



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Modelo ontológico de competências pessoais de discentes graduandos em Computação

Miguel Ângelo Montagner Filho

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Profa. Dra. Germana Menezes da Nóbrega

Brasília
2022



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Modelo ontológico de competências pessoais de discentes graduandos em Computação

Miguel Ângelo Montagner Filho

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Profa. Dra. Germana Menezes da Nóbrega (Orientadora)
CIC/UnB

Profa. Dra. Prof. Dr.
Fernanda Lima Fernando W. Cruz

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 06 de maio de 2022

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

M758m Montagner, Miguel Angelo
Modelo ontológico de competências pessoais de discentes
graduandos em Computação / Miguel Angelo Montagner;
orientador Germana Menezes da Nóbrega. -- Brasília, 2022.
53 p.

Monografia (Graduação - Ciência da Computação) --
Universidade de Brasília, 2022.

1. Competência. 2. Ontologia. 3. Informática na Educação.
4. Modelo de Competências. I. Menezes da Nóbrega, Germana,
orient. II. Título.

Agradecimentos

Agradeço aos meus pais, por me auxiliarem nos meus momentos difíceis e estarem comigo para compartilhar os meus bons.

Agradeço à Prof. Dra. Germana Menezes da Nóbrega, por ser uma pessoa iluminada, sempre presente e disposta a estender uma mão auxiliadora.

Agradeço à minha companheira Helena Maria Ceccatelli, por ser uma pessoa que profundamente amo e me proporciona o privilégio de ser amado por ela.

Agradeço aos membros da banca examinadora, por se disporem a auxiliar com seu tempo e conhecimento para tecer críticas importantes e necessárias para o aprimoramento e conclusão do trabalho.

Resumo

Na educação formal, os alunos passam por processos educacionais que visam desenvolver competências. Estas competências, formalmente definidas por instituições competentes, são esperadas serem desenvolvidas por eles ao longo de sua jornada educacional. Contudo, muitas vezes os alunos estão alheios ao processo pedagógico como um todo, impactando negativamente em sua motivação e entendimento dos elementos de educação apresentados à eles. Este trabalho tem como objetivo avançar o ecossistema SmartUnB.ECOS ao apresentar um modelo de competências destes alunos a fim de facilitar o entendimento e uso destas informações acerca das competências. Em especial, foi utilizada a linguagem OWL através do software Protégé para implementar uma ontologia que modela competências formais, mas que também permite o uso para competências não-formais e informais. Foram abordados os conceitos que permeiam a educação na informática, trabalhos relacionados ao tema e outros trabalhos com propostas similares. Após, foi apresentado o método utilizado: *Ontology 101*. Com a aplicação do método, o modelo resultante foi inserido no *software* Protégé. O método levanta questões de teste, no qual o modelo foi bem sucedido.

Palavras-chave: Ontologia, Competência, Formal, protégé, Educação, Informática

Abstract

In formal education, students undergo through educational processes that seek to develop competencies. These competencies, when defined by formal institutions, are expected to be acquired by students throughout the academic process. Yet, students commonly are oblivious to them. This, in turn, impacts negatively in their motivation and their understanding of the whole educational process. This work further expands the SmartUnB.ECOS by developing a competency model of said students. This model should aid the student's understanding of their own formal competencies. In this work, OWL language was used and implemented by the *Protégé* software to create the formal competency model. It is worth to add that the model is not restrained to formal competencies only. This work firstly addresses the concepts behind education in informatics and related or similar papers. Secondly, it introduces the used methodology: Ontology 101. After applying the method, the resulting competency model is added to the Protégé software. Lastly, the tests produced by the method were applied and the model withstood.

Keywords: Ontology, Competency, Formal

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Projeto smartUnB.ECOS	1
1.3	Motivação/Justificativa	2
1.4	Questão de Pesquisa	2
1.5	Objetivos	3
1.5.1	Objetivo Geral	3
1.5.2	Objetivo(s) Específico(s)	3
1.6	Classificação do Trabalho	3
1.7	Escopo	4
1.8	Metodologia	5
1.9	Estrutura deste documento	5
2	Trabalhos Relacionados	6
2.1	Modelagem de competências	6
2.1.1	<i>Moodle</i>	6
2.1.2	<i>HR Open Standards Competency XML</i>	7
2.1.3	Gestão de Pessoas por Competências no Ambiente AulaNet	8
2.1.4	<i>TENCompetence</i>	9
2.2	Ontologias na Educação	11
2.2.1	<i>Ontology-Based Competency Management: Infrastructures for the Knowledge Intensive Learning Organization</i>	11
2.2.2	<i>Plugin</i> de extensão de funcionalidades do <i>Moodle</i>	11
2.2.3	<i>Ontologies in education – state of the art</i>	12
2.3	Generalidades sobre ontologias	12
2.3.1	Modelagem de uma Ontologia de Domínio com Foco em Competências para Sistemas de Recomendação na Educação	12
2.4	Uma revisão sistemática recente	12
2.4.1	<i>An ontology-based model for competence management</i>	12

2.4.2	<i>A new competency ontology for learning environments personalization</i>	13
2.5	Trabalhos recentes na equipe smartUnB.ECOS	13
2.5.1	Manutenção de portfólio de competências para cursos de graduação em computação por usuário gestor	13
2.5.2	Uma ontologia de perfil holístico para estudantes de graduação	14
2.5.3	Incorporando a Edição de Competências a uma Rede Social Descentralizada para Utilização em Local de Trabalho	14
3	Fundamentação para a Proposta	15
3.1	Definições	15
3.1.1	Ontologia	16
3.1.2	Competência	16
3.1.3	Conteúdo	18
3.2	Base Formal	19
3.2.1	Fontes Internacionais	19
3.2.2	Fontes Nacionais	19
3.3	Modelagem	20
3.3.1	Ontology 101	20
3.4	Protégé	23
4	Proposta	24
4.1	Modelagem da Ontologia	24
4.1.1	Aplicação do Método	24
4.1.2	Modelo Resultante	26
4.2	Persistência no Protégé	26
4.3	Testes das questões de competência	27
5	Conclusão	31
5.1	Contribuições	31
5.2	Limitações	32
5.3	Trabalhos Futuros	32
5.4	Considerações Finais	32
	Referências	33
	Apêndice	36
	A Código Turtle da Ontologia	37

Lista de Figuras

1.1 Diagrama de Gomes para a classificação de pesquisa	4
2.1 Inserção de nova competência em um <i>framework</i> na plataforma <i>Moodle</i> . . .	7
2.2 Níveis taxonômicos de um <i>framework</i> na plataforma <i>Moodle</i>	8
2.3 Representação visual do esquema em <i>XML</i> da HR Open	9
2.4 Modelo de competência da extensão do <i>AulaNet</i>	10
2.5 Elementos básicos de uma Ontologia em competências para Sicília[1]	11
4.1 Diagrama de Classes resultante, gerado pelo <i>Plugin OntoGraf</i>	27
4.2 Tela inicial da versão Java do Protégé	28
4.3 Aba de Entidades no <i>Protégé</i>	28
4.4 Diagrama parcial da Ontologia	29
4.5 diagrama de propriedades de uma Disciplina e seus conteúdos abordados. . .	30

Lista de Abreviaturas e Siglas

CIC Departamento de Ciência da Computação.

RF2017 Referenciais de Formação para os Cursos de Graduação em Computação 2017.

UnB Universidade de Brasília.

Capítulo 1

Introdução

1.1 Contextualização

Este trabalho foi desenvolvido para empregar os conhecimentos adquiridos ao longo da graduação em Bacharelado em Ciência da Computação e é o resultado final das disciplinas Trabalho de Graduação 1 e Trabalho de Graduação 2.

O projeto foi escrito durante o segundo semestre letivo de 2020 da Universidade de Brasília, realizado no período letivo de 4 de janeiro até 28 de maio de 2021. O trabalho foi finalizado no segundo semestre letivo de 2021, ministrado entre as datas de 17 de janeiro de 2022 a 05 de maio de 2022. Ambos os semestres foram ministrados à distância e no ano seguinte ao seus períodos nomeados, em virtude da pandemia de COVID-19.

1.2 Projeto smartUnB.ECOS

Este trabalho está inserido dentro de uma iniciativa que visa trazer um ecossistema educacional apoiado por computador (também conhecidos como ecossistemas de *e-Learning*). O projeto aponta os benefícios de tornar tal plataforma disponível para alunos, bem como para os educadores em virtude do incremento na capacidade didática e gerência dos processos pedagógicos.

Além disso, também foi verificado que promover o acoplamento de sistemas e-Learning com redes sociais tem uma capacidade potencializadora. Entretanto, houve uma apreensão quanto ao uso de redes sociais privadas quanto à segurança dos dados. A solução encontrada foi o uso de redes sociais descentralizadas com código aberto.

Dado este contexto, foram avaliados candidatos capazes de suprir a demanda levantada. A plataforma *Friendica* foi escolhida por satisfazer tanto a necessidade de código aberto quanto ser uma rede social descentralizada. Para que haja uma interoperabilidade capaz de promover robustez entre a rede social e outras plataformas como o *Moodle*

(uma plataforma virtual de aprendizagem), foi adotado o padrão LTI (*Learning Tools Interoperability*) desenvolvido pela IMS *Global Learning Consortium*.

1.3 Motivação/Justificativa

Os discentes, ao ingressarem no curso de Ciência da Computação, se deparam com plurais adversidades oriundas das mais diversas fontes. Podemos citar o diferente modelo do processo letivo em relação ao ensino médio brasileiro; lacunas de conhecimento que serão então requeridas; grade horária distinta; e necessidade de desenvolver apressadamente novas competências informais.

Com isso, há um movimento por parte dos docentes para mitigar estes empecilhos da transição, através de projetos, trabalhos de extensão e iniciativas, como o Programa de Treinamento Especial da CAPES (PET), Guia do Calouro, sítio web "<http://boasvindas.unb.br>", o evento de Boas-vindas no Centro Comunitário Athos Bulcão (CCAB) da UnB.

Entretanto, foi percebido que há uma falta que tange os requisitos formais do graduando, ou seja, o graduando não coleta adequadamente as expectativas impostas a ele quando se formar, não lhe permitindo compreender de forma plena a motivação para certos elementos pedagógicos presentes em sua formação. Isso pode provocar uma desmotivação perante certas disciplinas e/ou processos acadêmicos, que poderia ser evitada caso houvesse uma forma de apresentar-lhe o perfil de egresso no início de sua jornada universitária.

1.4 Questão de Pesquisa

Com a motivação descrita, diversas hipóteses de como solucionar o problema podem ser levantadas. Boa parte dos anseios iniciais de solução gravitam em uma indagação central, que pode ser descrita na seguinte frase: "Como organizar as competências formais do curso para que possam ser fornecidas ao discente?".

Desta pergunta decorre a necessidade de se entender o que é uma competência neste contexto, como se dá a forma de organizá-las, qual seria uma forma de representá-las digitalmente e, não menos importante, quem é o discente alvo.

Há inúmeras formas de se definir competência e elas serão abordadas mais a frente, contudo é possível antecipar que o escopo previamente discutido leva a escolha de uma definição que se adéque aos requisitos formais advindos do perfil de egresso do curso de órgãos competentes.

Isto posto, uma forma usual e frutuosa de organizar informações, e portanto competências, é através de ontologias, pois elas permitem uma representação semântica dos termos

não só humana mas também digital, facilitando a intercomunicabilidade entre aplicações, modernizando todo o sistema a ser elaborado.

O discente alvo nos indica qual perfil de egresso estará em voga, auxiliando a balizar por quais documentos a busca pelos requisitos formais a busca será feita. Seria de grande valor estender o grupo alvo para todos os discentes de uma universidade, contudo a análise técnica de documentos formais em múltiplas áreas não é cabível para a dimensão deste trabalho. Logo, foram escolhidos os discentes do curso de Ciência da Computação.

1.5 Objetivos

1.5.1 Objetivo Geral

Este projeto tem o propósito de apresentar um modelo de competências por meio de uma ontologia decorrente de documentos formais, capaz de apresentar uma camada transparente para os discentes que permita posteriormente que aplicações inseridas no ecossistema possam armazenar e computar as competências dos discentes.

1.5.2 Objetivo(s) Específico(s)

Declinado este objetivo geral, tem-se que é necessário modelar uma ontologia das competências, um modo para persisti-la, apurar os documentos formais vigentes e definir os limites de utilização, não necessariamente nesta ordem.

1.6 Classificação do Trabalho

Segundo a classificação de Gomes[2], o trabalho está sob o paradigma projetivo, pois "a hipótese material é um sistema em desenvolvimento"[2](Página 8), trazendo conjuntamente a epistemologia do Design.

O tipo do método é dedutivo, a partir de conhecimentos prévios faz-se um novo conhecimento. Neste caso, conhecimento a respeito de competências formais brasileiras em meio digital.

Quanto a relação com o objeto de pesquisa, trata-se de um trabalho empírico em ambiente controlado, pois a análise e testes serão feitos dentro de um arcabouço tecnológico específico.

Para a profundidade dos objetivos, é um trabalho descritivo, pois há literatura consolidada na área não necessitando haver uma exploração de novos conhecimentos, assim não são elaborados novos conceitos, eles apenas são empregados em um contexto específico.

Como o próprio autor cita na Página 19[2], a área de Informática na Educação desenvolve trabalhos aplicados. Como este está totalmente inserido em tal área, pode-se dizer que também é aplicado.

Quanto ao tempo decorrido, é uma pesquisa transversal, dado que propõe uma solução num espaço único de tempo, não havendo iterações seguidas de mudanças. Contudo, há uma preocupação para que trabalhos futuros possam fazer o reuso dos artefatos aqui produzidos, tornando tal definição ligeiramente imprecisa.

Por fim, a relação entre ciência, tecnologia e mercado é de caráter tecnológico, pois não há mudança nos conhecimentos da área contudo há uma aplicação em tecnologia destes. A figura 1.1 mostra todas as possíveis denominações de uma pesquisa segundo o autor.

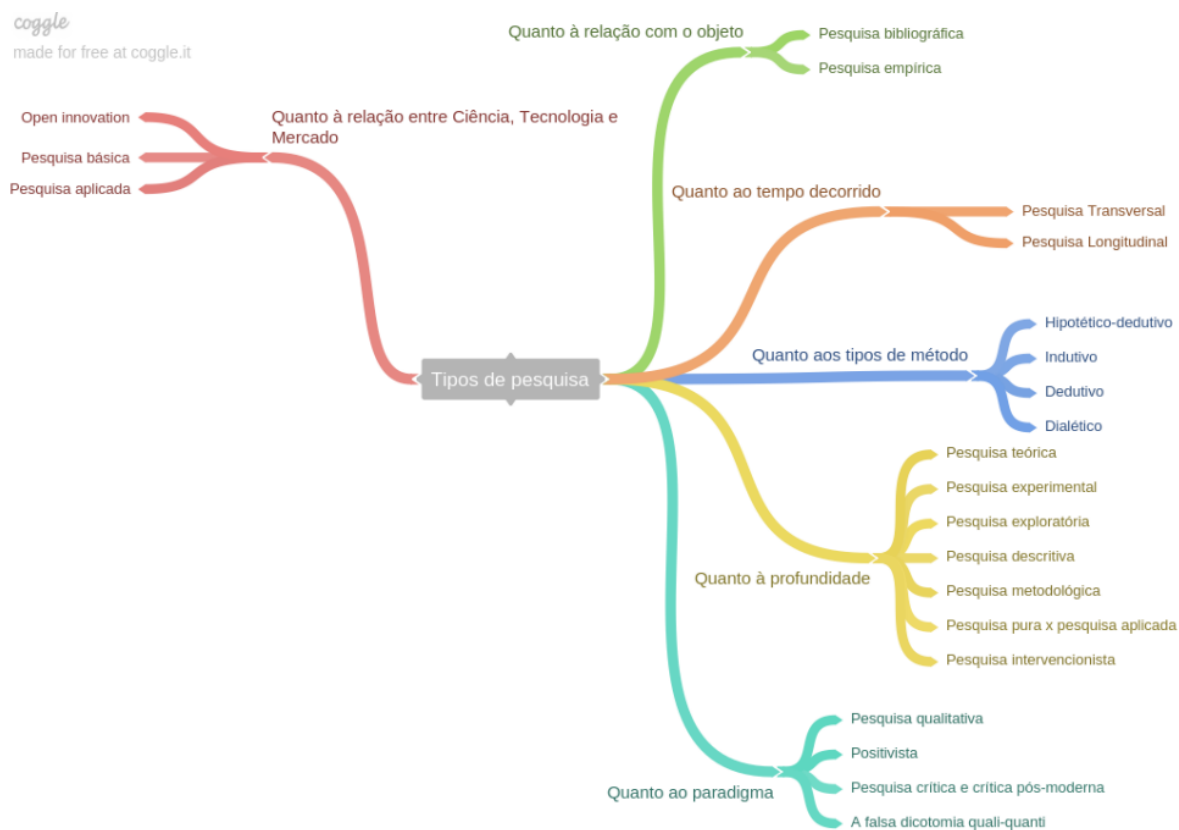


Figura 1.1: Diagrama de Gomes para a classificação de pesquisa

1.7 Escopo

A pesquisa estará focada no curso de bacharelado em Ciência da Computação oferecido pelo Departamento de Ciência da Computação (CIC) vinculado ao Instituto de Ciências Exatas (IE) da Universidade de Brasília (UnB), departamento localizado no campus Darcy Ribeiro no edifício CIC/EST.

Neste contexto temos como população alvo o grupo de indivíduos discentes deste curso, com o foco naqueles que estão nos semestres iniciais. As competências modeladas pela ontologia são aquelas postuladas por documentos oficiais do governo brasileiro e/ou pela Sociedade Brasileira de Computação para este grupo desenvolver durante sua trajetória na graduação em bacharelado em Ciência da Computação.

1.8 Metodologia

Após abordar especificamente cada termo pertinente a este trabalho, trazendo consigo suas referências e motivações, será então apresentada a metodologia utilizada para o desenvolvimento da ontologia e em seguida o contexto tecnológico e a ferramenta alvo a ser utilizada.

O método de elaboração de ontologias é chamado *Ontology 101* [3], desenvolvido no início do século vigente e amplamente utilizado desde então. Por se tratar de um método relativamente abrangente, ele se mantém atual e robusto para a utilização. Outras metodologias foram consideradas, contudo o *Ontology 101* traz consigo o uso de uma ferramenta computacional que também será utilizada neste trabalho, permitindo então uma sinergia natural.

Para a persistência da ontologia em meio computacional, foi escolhida a ferramenta *Protégé*. Ela é uma das mais importantes disponíveis gratuitamente, sendo amplamente utilizada, em especial, na área médica e biológica, contudo segue extremamente útil para todas as áreas. Ela oferece não só uma plataforma de elaboração de ontologias como uma aplicação web e um software computacional multiplataforma, sendo então uma ótima alternativa para o reuso e evolução de ontologias nela desenvolvidas.

1.9 Estrutura deste documento

Este documento apresenta cinco capítulos: O primeiro é este próprio de caráter introdutório; o segundo apontará estudos relacionados que forneceram conhecimentos sobre o domínio desta pesquisa, exemplos de elaborações de ontologias em contextos diferentes, que apontam estudos recentes na área, ou pesquisas relacionadas dentro do ecossistema deste trabalho, dentre outros motivos; o terceiro introduz e justifica o uso de termos específicos, bem como demonstra a metodologia e a ferramenta de persistência empregada; o quarto desenvolve o método apontando os passos e atitudes tomadas; e por último, o quinto avalia o processo e conclui o trabalho.

Capítulo 2

Trabalhos Relacionados

Nesta seção serão abordados trabalhos relacionados. Os motivos por serem relevantes serão organizados da seguinte maneira: aqueles que possuem como objetivo representar competências ou que discutem suas representações; os que abordam a ligação entre educação e ontologias ou aplicam uma ontologia em um meio educacional (não necessariamente competências); os que compartilham aspectos metodológicos na formação de ontologias; por serem trabalhos recentes de autores notórios na área, trazendo atualizações e/ou observações sobre as metodologias de base; os que agregam ao ecossistema de trabalhos formulados pela equipe smartUnB.ECOS, no qual este trabalho está inserido.

2.1 Modelagem de competências

Competência, como mais a frente será mais bem abordado, é um conceito deveras complexo e trazê-lo para um meio digital é desafiador. Há diversos paradigmas para conduzir essa representação, sejam eles mais abrangentes e menos específicos ou mais precisos, porém menos versáteis. Nesta seção serão abordados os trabalhos que almejavam as exprimir computacionalmente.

2.1.1 *Moodle*

Temos como exemplar de modelagem mais abrangente aquele desenvolvido pelo *Moodle*. Este ambiente, por ser um sistema de gestão de aprendizagem (*Learning Management System, LMS*) mundialmente utilizado com mais de trezentos milhões de usuários [4], é de interesse neste trabalho e procuramos nos ambientar com a solução de modelagem de competências oferecida por ele.

Sua representação de competência é, relativamente a outros modelos, genérica. Uma competência possui nome, identificador, descrição, escala, identificador da competência

pai e *framework* relacionada, como mostra a Figura 2.1, retirada através da utilização da ferramenta *Moodle Sandbox*[5]. Vemos então que o *Moodle* oferece suporte à hierarquia entre competências, bem como escalas de proficiência.

Add competency

General



Competency framework	Competency Framework 1
Parent	No parent (top-level competency) 
Name	<input type="text"/>
Description	<div style="border: 1px solid #ccc; padding: 5px;"><div style="border-bottom: 1px solid #ccc; padding-bottom: 5px;">↴ A B I ☰ ☷ 🔗 🔄 😊 🖼️ H-P</div><div style="height: 40px;"></div></div>
ID number	<input type="text"/>
Scale	<input type="text" value="Inherit from competency framework (default)"/> 
<input type="button" value="Configure scales"/>	
<input type="button" value="Save changes"/> <input type="button" value="Cancel"/>	

Figura 2.1: Inserção de nova competência em um *framework* na plataforma *Moodle*

Contudo, o *Moodle* carece de representar modelos taxonômicos heterogêneos, ou seja, modelos em a hierarquia entre competências não segue um único padrão taxonômico. Como podemos observar na Figura 2.2, pode-se somente atribuir valores para cada nível hierárquico, não podendo, portanto, existir um caso de competências nível 1 com valores taxonômicos diferentes.

Isto é um exemplo de perda de poder representativo que a plataforma apresenta, tornando-a útil para contemplar uma grande variedade de representações de competências, mas para modelos mais complexos como o proposto por este trabalho, isso acaba se tornando insuficiente. Logo, foi tomada a decisão de não utilizar este modelo, mesmo sendo oriundo de uma plataforma promissora.

2.1.2 *HR Open Standards Competency XML*

A organização *HR Open Standards* tem como principal objetivo padronizar as trocas de dados digitais relativos aos recursos humanos[6]. Dentre diversas especificações e padrões,

▼ Taxonomies

Level 1	Competency ⇅
Level 2	Behaviour ⇅
Level 3	Practice ⇅
Level 4	Domain ⇅

Figura 2.2: Níveis taxonômicos de um *framework* na plataforma *Moodle*

é oportuno apontar àquela relacionada a competências[7], como podemos ver na Figura 2.3.

Este esquema é pertinente a este trabalho, pois foi amplamente utilizado na modelagem de competências, trazendo um contexto histórico do estado da arte na época (entretanto, não necessariamente deixou de ser utilizado). Ele apresenta um modelo de competência que possui *CompetencyId*, um elemento identificador; *TaxonomyId*, um elemento que permite a descrição de uma taxonomia, esta a ser desenvolvida pelo usuário; *CompetencyEvidence*, um elemento que descreve a evidência para a competência; *AssignedWeight*, um elemento que pode representar a importância da competência em seu contexto; *Competency*, elemento que descreve outra competência, permitindo hierarquia e/ou relações entre competências; e atributos *name*, *description* e *required*, que descrevem o nome, a descrição e se é necessário ou não em seu contexto, respectivamente.

2.1.3 Gestão de Pessoas por Competências no Ambiente Aula-Net

Conduzida em 2004, a dissertação apresenta uma extensão para a ferramenta de aprendizagem virtual *AulaNet*. Sua principal funcionalidade é a representação e gestão de competências para fins administrativos[8].

AulaNet é uma aplicação desenvolvida pelo Departamento de Informática da PUC-Rio em 1997, que implementa um sistema LMS colaborativo, capaz de organizar cursos e

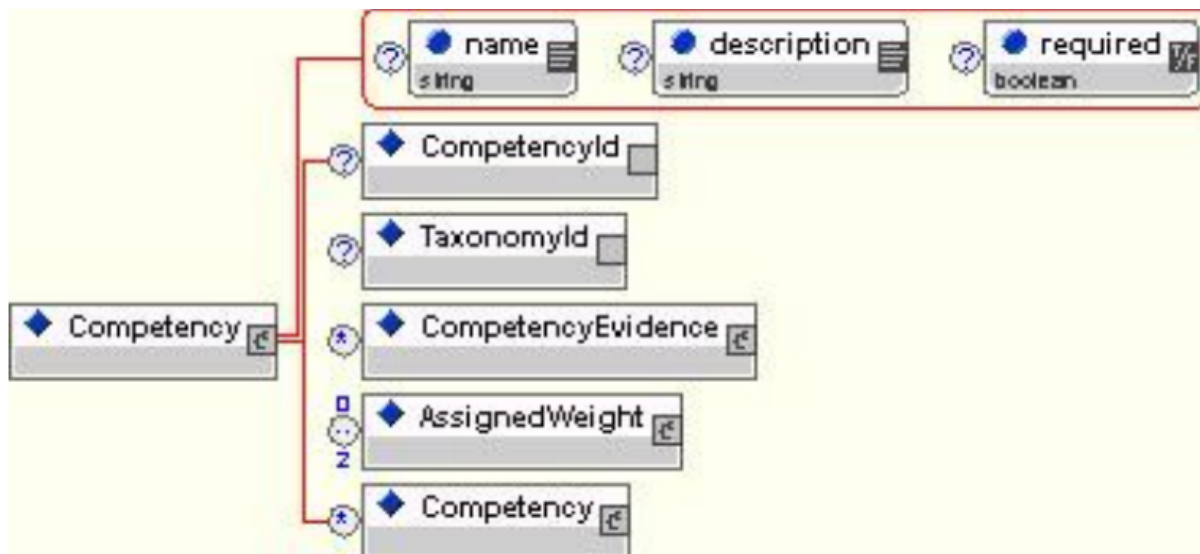


Figura 2.3: Representação visual do esquema em XML da HR Open

admitir participantes, sejam eles discentes ou docentes. Desenvolvido em *Java* e aplicado para a *Web*, tem como forte premissa a comunicação entre seus gestores.

Com isso, vemos que implementar competências pode ajudar substancialmente a integração dos usuários, logo o trabalho propôs a ontologia de competências representada pelo diagrama presente na Figura 2.4. É notória a distinção entre os verbetes ingleses *Competence* e *Competency*, algo não presente na língua portuguesa. Esta discussão será desenvolvida no capítulo 3.

2.1.4 *TENCompetence*

O projeto denominado *TENCompetence* tem como objetivo disponibilizar à pessoas, grupos e organizações serviços que auxiliem no aprendizado ao longo de suas vidas[9]. Para tal, apresenta diversos artefatos tecnológicos e ecossistema teórico.

Uma solução oferecida pelo projeto é o *Personal Competence Development Manager* (PCM)[10], um *software* de código livre, interoperável e orientado a serviços. Provendo uma infraestrutura, permite que serviços adicionais sejam acoplados garantindo que o projeto *TENCompetence* tenha margem para evolução.

Um experimento foi conduzido utilizando a plataforma *PCM* de modo a averiguar a eficácia do uso de ferramentas de gestão de competências e a influência desta em seus usuários[11]. Foi esperado que houvesse efeito positivo em seus usuários em cinco pontos: maior controle no próprio aprendizado (1), apreço pela trajetória(2) e materiais didáticos(3), no desenvolvimento de suas competências(4) e possibilidade de colaboração(5).

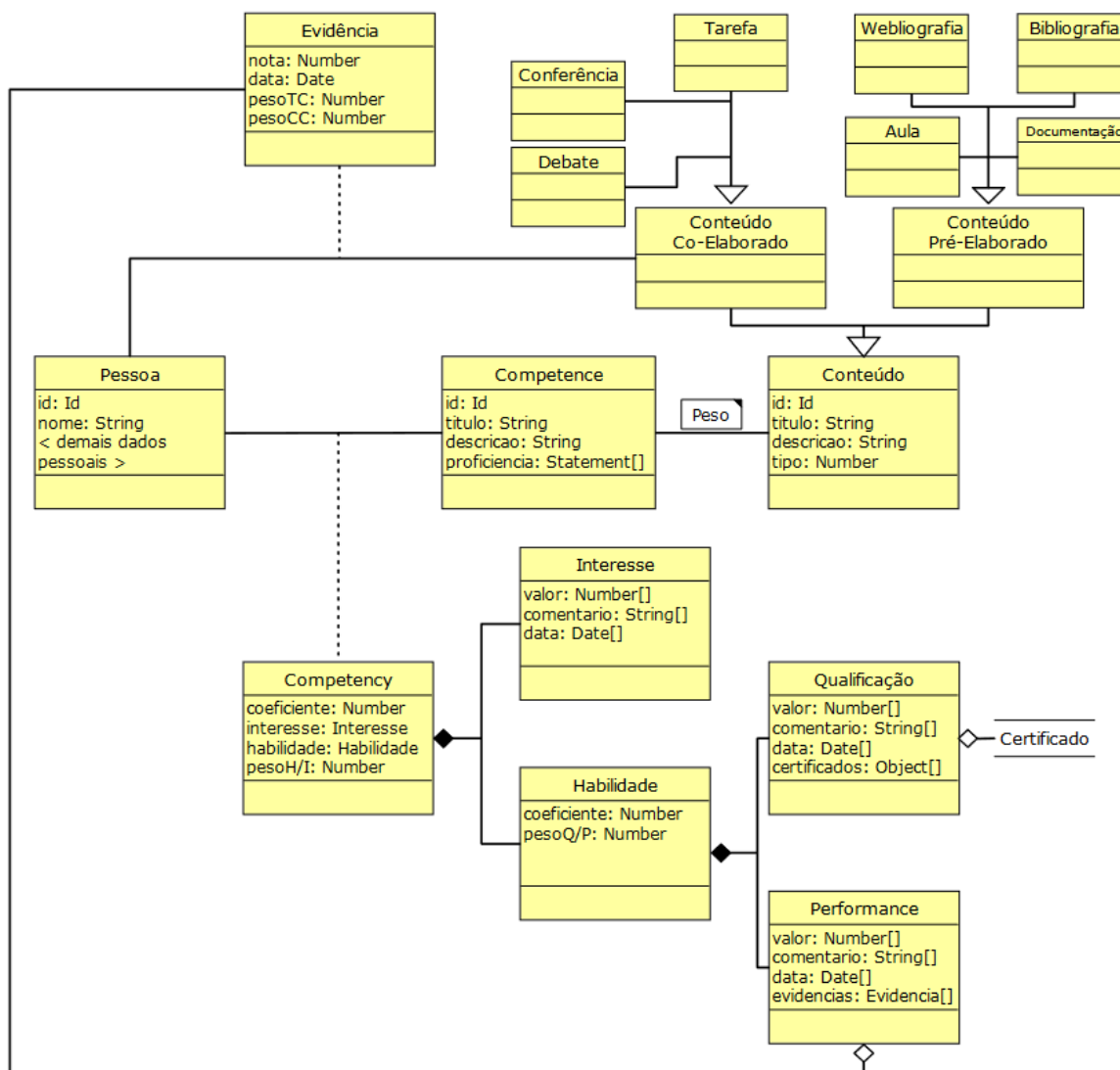


Figura 2.4: Modelo de competência da extensão do *AulaNet*

Entre o grupo controle e usuários do sistema, não foram constatadas diferenças nos pontos (2), (3) e (5), necessitando averiguação futura. Contudo, foi documentada melhora nos quesitos (1) e (4), ou seja, os participantes se sentiram com melhor controle sob seus estudos e melhor desenvolvimento de suas competências. Isto traz, portanto, um direcionamento interessante para futuros projetos envolvendo gestão de competências.

2.2 Ontologias na Educação

2.2.1 *Ontology-Based Competency Management: Infrastructures for the Knowledge Intensive Learning Organization*

Sicilia[1] tem como contexto organizações e empresas que necessitam gerir competências de seus recursos humanos, em especial aquelas críticas aos seus funcionamentos. Esta premissa levemente destoa com a deste trabalho, contudo podemos perceber certas necessidades mútuas quanto às características de competências a serem representadas.

O trabalho faz menção à ontologia do HR Consortium [7], fazendo uma interessante ressalva que é possível realizar o aninhamento de competências em *XML* para conferir hierarquia. Vemos, portanto, que há uma preocupação pelo autor em implementar tal relação.

Como podemos verificar na Figura 2.5, o autor aponta que por mais que termos como *knowledge* (saber), *skills* (habilidades) e *attitudes* (atitudes) sejam importantes para definirmos uma competência, como há uma ausência de consenso dentre as pesquisas analisadas, estes não foram inclusos como elementos básicos.

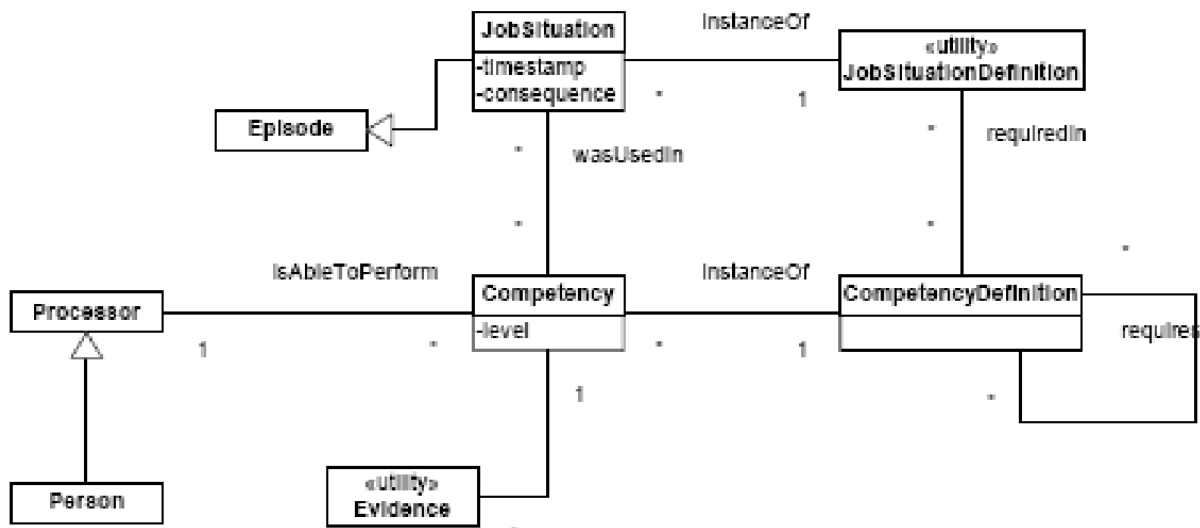


Figura 2.5: Elementos básicos de uma Ontologia em competências para Sicilia[1]

2.2.2 *Plugin de extensão de funcionalidades do Moodle*

O trabalho de título "*Extending Moodle functionalities with ontology-based competency management*"[12] (Tradução livre: Estendendo funcionalidades do *Moodle* com gestão de competências por ontologias) apresenta uma proposta de evolução do modelo nativo do *Moodle* ao incorporar informações adicionais e melhorar outras já presentes. Implementa também um artefato computacional capaz de se integrar a plataforma *Moodle* versão 2.4.

2.2.3 *Ontologies in education – state of the art*

O artigo[13] (2020) apresenta uma revisão teórica sobre ontologias na educação, bem como faz uma revisão sistemática de projetos e pesquisas contemporâneas ao artigo sobre o assunto, precedendo no máximo cinco anos.

Dentre os assuntos abordados, os principais são: a definição, os tipos mais comuns e metodologias de construção mais utilizadas de ontologias; e uma visão geral dos sistemas disponíveis que as usam no âmbito da educação.

Também, de notória importância, aborda sobre o uso de *Web Ontology Languages* (OWLs) para representação, organiza metodologias para elaboração de ontologias em três grupos, mostra gráficos sobre os usos designados a elas e artigos exemplo para cada um desses tópicos.

2.3 Generalidades sobre ontologias

2.3.1 Modelagem de uma Ontologia de Domínio com Foco em Competências para Sistemas de Recomendação na Educação

O artigo[14] propõe realizar uma integração entre Sistemas de Recomendação (SRs) e Ontologias voltadas à educação. SRs são descritos como um conjunto de ferramentas e técnicas capazes de sugerir itens de interesse do usuário. Quando voltado para a educação, como é o caso do artigo, os autores consideraram necessário acrescentar o conceito de competências.

Um ponto de interesse do artigo[14] é a metodologia empregada. Ela consiste em adaptar o método descrito pelo artigo *Ontology 101*[3], trazendo não só uma metodologia revisada e potencialmente mais adequada para a proposta de nosso trabalho como também uma versão adaptada para a língua portuguesa.

2.4 Uma revisão sistemática recente

2.4.1 *An ontology-based model for competence management*

O artigo[15] tem como proposta desenvolver um modelo baseado em ontologia para representar competências para propósito de uso geral no escopo profissional. Além disso, propõe que seja capaz de ser interoperável com outras ferramentas, provendo a capacidade de realizar consultas e operações de inferências nas competências.

A especificação de modelagem não é definida por um trabalho específico, contudo cita que se utilizou de recomendação de diversos projetos e iniciativas europeias para elaborar uma própria.

Algo notável é que uma das três principais premissas é o reconhecimento que há três formas de ser competente: formal, não formal e informal. Esta separação é importante para a *SmartUnB.ECOS*, pois o ecossistema aborda competências sob todos estes diferentes parâmetros.

Por último, o trabalho leva em conta os documentos formais do governo italiano para a educação, trazendo um ponto em comum com este trabalho, já que este se baseia nos documentos oficiais do Brasil.

Quanto à metodologia, para a ontologia, utilizaram a NeOn[16]; quanto ao sistema de gerenciamento foi utilizada um processo que define cinco grandes etapas: viabilidade, início, refinamento, avaliação e aplicação/evolução[17].

Em relação ao modelo ontológico gerado, temos de interessante a presença da decomposição da competência nos aspectos do Saber, Habilidade e Atitude. Similarmente, apresenta a diferenciação entre os termos *Competence* e *Competency*.

2.4.2 A new competency ontology for learning environments personalization

O artigo[18] possui um contexto e objetivos similares ao realizado por Miranda [15], com um foco mais direcionado à *Web Semântica*, trazendo maior cautela para que a ontologia seja facilmente lida por *softwares*. Foram analisados cinco outros trabalhos, para então produzirem a ontologia *COMP2*.

Diferente de outros trabalhos, não foi utilizado *OWLs* para a elaboração da ontologia e sim na linguagem *RDFS*. Contudo, isto não promove perda de congruência com este trabalho, pois *OWL* nada mais é uma adaptação da *RDFS*, trazendo algumas ferramentas adicionais.

2.5 Trabalhos recentes na equipe smartUnB.ECOS

2.5.1 Manutenção de portfólio de competências para cursos de graduação em computação por usuário gestor

O trabalho[19] propõe um artefato computacional capaz de fornecer uma plataforma de gestão de competências para gestores do curso de computação na UnB. Seu enfoque é lhes proporcionar a capacidade de representar as competências exercitadas durante o curso e

a partir disto manter e evoluir de acordo com as mudanças pedagógicas que fatalmente ocorrem ao longo dos anos.

A representação das competências é feita através de um modelo proposto pelo autor, baseando-se nas particularidades das informações presentes nos Referenciais de Formação da SBC RF2017. Isto posto, é de grande interesse ressaltar que a persistência destes dados foi feita utilizando um banco de dados relacional convencional, abrindo margem para se perscrutar novas tecnologias capazes de melhor persistir uma ontologia sem que haja perda de informação.

O artefato computacional também oferece suporte para interoperabilidade, ou seja, capacidade de se acoplar a outros artefatos. Isto se dá por adotar os protocolos de interoperabilidade propostos pela IMS Global denominado *Learning Tools Interoperability* (LTI)[20], em sua versão 1.3.

2.5.2 Uma ontologia de perfil holístico para estudantes de graduação

O trabalho[21] propõe a realização de uma ontologia para representar o perfil de um estudante graduando em cursos do CIC. Ou seja, deseja representar informações sobre o estudante como seu estilo de aprendizagem, seja este ativo, global, intuitivo, reflexivo, entres outros; bem como seus dados pessoais.

Em especial, pode-se verificar que foi utilizada a ferramenta de confecção e manutenção de ontologias *Protégé*, a mesma utilizada neste trabalho. A metodologia utilizada também é compartilhada por ambos, sendo ela a *Ontology 101*[3].

2.5.3 Incorporando a Edição de Competências a uma Rede Social Descentralizada para Utilização em Local de Trabalho

A proposta do trabalho[22] é de oferecer um sistema de edição de competências em redes sociais voltadas para o ambiente corporativo. A rede social alvo foi a *Friendica*, uma rede descentralizada de código aberto capaz de interoperar entre diversas outras plataformas[23].

A definição de competência utilizada é baseada naquela da *engageGrid*[24], projeto que apresenta um conjunto de serviços que proporciona aos usuários uma rede de compromissos na esfera profissional. A persistência se deu através do sistema *ARC2*, capaz de lidar com ontologias descritas em linguagem RDF/XML e armazená-las em bancos de dados relacionais.

Capítulo 3

Fundamentação para a Proposta

Para permitir que este trabalho consiga satisfazer o objetivo, é de suma importância que a fundamentação teórica e entendimento das ferramentas utilizadas esteja evidente. Nesta seção, expomos a fundamentação teórica e tecnológica da proposta, divididas em quatro subseções.

Primeiramente, foram abordadas as definições de termos importantes para o desenvolvimento deste projeto. Cabe ressaltar que o mérito das definições não foi contemplado durante a escolha. Por vezes, a escolha tinha como objetivo apenas o alinhamento com textos pertinentes a este, logo não se deve induzir que as definições adotadas estão em consonância com o exato estado da arte por padrão.

Na segunda subseção, encontram-se os documentos que trazem as competências formais relacionadas com o escopo deste trabalho. Foi feita uma trajetória do processo de elicitação deles, trazendo opções e alternativas que não foram utilizadas, embora sejam relevantes.

Em terceiro, a seção de modelagem traz a bagagem teórica que permeou o processo de criação da ontologia, aportando portanto o método *Ontology 101* traduzido livremente e com comentários onde for necessário. Com isso, será possível dar continuidade ao trabalho e iniciar o uso da ferramenta tecnológica.

Por último, foi elucidado o contexto tecnológico em que a ferramenta de desenvolvimento de ontologias está inserida. Naturalmente também foi discutido o uso da ferramenta em si.

3.1 Definições

Visto a abrangência de temas que permeiam a pedagogia e, por consequência, a educação aplicada à informática, há diversos usos para as mesmas palavras, podendo estas represen-

tar diferentes conceitos. Assim, faz-se necessário desarmar as ambiguidades que surgem em virtude desta amplidão, em especial para palavras-chave desta pesquisa.

3.1.1 Ontologia

Ontologia, segundo o dicionário *Michaelis*, pode se referir ao "*estudo ou conhecimento dos seres e dos objetos enquanto eles mesmos, em oposição ao estudo de suas aparências e atributos.*"[25]. Tal definição nos ajuda a entender a etimologia detrás do uso que será feito neste trabalho.

Contudo, esta escolha de definição é deveras abrangente e não nos auxiliam para com o objetivo proposto. Tendo em vista a escolha da metodologia, é de interesse promover o alinhamento com ela, já que este trabalho não tem como objetivo realizar análises semânticas. Com isso, foi adotada a própria definição presente na obra da metodologia[3] (Página 3):

ontology is a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (sometimes called role restrictions)).

Cabe ressaltar que tal definição já está inserida no contexto informático, e não é uma definição que possa ser aplicada diretamente em outras áreas como a própria filosofia. Inclusive, deve-se atentar que está na língua inglesa, portanto a tradução foi conduzida através da tradução literal, aliada com o uso presente na obra de Simon et al [14], que utiliza a mesma metodologia.

Portanto, temos que uma ontologia é uma definição formal de conceitos presentes em um escopo contendo classes (classes), suas propriedades (slots) e descrições (facets). Classes se referem a algum elemento do domínio, as propriedades trazem seus atributos e descrições se remetem às informações relativas às propriedades. Cabe ressaltar que propriedades também podem se dar na forma de relacionamentos entre classes, resultando em hierarquia de classes, algo importante para o desenvolvimento de ontologias. Como apontado por Simon et al[14], tais definições são interessantes por estarem também alinhadas com a ferramenta *Protégé*.

3.1.2 Competência

Este termo é decerto maleável perante o contexto que está inserido. Contudo, em observância ao escopo deste trabalho, é interessante utilizar a definição presente no RF2017[26] (Página 10): "Ter competência é a capacidade de um indivíduo em mobilizar recursos,

tais como conhecimentos, habilidades, atitudes e valores, para a sua atuação em situações reais complexas."

Baseando-se na taxonomia apresentada por Bloom[27], foi sugerida uma taxonomia revisada por Anderson[28] que apresenta seis níveis de capacidade: lembrar[1], entender[2], aplicar[3], analisar[4], avaliar[5] e criar[6].

Estes níveis foram sintetizados em três **categorias**: conhecimento, competência e habilidade[29]. Conhecimento engloba os níveis [1] e [2], competência [3] e [4] e habilidade [5] e [6]. Para evitar ambiguidade com o conceito de competência empregado neste trabalho, foi feita a mudança sintática do termo competência aqui descrito para atitudes, como também realizado por Simon[14].

Estas categorias possuem uma relação direta com termos descritos nos RF2017, ligando a categoria conhecimento a "saber", atitude a "saber fazer" e habilidade a "saber ser".

Para fins de exemplo, exercita-se o seguinte cenário: ao concluir através da ontologia que um discente é competente em "Conhecer os limites da computação"(Um exemplo de competência formal), podemos esperar que este aluno saiba o porquê e quais são as limitações de uma computação (saber), consiga determinar se um problema específico é ou não computável (saber fazer), e ao ter estes conhecimentos, agir de forma a evitar se empenhar com problemas não computáveis e empregar suas outras competências apenas em problemas computáveis (saber ser).

Sampson e Fytros[30] nos trazem algumas ponderações e sugerem algumas características para descrever uma competência: **Nome**, **Descrição**, **Proficiência** e **Contexto**. Nome e Descrição são autoexplicativos. Proficiência deverá remeter a uma **Escala**. O Contexto deverá informar o âmbito no qual a competência está inserida. Em exemplo, empresarial, acadêmico ou social.

Um aspecto importante de uma competência é sua **Natureza**, pois ela pode ser formal, informal ou não-formal. Há uma discussão acadêmica quanto a estes termos, principalmente em virtude da distinção entre não-formal e informal, como aponta Gadotti[31], mas não é pertinente a este trabalho. Contudo isso não exime a responsabilidade deste em ser capaz de representar tal característica de uma competência.

Para que uma competência de um indivíduo seja acessível e mensurável, deve-se atribuir às competências uma forma de evidenciá-las. Logo, o modelo também deve contemplar **Evidências**. Elas poderão advir de diversas fontes, contudo este trabalho traz apenas competências formais, logo as evidências serão aprovações em disciplinas oficiais.

Há uma diferença semântica na língua inglesa entre as palavras *Competence* e *Competency*. A primeira remete a capacidade de ser efetivo no cumprimento de tarefas, ou seja, ter *Competence* diminui os custos operacionais (seja por exemplo tempo ou material) e

aumenta a efetividade de uma dada ação. A segunda traz a noção de um conjunto de características que quando unidas promovem um aumento de performance [32].

Logo, pode-se observar que *competency* não possui um agente enquanto *competence* sim. Tornando este o principal fator para o modelo: *competence* será o objeto no qual discentes se relacionarão, enquanto *competency* será um elemento que se relacionará conceitualmente com *competence*. No ambiente *AulaNet* não foi feita a tradução, como vemos na Figura 2.4, contudo para melhor leitura será adotado o termo Competência para *competency* e Competência Efetiva para *Competence*.

Competência de Formação

Será definida como uma competência de formação aquelas presentes nas Diretrizes Curriculares Nacionais, homologadas em novembro de 2016, por meio da Resolução N°05 de 16/11/2016[33]. Estas competências formais são as esperadas de um discente egresso em bacharelado em ciência da computação. Esta definição visa diferenciar a origem da competência ao longo deste trabalho.

Competência Derivada

Competências Derivadas estão presentes no RF2017, são aquelas presentes em um eixo de formação. Elas portanto pertencem à um eixo e se relacionam com conteúdos de forma taxonômica descrita anteriormente. Há também uma relação direta com as Competências de Formação feita pelos autores, facilitando o relacionamento entre os documentos.

3.1.3 Conteúdo

Também descritos como Conteúdos Curriculares[34] e declarado como sinônimo de *matéria* no RF2017, são elementos de conhecimento a serem absorvidos pelos discentes. Não foi apresentada uma definição por nenhum dos documentos formais, logo infere-se que o uso comum da palavra esteja sendo utilizado: conjunto de informações que pertencem a um tema.

Extensivamente declarados pelo Ministério da Educação[34], não há uma definição de cada conteúdo específico. Logo, deve-se exercitar a razoabilidade e consenso entre o corpo docente para determinar a exata expressão de cada conteúdo.

3.2 Base Formal

Durante a pesquisa por modelos de representação de competências e suas instâncias, um impasse se demonstrou presente: devem eles adequar aos moldes formais apresentados por instituições internacionais ou nacionais?

Visto a natureza do trabalho com seu contexto nacional, optou-se por se ater aos documentos nacionais. Apesar disso, é interessante apresentar todos os documentos que foram apreciados.

3.2.1 Fontes Internacionais

A instituição *Institute of Electrical and Electronics Engineers* (IEEE) possui grande notoriedade no âmbito internacional, logo foram pesquisados documentos por ela gerados na questão de competências e ontologias.

O projeto P1484.20.2 *Recommended Practice for Defining Competencies*[35] mostrou-se promissor, pois baseia-se no padrão IEEE 1484.20.1-2007 *IEEE Standard for Learning Technology-Data Model for Reusable Competency Definitions*[36], que apresenta um modelo de dados de competências capaz de realizar diversas operações; contudo estes documentos estão indisponíveis de forma gratuita, e não possuem parceria com o portal de periódicos da CAPES por acesso via Comunidade Acadêmica Federada (CAFe).

A IEEE, em conjunto com a ACM, também oferece um guia para currículos de graduações em computação[37], que se apresentou um forte candidato para a elaboração deste projeto. Verificamos que já existem trabalhos que propõem ontologias para este guia, como a apresentada por Pietra (2018), tornando esta linha de pesquisa interessante.

Uma instituição também importante internacionalmente é a *Association for Computing Machinery* (ACM), com o notório documento atualizado sobre o currículo de cursos em computação[38]. Possui definições e modelo de competências (p.44-54), contudo não foi utilizado para este projeto, pois, no momento de seu acesso, a decisão de se manter em escopo nacional já havia sido feita.

3.2.2 Fontes Nacionais

Para ater-se ao padrão nacional, o sítio web da Sociedade Brasileira de Computação foi acessado e foi encontrado o documento "Referenciais de Formação para os Cursos de Graduação em Computação SBC (RF-CC-2017)"[26]. Este documento foi amplamente utilizado para elaboração deste trabalho.

O RF2017 é fortemente relacionado às Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação MEC[33], documento que apresenta os padrões a

serem seguidos por instituições de ensino superior em âmbito nacional para implementação de cursos de graduação na área da computação.

3.3 Modelagem

Uma ontologia, como discutido anteriormente, é o estudo dos seres e objetos em sua essência. Para que seja possível realizar tal estudo em um âmbito computacional, é necessário que seja feita a representação deles digitalmente.

Para tal, não se pode tratar de forma leviana o método e processo para transportar algo da metafísica para o físico. Logo, abordaremos nas próximas seções o método e as ferramentas utilizadas para transportar as competências no domínio das ideias para uma representação computacional.

3.3.1 Ontology 101

Para que o ecossistema, no qual este projeto está inserido, possa se relacionar com este projeto é necessário que exista uma forma clara para se implementar um modelo de competências em um ambiente computacional. Com isso, percebe-se a necessidade de se adotar um método consistente e capaz de produzir um modelo de competências congruente e alinhado com o contexto formal.

Há diversas formas de representar conhecimento digitalmente. Contudo, espera-se do trabalho um modelo capaz não só de ser humanamente compreensível como por máquinas, já que é interessante promover a integração entre aplicações no SmartUnB.ECOS. Isso torna *Ontology Web Language* (OWL) uma solução interessante (o conceito de OWL será desenvolvido na próxima seção). Por consequência, percebe-se a necessidade de se modelar uma ontologia.

Portanto, para o desenvolvimento de uma ontologia capaz de representar competências, será utilizado o roteiro descrito por Noy e McGuinness intitulado *Ontology 101*. Nele, os autores explanam o contexto do método e determinam sete passos a serem seguidos para a elaboração.

Foi ponderado o uso de outras metodologias como a *NeOn*[39]. Este trabalho se enquadra no cenário dois desse método, pois já existe um arcabouço não ontológico (os documentos RF2017 e DCN2016). Também foi considerado a metodologia *Methontology*[40], notório método amplamente utilizado. Contudo, definiu-se que o uso da 101, não só por ser amplamente testada mas por ser um método endossado pela documentação da ferramenta *Protégé*[41]. Logo tal adoção traz uma melhor consonância entre as partes do nosso trabalho.

Parte da metodologia 101 advém da literatura de desenho em orientação a objetos, contudo os autores ressaltam que a terminologia compartilhada são falsos cognatos. Semelhantes, porém com significados distintos, palavras como classe ou atributo desempenham papéis ligeiramente diferentes no esquema resultante dos métodos.

Antes de expor os passos, os autores[3] apontam três regras fundamentais que devem permear todo o processo. A primeira anuncia que não há um único modelo viável resultante, com o melhor resultado estando a depender da aplicação da ontologia; a segunda, que o desenvolvimento é necessariamente iterativo, ou seja, deve-se revisitar os passos anteriores para realizar o desenho; a terceira, que os conceitos na ontologia devem estar bem próximos aos objetos e relacionamentos, verificando que sentenças descritivas do domínio quase sempre têm os substantivos sendo traduzidos a objetos e verbos relacionados (por exemplo, "os alunos possuem competências" sugere que a ontologia apresentará objetos chamados alunos e competências e um relacionamento de *possui/éPossuídoPor* existirá entre tais objetos).

Após esta introdução ao método, os autores[3] expõem os passos a serem seguidos para a construção da ontologia:

Passo 1: Determinar o domínio e escopo da ontologia. O autor elabora quatro perguntas chave que devem auxiliar a determinação do domínio e escopo. as perguntas são: *Qual o domínio que a ontologia irá cobrir? Qual o uso da ontologia? Para quais perguntas as informações contidas na ontologia fornecerão respostas? Quem usará e manterá a ontologia?* O autor também sugere elaborar questões de competência para averiguar se a ontologia resultante está de fato sendo suficiente para o escopo.

Passo 2: Considere o reuso de ontologias. O passo é relativamente autoexplicativo, por vezes o domínio e/ou escopo alvo já foi explorado e devidamente desenhado em uma ontologia. Basta então realizar adequações, não necessitando a construção do zero. Pode acontecer também do uso da ontologia já estar inserida em um contexto que já possui outras ontologias, logo para haver interação é necessário se utilizar delas e/ou de seus vocabulários.

Passo 3: Enumerar termos importantes para a ontologia. O método sugere que se elabore uma lista de termos que são de interesse em se deixar claros ou importantes que o usuário da ontologia os conheça. De início, não é necessário se preocupar se os termos representam classe, propriedades ou atributos, ou mesmo se os termos representam conceitos parecidos.

Passo 4: Definir as classes e a suas hierarquias. Este passo possui diversas abordagens válidas, e o autor cita três destas: "de cima pra baixo", "de baixo pra cima" e "uma combinação destas". No primeiro caso, define-se as classes mais genéricas e então se elabora as mais específicas em seguida. No segundo, descreve-se as específicas primeiro

para então prosseguir generalizando. No terceiro, elabora-se as classes mais genéricas bem como as mais específicas, para aí então traçar uma estratégia para conectá-las.

Segundo os autores[3], não há abordagem superior, deve-se avaliar qual a melhor estratégia levando em consideração o domínio e até mesmo a preferência do desenhista. Havendo a escolha da abordagem, deve-se começar então com a definição das classes. Utiliza-se a lista elaborada no Passo 3, coletando os termos que existem por si só e não são descrições destes.

Passo 5: Defina as propriedades das classes. Como a classe sozinha é incapaz de responder as questões elicitadas no Passo 1, há a necessidade de descrever a estrutura interna dos conceitos relacionados a elas. Dos elementos da lista elaborada no Passo 3, aqueles ainda não utilizados muito provavelmente são propriedades das classes. Neste passo devemos relacionar tais propriedades às suas respectivas classes. Estas podem ser *Object Properties*, que relaciona classes entre si, não possuindo restrições lógicas entre domínio e imagem, permitindo portanto relacionamentos entre a mesma classe ou até mesmo entre múltiplas classes. Ou *DataProperties*, análogo a atributos, sendo, portanto, elementos que relacionam valores finais como inteiro ou sequência de caracteres à classes.

Passo 6: Definir as descrições das propriedades. Neste momento deve-se determinar os tipos de valores aceitos por uma propriedade, seus limites, a quantidade de cada um por classe (cardinalidade) etc.

Destas descrições possíveis, o autor enfatizou a cardinalidade, quantos valores uma propriedade pode ter; o tipo de valor, como número, booleano, caracteres ou até mesmo uma outra classe; os limites de valores, como um número não poder ser negativo ou uma sequência de caracteres não ser maior que 30.

Em sistemas onde acoplar uma propriedade a uma classe é equivalente a adicionar a classe ao domínio da propriedade, ou seja, a propriedade é uma entidade independente de classe e pode ser reutilizada em outras classes, é interessante deixar as descrições genéricas o suficiente para satisfazer o caso mais geral. Visando isso, é de interesse também sempre colocar as propriedades e suas descrições na parte mais alta da hierarquia sem que haja perda de coerência no domínio.

Passo 7: Criar instâncias. Neste momento, deve-se povoar a ontologia com exemplos reais aos quais ela se propõe a descrever. No nosso contexto, seria equivalente a colocar as competências presentes no RF2017 e DCN2016 dentro da ontologia. Este passo é importante, pois as questões levantadas no Passo 1 podem ser testadas e então verificar se a ontologia é de fato capaz de descrever aquilo que ela propõe.

Após a elaboração da ontologia, deve-se procurar uma forma de representá-la computacionalmente a fim de contemplar o objetivo de nosso trabalho. Para tal, foi escolhida a ferramenta *Protégé* descrita a seguir.

3.4 Protégé

A ferramenta *Protégé* é um software gratuito desenvolvido pela *Stanford Center for Biomedical Informatics Research* localizado na *Stanford University School of Medicine*[42].

Desde o ano 2000, foi desenvolvido em Java garantindo a capacidade multiplataforma. Há também uma versão para web chamada *WebProtégé*. Atualmente encontra-se na versão 5. A versão e subversão utilizada neste trabalho é a 5.5.0.

O uso principal desta ferramenta é promover o desenvolvimento de ontologias e instâncias. Ontologias, por sua vez, são representadas através da linguagem OWL. Esta linguagem foi elaborada para ser possível construir ontologias inseridas num contexto de *Web Semântica*.

Por possuir uma interface gráfica, não é necessário que o usuário domine a linguagem OWL, em específico a OWL2 definida pela W3C[43], linguagem utilizada pela aplicação.

Um aspecto importante do ecossistema de OWL é a utilização de ferramentas denominadas *Reasoners*. Elas analisam os elementos de uma ontologia e averiguam a consistência interna, sinalizando ao projetista erros que possivelmente faria um leitor eletrônico ser induzido ao erro. A versão do *Protégé* utilizada é acompanhada pelo *Reasoner* HermiT em sua versão 1.4.3.456.

A ferramenta apresenta a possibilidade de incorporar aplicações acessórias, também conhecidas como *Plugins*. Isso torna o uso mais robusto, permitindo maior personalização. Este trabalho irá utilizar em maior escala o *Plugin* OntoGraf. Ele permite a geração de diagramas visuais da ontologia modelada utilizando as classes como nós e as propriedades de objeto como arestas.

Capítulo 4

Proposta

4.1 Modelagem da Ontologia

4.1.1 Aplicação do Método

Passo 1: Determinar o domínio e escopo da ontologia. O domínio da ontologia é o corpo discente do curso de bacharelado em Ciência da Computação da Universidade de Brasília, tendo em seu escopo as competências formais destes. A ontologia será utilizada pelos próprios discentes para consulta e gerência de suas próprias competências. Cabe ressaltar que isso não impede de a ontologia ser capaz de ser utilizada em domínios e escopos relativos, como o curso de Engenharia da Computação ou representar competências informais, seja através do uso direto ou após sujeita a breves alterações.

Perguntas que a ontologia deve responder são: Quais competências o aluno possui? Qual o nível de proficiência do aluno para uma certa competência? O estudante adquiriu uma competência de eixo dadas suas competências derivadas? Qual a natureza de uma certa competência? Qual evidência valida a competência do aluno? Qual o contexto de uma competência? Ao concluir a disciplina de estruturas de dados, quais competências serão adquiridas? Quais são as competências de um curso?

Passo 2: Considere o reuso de ontologias. Por ser específico para os documentos formais do curso de Ciência da Computação, não houve ontologia pesquisada que fosse capaz de exprimir as nuances das informações pertinentes a este contexto, bem como as competências em computação neles descritas.

Decerto haverá grande influência de certas ontologias, como as presentes no trabalho de Miranda *et al* [15], Paquette [18], por não só compartilhar boa parte dos conceitos mas também estarem inseridos no âmbito educacional representado computacionalmente.

Passo 3: Enumerar termos importantes para a ontologia. Alguns termos a seguir já estão presentes na fundamentação teórica, enquanto outros serão introduzidos.

Para eles, se cabível, será feita uma breve descrição. Alguns termos foram omitidos pois descreviam relações relativamente genéricas como "possui", acarretando uma lista prolixa. Segue a lista:

Competência, Competência Efetiva, Evidência, Eixo de Formação, Taxonomia, Discente, Curso, Disciplina, Conteúdo, Competência Derivada (RF2017), Competência Geral(DCN2016) e Contexto.

No que tange a uma competência, relaciona-se palavras como Escala, é composta por, é equivalente a, Tipo, Formal, Informal, Não-Formal, Contexto. Para a competência Efetiva, temos também "é instância de".

Quanto a Escala, temos também Valor, Mínimo, Máximo e Descrição.

Passo 4: Definir as classes e a suas hierarquias. Por ser um contexto específico com o escopo relativamente estrito, não há grande margem para a elaboração de hierarquia. Logo, as classes tendem a ter relações horizontais. Isso acarreta não haver necessidade de determinar uma abordagem, como *Top-down* ou *Bottom-up*.

As classes a serem modeladas são: Competência, Competência Efetiva, Evidência, Discente, Curso, Disciplina, Tipo, Contexto, Escala.

Passo 5: Defina as propriedades das classes. & Passo 6: Definir as descrições das propriedades. Por estes passos estarem relacionados, para fins de brevidade, serão apresentados simultaneamente, mesmo não tendo sido desenvolvidos de tal forma.

Como pontuado no passo três, algumas das propriedades não estão citadas nominalmente na lista devido a serem genéricas quando não inseridas em uma relação de classe. As propriedades que não são relacionamentos também não foram apresentadas, pois são pouco numerosas e apenas descrevem propriedades genéricas como nome e descrição. Alguns termos aqui podem aparecer mesmo não estando presentes na lista. Relacionamentos inversos também foram omitidos (como por exemplo, "possui" e "é_possuído_por"). Seguem as classes e suas propriedades:

Competência: **é_composta_por** e **é_equivalente_a**, estas se relacionam a outras zero ou mais Competências, conferindo a possibilidade de associação entre elas, uma característica desejável[18]; **Possui_tipo**, se referindo a uma classe Tipo, permitindo ser categorizada por formal, informal e não-formal; **é_contextualizado_por**, possuindo zero ou mais Contextos, serve para evitar ambiguidade, por exemplo, traduzir pode se remeter a línguas humanas ou computacionais, contextos que envolvem conhecimentos diferentes. Tipo, sendo este uma propriedade de dados de tipo *string*, podendo ser "formal", "informal" ou "não-formal".

Competência Efetiva: **é_instância_de**, trazendo uma referência a competência relacionada; **é_categorizado_por**, que relaciona a uma classe de Taxonomia; **é_validado_por**,

relacionando com zero ou mais Evidências; **possui_nível**, relacionando com zero ou uma Escala.

Evidência: **é_contextualizado_por**, possuindo zero ou mais Contextos;

Discente: **curso**, relacionando com zero ou mais Disciplinas; **possui_competência**, relacionando com zero ou mais Competências Efetivas.

Curso: **possui_eixos**, relacionando com zero ou mais Eixos de Formação.

Disciplina: **possui_nível**, relacionando com zero ou uma Escala, descrevendo a nota obtida pelo Discente; **abrange**, relacionando com zero ou mais Conteúdos.

Conteúdo: **aborda**, relacionando com zero ou mais Competências.

Contexto: Por ser uma classe descritiva, não há nenhuma propriedade de objeto não inverso partindo de Contexto.

Escala: Por ser uma classe descritiva, não há nenhuma propriedade de objeto não inverso partindo de Escala.

Taxonomia: Por ser uma classe descritiva, não há nenhuma propriedade de objeto não inverso partindo de Taxonomia. Contudo vale ressaltar que pode apresentar como propriedade de dados valores relativos aos níveis taxonômicos presentes em [28] ou em [29]. Cabendo também evolução para outros projetos, podendo abarcar outras definições.

Passo 7: Criar instâncias. Devido ao fato da persistência se dar em uma ferramenta, este passo será feito concomitantemente com a próxima seção do capítulo.

4.1.2 Modelo Resultante

A aplicação da metodologia 101 foi feita em meio analógico com papel e caneta. Por se tratar de um meio não definitivo, houve o descarte do modelo resultante direto, contudo, a Figura 4.1 apresenta o modelo parcial gerado na ferramenta *Protégé*.

A imagem não traz as propriedades de dados das classes pois comprometeria a leitura e eles foram melhor abordados na próxima seção.

4.2 Persistência no Protégé

Com o desenho da ontologia feito, deve-se então persisti-lo em meio digital. A ferramenta utilizada é chamada Protégé. Ela possui uma versão em linguagem Java multiplataforma e outra versão Web. Será utilizada a versão Java no ambiente Windows.

A tela inicial do programa pode ser visualizada na figura 4.2. Dentre diversas funções, as de interesse explicativo presentes no menu superior são o sub-menu *File*, capaz de salvar e carregar ontologias, *Reasoner* onde há as funções de escolha de um motor de inferências e seus comandos e *Windows*, permitindo a inicialização de *plugins*.

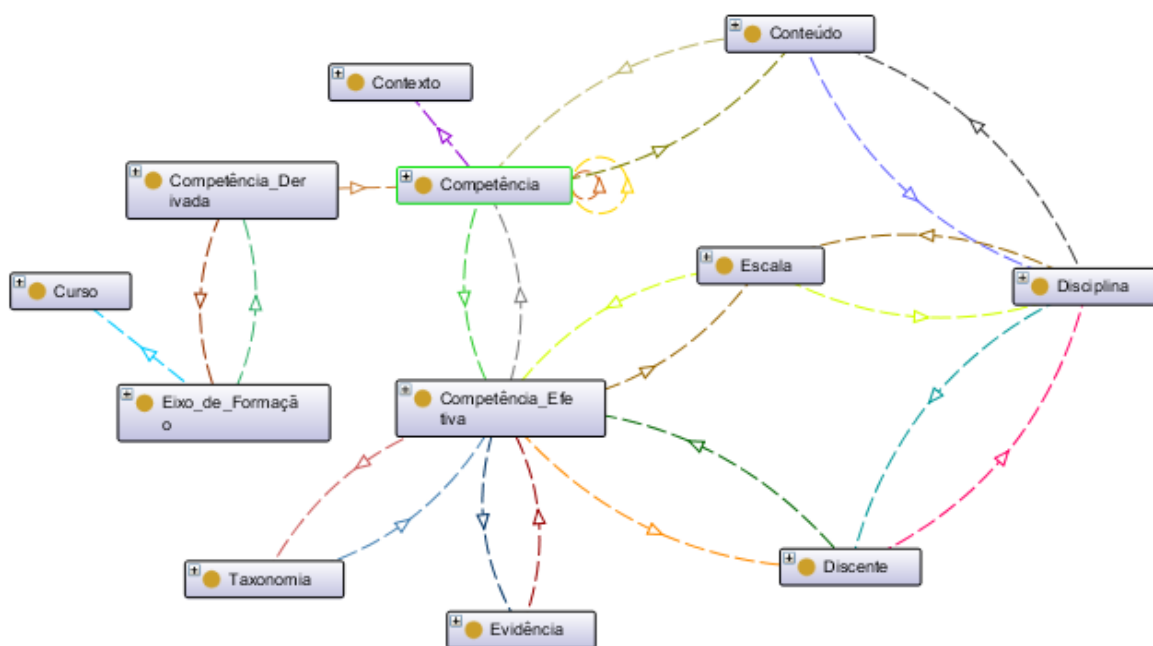


Figura 4.1: Diagrama de Classes resultante, gerado pelo *Plugin OntoGraf*

Vemos mais abaixo as guias *Active Ontology(1)*, *Entities(2)*, *Individuals by Class(3)* e as abas referentes aos plugins em execução. Cada uma delas permite:

(1) Visível na Figura 4.2, mostrando informações gerais sobre a ontologia que está sendo trabalhada.

(2) Como visto na Figura 4.3 permite a declaração e modificação de entidades, sejam elas classes, propriedades de objetos, propriedades de dados, metapropriedades, tipos de dados e indivíduos (objetos).

(3) Permite manipular os indivíduos, também conhecido por objetos em um paradigma de orientação a objeto. Nele os indivíduos são agrupados por classe, facilitando a organização e manipulação.

A Figura 4.1 mostra as classes que foram inseridas no modelo, e cada seta representa uma propriedade de objeto que relaciona duas classes. Para maiores detalhes de cada nome e relação, segue em anexo o RDF da ontologia, a fim de carregá-lo no software Protégé.

4.3 Testes das questões de competência

O passo 1 da metodologia 101 prevê a realização de perguntas para que se permita a validação do modelo. Portanto, aqui serão demonstradas as formas de se responder tais

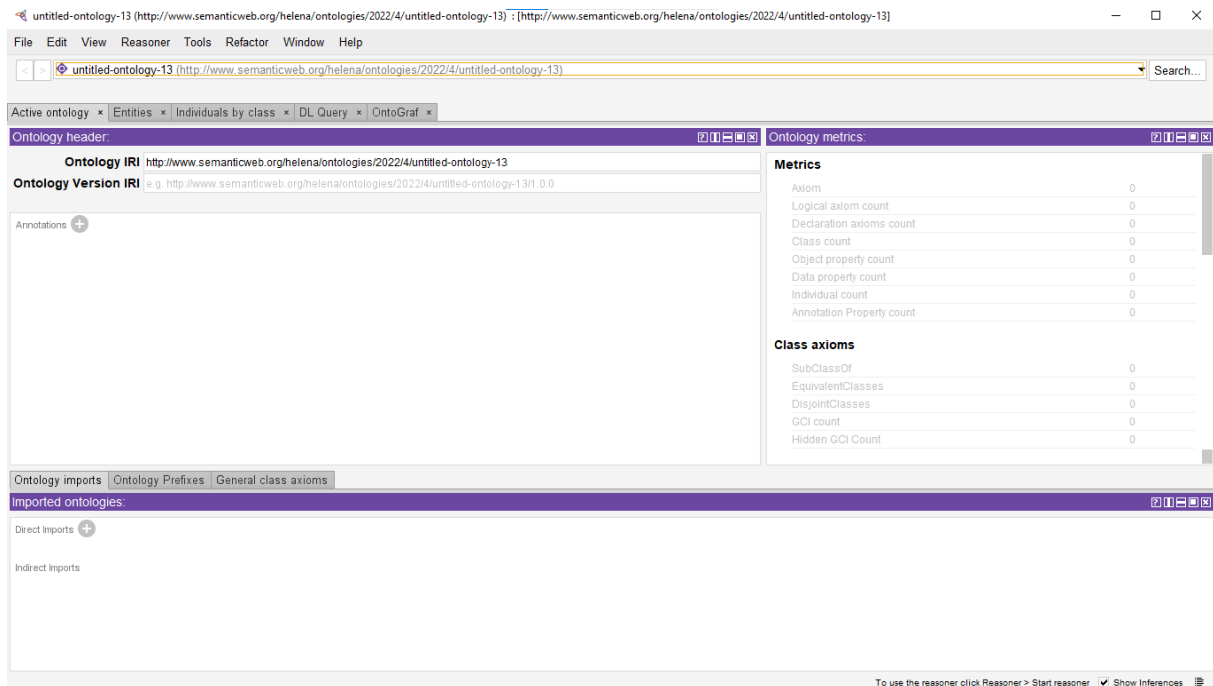


Figura 4.2: Tela inicial da versão Java do Protégé

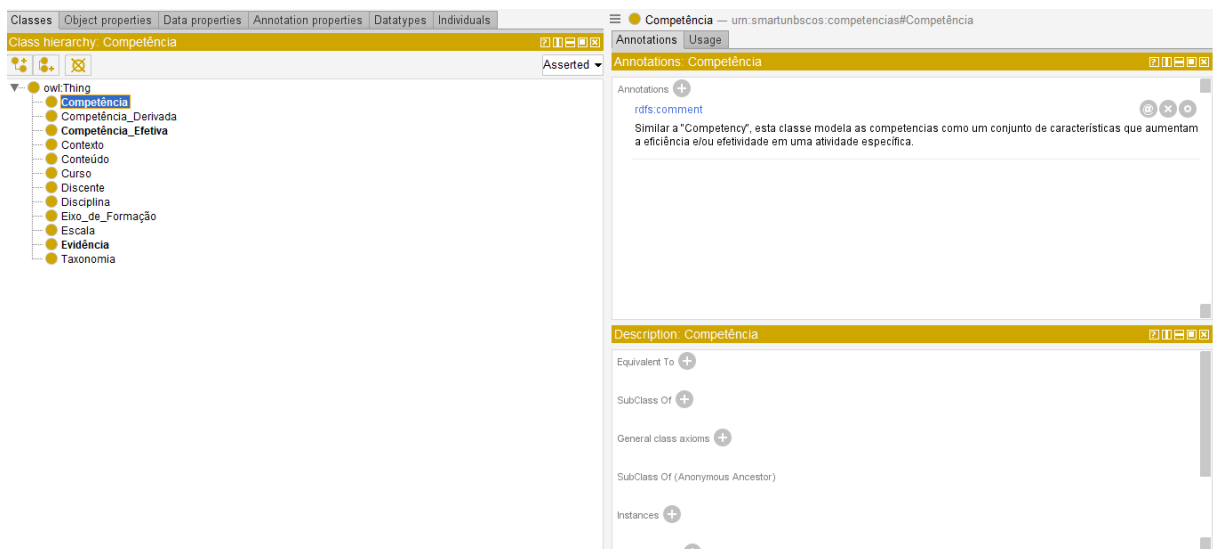


Figura 4.3: Aba de Entidades no Protégé

perguntas feitas utilizando o modelo gerado.

As perguntas e respostas estão em alto nível, pois a forma exata depende do meio computacional em que a ontologia será utilizada. Por exemplo, o próprio Protégé fornece *Plugins* como SQWRLTab e a aba DL Query que são capazes de acessar as instâncias. Contudo, em um ambiente de programação Java, por exemplo, há opções como a API Jena[44] para interagir com os dados.

Quais competências o aluno X possui?

Para responder, deve-se selecionar todos os relacionamentos "possui_competência" do "Discente" X. Devendo então mostrar a lista de "Competência_Efetiva" relacionada.

Qual o nível de proficiência do aluno X para uma certa competência C?

deve-se procurar pelo dado "valor" do indivíduo "Escala" que possui relacionamento "possui_nível" com "Competência_Efetiva" C relacionada por "possui_competência" com o "Discente" X

O estudante X adquiriu uma competência de eixo dadas suas competências derivadas?

para obter esta resposta, se baseando na Figura 4.4 para melhor entendimento, deve-se verificar a interseção entre as "Competências" que "é_metaclassa_de" (laranja) "Competências_efetivas" que "pertence_a" (roxo) "Discente" X com as "Competências" que "é_relativo_a" (verde) "Competências_Derivada" para toda "Competência_Derivada" que "é_incorporado_por" (azul à direita) "Eixo_de_Formação". A Figura exemplifica melhor as relações entre estas classes.

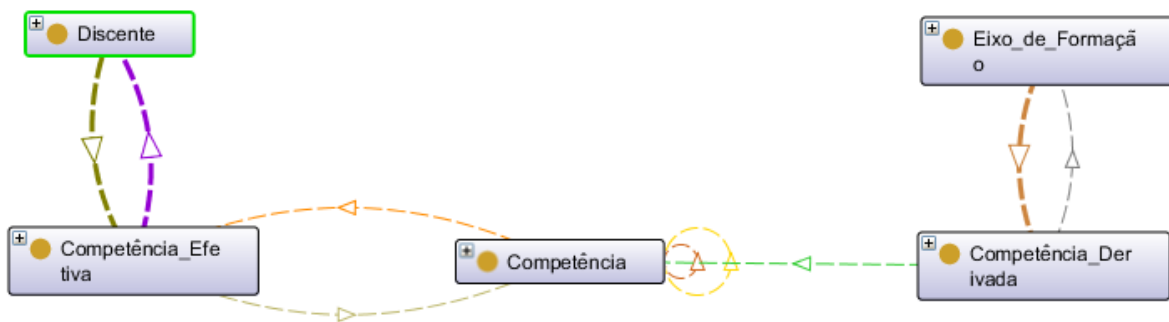


Figura 4.4: Diagrama parcial da Ontologia

Qual a natureza de uma certa competência C?

Em resposta, procura-se o valor da propriedade de dado "tipo_competencia" na "Competência" C. Sugere-se que valor seja Formal, Informal ou Não-Formal.

Qual evidência valida a competência E do aluno?

Conhecendo a "Competência_Efetiva" do "Discente", procura-se o relacionamento "é_validada_por", retornando o indivíduo que causa a validação.

Qual o contexto de uma competência C?

Similar à resposta anterior, a "Competência" possui um relacionamento "é_contextualizado_por", retornando um indivíduo "Contexto" com valores de "nome" e "descrição".

Ao concluir a disciplina de Estruturas de Dados, quais competências serão adquiridas?

para chegar a esta informação, como podemos observar na Figura 4.5, deve-se procurar pela entidade "Disciplina" Estrutura de Dados que se refere ao aluno em questão

(no caso, Alan Turing), e então verificar todas as propriedades de objeto "abrange" para obter os "Conteúdos" relacionados. Sabendo os Conteúdos, verifica-se as "Competências" estão relacionadas por "aborda". Com esta informação, cria-se uma entidade "Competência_Efetiva" para descrever essas novas competências.

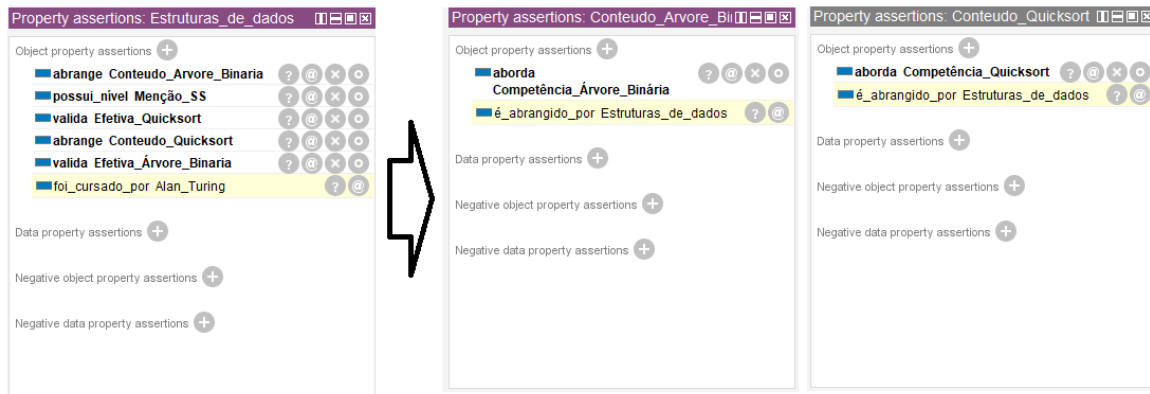


Figura 4.5: diagrama de propriedades de uma Disciplina e seus conteúdos abordados.

Quais são as competências de um curso?

Partindo do "Curso", verifica o "Eixo_de_Formação" dele por "possui_eixo". Com o eixo, seleciona-se todas as propriedades "é_incorporado_por" para obter as "Competências_Derivadas" e assim por "é_relativo_a" obtém-se as competências. Alternativamente, pode-se utilizar as competências derivadas diretamente, sem necessitar do último passo.

Capítulo 5

Conclusão

Visto o impasse em que o graduando em ciência da computação potencialmente encontra ao tentar buscar significado em certos elementos curriculares, foi elaborada uma solução e materializada uma ontologia inserida digitalmente através da metodologia 101[3] e implementada pela ferramenta *Protégé*.

Com isso, percebemos que o trabalho se demonstrou capaz de oferecer uma abordagem exploratória para a elaboração de um modelo de persistência de uma ontologia de competências para discentes de graduação do curso de ciência da computação oferecido pela Universidade de Brasília.

Nota-se que a definição de competência é relativamente ampla e há pequenas diferenças de definição entre autores. Contudo, todas as obras analisadas não apresentaram discrepância tamanha capaz de criar divergências teóricas intransponíveis. Com isso, o modelo apresenta capacidade de exprimir a grande maioria das competências de um discente no devido escopo deste trabalho.

5.1 Contribuições

Visando ampliar o ecossistema SmartUnB.ECOS, houve diversos avanços na sua expressibilidade. Pode-se citar a busca sobre o estado da arte na representação de competências e nos métodos para a realização de um modelo ontológico em Web Semântica a fim de permitir a interoperabilidade entre sistemas de informação do ecossistema.

A ontologia modelada também fornece valor ao seu contexto, pois trabalhos futuros podem vir a utilizá-la, empregando então o Passo 2 do método que se refere ao reuso, facilitando então a evolução da ontologia ou até mesmo empregá-la sem alterações.

A seção de trabalhos relacionados expõe diversas fontes para leitura e aprofundamento do assunto, contendo por exemplo artigos de revisão de literatura[45] que apresenta diversos outros sistemas similares ao implementado em nosso trabalho.

5.2 Limitações

Em virtude da vasta literatura que descreve não só modelos de competências mas também modelos de currículos de graduação, percebe-se que este trabalho não é capaz de garantir a integridade e eficácia perante fontes de competências formais diferente das apresentadas bem como o escopo se ampliar para outros cursos ou até mesmo para o próprio curso em outra instituição.

Não há uma interface direta para sistemas LTI, necessitando a utilização de ferramentas externas para que a interação aconteça.

5.3 Trabalhos Futuros

Há inúmeras formas de evolução deste trabalho. Inicialmente, como apontado como limitação no parágrafo anterior, ampliar a metodologia para documentos internacionais, em exemplo o currículo proposto em 2013 para cursos em computação da ACM e IEEE[46] e ACM (2020). Desta forma, o trabalho estaria globalizado e capaz de se integrar com diversos outros de forma fluída.

Há alguns termos que foram propositalmente deixados vagos a fim de não comprometer a expressabilidade da ontologia (como por exemplo taxonomia e contexto). Contudo, pode-se averiguar se é ou não necessário mantê-los desta forma, trazendo então uma evolução teórica ao modelo.

Também é possível continuar o trabalho ao pesquisar formas de interoperar uma ontologia OWL desenvolvida no *Protégé* para outras linguagens. Foi indicada a API Jena para Java[44], contudo não foi testado sua conveniência para o contexto nem alternativas para outras linguagens.

5.4 Considerações Finais

Consideramos que o trabalho apresentou de forma satisfatória os elementos necessários para o entendimento sobre a elaboração de uma ontologia no contexto da informática aplicada à educação para a representação de competências de discentes em Ciência da Computação.

Também foram expostos o método e a ferramenta de persistência escolhida, bem como suas peculiaridades. A aplicação do método foi documentada e reportada no texto, com o auxílio de imagens para melhor entendimento do leitor. Testes foram conduzidos para verificar se os casos de uso apontados pela metodologia foram contemplados.

Referências

- [1] Sicilia, Miguel Angel: *Ontology-based competency management: infrastructures for the knowledge intensive learning organization*. Em *Intelligent learning infrastructure for knowledge intensive organizations: A semantic Web Perspective*, páginas 302–324. IGI Global, 2005. viii, 11
- [2] Gomes, Alex e Claudia Gomes: *Classificação dos Tipos de Pesquisa em Informática na Educação*. junho 2019. 3, 4
- [3] Noy, Natalya F, Deborah L McGuinness *et al.*: *Ontology development 101: A guide to creating your first ontology*, 2001. 5, 12, 14, 16, 21, 22, 31
- [4] *Moodle statistics*, Mar 2022. <https://stats.moodle.org>, Online, acessado em 07/03/2022. 6
- [5] *Moodle sandbox demo*, Mar 2022. <https://sandbox.moodledemo.net>, Online, acessado em 07/03/2022. 7
- [6] *Foundation - HR Open Standards Consortium, Inc.*, Mar 2022. <https://www.hropenstandards.org/foundation>, [Online; accessed 23. Mar. 2022]. 7
- [7] Allen, Chuck: *Competencies 1.0 (measurable characteristics)*, Oct 2001. http://xml.coverpages.org/HR-XML-Competencies-1_0.pdf, Online, acessado em 23/03/2022. 8, 11
- [8] Mitchell, Luís Henrique Raja Gabaglia: *Gestão de pessoas por competências no ambiente aulanet*. 2004. 8
- [9] Koper, Rob e Marcus Specht: *Ten-competence: Life-long competence development and learning*. Em *Competencies in Organizational e-learning: concepts and tools*, páginas 234–252. IGI Global, 2007. <https://www.academia.edu/download/30781365/11.pdf>, Online, acessado em 24/03/2022. 9
- [10] Kew, Christopher: *The tencompetence personal competence manager.*, 2007. <http://ceur-ws.org/Vol-280/p08.pdf>, Online, acessado em 24/03/2022. 9
- [11] Schoonenboom, Judith, Henk Sligte, Ayman Moghnieh, Krassen Stefanov, Christian Glahn, Marcus Specht, Ruud Lemmers *et al.*: *Supporting life-long competence development using the tencompetence infrastructure: a first experiment*. *International Journal of Emerging Technologies in Learning (iJET)*, 3(2008), 2008. https://www.learntechlib.org/p/45174/article_45174.pdf, Online, Acessado em 24/03/2022. 9

- [12] Rezgui, Kalthoum, Hédia Mhiri e Khaled Ghédira: *Extending moodle functionalities with ontology-based competency management*. *Procedia Computer Science*, 35:570–579, 2014. 11
- [13] Stancin, Kristian, Patrizia Poscic e Danijela Jaksic: *Ontologies in education—state of the art*. *Education and Information Technologies*, 25(6):5301–5320, 2020. 12
- [14] Simon, Augusto, Patricia Alejandra Behar, Cristina Alba Wildt Torrezan, Bruna Kin Slodkowski, Sílvio César Cazella e Daisy Schneider: *Modelagem de uma ontologia de domínio com foco em competências para sistemas de recomendação na educação*. *Revista brasileira de informática na educação*. Florianópolis. Vol. 28 (2020), p. 644–663, 2020. 12, 16, 17
- [15] Miranda, Sergio, Francesco Orciuoli, Vincenzo Loia e Demetrios Sampson: *An ontology-based model for competence management*. *Data & Knowledge Engineering*, 107:51–66, 2017. 12, 13, 24
- [16] Haase, Peter, Holger Lewen, Rudi Studer, Duc Thanh Tran, Michael Erdmann, Mathieu d’Aquin e Enrico Motta: *The neon ontology engineering toolkit*. WWW, 2008. 13
- [17] Steffen, Staab e Studer Rudi: *Handbook on ontologies, international handbooks on information systems*, 2009. 13
- [18] Paquette, Gilbert, Olga Marino e Rim Bejaoui: *A new competency ontology for learning environments personalization*. *Smart Learning Environments*, 8(1):1–23, 2021. 13, 24, 25
- [19] Péres, Alison de Miranda: *Manutenção de portfólio de competências para cursos de graduação em computação por usuário gestor*. 2021. <https://bdm.unb.br/handle/10483/29467>, Online, Acessado em 12/04/2022. 13
- [20] *Learning tools interoperability | ims global learning consortium*. <https://www.imsglobal.org/activity/learning-tools-interoperability>. Online, Acessado em 13/04/2022. 14
- [21] Silva, Claiton Custódio da: *Uma ontologia de perfil holístico para estudantes de graduação*. 2021. <https://bdm.unb.br/handle/10483/29472>, Online, Acessado em 12/04/2022. 14
- [22] Caires Junior, Norberto Luz: *Incorporando a edição de competências a uma rede social descentralizada para utilização em local de trabalho*. 2019. <https://bdm.unb.br/handle/10483/26500>, Online, Acessado em 12/04/2022. 14
- [23] *About – friendica*. <https://friendi.ca/about/>. Online, Acessado em 13/04/2022. 14
- [24] Wilson, Bonita: *Social networking gets serious: Social networking moves beyond time-wasting to become a powerful business tool*. *Human Resource Management International Digest*, 21:6–8, 2013. 14

- [25] Editora Melhoramentos: *Ontologia*. <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/ontologia/>. Online, Acessado em 13/04/2022. 16
- [26] Zorzo, A. F., D. Nunes, E. Matos, I. Steinmacher, J. Leite, R. M. Araujo, R. Correia e S. Martins: *Referenciais de formação para os cursos de graduação em computação*. 2017. <https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1234-referencias-de-formacao-pgcc-2019>, Online, Acessado em 12/04/2022. 16, 19
- [27] Bloom, Benjamin Samuel, Committee of College e University Examiners: *Taxonomy of educational objectives*, volume 2. Longmans, Green New York, 1964. 17
- [28] Anderson, Lorin W., David R. Krathwohl e Peter W. Airasian: *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Abridged Edition*. Addison Wesley Longman, 1ª edição, 2001. 17, 26
- [29] Ferraz, Ana Paula do Carmo Marcheti e Renato Vairo Belhot: *Taxonomia de bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais*. Gestão & produção, 17:421–431, 2010. 17, 26
- [30] Sampson, Demetrios e Demetrios Fytros: *Competence models in technology-enhanced competence-based learning*. Em *Handbook on information technologies for education and training*, páginas 155–177. Springer, 2008. 17
- [31] Gadotti, Moacir: *A questão da educação formal/não-formal*. Sion: Institut International des Droits de 1º Enfant, páginas 1–11, 2005. 17
- [32] Teodorescu, Tina: *Competence versus competency: What is the difference?* Performance Improvement, 45(10):27–30, 2006. <https://onlinelibrary.wiley.com/doi/abs/10.1002/pfi.4930451027>. 18
- [33] Ministério da Educação: *Diretrizes curriculares nacionais para os cursos de graduação em computação*, 2016. http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug=novembro-2016-pdf&Itemid=30192, Online, Acessado em 13/04/2022. 18, 19
- [34] Educação, Ministério da: *Proposta de diretrizes curriculares nacionais para os cursos de graduação em computação*, 2012. http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=11205-pces136-11-pdf&category_slug=julho-2012-pdf&Itemid=30192, Online, Acessado em 13/04/2022. 18
- [35] Institute of Electrical and Electronics Engineers: *IEEE P1484.20.2: IEEE draft recommended practices for defining competencies*, 2020. <https://standards.ieee.org/ieee/1484.20.2/10743/>, Online, Acessado em 13/04/2022. 19
- [36] Institute of Electrical and Electronics Engineers: *IEEE standard for learning technology-data model for reusable competency definitions*, 2008. IEEE Std 1484.20.1-2007. 19

- [37] ACM e IEEE-CS: *Joint task force on computing curricula*, 2013. DOI: <http://dx.doi.org/10.1145/2534860>. 19
- [38] Force, CC2020 Task: *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery, New York, NY, USA, 2020, ISBN 9781450390590. <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>, Online, Acessado em 13/04/2022. 19
- [39] Suárez-Figueroa, Mari Carmen, Asunción Gómez-Pérez e Mariano Fernández-López: *The neon methodology for ontology engineering*. Em *Ontology engineering in a networked world*, páginas 9–34. Springer, 2012. 20
- [40] Fernández-López, Mariano, Asunción Gómez-Pérez e Natalia Juristo: *Methontology: from ontological art towards ontological engineering*. 1997. 20
- [41] Stanford University: *Página de suporte protégé*. <https://protege.stanford.edu/support.php>. Online, Acessado em 13/04/2022. 20
- [42] Musen, Mark A: *The protégé project: a look back and a look forward*. *AI matters*, 1(4):4–12, 2015. 23
- [43] Motik, Boris, Bijan Parsia e Peter F Patel-Schneider: *Owl 2 web ontology language xml serialization*. World Wide Web Consortium, 2009. <https://www.w3.org/TR/owl2-xml-serialization/>, Online, Acessado em 13/04/2022. 23
- [44] *Apache jena - jena ontology api*. <https://jena.apache.org/documentation/ontology/>. Online, Acessado em 03/05/2022. 28, 32
- [45] Stancin, Kristian, Patrizia Poscic e Danijela Jaksic: *Ontologies in education—state of the art*. *Education and Information Technologies*, 25(6):5301–5320, 2020. 31
- [46] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society: *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery, New York, NY, USA, 2013, ISBN 9781450323093. 32

Apêndice A

Código Turtle da Ontologia

```
@prefix : <urn:smartunbecos:competencias#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <urn:smartunbecos:competencias> .

<urn:smartunbecos:competencias> rdf:type owl:Ontology ;
                                owl:versionIRI <urn:smartunbecos:competencias> .

#####
#   Object Properties
#####

### urn:smartunbecos:competencias#aborda
:aborda rdf:type owl:ObjectProperty ;
        owl:inverseOf :é_abordado_por ;
        rdfs:domain :Conteúdo ;
        rdfs:range <urn:smartunbscos:competencias#Competência> .

### urn:smartunbecos:competencias#abrange
:abrange rdf:type owl:ObjectProperty ;
        owl:inverseOf :é_abrangido_por ;
        rdfs:domain :Disciplina ;
```

rdfs:range :Conteúdo .

```
### urn:smartunbecos:competencias#categoriza
:categoriza rdf:type owl:ObjectProperty ;
            owl:inverseOf :é_categorizado_por ;
            rdfs:domain :Taxonomia ;
            rdfs:range :Competência_Efetiva .
```

```
### urn:smartunbecos:competencias#contextualiza
:contextualiza rdf:type owl:ObjectProperty ;
              owl:inverseOf :é_contextualizado_por ;
              rdf:type owl:InverseFunctionalProperty ;
              rdfs:domain :Contexto .
```

```
### urn:smartunbecos:competencias#cursou
:cursou rdf:type owl:ObjectProperty ;
        owl:inverseOf :foi_cursado_por ;
        rdfs:domain :Discente ;
        rdfs:range :Disciplina .
```

```
### urn:smartunbecos:competencias#define_tipo
:define_tipo rdf:type owl:ObjectProperty ;
            owl:inverseOf :possui_tipo ;
            rdfs:range <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#foi_cursado_por
:foi_cursado_por rdf:type owl:ObjectProperty ;
                rdfs:domain :Disciplina ;
                rdfs:range :Discente .
```

```
### urn:smartunbecos:competencias#incorpora
```



```
:incorpora rdf:type owl:ObjectProperty ;
    owl:inverseOf :é_incorporado_por ;
    rdfs:domain :Competência_Derivada ;
    rdfs:range :Eixo_de_Formação .
```

```
### urn:smartunbecos:competencias#nivela
:nivela rdf:type owl:ObjectProperty ;
    owl:inverseOf :possui_nível ;
    rdfs:domain :Escala ;
    rdfs:range :Disciplina ,
        :Evidência .
```

```
### urn:smartunbecos:competencias#pertence_a
:pertence_a rdf:type owl:ObjectProperty ;
    rdfs:domain :Competência_Efetiva ;
    rdfs:range :Discente .
```

```
### urn:smartunbecos:competencias#possui_competência
:possui_competência rdf:type owl:ObjectProperty ;
    rdfs:domain :Discente ;
    rdfs:range :Competência_Efetiva .
```

```
### urn:smartunbecos:competencias#possui_eixo
:possui_eixo rdf:type owl:ObjectProperty ;
    owl:inverseOf :remete_a ;
    rdfs:domain :Curso ;
    rdfs:range :Eixo_de_Formação .
```

```
### urn:smartunbecos:competencias#possui_nível
:possui_nível rdf:type owl:ObjectProperty ;
    rdfs:domain :Disciplina ,
        :Evidência ;
```

rdfs:range :Escala .

```
### urn:smartunbecos:competencias#possui_tipo
:possui_tipo rdf:type owl:ObjectProperty ;
             rdfs:domain <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#remete_a
:remete_a rdf:type owl:ObjectProperty ;
          rdfs:domain :Eixo_de_Formação ;
          rdfs:range :Curso .
```

```
### urn:smartunbecos:competencias#valida
:valida rdf:type owl:ObjectProperty ;
        owl:inverseOf :é_validada_por ;
        rdfs:domain :Evidência ;
        rdfs:range :Competência_Efetiva .
```

```
### urn:smartunbecos:competencias#é_abordado_por
:é_abordado_por rdf:type owl:ObjectProperty ;
                rdfs:domain <urn:smartunbscos:competencias#Competência> ;
                rdfs:range :Conteúdo .
```

```
### urn:smartunbecos:competencias#é_abrangido_por
:é_abrangido_por rdf:type owl:ObjectProperty ;
                 rdfs:domain :Conteúdo ;
                 rdfs:range :Disciplina .
```

```
### urn:smartunbecos:competencias#é_categorizado_por
:é_categorizado_por rdf:type owl:ObjectProperty ;
                    rdfs:domain :Competência_Efetiva ;
                    rdfs:range :Taxonomia .
```

```
### urn:smartunbecos:competencias#é_competência_equivalente_a
:é_competência_equivalente_a rdf:type owl:ObjectProperty ,
                                owl:AsymmetricProperty ;
                                rdfs:domain <urn:smartunbscos:competencias#Competência>
                                rdfs:range <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#é_composta_por
:é_composta_por rdf:type owl:ObjectProperty ;
                 rdfs:domain <urn:smartunbscos:competencias#Competência> ;
                 rdfs:range <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#é_contextualizado_por
:é_contextualizado_por rdf:type owl:ObjectProperty ,
                        owl:FunctionalProperty ;
                        rdfs:domain <urn:smartunbscos:competencias#Competência> ;
                        rdfs:range :Contexto .
```

```
### urn:smartunbecos:competencias#é_incorporado_por
:é_incorporado_por rdf:type owl:ObjectProperty ;
                  rdfs:domain :Eixo_de_Formação ;
                  rdfs:range :Competência_Derivada .
```

```
### urn:smartunbecos:competencias#é_instância_de
:é_instância_de rdf:type owl:ObjectProperty ;
                owl:inverseOf :é_metaclasse_de ;
                rdf:type owl:FunctionalProperty ,
                        owl:AsymmetricProperty ;
                rdfs:domain :Competência_Efetiva ;
                rdfs:range <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#é_metaclassede
:é_metaclassede rdf:type owl:ObjectProperty ,
                owl:AsymmetricProperty ;
                rdfs:domain <urn:smartunbscos:competencias#Competência> ;
                rdfs:range :Competência_Efetiva .
```

```
### urn:smartunbecos:competencias#é_relativo_a
:é_relativo_a rdf:type owl:ObjectProperty ;
                rdfs:domain :Competência_Derivada ;
                rdfs:range <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#é_validada_por
:é_validada_por rdf:type owl:ObjectProperty ;
                 rdfs:domain :Competência_Efetiva ;
                 rdfs:range :Evidência .
```

```
#####
# Data properties
#####
```

```
### urn:smartunbecos:competencias#descrição
:descrição rdf:type owl:DatatypeProperty ;
            rdfs:range xsd:string .
```

```
### urn:smartunbecos:competencias#max
:max rdf:type owl:DatatypeProperty ;
      rdfs:range xsd:positiveInteger .
```

```
### urn:smartunbecos:competencias#min
:min rdf:type owl:DatatypeProperty ;
      rdfs:range xsd:positiveInteger .
```

```
### urn:smartunbecos:competencias#nome
:nome rdf:type owl:DatatypeProperty ;
      rdfs:range xsd:string .
```

```
### urn:smartunbecos:competencias#tipo_competencia
:tipo_competencia rdf:type owl:DatatypeProperty ;
                  rdfs:domain <urn:smartunbscos:competencias#Competência> .
```

```
### urn:smartunbecos:competencias#valor
:valor rdf:type owl:DatatypeProperty ;
       rdfs:range xsd:int .
```

```
#####
```

```
# Classes
```

```
#####
```

```
### urn:smartunbecos:competencias#Competência_Derivada
:Competência_Derivada rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Competência_Efetiva
:Competência_Efetiva rdf:type owl:Class ;
                    owl:disjointWith <urn:smartunbscos:competencias#Competência> ;
```

```
### urn:smartunbecos:competencias#Contexto
:Contexto rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Conteúdo
:Conteúdo rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Curso
:Curso rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Discente
:Discente rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Disciplina
:Disciplina rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Eixo_de_Formação
:Eixo_de_Formação rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Escala
:Escala rdf:type owl:Class .
```

```
### urn:smartunbecos:competencias#Evidência
:Evidência rdf:type owl:Class ;
```

```
### urn:smartunbecos:competencias#Taxonomia
:Taxonomia rdf:type owl:Class .
```

```
### urn:smartunbscos:competencias#Competência
<urn:smartunbscos:competencias#Competência> rdf:type owl:Class ;
```

```
### Generated by the OWL API (version 4.5.9.2019-02-01T07:24:44Z)
```

```
### https://github.com/owlcs/owlapi
```