

Aplicação MVC + Entity + MySQL

Criar a Classe Cidade

Adicionar a classe Cidade na pasta Models

```
public class Cidade
{
    [Key]
    public int CidadeID { get; set; }

    [Required(ErrorMessage = "Preencha o nome da cidade")]
    [DisplayName("Nome da cidade")]
    [StringLength(100, MinimumLength = 3, ErrorMessage = "O nome da cidade deve ter
entre 3 e 100 caracteres.")]
    public string Nome { get; set; }
}
```

Adicionar o Relacionamento entre Cidade e Estado que é de 1:N

Na Classe Cidade, adicionar

```
public class Cidade
{
    [Key]
    public int CidadeID { get; set; }

    [Required(ErrorMessage = "Preencha o nome da cidade")]
    [DisplayName("Nome da cidade")]
    [StringLength(100, MinimumLength = 3, ErrorMessage = "O nome da cidade deve ter
entre 3 e 100 caracteres.")]
    public string Nome { get; set; }

    public int EstadoID { get; set; }
    public virtual Estado Estado { get; set; }
}
```

Na Classe Estado, adicionar

```
public class Estado
{
    [Key]
    public int EstadoID { get; set; }

    [Required(ErrorMessage = "Preencha a descrição do estado")]
    [DisplayName("Descrição")]
    [StringLength(50, MinimumLength = 3, ErrorMessage = "A descrição do estado deve ter
entre 3 e 50 caracteres.")]
    public string Descricao { get; set; }

    [Required(ErrorMessage = "Preencha a sigla do estado")]
    [StringLength(2, MinimumLength = 2, ErrorMessage = "A sigla do estado deve ter 2
caracteres.")]
    public string Sigla { get; set; }

    public virtual ICollection<Cidade> Cidades { get; set; }
}
```

Adicionar Cidade ao Context

```
public DbSet<Estado> Estados { get; set; }  
public DbSet<Cidade> Cidades { get; set; }
```

Adicionar Migration

Add-Migration cidade

Caso dê o erro `The provider did not return a ProviderManifestToken string.` Verifique se o serviço do MySQL está Iniciado.

Atualizar o Banco de Dados

Update-Database

Verifique no banco de dados se as tabelas Estado e Cidade estão corretas.

Gerar as Telas de Cidade

Comentar os blocos system.data e entityFramework no web.config.

Adicionar um Novo Controller na pasta Controllers.

- MVC 5 Controller with views, using EntityFramework
 - Model Class: Cidade
 - Data Context class: Context
 - Controller name: CidadesController
 - Deixar marcado: Generate views; Reference script libraries; Use a layout page.

Descomentar os blocos system.data e entityFramework no web.config.

Altere o Layout principal

Na pasta Views/Shared/_Layout.cshtml

Adicione o item no menu

```
<li>@Html.ActionLink("Cidades", "Index", "Cidades")</li>
```

Execute a aplicação.

Acesse a lista de Cidades.

Acesse o Cadastro de Cidade (já deve ter gerado com a Lista de Estado).

Alterando a tela de Adicionar Cidade

Altere a div que mostra o DropDownList de Estado de

```
<div class="form-group">
@Html.LabelFor(model => model.EstadoID, "EstadoID", htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownList("EstadoID", null, htmlAttributes: new { @class = "form-control" })
        @Html.ValidationMessageFor(model => model.EstadoID, "", new { @class = "text-danger" })
    </div>
</div>
```

Para

```
<div class="form-group">
@*@Html.LabelFor(model => model.EstadoID, "EstadoID", htmlAttributes: new { @class = "control-label col-md-2" })*@
<label class="control-label col-md-2">
    Estado:
</label>

<div class="col-md-10">
@*@Html.DropDownList("EstadoID", null, htmlAttributes: new { @class = "form-control" })*@
@Html.DropDownList("EstadoID", null, "Selecione um estado", new { @class = "form-control" })
@Html.ValidationMessageFor(model => model.EstadoID, "", new { @class = "text-danger" })
</div>

</div>
```

Execute a aplicação.

Acesse o Cadastro de Cidade.

- Não preencha a Cidade e clique em Salvar – Verifique a mensagem
- Não preencha só o Estado e clique em Salvar – Verifique a mensagem
- Preencha os dois campos e Salve

Alterar a mensagem de Erro do Estado

Na Classe Cidade, acrescentar annotations em EstadoID

```
public class Cidade
{
    [Key]
    public int CidadeID { get; set; }

    [Required(ErrorMessage = "Preencha o nome da cidade")]
    [DisplayName("Nome da cidade")]
    [StringLength(100, MinimumLength = 3, ErrorMessage = "O nome da cidade deve ter entre 3 e 100 caracteres.")]
    public string Nome { get; set; }

    [Required(ErrorMessage = "Selecione um estado")]
    [DisplayName("Estado")]
    public int EstadoID { get; set; }

    public virtual Estado Estado { get; set; }
}
```

Adicionar Migration

Add-Migration cidade2

Atualizar o Banco de Dados

Update-Database

Execute a aplicação.

Acesse o Cadastro de Cidade.

- Não preencha a Cidade e clique em Salvar – Verifique a mensagem
- Não preencha só o Estado e clique em Salvar – Verifique a mensagem
- Preencha os dois campos e Salve

Alterando a Lista de Cidades

Pode-se observar que a Lista já exibe o Estado e a Cidade, porém, o Estado está como “Descrição”. Na tela Views/Cidades/Index.cshtml, altere a primeira coluna da tabela.

De

```
<table class="table">
  <tr>
    <th>
      @Html.DisplayNameFor(model => model.Estado.Descricao)
    </th>
    <th>
      @Html.DisplayNameFor(model => model.Nome)
    </th>
  </tr>
</table>
```

Para

```
<table class="table">
  <tr>
    <th>
      Estado
    </th>
    <th>
      @Html.DisplayNameFor(model => model.Nome)
    </th>
  </tr>
</table>
```

Execute a aplicação.

Acesse a Lista de Cidades.

Realizar as demais alterações do Layout das telas da Cidade (Index, Create e Edit).