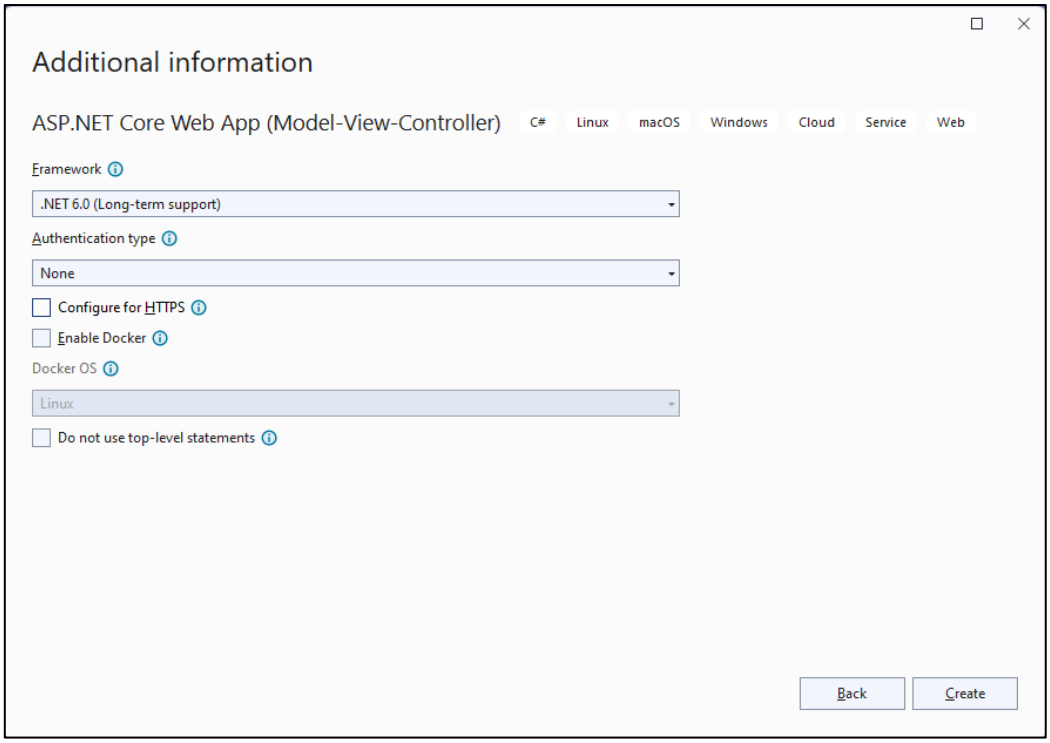
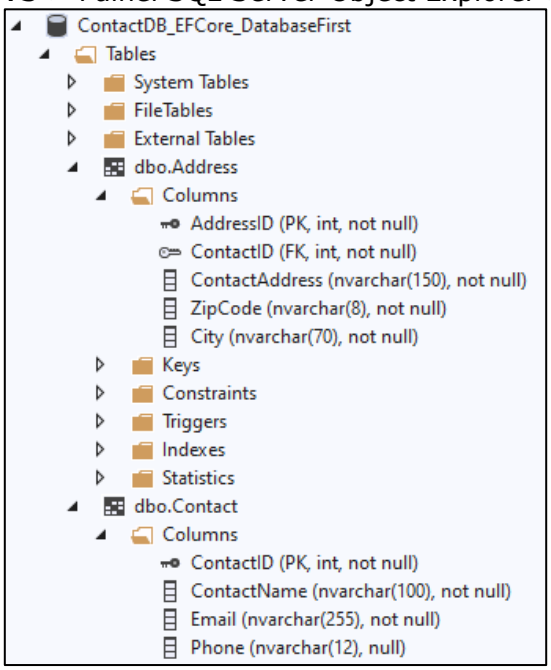
1. Criar um projeto novo

- ASP.NET Core Web Application (Model-View-Controller)

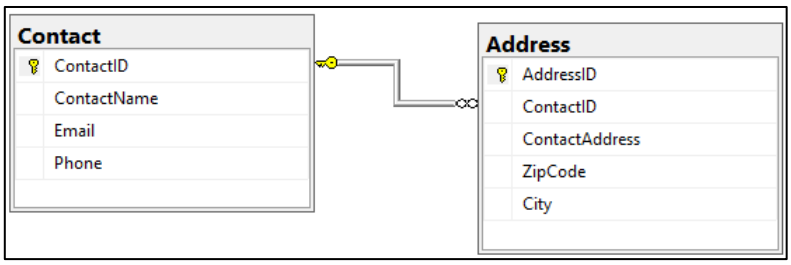


2. Criar a base dados

- VS > Abrir ContactDB_Script_All.sql > Connect ao servidor > Execute
- VS > Painel SQL Server Object Explorer > Conetar ao servidor > Verificar a base de dados



- Diagrama ER:

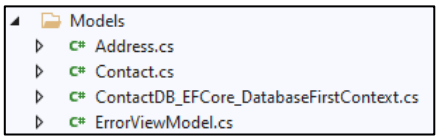


3. Instalar packages

- Abrir package manager console (PM)
```
Install-Package Microsoft.EntityFrameworkCore.SqlServer -Version 6.0.29
Install-Package Microsoft.EntityFrameworkCore.Tools -Version 6.0.29
Install-Package Microsoft.EntityFrameworkCore.Design -Version 6.0.29
```
- Verificar os packages instalados.

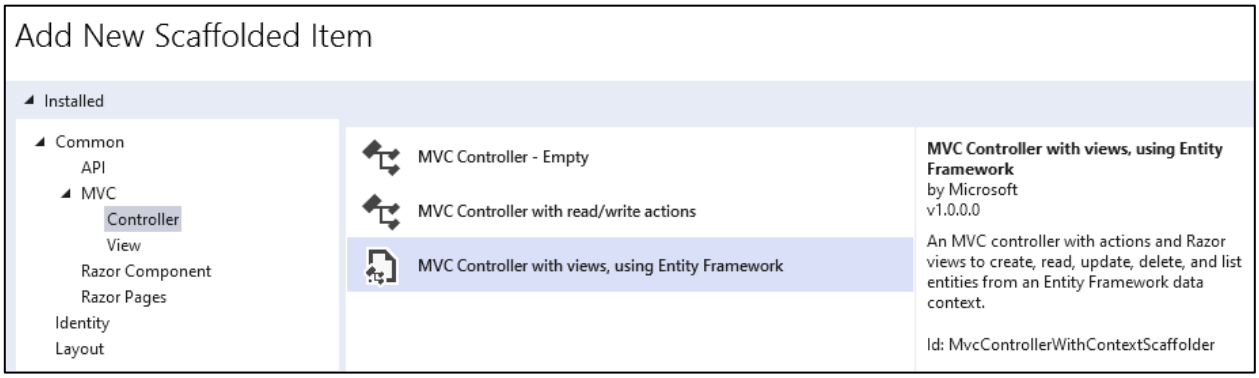
4. Fazer scaffolding à base de dados

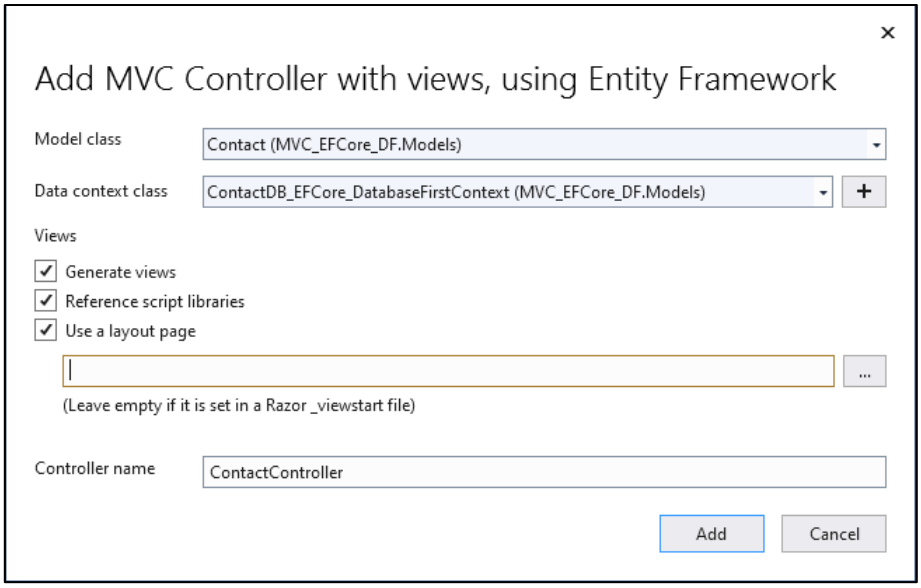
- Abrir package manager console (PM)
```
Scaffold-DbContext "Server=NomeServidor;Database= ContactDB_EFCore_DatabaseFirst;Trusted_Connection=True;"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```
- Verificar os modelos e a classe do contexto:



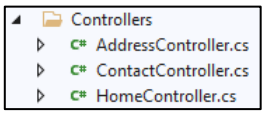
5. Criar os controladores para os modelos

- Controllers > Right click > Add > Controller > MVC Controller with views, using Entity Framework > Add

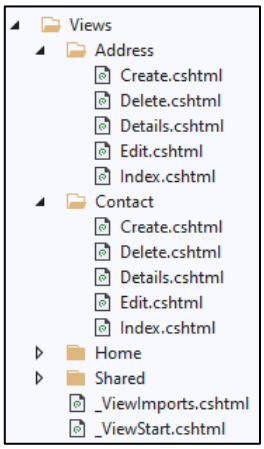




- Repetir a operação para todos os modelos
- Verificar os controladores:



- Verificar as vistas:



6. Alterar a localização da connection string

- ContactDB_EFCore_DatabaseFirstContext.cs > Comentar o conteúdo do método OnConfiguring
- Editar appsettings.json > Acrescentar a secção da connection string

```
"ConnectionStrings": {
    "ContactDB_EFCore_DatabaseFirstContext":
"Server=NomeServidor;Database=ContactDB_EFCore_DatabaseFirst;Trusted_Connection=True"
  }
```

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  // TODO MRS: acrescentar a connection string
  "ConnectionStrings": {
    "ContactDB_EFCore_DatabaseFirstContext": "Server=[redacted];Database=ContactDB_EFCore_DatabaseFirst;Trusted_Connection=True"
  }
}
```

7. Adicionar o serviço da EF

- Program.cs > Incluir as namespaces Microsoft.EntityFrameworkCore e a do modelo > Registar a connection string > Configurar a EF

```
// TODO MRS: ler connection string de appsettings.json
var connectionString = builder.Configuration.GetConnectionString("ContactDB_EFCore_DatabaseFirstContext");

// TODO MRS: configurar a EF
builder.Services.AddDbContext<ContactDB_EFCore_DatabaseFirstContext>(options =>
options.UseSqlServer(connectionString));
```

Program.cs

EF_DF

```
1  using EF_DF.Models;
2  using Microsoft.EntityFrameworkCore;
3  using System.Configuration;
4
5  var builder = WebApplication.CreateBuilder(args);
6
7  // Add services to the container.
8
9  // TODO MRS: ler connection string de appsettings.json
10 var connectionString = builder.Configuration.GetConnectionString("ContactDB_EFCore_DatabaseFirstContext");
11
12 // TODO MRS: configurar a EF
13 builder.Services.AddDbContext<ContactDB_EFCore_DatabaseFirstContext>(options => options.UseSqlServer(connectionString));
14
15 builder.Services.AddControllersWithViews();
16
17 var app = builder.Build();
18
19 // Configure the HTTP request pipeline.
20 if (!app.Environment.IsDevelopment())
21 {
22     app.UseExceptionHandler("/Home/Error");
23 }
24 app.UseStaticFiles();
25
26 app.UseRouting();
27
28 app.UseAuthorization();
29
30 app.MapControllerRoute(
31     name: "default",
32     pattern: "{controller=Home}/{action=Index}/{id?}");
33
34 app.Run();
```

8. Alterar o layout

- _Layout.cshtml > Incluir os links na navigation bar para os novos controladores

```
<li class="nav-item">
   <a class="nav-link text-dark" asp-area="" asp-controller="Contact" asp-action="Index">Contact</a>
</li>
<li class="nav-item">
   <a class="nav-link text-dark" asp-area="" asp-controller="Address" asp-action="Index">Address</a>
</li>
```

```
<!--TODO MRS: acrescentar links à nav bar-->
<li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-controller="Contact" asp-action="Index">Contacts</a>
</li>
<li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-controller="Address" asp-action="Index">Addresses</a>
</li>
```

9. Testar a aplicação

- Fazer os ajustes nas views que considerar pertinentes: