

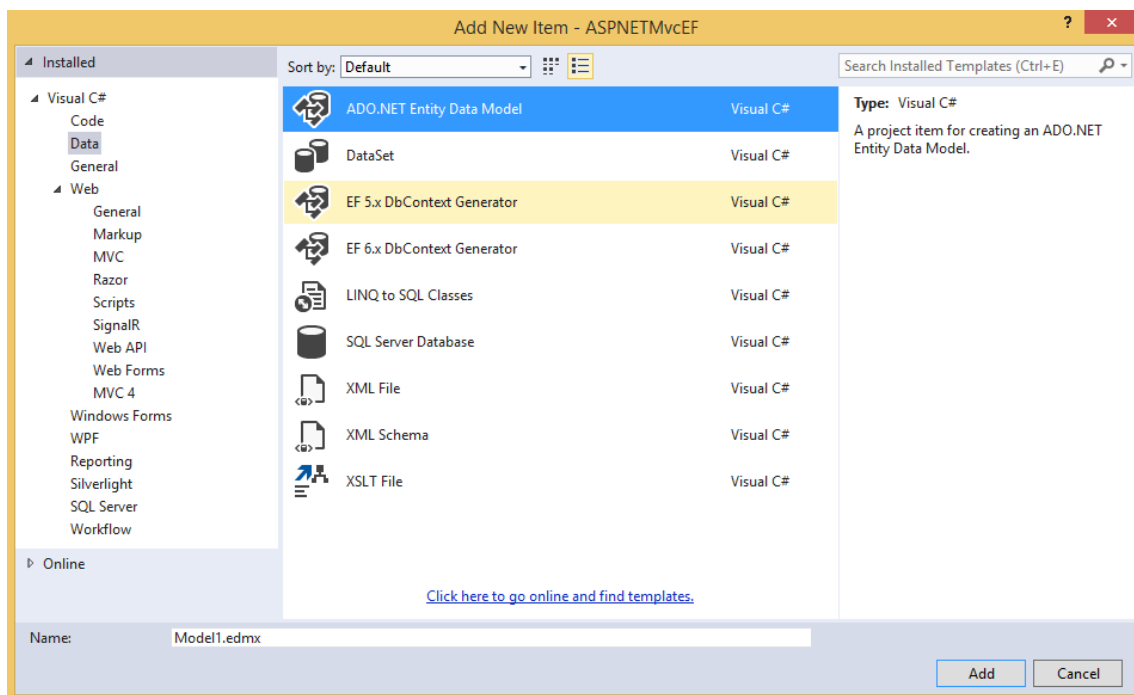
PROJETOS EXEMPLO DE ASP.NET MVC

Projeto 1:

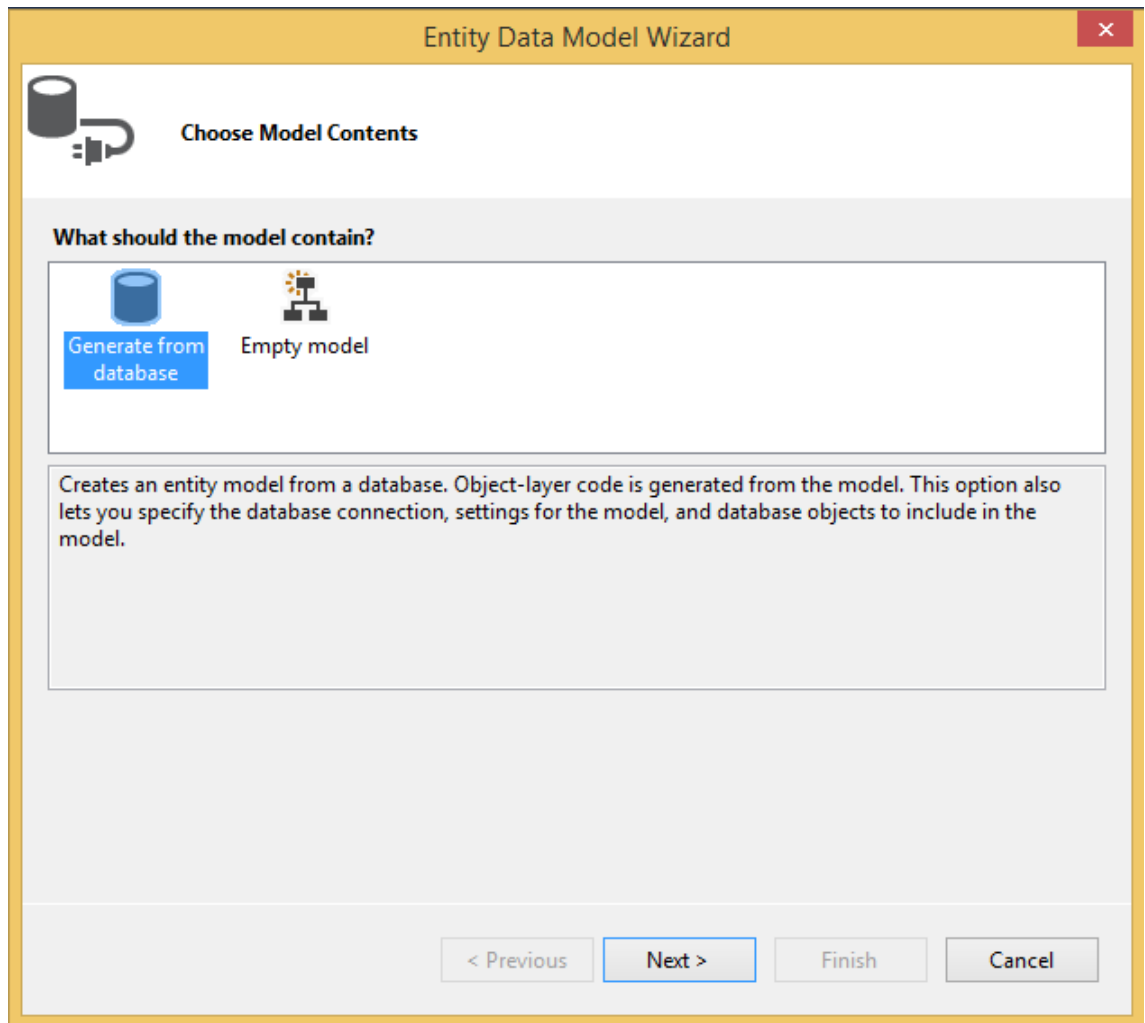
Projeto 2: Implementando o CRUD em ASP.NET MVC, utilizando o ADO.NET Entity Framework.

Passos:

1. Crie um novo projeto Web, ASP.NET MVC4, no Visual Studio.
2. Clique com o botão direito do mouse na pasta Models → adicione um novo item → selecione a pasta Data → Escolha o template → ADO.NET Framework. Nomeie com sua preferência. Veja a figura 1.



3. Clique em ADD.
4. Na janela que surge → selecione o modelo Generate from database. Veja a figura 2.



5. Clique em Next.
6. Na janela que surge → faça uma nova conexão e selecione o banco de dados Northwind. Veja a figura 3.

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
localhost Refresh

Log on to the server

Use Windows Authentication
 Use SQL Server Authentication

User name:
Password:
 Save my password

Connect to a database

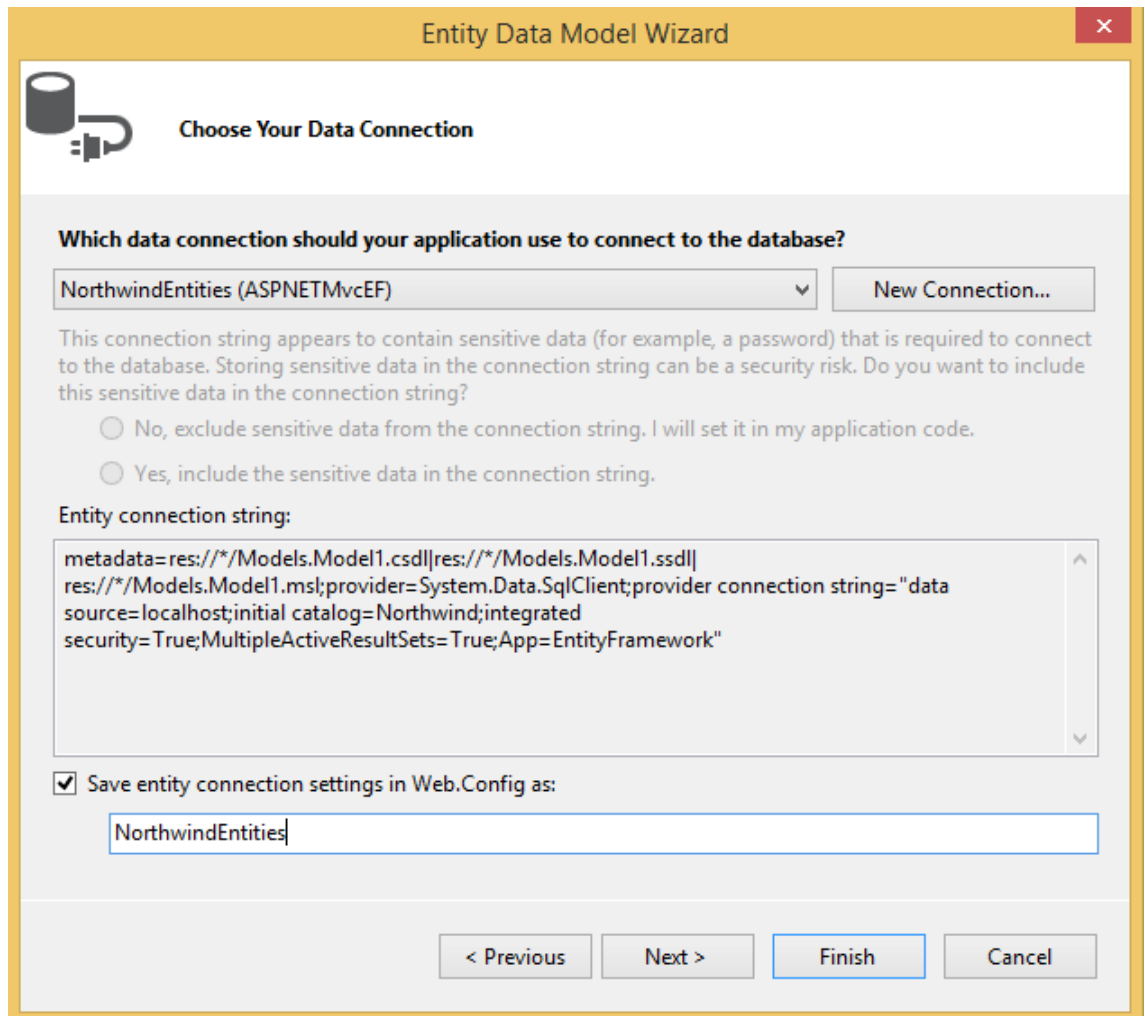
Select or enter a database name:
Northwind

Attach a database file:
 Browse...
Logical name:

Advanced...

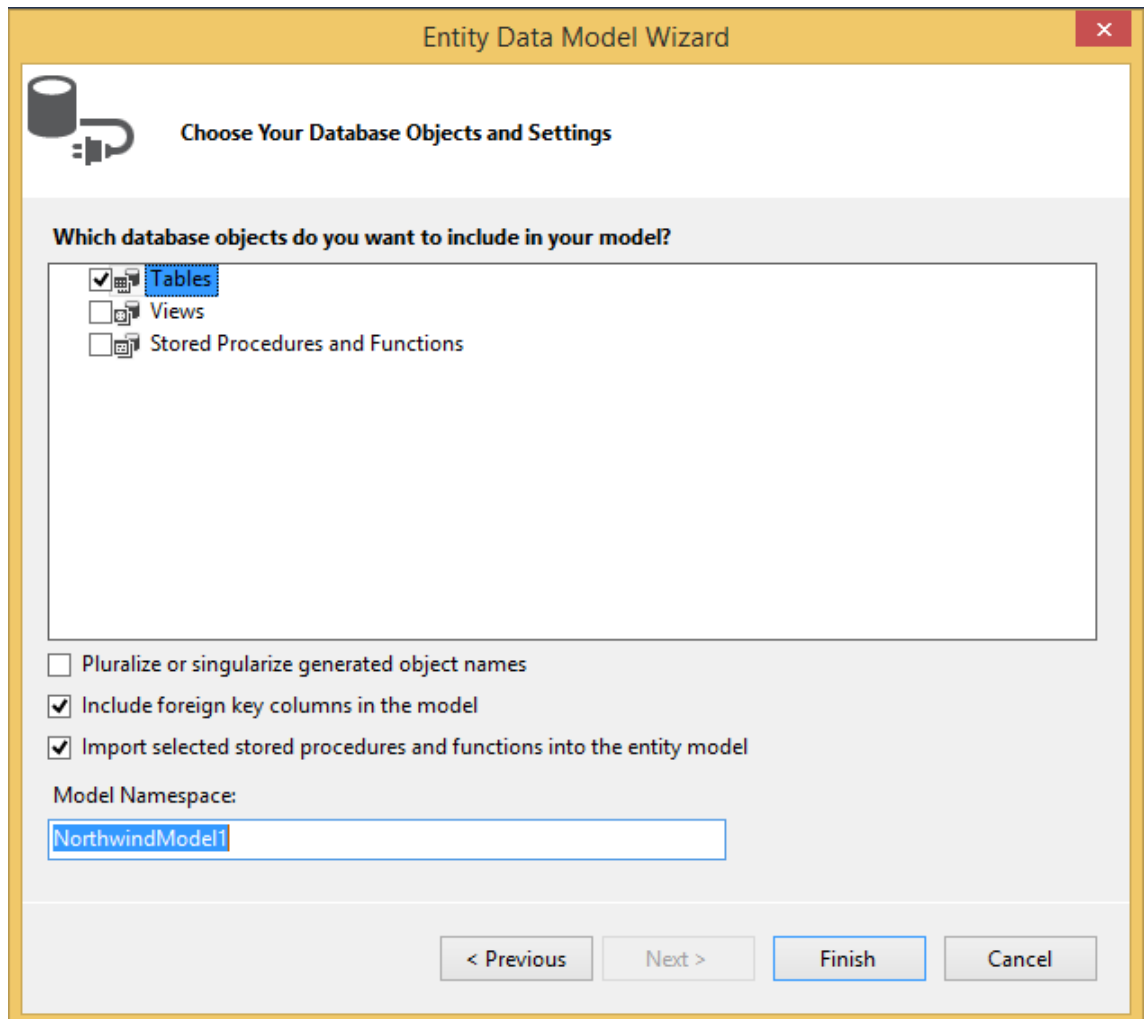
Test Connection OK Cancel

7. Clique em OK.
8. Na janela seguinte, nomeie o modelo como NorthwinEntitie. Veja a figura 4

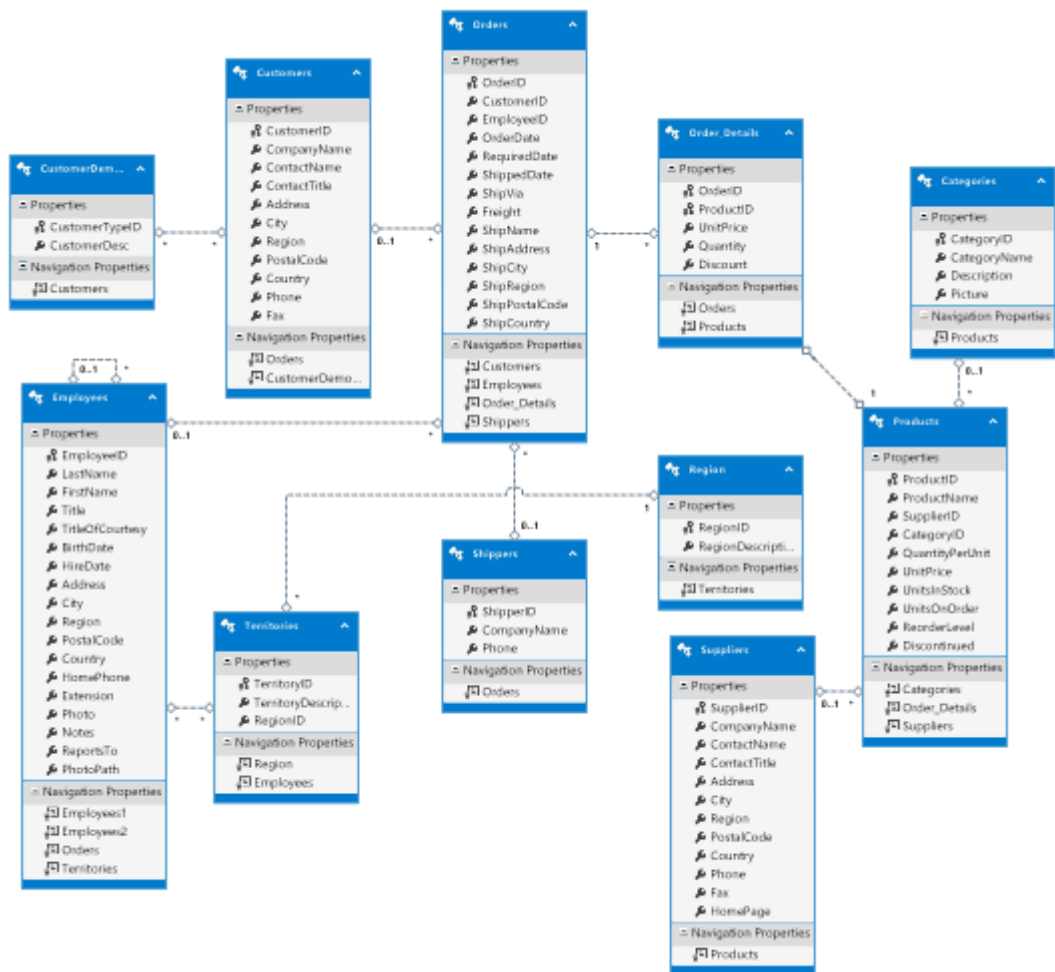


9. Clique em Next.

10. Na janela seguinte selecione a caixa table. Veja a figura 5.



11. Clique Finish. A figura 6 mostra o diagrama de classes que mapeia o banco de dados Northwind.



12. Feito o modelo de dados, o próximo passo é desenvolver o controle para acessar e manipular os dados. A listagem 1 mostra o código para manipular o crud na tabela categories.

Listagem 1 Código do controle CategoryController

```
public class CategoriaController : Controller
{
    //
    // GET: /Categoria/
    private NorthwindEntities bd = new NorthwindEntities();

    public ActionResult Index()
    {
        try
        {
            var model = bd.Categories.AsParallel().ToList();
            if (model == null)
                return View("NotFound");
            else
                return View(model);
        }
    }
}
```

```

    catch { }
    finally {
        if (bd != null) bd.Dispose();
    }

    return View();
}

//Método que retorna os nomes das categorias.
public ActionResult List()
{
    try
    {
        var model = bd.Categories.AsParallel().ToList();
        if (model == null)
            return View("NotFound");
        else
            return View(model);
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}

// GET: /Categorie/Details/5

//Método que retorna todos os produtos da categoria selecionada (passada).
public ActionResult Details (string categoria)
{
    try
    {
        ViewBag.Message = categoria;
        var model = (from p in bd.Products
                     where p.Categories.CategoryName == categoria
                     select p).ToList();
        if (model == null)
            return View("NotFound");
        else
            return View(model);
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}

//
// GET: /Categorie/Create

public ActionResult Create()
{
    return View();
}

//
// POST: /Categorie/Create

```

```

[HttpPost]
public ActionResult Create([Bind(Exclude= "CategoryID")]Categories p)
{
    try
    {
        if (!ModelState.IsValid) return View();
        bd.Categories.Add(p);
        bd.SaveChanges();

        return RedirectToAction("Index");
    }
    catch (System.Data.EntitySqlException)
    {
        ModelState.AddModelError("", "Erro EntitySqlException.");
        return View();
    }
    catch (System.Data.EntityCommandExecutionException)
    {
        ModelState.AddModelError("", "Erro EntityCommandExecutionException.");
        return View();
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}

//
// GET: /Categorie/Edit/5

public ActionResult Edit(int? id)
{
    try
    {
        if (!id.HasValue) return View("NotFound");

        var model = bd.Categories.Where(c => c.CategoryID == id).FirstOrDefault();

        if (model == null)
            return View("NotFound");
        else
            return View(model);
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}

//
// POST: /Categorie/Edit/5

[HttpPost]
public ActionResult Edit(int? id, FormCollection collection)
{
    try
    {

```



```

        if (!id.HasValue) return View("NotFound");

        var model = bd.Categories.Where(c => c.CategoryID == id).FirstOrDefault();

        if (model == null)
        {
            return View("NotFound");
        }
        else
        {
            this.TryUpdateModel(model);
            bd.SaveChanges();
        }

        return RedirectToAction("Index");
    }
    catch (System.Data.EntitySqlException)
    {
        ModelState.AddModelError("", "Erro EntitySqlException.");
        return View();
    }
    catch (System.Data.EntityCommandExecutionException)
    {
        ModelState.AddModelError("", "Erro EntityCommandExecutionException.");
        return View();
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}

// POST: /Categorie/Delete/5

public ActionResult Delete(int? id)
{
    try
    {
        if (!id.HasValue) return View("NotFound");

        var model = bd.Categories.Where(c => c.CategoryID == id).FirstOrDefault();
        if (model == null)
        {
            return View("NotFound");
        }
        else
        {
            return View(model);
        }
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}
}

```

```

[HttpPost]
public ActionResult Delete(int? id, FormCollection collection)
{
    try
    {
        if (!id.HasValue) return View("NotFound");

        var model = bd.Categories.Where(c => c.CategoryID == id).FirstOrDefault();
        if (model == null)
        {
            return View("NotFound");
        }
        else
        {
            bd.Categories.Remove(model);
            bd.SaveChanges();
        }

        return RedirectToAction("Index");
    }
    catch (System.Data.EntitySqlException)
    {
        ModelState.AddModelError("", "Erro EntitySqlException.");
        return View();
    }
    catch (System.Data.EntityCommandCompilationException)
    {
        ModelState.AddModelError("", "Erro EntityCommandExecutionException.");
        return View();
    }
    finally
    {
        if (bd != null) bd.Dispose();
    }
}
}

```

13. Feito o controle é hora de criarmos as janelas.

Janela listar todas as categorias (Index.cshtml).

```
@model IEnumerable<ASPNETMvcEF.Models.Categories>
```

```
@{
    ViewBag.Title = "Index";
}
```

```
<h2>Index</h2>
```

```
<p>
    @Html.ActionLink("Nova categoria", "Create")
</p>
<table>
```

```

<tr>
  <th>
    @Html.DisplayNameFor(model => model.CategoryID)
  </th>
  <th>
    @Html.DisplayNameFor(model => model.CategoryName)
  </th>
  <th>
    @Html.DisplayNameFor(model => model.Description)
  </th>
</th></th>
</tr>

@foreach (var item in Model) {
  <tr>
    <td>
      @Html.DisplayFor(modelItem => item.CategoryID)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.CategoryName)
    </td>
    <td>
      @Html.DisplayFor(modelItem => item.Description)
    </td>
    <td>
      @Html.ActionLink("Edit", "Edit", new { id=item.CategoryID }) |
      @Html.ActionLink("Details", "List", new { id=item.CategoryName }) |
      @Html.ActionLink("Delete", "Delete", new { id=item.CategoryID })
    </td>
  </tr>
}

</table>

```

Janela criar nova categoria (create.cshtml)

```

@model ASPNETMvcEF.Models.Categories

@{
  ViewBag.Title = "Inserindo novos registros";
}

<h2>Inserir novas categorias</h2>

@using (Html.BeginForm()) {
  @Html.ValidationSummary(true)

  <fieldset>
    <legend>Categories</legend>

    <div class="editor-label">
      @Html.LabelFor(model => model.CategoryID)
    </div>
    <div class="editor-field">
      @Html.EditorFor(model => model.CategoryID)
    </div>
  </fieldset>
}

```

```

        @Html.ValidationMessageFor(model => model.CategoryID)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.CategoryName)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.CategoryName)
        @Html.ValidationMessageFor(model => model.CategoryName)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.Description)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.Description)
        @Html.ValidationMessageFor(model => model.Description)
    </div>

    <p>
        <input type="submit" value="Salvar" />
    </p>
</fieldset>
}

<div>
    @Html.ActionLink("Voltar para a página anterior", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Janela Editar categoria (Edtit.cshtml)

```

@model ASPNETMvcEF.Models.Categories

@{
    ViewBag.Title = "Editar";
}

<h2>Editar</h2>

@using (Html.BeginForm()) {
    @Html.ValidationSummary(true)

    <fieldset>
        <legend>Categorias</legend>

        <div class="editor-label">
            @Html.LabelFor(model => model.CategoryID)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.CategoryID)
            @Html.ValidationMessageFor(model => model.CategoryID)

```

```

</div>

<div class="editor-label">
    @Html.LabelFor(model => model.CategoryName)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.CategoryName)
    @Html.ValidationMessageFor(model => model.CategoryName)
</div>

<div class="editor-label">
    @Html.LabelFor(model => model.Description)
</div>
<div class="editor-field">
    @Html.EditorFor(model => model.Description)
    @Html.ValidationMessageFor(model => model.Description)
</div>

<p>
    <input type="submit" value="Salvar" />
</p>
</fieldset>
}

<div>
    @Html.ActionLink("Voltar para a página anterior", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Janela Detalhes da categoria (Details.cshtml)

```

<%@ Page Language="C#"
Inherits="System.Web.Mvc.ViewPage<IEnumerable<ASPNETMvcEF.Models.Products>>" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <meta name="viewport" content="width=device-width" />
    <title>Details</title>
</head>
<body>
    <div>
        <h2><%: ViewBag.Message %></h2>
        <ul>
            <% foreach (var item in Model) { %>
                <li>
                    <%: item.ProductName %>
                </li>
            <% } %>
        </ul>
    </div>
    <p><%: Html.ActionLink("Voltar para a página anterior", "List") %></p>

```

```
</body>
</html>
```

Janela listar categorias (List.cshtml)

```
<%@ Page Language="C#"
Inherits="System.Web.Mvc.ViewPage<IEnumerable<ASPNETMvcEF.Models.Categories>>" %>

<!DOCTYPE html>

<html>
<head runat="server">
  <meta name="viewport" content="width=device-width" />
  <title>List</title>
</head>
<body>
  <p>Lista de categorias</p>

  <% foreach (var item in Model) { %>
    <%: Html.ActionLink(item.CategoryName, "Details",
      new { categoria = item.CategoryName }) %><br />
  <% } %>
</body>
</html>
```