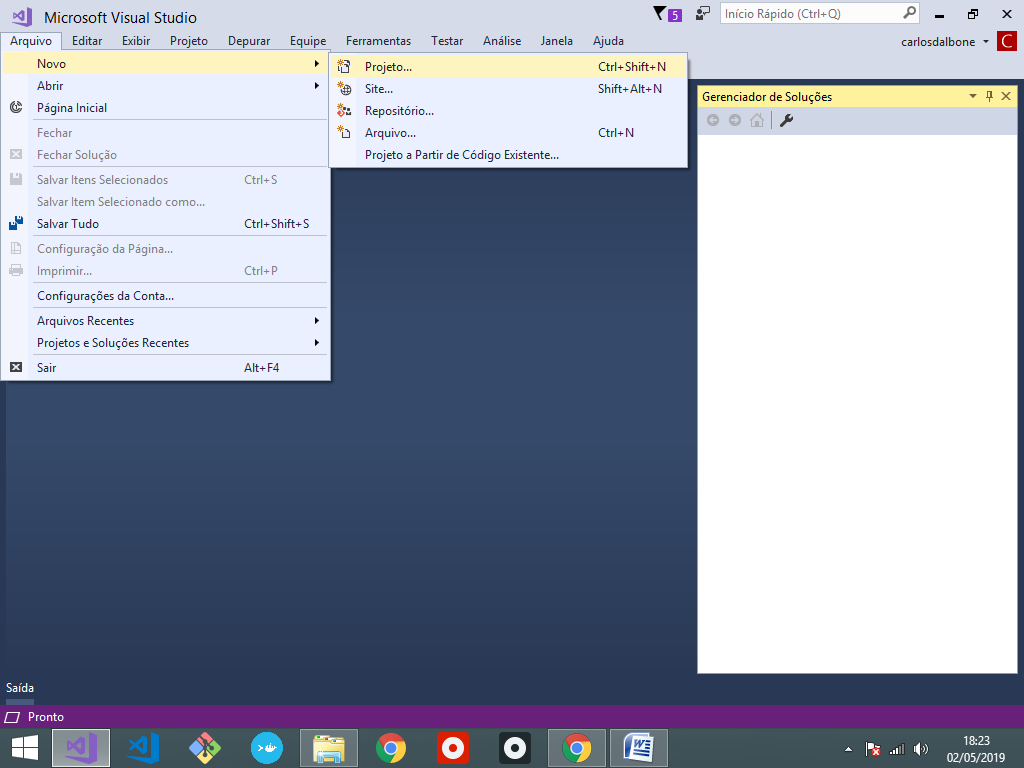
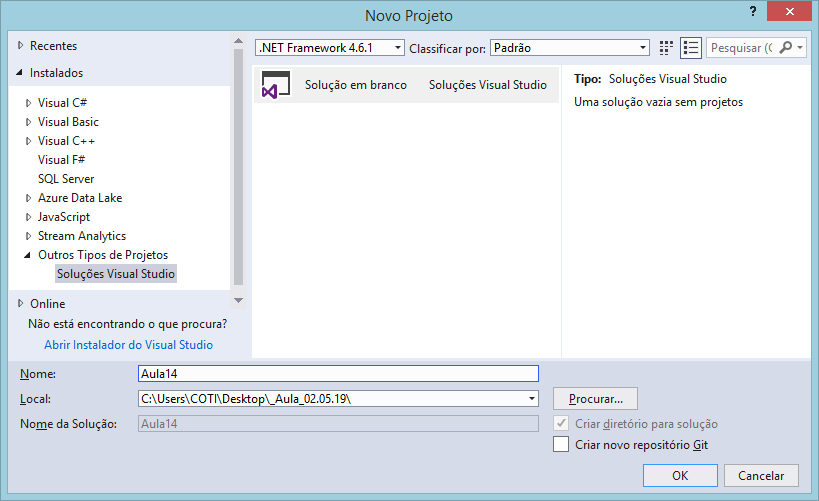
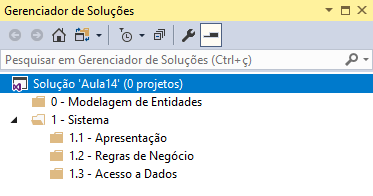
**Criando uma nova solution em branco:**

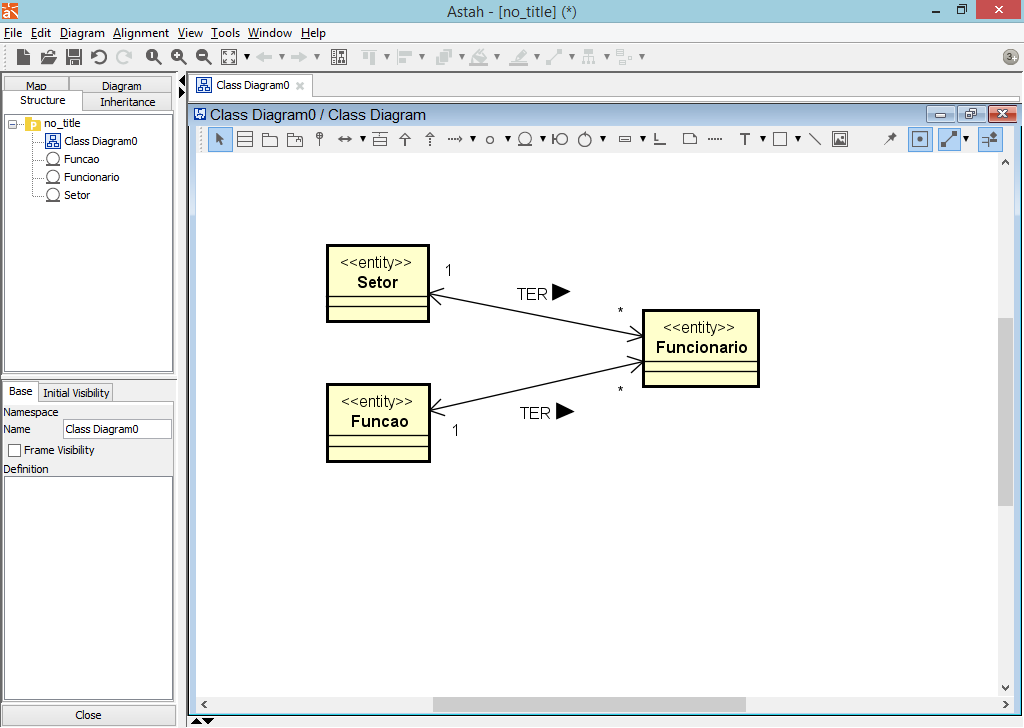


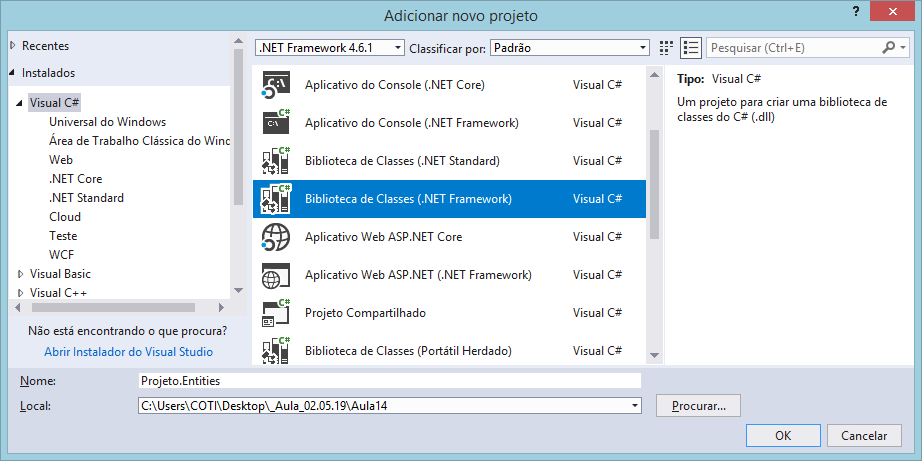




0 - Modelagem de entidades

Biblioteca de Classes .NET Framework





using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto.Entities

{

public class Funcionario

{

public int IdFuncionario { get; set; }

public string Nome { get; set; }

public decimal Salario { get; set; }

public DateTime DataAdmissao { get; set; }

public int IdSetor { get; set; } //FK no banco de dados

public int IdFuncao { get; set; } //FK no banco de dados

//Relacionamento com as demais classes

public Setor Setor { get; set; } //Funcionario TEM 1 Setor

public Funcao Funcao { get; set; } //Funcionario TEM 1 Função

public Funcionario()

{

}

public Funcionario(int idFuncionario, string nome, decimal salario,

DateTime dataAdmissao)

{

IdFuncionario = idFuncionario;

Nome = nome;

Salario = salario;

DataAdmissao = dataAdmissao;

}

public override string ToString()

{

return $"Id: {IdFuncionario}, Nome: {Nome},

Salário: {Salario}, Admissão: {DataAdmissao}";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto.Entities

{

public class Funcao

{

public int IdFuncao { get; set; }

public string Nome { get; set; }

//Relacionamento TER-MUITOS

public List<Funcionario> Funcionarios { get; set; }

public Funcao()

{

}

public Funcao(int idFuncao, string nome)

{

IdFuncao = idFuncao;

Nome = nome;

}

public override string ToString()

{

return $"Id: {IdFuncao}, Nome: {Nome}";

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Projeto.Entities

{

public class Setor

{

public int IdSetor { get; set; }

public string Nome { get; set; }

//Relacionamento TER-MUITOS

public List<Funcionario> Funcionarios { get; set; }

public Setor()

{

}

public Setor(int idSetor, string nome)

{

IdSetor = idSetor;

Nome = nome;

}

public override string ToString()

{

return $"Id: {IdSetor}, Nome: {Nome}";

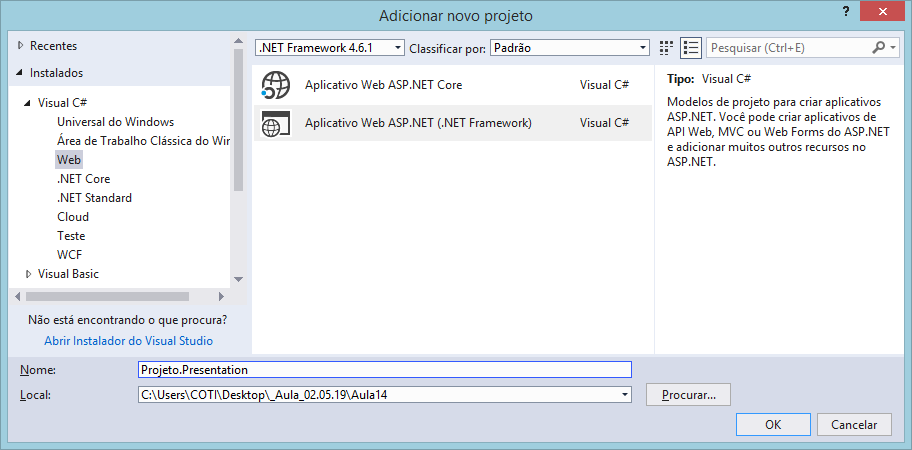
}

}

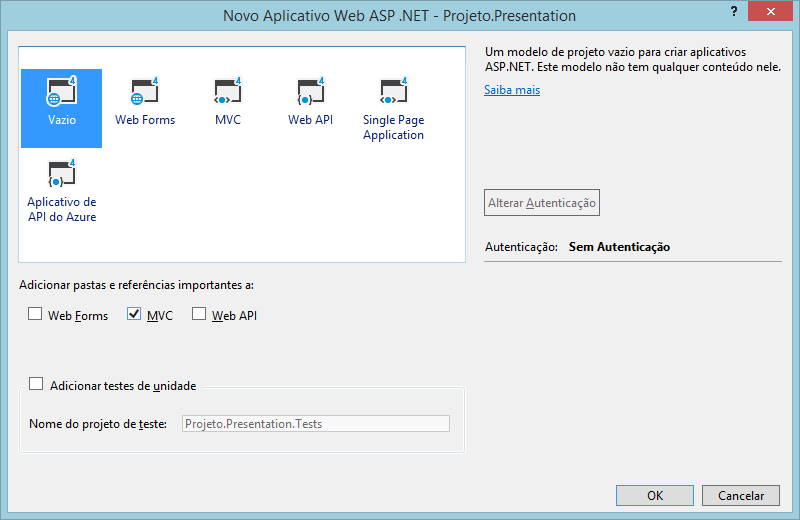
}

1.1 - Camada de Apresentação

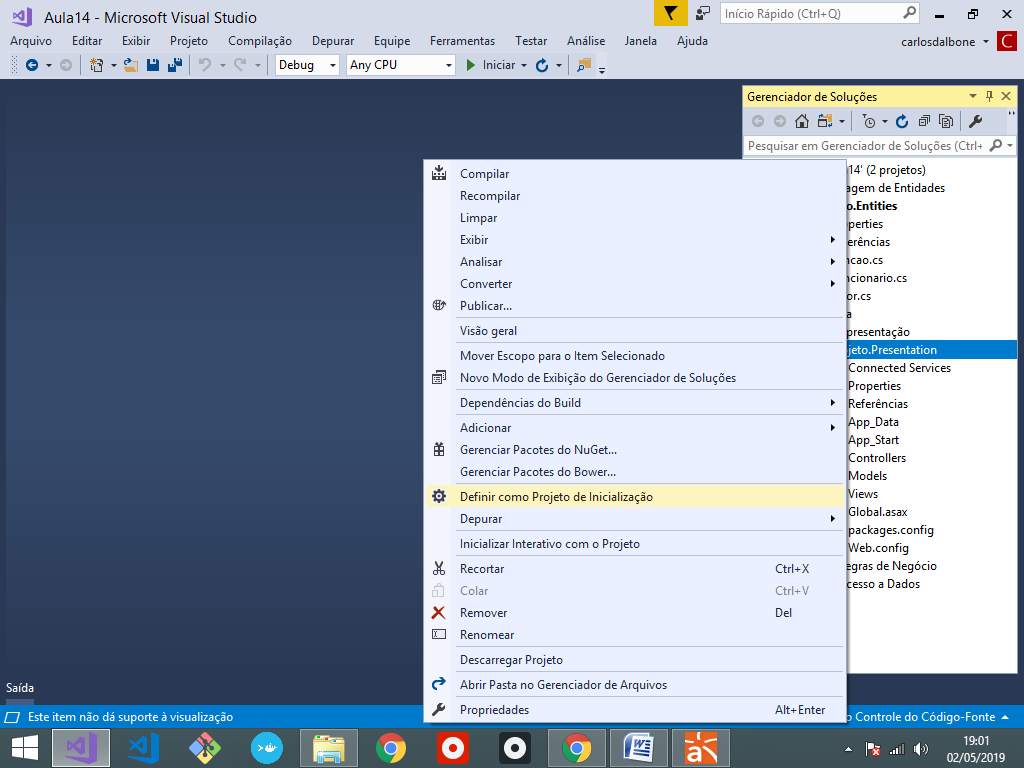
Projeto Asp.Net MVC



Escolha: Vazio / MVC

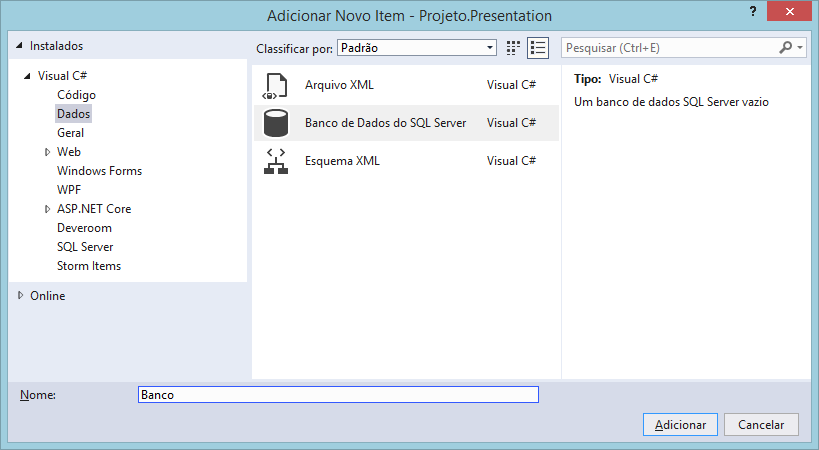


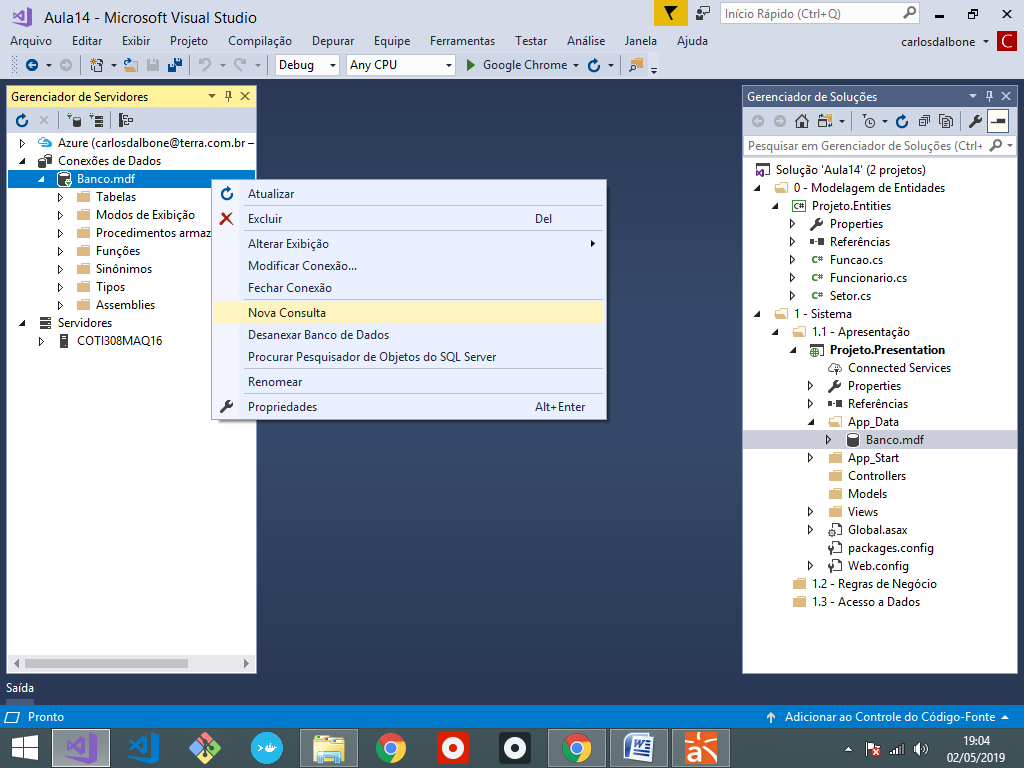
**Definindo o projeto de inicialização da solution:**



Criando um banco de dados

**MDF - Master Database File**





create table Setor(

IdSetor integer identity(1,1),

Nome nvarchar(100) not null,

primary key(IdSetor))

insert into Setor(Nome) values('Recursos Humanos')

insert into Setor(Nome) values('Desenvolvimento de Sistemas')

insert into Setor(Nome) values('Contabilidade')

insert into Setor(Nome) values('Engenharia de Produção')

create table Funcao(

IdFuncao integer identity(1,1),

Nome nvarchar(100) not null,

primary key(IdFuncao))

insert into Funcao(Nome) values('Estagiario')

insert into Funcao(Nome) values('Gerente')

insert into Funcao(Nome) values('Supervisor')

insert into Funcao(Nome) values('Operacional')

insert into Funcao(Nome) values('Analista')

create table Funcionario(

IdFuncionario integer identity(1,1),

Nome nvarchar(150) not null,

Salario decimal(18,2) not null,

DataAdmissao date not null,

IdSetor integer not null,

IdFuncao integer not null,

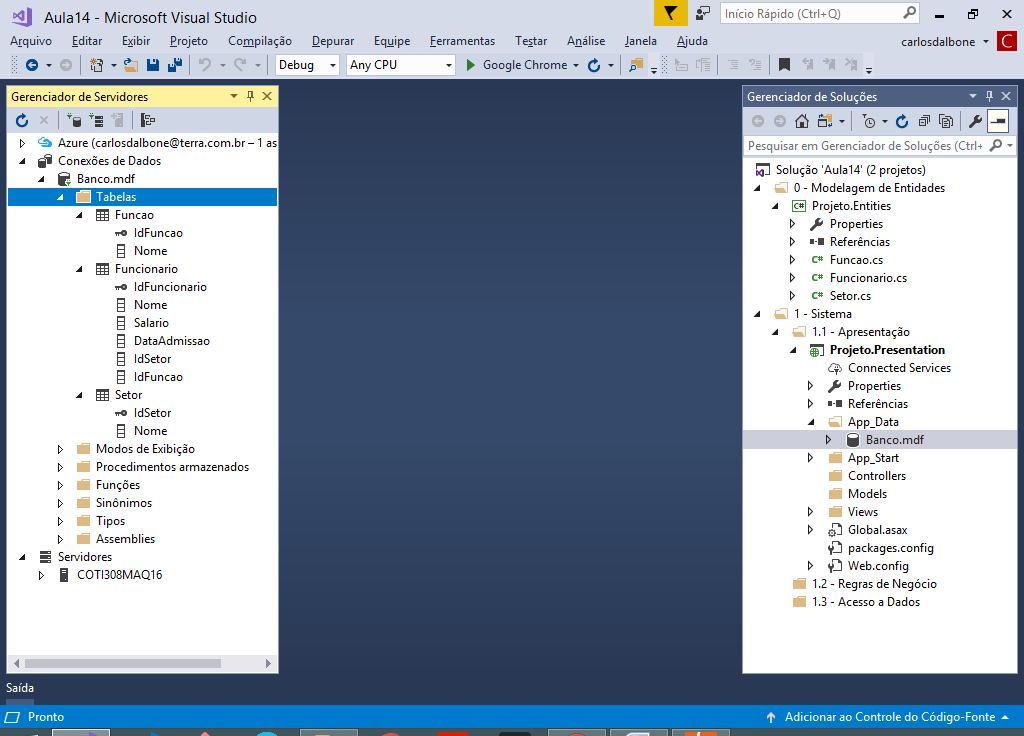
primary key(IdFuncionario),

foreign key(IdSetor) references Setor(IdSetor),

foreign key(IdFuncao) references Funcao(IdFuncao))

-------------------

**Tabelas criadas:**



Web.config.xml

Mapeando a string de conexão com o banco de dados

<connectionStrings>

<add

name="projeto"

connectionString="Data Source=(LocalDB)\MSSQLLocalDB;

AttachDbFilename=C:\Users\COTI\Desktop\

\_Aula\_02.05.19\Aula14\Projeto.Presentation\

App\_Data\Banco.mdf;Integrated Security=True"

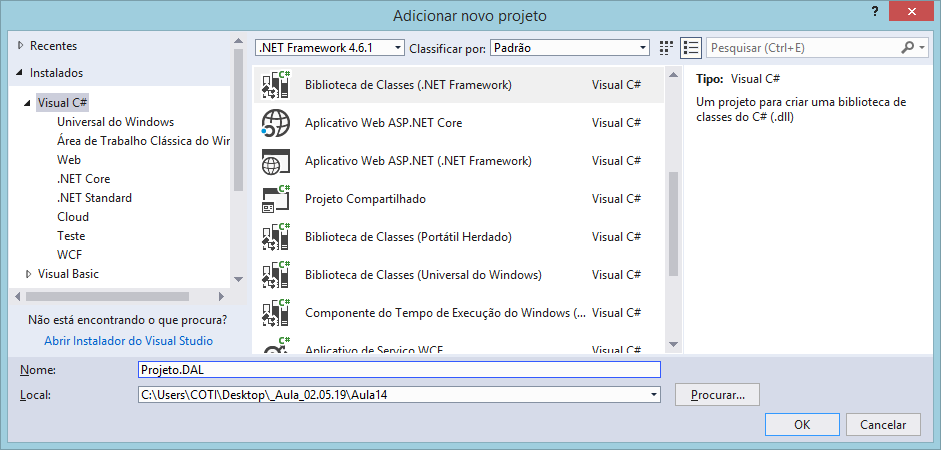
/>

</connectionStrings>

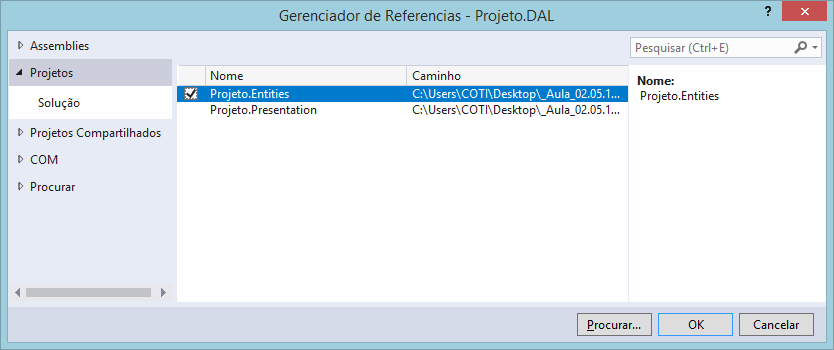
-----------------------

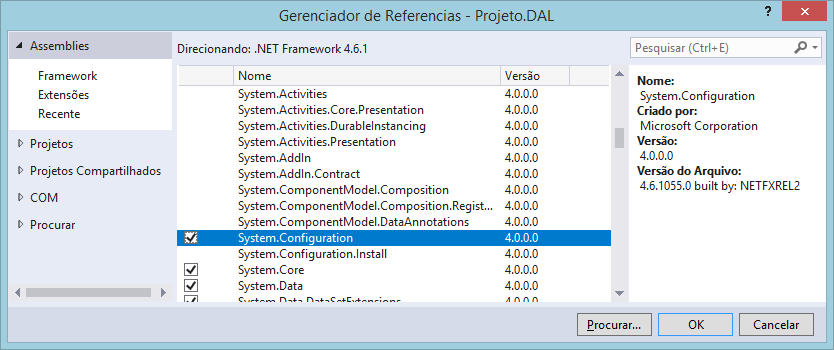
1.3 - Camada de Acesso a Dados

Repositório de banco de dados



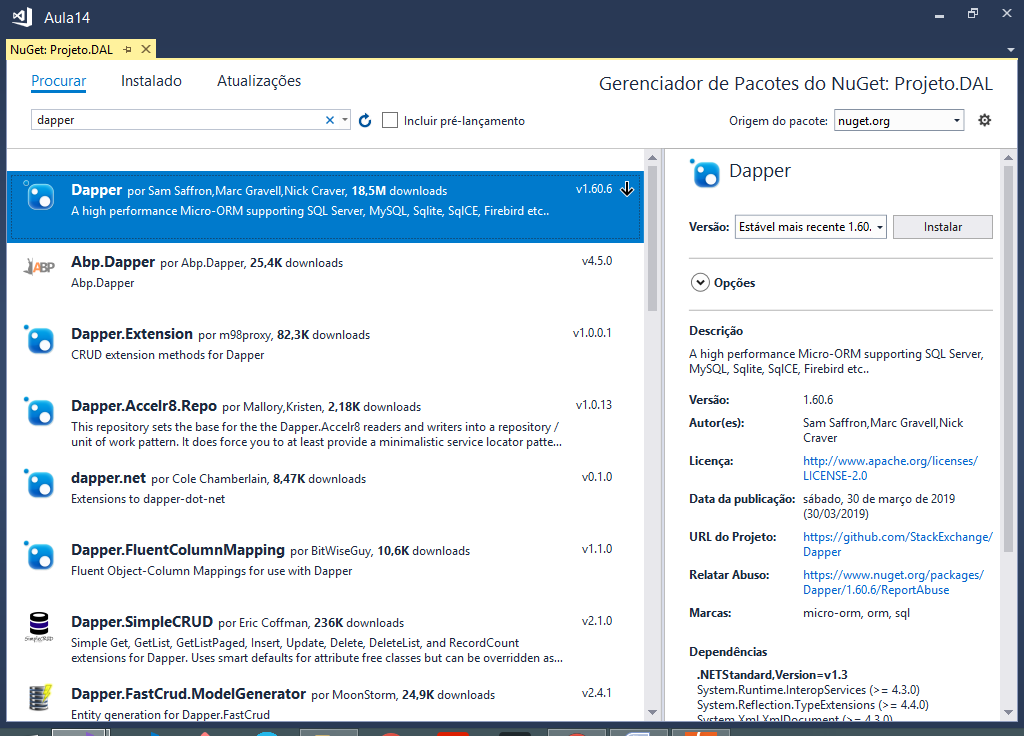
**Adicionando referencias no projeto:**





Instalando o Dapper:

Gerenciador de pacotes do NuGet



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient; //importando

using System.Configuration; //importando

using Projeto.Entities; //importando

using Dapper; //importando

namespace Projeto.DAL

{

public class FuncionarioRepository

{

//atributo

private string connectionString;

//construtor

public FuncionarioRepository()

{

connectionString = ConfigurationManager.ConnectionStrings

["projeto"].ConnectionString;

}

//método para inserir um funcionario na base de dados

public void Insert(Funcionario funcionario)

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

string query = "insert into Funcionario(Nome, Salario, "

+ "DataAdmissao, IdSetor, IdFuncao) "

+ "values(@Nome, @Salario, @DataAdmissao, "

+ "@IdSetor, @IdFuncao)";

conn.Execute(query, funcionario);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

using Dapper;

using Projeto.Entities;

namespace Projeto.DAL

{

public class FuncaoRepository

{

//atributo

private string connectionString;

//construtor

public FuncaoRepository()

{

connectionString = ConfigurationManager.ConnectionStrings

["projeto"].ConnectionString;

}

public List<Funcao> FindAll()

{

using (SqlConnection conn = new SqlConnection(connectionString))

{

string query = "select \* from Funcao order by Nome";

return conn.Query<Funcao>(query).ToList();

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

using Dapper;

using Projeto.Entities;

namespace Projeto.DAL

{

public class SetorRepository

{

//atributo..

private string connectionString;

public SetorRepository()

{

connectionString = ConfigurationManager.ConnectionStrings

["projeto"].ConnectionString;

}

public List<Setor> FindAll()

{

using (SqlConnection conn = new SqlConnection())

{

string query = "select \* from Setor order by Nome";

return conn.Query<Setor>(query).ToList();

}

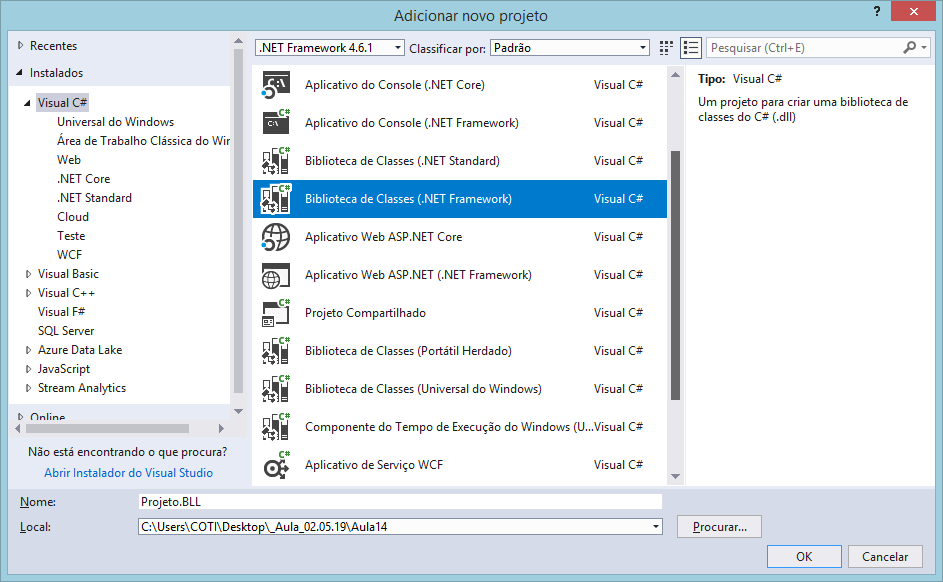
}

}

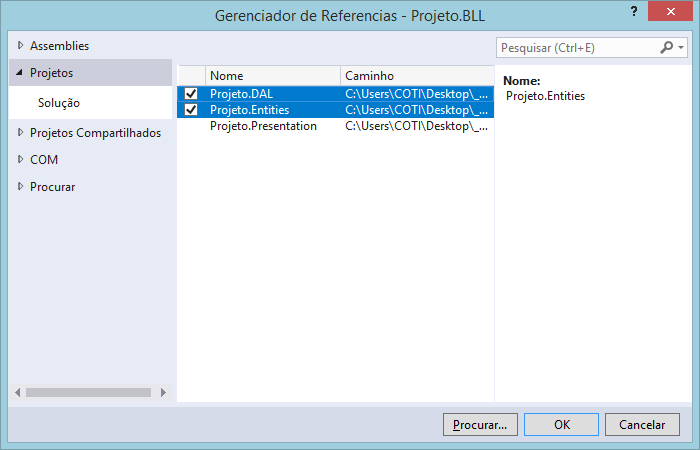
}

1.2 - Camada de Regras de Negócio:

Biblioteca de Classes .NET Framework



**Adicionando referencias:**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto.Entities; //importando

using Projeto.DAL; //importando

namespace Projeto.BLL

{

public class FuncionarioBusiness

{

//atributo

private FuncionarioRepository repository;

//construtor -> ctor + 2x[tab]

public FuncionarioBusiness()

{

repository = new FuncionarioRepository();

}

//método para cadastrar funcionario

public void CadastrarFuncionario(Funcionario funcionario)

{

repository.Insert(funcionario);

}

}

}

-----------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto.Entities;

using Projeto.DAL;

namespace Projeto.BLL

{

public class FuncaoBusiness

{

//atributo

private FuncaoRepository repository;

//construtor

public FuncaoBusiness()

{

repository = new FuncaoRepository();

}

//método para executar a consulta de funções

public List<Funcao> ConsultarFuncoes()

{

return repository.FindAll();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Projeto.DAL;

using Projeto.Entities;

namespace Projeto.BLL

{

public class SetorBusiness

{

//atributo

private SetorRepository repository;

public SetorBusiness()

{

repository = new SetorRepository();

}

public List<Setor> ConsultarSetores()

{

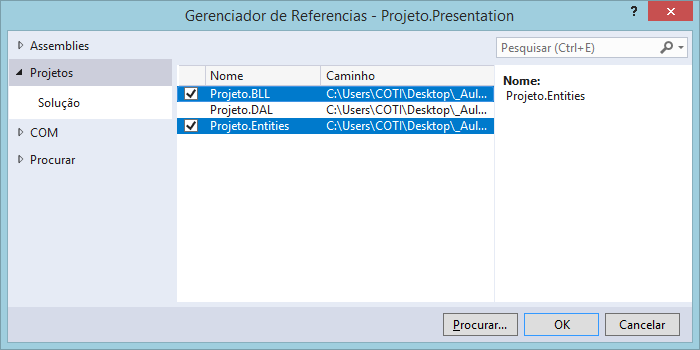
return repository.FindAll();

}

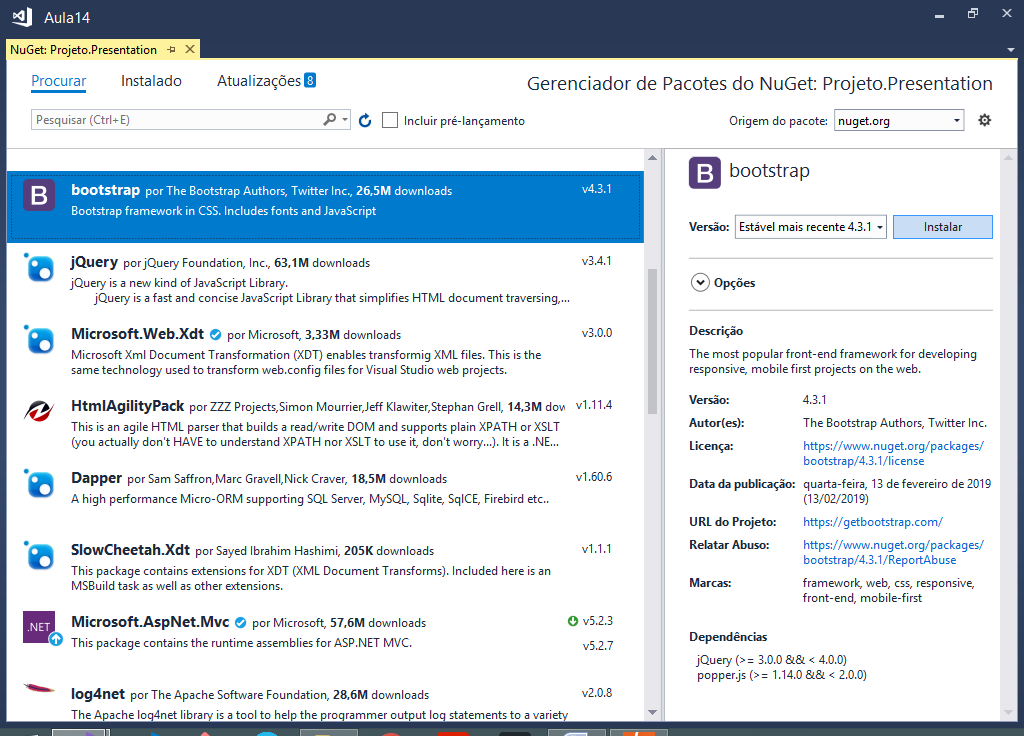
}

}

**Adicionando referências no projeto Asp.Net MVC:**

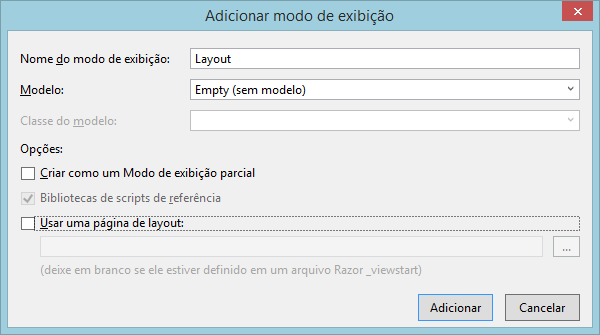


**Instalando o bootstrap:**



Criando uma página de layout mestre

/Views/Shared/Layout.cshtml



@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Projeto</title>

<!-- folhas de estilo CSS -->

<link href="~/Content/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

<div class="container">

<div class="card card-body bg-dark">

<h3 class="text-white">Sistema de Controle de Funcionários</h3>

</div>

<br/>

@RenderBody()

</div>

<!-- arquivos javascript -->

<script src="~/Scripts/jquery-3.0.0.min.js"></script>

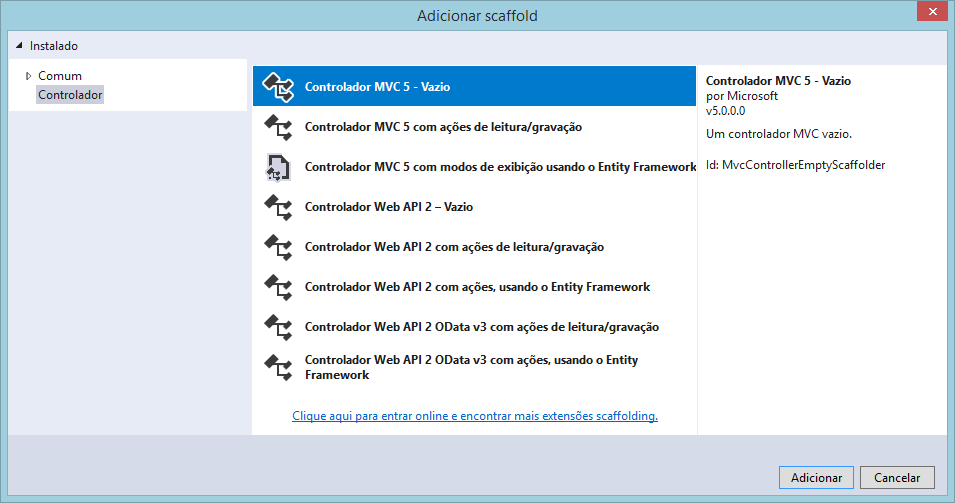
<script src="~/Scripts/bootstrap.min.js"></script>

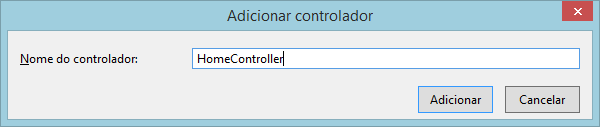
</body>

</html>

/Home/Index

[Controller] [View]





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Projeto.Presentation.Controllers

{

public class HomeController : Controller

{

// GET: Home/Index

public ActionResult Index()

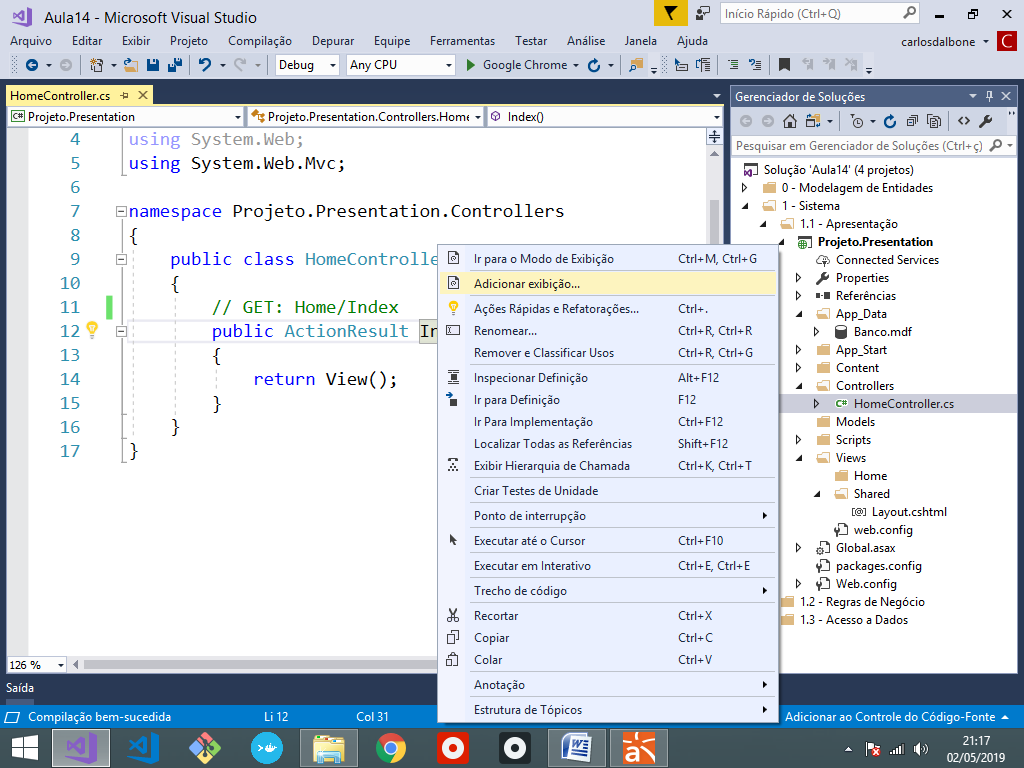
{

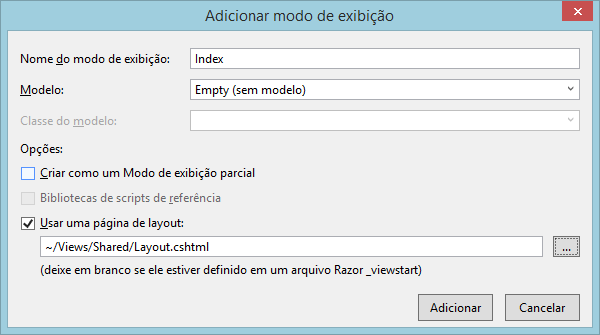
return View();

}

}

}





@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/Layout.cshtml";

}

<h4>Seja bem vindo ao Projeto</h4>

<hr/>

<a href="/Funcionario/Cadastro">

Cadastrar Funcionários

</a>

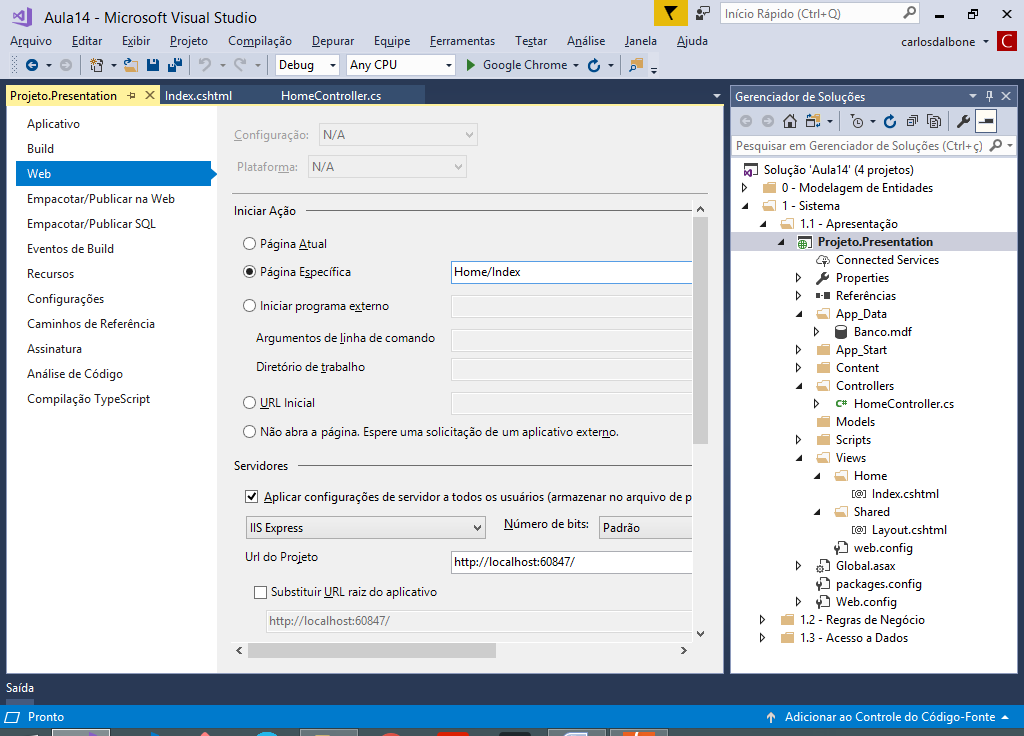
<br/>

<a href="/Funcionario/Consulta">

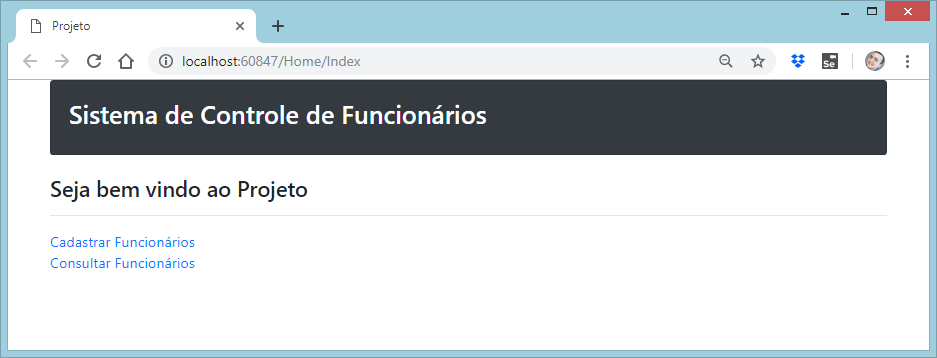
Consultar Funcionários

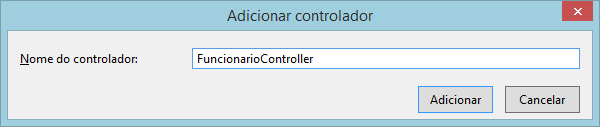
</a>

**Definindo a página inicial do projeto:**



<http://localhost:60847/Home/Index>





using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Projeto.Presentation.Controllers

{

public class FuncionarioController : Controller

{

// GET: Funcionario/Cadastro

public ActionResult Cadastro()

{

return View();

}

// GET: Funcionario/Consulta

public ActionResult Consulta()

{

return View();

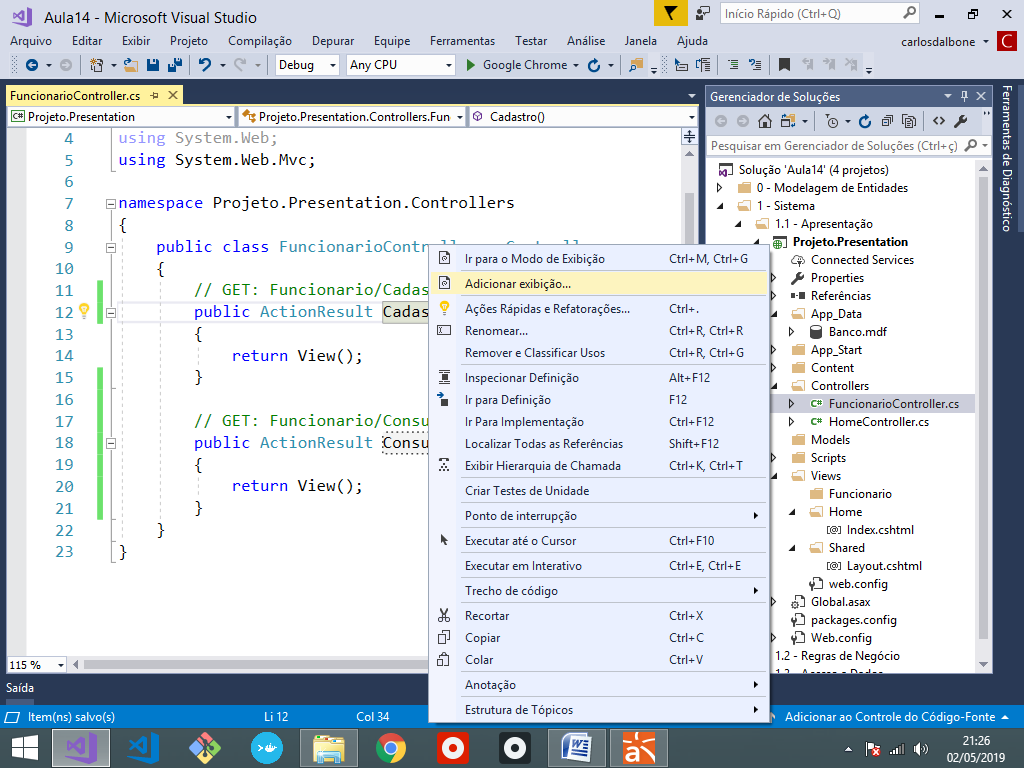
}

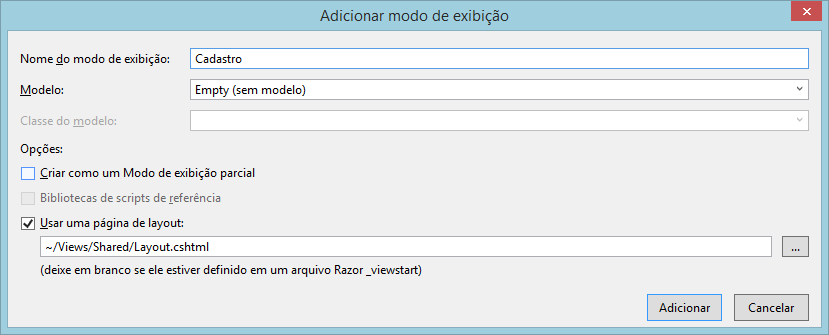
}

}

----------------------------------------

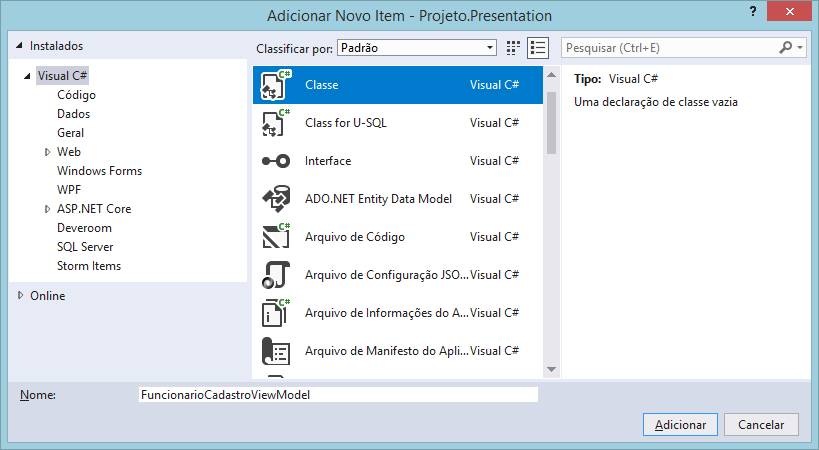
Criando as Views de Funcionario:





Classe de modelo (Model)

Criando uma ViewModel para cadastro de Funcionario



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.ComponentModel.DataAnnotations; //validações

namespace Projeto.Presentation.Models

{

public class FuncionarioCadastroViewModel

{

[Required(ErrorMessage = "Campo obrigatório")]

public string Nome { get; set; }

[Required(ErrorMessage = "Campo obrigatório")]

public decimal Salario { get; set; }

[Required(ErrorMessage = "Campo obrigatório")]

public DateTime DataAdmissao { get; set; }

}

}

**Criando o formulário para cadastro de funcionário contendo por enquanto os campos Nome, Salario e DataAdmissao:**

@model Projeto.Presentation.Models.FuncionarioCadastroViewModel

@{

ViewBag.Title = "Cadastro";

Layout = "~/Views/Shared/Layout.cshtml";

}

<h4>Cadastro de Funcionários</h4>

<a href="/Home/Index">Página inicial</a>

<hr/>

<div class="row">

<div class="col-md-4">

@using (Html.BeginForm())

{

<label>Nome do Funcionário:</label>

@Html.TextBoxFor(model => model.Nome,

new { @class = "form-control" })

<span class="text-danger">

@Html.ValidationMessageFor(model => model.Nome)

</span>

<br/>

<label>Salário:</label>

@Html.TextBoxFor(model => model.Salario,

new { @class = "form-control" })

<span class="text-danger">

@Html.ValidationMessageFor(model => model.Salario)

</span>

<br />

<label>Data de Admissão:</label>

@Html.TextBoxFor(model => model.DataAdmissao,

new { @class = "form-control", @type = "date" })

<span class="text-danger">

@Html.ValidationMessageFor(model => model.DataAdmissao)

</span>

<br />

<input type="submit" value="Cadastrar Funcionário"

class="btn btn-success"/>

<br />

<br />

<strong>@TempData["Mensagem"]</strong>

}

</div>

</div>



**Criando o método HttpPost na classe FuncionarioController:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Projeto.Entities; //importando

using Projeto.BLL; //importando

using Projeto.Presentation.Models; //importando

namespace Projeto.Presentation.Controllers

{

public class FuncionarioController : Controller

{

//atributo

private FuncionarioBusiness business;

//construtor -> ctor + 2x[tab]

public FuncionarioController()

{

business = new FuncionarioBusiness();

}

// GET: Funcionario/Cadastro

public ActionResult Cadastro()

{

return View();

}

**// POST: Funcionario/Cadastro**

**[HttpPost] //método recebe SUBMIT do formulário**

**public ActionResult Cadastro(FuncionarioCadastroViewModel model)**

**{**

**return View();**

**}**

// GET: Funcionario/Consulta

public ActionResult Consulta()

{

return View();

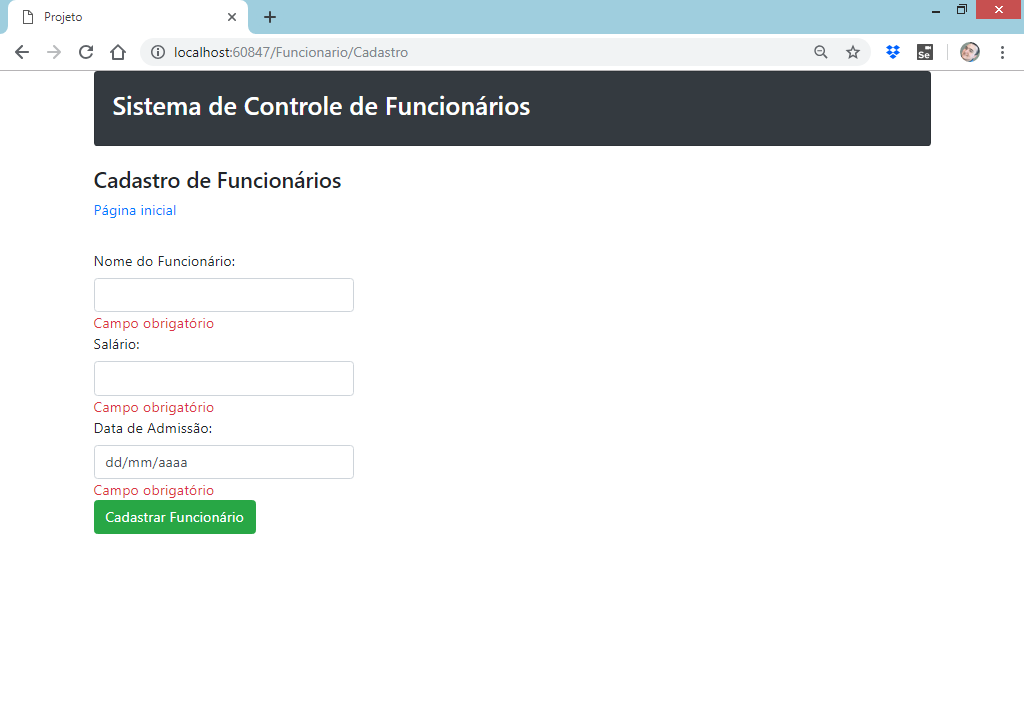
}

}

}

**Executando:**

<http://localhost:60847/Funcionario/Cadastro>



Continua...