSR), que estrutura a

pesquisa por meio do ciclo de construção e validação de soluções tecnológicas (Bax, 2013). A ontologia utilizada na PoC apresentada na seção 4.1, foi construída a partir da metodologia ReBORM apresentada por Farinelli (2017, 2020).

A implementação da PoC utilizou as tecnologias Python, RDFLib e Flask, e sua validação considerou a geração dinâmica de formulários com base na coerência semântica, nas restrições definidas na ontologia e na adaptabilidade à estrutura conceitual do domínio.

## 4 RESULTADOS

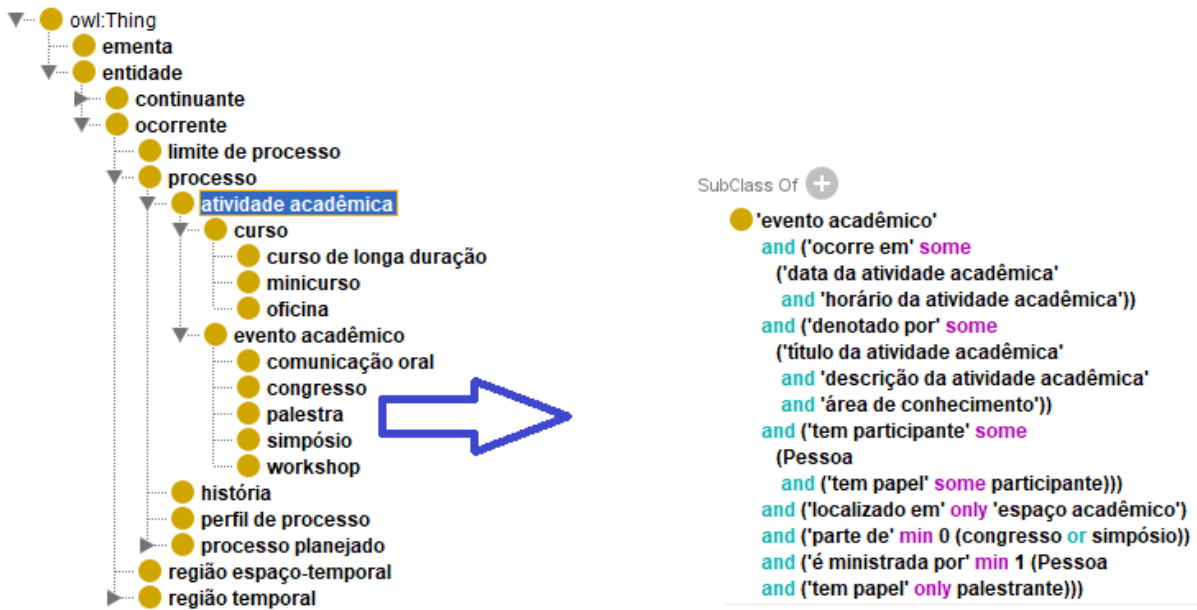
Como parte dos resultados, são apresentados a ontologia, a arquitetura do sistema e a PoC, que comprovam a aplicação prática da abordagem proposta.

### 4.1 A Ontologia do Domínio de Atividades Acadêmicas

A ontologia utilizada como base para a PoC foi denominada ONTAE (Ontologia de Eventos Acadêmicos) e tem como foco a representação formal do domínio de atividades acadêmicas, com ênfase em eventos como minicursos, oficinas, palestras e seminários. A construção da ONTAE seguiu a ReBORM (*Realism-Based Ontology engineering Methodology*), ou Metodologia de Engenharia de Ontologia Baseada em Realismo, apresentada em Farinelli (2017, 2020). A ReBORM combina fundamentos do realismo ontológico com práticas da engenharia de ontologias, organizando o processo em um ciclo de vida iterativo e incremental com cinco fases: conceitual, iniciação, projeto, implementação e disponibilização. Durante a fase de projeto, conforme previsto na ReBORM, é necessário selecionar uma ontologia de alto nível que sirva de referência para garantir a coerência e a reusabilidade da ontologia de domínio. No caso da ONTAE, optou-se pela utilização da BFO (*Basic Formal Ontology*) como ontologia de referência, considerando sua ampla adoção na construção de ontologias científicas e seu alinhamento com os princípios do realismo ontológico. A ONTAE foi formalizada no editor Protégé 5.5, utilizando a linguagem OWL no formato RDF/XML. O uso da OWL permitiu definir restrições de cardinalidade e tipos de dados, garantindo maior rigor semântico na representação ontológica (Hitzler *et al.*, 2012).

A ONTAE define classes hierárquicas, *object properties* para relações entre entidades e *data properties* para atributos como título, data e local. A Figura 2 apresenta a hierarquia de classes no Protégé, com destaque para *atividade acadêmica* e suas subclasses. À direita, observam-se as restrições da classe *palestra*, com propriedades como *tem participante*, *é ministrada por* e *ocorre em*, além de atributos descritivos como *título*, *descrição* e *horários*.

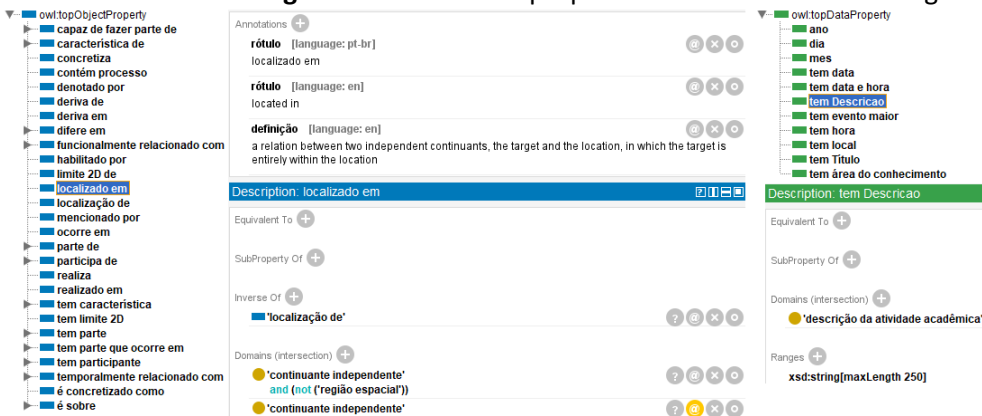
Figura 2 - Recorte das classes da ONTAE no Protégé



Fonte: elaborada pelos autores (2025)

A Figura 3 mostra um recorte das propriedades da ONTAE no Protégé, com uma *object property* à direita e uma *data property* à esquerda. A propriedade *localizado em*, adaptada da *Relations Ontology (RO)*, relaciona a atividade acadêmica ao local de realização, com rótulos em português e inglês. Já a propriedade *tem Descrição*, associada à classe *Descrição da Atividade Acadêmica*, utiliza o tipo *xsd:string* com limite de 250 caracteres, garantindo padronização na entrada de textos.

Figura 3 - Recorte das propriedades da ONTAE no Protégé



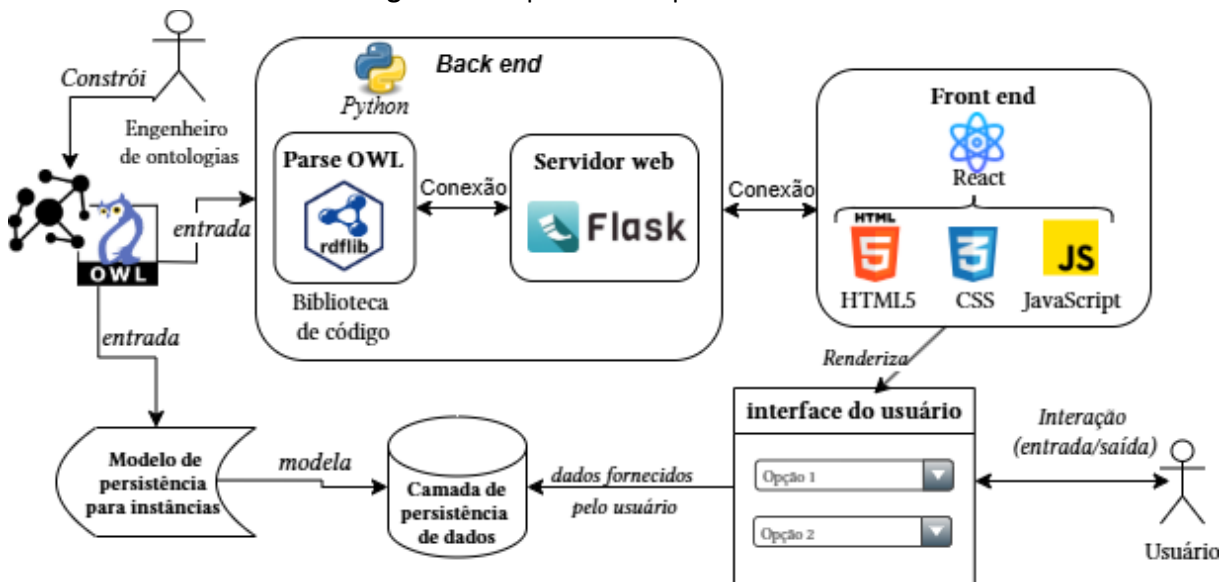
Fonte: elaborada pelos autores (2025)

Essas definições orientam a construção de formulários semanticamente alinhados à estrutura conceitual da ontologia.

#### 4.2 Arquitetura proposta de Interface Adaptativa do Usuário (IAU)

A arquitetura proposta (Figura 4) é projetada para utilizar uma ontologia como base semântica para a geração automatizada de interfaces adaptativas de cadastro. A solução é composta por dois principais componentes: *back-end* e *front-end*.

Figura 4 - Proposta de Arquitetura de IAU



Fonte: elaborada pelos autores (2025)

O *back-end* foi desenvolvido utilizando o *framework* *Flask* em *Python*, utilizando a biblioteca *rdflib* para realizar o *parse* da ontologia em OWL. A partir dela, são extraídas subclasses, relacionamentos (*object properties*) e propriedades de dados (*data properties*), que definem como os campos da interface devem ser apresentados e quais tipos de dados devem aceitar.

Essas informações alimentam a camada de persistência e orientam a construção dinâmica da interface do usuário (*front-end*), implementada em *React*, com tecnologias HTML5, CSS e JavaScript. A interface resultante é capaz de adaptar seus elementos com base nas definições ontológicas, aprimorando o alinhamento semântico e maior precisão na entrada de dados pelo usuário.

Para apoiar essa abordagem, diversos trabalhos foram analisados na revisão da literatura, destacam-se Cannon *et al.* (2004) e Liu (2009), que retratam o estado da arte na geração de interfaces orientadas por ontologias. Em geral, a ontologia é utilizada como apoio à modelagem de domínio, sem estar formalizada em OWL nem integrada diretamente ao fluxo de geração da interface. Quando há formalização, observa-se o uso de bibliotecas Java para construção de GUI (*Graphical User Interface*), o que, embora permita representação

semântica e adaptação ao domínio, limita-se a ferramentas convencionais de desktop, sem considerar tecnologias de *front-end* baseadas em *web* com suporte para *mobile* como *React* ou *Angular*.

#### 4.2.1 Parse e renderização

A arquitetura proposta utiliza a biblioteca *RDFLib* para realizar o parse de triplas RDF (sujeito-predicado-objeto) a partir de uma ontologia OWL. O processo tem início com a seleção de uma classe de entrada (sujeito), a partir da qual o sistema identifica todas as propriedades de objeto (predicados) e as classes relacionadas a ela (objetos). Com essas informações, são gerados componentes de interface em HTML (*HyperText Markup Language*) e de persistências em JSON (*JavaScript Object Notation*), permitindo a renderização dinâmica de formulários em aplicações web e a persistência em arquivos ou sistemas de gerenciamento de banco de dados baseado em documentos ou em grafos.

O Quadro 1 apresenta o mapeamento entre os elementos da OWL e os componentes da interface HTML. Por exemplo, *owl:Class* estrutura o formulário principal, *owl:DatatypeProperty* define o tipo de dado (texto, número, data), e *rdfs:label* e *rdfs:comment* contribuem para a clareza e usabilidade dos campos.

**Quadro 1 – Mapeamento dos elementos OWL para atributos e componentes HTML**

Elemento OWL	Elemento na Interface HTML	Comportamento na Interface
owl:Class	Formulário principal, campo de formulário	Estrutura base do formulário
owl:DatatypeProperty	Tipo de dado relacionado a uma classe	Define se um campo é texto ou numérico, em formato de data ou horário, além de informar mínimo e máximo de caracteres esse campo pode receber.
owl:ObjectProperty	Subformulário, campo de formulário	Campo composto ou aninhado com repetição
rdfs:label	<input value="label">	Nome exibido do campo
rdfs:comment	<label for="campo">Nome <span title="Descrição baseada no rdfs:comment">? </span> </label>	Ajuda contextual ou explicação
xsd:string[maxLength=50]	<input type="text">	Campo de texto curto
xsd:string[maxLength=250]	<input type="text">	Campo de texto longo
xsd:time	<input type="time">	Campo de horário
xsd:date	<input type="date">	Campo de data

**Fonte:** elaborado pelos autores (2025), com base em OWL, HTML5.

O *front-end*, foi projetado para operar com um único componente de código para o formulário dinâmico, capaz de interpretar a estrutura em *JSON* gerada pelo *back-end* e renderizar automaticamente os formulários com base na ontologia. Essa abordagem representa uma mudança em relação ao modelo tradicional de desenvolvimento, no qual o desenvolvedor precisa modelar bancos de dados, criar objetos e construir formulários manualmente, muitas vezes sem apoio de uma estrutura conceitual formalizada. Na proposta orientada por ontologias, a modelagem conceitual continua essencial, mas ela é explicitada e formalizada em um artefato ontológico, que passa a guiar diretamente a geração das interfaces e a estrutura de dados, promovendo maior alinhamento semântico com o domínio.

Nos sistemas convencionais, os relacionamentos entre entidades são representados por *chaves estrangeiras* em bancos relacionais, que garantem integridade estrutural, mas não expressam o *significado semântico* das conexões. Em contraste, na abordagem baseada em *ontologias*, os relacionamentos são descritos por *triplas RDF*, por exemplo: “*Palestra*”, “*tem título*”, “*Título da Atividade Acadêmica*”, permitindo uma representação clara e semântica do papel de cada elemento no domínio. Isso enriquece a estrutura informacional do sistema e promove maior coerência na organização dos dados.

Em abordagens tradicionais, desenvolvedores geralmente criam múltiplos formulários ou recorrem a versões genéricas com muitos campos opcionais para lidar com diferentes tipos de dados. Isso pode tornar o preenchimento confuso e comprometer a padronização das informações. Na proposta adotada, essa complexidade é resolvida com a definição dos campos diretamente na ontologia. O desenvolvedor implementa apenas um formulário de código, e os campos são gerados automaticamente com base nas propriedades e restrições ontológicas, resultando em formulários mais claros, precisos e alinhados ao domínio.

#### **4.3 Prova de Conceito**

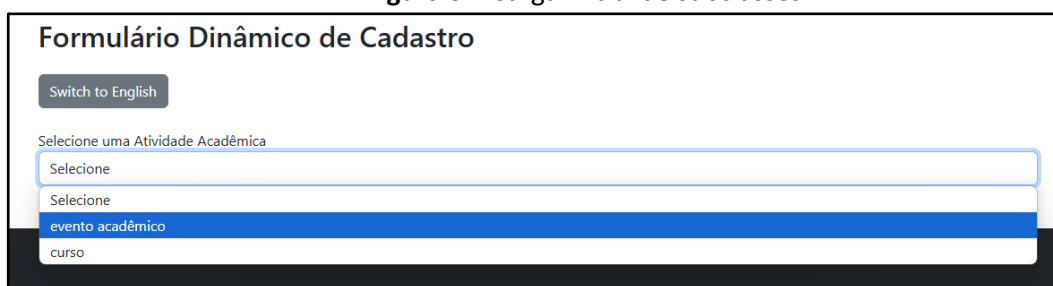
A prova de conceito (PoC) foi desenvolvida com o objetivo de validar a viabilidade técnica da arquitetura proposta. O protótipo implementado consome diretamente a ontologia ONTAE e é capaz de gerar dinamicamente formulários de cadastro adaptativos, cujos campos, rótulos e restrições são derivados automaticamente da estrutura semântica definida na ontologia. Essa geração dinâmica ocorre a partir da interpretação de classes, propriedades e restrições modeladas em OWL, garantindo que a interface reflita fielmente os conceitos do domínio. Com isso, reduz-se a necessidade de codificação manual e assegura-se maior

padronização, reutilização e integridade na entrada de dados. A PoC demonstra, assim, o potencial da ontologia como motor semântico para a construção de interfaces guiadas por conhecimento.

A comunicação entre o *front-end* e a ontologia é realizada por meio de rotas disponibilizadas em um servidor *Flask*. A rota */get\_subclasses*<sup>6</sup> permite a recuperação recursiva das subclasses de uma determinada classe. Já a rota */get\_class\_details*<sup>7</sup> é ativada quando o usuário seleciona uma classe terminal (classe-folha), como “Palestra”. Nesse momento, o sistema consulta a ontologia utilizando a biblioteca *rdflib* e retorna os metadados relacionados, incluindo propriedades objetais, classes associadas e atributos da atividade — que são convertidos em campos do formulário (Figuras 6, 7 e 8).

O fluxo de navegação da interface ocorre em três etapas: (i) exibição das subclasses da classe raiz *Atividade Acadêmica* (Figura 6); (ii) seleção de uma subcategoria, como *Evento Acadêmico*, com carregamento de subclasses específicas (Figura 7); e (iii) geração automática do formulário conforme as propriedades da classe terminal escolhida, como *Palestra* (Figura 8). Os campos são rotulados com base nas anotações da ontologia, como *rdfs:label*, e seguem as restrições semânticas definidas em OWL, aprimorando o alinhamento conceitual entre a atividade selecionada e a interface apresentada.

Figura 6 – Carga inicial de subclasses



Formulário Dinâmico de Cadastro

Switch to English

Selecione uma Atividade Acadêmica

Selecione

Selecione

evento acadêmico

curso

Fonte: elaborada pelos autores (2025).

<sup>6</sup> Disponível em: */get\_subclasses?class={URI\_da\_classe\_da\_ontologia}*

<sup>7</sup> Disponível em: */get\_class\_details?class={URI\_da\_classe\_da\_ontologia}*

Figura 7 – Carga de subclasses da classe “Evento Acadêmico”

The screenshot shows a web form titled "Formulário Dinâmico de Cadastro". At the top left, there is a "Switch to English" button. Below it, there is a label "Selecione uma Atividade Acadêmica" followed by a dropdown menu containing "evento acadêmico". Below that, another label "Selecione um(a) evento acadêmico" is followed by a larger dropdown menu. This menu is open, showing a list of options: "Selecione", "congresso", "simpósio", "workshop", "palestra" (highlighted in blue), and "comunicação oral".

Fonte: elaborada pelos autores (2025)

Figura 8 – Formulário de registro de “Palestra”

The screenshot shows a registration form for a "Palestra". At the top, there is a dropdown menu for "Selecione um(a) evento acadêmico" with "palestra" selected. Below this are several required fields (marked with an asterisk): "Denotado por (Área de conhecimento)" with a "Máximo de 50 caracteres" limit; "Denotado por (Título da atividade acadêmica)" with a "Máximo de 50 caracteres" limit; "Denotado por (Descrição da atividade acadêmica)" with a "Máximo de 250 caracteres" limit. Below these are "Localizado em" (with a "Selecione" dropdown), "Ocorre em (Data da atividade acadêmica)" (with a "dd/mm/aaaa" format and a calendar icon), "Ocorre em (Horário de início da atividade acadêmica)" (with "--:--" format and a clock icon), and "Ocorre em (Horário de fim da atividade acadêmica)" (with "--:--" format and a clock icon). At the bottom left, there is a blue "Enviar" button.

Fonte: elaborada pelos autores (2025)

Ao selecionar uma *classe-folha*, isto é, uma classe sem subclasses, como *Palestra*, o sistema executa a rota `/get_class_details`, que utiliza a biblioteca *rdflib* para extrair todas as *object properties* diretamente associadas à classe escolhida. Para cada propriedade, também são identificadas as classes relacionadas, como *Título da Atividade Acadêmica* e *Palestrante*, que atuam como metadados no contexto da geração da interface. Com base nesses dados, o formulário é construído automaticamente, respeitando a estrutura semântica da ontologia, conforme ilustrado na Figura 8.

A Figura 9 ilustra o JSON gerado pela rota `/get_class_details`, acionada ao se atingir uma classe-folha como *Palestra*. Esse retorno inclui os campos *property* (URI da propriedade), *label*, *cardinality*, *dataType*, *relatedClass* e *restrictions*, permitindo à interface aplicar, por exemplo, limites de tamanho de caracteres (`xsd:maxLength`), tipos de dados como `xsd:string`, `xsd:date` e `xsd:time`, e regras de obrigatoriedade.

Figura 9 – JSON de propriedades da classe folha

```
▼ 3: {property: "http://purl.obolibrary.org/obo/IAO_0000235", label: "denotado por",...}
  cardinality: "only"
  ▶ dataType: [{"http://www.w3.org/2001/XMLSchema#string"}]
  label: "denotado por"
  property: "http://purl.obolibrary.org/obo/IAO_0000235"
  relatedClass: "título da atividade acadêmica"
  ▶ restrictions: {xsd:maxLength: "50"}
▼ 4: {property: "http://purl.obolibrary.org/obo/IAO_0000235", label: "denotado por",...}
  cardinality: "only"
  ▶ dataType: [{"http://www.w3.org/2001/XMLSchema#string"}]
  label: "denotado por"
  property: "http://purl.obolibrary.org/obo/IAO_0000235"
  relatedClass: "descrição da atividade acadêmica"
  ▼ restrictions: {xsd:maxLength: "250"}
    xsd:maxLength: "250"
▼ 5: {property: "http://purl.obolibrary.org/obo/RO_0001025", label: "localizado em",...}
  cardinality: "only"
  label: "localizado em"
  property: "http://purl.obolibrary.org/obo/RO_0001025"
  relatedClass: "espaço acadêmico"
  ▼ subclasses: [{"uri": "http://www.semanticweb.org/ontologias/ONTAE/ONTAE_00000012", label: "laboratório",...},...]
    ▼ 0: {uri: "http://www.semanticweb.org/ontologias/ONTAE/ONTAE_00000012", label: "laboratório",...}
      ▶ dataProperties: [{"property": "http://www.semanticweb.org/ontologias/ONTAE/ONTAE_00000036",...},...]
        label: "laboratório"
      ▶ restrictions: [{"property": "http://purl.obolibrary.org/obo/RO_0001015",...}]}]
```

Fonte: elaborada pelos autores (2025)

Essas *restrictions* são definidas diretamente nas *data properties* da ontologia ONTAE e são fundamentais para que o *front-end* construa e valide os campos de entrada conforme a semântica do domínio modelado. Vale destacar que, embora algumas propriedades possam ser representadas de forma direta e inversa na ontologia, o sistema evita duplicidades ao seguir, no processo de *parsing*, exclusivamente o caminho *classe sujeito* → *predicado* → *classe objeto*. Isso garante a geração de formulários claros, concisos e semanticamente consistentes.

Com esse mecanismo, a arquitetura proposta demonstra sua capacidade de gerar interfaces adaptadas automaticamente com base na ontologia de domínio. A entrada de dados se torna mais precisa, controlada e semanticamente orientada, assegurando a integridade das informações e favorecendo sua futura recuperação em sistemas interoperáveis.

#### 4.4 Avaliação dos Resultados

Os testes realizados demonstraram que o uso da ontologia como base para a geração de interfaces contribuiu significativamente para a consistência dos dados inseridos. A construção dos formulários a partir da estrutura ontológica possibilitou maior padronização, clareza na apresentação dos campos e validação semântica no momento da entrada das informações. Esse alinhamento reduziu ambiguidades e inconsistências, reforçando a capacidade das ontologias de melhorar a qualidade da informação nos sistemas computacionais, conforme apontado por Farinelli e Almeida (2014).

A adaptação dinâmica da interface a partir das classes e propriedades da ontologia também favoreceu a usabilidade, permitindo que o usuário interagisse apenas com os campos

relevantes para o contexto selecionado. O resultado foi uma experiência de cadastro mais precisa e orientada ao domínio, com potencial para facilitar a interoperabilidade entre sistemas e a recuperação qualificada dos dados no futuro.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho demonstrou como a organização e a representação formal do conhecimento, por meio de ontologias, podem contribuir efetivamente para o desenvolvimento de interfaces adaptativas de usuário. A partir da modelagem ontológica do domínio de *atividades acadêmicas* e da implementação de uma PoC, foi possível validar a viabilidade técnica de uma arquitetura capaz de gerar formulários de cadastro dinâmicos e semanticamente coerentes com as classes e propriedades descritas na ontologia *ONTAE*.

Diferentemente das abordagens tradicionais, centradas em modelagens genéricas de banco de dados e na construção manual de formulários, a proposta apresentada transfere à ontologia a responsabilidade de estruturar semanticamente os dados e suas restrições. Isso permite que o sistema atue com maior precisão na coleta de informações, reduzindo ambiguidades e inconsistências na entrada de dados.

A utilização da biblioteca *rdflib*, associada à linguagem *OWL* e ao framework *Flask*, possibilitou o desenvolvimento de uma arquitetura modular e escalável, na qual um único componente de formulário adapta-se automaticamente às definições da ontologia. A PoC evidenciou que essa abordagem melhora a usabilidade, promove a padronização e fortalece o controle semântico sobre os dados registrados.

Destaca-se que a camada de persistência, poderá ser implementada por meio do armazenamento dos dados em arquivos *JSON* ou pela integração com bancos de dados *NoSQL*, orientados a grafos ou documentos. Ressalta-se que é um trabalho em andamento, e a proposta ainda será ampliada com a implementação da camada de persistência, o aprimoramento da interface com agrupamento semântico e adaptação visual, além da integração de mecanismos de interoperabilidade ontológica. Também será incorporado um modelo de contexto que permita personalização da interface conforme o domínio e a aplicação.

## REFERÊNCIAS

ALMEIDA, Mauricio Barcellos. **Ontologia em ciência da informação: teoria e método**. 1. ed. Curitiba: CRV, 2021. v. 1.

ALMEIDA, Mauricio Barcellos. Revisiting ontologies: A necessary clarification. **Journal of the American Society for Information Science and Technology**, v. 64, n. 8, p. 1682–1693, ago. 2013.

ARP, Robert; SMITH, Barry; SPEAR, Andrew D. **Building ontologies with basic formal ontology**. [S. l.]: MIT Press, 2015.

BATINI, Carlo; SCANNAPIECO, Monica. Methodologies for Data Quality Assessment and Improvement. **ACM Computing Surveys**, v. 41, n. 3, p. 1–52, 2009.

BAX, Marcelo Peixoto. Design science: filosofia da pesquisa em ciência da informação e tecnologia. **Ciência da Informação** (Online), v. 42, p. 298-312, 2013.

CANNON, Alan; KENNEDY, Jessie B.; PATERSON, Trevor; WATSON, Mark F. Ontology-Driven Automated Generation of Data Entry Interfaces to Databases. *In*: WILLIAMS, Howard; MACKINNON, Lachlan (ed.). **Key Technologies for Data Management**. BNCOD 2004. Lecture Notes in Computer Science, v. 3112. Berlin: Springer, 2004. p. 150–164.

DEUSCHEL, Tilman. On the Influence of Human Factors in Adaptive User Interface Design. *In*: CONFERENCE ON USER MODELING, ADAPTATION AND PERSONALIZATION ADJUNCT, 26., 2018, Singapore. **Proceedings** [...] New York: ACM, 2018. p. 1-5.

FARINELLI, Fernanda; ALMEIDA, Maurício Barcellos. Interoperabilidade semântica em sistemas de informação de saúde por meio de ontologias formais e informais: um estudo da norma OpenEHR. *In*: CONFERÊNCIA INTERNACIONAL ACESSO ABERTO, PRESERVAÇÃO DIGITAL, INTEROPERABILIDADE, VISIBILIDADE E DADOS CIENTÍFICOS, 4., 2014, Porto Alegre. **Anais** [...]. Porto Alegre: [s. n.], 2014. Disponível em: [https://mba.eci.ufmg.br/downloads/Biredial2014\\_144\\_web.pdf](https://mba.eci.ufmg.br/downloads/Biredial2014_144_web.pdf). Acesso em: 25 maio 2025.

FARINELLI, Fernanda. **Realismo ontológico aplicado a interoperabilidade semântica entre sistemas de informação**: um estudo de caso do domínio obstétrico e neonatal. 2017. 256 f. Tese (Doutorado em Gestão e Organização do Conhecimento) - Pós-Graduação em Gestão & Organização do Conhecimento da Escola de Ciência da Informação, Universidade Federal de Minas Gerais, Belo Horizonte, 2017. Disponível em: <https://repositorio.ufmg.br/handle/1843/BUBD-AX2J5B>. Acesso em: 25 maio 2025.

FARINELLI, Fernanda. Um diálogo entre o realismo ontológico e a engenharia de ontologias na construção de artefatos de representação. *In*: ALMEIDA, Maurício Barcellos. **Representação do conhecimento, ontologias e linguagem**: pesquisa aplicada em ciência da informação. 1. ed. Curitiba, PR: Editora CRV, 2020. p. 277–294.

GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6. ed. ed. São Paulo: Editora Atlas SA, 2008.

GRUBER, Thomas R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, n. 2, p. 199–220, 1 jun. 1993.

GUARINO, Nicola. Formal ontology in information systems. *In*: FORMAL ONTOLOGY IN INFORMATION SYSTEMS, 1., 1998, Trento, Italy. **Proceedings** [...]. Trento, Italy: IOS press, 1998. p. 3–15.

HITZLER, Pascal *et al.* **OWL 2 Web Ontology Language Primer (Second Edition)**: W3C Recommendation. 2012. Disponível em: <https://www.w3.org/TR/owl2-primer/>. Acesso em: 25 maio 2025.

LIU, Fangfang. **An ontology-based approach to Automatic Generation of GUI for Data Entry**. 2009. Thesis (Master in Computer Science)—University of New Orleans, New Orleans, 2009. Disponível em: <https://scholarworks.uno.edu/td/1094/>. Acesso em: 25 maio 2025.

SHAHZAD, Syed Khuram. Ontology-based User Interface Development: User Experience Elements Pattern. **Journal of Universal Computer Science**, v. 17, n. 7, p. 1078-1088, 2011. Disponível em: DOI: 10.3217/jucs-017-07-1078. Acesso em: 25 maio 2025.

SØERGEL, Dagobert. The rise of ontologies or the reinvention of classification. **Journal of the Association for Information Science and Technology**, v. 50, n. 12, p. 1119, 1999.

SOUZA, Renato Rocha; TUDHOPE, Douglas; ALMEIDA, Mauricio Barcellos. The KOS spectra: A tentative typology of knowledge organization systems. *In*: INTERNATIONAL CONFERENCE ON INTERNATIONAL SOCIETY FOR KNOWLEDGE ORGANIZATION, ISKO, 11., 2010, Rome, Italy. **Proceedings** [...] Rome, Italy: ISKO, 2010. p. 122-128.

VICKERY, Brian C. Ontologies. **Journal of Information Science**, v. 23, n. 4, p. 277-286, 1997.

ZOUHAIER, Mohamed *et al.* Towards intelligent adaptive user interfaces: a semantic model for user interaction personalization. **Procedia Computer Science**, v. 206, p. 75–86, 2022.