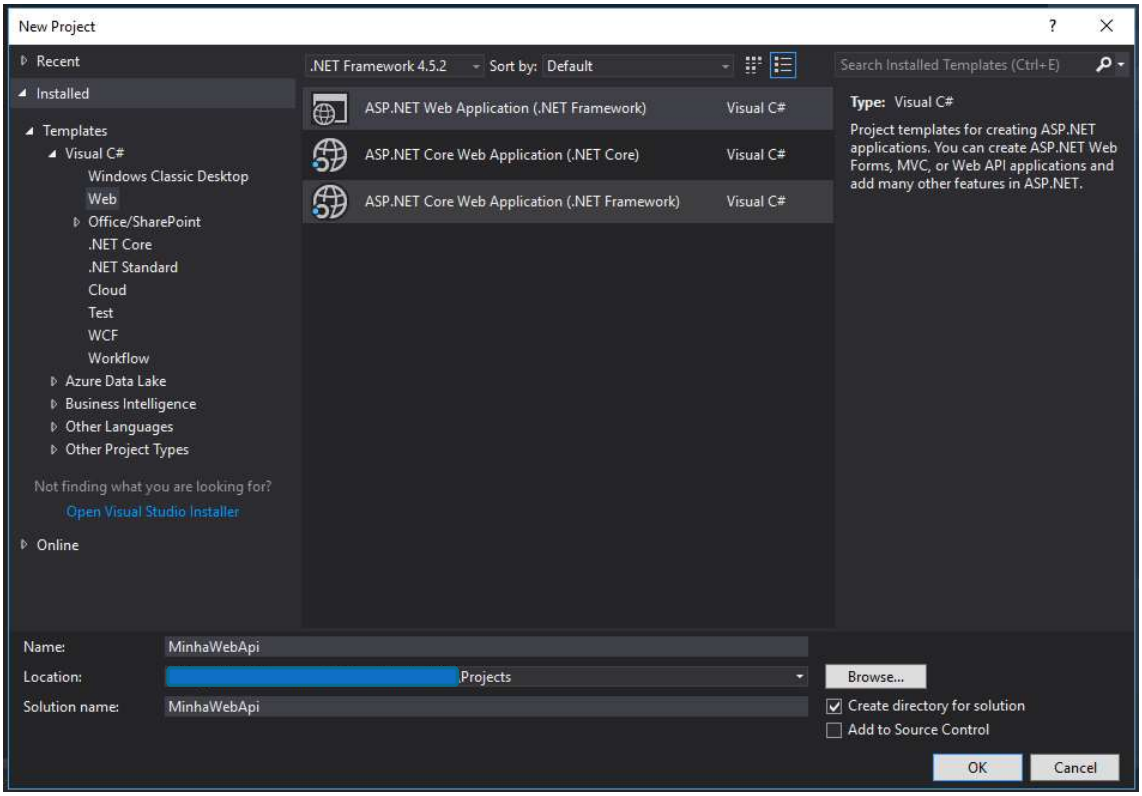
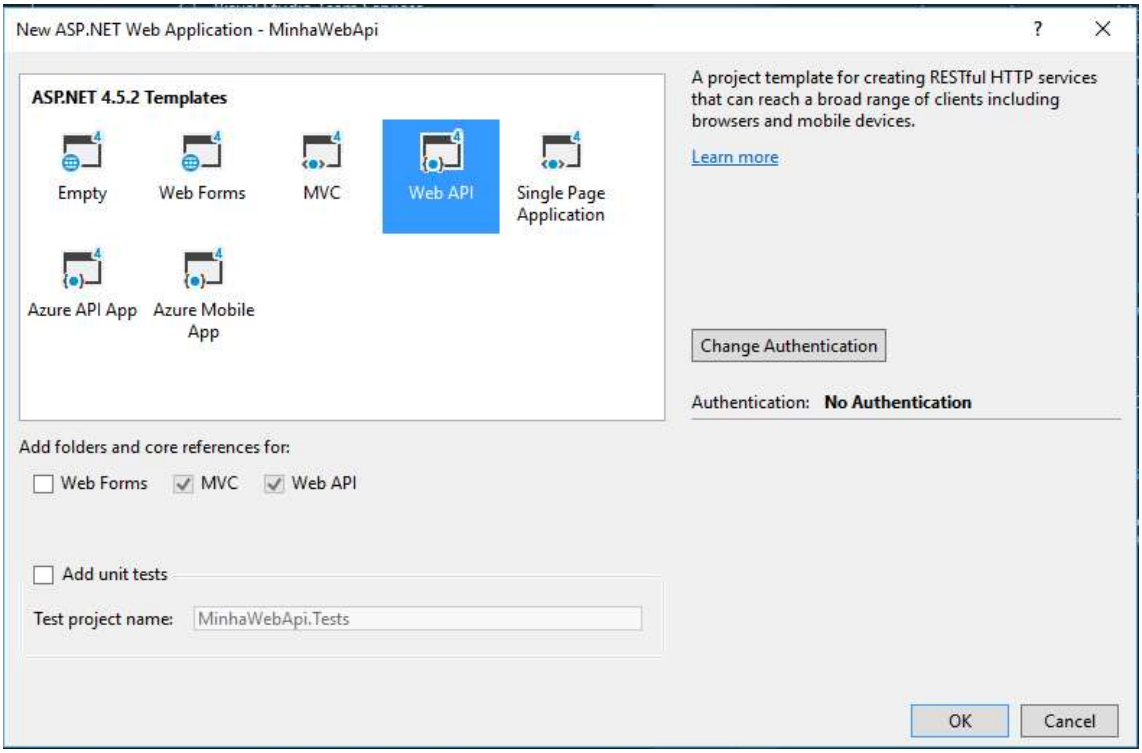
Ferramentas: Internet, Conta no Azure com Azure AD (pode ser a versão free), Visual Studio 2017, SharePoint Online

## 1.Criar e Configurar uma webApi autenticando o Azure AD

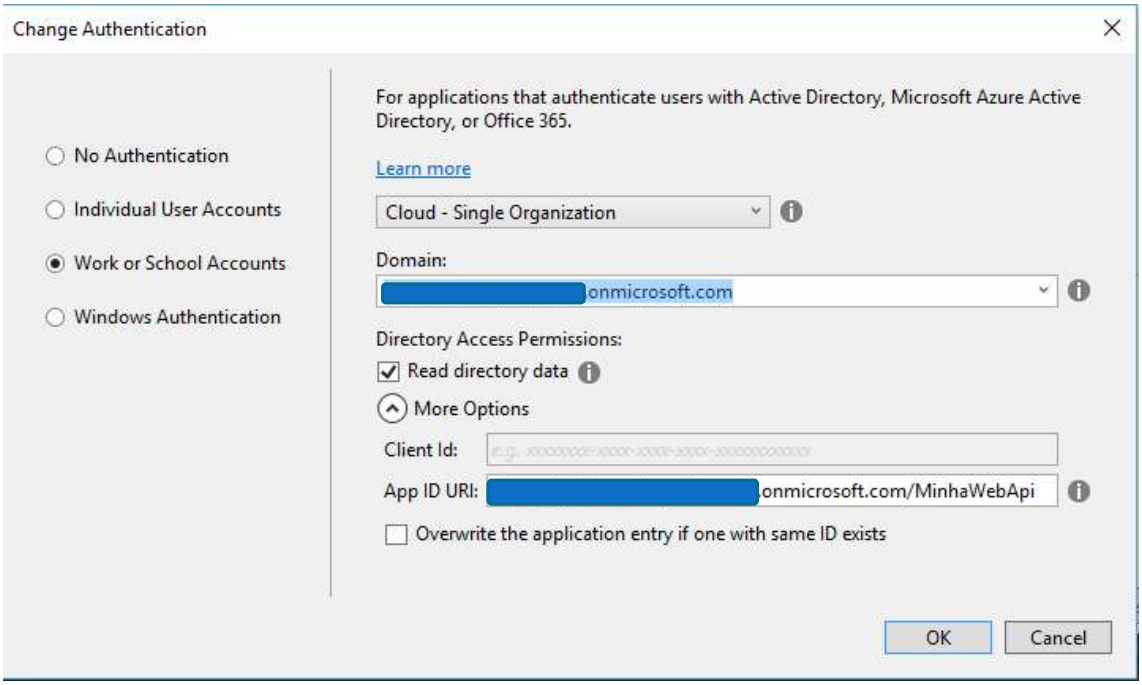
- Crie um novo projeto do tipo web



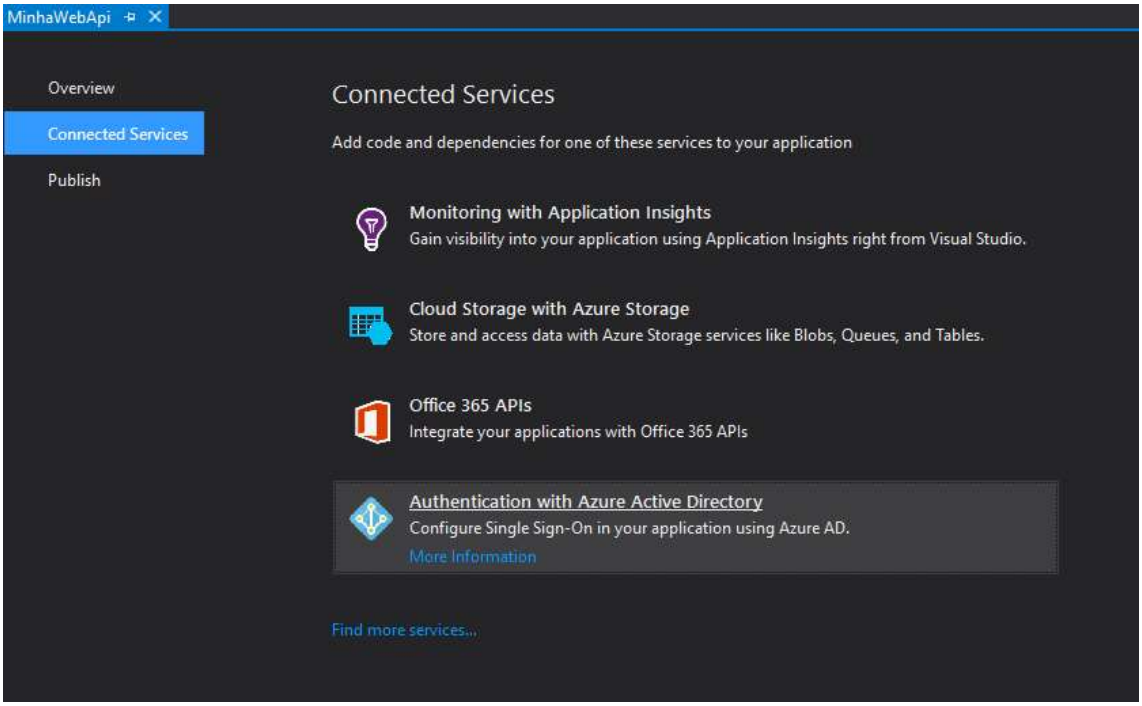

- Selecione WebApi e clique em Change Authentication



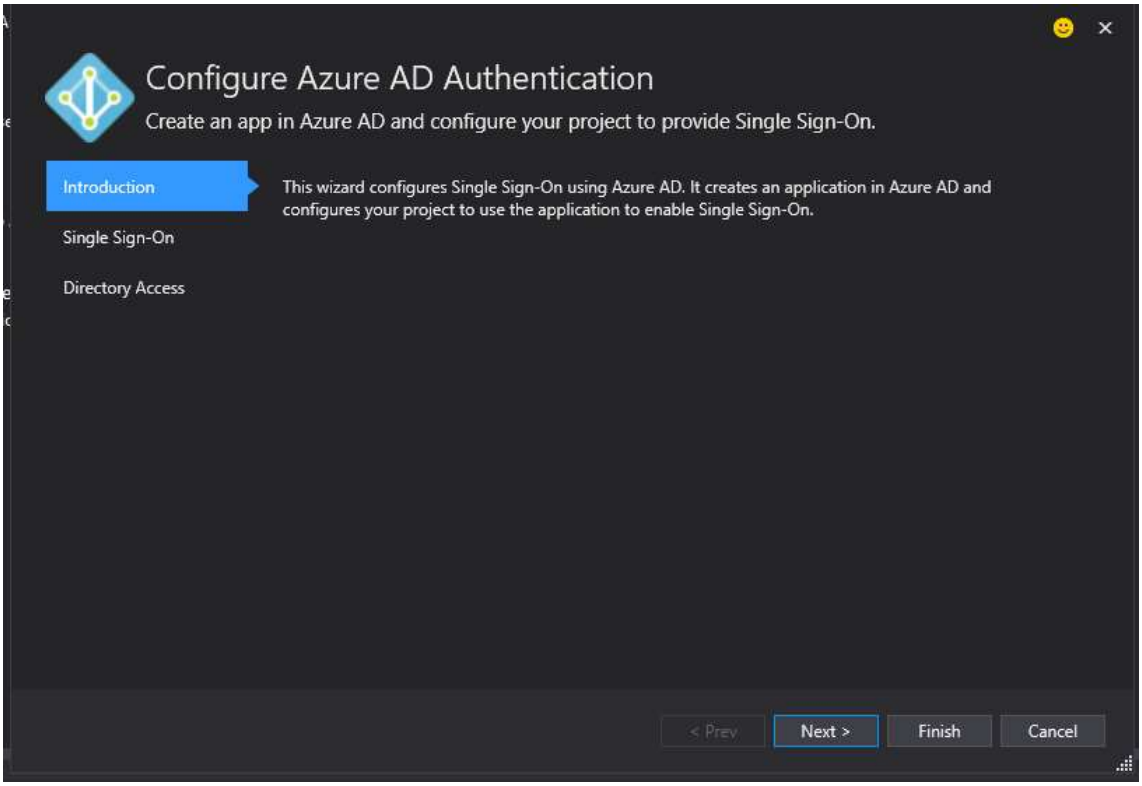
- Selecione o seu domínio do azure, marque a opção de Read directory data



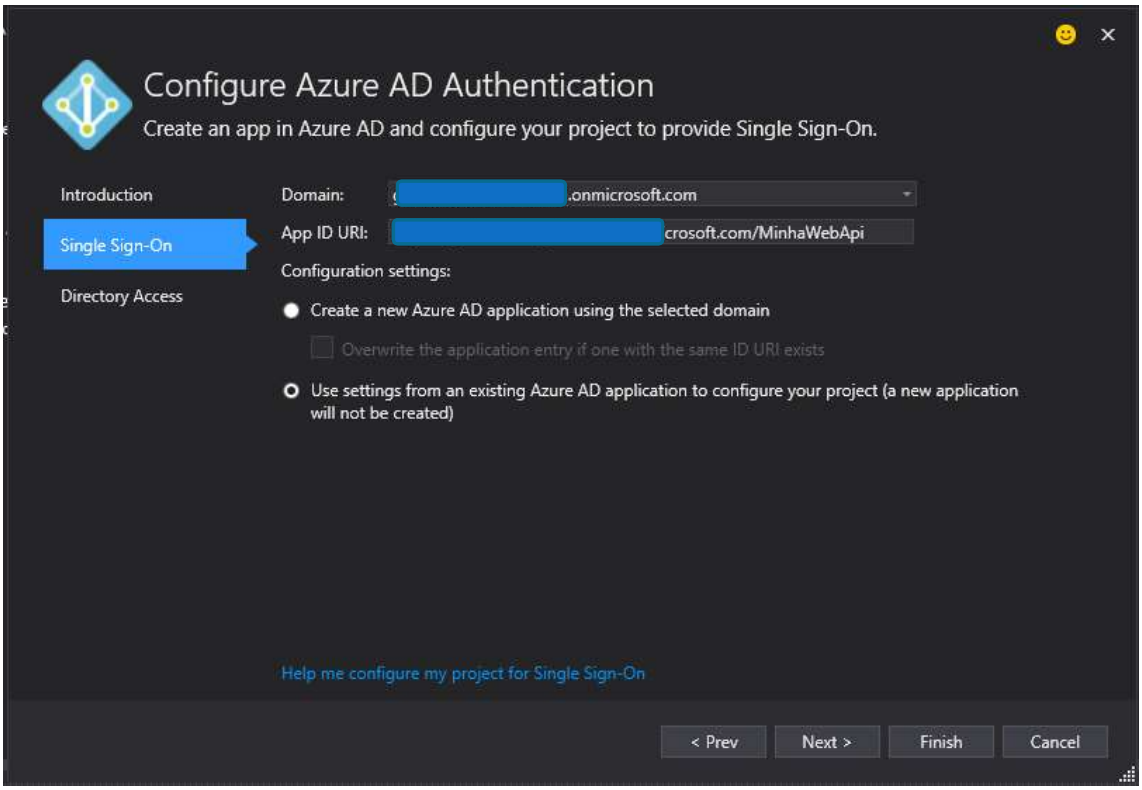

- Após criar o projeto, selecione, na tela inicial, "Connected Services" e na sequência "Authentication with Azure Active Directory"



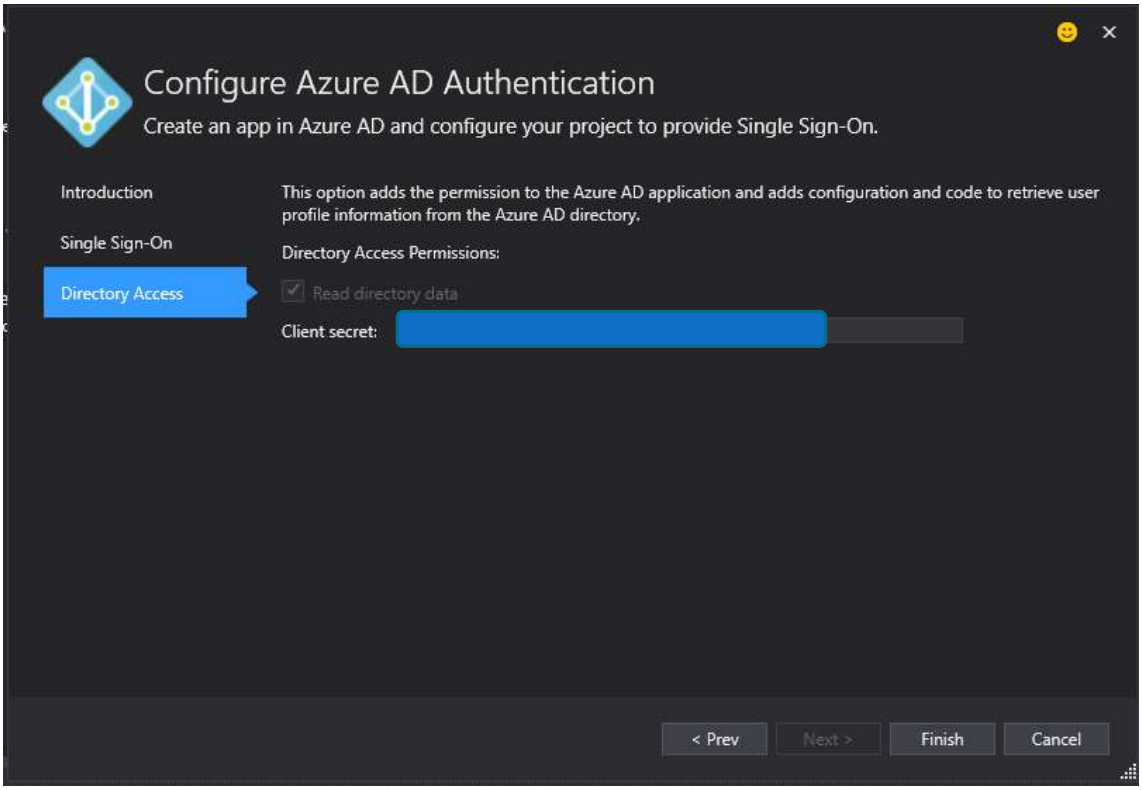
- Selecione Next



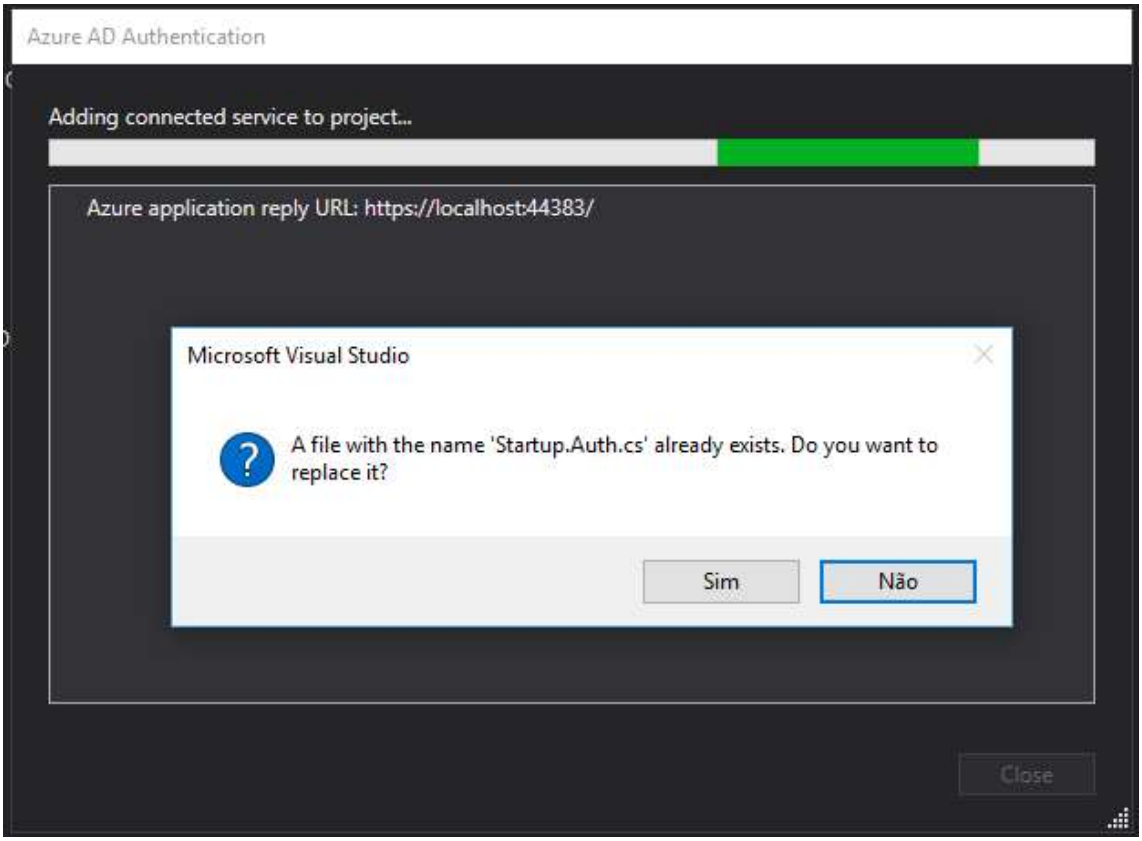

- Selecione Next



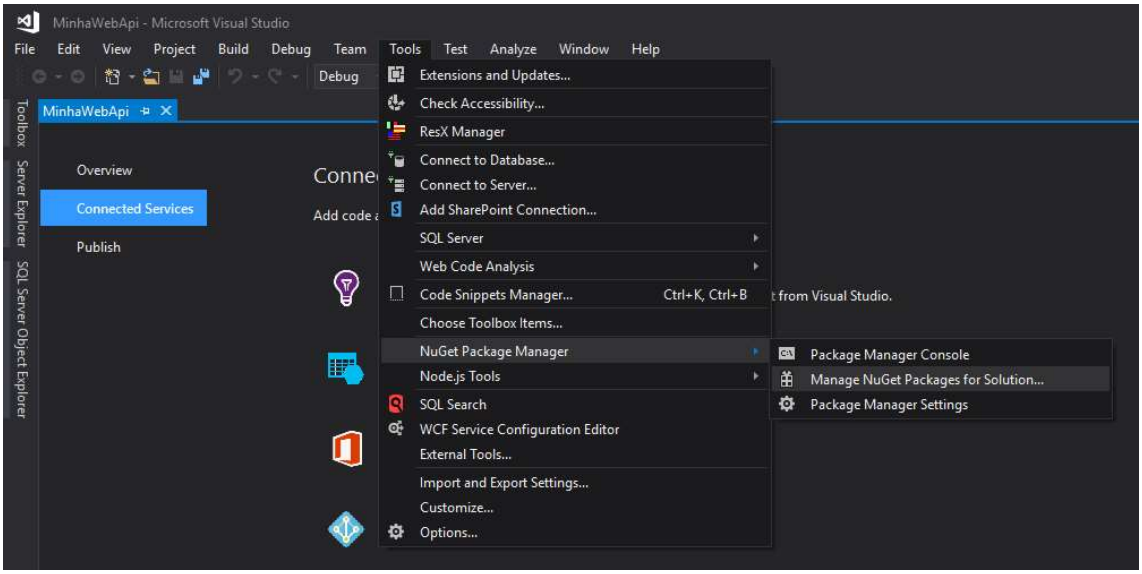
- Selecione Finish



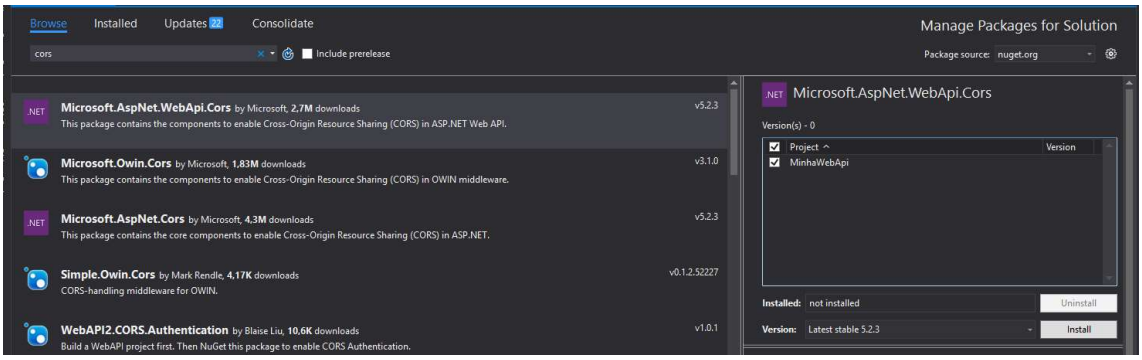

- Selecione Sim



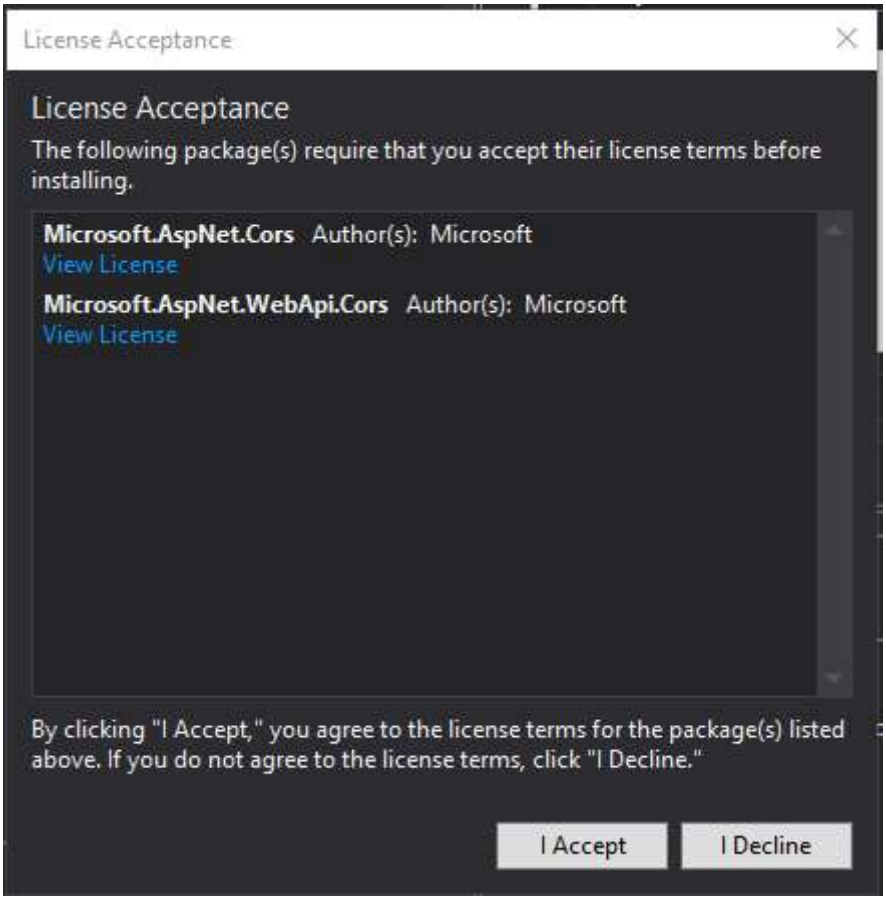
- Abra o gerenciador de pacotes do Nuget



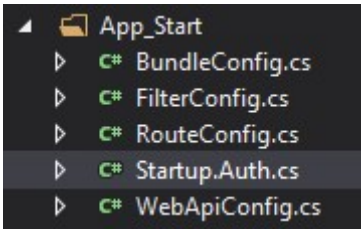

- Em Browse, busque por "cors", selecione a opção "Microsoft.AspNet.WebApi.Cors", marque o seu projeto e clique em instalar (Install) e aguarde.



- Ao aparecer a tela de aceite de licenças, clique em aceitar e espere o fim da intalação do pacote.




- Após concluido o processo de instalação do pacote, abra o arquivo WebApiConfig.cs no diretório App_Start da sua solution




- Adicione as seguintes linhas no começo do método register:

  var cors = new EnableCorsAttribute("<SUA URL QUE ACESSARÁ A WEBAPI>", "Accept , Authorization, OutroHeaderCustomizado", "GET,POST,PUT,OPTIONS,DELETE");

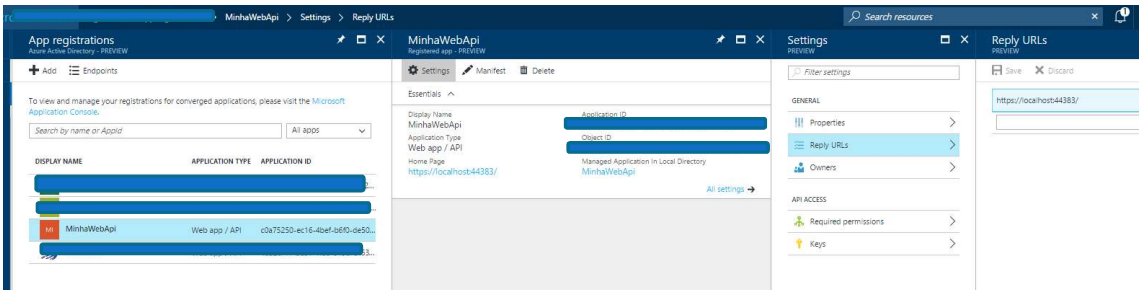
  config.EnableCors(cors);

  O accept no header é importante pra funcionar nas chamadas originadas no Edge e no Internet Explorer.

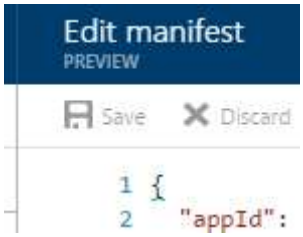
  O Authorization é o header onde será passado o token de autorização

- Corrija as referências para que seja adicionado o CORS nessa classe.


- Não é necessário adicionar as configurações de CORS no web.config
- A configuração do CORS no Register é global, ela atende a toda a aplicação. Pode-se adicionar um atributo de cors em uma classe ou método.
  A sequência de verificação da configuração é:
  - Primeiro no método, se não houver, na classe, se não houver vale a configuração global.
- Basta colocar o atributo [Authorize] (classe ou método) onde você deseja que o acesso seja permitido somente a chamadas autenticadas.
- Você pode publicar ou rodar seu webapi de qualquer servidor (localhost do seu VS, publicado em um servidor on-premise ou ainda no azure).
- Caso a url do seu webapi mude do localhost do visual studio para o local de publicação, é necessário entrar nas configurações de aplicativos registrados do azure ad (https://portal.azure.com) e modificar ou adicionar o novo endereço ao reply url.

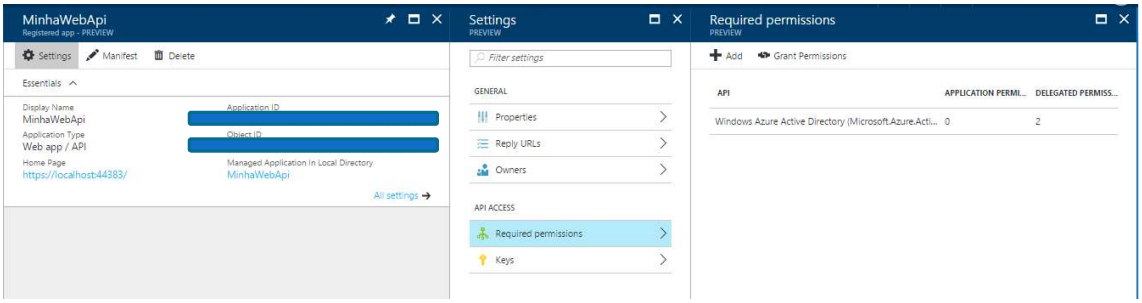



- Outra configuração necessária é modificar o manifest da aplicação. Na mesma tela de app registrations do azure, selecione a aplicação e depois clique em manifest. Encontre a opção oauth2AllowImplicitFlow e mude o valor de false para true 

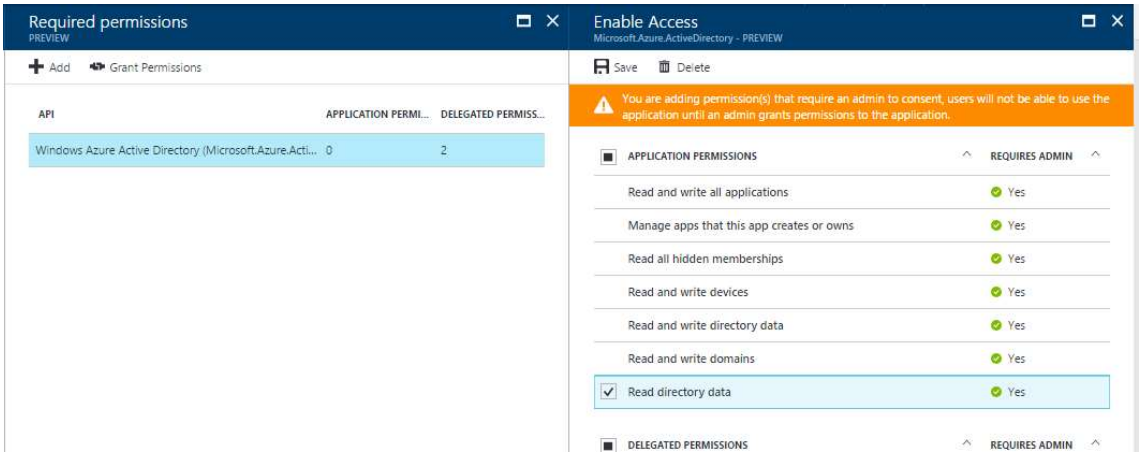
e clique em Save. 


- É necessário autorizar os usuários do AD também, sem pedir a autorização explicita. Para isso, na mesma tela onde tem o manifest,

selecionar Settings, e Required Permissions.




Selecione o Windows Azure Active Directory, marque a opção Read Directory data e depois salve a alteração.



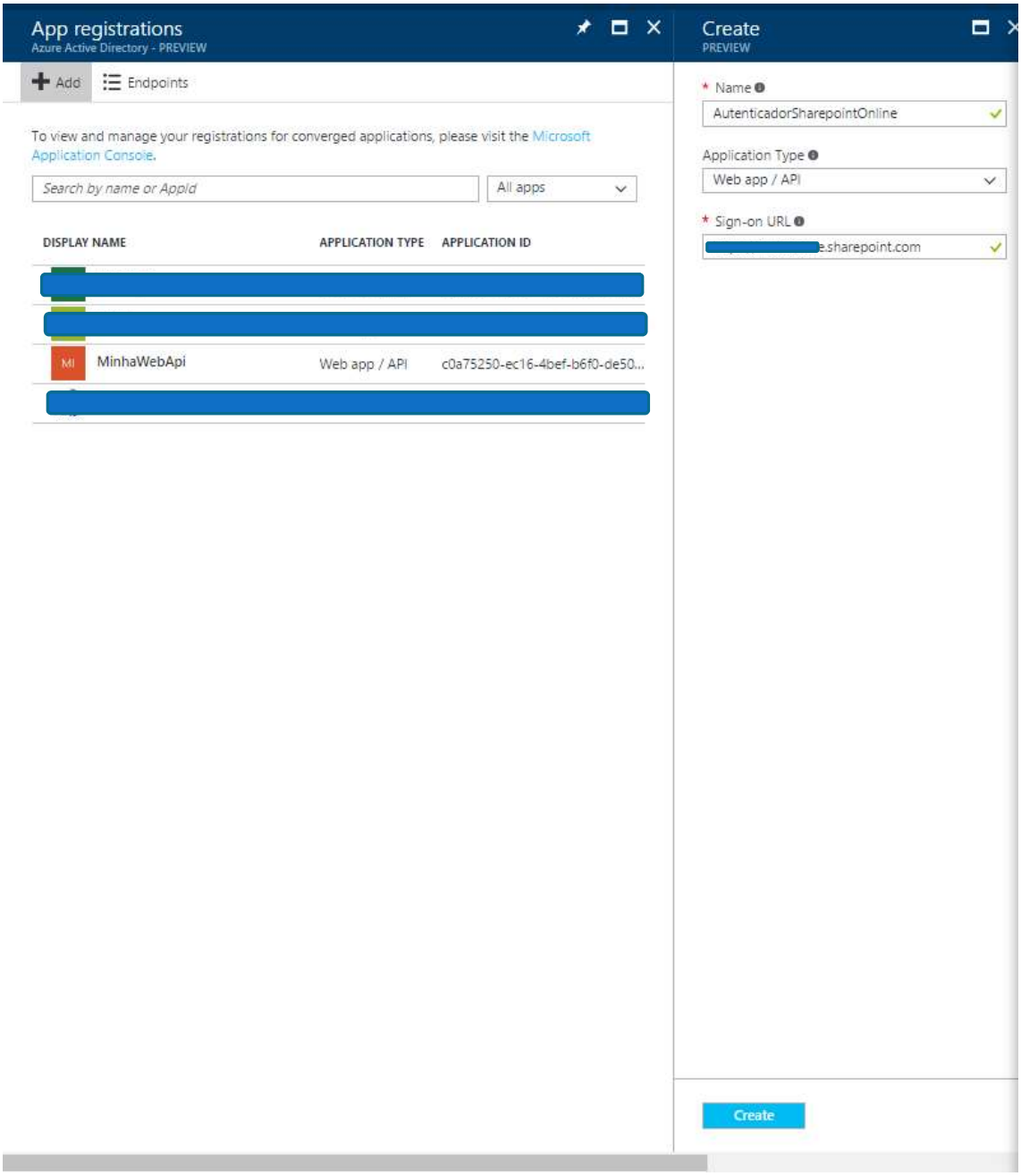

Na sequencia clique em Grant Permissions, e clique em Yes


A web api está pronta para ser usada.

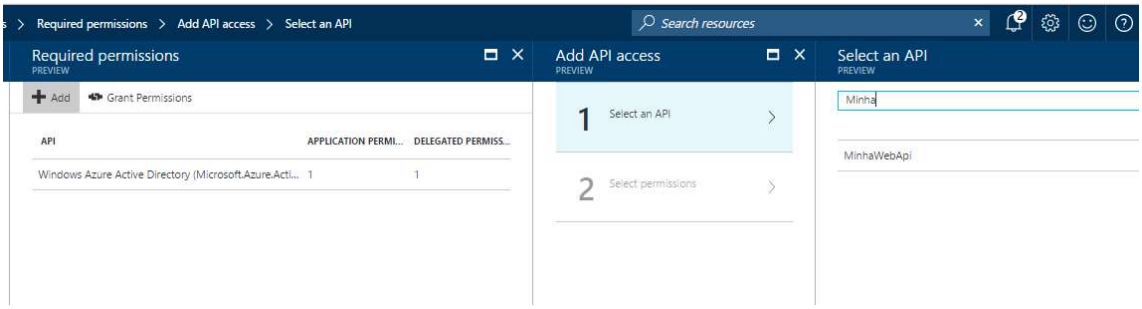
## 2. Aplicação intermediária para autenticar a partir do SharePoint online.

Essa etapa é toda realizada no azure AD.

- No azure Ad, selecione App Registrations, clique em Add e preencha os dados, sendo que no campo Sign-on URL deve ser a url do Sharepoint

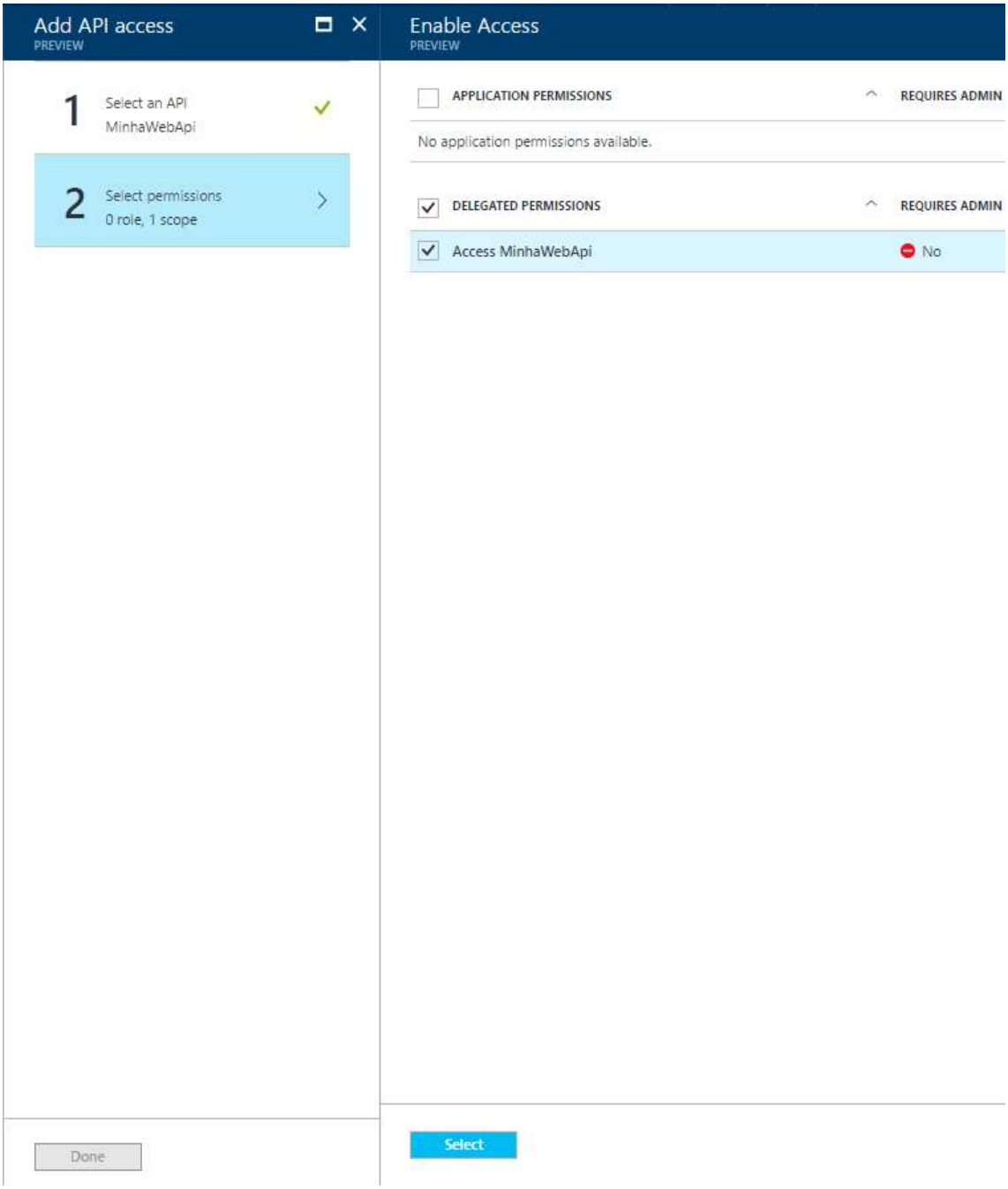



- Altere o manifest de forma igual ao do WebApi, modificando o parâmetro oauth2AllowImplicitFlow de false para true
- Em Required Permissions, faça igual ao WebApi, autorizando Read directory data e salve
- Ainda em required permissions, clique em Add, Select an API, e na barra de busca digite o nome da sua webApi

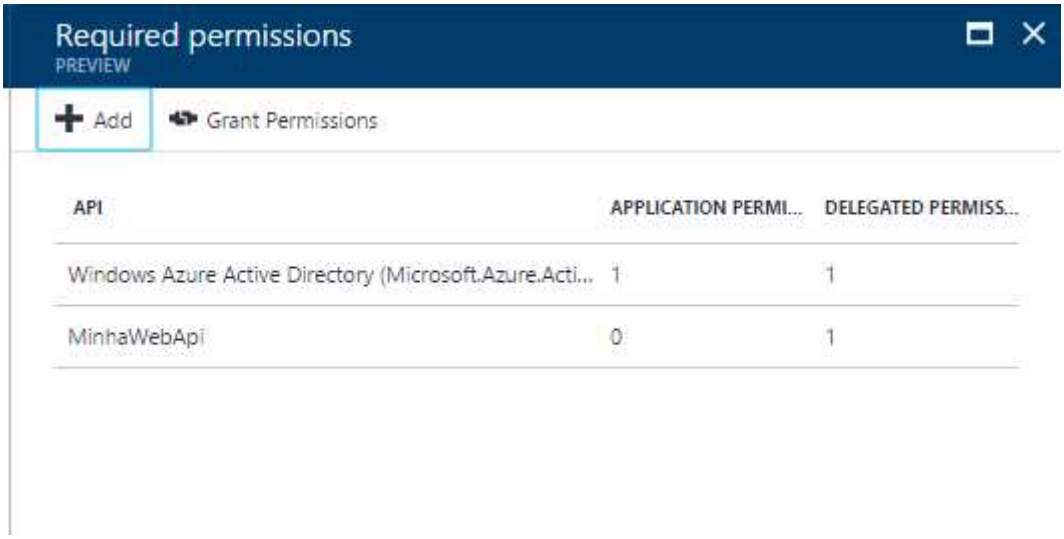



Selecione a api listada, clique em Select na parte inferior.

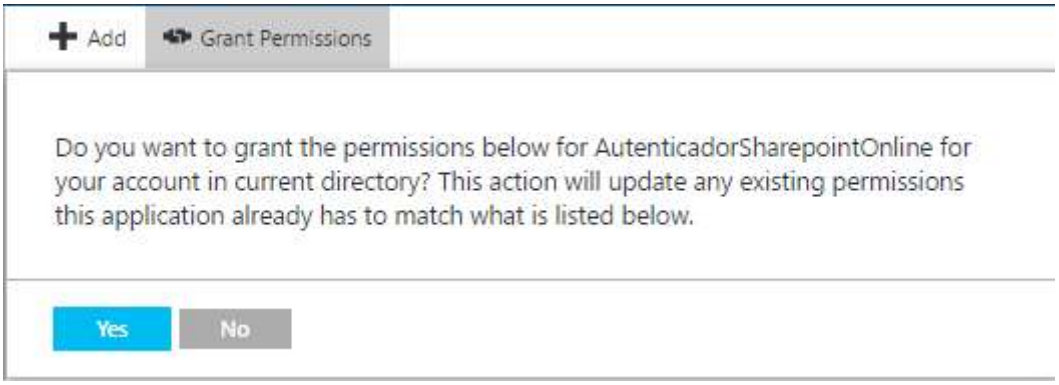
- Na segunda etapa marque em Access MinhaWebApi e clique em Select e na sequencia em Done



- Ainda na tela de Required Permissions




clique em Grant Permissions e depois em Yes

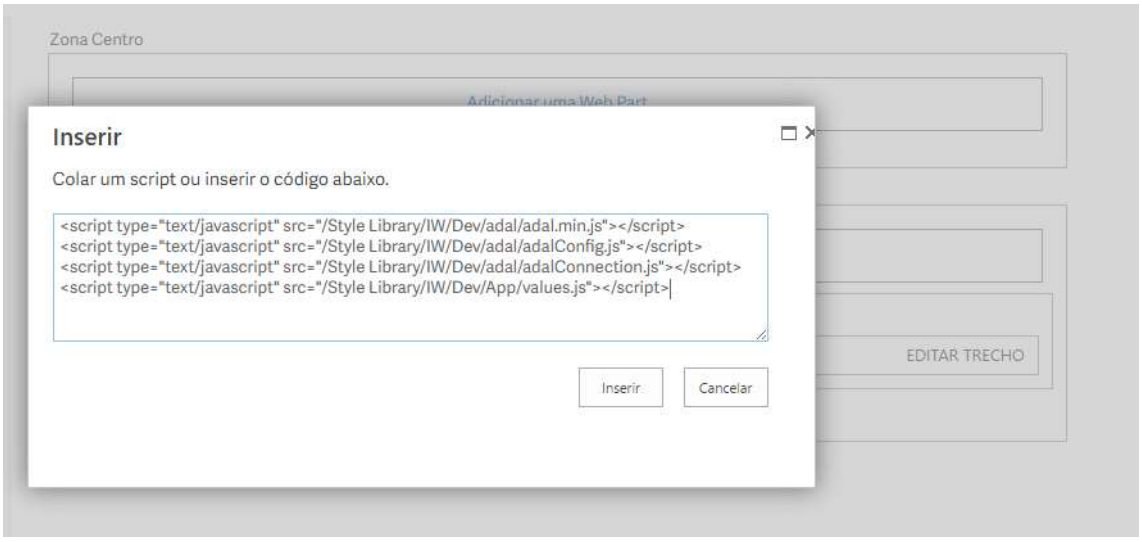



A sua aplicação intermediária está configurada. Agora é utilizar a biblioteca adal.js no lado do SharePoint para chamar a webapi de forma autenticada.

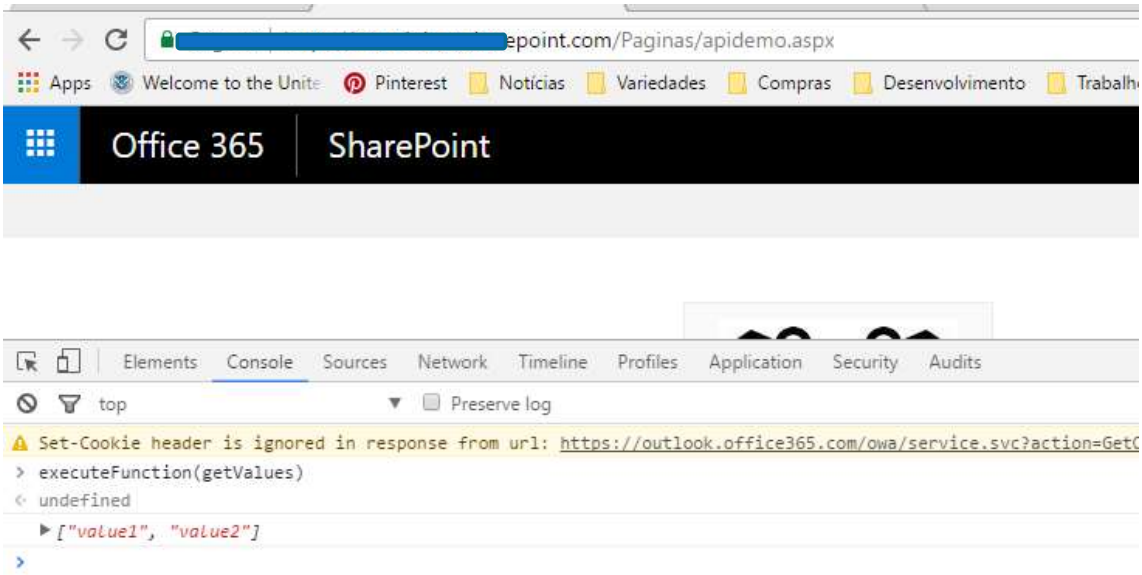
### 3. Uso da biblioteca adal.js no sharepoint

O exemplo a seguir será demonstrado em um ambiente onde foi implementado uma webapi igual a demonstrada nesse tutorial e utilizando uma aplicação intermediária similar a demonstrada nesse tutorial.

- No Sharepoint foi criada uma página para adicionar os scripts para exemplificar. Nessa página foi adicionada uma webpart Editor de Scripts




  - A biblioteca adal.min.js é disponibilizada no github e você pode acessar ao site para obter mais informações https://github.com/AzureAD/azure-activedirectory-library-for-js
  - O arquivo adalConfig.js e adalConnection.js foi feito com base em alguns exemplos obtidos na internet (geralmente nos exemplos é utilizado angular, por este motivo fiz o tutorial, pois não encontrei nenhum sem a utilização do angular com essa biblioteca do adal.js)
  - O arquivo values.js são métodos simples com chamadas ajax ao webapi, no fim do arquivo tem as chamadas as functions implementadas no próprio arquivo
  - Os arquivos você pode obter nesse endereço do github <meuprojeto>
- Chamada executada do SharePoint



- adalConfig.js

  Nesse arquivo tem todos os parâmetros necessários para obter a autorização e chegar aos métodos do webapi. A configuração deve ser feita apenas uma vez. Cada parâmetro está com um comentário do que deve ser colocado.

- adalConnection.js
  Nesse arquivo tem o método que gera o token de autorização para a chamada autenticada e chama a function desejada com o token obtido.

  Para executar uma function que faz uma requisição a WebApi basta chamar da seguinte forma:

  - Sem parâmetros: executeFunction(nomedaFunction);
  - Com um parâmetro: executeFunction(nomedaFunction, 1);
  - Com vários parâmetros: executeFunction(nomedaFunction, [1,2,{nome: 'fulano', telefone: '12345-1234'}])