5 Exemplo de desenvolvimento de uma aplicação

Neste item será exemplificado um fluxo de desenvolvimento de uma aplicação SHDM .Net.

5.1. Ambiente de Desenvolvimento

Todo o desenvolvimento de uma aplicação SHDM .Net é efetuado dentro do Visual Studio 2005. Para iniciarmos o processo devemos abrir o VS 2005 e criar um novo projeto através do link *File* → *New* → *Project*. A Figura 30 ilustra a ação.

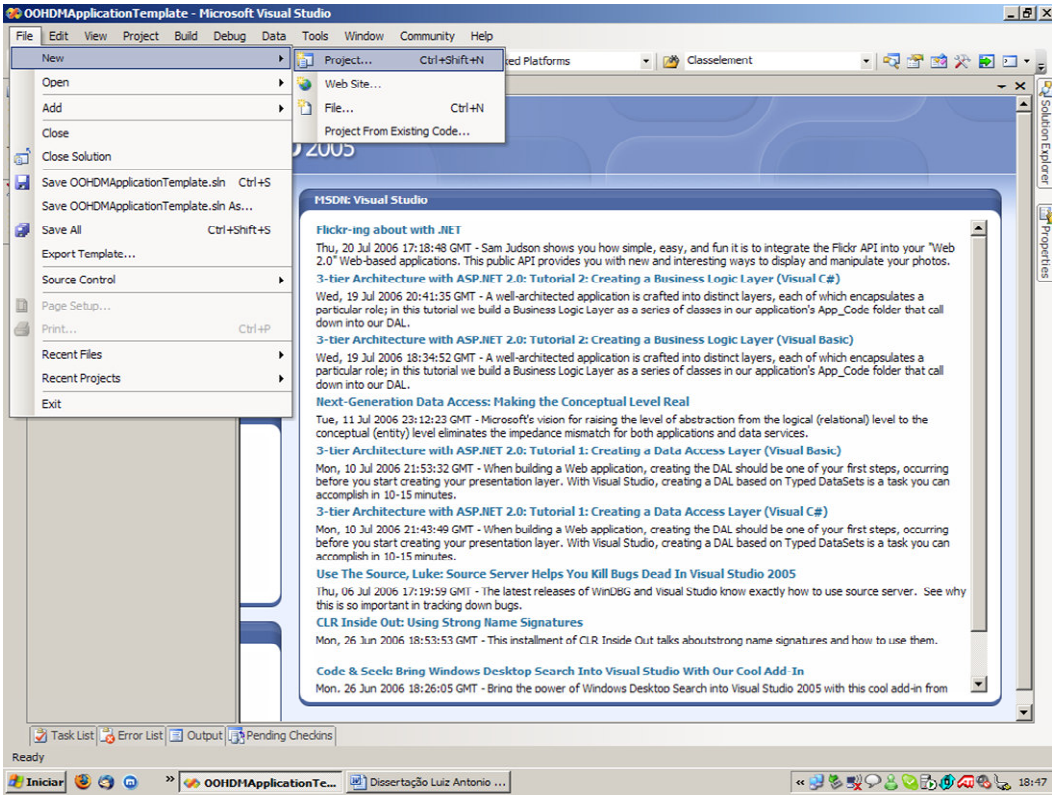

Figura 30 – Tela inicial Visual Studio

A tela da Figura 31 é apresentada para que sejam preenchidos, o nome do projeto, o caminho de diretórios onde o projeto será armazenado e o nome da solução da qual o projeto fará parte.

O tipo do projeto a ser selecionado é o Visual C# e o *template* do projeto deve ser *Class Library*. Este projeto armazenará a lógica de negocio da aplicação.

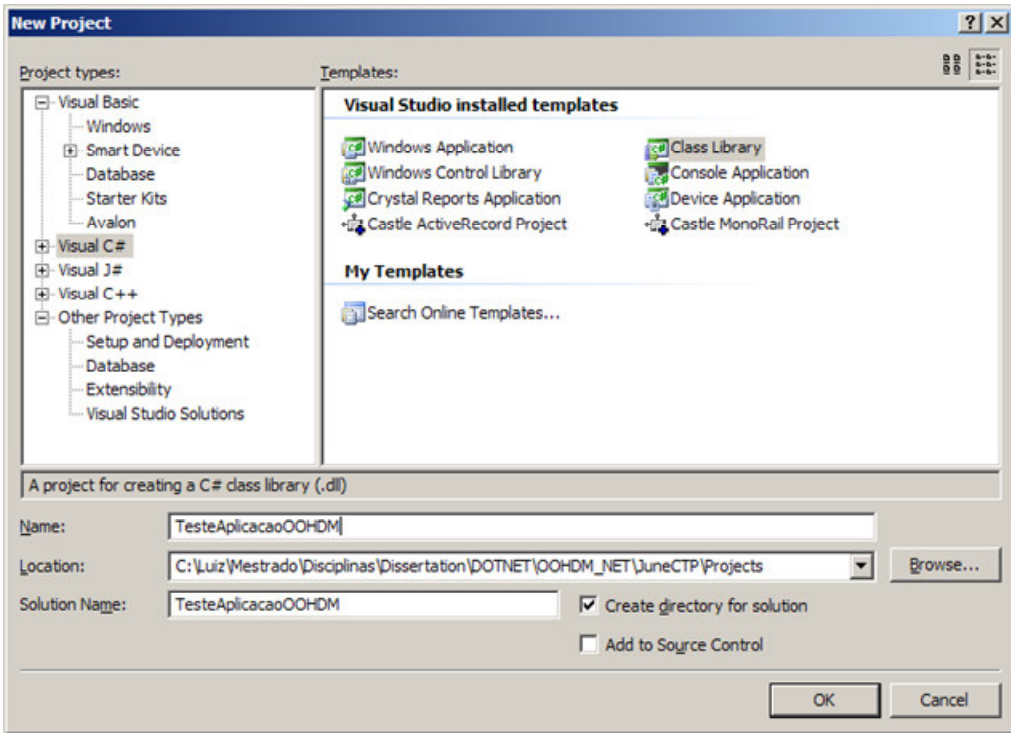

Figura 31 – Criando um novo projeto

Após a criação do projeto o Diagrama de Classes Navegacionais deve ser criado. Para isto deve-se clicar com o botão direito do mouse no projeto e no menu que será apresentado deve-se selecionar *Add* → *New Item* como ilustra a Figura 32.

Na tela de seleção do item a ser adicionado deve ser selecionado o elemento Navigational, como demonstra a Figura 33. Um nome deve ser preenchido para o item, por exemplo, NavigationalDiagram.

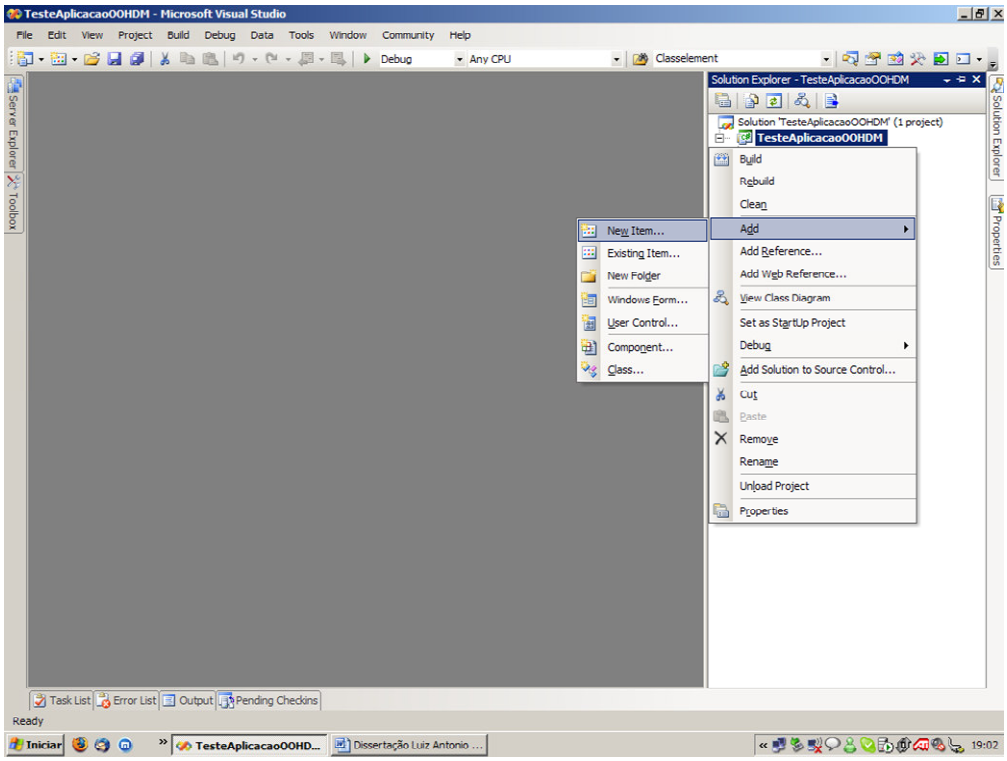

Figura 32 – Adicionando um novo item no projeto

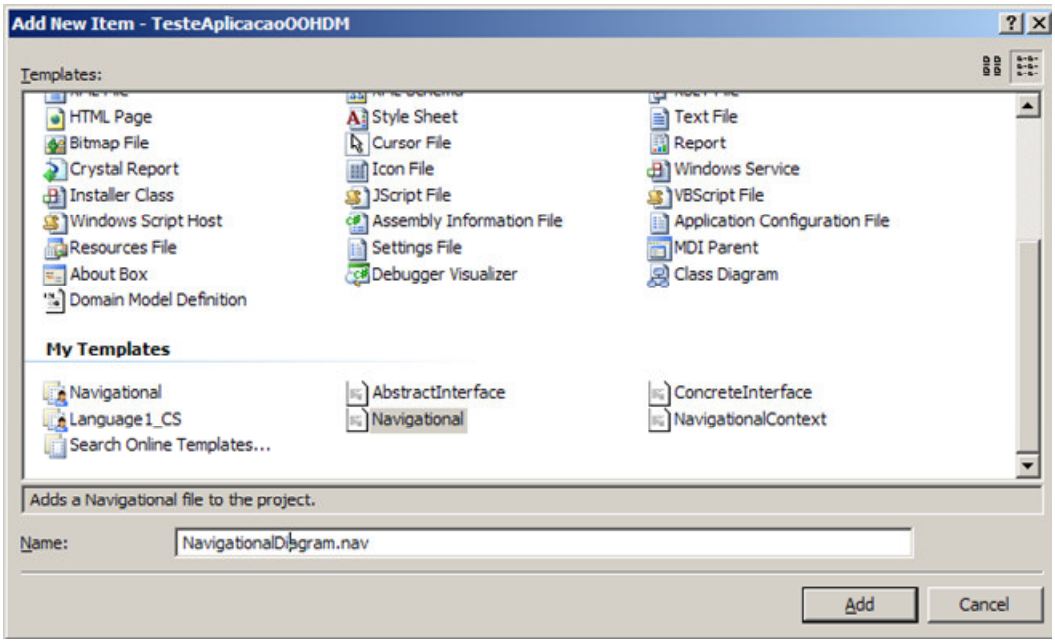

Figura 33 – Adicionando um diagrama de classes navegacionais

Uma tela de edição do diagrama de classes navegacionais é apresentada. Na Toolbox desta tela aparecem os elementos que fazem parte do diagrama. Para se criar uma classe navegacional é necessário clicar no item “Navigational Class” da

toolbox (quadrado vermelho na Figura 34), e arrastá-lo para a área de design do diagrama. Os relacionamentos entre as classes navegacionais são criados através do clique no relacionamento desejado na toolbox seguido do clique no elemento de origem do relacionamento sendo finalizado por um clique no elemento de destino do relacionamento.

Todos os diagramas do SHDM .Net, exceto o diagrama de interface concreta, funcionam de maneira análoga, ou seja, as classes navegacionais, estruturas de acesso, contextos e elementos de interface abstrata devem ser adicionados ao diagrama da mesma forma; clicando-se no ícone do elemento desejado na toolbox e arrastando-o para a área de design. No caso dos links ou relacionamentos entre elementos, estes deve ser adicionados efetuando-se um clique na toolbox, seguido de um segundo clique no elemento de origem do relacionamento, finalizando com um clique no elemento de destino.

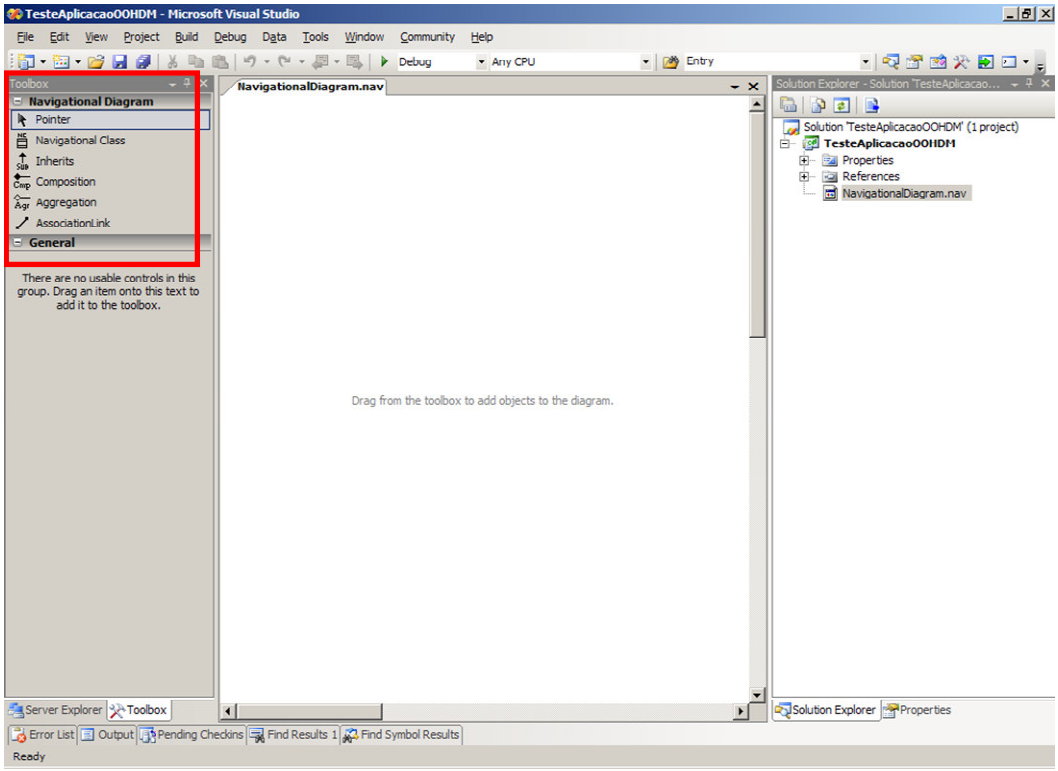

Figura 34 – Tela de Edição do diagrama de classes navegacionais

No OOHDM / SHDM um projeto navegacional envolve o desenvolvimento de dois diagramas quase simultaneamente. No SHDM .Net este desenvolvimento deve ser simultâneo, pois o diagrama de classes navegacionais referencia elementos criados no diagrama de contextos navegacionais e vice-versa. Devido a

esta comunicação é recomendável que seja adicionado um diagrama de contextos navegacionais ao projeto.

Para isto deve-se clicar com o botão direito do mouse no projeto e no menu que será apresentado deve-se selecionar *Add* → *New Item* como ilustra a Figura 32.

Na tela de seleção do item a ser adicionado deve ser selecionado o elemento NavigationalContext, como demonstra a Figura 35. Um nome deve ser preenchido para o item, por exemplo, NavigationalContext.

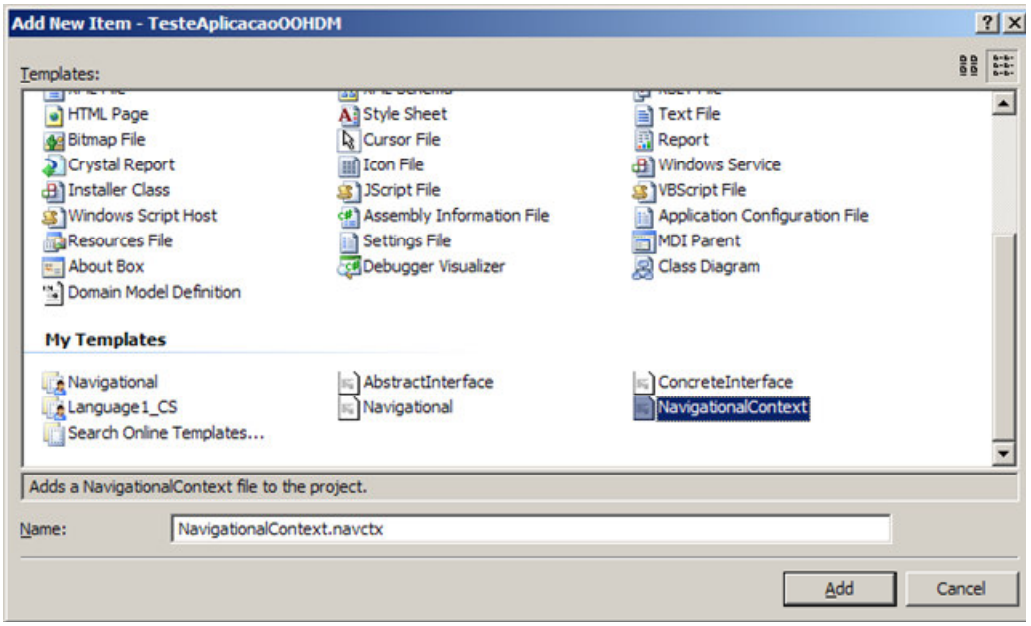

Figura 35 – Adicionando um diagrama de contextos navegacionais

O SHDM .Net possui uma integração entre os diagramas de classes navegacionais e o diagrama de contextos navegacionais. Esta integração permite que elementos definidos em um diagrama possam ser visualizados e utilizados no outro. Para que esta integração seja habilitada é necessário fazer a vinculação entre os diagramas. Esta vinculação é efetuada clicando-se na área branca do diagrama de classes navegacionais e na aba Properties deve ser selecionado, na propriedade NavigationalContextDiagram, o arquivo que contém o diagrama de contextos navegacionais. Como ilustra a Figura 36.

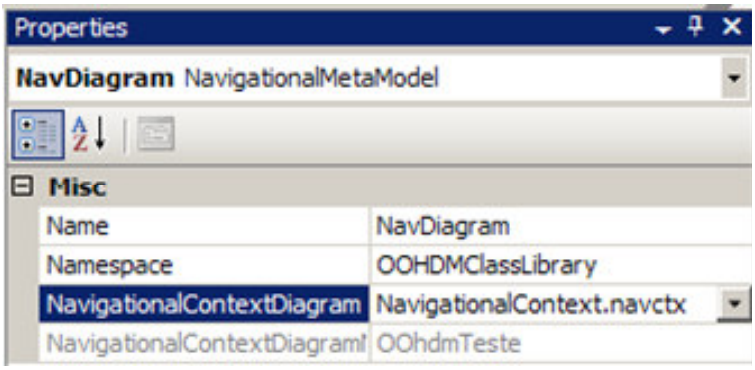

Figura 36 – Vinculação do diagrama de contextos navegacionais no diagrama de classes navegacionais.

Um processo análogo deve ser efetuado no diagrama de contextos navegacionais para que este visualize as classes definidas no diagrama de classes navegacionais. Esta vinculação é efetuada clicando-se na área branca do diagrama de contextos navegacionais e na aba Properties deve ser selecionado, na propriedade NavigationalClassDiagram, o arquivo que contém o diagrama de classes navegacionais. Como ilustra a Figura 37.

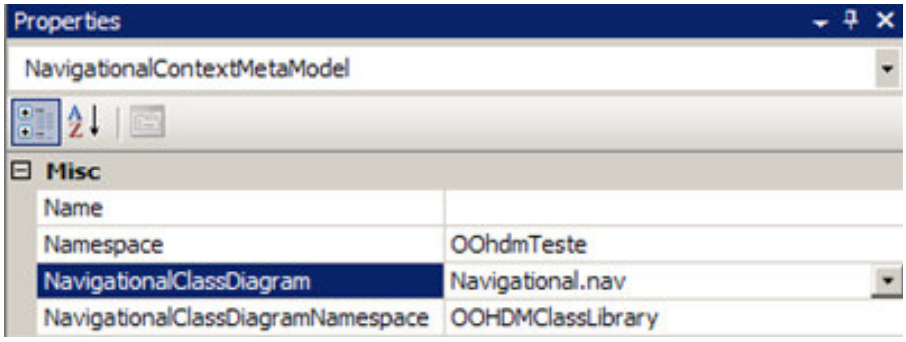

Figura 37 - Vinculação do diagrama de classes navegacionais no diagrama de contextos navegacionais.

5.2. Criação do Diagrama de classes Navegacionais

Exemplo da criação de um diagrama de classes navegacionais utilizando a ferramenta. Na Figura 38 é demonstrada a forma de criação de um atributo ou operação em uma classe navegacional. Para isto deve-se clicar com o botão direito no elemento selecionar o item de menu *Add,* e selecionar o tipo de atributo ou operação desejado.

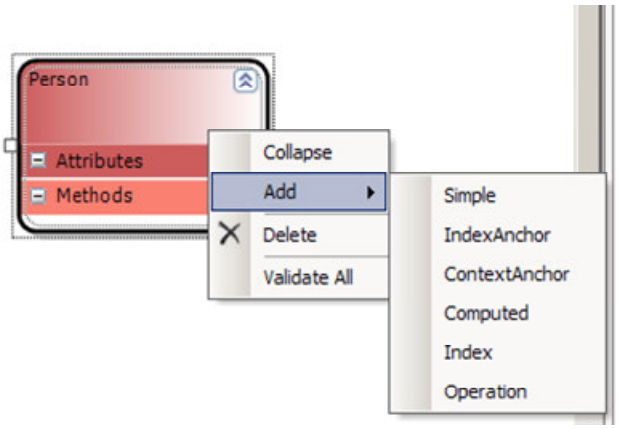

Figura 38 – Adicionando um atributo a uma classe navegacional

Todo o elemento criado no diagrama tem propriedades a serem preenchidas. Estas propriedades estão disponíveis para edição na aba *Properties* do Visual Studio, como demonstra a Figura 39.

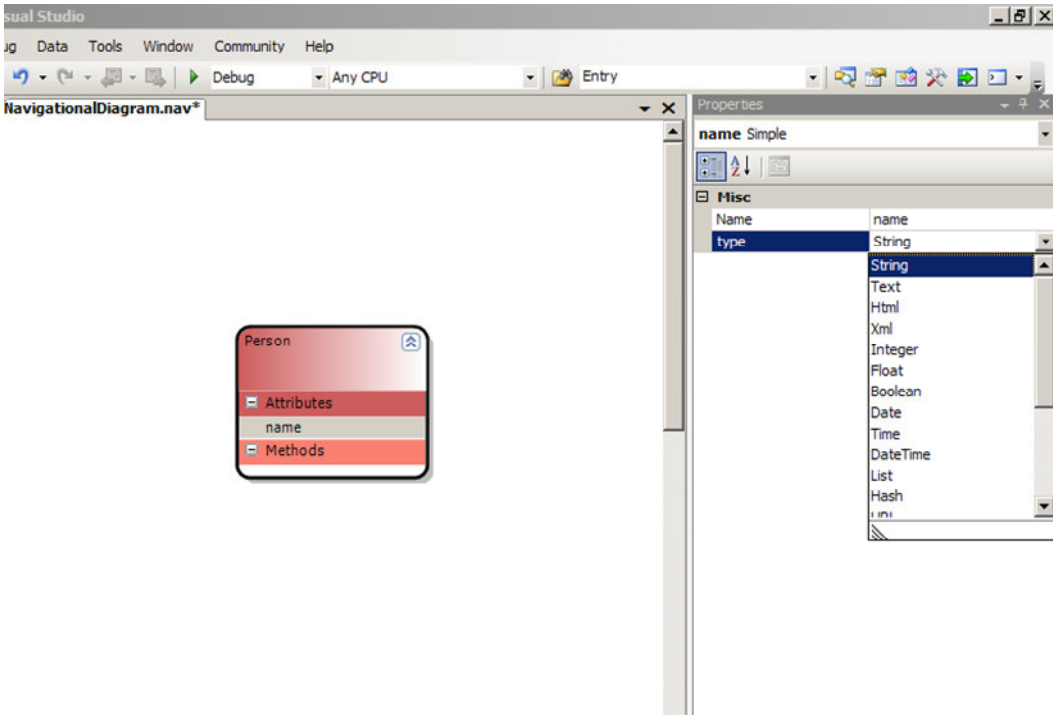

Figura 39 – Alterando as propriedades do atributo

Quando o diagrama de classes navegacionais ou o diagrama de contextos navegacionais são salvos, automaticamente a DSL-SHDM dos elementos inseridos no diagrama é produzida.

Um exemplo da DSL-SHDM produzida pelo diagrama apresentado na Figura 39 é apresentado no Quadro 4. Toda a instância de uma classe produzida pelo SHDM .Net tem um identificador único, este identificador e utilizado em

consultas RDF e na parte de persistência dos dados em RDF. No código podemos verificar que foi gerada uma propriedade globalIdentifier. Esta propriedade é responsável pela identificacao unívoca de uma instância de classe navegacional.

```
      #region Imports
      using System;
      using System.Collections;
      using System.Collections.Generic;
      #endregion

namespace NavigationalClassTest
{

    #region Classes Auxiliares

///Classe reponsável por armazenar uma âncora para um contexto ou
índice.
    public partial class Anchor
    {
        string _link = "";
        string _text = "";

        public Anchor(object targetContext, object targetElement,
Hashtable parameters)
        {
            ContextBehaviour context = targetContext as
ContextBehaviour;
            this._link = context.getLinkForElement(targetElement,
parameters);
        }
        public Anchor(){ }

        public string link
        {
            get { return this._link; }
            set { this._link = value; }
        }

        public string text
        {
            get { return this._text; }
            set { this._text = value; }
        }
    }
```

```
      #endregion

    #region Person
    public partial class Person :
OOHDMLibrary.DevelopmentBase.IGlobalIdentifier
    {
        #region Attributes
        private int _globalIdentifier;
        public int Identifier()
        {  return this.globalIdentifier;
        }
        public void GenerateNewIdentifier()
        {
            Random randomicNumber = new Random();
            this._globalIdentifier = randomicNumber.Next();
        }

        public int globalIdentifier
        {  get { return this._globalIdentifier; }
        }

        public Person(int globalIdentifier)
        {  this._globalIdentifier = globalIdentifier;
        }

        public Person()
        {   Random randomNumber = new Random();
            this._globalIdentifier = randomNumber.Next();
        }

        public static List<Person> getList()
        {
            return new List<Person>();
        }

        // Atributo name
        private string a_name = "";
        public string name
        {
            get { return this.a_name; }
            set { this.a_name = value; }
        }// Fim Atributo name
        #endregion
```

```
        #region Attributes Generated by Association Relationship
        #endregion

        #region Attributes Generated by Composite Relationship
        #endregion

        #region Attributes Generated by Aggregation Relationship
        #endregion
    }
    #endregion
}
```

Quadro 4 – Exemplo da DSL-SHDM produzida pelo diagrama da Figura 39.

Na Figura 40 é apresentado um exemplo de um diagrama de classes navegacionais finalizado. No exemplo, temos duas classes Estudante (*Student*) e Professor (*Professor*) que são subclasses de Pessoa (*Person*) e um relacionamento entre as subclasses cuja semântica diz que um professor orienta diversos alunos e um aluno é orientado por um professor. Um aluno tem zero ou uma área de pesquisa (*ResearchArea*) e um professor tem diversas áreas de pesquisa. O Quadro 5 mostra um parte do código gerado da classe *Student*.

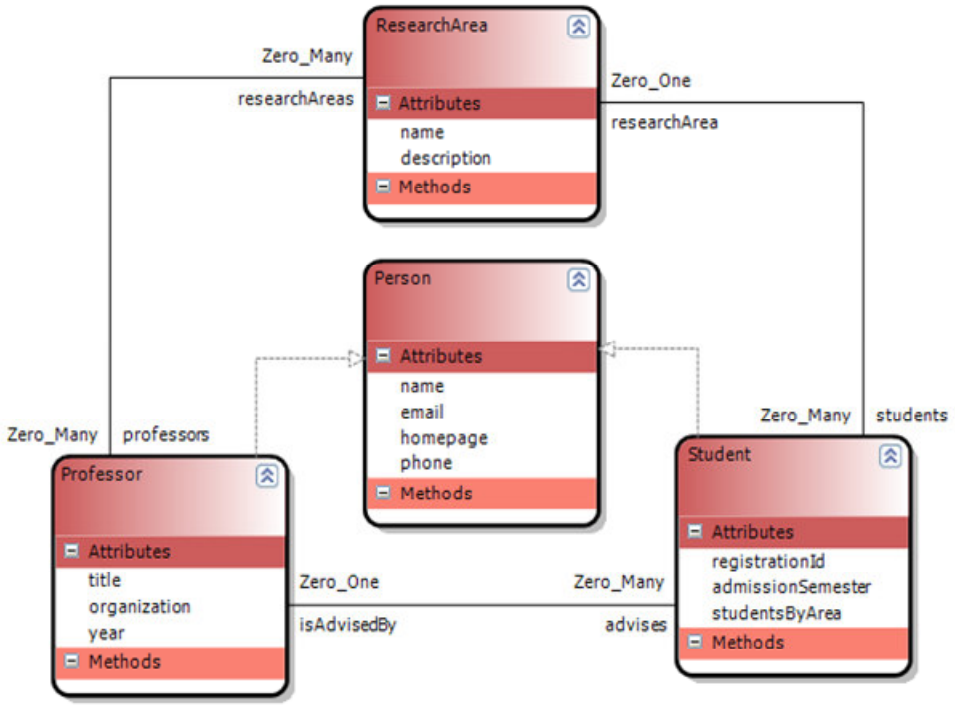

Figura 40 – Diagrama de classes navegacionais finalizado

```
public partial class Student: Person,
OOHDMLibrary.DevelopmentBase.IGlobalIdentifier
        {
            #region Attributes

        // Atributo registrationId
            private int a_registrationId = 0;
            public int registrationId
            {
                  get{return this.a_registrationId;}
                  set{this.a_registrationId = value;}
            }// Fim Atributo registrationId

            // Atributo Computado admissionSemester
            public string  admissionSemester
            {
                get{ string regID =
this.registrationId.ToString();  return regID.Substring(0, 3); }

            }//Fim Atributo Computado admissionSemester

      // Atributo Âncora para o contexto StudentByReserchArea
            Anchor _studentsByArea = null;
            Public      Anchor  studentsByArea
            {
              get {
               if (_studentsByArea == null)
              {
                Hashtable parameters = new Hashtable();
                    parameters["targetElement"] = this;
                    parameters["area"] = this.researchArea;
                    this._studentsByArea = new Anchor(new
StudentByResearchArea(), parameters["targetElement"],
parameters);
                _studentsByArea.text = this.researchArea.name;
              }
                return _studentsByArea;
             }
            }
//Fim do Atributo Âncora para o contexto StudentByReserchArea
        #endregion

      #region Attributes Generated by Association Relationship
```

```
            #region  Relationship Advises
                private Professor a_isAdvisedBy =  null;
                      public Professor isAdvisedBy
                      {
                          get{return this.a_isAdvisedBy;}
                          set{this.a_isAdvisedBy = value;}
                      }
            #endregion

            #region  Relationship Works
                private ResearchArea a_researchArea =  null;
                      public ResearchArea researchArea
                      {
                            get{return this.a_researchArea;}
                            set{this.a_researchArea = value;}
                      }
            #endregion
        #endregion
    }
    #endregion
```

Quadro 5 – Exemplo de parte da DSL-SHDM da classe Student

5.3.Diagrama de Contextos Navegacionais

No diagrama da Figura 41 é apresentado um diagrama de contextos navegacionais. Neste diagrama podemos observar os índices: StudentsAlpha, StudentsByProfessor, ProfessorsAlpha e ResearchAreasAlpha; os contextos StudentAlpha, StudentByProfessor, StudentByResearchArea, ProfessorAlpha e ResearchAreaAlpha. Os índices StudentsAlpha, ProfessorsAlpha e ResearchAreasAlpha são landmarks, ou seja, podem ser acessados a qualquer momento na navegação da aplicação. Os escopos são responsáveis pela simplificação dos diagramas. Podemos observar que o escopo ScopeStudent tem uma âncora para o contexto StudentByResearchArea, esta âncora saindo do escopo equivale a dizer que todos os contextos que estão aninhados dentro do ScopeStudent possuem este link para o contexto, logo para fins de simplificação do diagrama, esta âncora é representada uma única vez através do escopo.

Uma parte da DSL-SHDM produzida tendo como base o diagrama da Figura 41 é apresentado no Quadro 6 e Quadro 7.

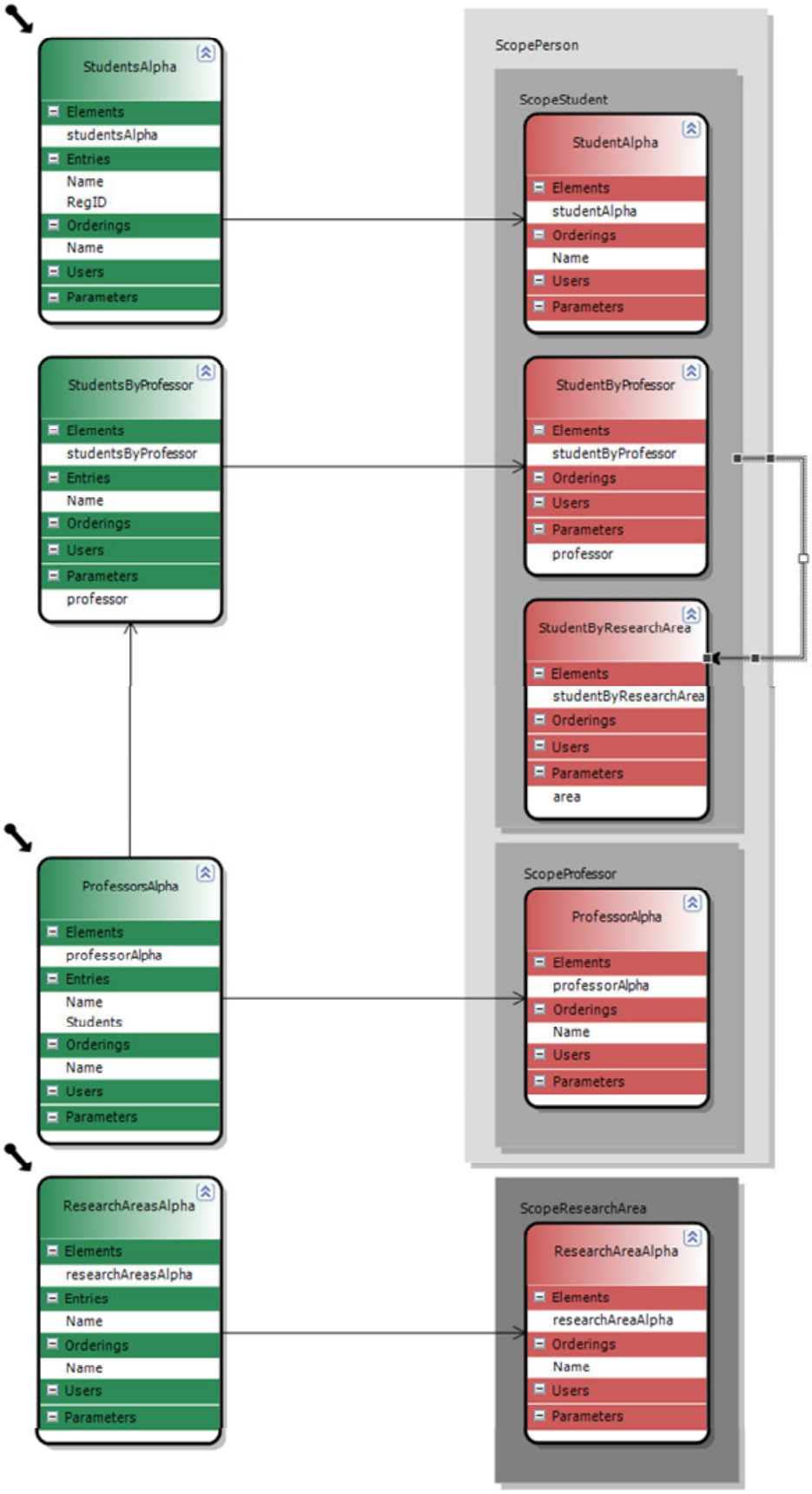

Figura 41 – Diagrama de contextos navegacionais

```
#region Access Structure StudentsAlpha
public partial class StudentsAlpha
 {
     StudentAlpha studentsAlpha = null;
      public  StudentsAlpha(){
      studentsAlpha = new StudentAlpha();
      }

     public new Hashtable show(Hashtable parameters)
     {
         Hashtable result = new Hashtable();
         Landmark land = new Landmark();
         result =
HashHelper.joinHashes(result,land.load(parameters));

          if (parameters["View"] == null)
            {
                result["%View%"] = "StudentsAlpha.aspx";
            }
            else
            {
                result["%View%"] = parameters["View"].ToString();
            }
         return result;
      }

      public string getLinkForElement( Hashtable parameters)
     {
         string result = "?";
         result += "Controller=" + this.GetType().FullName ;
                  return result;
     }
       public Hashtable load(Hashtable parameters)
       {
         System.Collections.Generic.SortedList<string,
StudentsAlpha_Data> resultList = new SortedList<string,
StudentsAlpha_Data>();
            Hashtable result = new Hashtable();
            Hashtable parametersLink = new Hashtable();
            StudentsAlpha_Data elementData = null;
            IEnumerator<Student> enumera =
studentsAlpha.getAllElementsOfContext(parameters).GetEnumerator()
;
        while (enumera.MoveNext())
        {
            parametersLink = new Hashtable();
```

```
            elementData = new StudentsAlpha_Data();
            elementData.id =  enumera.Current.globalIdentifier;
            elementData.Name = enumera.Current.name.ToString();
            elementData.Name_lnk = new
StudentAlpha().getLinkForElement(enumera.Current, parameters);
            elementData.id =  enumera.Current.globalIdentifier;
            elementData.RegID =
enumera.Current.registrationId.ToString();
            resultList.Add(elementData.Name ,elementData);
        }
        result["StudentsAlpha"]= resultList.Values;
        return result;
       }
  }
 #endregion
```

Quadro 6 – Exemplo do código da DSL-SHDM produzido da Estrutura de Acesso StudentsAlpha.


```
#region Context StudentAlpha
 public partial class StudentAlpha : ContextBehaviour
 {
      public Student current = null;
      public Student previous = null;
      public Student next = null;
      public string Name = "StudentAlpha";
      List<Student> content = null;
      OOHDMLibrary.DevelopmentBase.DatabaseInstance dtb = null;
      private Hashtable creationParameters = new Hashtable();

      /* Constructor */
      public  StudentAlpha(){
      dtb =
OOHDMLibrary.DevelopmentBase.DatabaseInstance.getInstance();
      }

       private new Hashtable load(Hashtable parameters)
       {
         creationParameters = (Hashtable) parameters.Clone();
         loadParameters(parameters);
         content = selectAllContextContent(parameters);
         int currentposition = -1;
         current =
dtb.getObjectByGlobalIdentifier(Convert.ToInt32(parameters["Stu
dent"])) as Student;
         currentposition = content.IndexOf(current);
```

```
        previous = ((currentposition - 1) >= 0) ?
content[currentposition-1] : null;
        next = ((currentposition + 1) < content.Count) ?
content[currentposition + 1] : null;
        Hashtable result = new Hashtable();
        result.Add("StudentAlpha", this);
        return result;
      }

    public List<Student> selectAllContextContent(Hashtable
parameters)
      {
        creationParameters = (Hashtable) parameters.Clone();
        loadParameters(parameters);
        return
(List<Student>)dtb.selectAllObjectsOfType(typeof(Student));
      }

    private bool loadParameters(Hashtable parameters)
    {
      return true;
    }

    public new Hashtable show(Hashtable parameters)
     {
        creationParameters = (Hashtable) parameters.Clone();
        Hashtable result = new  Hashtable();
        Landmark land = new Landmark();
        result = land.load(parameters);
        result = HashHelper.joinHashes(result,
this.load(parameters));
        if (parameters["View"] == null)
        {
          result.Add("%View%", "StudentAlpha.aspx");
        }
        else
        {
          result.Add("%View%", parameters["View"].ToString());
        }
       return result;
      }

     public IList<Student> getAllElementsOfContext(Hashtable
parameters)
      {
        creationParameters = (Hashtable) parameters.Clone();
```

```
             this.load(parameters);
            return content;
       }

       public string getLinkForElement(int relativePosition)
       {
         int pos = content.IndexOf(current);
         int linkposition = int.MinValue;
         if (pos > -1)
         {
             linkposition = pos + relativePosition;
         }
         if ( (linkposition >= 0 ) && ( linkposition <
content.Count) ){
             return getLinkForElement(content[linkposition],
creationParameters);
         }
         else
         return null;
       }

     public string getLinkForElement(object currentObject,
Hashtable parameters)
     {
      Student currentElement = currentObject as Student;
      return this.getLinkForElement(currentElement,  parameters);
     }

     public string getLinkForElement(Student currentElement,
Hashtable parameters)
     {
        creationParameters = (Hashtable) parameters.Clone();
        string result = "?";
        result += "Controller=" + this.GetType().FullName ;
        result += "&Student=" + currentElement.globalIdentifier;
        return result;
     }
}
 #endregion
```

Quadro 7 – Exemplo do código da DSL-SHDM produzido do Contexto StudentAlpha.

5.4.Diagrama de Interface Abstrata

Um diagrama de interface abstrata referente ao contexto ProfessorAlpha é apresentado na Figura 42. Neste diagrama podemos verificar que existe um elemento no qual todos os CompositeInterfaceElements estão inseridos. Este elemento equivale ao corpo da página. Dentro deste elemento estão os índices que são landmarks: ProfessorsAlpha, ResearchAreasAlpha e StudentsAlpha, e o contexto navegacional ProfessorAlpha com os elementos abstratos responsáveis por estabelecer uma relação com as instâncias das classes a serem apresentadas.

Esta relação é mostrada na Figura 43 onde um CompositeInterfaceElement é associado através da propriedade SourceValue com a Estrutura de Acesso ProfessorsAlpha. A propriedade Repeatable com o valor *true,* representa que o item é um conjunto de dados que deve ser percorrido e a propriedade DataBound identifica que a origem dos dados provém de um objeto do modelo.

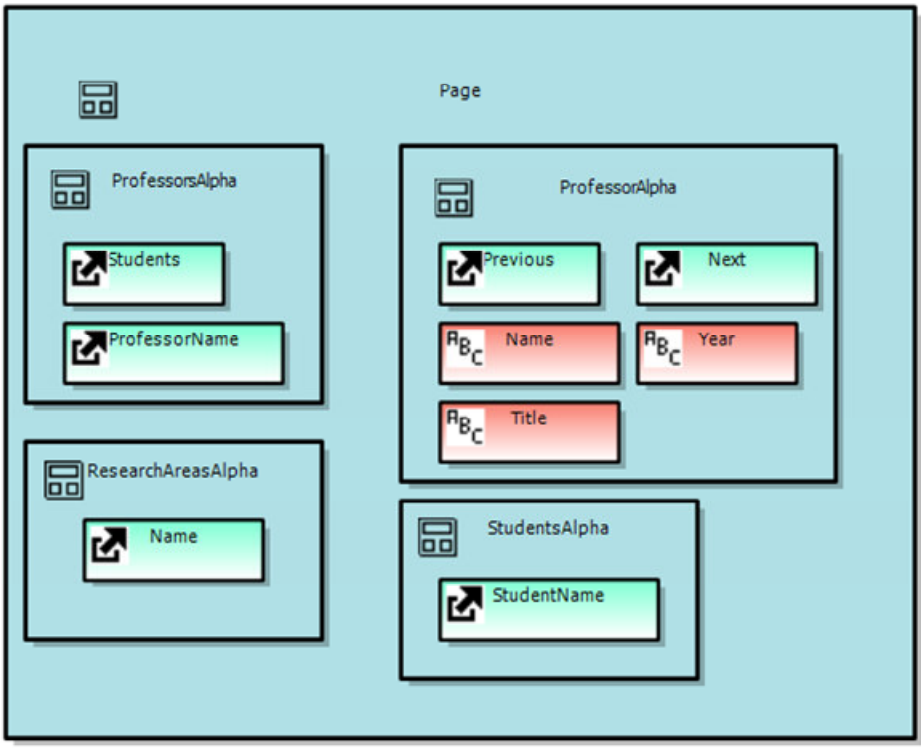

Figura 42 – Diagrama de interface abstrata do contexto ProfessorAlpha

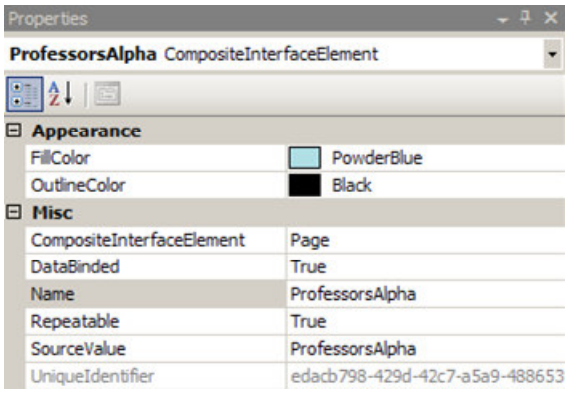

Figura 43 – Exemplo do mapeamento do CompositeInterfaceElement ProfessorsAlpha com a estrutura de acesso ProfessorsAlpha.

5.5. Diagrama de Interface Concreta

O diagrama de interface concreta produzido automaticamente pelo *framework* é similar ao diagrama apresentado na Figura 42, a diferença encontra-se apenas nas propriedades dos elementos. Foram adicionadas propriedades que, quando preenchidas, definem o mapeamento do elemento abstrato com algum elemento tecnológico, definido pela ontologia de interface concreta.

Na Figura 44 são apresentadas as propriedades preenchidas de um diagrama de interface concreta. Na propriedade ConcreteWidgetOntology, foi selecionada a ontologia de mapeamento da interface concreta. Baseado neste mapeamento que são apresentados para a seleção do desenvolvedor da interface os possíveis *widgets* concretos com os quais cada elemento abstrato pode ser mapeado. Na Figura 45 o ConcreteWidget selecionado foi Repeater da ontologia ASPX ontology, este elemento é responsável por iterar sobre o conjunto de instâncias de objetos que foram associadas na propriedade SourceValue. Esta propriedade está desabilitada na Figura 45 pois, deve ser definida apenas no diagrama de interface abstrata. A interface concreta produzida pelo diagrama de interface concreta após o mapeamento dos elementos do diagrama com a ontologia de interface concreta é apresentado no Quadro 8.

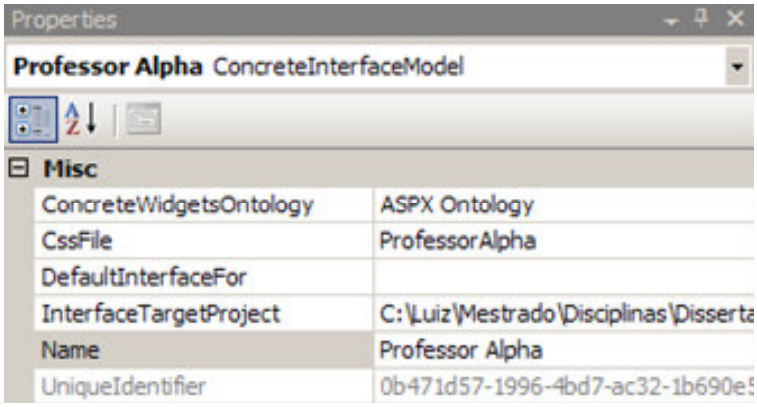

Figura 44 – Propriedades do diagrama de interface concreta.

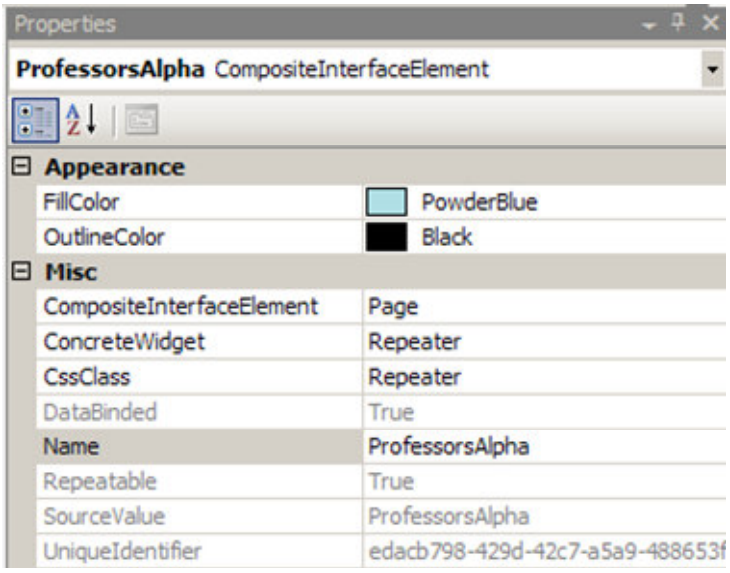

Figura 45 - Propriedades do CompositeInterfaceElement ProfessorsAlpha

```
<%@ Page Language="VB" Strict="false"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><script
runat="server">

Dim DataStore As Hashtable

Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs)

DataStore = CType(Context.Items.Item("UserSession"), Hashtable)

Me.DataBind()

End Sub

</script>

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

    <title>Page</title>
```

```
    <link rel="stylesheet" href="ProfessorAlpha.css"
type="text/css">
</head>
<body class="">
    <form id="form1" runat="server">
    <div>
    <div class=''><asp:Repeater id='ResearchAreas' runat='server'
DataSource='<%# DataStore.Item("ResearchAreasAlpha")
%>'><ItemTemplate>
<asp:HyperLink id="Name" CssClass="" runat="server"
NavigateUrl='<%#Eval("Name_lnk") %>'><%#Eval("Name")
%></asp:HyperLink>
</ItemTemplate></asp:Repeater></div>
<div class=''><asp:Repeater id='StudentsAlpha' runat='server'
DataSource='<%# DataStore.Item("StudentsAlpha") %>'><ItemTemplate>
<asp:HyperLink id="StudentName" CssClass="" runat="server"
NavigateUrl='<%#Eval("Name_lnk") %>'><%#Eval("Name")
%></asp:HyperLink></ItemTemplate></asp:Repeater></div>
<div class=''><asp:Repeater id='ProfessorsAlpha' runat='server'
DataSource='<%# DataStore.Item("ProfessorsAlpha")
%>'><ItemTemplate>
<asp:HyperLink id="ProfessorName" CssClass="" runat="server"
NavigateUrl='<%#Eval("Name_lnk") %>'><%#Eval("Name")
%></asp:HyperLink>
<asp:HyperLink id="Students" CssClass="" runat="server"
NavigateUrl='<%#Eval("Name_lnk") %>'><%#Eval("Name")
%></asp:HyperLink></ItemTemplate></asp:Repeater></div>
<div class='' id='ProfessorAlpha'>
<div class=''><%# DataStore.Item("ProfessorAlpha").current.title
%></div>
<div class=''><%# DataStore.Item("ProfessorAlpha").current.name
%></div>
<div class=''><%# DataStore.Item("ProfessorAlpha").current.year
%></div>
<asp:HyperLink id="Next" CssClass="" runat="server"
NavigateUrl='<%#
DataStore.Item("ProfessorAlpha").getLinkForElement(+1)
%>'>Next</asp:HyperLink>
<asp:HyperLink id="Previous" CssClass="" runat="server"
NavigateUrl='<%#
DataStore.Item("ProfessorAlpha").getLinkForElement(-1)
%>'>Previous</asp:HyperLink>
</div>
</div>
</form>
</body>
</html>
```

Quadro 8 – Código da interface gerada pelo diagrama de interface abstrata ProfessorAlpha

5.6.Exemplo de uma Instância da Ontologia de Interface Concreta

No Quadro 9 é apresentado um exemplo de uma instância da ontologia de interface concreta com os elementos concretos responsáveis pela renderização da interface utilizando ASP.Net como tecnologia. No exemplo abaixo é possível observar algumas palavras com delimitadores **<#$ $#>**. Estas palavras são substituídas, em tempo de geração da interface concreta, por valores definidos no diagrama de interface concreta.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
       <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-
ns#'>
       <!ENTITY awo 'http://protege.stanford.edu/rdf/awo#'>
      <!ENTITY cwo 'http://protege.stanford.edu/rdf/cwo#'>
 <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
       xmlns:awo="&awo;"
       xmlns:cwo="&cwo;"  xmlns:rdfs="&rdfs;">
<rdf:Description rdf:about="&cwo;cwo_Instance_0"
       cwo:fileExtension="aspx"
       cwo:name="ASPX Ontology"
       rdfs:label="ASPX Ontology">
      <cwo:dataSourceFunction xml:space='preserve'><![CDATA[<%#
DataStore.Item("<#$OBJECT$#>")<#$VALUEPATH$#>
%>]]></cwo:dataSourceFunction>
      <cwo:evaluateFunction
xml:space='preserve'><![CDATA[<%#Eval("<#$OBJECT$#>")<#$VALUEPATH
$#> %>]]></cwo:evaluateFunction>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_1"/>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_11"/>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_12"/>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_13"/>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_14"/>
      <cwo:concrete widgets rdf:resource="&cwo;cwo_Instance_15"/>
```

```
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_2"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_3"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_4"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_5"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_6"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_7"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_8"/>
      <cwo:concrete*Widgets* rdf:resource="&cwo;cwo_Instance_9"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteInterface"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_1"
       cwo:name="Page"
       rdfs:label="Page">
      <cwo:code xml:space='preserve'><![CDATA[<%@ Page
Language="VB" Strict="false"%><!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">Dim DataStore As HashtableProtected Sub
Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs)DataStore =
CType(Context.Items.Item("UserSession"),
Hashtable)Me.DataBind()End Sub<#$PageCode$#></script><html
xmlns="http://www.w3.org/1999/xhtml" ><head runat="server">
<title><#$ELEMENTNAME$#></title></head><body
class="<#$ELEMENTCLASSNAME$#>"> <form id="form1" runat="server">
<div> <#$NESTEDELEMENTS$#> </div>
</form></body></html>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:CompositeInterfaceElement"/>
      <rdf:type rdf:resource="&awo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_11"
       cwo:name="DropDownList"
       rdfs:label="DropDownList">
      <cwo:code xml:space='preserve'><![CDATA[<asp:DropDownList
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
DataSource="<#$ELEMENTDATASOURCE$#>"
```

```
DataTextField="<#$CAPTION$#>" DataValueField="<#$VALUE$#>"
></asp:DropDownList>]]></cwo:code>

      <cwo:abstractElements
rdf:resource="&awo;awo:SingleChoice"/>

      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>

</rdf:Description>

<rdf:Description rdf:about="&cwo;cwo_Instance_12"

       cwo:name="RadioButton"

       rdfs:label="RadioButton">

      <cwo:code xml:space='preserve'><![CDATA[<asp:RadioButton
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
Text="<#$CAPTION$#>"></asp:RadioButton>]]></cwo:code>

      <cwo:abstractElements
rdf:resource="&awo;awo:SingleChoice"/>

      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>

</rdf:Description>

<rdf:Description rdf:about="&cwo;cwo_Instance_13"

       cwo:name="CheckBoxList"

       rdfs:label="CheckBoxList">

      <cwo:code xml:space='preserve'><![CDATA[<asp:CheckBoxList
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
Text="<#$CAPTION$#>"></asp:CheckBoxList>]]></cwo:code>

      <cwo:abstractElements
rdf:resource="&awo;awo:MultipleChoices"/>

      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>

</rdf:Description>

<rdf:Description rdf:about="&cwo;cwo_Instance_14"

       cwo:name="RadioButtonList"

       rdfs:label="RadioButtonList">

      <cwo:code
xml:space='preserve'><![CDATA[<asp:radiobuttonlist
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
Text="<#$CAPTION$#>"></asp:radiobuttonlist>]]></cwo:code>

      <cwo:abstractElements
rdf:resource="&awo;awo:SingleChoice"/>
```

```
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_15"
       cwo:name="ListBox"
       rdfs:label="ListBox">
      <cwo:code xml:space='preserve'><![CDATA[<asp:listbox
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
DataSource="<#$ELEMENTDATASOURCE$#>"
DataTextField="<#$CAPTION$#>" DataValueField="<#$VALUE$#>"
></asp:listbox>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:MultipleChoices"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_2"
       cwo:name="HyperLink"
       rdfs:label="HyperLink">
      <cwo:code xml:space='preserve'><![CDATA[<asp:HyperLink
id="<#$ELEMENTNAME$#>" CssClass="<#$ELEMENTCLASSNAME$#>"
runat="server"
NavigateUrl='<#$TARGET$#>'><#$CAPTION$#></asp:HyperLink>]]></cwo:
code>
      <cwo:abstractElements
rdf:resource="&awo;awo:SimpleActivator"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_3"
       cwo:name="Label"
       rdfs:label="Label">
      <cwo:code xml:space='preserve'><![CDATA[<div
class='<#$ELEMENTCLASSNAME$#>'><#$CAPTION$#></div>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:ElementExihibitor"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_4"
```

```
       cwo:name="Repeater"
       rdfs:label="Repeater">
      <cwo:code xml:space='preserve'><![CDATA[<div
class='<#$ELEMENTCLASSNAME$#>'><asp:Repeater
id='<#$ELEMENTNAME$#>' runat='server'
DataSource='<#$ELEMENTDATASOURCE$#>'><ItemTemplate><#$NESTEDELEME
NTS$#></ItemTemplate></asp:Repeater></div>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:CompositeInterfaceElement"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_5"
       cwo:name="Panel"
       rdfs:label="Panel">
      <cwo:code xml:space='preserve'><![CDATA[<div
class='<#$ELEMENTCLASSNAME$#>'
id='<#$ELEMENTNAME$#>'><#$NESTEDELEMENTS$#></div>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:CompositeInterfaceElement"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_6"
       cwo:name="TextBox"
       rdfs:label="TextBox">
      <cwo:code xml:space='preserve'><![CDATA[<asp:TextBox
ID="<#$ELEMENTNAME$#>" runat="server" Text="<#$CAPTION$#>"
CssClass="<#$ELEMENTCLASSNAME$#>" ></asp:TextBox>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:FreeValueInput"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_7"
       cwo:name="TextArea"
       rdfs:label="TextArea">
      <cwo:code xml:space='preserve'><![CDATA[<asp:TextBox
ID="<#$ELEMENTNAME$#>" runat="server" Text="<#$CAPTION$#>"
```

```
TextMode="MultiLine" CssClass="<#$ELEMENTCLASSNAME$#>"
></asp:TextBox>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:FreeValueInput"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_8"
       cwo:name="PasswordBox"
       rdfs:label="PasswordBox">
      <cwo:code xml:space='preserve'><![CDATA[<asp:TextBox
ID="<#$ELEMENTNAME$#>" runat="server" Text="<#$CAPTION$#>"
TextMode="Password" CssClass="<#$ELEMENTCLASSNAME$#>"
></asp:TextBox>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:FreeValueInput"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
<rdf:Description rdf:about="&cwo;cwo_Instance_9"
       cwo:name="CheckBox"
       rdfs:label="CheckBox">
      <cwo:code xml:space='preserve'><![CDATA[<asp:CheckBox
ID="<#$ELEMENTNAME$#>" runat="server"
CssClass="<#$ELEMENTCLASSNAME$#>"
Text="<#$CAPTION$#>"></asp:CheckBox>]]></cwo:code>
      <cwo:abstractElements
rdf:resource="&awo;awo:MultipleChoices"/>
      <rdf:type rdf:resource="&cwo;cwo:ConcreteWidget"/>
</rdf:Description>
</rdf:RDF>
```

Quadro 9 – Exemplo de uma instância de ontologia de interface concreta

5.7.Execução da Aplicação

A aplicação quando executada carrega os dados que estão no banco de dados semântico e os apresenta através da interface produzida. Um exemplo da interface produzida pelo diagrama de interface concreta do contexto

ProfessorAlpha é apresentado na Figura 46, sem a customização de um CSS ser aplicada. Na Figura 47 a mesma interface é apresentada com a formatação aplicada por um CSS.

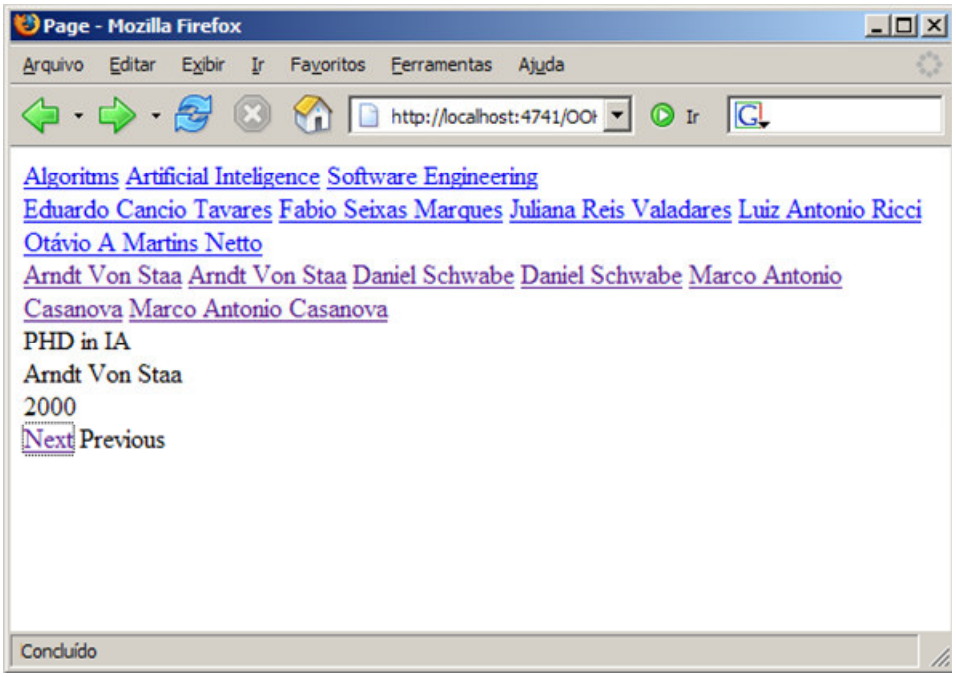

Figura 46 - Interface produzida pelo diagrama de interface concreta sem o CSS

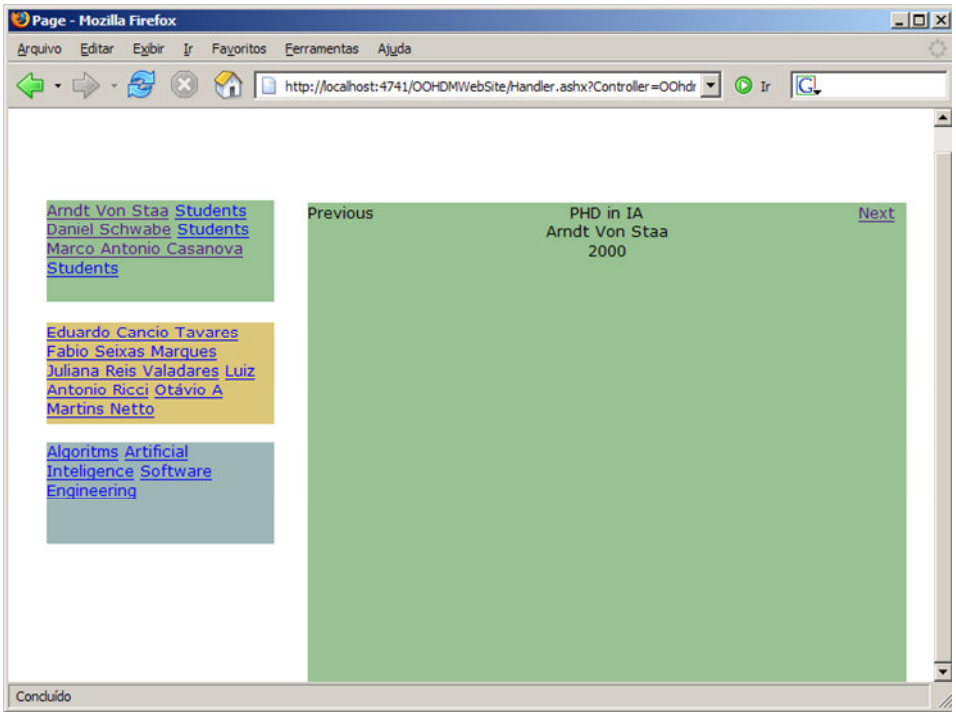

Figura 47 - Interface produzida pelo diagrama de interface concreta com o CSS