

# Criando projetos (classlib e webapi) adicionando suas referências e conectando a *Solution* (SLN)

Criador por **Ellison W. M. Guimarães**

**LinkedIn:** <https://www.linkedin.com/in/ellisonguimaraes/>

**E-mail:** [ellison.guimaraes@gmail.com](mailto:ellison.guimaraes@gmail.com)

Quando criamos uma *Solution* (SLN) com vários projetos dentro, sejam eles *webapi*, *classlib*, etc, é necessários informar ao *solution* que existem esses projetos. Além disso, precisamos adicionar as referências de um projeto a outros.

Para exemplificar, iremos criar inicialmente uma *Solution*:

```
1 dotnet new sln --name "ProjectSLN"
```

Também criaremos mais quatro projetos nesta mesma pasta, três deles são *classlib* e um *webapi*:

```
1 dotnet new classlib -o "Project.Domain"
2 dotnet new classlib -o "Project.Interface"
3 dotnet new classlib -o "Project.Repository"
4 dotnet new webapi -o "Project.API"
```

Após criamos a *solution* e os projetos o nosso diretório fica da seguinte forma:

```
1 |──ProjectSLN.sln
2 |
3 |──Project.API
4 | |   appsettings.Development.json
5 | |   appsettings.json
6 | |   Program.cs
7 | |   Project.API.csproj
8 | |   Startup.cs
9 | |   WeatherForecast.cs
10 | |
11 | |   └──Controllers
12 | |       WeatherForecastController.cs
13 | |
14 | |   └──Properties
15 | |       launchSettings.json
16 | |
17 |──Project.Domain
18 | |   Class1.cs
19 | |   Project.Domain.csproj
20 | |
21 |──Project.Interface
22 | |   Class1.cs
23 | |   Project.Interface.csproj
24 | |
25 |──Project.Repository
```

```
26 | Class1.cs
27 | Project.Repository.csproj
```

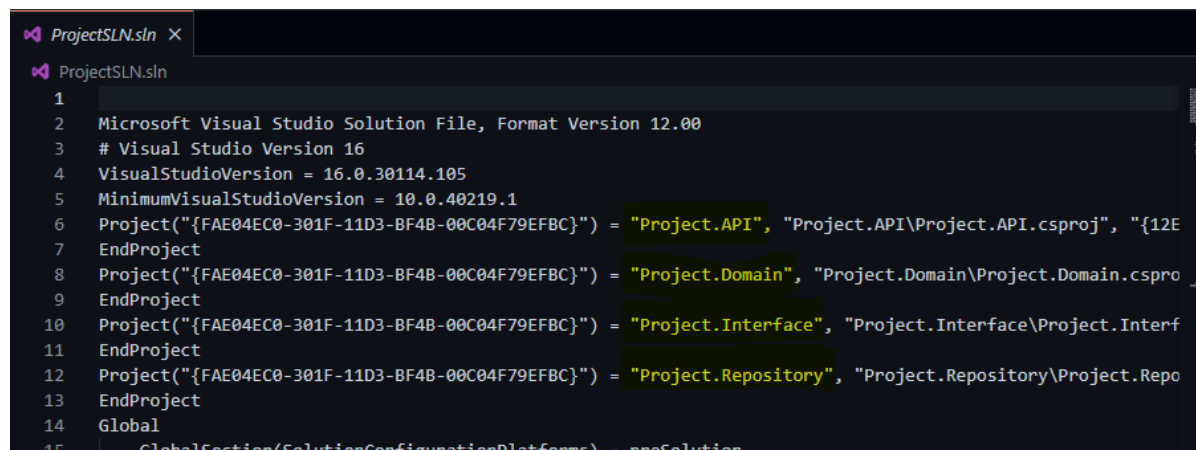
Agora precisamos fazer as conexões entre os projetos, que até então, estão totalmente independentes.

## Configurando os projetos na *solution* (SLN)

Para adicionar os projetos a *solution* usando o comando `dotnet sln add`, onde após o `add` colocamos os respectivos arquivos `.csproj` de cada projeto. Abaixo vamos adicionar os projetos:

```
1 dotnet sln add .\Project.API\Project.API.csproj
2 dotnet sln add .\Project.Domain\Project.Domain.csproj
3 dotnet sln add .\Project.Interface\Project.Interface.csproj
4 dotnet sln add .\Project.Repository\Project.Repository.csproj
```

A partir deste momento podemos verificar que no arquivo *sln* estão incluídos:



```
ProjectSLN.sln
ProjectSLN.sln
1
2 Microsoft Visual Studio Solution File, Format Version 12.00
3 # Visual Studio Version 16
4 VisualStudioVersion = 16.0.30114.105
5 MinimumVisualStudioVersion = 10.0.40219.1
6 Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Project.API", "Project.API\Project.API.csproj", "{12E
7 EndProject
8 Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Project.Domain", "Project.Domain\Project.Domain.csproj",
9 EndProject
10 Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Project.Interface", "Project.Interface\Project.Interface.csproj",
11 EndProject
12 Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "Project.Repository", "Project.Repository\Project.Repository.csproj",
13 EndProject
14 Global
15 GlobalSection(SolutionConfigurationPlatforms) = preSolution
```

## Referenciando projetos a outros

Cada um dos projetos precisam referenciar outros, e para isso utilizamos o comando `dotnet add reference [file.csproj]` também referenciando o arquivo `csproj` dos outros projetos, por exemplo:

- O projeto `Project.Interface` precisa da referência para o projeto `Project.Domain`, precisamos então ir até o diretório de `Project.Interface` e usar o comando:

```
1 dotnet add reference ..\Project.Domain\Project.Domain.csproj
```

- O projeto `Project.Repository` precisa da referência para o projeto `Project.Domain` e `Project.Interface`, precisamos então ir até o diretório de `Project.Repository` e usar os comandos:

```
1 dotnet add reference ..\Project.Domain\Project.Domain.csproj
2 dotnet add reference ..\Project.Interface\Project.Interface.csproj
```

- O projeto `Project.API` precisa da referência para todos os outros três projetos, logo, precisamos ir até o diretório do `Project.API` e usar todos os comandos:

```
1 dotnet add reference ..\Project.Domain\Project.Domain.csproj
2 dotnet add reference ..\Project.Interface\Project.Interface.csproj
3 dotnet add reference ..\Project.Repository\Project.Repository.csproj
```

É possível visualizar as referências de um projeto usando o comando `dotnet list reference`. Veja o exemplo do comando no projeto `Project.API`:

```
ellis@ELLISON ~ > Desktop > Project > Project.API
> dotnet list reference
Referências de projeto
-----
.. \Project.Domain\Project.Domain.csproj
.. \Project.Interface\Project.Interface.csproj
.. \Project.Repository\Project.Repository.csproj
ellis@ELLISON ~ > Desktop > Project > Project.API
>
```

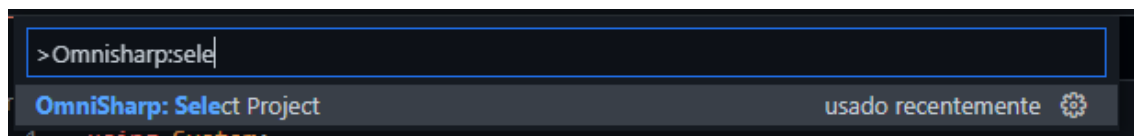
Não confunda com `dotnet list package`, que mostram os pacotes:

```
ellis@ELLISON ~ > Desktop > Project > Project.API
> dotnet list package
O projeto 'Project.API' tem as seguintes referências de pacote
[net5.0]:
Pacote de Nível Superior      Solicitado  Resolvido
> Swashbuckle.AspNetCore      5.6.3       5.6.3
```

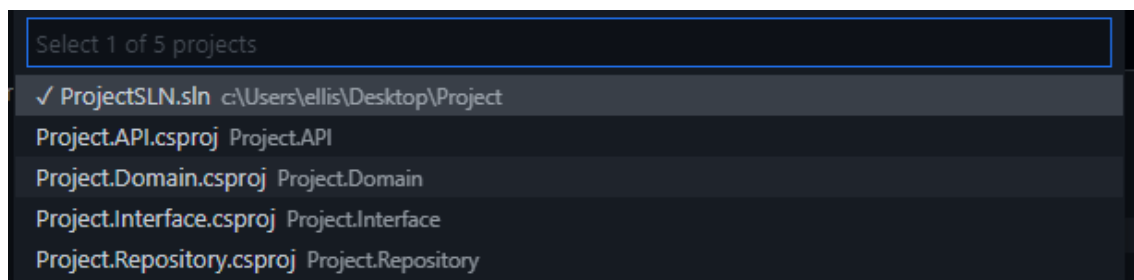
## Selecionando o projeto no VSCode para uso do Intellisense

Para selecionar o arquivo `csproj` no Visual Studio Code, é necessário:

- Acessar o menu com `Ctrl + Shift + P`;
- Digitar por `OmniSharp: Select Project`:



- Selecionar o projeto que deseja utilizar o *intellisense*:



Podemos utilizar o *Intellisense* em um projeto em específico ou podemos utilizar a *solution* geral que engloba todos os projetos, no qual é o mais recomendado.

