

# Isolando o ASP.Net Identity

- 1- Criar uma solução em branco
  - a. menu: File > New > Project; Abrirá a janela *New Project*:
  - b. painel esquerdo: Installed > Templates > Other Project Types > Visual Studio Solutions
  - c. painel direito: Clicar em “Blank Solution”
    - i. Name: *EDCT1T.IsolandIdentity.SEU\_NOME.Solution*
    - ii. Location: defina um local para o diretório da solução.
- 2- Adicionar as pastas da solução. Pastas de solução servem para organizar a solução no VS exclusivamente, não refletem pastas físicas.
  - a. Botão direito sobre a solução: Add > New Solution Folder
    - i. 1- Presentation
    - ii. 2- Service
    - iii. 3- Application
    - iv. 4- Domain
    - v. 5- Infra
  - b. A camada de Infra, deve ser dividida em duas camadas, então, devemos clicar com o botão direito sobre a pasta “5- Infra”: Add > New Solution Folder
    - i. 5.1- Data
    - ii. 5.2- CrossCutting
  - c. Como a camada de CrossCutting tende a ter vários projetos, talvez você queira criar novas pastas de solução para subdividir esta, fica a seu critério.
- 3- **Vamos adicionar um projeto MVC na pasta “1-Presentation”**. Add > New Project
  - a. Na janela para New Project: Installed > Visual C# > Web: painel direito: ASP.NET Web Application
    - i. Name: *EDCT1T.SEU\_NOME.Presentation.MvcApp*
    - ii. Location: mantenha o diretório da solução
    - iii. Atenção: Desmarque a opção (checkbox) “Add Application Insights to Project”, no painel da direita. Isso será visto no futuro.
  - b. Na janela seguinte New ASP.NET Project, selecione o template “MVC”.  
Se você estiver usando um Visual Studio que não seja o 2017, clique em “Change Authentication” e mude para “No Authentication”.  
Clique em “OK”.
- 4- **Vamos adicionar um projeto Class Library na pasta “4-Domain”**. Add > New Project
  - a. Na janela para New Project: Installed > Visual C#: painel direito: “Class Library”.  
No Visual Studio 2017: “Class Library (.Net Framework)”
    - i. Name: *EDCT1T.SEU\_NOME.Domain*
    - ii. Exclua o arquivo “class1.cs”
    - iii. Expanda “References” e remova as referências, exceto “System” e “Analyzers”.



5- Agora vamos adicionar um projeto Class Library na pasta “5.2-CrossCutting”.

No Visual Studio 2017: “Class Library (.Net Framework)”

- a. Name: *EDCT1T.SEU\_NOME.Infra.CrossCutting.Security*
- b. Expanda “References” e remova as referências, exceto “System” e “Analyzers”.
- c. Via Package Manager Console, instalar os pacotes abaixo para a camada do *Security*:
  - i. `PM> Install-Package EntityFramework` (foto 2016-01-23)
  - ii. `PM> Install-Package Microsoft.AspNet.Identity.EntityFramework`
  - iii. `PM> Install-Package Microsoft.AspNet.Identity.Owin` (foto 2016-01-23 (1))
- d. Add reference
  - i. System.Web.Mvc (Use Browse, se o namespace não aparecer na lista de Assemblies, e vá até a pasta Bin do projeto MvcApp)
  - ii. System.Configuration
  - iii. System.Web
- e. Precisamos anexar os arquivos do Projeto Security (Identity).
  - A ideia é não ter que adicionar de maneira individual cada um dos arquivos necessários ao projeto. Arquivos que estariam no projeto MVC, se você tivesse criado um projeto MVC com autenticação. É mais simples corrigir o namespace nos arquivos anexados.
  - Então, pegue o pacote zipado: “ArquivosDoProjetoIdentity.zip” e coloque seu conteúdo dentro da pasta do projeto Security.
  - Exemplo:  
`E:\VSO\EDCT1T.IsolandIdentity.SEU_NOME.Solution\EDCT1T.SEU_NOME.Infra.CrossCutting.Security`
  - Você terá as pastas: **Configurations, Contexts e Models**
  - Precisamos fazer os arquivos aparecerem no projeto: clique no projeto e depois, clique no botão Show All Files, na barra de botões da Solution Explorer.
  - Clique com o botão direito sobre cada uma das pastas acima e depois em Include In Project.
  - Clique novamente sobre Show All Files, para que as demais pastas ocultas, permanecem ocultas. :)
  - Agora precisamos acertar o namespace de todos os arquivos e as referências feitas com a instrução using se necessário. Se fez tudo certo, só precisará trocar o início do namespace.  
Você pode usar o recurso de “replace” no “Project current” (Ctrl+H). Conforme fiz em sala.
  - Execute um Build exclusivamente sobre o projeto. Para garantir que não fez nada de errado. Revise todos os arquivos, mesmo que não seja apresentado nenhum erro.

6- Vamos realizar o processo de anexar os arquivos do projeto do Domain. Temos que realizar as mesmas operações do projeto Security.

- a. Descompacte o conteúdo do arquivo ArquivosDoProjetoDomain.zip  
Terá as pastas: **Entities e Interfaces**
- b. Ajuste os namespaces:
  - i. Domain.Entities.*Usuario*  
Domain.Interface.Repository.*IUsuarioRepository*



Como eu falei em sala, a Microsoft não oferece uma gestão para as Entidades do Identity (tabelas). O que estou oferecendo aqui, é uma forma de manipular os dados nas tabelas do Identity utilizando uma estrutura própria. Abaixo, segue a lista de entidades do Namespace Microsoft.AspNet.Identity.EntityFramework.

A nossa classe **Usuário** acima, representa a classe **IdentityUser** do Identity.

## O AspNet.Identity tem suas classes definidas no Namespace Microsoft.AspNet.Identity.EntityFramework

[https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework(v=vs.108).aspx)

Onde temos as definições das classes:

- **IdentityRole** Class (AspNetRoles Table)  
([https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityrole\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityrole(v=vs.108).aspx))
  - **Id** - Obtém ou define o ID de papel.
  - **Name** - Obtém ou define o nome do papel.
  - **Users** - Obtém a coleção de usuários no papel.
  
- **IdentityUser** Class (AspNetUsers Table)  
([https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuser\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuser(v=vs.108).aspx))
  - **Id** - Obtém ou define o identificador do usuário.
  - **UserName** - Obtém ou define o nome do usuário.
  - **Email** - Obtém ou define o e-mail para o usuário.
  - **EmailConfirmed** - Obtém ou define um valor que indica se o e-mail é confirmada.
  - **PasswordHash** - Obtém ou define a forma salted/hashed da senha do usuário.
  - **SecurityStamp** - Obtém ou define um valor aleatório que muda quando as credenciais de um usuário mudam.
  - **PhoneNumber** - Obtém ou define o número de telefone para o usuário.
  - **PhoneNumberConfirmed** - Obtém ou define o valor que indica se o número de telefone é confirmada.
  - **TwoFactorEnabled** - Obtém ou define um valor que indica se a autenticação de dois fatores está habilitado para o usuário.
  - **LockoutEndDateUtc** - Obtém ou define o valor de data e hora (em UTC), quando termina bloqueio, a qualquer momento no passado não é considerado bloqueado.
  - **LockoutEnabled** - Obtém ou define um valor que indica se o bloqueio habilitado para esse usuário.
  - **AccessFailedCount** - Obtém ou define o número de falhas para os efeitos do bloqueio.
  - **Claims** - Obtém a coleção de claims do usuário.
  - **Logins** - Obtém a coleção de logins para o usuário.
  - **Roles** - Obtém a coleção de papéis para o usuário.



- **IdentityUserRole** Class (AspNetUserRoles Table)  
([https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserrole\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserrole(v=vs.108).aspx))
  - **RoleId** - Obtém ou define o ID da função para o papel.
  - **UserId** - Obtém ou define a identificação para o usuário que está no papel.
  
- **IdentityUserClaim** Class (AspNetUserClaims Table)  
([https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserclaim\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserclaim(v=vs.108).aspx))
  - **Id** - Obtém ou define a chave primária.
  - **UserId** - Obtém ou define o ID do usuário para o usuário que possui esse login.
  - **ClaimType** - Obtém ou define o tipo de reivindicação.
  - **ClaimValue** - Obtém ou define o valor pedido.
  
- **IdentityUserLogin** Class (AspNetUserLogins Table)  
([https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserlogin\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.identity.entityframework.identityuserlogin(v=vs.108).aspx))
  - **LoginProvider** - Obtém ou define o provedor de login para o login.
  - **ProviderKey** - Obtém ou define a chave que representa o início de sessão para o provedor.
  - **UserId** - Obtém ou define o ID do usuário para o usuário que possui esse login.



- 7- Vamos adicionar o projeto *EDCT1T.SEU\_NOME.Infra.Data* e repetir os processos anteriores. Descompactando o arquivo zipado do projeto *Infra.Data*: *ArquivosDoProjetoInfraData.zip*. Você terá as pastas: **Contexts, Mappings e Repositories**. Instale o pacote abaixo e adicione a referência.
- `PM>Install-Package EntityFramework`
  - Adicionar referência ao `PSTI.IsolandoIdentiy.Domain`
- 8- Vamos agora adicionar o projeto de IoC. *EDCT1T.SEU\_NOME.Infra.CrossCutting.IoC*. Precisaremos instalar o EntityFramework, SimpleInjector, e as referências aos projetos **Domain, Infr.CrossCutting.Security e Infra.Data**, mas antes exclua todas as referências, menos System.
- `PM>Install-Package EntityFramework`
  - `PM> Install-Package SimpleInjector.Integration.Web`. O SimpleInjector será instalado por ser dependência.
  - `PM> Install-Package Microsoft.AspNet.Identity.EntityFramework`
  - `PM> Install-Package Microsoft.AspNet.Identity.Owin`
  - Adicionar a classe StartupIoC fornecida.
- 9- Bom... Agora é com a camada de apresentação MVC.
- Primeiro, precisamos instalar as referências:
  - `Install-Package EntityFramework`
  - `Install-Package SimpleInjector.Integration.Web.Mvc`. Este depende dos outros dois (SimpleInjector e Integration.Web) que serão instalados automaticamente.
  - `Install-Package Microsoft.AspNet.Identity.EntityFramework`
  - `Install-Package Microsoft.AspNet.Identity.Owin`
  - `Install-Package WebActivatorEx`
  - `Install-Package Microsoft.Owin.Host.SystemWeb`
  - Infra.CrossCutting.Security*
  - Infra.CrossCutting.IoC*
  - Domain*
  - Instalar os arquivos fornecidos no arquivo *ArquivosDoProjetoMvcApp.zip* e corrigir os namespaces:
    - Startup.Auth.cs* dentro do diretório *App\_Start*
    - SimpleInjectorInitializer* dentro do diretório *App\_Start*
    - Startup.cs* na raiz.
    - Adicionar os Controllers `AccountController`, `ManageController` e `UsuariosController`.
    - Adicionar as Views
    - Substituir o conteúdo da pasta *Shared* pelo conteúdo fornecido.
  - Não esqueça da string de conexão para colocar no *web.config*. Estou fornecendo um arquivo *web.config* para você localizar o local de se adicionar a tag `<connectionStrings>`  
Exemplos de *ConnectionsStrings*:

```
<connectionStrings>
```

```
<add name="DefaultConnection"
```

```
connectionString="Data
```

```
Source=(localdb)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\IdentityDb.mdf; Integrated Security=True"
```



Elaborado por: Paulo Maurício: [paulo.mauricio@prof.infnet.edu.br](mailto:paulo.mauricio@prof.infnet.edu.br)  
[professor@paulomau.eti.br](mailto:professor@paulomau.eti.br)

```
        providerName="System.Data.SqlClient" />
    </connectionStrings>

    <connectionStrings>
        <add name="DefaultConnection" connectionString="Data Source=.\sqlexpress2014;Initial
        Catalog=Users4IdentityDb;Integrated Security=True" providerName="System.Data.SqlClient" />
    </connectionStrings>
```

m. Acerte página \_Layout, adicionando o link do menu:

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

10- FINALIZADO COM SUCESSO