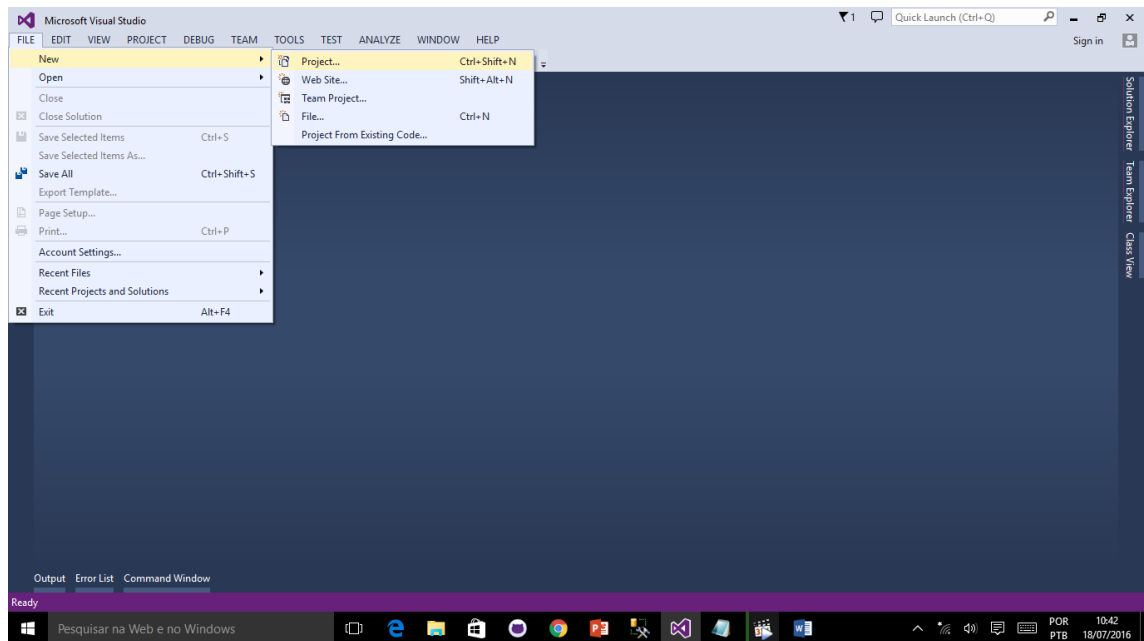
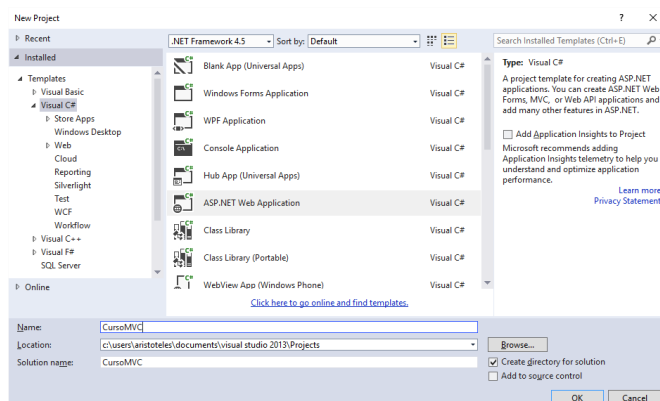


Prática 11

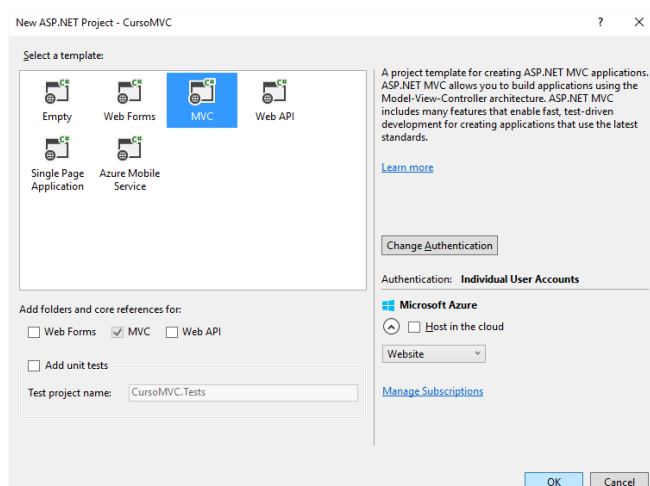
❖ Crie um novo item ao projeto



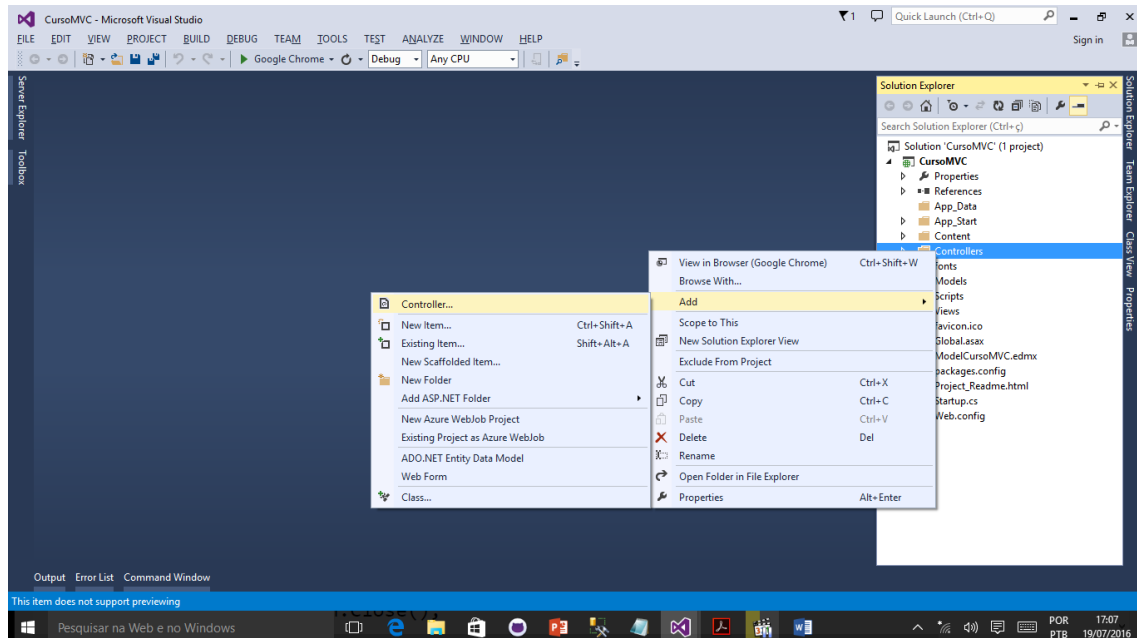
❖ Selecione ASP.NET Web Application e adicione o nome cursoMVC



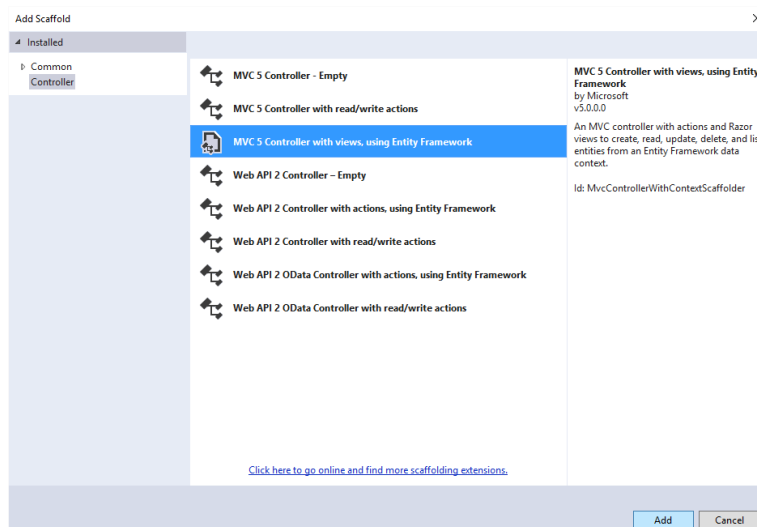
❖ Escolhe a opção MVC



- ❖ Examine o projeto
- ❖ Adicione um ADO.NET Entity Data Model do banco de dados cursomvc
- ❖ Adicione um controller ao projeto



- ❖ Selecione a opção MVC 5 Controller with views, using Entity Framework



- ❖ Informe o nome o Model Class, Data context class e Controller name

Add Controller

Model class:

Data context class: +

☐ Use async controller actions

Views:

☒ Generate views

☒ Reference script libraries

☒ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

Controller name:

Add Cancel

❖ Adicione a opção Aluno no menu

CourseMVC - Microsoft Visual Studio

FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP

Quick Launch (Ctrl+Q) Sign in

Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>@ViewBag.Title - My ASP.NET Application</title>
<@Styles.Render("~/Content/css")>
<@Scripts.Render("~/bundles/modernizr")>
</head>
<body>
<div class="navbar navbar-inverse navbar-fixed-top">
<div class="container">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<@Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })>
</div>
<div class="navbar-collapse collapse">
<ul class="nav navbar-nav">
<li>@Html.ActionLink("Home", "Index", "Home")</li>
<li>@Html.ActionLink("Aluno", "Index", "Aluno")</li>
<li>@Html.ActionLink("About", "About", "Home")</li>
<li>@Html.ActionLink("Contact", "Contact", "Home")</li>
</ul>
<@Html.Partial("_LoginPartial")>
</div>
</div>
<div class="container body-content">
```

Solution Explorer

Search Solution Explorer (Ctrl+G)

- Properties
- References
- App_Data
- App_Start
- Content
- Controllers
 - AccountController.cs
 - AlunoController.cs
 - HomeController.cs
 - ManageController.cs
- fonts
- Models
- Scripts
- Views
 - Account
 - Aluno
 - Create.cshtml
 - Delete.cshtml
 - Details.cshtml
 - Edit.cshtml
 - Index.cshtml
 - Lockout.cshtml
 - Manage
 - Shared

Item(s) Saved

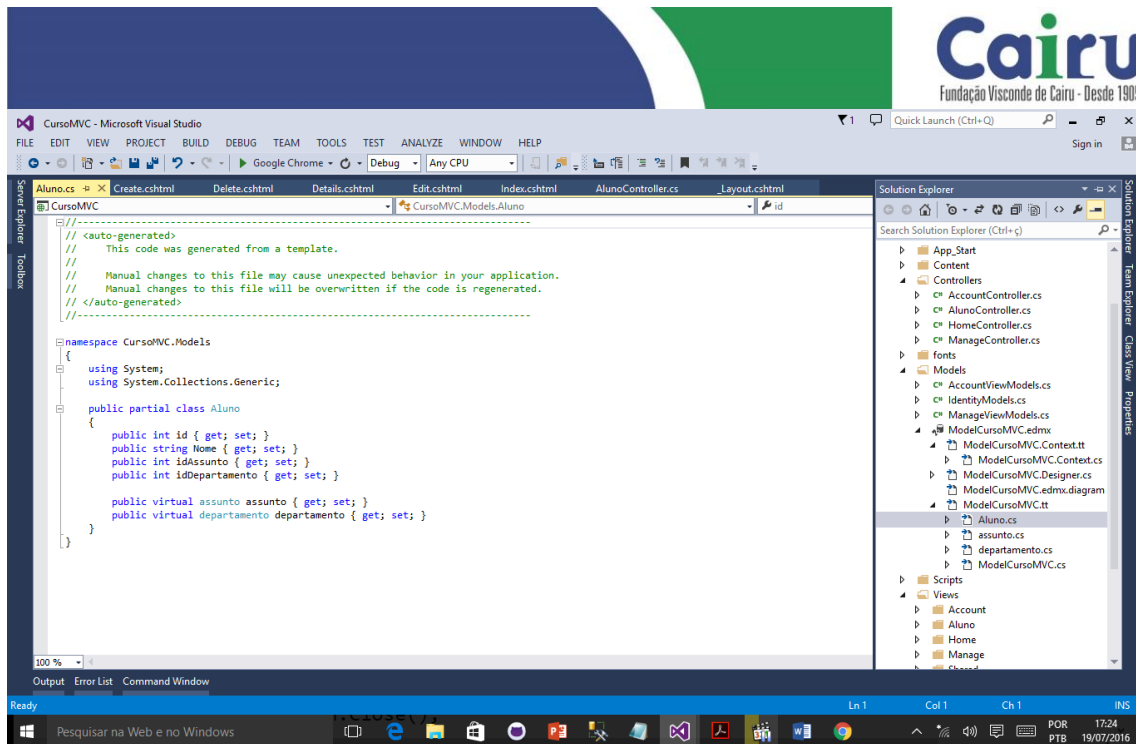
Ln 25 Col 21 Ch 21 INS

Pesquisar na Web e no Windows

17:15 19/07/2016

❖ Compile e teste

❖ Abra o Arquivo Alunos que se encontra na pasta Model



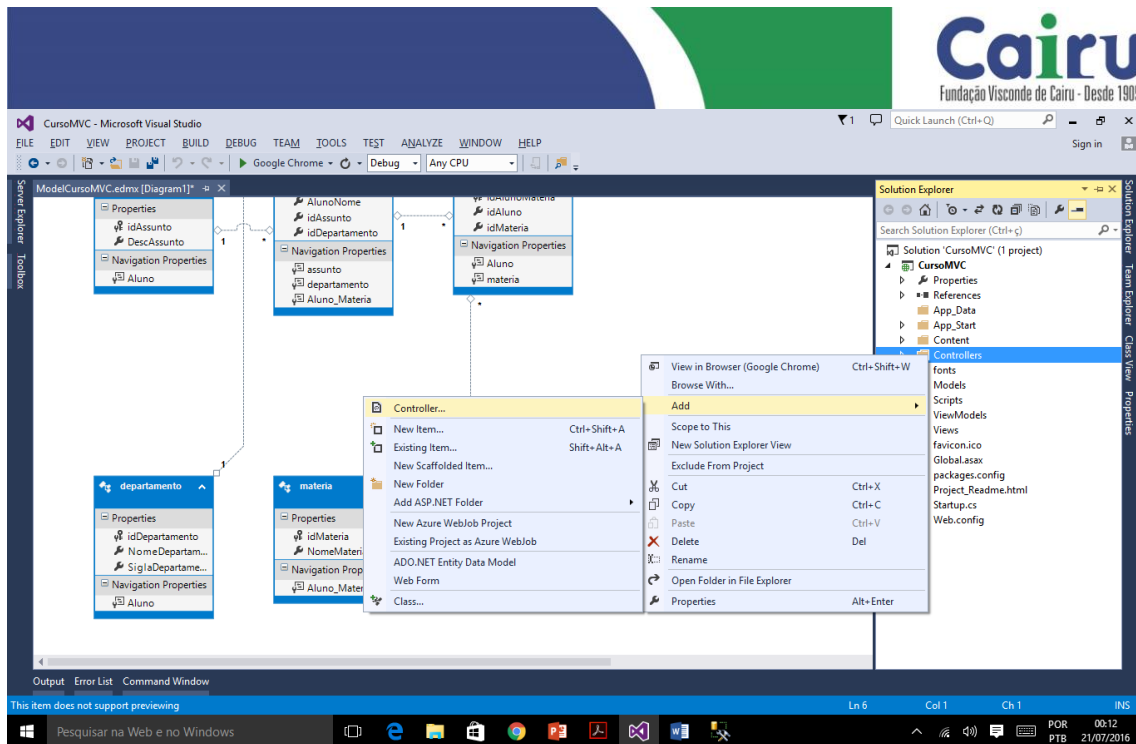
- ❖ Adicione as Data Annotations para alterar o título dos campos, antes importe a biblioteca DataAnnotation

`using System.ComponentModel.DataAnnotations;`

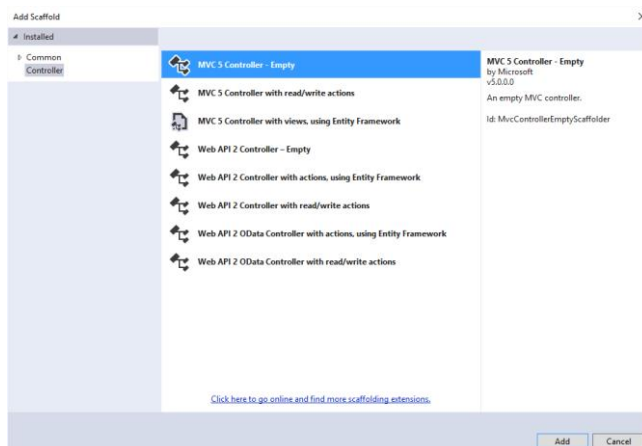
```
public partial class Aluno
{
    public int id { get; set; }
    [Display(Name="Nome")]
    [Required]
    public string Nome { get; set; }
    [Display(Name="Código do Assunto")]
    [Required]
    public int idAssunto { get; set; }
    [Display(Name="Código do Departamento")]
    [Required]
    public int idDepartamento { get; set; }

    public virtual assunto assunto { get; set; }
    public virtual departamento departamento { get; set; }
}
```

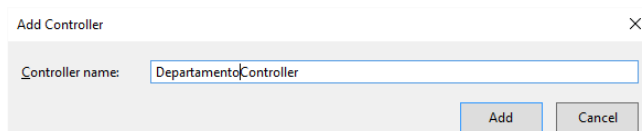
- ❖ Compile e teste
- ❖ Crie o Controller para Departamento



- ❖ Escolha a opção **MVC 5 Controller – Empty** para criar um Controller vazio



- ❖ Adicione o nome DepartamentoController



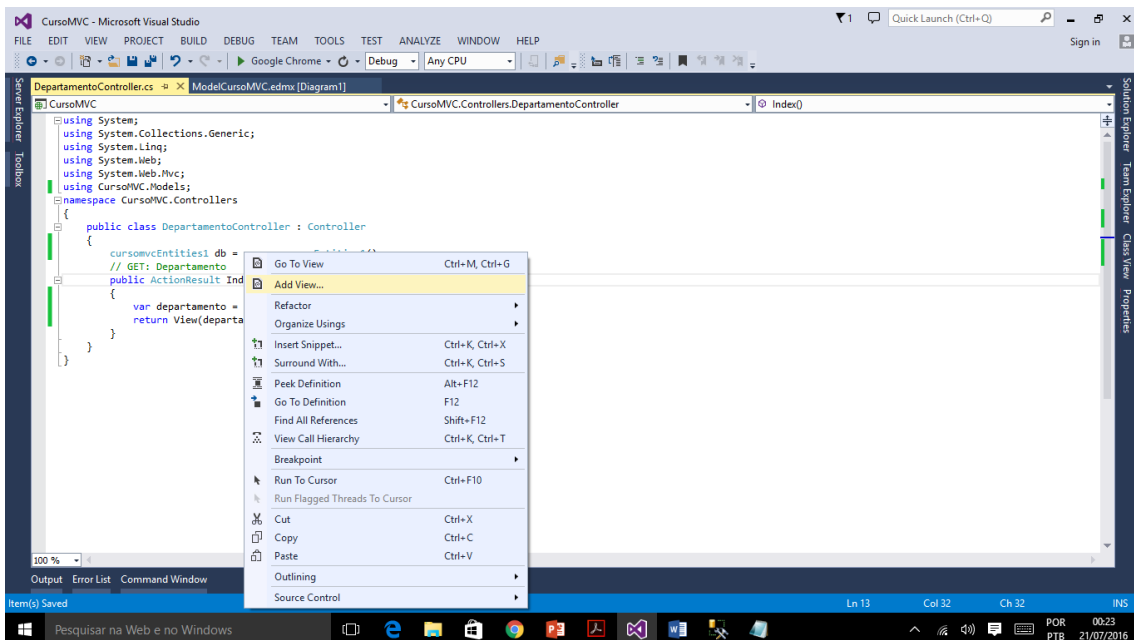
- ❖ Será exibida a classe DepartamentoController

```
namespace CursoMVC.Controllers
{
    public class DepartamentoController : Controller
    {
        // GET: Departamento
        public ActionResult Index()
        {
            return View();
        }
    }
}
```

- ❖ Crie um objeto do entity criado e no método Index() consulte e retorne a lista de departamentos

```
using CursoMVC.Models;
namespace CursoMVC.Controllers
{
    public class DepartamentoController : Controller
    {
        cursomvcEntities1 db = new cursomvcEntities1();
        // GET: Departamento
        public ActionResult Index()
        {
            var departamento = db.departamento.ToList();
            return View(departamento);
        }
    }
}
```

- ❖ Adicione uma view de Index a partir do Controller



- ❖ Informe os dados da view

Add View ✕

View name:

Template:

Model class:

Data context class:

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

...

(Leave empty if it is set in a Razor _viewstart file)

- ❖ Será exibida as informações da view



```
@model IEnumerable<CursoMVC.Models.departamento>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.NomeDepartamento)
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.NomeDepartamento)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.idDepartamento }) |
                @Html.ActionLink("Details", "Details", new { id=item.idDepartamento })
            </td>
        </tr>
    }
</table>
```

- ❖ Agora crie o método Edit na classe DepartamentoController

```
public ActionResult Edit(int id)
{
    var departamento = db.departamento.Find(id);
    return View(departamento);
}
```

- ❖ Adicione a view para Edit

- ❖ Na classe DepartamentoController adicione o método para fazer a alteração

```
[HttpPost]
public ActionResult Edit(departamento depart)
{
    if (ModelState.IsValid)
    {
        db.Entry(depart).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(depart);
}
```

- ❖ Agora crie o método Delete na classe DepartamentoController

```
public ActionResult Edit(int id)
```



```
{  
    var departamento = db.departamento.Find(id);  
    return View(departamento);  
}
```

❖ Adicione a view para Edit

❖ Na classe DepartamentoController adicione o método para fazer a remoção

```
[HttpPost, ActionName("Delete")]  
public ActionResult DeleteConfirmed(int id)  
{  
    var departamento = db.departamento.Find(id);  
    db.departamento.Remove(departamento);  
    db.SaveChanges();  
    return RedirectToAction("Index");  
}
```




- ❖ Agora crie o método Details na classe DepartamentoController

```
public ActionResult Details(int id)
{
    var departamento = db.departamento.Find(id);
    return View(departamento);
}
```

- ❖ Adicione a view para Details

- ❖ Agora crie o método Create na classe DepartamentoController

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

- ❖ Adicione a view para Create

- ❖ Na classe DepartamentoController adicione o método para fazer a inserção

```
[HttpPost]
public ActionResult Create([Bind(Include = "idDepartamento,
NomeDepartamento, SiglaDepartamento")] departamento departamento)
{
    if (ModelState.IsValid)
    {
        db.departamento.Add(departamento);
        db.SaveChanges();
    }

    return RedirectToAction("Index");
}
```





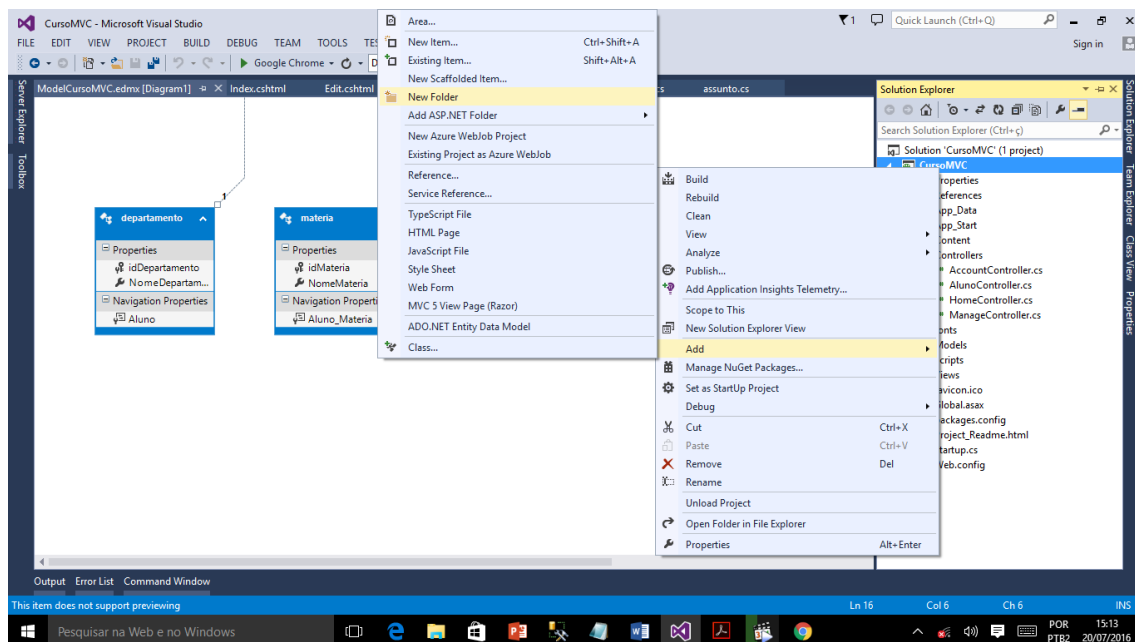








❖ Crie a pasta ViewModels



❖ Crie a classe AlunoMateriaViewModel

```
using CursoMVC.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CursoMVC.ViewModels
{
    public class AlunoMateriaViewModel
    {
        public Aluno aluno { get; set; }
        public Aluno_Materia aluno_Materia { get; set; }
    }
}
```

❖ Abra o Arquivo Alunos que se encontra na pasta Model