

# Aula 06 – Criar Aplicação MVC com Banco de Dados – Relacionamento N:N (code first)

---

Disciplina: Programação Web

Prof. Allbert Velleniche de Aquino Almeida

E-mail: [allbert.almeida@fatec.sp.gov.br](mailto:allbert.almeida@fatec.sp.gov.br)

Site: <http://www.allbert.com.br>



/allbert.almeida

# Relacionamento N:N

---

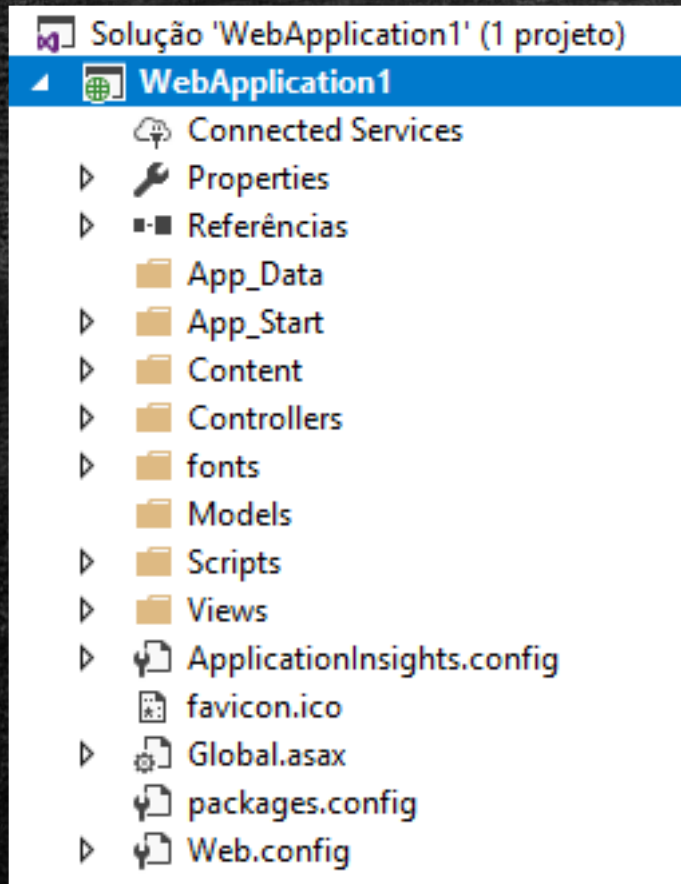
- O objetivo dessa aula é construir uma aplicação utilizando EntityFramework com relacionamento N:N;

# Criando uma nova aplicação

---

- Vamos abrir o Visual Studio;
- File >> New >> Project >> Aplicativo Web;
- Tipo de aplicação MVC;
- Change Authentication >> No Authentication

# Exemplo de uma aplicação



# Adicionar pacotes à solução

---

- MySql Data Entity 6.9.12

**Install-Package MySql.Data.Entity  
-Version 6.9.12**

**EntityFramework** por Microsoft



Entity Framework is Microsoft's recommended data access technology for new applications.



**MySql.Data.Entity** por Oracle



Entity Framework 6.0 supported

# Modelo

---

Usuario
Id
Nome
Email
Senha

UsuarioPerfil
Id
UsuarioId
PerfilId

Perfil
Id
Descricao



# Criar uma classe Usuario

---

```
public class Usuario {
    public int Id { get; set; }
    [MaxLength(255)]
    public string Nome { get; set; }
    [MaxLength(100)]
    [EmailAddress]
    public string Email { get; set; }
    [Required]
    [DataType(DataType.Password)]
    [RegularExpression("(?=.*\\d)(?=.*[a-z])(?=.*[A-
Z]).{6,10}")]
    public string Senha { get; set; }
    [Required]
    [DataType(DataType.Password)]
    [Compare("Senha")]
    public string ConfirmaSenha { get; set; }
}
```

# Criar uma classe Perfil

---

```
public class Perfil
{
    public int Id { get; set; }
    [Required]
    [DisplayName("Descrição")]
    [MaxLength(255)]
    public string Descricao { get; set; }
}
```



# Criar uma classe UsuarioPerfil

---

```
public class UsuarioPerfil {  
    public int Id { get; set; }  
    [DisplayName("Usuário")]  
    public int UsuarioId { get; set; }  
    [DisplayName("Perfil")]  
    public int PerfilId { get; set; }  
    public virtual Usuario Usuario { get; set; }  
    public virtual Perfil Perfil { get; set; }  
}
```

# Criar uma classe Contexto

---

```
public class Contexto:DbContext
{
    public Contexto() : base(nameOrConnectionString:
"StringConexao") { }

    public DbSet<Usuario> Usuario { get; set; }
    public DbSet<Perfil> Perfil { get; set; }
    public DbSet<UsuarioPerfil> UsuarioPerfil { get; set; }
}
```

# Adicionar a String de Conexão no WebConfig

---

```
<connectionStrings>
```

```
    <add name="StringConexao"  
providerName="MySql.Data.MySqlClient"  
connectionString="server=localhost;dat  
abase=teste;uid=root;pwd='root'" />
```

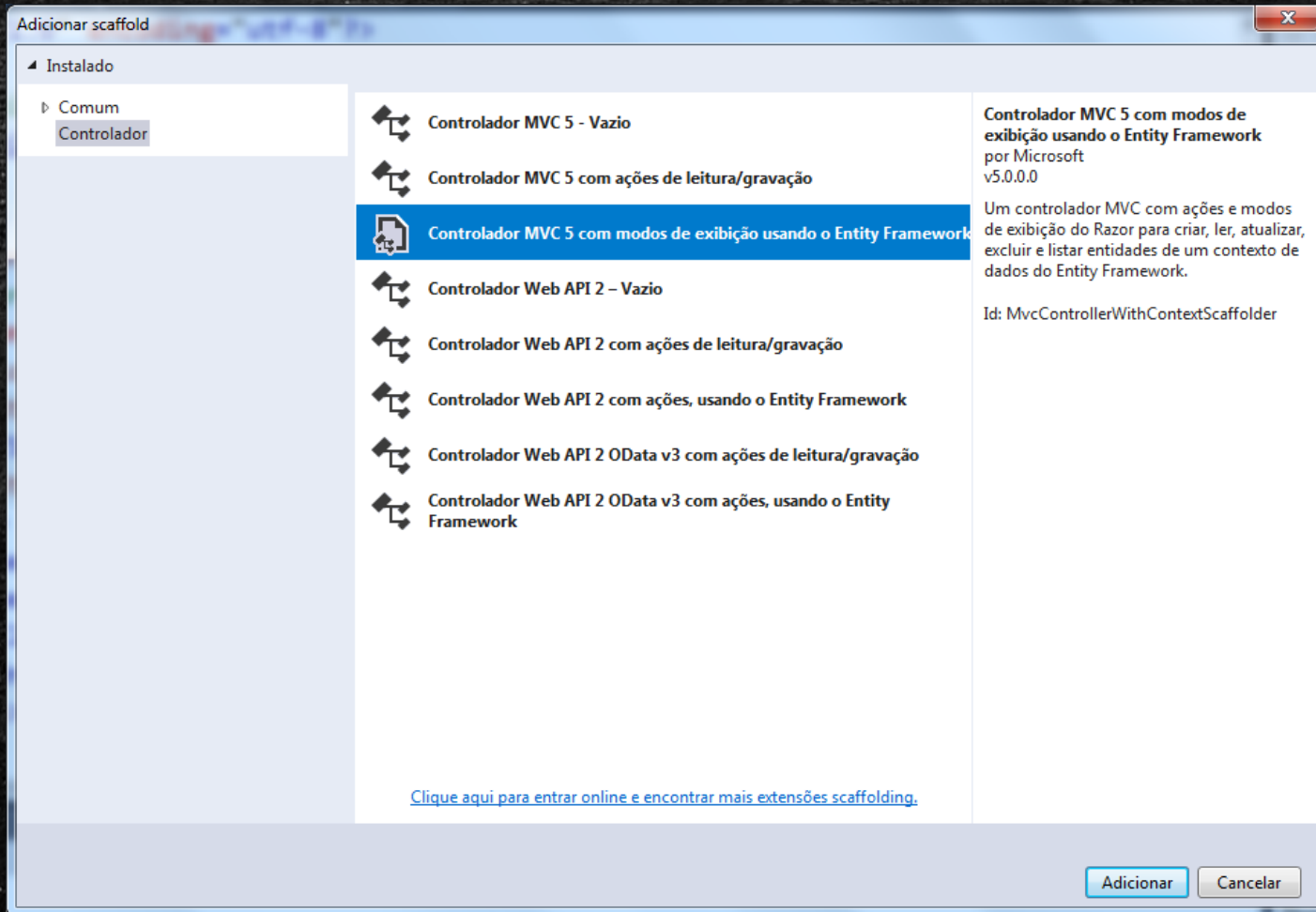
```
</connectionStrings>
```

# Compilar a solução

---

- Compilar a solução para que os modelos sejam validados e seja possível criar o “CRUD” da classe Usuario; Perfil; UsuarioPerfil

# Adicionar Controller



# Adicionar Controller

Adicionar controlador

Classe do modelo: Usuario (WebApplication1.Models)

Classe de contexto de dados: Contexto (WebApplication1.Models) +

Usar ações assíncronas do controlador

Modos de exibição:

Gerar modos de exibição

Bibliotecas de scripts de referência

Usar uma página de layout:

...

(deixe em branco se ele estiver definido em um arquivo Razor \_viewstart)

Nome do controlador: UsuariosController

Adicionar Cancelar

# Executar a solução

---

- Na URL entre com o caminho da aplicação +  
/Perfils/
- Na URL entre com o caminho da aplicação +  
/Usuarios/
- Na URL entre com o caminho da aplicação +  
/UsuarioPerfils/

# Adicionar a propriedade virtual ICollection - Usuario

---

```
public virtual
```

```
ICollection<UsuarioPerfil>
```

```
UsuarioPerfil { get; set; }
```



# Alterar a View Index na pasta Usuarios

---

```
@foreach(var item2 in item.UsuarioPerfil)
{
    @item2.Perfil.Descricao<br />
}
```

# Compilar a solução novamente

---

- Compilar a solução novamente para que os modelos sejam validados e seja possível ver a lista de perfis de um usuário;

# Atividade Prática

---

- Construir uma nova aplicação com BD utilizando o conceito code first, com os relacionamentos das entidades e gere o CRUD;
- Crie as classes Livro (Título, Edição, Nro páginas e ano), Editora(Nome, telefone e Cnpj) e Autor(Nome, país, sexo e breve nota biográfica.);