

MODELAGEM DE UMA ONTOLOGIA PARA CONSTRUÇÃO DE ALGORITMOS

MODELING OF AN ONTHOLOGY FOR THE CONSTRUCTION OF ALGORITHMS

Matheus Gomes Terra¹, Ana Paula Canal², Gustavo Stangherlin Cantarelli³, Henrique Gabriel Goularte Pereira⁴

RESUMO: Com a complexidade das linguagens de programação, estudantes iniciantes na programação de computadores enfrentam dificuldades e, por isso, sistemas que auxiliem na compreensão desses conteúdos são ferramentas úteis. As ontologias fazem a representação dos conceitos, as quais, por meio de consultas, entregam informações relevantes ao usuário. Nesse sentido, este estudo tem como objetivo apresentar uma modelagem de uma ontologia, baseada nas disciplinas de algoritmos, com a ferramenta Protégé, para a construção e o desenvolvimento de algoritmos na linguagem de programação C.

Palavras-chave: Ontologia, Protégé, Algoritmos.

ABSTRACT: With the complexity of programming languages, beginner level students in computer programming are facing difficulties and for this reason, systems that help in the comprehension of these contents are useful tools. Ontologies are the representation of concepts, which through consultations deliver relevant information to the user. In this sense, the present study aims to present a modeling of an ontology based on the subjects about algorithms with the Protégé tool for the construction and development of algorithms in the C programming language.

Keywords: Ontology, Protégé, Algorithms.

1 INTRODUÇÃO

A programação de computadores é uma atividade altamente complexa, que envolve subtarefas ligadas a diferentes domínios de conhecimento (AMBRÓSIO, 2011). Alunos ingressantes em cursos como o da computação apresentam comportamentos distintos nas disciplinas introdutórias relacionadas à programação. Enquanto alguns conseguem aprender a programar rapidamente, outros encontram enormes dificuldades (AMBRÓSIO, 2011).

Os alunos entendem os enunciados dos exercícios apresentados a eles, no entanto encontram dificuldades em definir uma sequência de comandos que leve ao resultado desejado (AMBRÓSIO, 2011). Durante o desenvolvimento de um algoritmo, muitas informações são produzidas e requeridas, e, muitas vezes, é essencial estabelecer conexões entre recursos de informação a fim de se obter o conjunto necessário de informações para apoiar a realização de alguma atividade (SOUZA et al., 2010), e, para desempenhar esse papel, pode-se usar as ontologias.

¹Graduando em Ciência da Computação - Centro Universitário Franciscano
{matheusterra@outlook.com}

²Msc. em Ciência da Computação - Centro Universitário Franciscano
{apc@unifra.br}

³Msc. em Ciência da Computação - Centro Universitário Franciscano
{gus.cant@unifra.br}

⁴Professor dos cursos de Ciência da Computação e Sistemas de Informação
{henriquep@acm.org}

Como um meio de representação, as ontologias são muito utilizadas para representar um contexto específico (GUIZZARDI, 2000). Além disso, elas possibilitam a comunicação entre pessoas acerca de determinado conhecimento, pois permitem raciocínio e entendimento sobre um domínio, cuja relação auxilia na obtenção de consenso, principalmente sobre os termos técnicos (GUIZZARDI, 2000).

São diversas as áreas de aplicação das ontologias, tais como: Recuperação de Informações, Gestão do Conhecimento, Educação, Processamento de Linguagem Natural, Mineração de Dados, entre outras. Entre as áreas de aplicação, a *Web Semântica* tem obtido grandes avanços, tornando-se (BERNERS-LEE, 2001) um padrão de vocabulário comum para a troca de dados, o conhecimento reutilizável e a facilidade na comunicação de sistemas heterogêneos (LIU; ÖZSU, 2009).

Sendo assim, este trabalho tem como objetivo elaborar uma ontologia que será modelada a partir de conteúdos da disciplina de algoritmos com a linguagem de programação C, com a finalidade de mostrar informações por meio de consultas e representar o contexto do desenvolvimento desses algoritmos.

1.1 JUSTIFICATIVA

A ontologia é um método de representação do conhecimento muito relevante no campo da educação (PIERRAKEAS; SOLOMOUR; KAMEAS, 2012). As ontologias são consideradas um meio para facilitar a reutilização e o compartilhamento de conhecimento em aplicativos e grupos de pessoas. De acordo com a literatura, elas estão sendo utilizadas para representar domínios de conhecimentos, estratégias de ensino, perfis de estudantes, sistemas de recomendação, entre outros (PIERRAKEAS; SOLOMOUR; KAMEAS, 2012).

A ontologia pode ser um modelo interessante para a representação do conhecimento em algoritmos, pois percebe-se a falta de domínio da linguagem de programação por parte dos novos alunos (AMBRÓSIO, 2011). Nesse sentido, disponibilizar informações de estruturas computacionais poderá auxiliar na construção do código.

Sendo assim, este estudo visa implementar uma ontologia, a qual poderá ser utilizada como uma base de conhecimento por ferramentas que entreguem elementos e informações a serem utilizadas na elaboração de algoritmos.

2 REFERENCIAL TEÓRICO

2.1 WEB SEMÂNTICA

A *World Wide Web* foi criada por Tim Barner-Lee, entre 1989 e 1991, baseada nos trabalhos sobre hipertexto realizados por Bush e Ted Nelson. A *Web*, conforme Cunha (2002), surgiu com a visão de que ela seria um espaço em que a informação poderia adquirir um significado bem definido, de forma que facilitasse a cooperação e a comunicação entre as pessoas e os agentes computacionais.

A *Web* Sintática, conforme Breitman e Leite (2004), é apenas a apresentação de informações, enquanto o processo de avaliar, classificar e selecionar fica a cargo dos seres humanos. Nesse sentido, os mecanismos de busca são extremamente ricos em quantidade de sites indexados em suas bases de dados, embora sequer os melhores deles consigam abranger a totalidade de conteúdo disponível na *Web* (PICKLER, 2007). Apesar da quantidade de informações recuperadas pelos mecanismos, apenas a parte da *Web* é pesquisada, enquanto uma parte considerável do conteúdo fica inacessível (PICKLER, 2007).

Dessa maneira, a *Web* Semântica visa, justamente, melhorar a satisfação do usuário no momento da busca, retornando-lhe as informações adequadas às suas necessidades. Ao contrário da *Web* sintática, a *Web* Semântica busca capturar o significado das páginas, criando um ambiente em que os computadores possam processar e relacionar conteúdos provenientes de várias fontes. Para isso, é necessário adicionar semânticas aos documentos disponíveis na *Web* (BREITMAN; LEITE, 2004).

Em outras palavras, na *Web* Semântica, as informações publicadas na rede são preparadas para serem compreendidas tanto por humanos quanto por máquinas, o que resultaria em uma *web* mais eficiente e autônoma na busca e na associação de informações (LAMMEL; MIELNICZUK, 2012).

Para a *Web* Semântica tornar-se possível, os computadores necessitam ter acesso a coleções estruturadas de informações e de conjuntos de regras de inferência que ajudem no processo de dedução automática e na representação de conhecimento (PICKLER, 2007). Tais regras são especificadas por meio de ontologias que permitem representar a semântica dos dados (PICKLER, 2007).

2.2 ONTOLOGIA

A ontologia teve sua origem na Grécia e significa o estudo da existência de entidades tanto abstratas como concretas que formam o mundo (TURCATEL, 2014). Atualmente, na Inteligência Artificial (IA), utiliza-se a ontologia para representar um contexto com grande número de informações (NOY; MCGUINNESS, 2000).

Uma ontologia para a Inteligência Artificial é uma descrição explícita e formal das classes (ou conceitos) em um domínio. Ela também descreve as características, os atributos (ou funções ou propriedades) e as relações entre os membros da classe (NOY; MCGUINNESS, 2000). Uma ontologia, juntamente com um conjunto das instâncias das classes, constitui uma base de conhecimento (NOY; MCGUINNESS, 2000).

Conforme Studer, Benjamins e Fensel (1998) cita, pode-se classificar a ontologia em quatro tipos: ontologias de domínio, ontologias genéricas, ontologias de aplicação e ontologias de representação.

As ontologias de domínio são utilizadas para representar um domínio específico, descrevendo situações reais sobre ele. Já as ontologias genéricas são similares às de domínio, porém suas descrições são mais genéricas, já que estas são independentes de um domínio particular. Por sua vez, as ontologias de aplicação são modeladas para conceitos de uma determinada aplicação. Por último, as ontologias de representação não pertencem a domínio algum, em que apenas se fornecem inferências que serão usadas por outras ontologias, como a de domínio, por exemplo (STUDER; BENJAMINS; FENSEL, 1998).

2.3 ONTOLOGY WEB LANGUAGE (OWL)

Para modelar a ontologia, é necessário descrevê-la, a fim de que o computador identifique a linguagem e compreenda as relações. Com isso, a ferramenta *Protégé* utiliza, em seu núcleo, uma linguagem que descreve ontologias (NOY; MCGUINNESS, 2000).

A *Ontology Web Language* (OWL) é uma linguagem para descrever e criar instâncias de ontologias, ou seja, um objeto cujo comportamento e estado são definidos pela ontologia (PANDEY; DWIVEDI; VERMA, 2013). Ela pode incluir descrições de classes, propriedades, entre outras características. Na OWL, está especificada a semântica de maneira formal e como deve-se proceder com os fatos que não estão presentes na ontologia. Atualmente, existem três tipos de sublinguagens da OWL; são elas: *OWL Lite*, *OWL Description Logics* e *OWL Full* (HE; AN, 2011).

A *OWL Lite* é dedicada a usuários que necessitam, de maneira simples, uma classificação e características das relações. Ela só permite valores de cardinalidade de 0 ou 1 e, por ser mais simples, fornece caminhos rápidos entre os dados na ontologia (HE; AN, 2011).

Já a *OWL Description Logics* (OWL DL) foi projetada para usuários que buscam a máxima expressividade, sem perder a completude computacional, em que todas as vinculações são garantidas para serem computadas. Além disso, ela oferece decidibilidade, oferecendo métodos eficientes em problemas de decisão e sendo utilizada para apoiar segmentos em que existam sistemas de raciocínio. Esses sistemas são capazes de simular a habilidade humana de adaptação a experiências anteriores (HE; AN, 2011).

Por fim, a *OWL Full* é destinada a usuários que procuram por máxima expressividade e liberdade sintática, sem garantias computacionais do *Resource Description Framework* (RDF), uma linguagem de representação informal na internet. Ela permite que uma ontologia aumente o significado do vocabulário pré-definido (RDF ou OWL) (HE; AN, 2011).

2.4 SQWRL

Para verificar as informações na linguagem OWL, utilizam-se consultas. Existem diversos tipos de consultas, como a linguagem *XPath (XML Path Language)*, *SPARQL (SPARQL Protocol and RDF Query Language)*, *SWRL (Semantic Web Rule Language)*, entre outras.

A linguagem *Semantic Query-Enhanced Web Rule Language (SQWRL)* (O'CONNOR; DAS, 2008) permite a criação de consultas a ontologias de forma análoga ao *Structured Query Language (SQL)*. Diferentemente de outras consultas em ontologias, as consultas via SQWRL apenas retornam informações da ontologia que satisfazem determinadas restrições e que podem ser manipuladas, posteriormente, utilizando uma linguagem de programação qualquer, como a Java (GASSEN et al., 2008).

O SQWRL utiliza uma biblioteca com os métodos do *Semantic Web Rule Language (SWRL)*, podendo-se utilizar essas regras para fazer buscas em OWL (GASSEN et al., 2008). Por esse motivo, o SQWRL torna-se uma poderosa linguagem para buscas em OWL, podendo aproveitar ao máximo a expressividade semântica representada em OWL (GASSEN et al., 2008), sendo a mais indicada para consultas sobre ontologias expressas nessa linguagem (GASSEN et al., 2008).

A sintaxe dos comandos SQWRL é representada da seguinte forma:

$$\text{Classe} (?c) \wedge \text{propriedade} (?c, ?p) \rightarrow \text{sqwrl:select} (?c, ?p)$$

O exemplo acima mostra que os dados antecedidos por “?” são passados ao motor de busca, e o comando “^” significa que será utilizado outro campo de busca. Ao procurar dados, por exemplo, no elemento “propriedade”, utilizam-se dois valores. Esses valores são definidos como “*domain*” e “*range*” (domínio e alcance), ou seja, a partir de uma informação, acha-se a outra. Ao declarar qual será a busca, utiliza-se o comando “->” e o comando de busca “*sqwrl:select*”, no qual deverão ser selecionados os valores que se desejam retornar da ontologia (GASSEN et al., 2008).

2.5 ALGORITMOS

Um algoritmo pode ser definido como uma sequência finita de passos (instruções) para resolver um determinado problema (KOLIVER et al., 2009). São exemplos de algoritmos instruções de montagem, receitas, manuais de uso etc. Um algoritmo não é a solução do problema, pois, se assim fosse, cada problema teria um único algoritmo; um algoritmo é um caminho para a solução de um problema. Em geral, existem muitos (senão infinitos) caminhos que levam a uma solução satisfatória (TONET; KOLIVER, 2000).

Na computação, um algoritmo é utilizado para fazer com que o computador execute tarefas e, para isso, é necessário que ele execute um programa. Este é um conjunto de instruções que indicam ao

computador, passo a passo, o que ele precisa fazer. Logo, um programa é um algoritmo computacional descrito em uma linguagem de programação (TONET; KOLIVER, 2000).

Existem diversas linguagens de programação, como, por exemplo, a linguagem C, a qual, inicialmente, era utilizada para programação de sistemas. Em virtude de sua portabilidade e eficiência, a linguagem C ganhou grande popularidade (SCHILDT, 1997).

2.6 ONTOLOGY DEVELOPMENT 101

A metodologia utilizada para a construção do modelo ontológico está dividida em sete passos, conforme a *Ontology Development 101*, e foi realizada com o *software Protégé*, o qual possui, no núcleo do *software*, características que reúnem conceitos da OWL DL e da OWL Lite (NOY; MCGUINNESS, 2000). O objetivo dessa metodologia é a criação de novas ontologias. Algumas características da criação das novas ontologias são: compartilhar entendimento comum da estrutura da informação entre pessoas e agentes de *software*; permitir a reutilização do conhecimento de domínio; e analisar o conhecimento do domínio (NOY; MCGUINNESS, 2000).

Os passos da metodologia são:

1. Determinação do domínio e do escopo da ontologia;
2. Reutilização das ontologias;
3. Enumeração dos termos importantes na ontologia;
4. Definição de classes e hierarquia de classes;
5. Definição das propriedades das classes;
6. Definição das características/restrições das propriedades;
7. Criação das instâncias;

3 TRABALHOS RELACIONADOS

Foram identificados vários trabalhos relacionados, dos quais se citam somente três, considerados relevantes por utilizarem a ontologia de forma similar à utilizada neste estudo.

O estudo realizado por Lee, Ye e Wang (2005) aborda a modelagem de uma ontologia voltada à aprendizagem na linguagem Java. No trabalho do autor, utilizam-se informações baseadas em uma matriz curricular para ambientes de autoaprendizagem para cursos introdutórios de programação. O objetivo do autor com a ontologia é proporcionar a autoaprendizagem.

Já o trabalho de Pierrakeas, Solomour e Kameas (2012) desenvolve uma ontologia que se baseia no aprendizado de linguagens de programação. Utilizando a metodologia *Ontology Development 101*, o autor descreve duas linguagens como exemplos (linguagem C e Java). Com o objetivo de ensinar a

programação a possíveis alunos e facilitar a representação das linguagens pelos tutores, o autor descreve as funcionalidades e as características de cada uma das linguagens por meio de descrições e informações.

Por sua vez, o trabalho de Abuhassan e Almashaykhi (2012) realiza a modelagem de uma ontologia para descrever linguagens de programação orientadas a objeto. O objetivo do autor é auxiliar pesquisadores no desenvolvimento e no estudo das linguagens já existentes, além de centralizar informações sobre essas linguagens. Para isso, ele descreve as linguagens de programação com características específicas de cada uma delas, categorizando-as com o objetivo de cada uma.

Este estudo difere dos trabalhos dos autores, pois, além de representar a linguagem de programação C, também visa representar exercícios dados a alunos. Ele tem como objetivo representar a utilização da linguagem frente a exercícios, visto que, conforme Pierrakeas, Solomour e Kameas (2012), a linguagem C é uma das primeiras a ser apresentadas para novos estudantes do curso de computação. Por meio de consultas, é possível verificar, em exercícios propostos, a estrutura e a descrição de elementos da linguagem C para auxiliar o aluno na compreensão do conteúdo.

4 METODOLOGIA

A modelagem da ontologia foi realizada com a metodologia *Ontology Development 101*, utilizando a ferramenta *Protégé* na versão 5.0.0. Essa ferramenta utiliza, em seu núcleo, a linguagem OWL e permite, de forma estruturada, manipular as entidades da ontologia e realizar consultas na mesma, usando o *Protégé*. Além disso, na metodologia, os autores recomendam e utilizam essa ferramenta.

O conteúdo da ontologia é abordado a partir das disciplinas de Algoritmos I e II, no curso de graduação da área da Informática do Centro Universitário Franciscano. A ontologia baseia-se na linguagem C, pois, nessas disciplinas, os algoritmos são trabalhados e implementados nessa linguagem.

Para as consultas na ontologia, foi utilizada a linguagem SQWRL. Ela foi baseada nas relações dos indivíduos que representam as questões utilizadas na disciplina, consultando as descrições, os valores e a estrutura de cada indivíduo, a fim de mostrar cada elemento.

5 MODELAGEM DA ONTOLOGIA

No primeiro passo da metodologia *Ontology Development 101*, deve-se responder algumas perguntas para definir o domínio da ontologia. As perguntas e respostas podem ser visualizadas na tabela 1.

Tabela 1 – Perguntas e respostas elaboradas para a modelagem da ontologia

Perguntas	Respostas
Que domínio a ontologia irá cobrir?	As disciplinas de Algoritmos I e II do Centro Universitário Franciscano.
Por que usar ontologia?	Para a identificação de padrões na construção de algoritmos e de exercícios propostos, cujo objetivo é mostrar as estruturas envolvidas e as soluções dos exercícios.
Para que tipos de perguntas a informação na ontologia deve fornecer respostas?	Para dúvidas na construção e desenvolvimento dos códigos.
Quem vai usar e manter a ontologia?	Professores da disciplina selecionada e o autor deste projeto.

Após responder as questões acima, a metodologia recomenda criar algumas questões para complementar o escopo da ontologia. Essas questões foram respondidas e ajudaram a formar o domínio da ontologia. Algumas questões desenvolvidas são descritas abaixo:

- *Int* armazena números ou caracteres?
- Qual a melhor declaração para questões com palavras?
- Que tipo de variável é ideal para questões de cálculo?
- Quais as características de uma variável do tipo *float*?
- Em quais tipos de problemas deve-se usar uma variável do tipo *int*?
- Em quais tipos de questões de cálculo deve-se usar o *for*?
- Quais as características do *if*?
- Para questões com duas possibilidades, qual estrutura deve-se usar?
- Quais as características do *printf*?
- Para questões de cálculo, qual estrutura usar?
- Para contar letras ou números, o que usar?

O segundo passo da metodologia é a reutilização de ontologias. Existem diversas bibliotecas de ontologias reutilizáveis na *Web* e algumas comerciais. Para este estudo, não será reutilizada ontologia alguma, pois a proposta deste trabalho envolve o desenvolvimento de uma ontologia desde o seu início.

No terceiro passo, foram selecionados termos (Tabela 2) considerados importantes sobre os questionamentos realizados no passo um, e outros foram retirados de bibliografias, os quais são adicionados para ajudar a completar o domínio que a ontologia representará (SCHILDT, 1997; KOLIVER et al., 2009).

Tabela 2 – Exemplo de termos selecionados para serem utilizados como superclasses, classes e subclasses

Termos		
Abrindo Arquivos	Funções Data e Hora	Operadores Aritméticos
Abrir e Fechar	Funções de Alocação Dinâmica	Operadores de Ponteiros
Algoritmos	Funções de E/S	Operadores Relacionais e Lógicos
Alocação Dinâmica	Funções Gráficas	Outros Tipos de Dados
Argumentos de Funções	Funções Matemáticas	Parâmetros Formais
Argumentos para <i>main</i>	Funções Miscelâneas	Parênteses e Colchetes
Arquivos	Funções <i>String</i> e Caracteres	Ponteiro para Matriz
Arquivos Separados	Funções tipos <i>Void</i>	Ponteiros
Atribuições Múltiplas	Globais	Ponteiros de Arquivos
Bibliotecas	Ímpar ou Par	Ponteiros e Matrizes
Bibliotecas e Arquivos	Imprimir Palavras	<i>Print</i>
Blocos de Comandos	Incremento e Decremento	Protótipos

Em seguida, no passo quatro, é realizada a elaboração da hierarquia das classes. Para isso, primeiramente, é necessário definir as superclasses, as classes e as subclasses da ontologia. Essa definição será baseada na sua generalidade. As superclasses escolhidas foram “Algoritmos” e “Exercícios”, pois são mais genéricas e, sendo assim, representam uma gama maior de informações.

Já as classes escolhidas foram selecionadas por não serem tão genéricas quanto às superclasses e menos específicas que as subclasses (Tabela 3).

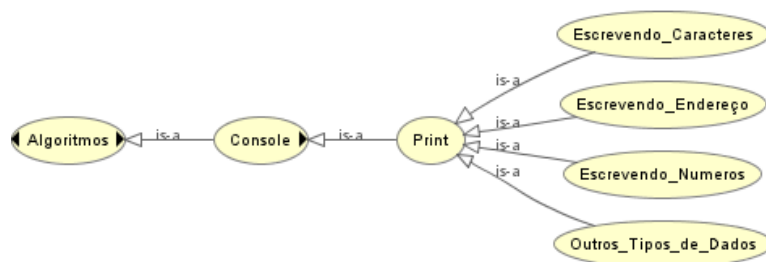
Tabela 3 – Termos selecionados para serem utilizados como classes

Classes		
Arquivos	Matrizes e <i>String</i>	Questões de Estrutura Repetição
Comandos	Ponteiros	Questões de Vetores e Matrizes
Console	Questões Aritméticas	Questões Sintáticas
Expressões	Questões de Arquivos	Questões de Estrutura <i>If Else</i>
Funções		

Os demais termos do passo três foram utilizados para a escolha das subclasses. Esses termos são mais específicos e relacionados às classes.

Como etapa final do passo quatro, foi elaborada a hierarquia das classes (Figura 1). Para tal, utilizou-se o modelo *Top-down*, pois ele apresenta uma visualização partindo dos termos mais genéricos para os mais específicos.

Figura 1 – Hierarquia *Top-Down* dos termos escolhidos como classes e subclasses



Conforme mostra a figura 1, as subclasses “Escrevendo_Endereço” e “Escrevendo_Caracteres”, por exemplo, são classes mais específicas das classes “Print”, “Console” e “Algoritmos”. Essas subclasses são tipos de “*print*” que a linguagem C possui. Nessa sequência, verifica-se que a classe “Algoritmos” é a mais genérica, e as subclasses “Escrevendo_Endereço” e “Escrevendo_Caracteres” são mais específicas.

No quinto passo, foram definidas as propriedades das classes. Nessa etapa, definem-se as propriedades intrínsecas e extrínsecas da ontologia. As propriedades intrínsecas foram definidas para realizar o relacionamento entre as classes e as instâncias. Na tabela 4, estão demonstradas algumas das propriedades intrínsecas da ontologia. É possível visualizar que a propriedade “Tem Algoritmo”, por exemplo, relaciona a classe “Problemas” com a classe “Algoritmos”. Essa relação é definida pela expressão: “Problemas Tem Algoritmo Algoritmos”. Na ontologia, há, também, propriedades intrínsecas que relacionam indivíduos, cuja relação define que um indivíduo possui outro indivíduo como uma propriedade.

Tabela 4 – Exemplo de propriedades intrínsecas definidas como relacionamento da ontologia

Classe	Relacionamento	Alcance
Exercícios	Tem Algoritmo	Algoritmos
Algoritmos	Manipula Arquivos	Arquivos
Arquivos	Funções de Arquivos	Subclasses de Arquivos
Algoritmos	Manipula Matrizes e <i>Strings</i>	Matrizes e <i>String</i>
Matrizes e <i>String</i>	Propriedades Matrizes e <i>String</i>	Subclasses de Matrizes e <i>String</i>
Algoritmos	Manipula Ponteiros	Ponteiros
Ponteiros	Propriedades de Ponteiros	Subclasses de Ponteiros
Ponteiros e Matrizes	Características das Matrizes de Ponteiros	Matrizes de Ponteiros
Algoritmos	Tem Comandos	Comandos

Ainda no quinto passo, tem-se a criação de propriedades extrínsecas (Tabela 5). Estas definem os conteúdos que estão fora do contexto da classe, como, por exemplo, descrição, valor, entre outros. Essas

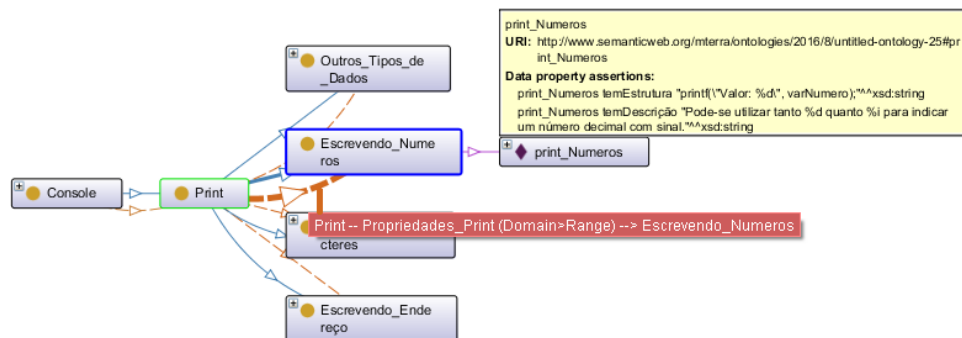
propriedades são atribuídas às instâncias da ontologia. Tais propriedades definem atributos complementares.

Tabela 5 – Propriedades extrínsecas dos indivíduos da ontologia

Propriedade	Objetivo
Tem Descrição	Atribui a um indivíduo uma descrição.
Tem Estrutura	Atribui a um indivíduo a forma estruturada a qual é usado.
Tem Valor	Atribui a um indivíduo um valor.
Tem Narrativa	Atribui aos indivíduos de questionamentos a narrativa da questão.

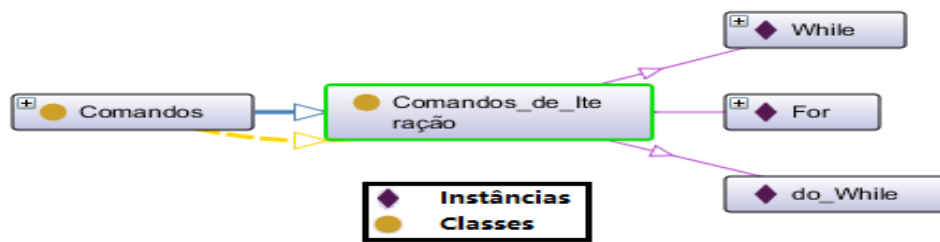
No sexto passo, devem-se atribuir as propriedades criadas no passo cinco aos elementos da ontologia (Figura 2). Atribuem-se as propriedades intrínsecas às superclasses com as classes e subclasses ou de indivíduo para indivíduo. Essas propriedades não possuem cardinalidade definida, e seu alcance é de direta relação, ou seja, relaciona a superclasse com a classe, a classe com a subclasse, e assim sucessivamente. As propriedades extrínsecas são atribuídas apenas aos indivíduos e possuem valores do tipo “String”, pois são descritas determinadas informações a esses indivíduos.

Figura 2 – Propriedades atribuídas à classe “Escrevendo_Numeros” e à instância “print_Numeros”



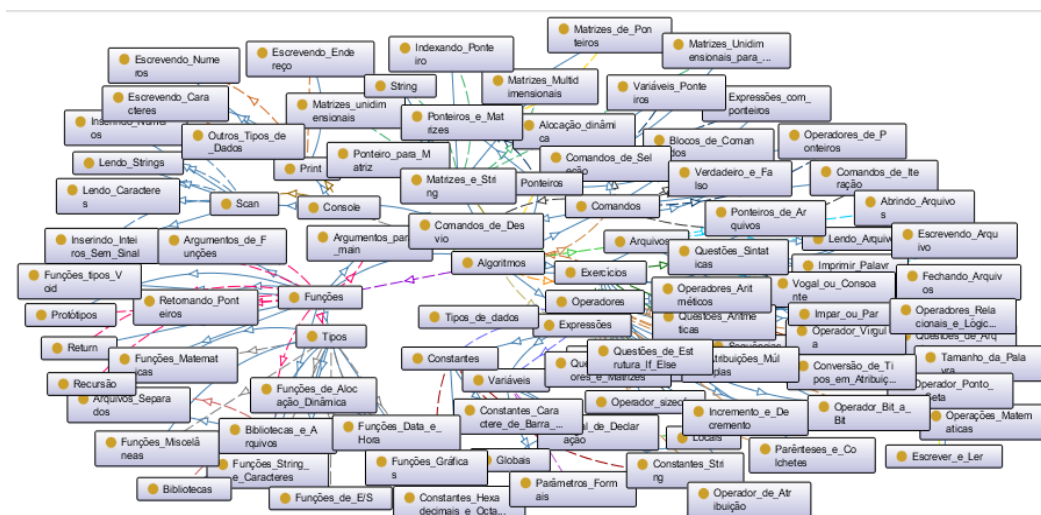
A última etapa da modelagem é a criação das instâncias. As instâncias são parte de uma única classe, as quais definirão um indivíduo da classe. Na Figura 3, é possível visualizar um exemplo de três instâncias atribuídas à classe “Comandos_de_Iteração” da qual esses indivíduos fazem parte. Assim, um indivíduo “For”, por exemplo, é um comando de iteração.

Figura 3 – Instâncias atribuídas à classe “Comandos_de_Iteração”



Todas as instâncias elaboradas para a modelagem da ontologia possuem pelo menos uma característica estabelecida pelas propriedades extrínsecas, contendo informações que serão coletadas pela consulta. Após a definição das superclasses, classes, subclasses, instâncias, características intrínsecas e extrínsecas e seus relacionamentos, tem-se a modelagem da ontologia. Na figura 4, tem-se a visualização de todas as classes relacionadas na ontologia.

Figura 4 – Visão geral da ontologia



6 CONSULTAS

Para verificar as informações atribuídas na modelagem da ontologia e responder as questões criadas no passo um, utilizam-se consultas que buscam, na ontologia, informações que respondem os questionamentos. Essas consultas foram elaboradas utilizando a linguagem SQWRL. Nelas, buscam-se, nas classes, os indivíduos de interesse, mostrando suas propriedades (descrição, valor etc.). Nesse sentido, tem-se como exemplo a consulta abaixo, que responde o questionamento: “qual é a característica do comando “if”?”:

Figura 5 – Consulta do Comando if

Comandos(if ^ temDescrição(if, ?descrição) ^ temEstrutura(if, ?estrutura) -> sqwrl:select(?descrição, ?estrutura)

Essa consulta busca o indivíduo “if” na classe a que ele pertence. Buscam-se, também, as suas propriedades extrínsecas, por meio da propriedade “temDescrição” e “temEstrutura”. Para isso, as variáveis “descrição” e “estrutura” devolvem a característica do indivíduo selecionado. Como resultado, tem-se o conteúdo do indivíduo:

Figura 6 – Retorno da consulta pela variável “descrição”

?descrição
A estrutura if é utilizada para verificar se uma determinada expressão é satisfatória. Caso o seja, ela executa um determinado comando.

Figura 7 – Retorno da consulta pela variável “estrutura”

?estrutura
A forma geral da sentença if é:
if(expressão) comando;
else comando;
onde comando pode ser um único comando, um bloco de comandos ou nada (no caso de comandos vazios).
A cláusula else é opcional.

Cada consulta possui um tipo de retorno especificado um a um. Na ontologia, também se descreveram exercícios. Cada exercício descrito na ontologia procurou mostrar as estruturas utilizadas, bem como suas descrições, a fim de mostrar o domínio de cada exercício em relação à linguagem de programação. Neste estudo, foram criados diversos tipos de exercícios, desde questões aritméticas até questões de arquivos. Para responder, por exemplo, o questionamento “qual estrutura utilizar em questões de cálculo?”, tem-se a consulta abaixo:

Figura 8 – Consulta para verificar qual estrutura utilizar em questões de cálculo

Questões_Aritméticas(Soma_de_Numeros_Pares) ^ temElementos(Soma_de_Numeros_Pares, ?elemento) ^ temDescrição(?elemento, ?descrição) -> sqwrl:select(?elemento, ?descrição)

Essa consulta busca o exercício “Soma_de_Numeros_Pares”. Para isso, busca-se, na propriedade intrínseca “temElementos”, todos os indivíduos ligados ao questionamento. Essa propriedade é consultada e retornada pela variável “elemento”. Para descrever cada um dos indivíduos, é consultada a propriedade extrínseca “temDescrição” de cada um dos elementos retornados por meio da variável “elemento”. Esta propriedade retorna à descrição de cada um dos elementos pela variável “descrição”. Ao final, tem-se a listagem de todos os elementos e suas descrições, à qual pertence a resposta do exercício, representada

pelo indivíduo “Soma_de_Numeros_Pares”. Para a ontologia, criou-se trinta consultas para validar as informações, descrições e valores da modelagem.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este estudo apresentou a modelagem de uma ontologia que representa os conteúdos das disciplinas de algoritmos com a linguagem C frente a exercícios.

Diante das diversas possibilidades existentes para o auxílio no desenvolvimento de algoritmos, optou-se pela modelagem de uma ontologia justamente por ela permitir o compartilhamento de informação e sua reutilização. Além disso, a ontologia serve de base de conhecimento de diversos sistemas, os quais buscam, na ontologia, relações entre as informações e, a partir disso, trabalham melhor com a informação contida no ambiente.

A respeito da metodologia utilizada no desenvolvimento da ontologia, optou-se pela *Ontology Development 101* por ela apresentar passos detalhados para a modelagem e uma ferramenta de desenvolvimento.

Com a ontologia desenvolvida neste estudo, permite-se aos estudantes buscar informações com maior facilidade frente aos desafios que lhes são apresentados no desenvolvimento dos algoritmos, pois as informações estão centralizadas e compartilhadas. Com isso, o desenvolvimento pode tornar-se mais dinâmico, permitindo que dúvidas como o que utilizar ou como utilizar determinados elementos da linguagem sejam definidas pela ontologia.

Durante o trabalho, encontraram-se algumas dificuldades, como problemas de desempenho e limitações técnicas com a utilização da ferramenta *Protégé* e com informações a respeito da estrutura das consultas na ontologia. Além disso, este estudo possui algumas limitações, como a necessidade de outro *software* para a utilização da ontologia em um sistema que auxilie no desenvolvimento dos algoritmos.

Como trabalho futuro, pretende-se utilizar a ontologia como uma base de informação para o desenvolvimento de um sistema de recomendação com o *framework* Jena. Esses sistemas buscam informações a partir de um determinado conhecimento para recomendá-los a usuários. Também se pretende aplicar essa ontologia em uma disciplina de algoritmos, a fim de verificar sua utilização junto aos estudantes.

REFERÊNCIAS

ABUHASSAN, I. A. O.; ALMASHAYKHI, A. M. O. Domain Ontology for Programming Languages. **Journal of Computations & Modelling**, Amman Arab University, v. 2, n. 4, p. 75-91, 2012.

AMBRÓSIO, A. P. L. et al. Programação de computadores: compreender as dificuldades de aprendizagem dos alunos. **Revista Galego-Portuguesa de Psicologia e Educación**, Universidade Federal de Goiás, v. 19, n. 1, p. 185-197, 2011.

BERNERS-LEE, J. H. T.; LASSILA, O. The semantic web. **Scientific American**, Nova Iorque, v. 284, n. 5, mai. 2001. Disponível em: <<https://www.scientificamerican.com/article/the-semantic-web/#>>. Acesso em: 01 de setembro de 2016.

BREITMAN, K.; LEITE, J. C. S. P. Ontologias: como e por que criá-las. Disponível em: <<http://www-di.inf.puc-rio.br/~julio/Slct-pub/JAI.pdf>>, mai. 2004.

CUNHA, L. M. S. Web Semântica: estudo preliminar. Embrapa Informática Agropecuária. <<https://www.embrapa.br/busca-de-publicacoes/-/publicacao/8670/web-semantica-estudo-preliminar>>, set. 2002.

GASSEN, J. B. et al. Uma comparação entre linguagens para consultas sobre ontologias OWL. Seminário de Informática – RS (SEMINFO RS 2008). Universidade Luterana do Brasil, Torres, 2008.

GUIZZARDI, G. **Desenvolvimento para e com reuso**: um estudo de caso no domínio de vídeo sob demanda. 2000. 202 f. Dissertação (Mestrado em 2000) – Centro Tecnológico da Universidade Federal do Espírito Santo, Espírito Santo, 2000.

HE, G.; AN, L. Ontology Language OWL Research Study. BaoDing University, BaoDing, China, 2011.

KOLIVER, C. et al. **Introdução à construção de algoritmos**: notas de aula. Caxias do Sul: Editora Educus, 2009.

LAMMEL, I.; MIELNICZUK, L. Aplicação da *Web Semântica* no Jornalismo. **Estudos em Jornalismo e Mídia**, Florianópolis, v. 9, n. 1, 2012. Disponível em: <<https://periodicos.ufsc.br/index.php/jornalismo/article/viewFile/1984-6924.2012v9n1p180/22315>>. Acesso em: 10 de setembro de 2016.

LEE, M.; YE, D. Y.; WANG, T. I. (2005). Java Learning Object Ontology. Laboratory of Intelligent Network Applications, National Chung Kung University. Disponível em: <<http://ieeexplore.ieee.org/document/1508750/?reload=true>>. Acesso em: 1 de outubro de 2016.

LIU, L.; ÖZSU, M. T. **Encyclopedia of Database Systems**. Springer, Estados Unidos: Editora Springer, 2009. (Vol. 1).

NOY, N. F.; MCGUINNESS, D. L. Ontology Development 101: a guide to creating your first ontology. Stanford University, 2000. Disponível em: <http://protege.stanford.edu/publications/ontology_development/ontology101.pdf>. Acesso em: 5 de maio de 2016.

PANDEY, R.; DWIVEDI, D. S.; VERMA, P. UnivPeopleProgram Ontology: a OWL based Structural definition for Semantic Web. Uttar Pradesh, 2013. Disponível em: <<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6558198>>. Acesso em: 10 de outubro de 2016.

PICKLER, M. E. V. *Web Semântica*: ontologias como ferramentas de representação do conhecimento. Universidade Estadual de Londrina, 2007. Disponível em: <<http://portaldeperiodicos.eci.ufmg.br/index.php/pci/article/view/251>>. Acesso em: 10 de setembro de 2016.

PIERRAKEAS, C.; SOLOMOUR, G.; KAMEAS, A. An Ontology-based Approach in Learning Programming Languages. **16th Panhellenic Conference on Informatics**, Piraeus Grécia, v. 1, n. 1, p. 393-398, 2012. Disponível em: <<http://ieeexplore.ieee.org/document/6377424/?reload=true>>. Acesso em: 7 de outubro de 2016.

O'CONNOR, M. J.; DAS, A. K. SQWRL: a query language for OWL. **Stanford Center for Biomedical Informatics Research**, Stanford, Estados Unidos, v. 259, n. 1, p. 208-215, 2008. Disponível em: <http://ceur-ws.org/Vol-529/owled2009_submission_42.pdf>. Acesso em: 13 de setembro de 2016.

SCHILDT, H. **C Completo e Total**. 3. ed. Local: Editora E.R.J. Informática Ltda, 1997.

SOUZA, F. P. et al. Estudo de Ontologias em Engenharia de *Software*, 2010. Disponível em: <http://ic.ufabc.edu.br/II_SIC_UFABC/resumos/paper_5_291.pdf>. Acesso em: 10 de setembro de 2016.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge Engineering: principles and methods, 1998. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0169023X97000566>>. Acesso em: 1 de maio de 2016.

TONET, B.; KOLIVER, C. Núcleo de Apoio Aprendizagem de Programação. Universidade de Caxias do Sul, 2000. Disponível em: <<https://www.ucs.br/portais/cent/nucleos/568/>>. Acesso em: 3 de outubro de 2016.

TURCATEL, I. O. **Ontologias para a descoberta de recursos na ciência**: análise de VIVO-ISF. 2014. 84 f. Trabalho de Conclusão de Curso (Especialização) –Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/112165>>. Acesso em: 15 de setembro de 2016.