

Capítulo 5

PROVA DE CONCEITO

Este capítulo apresenta a Prova de Conceito do sistema de gerenciamento de perfis. Ela foi desenvolvida para realizar as *queries* e as inserções de dados pertinentes a quatro casos de uso descritos neste capítulo, com o objetivo de validar a ideia proposta e a manipulação da ontologia criada para representar o conhecimento. Além disso, é apresentado o ambiente computacional usado para a realização da PoC e um breve tutorial dos componentes utilizados.

5.1 Ambiente computacional da PoC

O ambiente computacional para a implementação da PoC em um computador comum, que fez o papel de cliente e servidor, foi composto essencialmente por dois elementos:

- A interface HTTP Apache Jena Fuseki em sua versão 3.13.1 para a manipulação dos dados (<http://jena.apache.org/documentation/fuseki2/>). O Fuseki é um servlet, isto é, um componente que estende as funcionalidades de um servidor web. Ele é um subprojeto da API Jena que possibilita armazenar um grafo composto por triplas RDF e oferece suporte para que sejam realizadas *queries* e atualizações, como inserções e remoções (JENA, 2014). No caso desta implementação, as triplas foram exportadas da ontologia **CourseStructure** no formato *Turtle* (do inglês Terse RDF Triple Language), que é uma sintaxe para expressar dados no modelo RDF (BECKETT et al., 2014). O Fuseki foi uma alternativa à API Jena usada na Subseção 4.5.2.1, pois ela não possui suporte para a atualização do grafo quando este está armazenado localmente.
- O servidor Apache Tomcat em sua versão 9.0.29, que é um servidor web desenvolvido em Java e funciona como um contêiner de servlets (BRITTAIN; DARWIN, 2007). O Tomcat foi utilizado como contêiner para viabilizar a execução local do Fuseki e sua documentação pode ser encontrada em (<https://tomcat.apache.org/>).

Após a devida instalação dos dois componentes citados, o Fuseki pode ser acessado localmente em um *browser* comum por meio do endereço `<http://localhost:3030/>`. A página inicial do Fuseki é ilustrada pela Figura 5.1.

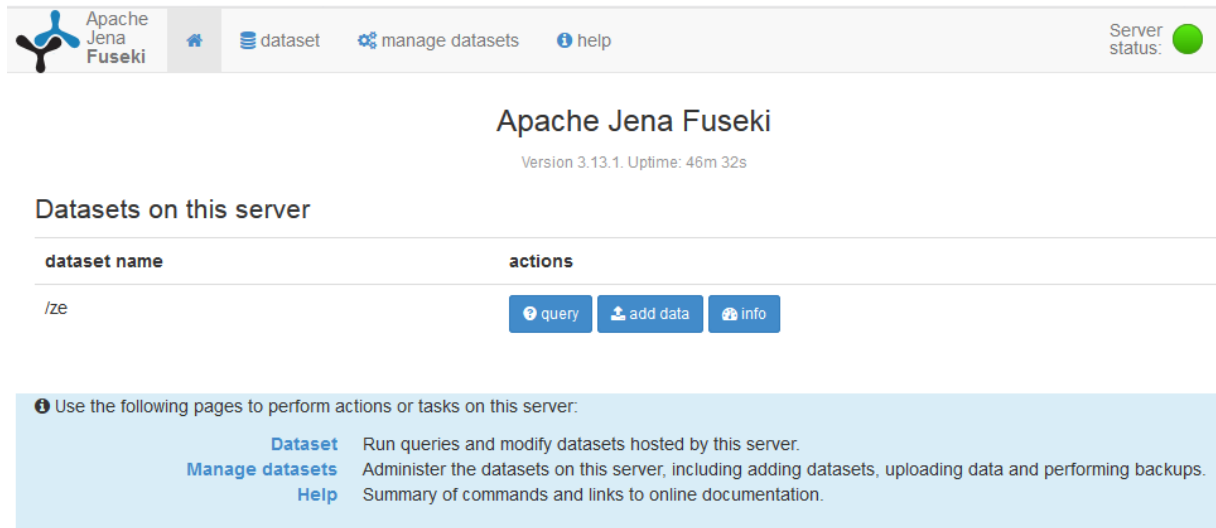


Figura 5.1: Página inicial do servidor local Fuseki.

Ao acessar o servidor, é necessário criar um novo banco de dados local na aba *manage datasets* e depois na aba *add new dataset*. Então, basta escolher um nome e selecionar uma das opções de persistência que mantêm os dados mesmo após a reinicialização, como mostrado na Figura 5.2.

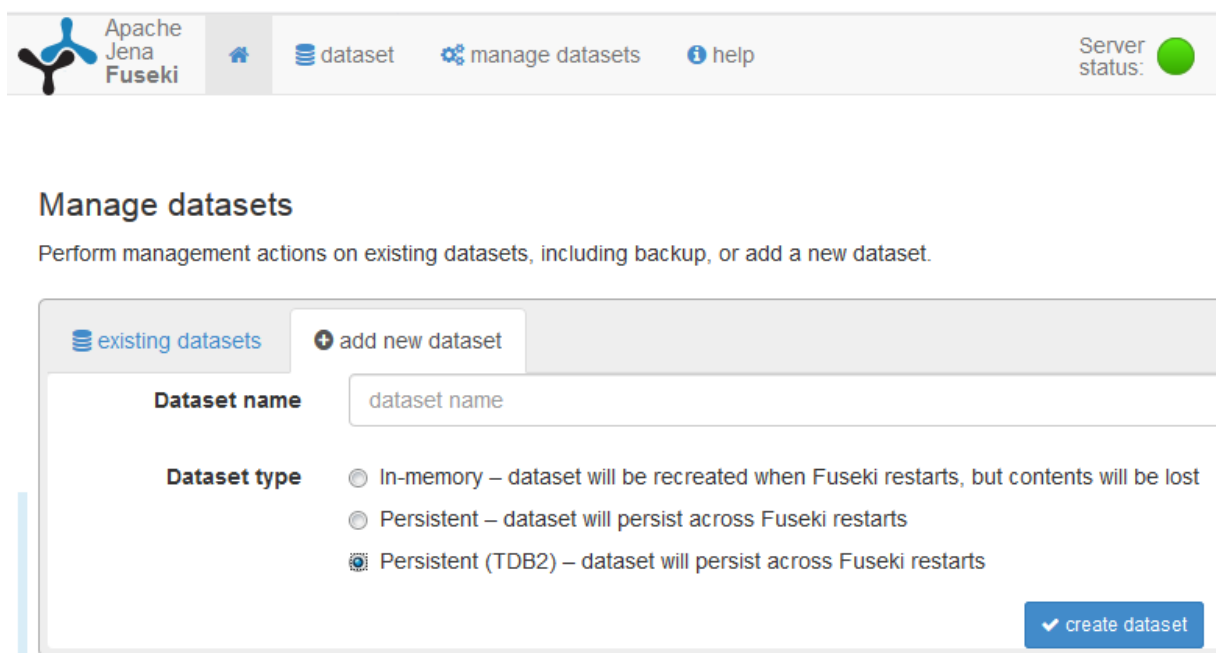


Figura 5.2: Página de inserção de banco de dados local.

Feito isso, é preciso fazer o *upload* do arquivo no formato Turtle correspondente à ontologia

escolhida para manipulação na aba *dataset* e depois na aba *upload files*. Ao clicar na aba, deve ser feita a escolha de um arquivo clicando no botão *select files* e, depois de selecionado, o arquivo pode ser colocado no servidor por meio dos botões *upload all* ou *upload now*. A Figura 5.3 ilustra a escolha do arquivo *CourseStructure_Turtle.ttl*, que é a ontologia utilizada.

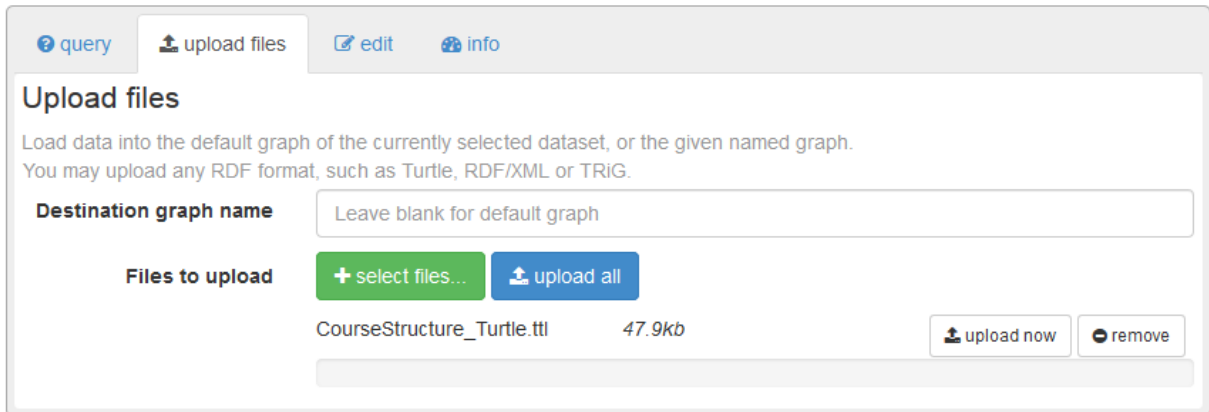


Figura 5.3: Página de *upload* de arquivo para o servidor.

Quando um arquivo que contém uma ontologia em um formato reconhecido pelo sistema é inserido, é possível visualizar e editar o código referente ao grafo por ele gerado na aba *edit*, como mostrado na Figura 5.4.

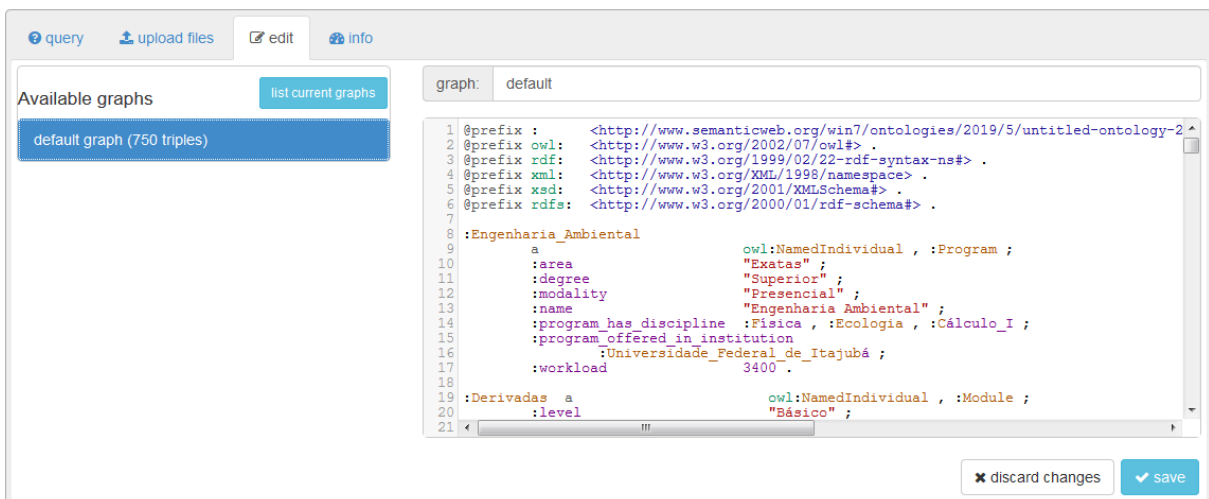


Figura 5.4: Página de visualização e edição do código referente ao grafo inserido.

Com o grafo criado, é possível executar as consultas usando a linguagem SparQL na aba *query*. Nessa página, também é possível editar o tipo de conteúdo nos campos correspondentes e determinar o tipo de comando SparQL que será executado, isto é, *query* ou *update*. Isso é definido no campo *SparQL endpoint*. A Figura 5.5 mostra a página com uma consulta genérica, com o campo *SparQL endpoint* definido para executar uma *query*. A Figura 5.6 destaca o mesmo campo definido para realizar uma atualização, ou *update*, no grafo.

The screenshot shows the 'query' tab of a web interface. At the top, there are links for 'upload files', 'edit', and 'info'. Below this is the 'SPARQL query' section with a prompt: 'To try out some SPARQL queries against the selected dataset, enter your query here.' There are two buttons for 'EXAMPLE QUERIES': 'Selection of triples' and 'Selection of classes'. Below these are 'PREFIXES' buttons for 'rdf', 'rdfs', 'owl', 'xsd', and a gear icon. The 'SPARQL ENDPOINT' field contains '/ze/query'. To its right are two dropdown menus: 'CONTENT TYPE (SELECT)' set to 'JSON' and 'CONTENT TYPE (GRAPH)' set to 'Turtle'. The main area is a code editor with a SPARQL query:


```

1
2
3 SELECT ?subject ?predicate ?object
4 WHERE {
5   ?subject ?predicate ?object
6 }
7 LIMIT 25
  
```

 On the right side of the code editor are icons for undo, redo, and a play button.

Figura 5.5: Página de inserção do código da *query* ou *update*.

This is a close-up of the 'SPARQL ENDPOINT' field. The text '/ze/update' is entered into the input box, which is highlighted with a blue border and a blue selection bar.

Figura 5.6: Campo que determina a finalidade do código definido como *update*.

Por fim, a Figura 5.7 apresenta a execução de uma *query* no grafo da ontologia. Após a inserção do código, a *query* é executada por meio do botão *play* mostrado no canto superior direito da figura. O resultado é exibido em forma de tabela ou de código, conforme selecionado nos botões *Table* e *Raw Response*. Neste caso, os dados resultantes da *query* são mostrados em uma tabela.

The screenshot shows the 'query' tab with a different query:


```

1 PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/untitled-ontology-2>
2 SELECT ?Objeto
3 WHERE
4 {
5   :Francisco :student_knows_learningObject ?Objeto .
6 }
  
```

 The 'QUERY RESULTS' section is visible below. It has two tabs: 'Table' (selected) and 'Raw Response'. There is a download icon. It shows 'Showing 1 to 2 of 2 entries'. A search bar and 'Show 50 entries' are also present. The results are displayed in a table:

	Objeto
1	:Introdução_à_Lógica_Video1
2	:Limites_Lista1

 At the bottom, it says 'Showing 1 to 2 of 2 entries'.

Figura 5.7: Página de inserção do código da *query* ou *update*.

5.2 Implementação da PoC

Para a criação da PoC, foi feita uma adaptação na ontologia **CourseStructure** descrita na Seção 4.5.4 para que ela abrangesse, além da estrutura de um curso e do conhecimento nele contido, os alunos cujos perfis precisam ser gerenciados. Assim, foi criada uma nova classe, **Student**, que representa os alunos e se relaciona com as seis classes originais. A Figura 5.8 apresenta um diagrama com a estrutura da ontologia adaptada.

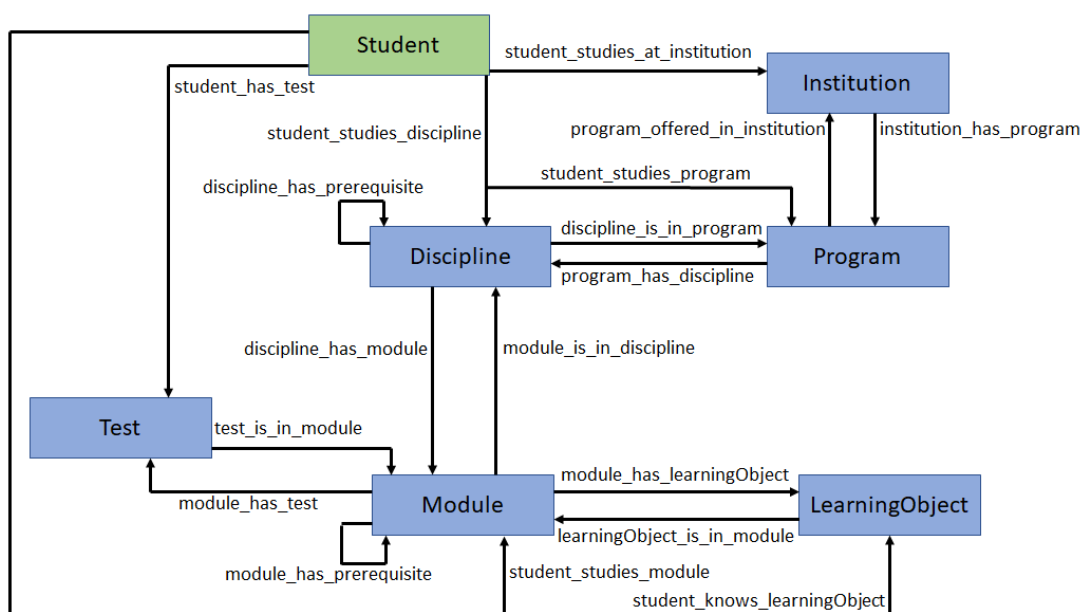


Figura 5.8: Diagrama de representação da ontologia **CourseStructure** adaptada com a classe **Student**.

Além disso, foram criados indivíduos com o objetivo de simular um ambiente real de uso da ontologia. Os indivíduos criados estão representados por meio de um grafo na Figura 5.9, desde a instituição de ensino até os objetos de aprendizagem. As arestas que ligam as instituições de ensino aos programas indicam que estes são oferecidos por aquelas, enquanto as arestas que conectam programas a disciplinas indicam que estas fazem parte da grade curricular daqueles. As demais arestas do grafo, entre disciplinas, módulos e objetos de aprendizagem, indicam uma relação de pertença, isto é, os objetos de aprendizagem pertencem aos módulos e os módulos pertencem às disciplinas. Além dos indivíduos listados na figura, também foi criado um indivíduo da classe **Test** para cada módulo. O endereço para o código OWL da ontologia já com os indivíduos pode ser visualizado no Apêndice C.

A Figura 5.10 apresenta a propriedade de objeto que expressa a relação de prerequisite entre as disciplinas, ao passo que a Figura 5.11 expressa a mesma relação entre os módulos.

Com relação aos indivíduos da classe **Student**, eles foram criados de acordo com a neces-

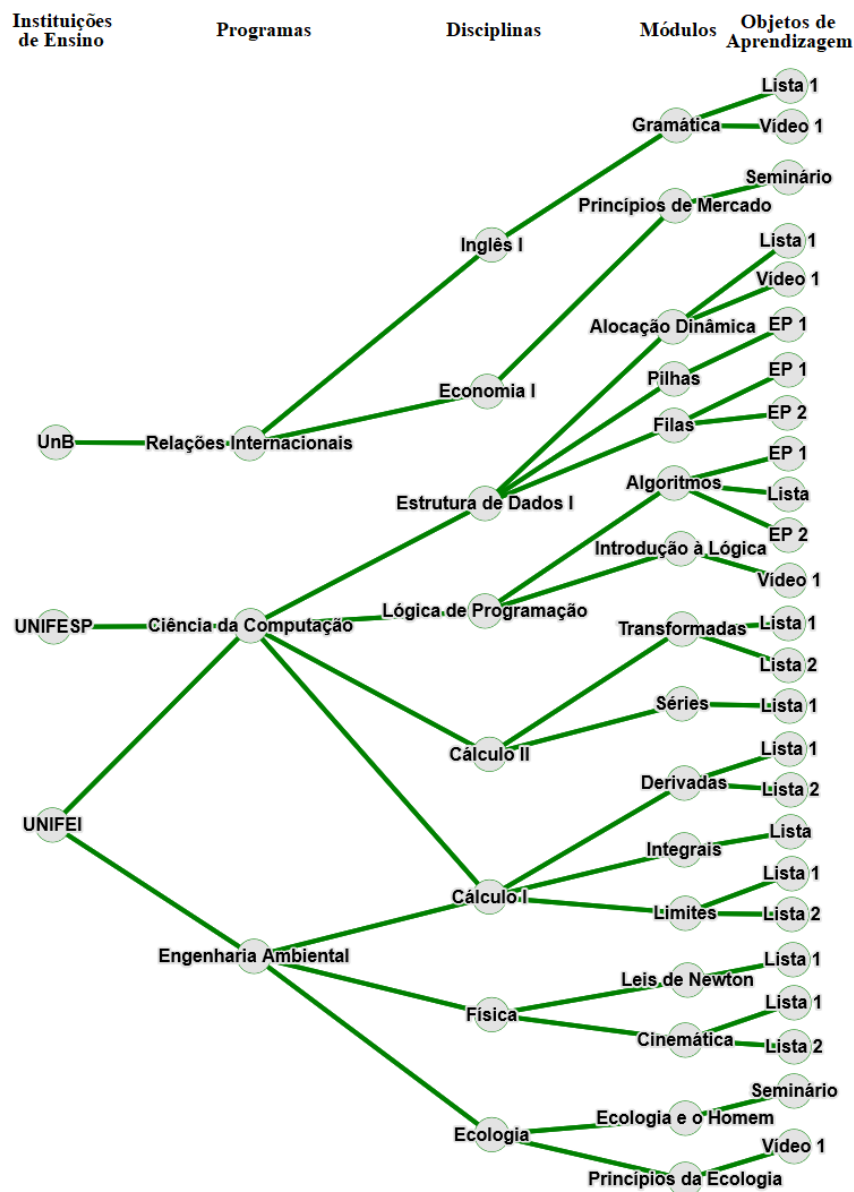


Figura 5.9: Grafo de representação dos indivíduos criados a partir da ontologia CourseStructure.

sidade dos casos de uso que serão apresentados nas seções a seguir. Um exemplo que pode ser analisado para o entendimento das estruturas de *query* e do retorno obtido com elas é o do aluno fictício Francisco.

Francisco é um aluno do programa de Ciência da Computação da Universidade Federal de São Paulo. Os dados dele podem ser visualizados por meio da *query* a seguir:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Predicado ?Objeto
WHERE{
    :Francisco ?Predicado ?Objeto .
}
```

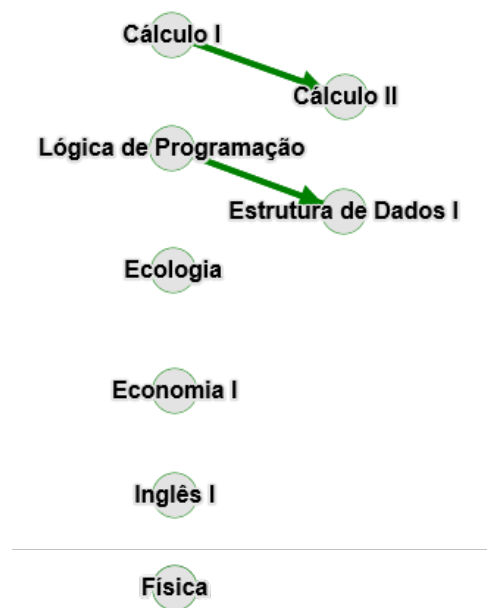


Figura 5.10: Grafo de representação das disciplinas e seus prerrequisitos. As setas representam o predicado “é prerrequisito de”.

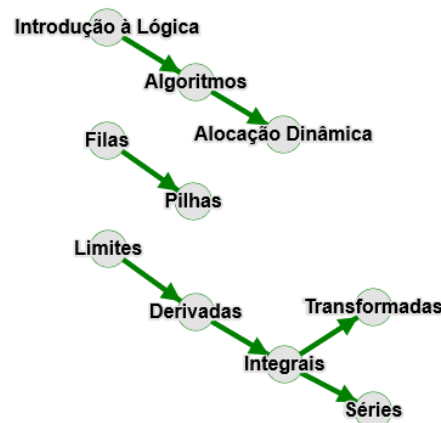


Figura 5.11: Grafo de representação dos módulos e seus prerrequisitos. As setas representam o predicado “é prerrequisito de”.

A Figura 5.12 apresenta as informações contidas nas triplas retornadas a partir da *query* acima, isto é, os dados atuais do aluno no sistema. Nesse caso, Francisco é o sujeito, a primeira coluna representa os predicados e a segunda os objetos. Segundo os dados, o sujeito é um aluno que pertence à classe **Student** (linha 8), se chama Francisco (linha 3), seu número de identificação é 111 (linha 11), ele está matriculado nas disciplinas de Lógica de Programação e Cálculo I (linhas 1 e 2), cursa Ciência da Computação na Universidade Federal de São Paulo (linhas 4 e 12), está estudando os módulos sobre Limites e Introdução à Lógica (linhas 6 e 5) e domina os objetos de aprendizagem Lista 1, sobre Limites, e Vídeo 1, sobre Introdução à Lógica (linhas 10 e 9).

	Predicado	Objeto
1	:student_studies_discipline	:Lógica_de_Programação
2	:student_studies_discipline	:Cálculo_I
3	:name	"Francisco"
4	:student_studies_program	:Ciência_da_Computação
5	:student_studies_module	:Introdução_à_Lógica
6	:student_studies_module	:Limites
7	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
8	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	:Student
9	:student_knows_learningObject	:Introdução_à_Lógica_Vídeo1
10	:student_knows_learningObject	:Limites_Lista1
11	:student_id	"111"^^xsd:integer
12	:student_studies_at_institution	:Universidade_Federal_de_São_Paulo

Showing 1 to 12 of 12 entries

Figura 5.12: Informações contidas nas triplas que representam os dados do Francisco.

Os mesmos dados apresentados na forma de tabela pela Figura 5.12 podem ser vistos na forma de triplas RDF, como explicado na Seção 4.2, da seguinte maneira:

```
@prefix :      <http://www.semanticweb.org/win7/ontologies/2019/5/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

:Francisco a owl:NamedIndividual , :Student ;
    :name "Francisco" ;
    :student_id 111 ;
    :student_knows_learningObject :Introducao_a_Logica_Video1 ,
    :Limites_Lista1 ;
    :student_studies_at_institution :Universidade_Federal_de_Sao_Paulo ;
    :student_studies_discipline :Logica_de_Programacao ,
    :Calculo_I ;
    :student_studies_module :Introducao_a_Logica ,
    :Limites ;
    :student_studies_program :Ciencia_da_Computacao .
```

5.3 Caso de uso 1: atualização do sistema

O primeiro caso de uso abordado envolve a atualização do sistema quando um aluno completa uma disciplina, um módulo ou um objeto de aprendizagem, de modo que em seu perfil conste a nova competência adquirida. Para ilustrar essa situação, foram utilizados os dados do mesmo aluno tomado como exemplo na Seção 5, o Francisco, e a atualização é referente aos

módulos e objetos de aprendizagem da disciplina de *Cálculo I*.

O módulo sobre *Limites* contém duas listas de exercícios como objetos de aprendizagem: *Lista 1* e *Lista 2*. Inicialmente, Francisco tinha concluído apenas a *Lista 1*. No decorrer do curso, Francisco concluiu a *Lista 2* e conseguiu realizar a avaliação do módulo, estando apto a iniciar o módulo sobre *Derivadas*, que tem *Limites* como prerequisite. Para fazer essa atualização, foi utilizada a seguinte *query*:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>

INSERT DATA {
    :Francisco :student_knows_learningObject :Limites_Lista2.
    :Francisco :student_has_test :Limites_Teste.
    :Francisco :student_studies_module :Derivadas.
} ;
```

O comando **INSERT DATA** faz a inserção de três triplas no grafo: a primeira indicando que Francisco já concluiu a *Lista 2*; a segunda corresponde à avaliação do módulo sobre *Limites*; e a terceira indica que ele iniciou o módulo sobre *Derivadas*. Com essa atualização, ao executar a *query* para obter os dados de Francisco, são obtidos os dados apresentados na Figura 5.13, que exhibe as atualizações nas linhas 5, 6 e 13.

Predicado	Objeto
1 :student_studies_discipline	:Lógica_de_Programação
2 :student_studies_discipline	:Cálculo_I
3 :name	"Francisco"
4 :student_studies_program	:Ciência_da_Computação
5 :student_has_test	:Limites_Teste
6 :student_studies_module	:Derivadas
7 :student_studies_module	:Introdução_à_Lógica
8 :student_studies_module	:Limites
9 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
10 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	:Student
11 :student_knows_learningObject	:Introdução_à_Lógica_Vídeo1
12 :student_knows_learningObject	:Limites_Lista1
13 :student_knows_learningObject	:Limites_Lista2
14 :student_id	"111"^^xsd:integer
15 :student_studies_at_institution	:Universidade_Federal_de_São_Paulo

Showing 1 to 15 of 15 entries

Figura 5.13: Informações contidas nas triplas que representam os dados do Francisco atualizados.

5.4 Caso de uso 2: verificação dos conhecimentos adquiridos

O segundo caso de uso consiste em uma consulta ao sistema para verificar quais são as disciplinas estudadas e os objetos de aprendizagem e módulos dominados por um aluno.

Para ilustrar essa situação, foram utilizados os dados de um aluno fictício chamado Raul. Para visualizar os dados que dizem respeito a ele, foi utilizada a *query* abaixo:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Predicado ?Objeto
WHERE{
    :Raul ?Predicado ?Objeto .
}
```

Predicado	Objeto
1 :student_studies_discipline	:Cálculo_II
2 :student_studies_discipline	:Cálculo_I
3 :name	"Raul"
4 :student_studies_program	:Ciência_da_Computação
5 :student_has_test	:Séries_Testes
6 :student_has_test	:Derivadas_Testes
7 :student_has_test	:Limites_Testes
8 :student_has_test	:Integrais_Testes
9 :student_studies_module	:Derivadas
10 :student_studies_module	:Limites
11 :student_studies_module	:Séries
12 :student_studies_module	:Integrais
13 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	<http://www.w3.org/2002/07/owl#NamedIndividual>
14 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>	:Student
15 :student_knows_learningObject	:Séries_Lista1
16 :student_knows_learningObject	:Integrais_Lista
17 :student_knows_learningObject	:Limites_Lista1
18 :student_knows_learningObject	:Limites_Lista2
19 :student_knows_learningObject	:Derivadas_Lista2
20 :student_knows_learningObject	:Derivadas_Lista1
21 :student_id	"050"^^xsd:integer
22 :student_studies_at_institution	:Universidade_Federal_de_São_Paulo

Showing 1 to 22 of 22 entries

Figura 5.14: Informações contidas nas triplas que representam os dados do Raul.

A Figura 5.14 apresenta as triplas retornadas a partir da *query* acima, isto é, os dados de Raul no sistema. Contudo, há mais informações do que as necessárias para o contexto deste caso de uso, que busca apenas as disciplinas estudadas e os objetos de aprendizagem e módulos que o aluno já domina. Assim, foi elaborada uma *query* para retornar apenas os dados interessantes

para o contexto:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Disciplina ?Modulo ?Objeto_de_Aprendizagem
WHERE{
    {
        :Raul :student_studies_discipline ?Disciplina .
    }UNION{
        :Raul :student_studies_module ?Modulo .
    }UNION{
        :Raul :student_knows_learningObject ?Objeto_de_Aprendizagem .
    }
}
```

	Disciplina	Modulo	Objeto_de_Aprendizagem
1	:Cálculo_II		
2	:Cálculo_I		
3		:Derivadas	
4		:Limites	
5		:Séries	
6		:Integrais	
7			:Séries_Lista1
8			:Integrais_Lista
9			:Limites_Lista1
10			:Limites_Lista2
11			:Derivadas_Lista2
12			:Derivadas_Lista1

Showing 1 to 12 of 12 entries

Figura 5.15: Informações contidas nas triplas que representam as disciplinas, os módulos e os objetos de aprendizagem que constam no perfil do Raul.

Desta vez, foram retornadas apenas as informações pertinentes ao caso de uso, como ilustrado na Figura 5.15. O comando **UNION** utilizado na pesquisa possibilita a visualização concomitante das três informações desejadas, representadas pelas variáveis “Disciplina”, “Modulo” e “Objeto_de_Aprendizagem” listadas no comando **SELECT**, pois ele faz a concatenação de consultas diferentes que retornam triplas diferentes. Neste caso específico, o sujeito, Raul, é o mesmo em todas as *subqueries*, o que muda são o predicado e o objeto para fazer a correspondência com as informações desejadas. As três consultas, caso feitas individualmente, retornariam as disciplinas, os módulos e os objetos de aprendizagem exibidos nas Figuras 5.16, 5.17 e 5.18, respectivamente.

Disciplina	
1	:Cálculo_II
2	:Cálculo_I
Showing 1 to 2 of 2 entries	

Figura 5.16: Informações contidas nas triplas que representam as disciplinas que constam no perfil do Raul.

Modulo	
1	:Derivadas
2	:Limites
3	:Séries
4	:Integrais
Showing 1 to 4 of 4 entries	

Figura 5.17: Informações contidas nas triplas que representam os módulos que constam no perfil do Raul.

Objeto_de_Aprendizagem	
1	:Séries_Lista1
2	:Integrais_Lista
3	:Limites_Lista1
4	:Limites_Lista2
5	:Derivadas_Lista2
6	:Derivadas_Lista1
Showing 1 to 6 of 6 entries	

Figura 5.18: Informações contidas nas triplas que representam os objetos de aprendizagem que constam no perfil do Raul.

Uma outra opção de *query* é relacionar as disciplinas, os módulos e os objetos de aprendizagem que constam no perfil de um aluno. Usando novamente o Raul como exemplo, a *query* seria:

```

PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Disciplina ?Modulo ?Objeto_de_Aprendizagem
WHERE{
    :Raul :student_knows_learningObject ?Objeto_de_Aprendizagem .
    ?Objeto_de_Aprendizagem :learningObject_is_in_module ?Modulo .
    ?Modulo :module_is_in_discipline ?Disciplina .
}

```

A Figura 5.19 exibe o resultado que relaciona as disciplinas, os módulos e os objetos de aprendizagem que constam no perfil do Raul.

	Disciplina	Modulo	Objeto_de_Aprendizagem
1	:Cálculo_I	:Integrais	:Integrais_Lista
2	:Cálculo_I	:Limites	:Limites_Lista1
3	:Cálculo_I	:Limites	:Limites_Lista2
4	:Cálculo_I	:Derivadas	:Derivadas_Lista2
5	:Cálculo_I	:Derivadas	:Derivadas_Lista1
6	:Cálculo_II	:Séries	:Séries_Lista1

Showing 1 to 6 of 6 entries

Figura 5.19: Informações contidas nas triplas que representam a relação entre disciplinas, módulos e objetos de aprendizagem que constam no perfil do Raul.

Além disso, é possível fazer essa relação para uma disciplina ou módulo específicos. Tomando como exemplo a disciplina de *Cálculo I* e ainda considerando o Raul, a *query* é:

```

PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Modulo ?Objeto_de_Aprendizagem
WHERE{
    :Raul :student_knows_learningObject ?Objeto_de_Aprendizagem .
    ?Objeto_de_Aprendizagem :learningObject_is_in_module ?Modulo .
    ?Modulo :module_is_in_discipline :Calculo_I .
}

```

A Figura 5.20 exibe o resultado que relaciona os módulos e os objetos de aprendizagem que constam no perfil do Raul para a disciplina de *Cálculo I*.

	Modulo	Objeto_de_Aprendizagem
1	:Derivadas	:Derivadas_Lista2
2	:Derivadas	:Derivadas_Lista1
3	:Integrais	:Integrais_Lista
4	:Limites	:Limites_Lista1
5	:Limites	:Limites_Lista2

Showing 1 to 5 of 5 entries

Figura 5.20: Informações contidas nas triplas que representam a relação entre os módulos e os objetos de aprendizagem referentes à disciplina de *Cálculo I* que constam no perfil do Raul.

5.5 Caso de uso 3: verificação dos conhecimentos que não foram cobertos

O terceiro caso de uso tem objetivo semelhante ao segundo, porém em sentido oposto, isto é, o aluno faz uma consulta ao sistema para listar o que ele ainda não aprendeu, considerando um determinado curso ou disciplina.

Para ilustrar esse cenário, foram utilizados novamente os dados do aluno tomado como exemplo na Seção 5.2, o Francisco. A verificação foi feita considerando o programa de Ciência da Computação, no qual Francisco está matriculado. A fim de melhor visualizar o contexto do caso de uso, a *query* abaixo pode ser usada para a obtenção das disciplinas contidas no programa:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Disciplinas_Comp
WHERE{
    :Ciencia_da_Computacao :program_has_discipline ?Disciplinas_Comp .
}
```

A Figura 5.21 apresenta todas as disciplinas contidas no programa de Ciência da Computação fictício usado para ilustrar o caso. Agora, pode-se fazer o mesmo para a obtenção das disciplinas que constam no perfil do Francisco:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Disciplinas_Francisco
WHERE{
    :Francisco :student_studies_discipline ?Disciplinas_Francisco .
}
```

A Figura 5.22 apresenta todas as disciplinas cursadas por Francisco. Assim, é possível fazer uma *query* que retorne todas as disciplinas do programa de Ciência de Computação que Francisco ainda não cursou, isto é, as disciplinas sobre as quais ele ainda não tem conhecimento:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Disciplinas_Faltantes
WHERE{
    {
        :Ciencia_da_Computacao :program_has_discipline ?Disciplinas_Faltantes .
    }MINUS{
        :Francisco :student_studies_discipline ?Disciplinas_Faltantes .
    }
}
```

Disciplinas_Comp	
1	:Banco_de_Dados
2	:Compiladores
3	:Cálculo_I
4	:Cálculo_II
5	:Cálculo_Numérico
6	:Engenharia_de_Software
7	:Estrutura_de_Dados_I
8	:Estrutura_de_Dados_II
9	:Grafos
10	:Lógica_de_Programação
11	:Probabilidade_e_Estatística
12	:Programação_Orientada_a_Objeto
13	:Sistemas_Distribuídos
14	:Sistemas_Embarcados
15	:Sistemas_Móveis
16	:Sistemas_Operacionais
Showing 1 to 16 of 16 entries	

Figura 5.21: Informações contidas nas triplas que representam as disciplinas do programa de Ciência da Computação.

A *query* descreve dois padrões de triplas no comando **WHERE** em duas linhas, uma que retorna as disciplinas do programa de Ciência da Computação e outra que retorna as disciplinas cursadas por Francisco, e usa o comando **MINUS** para listar, como retorno para a *query*, os resultados obtidos a partir da primeira linha excluindo os resultados da segunda (como em uma operação matemática de subtração). A Figura 5.23 apresenta todas as disciplinas do Programa de Ciência da Computação que ainda não foram estudadas por Francisco e, portanto, não constam no perfil dele. A mesma lógica pode ser utilizada para fazer, por exemplo, o levantamento dos módulos de uma disciplina que ainda faltam ser estudados, ou listar os objetos de aprendizagem de um módulo que ainda não foram dominados.

Disciplinas_Francisco	
1	:Lógica_de_Programação
2	:Cálculo_I
Showing 1 to 2 of 2 entries	

Figura 5.22: Informações contidas nas triplas que representam as disciplinas cursadas por Francisco.

5.6 Caso de uso 4: levantamento do que é mais estudado

O quarto caso de uso consiste em uma consulta para levantar um mapa geral de disciplinas, módulos ou objetos de aprendizagem, indicando quais são os mais estudados e dominados.

Para ilustrar esse caso, foi considerada uma turma de trinta alunos fictícios da disciplina de Cálculo I. Os alunos foram inseridos arbitrariamente para que o caso de uso fosse demonstrado e o código com a inserção pode ser visualizado no Apêndice D. Feitas as inserções, a *query* para visualizar os objetos de aprendizagem mais estudados pela turma é:

```
PREFIX : <http://www.semanticweb.org/win7/ontologies/2019/5/>
SELECT ?Objeto_de_Aprendizagem
      (COUNT(*) AS ?Quantidade_de_alunos_que_dominam)
WHERE{
    ?alunos :student_studies_discipline :Calculo_I .
    :Calculo_I :discipline_has_module ?modulo .
    ?modulo :module_has_learningObject ?Objeto_de_Aprendizagem .
    ?alunos :student_knows_learningObject ?Objeto_de_Aprendizagem .
}
GROUP BY ?Objeto_de_Aprendizagem ORDER BY DESC(COUNT(*))
```

A cláusula **WHERE** da *query* usa quatro linhas:

- a primeira busca alunos que estudam a disciplina de Cálculo I;
- a segunda busca os módulos existentes na disciplina de Cálculo I;
- a terceira busca os objetos de aprendizagem contidos nos módulos de Cálculo I;
- e a quarta relaciona os alunos da primeira linha com os objetos de aprendizagem da terceira linha.

O comando **COUNT** faz a contagem da variável que representa o número de alunos que

Disciplinas_Faltantes	
1	:Sistemas_Móveis
2	:Grafos
3	:Engenharia_de_Software
4	:Estrutura_de_Dados_I
5	:Sistemas_Embarcados
6	:Sistemas_Distribuídos
7	:Compiladores
8	:Probabilidade_e_Estatística
9	:Banco_de_Dados
10	:Cálculo_Numérico
11	:Estrutura_de_Dados_II
12	:Cálculo_II
13	:Sistemas_Operacionais
14	:Programação_Orientada_a_Objeto

Showing 1 to 14 of 14 entries

Figura 5.23: Informações contidas nas triplas que representam as disciplinas do programa de Ciência de Computação que Francisco ainda não cursou.

estudam os objetos de aprendizagem, enquanto o comando **GROUP BY**, da última linha da *query*, associado ao comando **ORDER BY DESC** faz o agrupamento dos resultados em ordem decrescente segundo a contagem feita. A Figura 5.24 apresenta o resultado da *query*, isto é, a quantidade de alunos que estudou cada um dos objetos de aprendizagem da disciplina de Cálculo I. Vale lembrar que, segundo a ontologia proposta, um aluno não necessariamente precisa estudar todos os objetos de aprendizagem de um módulo para dominá-lo e poder seguir adiante com o conteúdo. O domínio de um módulo, para fins de prerequisite, é indicado pelo teste do mesmo, que é um indivíduo da classe *Test*.

5.7 Considerações sobre o capítulo

Este capítulo apresentou a PoC com o objetivo de validar as *queries* realizadas na ontologia elaborada a partir de quatro casos de uso que utilizam *queries* e atualizações na linguagem SparQL.

	Objeto_de_Aprendizagem	⊗	Quantidade_de_alunos_que_dominam
1	:Limites_Lista1		"15" ^{AA} xsd:integer
2	:Derivadas_Lista1		"11" ^{AA} xsd:integer
3	:Derivadas_Lista2		"10" ^{AA} xsd:integer
4	:Limites_Lista2		"8" ^{AA} xsd:integer
5	:Integrais_Lista		"7" ^{AA} xsd:integer

Showing 1 to 5 of 5 entries

Figura 5.24: Informações contidas nas triplas que representam os objetos de aprendizagem mais estudados na disciplina de Cálculo I em uma turma de 30 alunos.

Para a implementação dos casos de uso, a ontologia *CourseStructure* foi adaptada para conter, além dos conhecimentos dos programas de aprendizagem, os perfis dos alunos e os conhecimentos por eles adquiridos. Além disso, foi utilizado um ambiente computacional com o servidor Apache Tomcat e o servlet Apache Jena Fuseki para a execução local das *queries*.

Os casos de uso abordaram operações que cobrem a atualização dos perfis dos alunos, a verificação de aquisição e lacuna de conhecimentos e o levantamento dos conteúdos mais estudados. Esses contextos servem como base para uma eventual implementação de casos mais específicos e contêm os comandos SparQL essenciais para viabilizá-los.