

4 Exemplo

Para o melhor entendimento dos conceitos expostos no capítulo 3, é apresentada aqui a declaração do mapeamento navegacional completo de uma aplicação SHDM. A aplicação utilizada como exemplo corresponde ao *website* de um departamento acadêmico, projetado para apresentação de informações sobre seus professores, alunos, áreas de pesquisa e publicações.

4.1 Modelo conceitual

O esquema conceitual da aplicação foi especificado como uma ontologia definida na linguagem OWL. A Figura 18 apresenta uma representação dessa ontologia como um diagrama no estilo UML, ilustrando o uso esperado das classes e propriedades dessa ontologia na geração dos dados da aplicação.

A definição completa dessa ontologia encontra-se disponível no apêndice “*Ontologia conceitual da aplicação exemplo*”. Um detalhe relevante para o mapeamento de classes navegacionais, descrito na próxima seção, é o fato de que todas as propriedades têm o seu domínio e contra-domínio definidos explicitamente, como na declaração da propriedade *ex:nome* que aparece no diagrama como um atributo da classe *ex:Pessoa*.

```
ex:nome
  a owl:DatatypeProperty ;
  rdfs:domain ex:Pessoa ;
  rdfs:range xsd:string .
```

Quadro 40 - Declaração da propriedade conceitual *ex:nome*

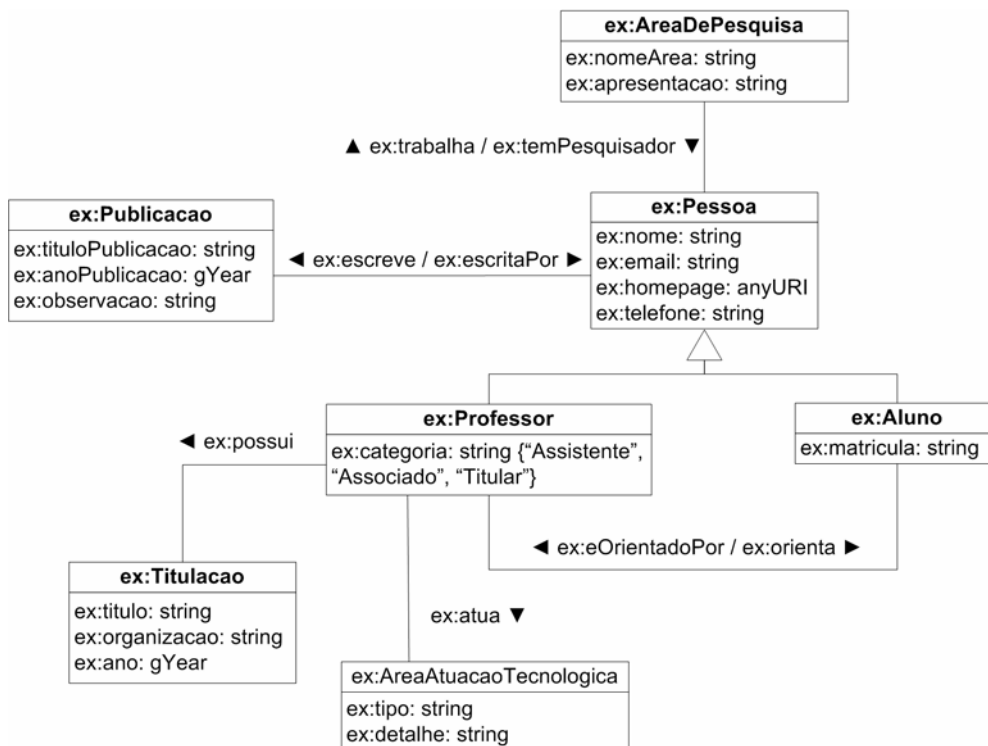


Figura 18 - Esquema conceitual para o website de um departamento acadêmico

4.2 Mapeamento navegacional

O mapeamento navegacional especificado para a aplicação define o esquema de classes navegacionais apresentado na Figura 19, correspondente a um reflexo do mapeamento de classes navegacionais e elos, cujas principais características são:

- Apenas as classes conceituais *ex:Aluno*, *ex:AreaDePesquisa*, *ex:Pessoa*, *ex:Professor* e *ex:Publicacao* foram mapeadas como classes navegacionais.
- Os atributos da classe conceitual *ex:AreaAtuacaoTecnologica* foram mapeados para uma lista (atributo *atuacao*) na classe navegacional *Professor*.
- Os atributos da classe conceitual *ex:Titulacao* foram mapeados para atributos simples da classe navegacional *Professor* (atributos *titulo*, *orgTitulacao* e *anoTitulacao*).
- O atributo *ex:categoria* de *ex:Professor* foi removido no mapeamento.

- Todos os demais atributos simples de todas as classes navegacionais foram mapeados diretamente dos atributos conceituais de mesmo nome.
- O relacionamento conceitual *ex:temPesquisador* foi mapeado em dois elos (*temAlunoPesquisador* e *temProfessorPesquisador*) que especializaram a relação em função das subclasses de *Pessoa*.
- Todos os demais elos foram mapeados diretamente dos relacionamentos conceituais de mesmo nome.
- Instâncias de *Aluno* apresentam uma âncora para a instância de *Professor* que representa seu orientador, identificado em função do elo *eOrientadoPor*.
- Instâncias de *AreaDePesquisa* apresentam, como valor do atributo *alunos*, um índice dos alunos que trabalham na área (*idxAlunosPorArea*) e, como valor do atributo *professores*, um índice dos professores que trabalham na área (*idxProfessoresPorArea*).
- As instâncias de *Pessoa* apresentam como valor do atributo *areas* um índice das áreas de pesquisa nas quais a pessoa trabalha (*idxAreasPorPessoa*), e como valor do atributo *publicacoes* um índice das publicações escritas pela pessoa (*idxPublicacoesPorPessoa*).
- As instâncias de *Professor* apresentam um índice dos alunos orientados pelo professor (*idxAlunosPorProfessor*).
- Instâncias de *Publicação* apresentam um índice das pessoas que são seus autores (*idxPessoasPorPublicacao*).

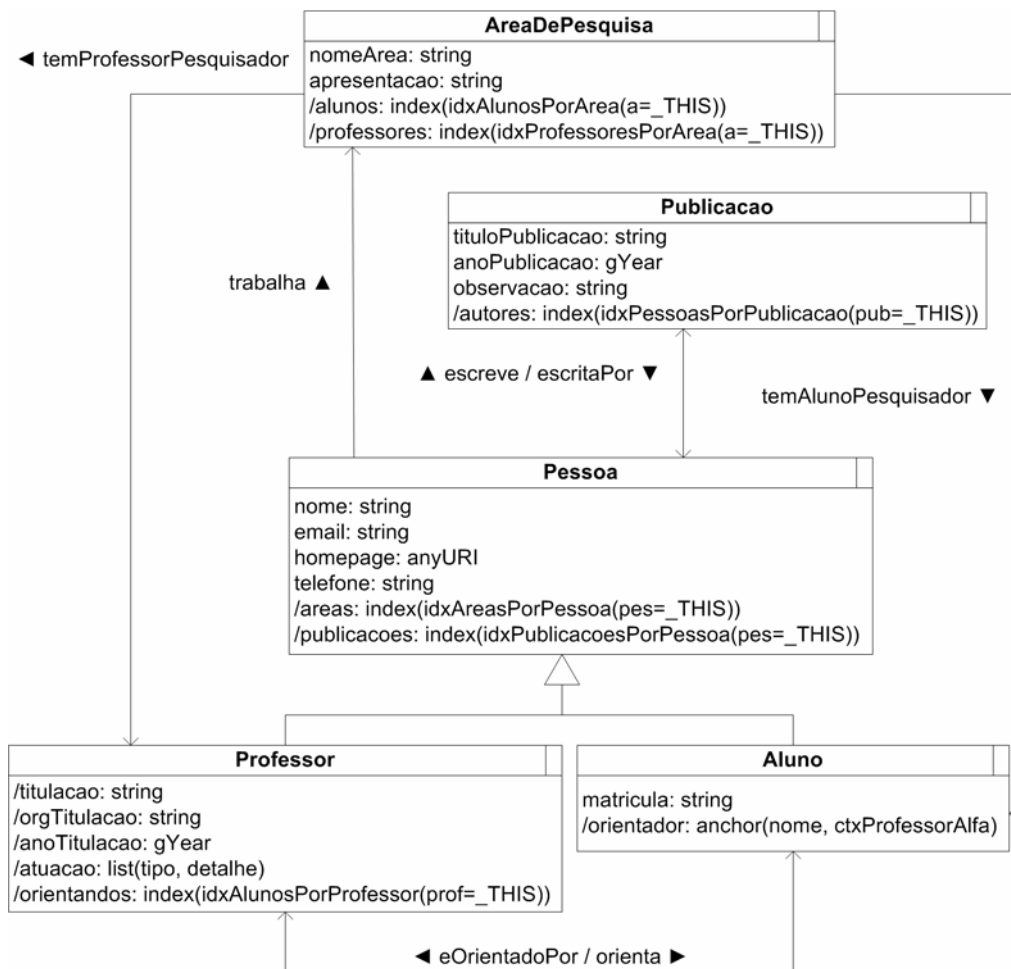


Figura 19 - Esquema de classes navegacionais para o *website* do departamento acadêmico

A especificação completa do mapeamento de classes navegacionais e elos encontra-se disponível no apêndice “*Especificação do mapeamento navegacional da aplicação exemplo*”. A seguir discutiremos alguns detalhes desta especificação.

Como o modelo conceitual foi definido em OWL com a declaração do domínio de todas as suas propriedades, as especificações dos mapeamentos das classes navegacionais foram declaradas deixando ativada por padrão a opção de mapeamento automático de atributos da classe conceitual base. Por isso, apenas os atributos que não foram mapeados diretamente do modelo conceitual foram declarados nos mapeamentos das classes. Isto pode ser visto na declaração do mapeamento da classe *Publicacao* apresentado a seguir, no qual apenas o atributo correspondente ao índice de autores precisou ser declarado explicitamente.

```

:Publicacao
  a shdm:NavClass ;
  shdm:baseclass ex:Publicacao ;
  shdm:attribute [
    a shdm:IndexAttribute ;
    shdm:attributeName "autores" ;
    shdm:indexValue :idxPessoasPorPublicacao ;
    shdm:parameter [
      a shdm:ParameterBinding ;
      shdm:parameterName "@pub";
      shdm:parameterValue "_THIS"
    ]
  ] .

```

Quadro 41 - Declaração do mapeamento da classe navegacional *Publicacao*

Abaixo se encontra um trecho da declaração do mapeamento de *Professor*, que mostra como o atributo *titulacao* foi mapeado a partir do atributo *ex:titulo* da classe conceitual *ex:Titulacao*, acessada através da propriedade *ex:possui*. Os atributos *orgTitulacao* e *anoTitulacao* foram mapeados de forma semelhante.

```

:Professor
  a shdm:NavClass ;
  shdm:baseclass ex:Professor ;

  . . .

  shdm:attribute [
    a shdm:MappedAttribute ;
    shdm:attributeName "titulacao" ;
    shdm:fromPath [
      a rdf:Seq ;
      rdf:li ex:possui , ex:titulo ;
    ]
  ]

  . . .

```

Quadro 42 - Declaração do mapeamento do atributo *titulacao* da classe navegacional *Professor*

A Figura 20 apresenta o diagrama de contextos definido para esta aplicação. Pode-se ver como os índices de áreas de pesquisa (*idxAreasDePesquisa*), alunos (*idxAlunos*), e professores (*idxProfessores*) podem ser acessados a partir de um menu principal da aplicação. Este menu principal por sua vez foi especificado como um índice arbitrário, podendo ser acessado de qualquer ponto da aplicação, como indicado pela presença do *landmark* correspondente.

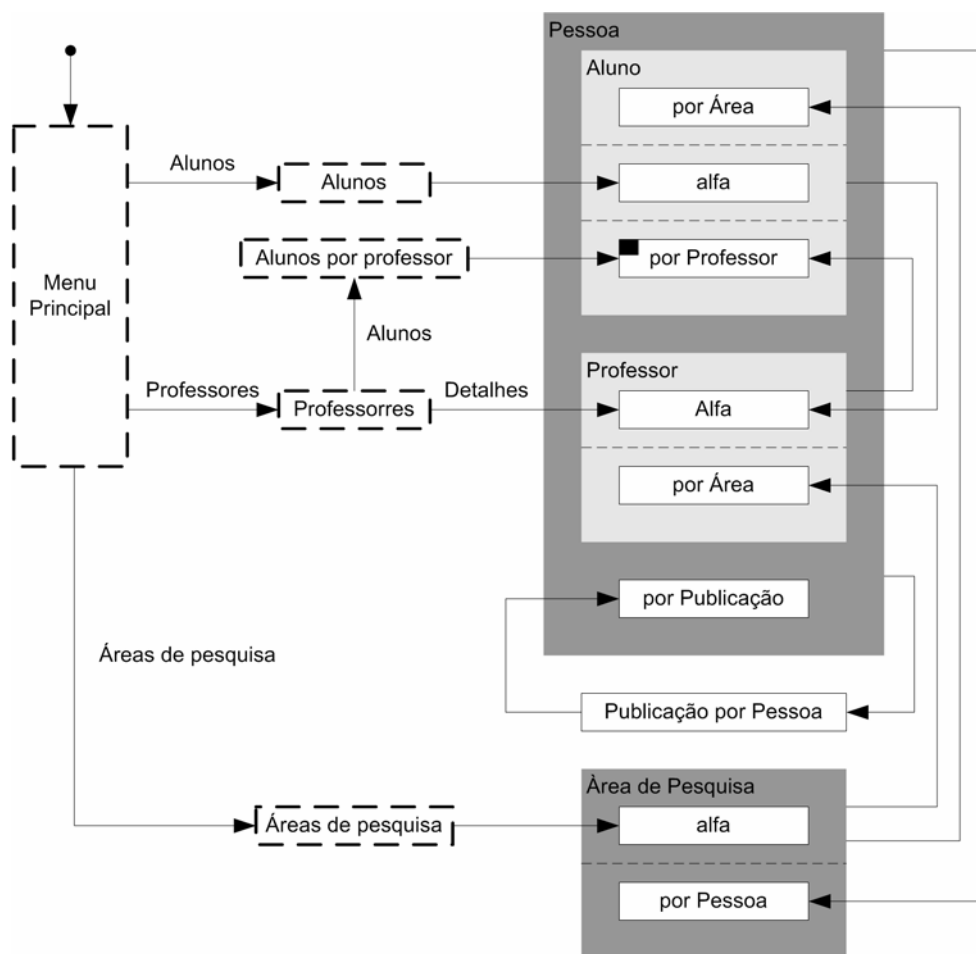


Figura 20 - Diagrama de contextos para o website do departamento acadêmico

Os contextos da aplicação foram especificados seguindo dois padrões de declaração, identificados de acordo com a estrutura de suas definições. O primeiro corresponde a contextos que selecionam todos os nós de um tipo específico. Este é o caso do contexto de alunos em ordem alfabética (*ctxAlunoAlfa*), cujo cartão de especificação e respectiva declaração são apresentados a seguir.

Contexto: ctxAlunoAlfa
Parâmetros: -
Elementos: nSchema:Alunos
Ordenação: nome ASC
Navegação interna: SEQ
Comentários:

Figura 21 - Cartão de contexto de alunos em ordem alfabética, *ctxAlunosAlfa*

```

:ctxAlunoAlfa
  a shdm:Context ;
  shdm:selects [
    a shdm:ElementsSelection ;
    shdm:elementType nSchema:Aluno ;
  ] ;
  shdm:defaultOrdering :ordenacaoPorNome ;
  shdm:navigation "SEQ" .

:ordenacaoPorNome
  a shdm:OrderingCriteria ;
  shdm:orderBy [
    a rdf:Seq ;
    rdf:li [
      a shdm:OrderingCriteriaTerm ;
      shdm:attributeName "nome" ;
      shdm:order "ASC"
    ]
  ] .

```

Quadro 43 - Declaração do contexto de alunos em ordem alfabética, *ctxAlunosAlfa*

Enquadram-se neste tipo de definição, além do exemplo já mencionado, os contextos *ctxProfessorAlfa*, e *ctxAreaDePesquisaAlfa*.

O outro padrão corresponde à definição de grupos de contextos, nos quais são selecionados nós de um único tipo em função de um nó associado que é passado como parâmetro para o contexto. Foram definidos desta forma os contextos de alunos por área (*ctxAlunoPorArea*), alunos por professor (*ctxAlunoPorProfessor*), área de pesquisa por pessoa (*ctxAreaDePesquisaPorPessoa*), professores por área (*ctxProfessorPorArea*), pessoas por publicação (*ctxPessoaPorPublicacao*), e publicações por pessoa (*ctxPublicacaoPorPessoa*). A seguir encontra-se o cartão de contexto e declaração para o contexto *ctxPessoaPorPublicação*.

Contexto: ctxPessoaPorPublicação
Parâmetros: @pub: nSchema:Publicação
Elementos: nSchema:Pessoa ?pessoa WHERE ?pessoa nSchema:escreve @pub
Ordenação: nome ASC
Navegação interna: CIRC
Comentários:

Figura 22 - Cartão de contexto de pessoa por publicação, *ctxPessoaPorPublicacao*

```

:ctxPessoaPorPublicacao
  a shdm:Context ;
  shdm:parameter [
    a shdm:ParameterDefinition ;
    shdm:parameterName "@pub" ;
    shdm:parameterType nSchema:Publicacao ;
    shdm:bindsTo nSchema:escreve
  ] ;
  shdm:selects [
    a shdm:ElementsSelection ;
    shdm:elementType nSchema:Pessoa ;
    shdm:elementVariable "?pessoa" ;
    shdm:condition "(?pessoa nSchema:escreve @pub)";
    shdm:queryLanguage "RDQL"
  ] ;
  shdm:defaultOrdering :ordenacaoPorNome ;
  shdm:navigation "CIRC" .

```

Quadro 44 - Declaração do contexto de pessoa por publicação, *ctxPessoaPorPublicacao*

Em sua maioria, as estruturas de acesso da aplicação foram especificadas como índices derivados dos contextos para os quais apontam, sendo definidos de forma análoga ao índice de áreas de pesquisa (*idxAreasDePesquisa*) cujo cartão e declaração são apresentados a seguir.

Estrutura de acesso: idxAreasDePesquisa	
Parâmetros: -	
Elementos: fromContext(ctxAreaDePesquisaAlfa)	
Atributos	
nome	valor
nome	Anchor(nSchema:nomeArea)
Ordenação: nome ASC	
Comentários:	

Figura 23 - Cartão da estrutura de acesso *idxAreasDePesquisa*


```
:idxAreasDePesquisa
  a shdm:Index ;
  shdm:fromContext :ctxAreaDePesquisaAlfa ;
  shdm:attribute [
    a shdm:AnchorAttribute ;
    shdm:attributeName "nome" ;
    shdm:anchorValue [
      a shdm:Anchor ;
      shdm:label nSchema:nomeArea
    ]
  ] ;
  shdm:defaultOrdering :ordenacaoPorNome .
```

Quadro 45 - Declaração da estrutura de acesso *idxAreasDePesquisa*

A exceção é o índice de professores (*idxProfessores*), que foi declarado como um índice derivado de consulta. Em sua definição foi declarada uma consulta para recuperação do URI, o nome e o e-mail de todos os recursos instâncias de *Professor*. Os valores recuperados nessa consulta serão utilizados para criação das entradas do índice no formato definido pela especificação de atributos. Neste caso, cada entrada irá apresentar uma âncora que exhibe o nome do professor e aponta para o nó correspondente no contexto de professores em ordem alfabética (*ctxProfessorAlfa*), o e-mail do professor, e uma âncora exibindo a etiqueta “Alunos”, apontando para o índice de alunos por professor (*idxAlunosPorProfessor*) e passando o professor correspondente como parâmetro. Segue o cartão de contexto e respectiva declaração deste índice.

Estrutura de acesso: idxProfessores	
Parâmetros: -	
Elementos: SELECT ?prof, ?nome?, ?email WHERE (?prof rdf:type nSchema:Professor) (?prof nSchema:nome ?nome) (?prof nSchema:email ?email)	
Atributos	
nome	valor
nome	Anchor(?nome, ctxProfessorAlfa, {_ID=?prof})
email	?email
alunos	Anchor("Alunos", idxAlunoPorProfessor, {@prof=?prof})
Ordenação: nome ASC	
Comentários:	

Figura 24 - Cartão da estrutura de acesso *idxProfessores*

```

:idxProfessores
  a shdm:Index ;
  shdm:query [
    a shdm:IndexQuery ;
    shdm:variable "?prof" , "?nome" , "?email" ;
    shdm:condition "
      (?prof rdf:type nSchema:Professor)
      (?prof nSchema:nome ?nome)
      (?prof nSchema:email ?email)" ;
    shdm:queryLanguage "RDQL"
  ] ;
  shdm:attribute [
    a shdm:AnchorAttribute ;
    shdm:attributeName "nome" ;
    shdm:anchorValue [
      a shdm:Anchor ;
      shdm:label "?nome" ;
      shdm:target :ctxProfessorAlfa ;
      shdm:parameter [
        a shdm:parameterBinding ;
        shdm:parameterName "_ID" ;
        shdm:parameterValue "?prof"
      ]
    ]
  ] , [
    a shdm:SimpleAttribute ;
    shdm:attributeName "email" ;
    shdm:simpleValue "?email"
  ] , [
    a shdm:AnchorAttribute ;
    shdm:attributeName "alunos" ;
    shdm:anchorValue [
      a shdm:Anchor ;
      shdm:label "Alunos" ;
      shdm:target :idxAlunosPorProfessor ;
      shdm:parameter [
        a shdm:parameterBinding ;
        shdm:parameterName "@prof" ;
        shdm:parameterValue "?prof"
      ]
    ]
  ] ;
  shdm:defaultOrdering :ordenacaoPorNome .

```

Quadro 46 - Declaração da estrutura de acesso *idxProfessores*

Após a especificação do mapeamento navegacional, o próximo passo é a declaração do projeto da interface abstrata, incluindo o mapeamento dos dados das classes navegacionais, contextos, estruturas de acesso e *landmarks* para os elementos de interface definidos nessa interface abstrata. Como mencionado anteriormente, o projeto da interface abstrata não é abordado no presente trabalho.

A seguir será apresentada a arquitetura de implementação projetada para o processamento das aplicações definidas segundo método SHDM, como a do exemplo que acabou de ser apresentado.