

Professor Orientador Projeto Final: Rômulo C. Silvestre

# Chamando a API

Usando Blazor e a extensão Newtonsoft.JSON

Usando o Visual Studio 2022

# Tutorial

- Esse tutorial utilizou o Visual Studio Community 2022
- Você pode usar também o VS CODE ou seu editor preferido
- Tutorial de apoio ao desenvolvimento do projeto final e trabalho complementar.

Chamando a **API REST** no aplicativo Blazor Server

## Web e Nuvem (4)



## ASP.NET e desenvolvimento Web

Crie aplicativos Web usando ASP.NET Core, ASP.NET, HTML/JavaScript e Contêineres, incluindo suporte a Doc...



## Desenvolvimento para Azure

SDKs, ferramentas e projetos do Azure para desenvolver aplicativos na nuvem e criar recursos usando o .NET e o ...



## Desenvolvimento Python

Edição, depuração, desenvolvimento interativo e controle do código-fonte para Python.



## Desenvolvimento em node.js

Crie aplicativos de rede escalonáveis usando Node.js, um tempo de execução do JavaScript conduzido por evento...



## Área de trabalho e Dispositivos móveis (5)



## Desenvolvimento de .NET Multi-Platform App UI

Crie aplicativos Android, iOS, Windows e Mac a partir de uma única base de código usando C# com .NET MAUI.



## Desenvolvimento para desktop com .NET

Crie WPF, Windows Forms e aplicativos de console usando C#, Visual Basic e F# com .NET e .NET Framework.



## Desenvolvimento para desktop com C++

Crie aplicativos C++ modernos para Windows usando as ferramentas de sua escolha, incluindo MSVC, Clang, CMa...



## Desenvolvimento com a Plataforma Universal do Wind...

Crie aplicativos para a Plataforma Universal do Windows usando C#, VB, JavaScript ou, como alternativa, C++.



## Detalhes da instalação

▸ Editor de núcleo do Visual Studio

▸ Desenvolvimento para desktop co... •

▸ ASP.NET e desenvolvimento Web •

▾ Processamento e armazenamentos... •

▾ Opcional

☒ SQL Server Data Tools☒ Ferramentas do Azure Data Lake e do Stre...☒ Ferramentas de desenvolvimento do .NET...☐ Suporte à linguagem F# da área de trabalho

## Localização

C:\Program Files\Microsoft Visual Studio\2022\Community

Ao continuar, você concorda com a [licença](#) da edição do Visual Studio selecionada. Também oferecemos a possibilidade de baixar outro software com o Visual Studio. Esse software é licenciado separadamente, conforme [Avisos de Terceiros](#) ou na licença que o acompanha. Ao continuar, você também concorda com essas licenças.

Espaço total necessário 2,58 GB

Instalar durante o download ▾

Modificar

## Outros conjuntos de ferramentas (5)



### Processamento e armazenamentos de dados

Conecte-se, desenvolva e teste as soluções de dados usando SQL Server, Azure Data Lake ou Hadoop.



### Aplicativos de ciência de dados e análise

Linguagens e ferramentas para criação de aplicativos de ciência de dados, incluindo Python e F#.



### Desenvolvimento de extensão do Visual Studio

Crie suplementos e extensões para Visual Studio, incluindo novos comandos, analisadores de código e jan...



### Desenvolvimento para Office/SharePoint

Crie complementos do Office e do SharePoint, soluções do SharePoint e complementos do VSTO usando C#, VB...

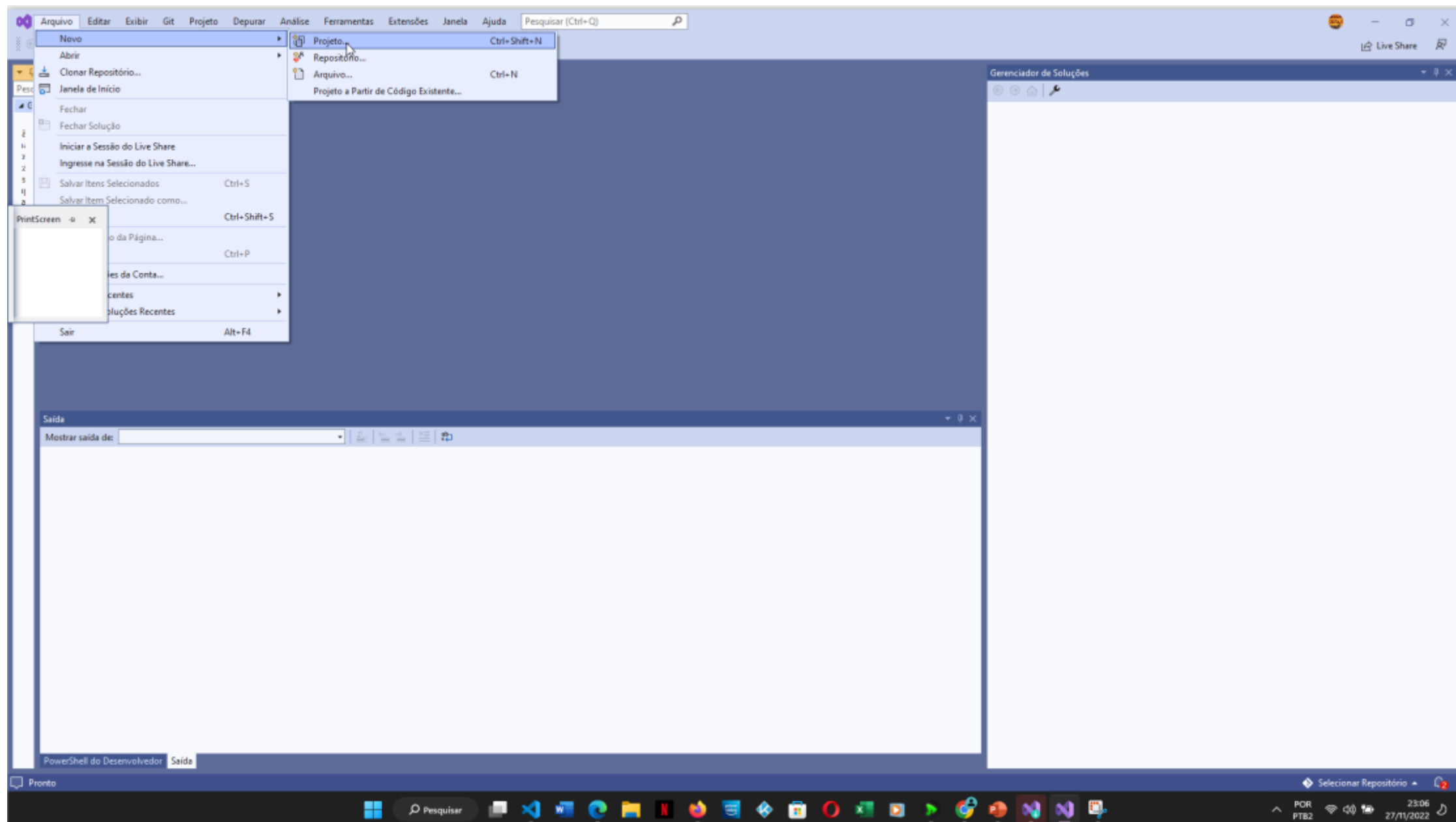


### Linux e desenvolvimento inserido com C++

Crie e depure aplicativos em execução em um ambiente Linux ou em um dispositivo inserido.



Localização



# Configurar seu novo projeto

Aplicativo Blazor Server

C#

Linux

macOS

Windows

Blazor

Nuvem

Web

Nome do projeto

VoeAirlines

Local

C:\Users\romul\source\repos

...

Nome da solução ⓘ

VoeAirlines

☐

Colocar a solução e o projeto no mesmo diretório

## Informações adicionais

Aplicativo Blazor Server

C#

Linux


macOS

Windows

Blazor

Nuvem


Web

Framework 


.NET 6.0 (Suporte de longo prazo)

Tipo de autenticação 


Nenhum

☒ Configurar para HTTPS 

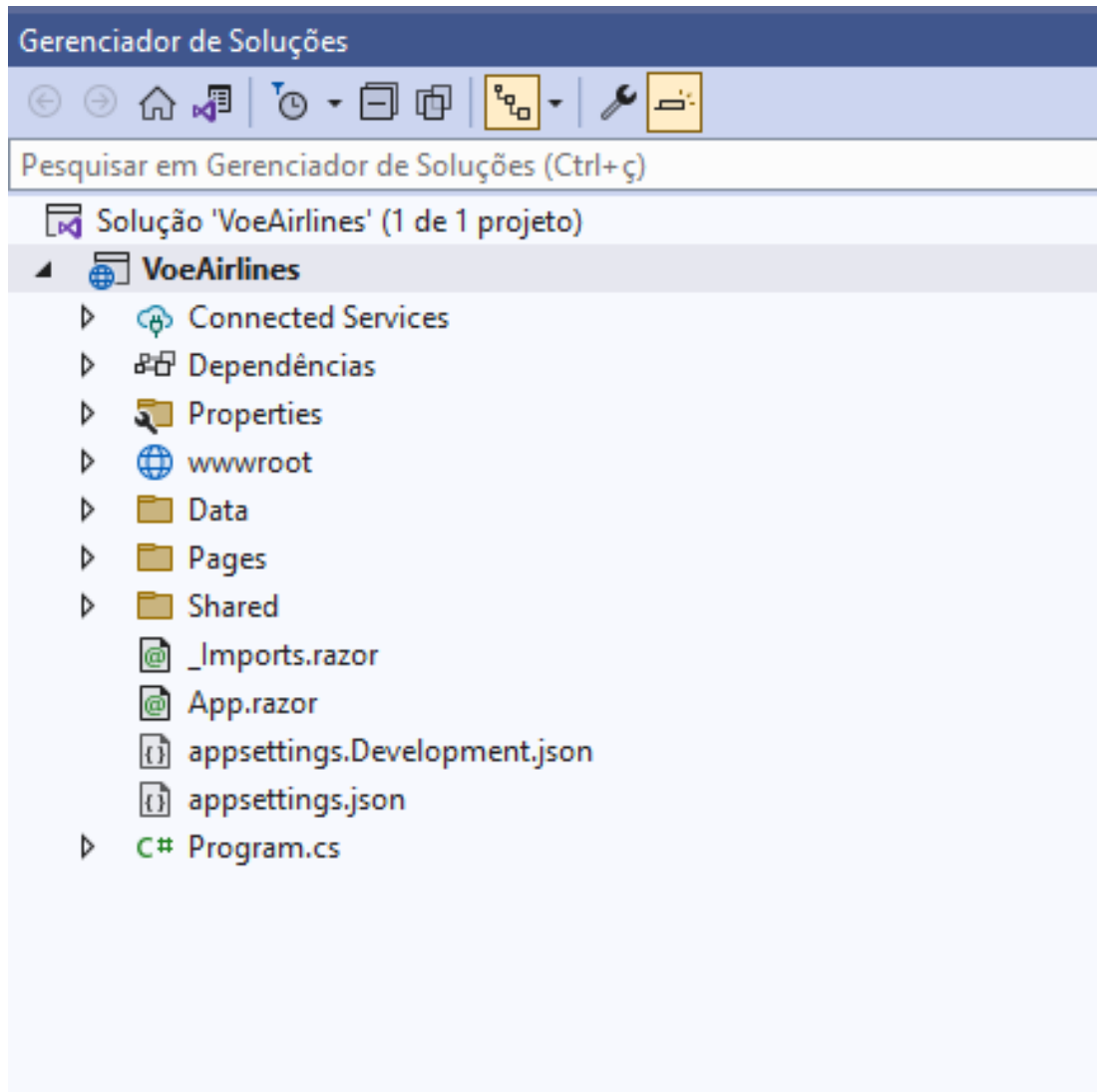
☐ Habilitar o Docker 

Sistema Operacional do Docker 

Linux

☐ Não use instruções de nível superior 





Program.cs → adicione a seguinte codificação

```
builder.Services.AddScoped
```

```
(
```

```
    s =>
```

```
{
```

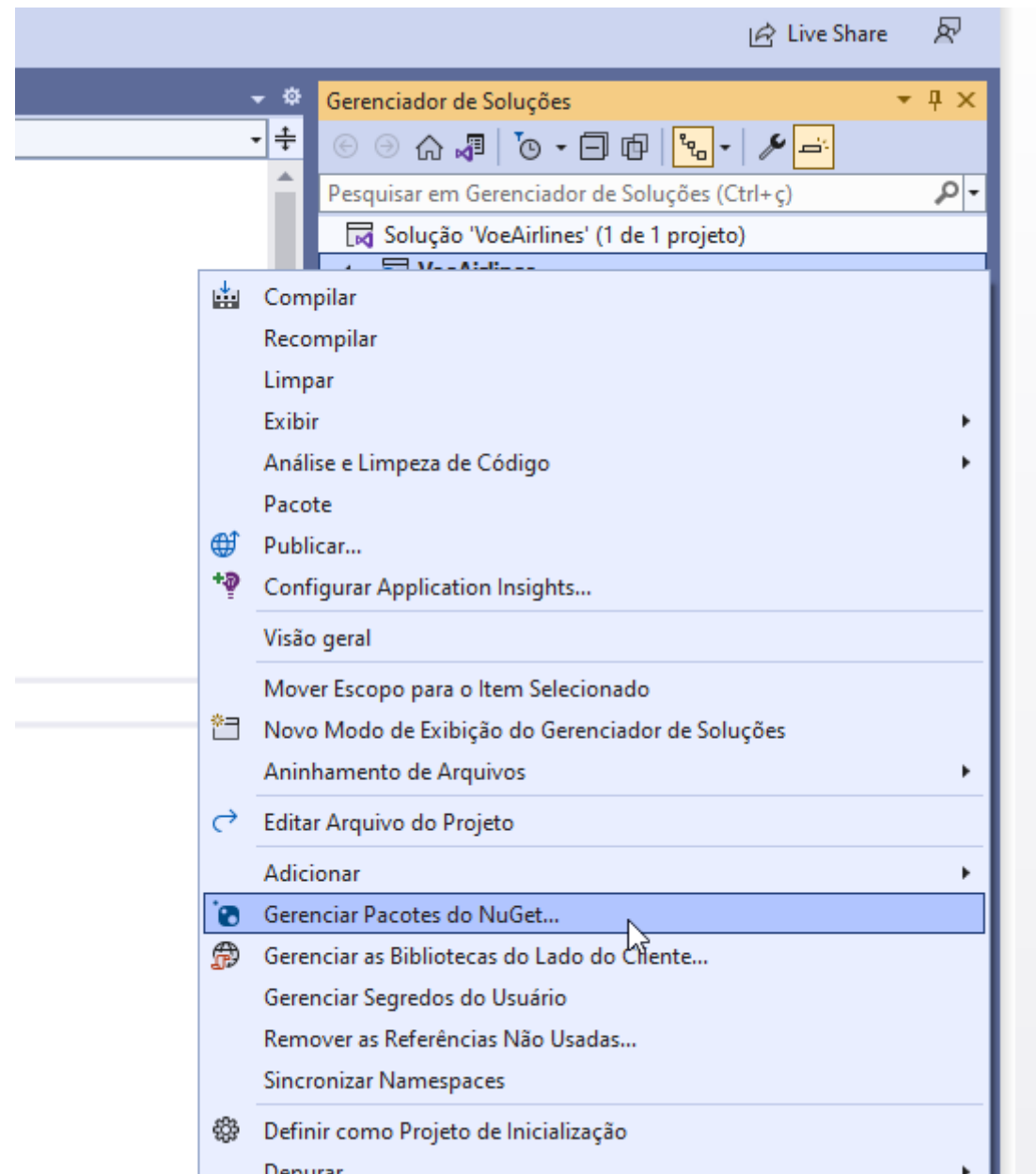
```
    return new HttpClient { BaseAddress = new Uri(@"https://www.edukacode.com.br") };
```

```
}
```

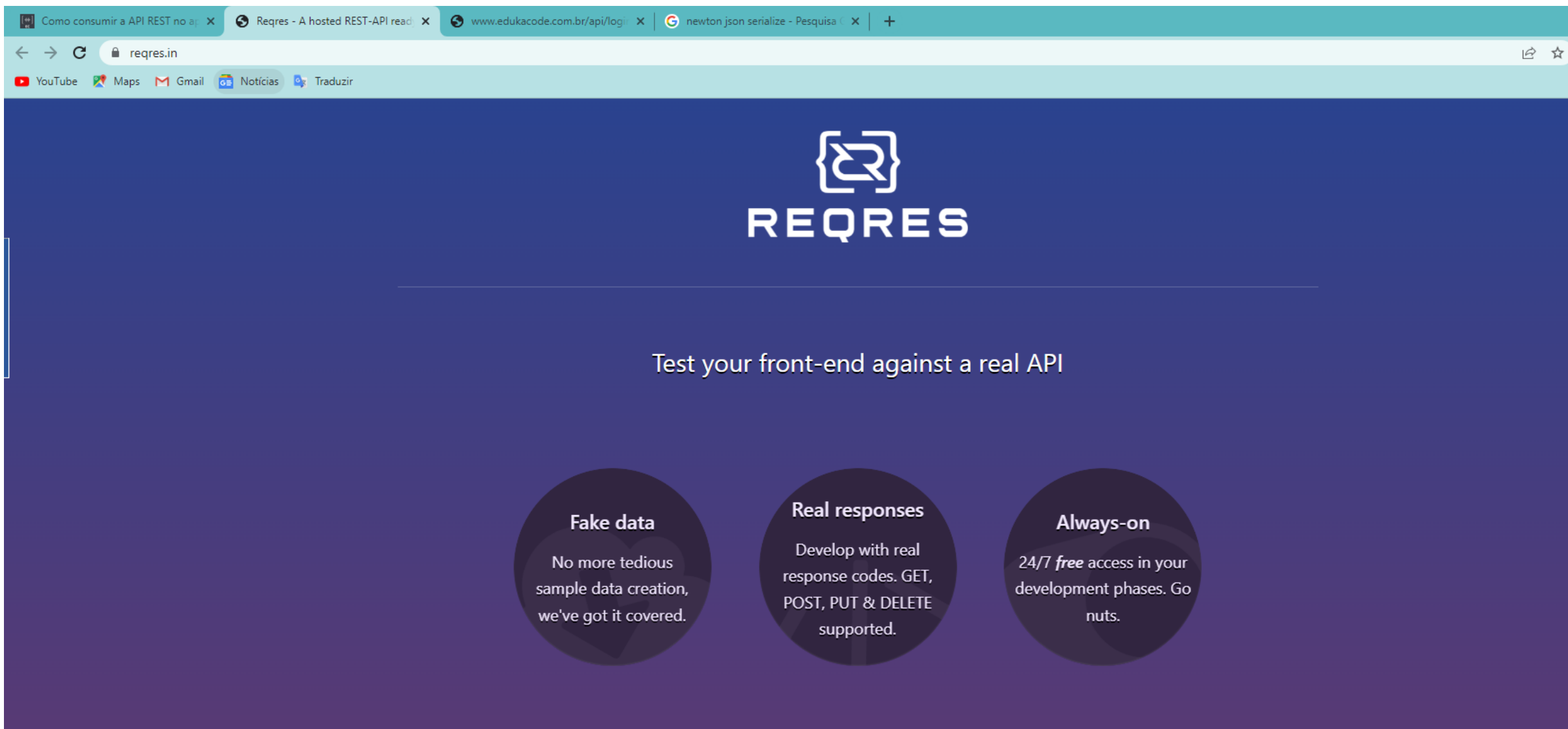
```
);
```

EndPoint da Api do professor:

<http://www.edukacode.com.br/api/login>



Apenas uma dica: caso queira usar outra api testar.



The image is a screenshot of a web browser displaying the Reqres website. The browser's address bar shows 'reqres.in'. The website has a dark blue background. At the top center is the Reqres logo, which consists of a stylized white icon of a computer monitor with a circular arrow inside, and the word 'REQRES' in white capital letters below it. Below the logo, the text 'Test your front-end against a real API' is centered. At the bottom, there are three dark blue circular callouts with white text. The first callout is titled 'Fake data' and contains the text 'No more tedious sample data creation, we've got it covered.' The second callout is titled 'Real responses' and contains the text 'Develop with real response codes. GET, POST, PUT & DELETE supported.' The third callout is titled 'Always-on' and contains the text '24/7 *free* access in your development phases. Go nuts.'

Como consumir a API REST no a... x

Reqres - A hosted REST-API read... x

www.edukacode.com.br/api/login x

newton json serialize - Pesquisa x

+ x

reqres.in

YouTube Maps Gmail Notícias Traduzir

# REQRES

Test your front-end against a real API

### Fake data

No more tedious sample data creation, we've got it covered.

### Real responses

Develop with real response codes. GET, POST, PUT & DELETE supported.

### Always-on

24/7 *free* access in your development phases. Go nuts.

# Deve instalar extensão do Newtonsoft.JSON

NuGet: VoeAirlines

Login.cs\*

WeatherForecastService.cs

WeatherForecast.cs

Program.cs\*

VoeAirlines: Visão geral

Procurar

Instalado













Atualizações


Newtonsoft.Json


Incluir pré-lançamento

Gerenciador de Pacotes do NuGet: VoeAirlines

Origem do pacote: nuget.org

	<b>Newtonsoft.Json.Bson</b>  por James Newton-King, 350M downloads	1.0.2	↓
Json.NET BSON adds support for reading and writing BSON			
	<b>Newtonsoft.Json.Schema</b>  por Newtonsoft, 26,4M downloads	3.0.14	
Json.NET Schema is a complete and easy-to-use JSON Schema framework for .NET			
	<b>Microsoft.AspNetCore.Mvc.Newtonsoft.Json</b>  por Microsoft, 201M downloads	7.0.0	
ASP.NET Core MVC features that use Newtonsoft.Json. Includes input and output formatters for JSON and JSON PATCH.			
	<b>Swashbuckle.AspNetCore.Newtonsoft</b> por Swashbuckle.AspNetCore.Newtonsoft, 41,1M downloads	6.4.0	
Swagger Generator opt-in component to support Newtonsoft.Json serializer behaviors			
	<b>GraphQL.Newtonsoft.Json</b> por GraphQL.Newtonsoft.Json, 4,45M downloads	7.1.1	
JSON.NET serializer for GraphQL.NET			
	<b>RestSharp.Newtonsoft.Json</b> por Adam Fisher, 2,91M downloads	1.5.1	⚠
Restores Newtonsoft.JSON as the default serializer for RestSharp. This package version is deprecated. Use <a href="#">RestSharp</a> instead.			
	<b>App.Metrics.Formatters.Json</b> por Allan Hardy and Contributors, 18,2M downloads	4.3.0	
App Metrics Formatters for metrics and environment data to JSON using Newtonsoft.Json.			
	<b>NServiceBus.Newtonsoft.Json</b>  por Particular Software. 5.74M downloads	3.0.0	



**Microsoft.AspNetCore.Mvc.Newtonsoft.Json** 

Versão:

Estável mais recente 7.0.0

Instalar

Opções

Descrição

ASP.NET Core MVC features that use Newtonsoft.Json. Includes input and output formatters for JSON and JSON PATCH.

This package was built from the source code at <https://github.com/dotnet/aspnetcore/tree/bb01bbf4433e27289b99001b7de6a582879d1835>

Versão:

7.0.0

Autor(es):

Microsoft

Licença:

MIT

Data da publicação:

segunda-feira, 7 de novembro de 2022 (07/11/2022)

URL do Projeto:

<https://asp.net/>

Relatar Abuso:

<https://www.nuget.org/packages/Microsoft.AspNetCore.Mvc.Newtonsoft.Json/7.0.0/ReportAbuse>

Marcas:

aspnetcore, aspnetcoremvc, json

Dependências

net7.0

Microsoft.AspNetCore.JsonPatch (>= 7.0.0)

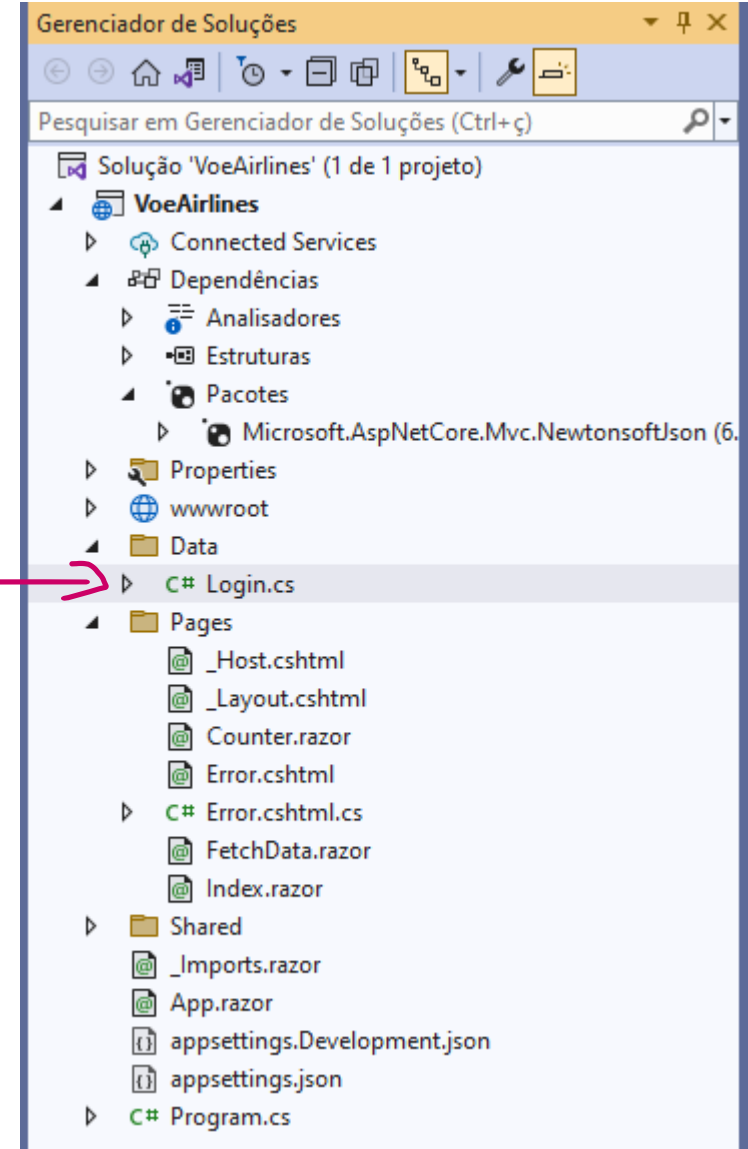
Newtonsoft.Json (>= 13.0.1)

Newtonsoft.Json.Bson (>= 1.0.2)

Pasta: Data

```

1  using Microsoft.AspNetCore.Mvc.NewtonsoftJson;
2  using Newtonsoft.Json;
3
4  namespace VoeAirlines.Data
5  {
6      3 referências
7      public class Login
8      {
9          [JsonProperty ("Usuario")]
10         1 referência
11         public string? Usuario { get; set; }
12
13         [JsonProperty ("DataCriacao")]
14         1 referência
15         public DateTime DataCriacao { get; set; }
16     }
17 }
    
```



```

Index.razor  X  NuGet: VoeAirlines  Login.cs  Program.cs  VoeAirlines: Visão geral
1  @page "/"
2  @inject HttpClient client
3  @using Newtonsoft.Json
4  @using Data
5
6  <div class="container-fluid">
7      <div class="row justify-content-center" style="margin-bottom:50px">
8          <h3 class="text-center">Chamando uma REST-API</h3>
9      </div>
10     <div class="row" style="margin:25px;">
11         <strong>Get Operation </strong><hr />
12         @if (Users != null)
13         {
14             <table class="table table-bordered table-striped">
15                 <thead class="thead-dark">
16                     <tr>
17                         <th>Usuario</th>
18                         <th>Data Criação</th>
19                     </tr>
20                 </thead>
21                 <tbody>
22                     @foreach (var user in Users)
23                     {
24                         <tr class="justify-content-center">
25                             <td>@user.Usuario</td>
26                             <td>@user.DataCriacao</td>
27                         </tr>
28                     }
29                 </tbody>
30             </table>
31         }
32
33     </div>
34     <div class="row" style="margin:10px;margin-bottom:50px;">
35         <div class="col"><button class="btn btn-info" @onclick="GetUser">Get Users</button></div>
36     </div>
37
38 </div>

```

Primeira parte!



Segunda parte!

```
40 @code {  
41     List<Login>? Users;  
42     private async void GetUser()  
43     {  
44         var apiName = "http://www.edukacode.com.br/api/login/";  
45         var httpResponse = await client.GetAsync(apiName);  
46         if (httpResponse.IsSuccessStatusCode)  
47         {  
48             Users = new List<Login>();  
49             var apiConteudo = await httpResponse.Content.ReadAsStringAsync();  
50             Users = JsonConvert.DeserializeObject<List<Login>>(apiConteudo);  
51             StateHasChanged();  
52         }  
53     }  
54 }  
55 }  
56
```

Resultado!

## Chamando uma REST-API

### Get Operation

Usuario	Data Criação
admin	01/01/0001 00:00:00
romulo.leite	01/01/0001 00:00:00
adriana-topshow	23/11/2022 21:13:25
daniel	23/11/2022 22:46:00
daniel	23/11/2022 22:46:00
cleiton	23/11/2022 23:07:03
Danilo	24/11/2022 15:11:29
nayara	25/11/2022 13:47:29

Get Users