



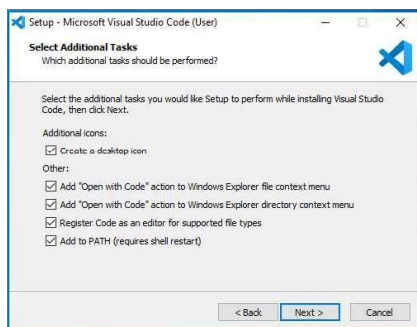
AULA 01 INAUGURAL

- Apresentação da disciplina
- Competências, habilidades e bases tecnológicas da disciplina
- Formas de Avaliação
- Introdução e desenvolvimento do conteúdo

Download e Instalação do Visual Studio Code

Faça o Download do Visual Studio Code acessando <https://code.visualstudio.com/download>

Instale deixando todas as opções selecionadas



Abra o PowerShell e utilize os comandos abaixo para checar as versões do .NET instaladas.



```
PS C:\Users\luiz> dotnet --version
7.0.100
```

```
PS C:\Users\luiz> dotnet --list-sdks
5.0.408 [C:\Program Files\dotnet\sdk]
6.0.301 [C:\Program Files\dotnet\sdk]
7.0.100 [C:\Program Files\dotnet\sdk]
```

Se seu computador não exibir nenhuma versão ou não reconhecer o comando, instale o .Net Core através do link a seguir: <https://dotnet.microsoft.com/en-us/download/dotnet>. A versão recomendada para as aulas é a 7.0

Version	Release type	Support phase	Latest release	Latest release date	End of support
.NET 7.0 (latest)	Standard Term Support ⓘ	Active ⓘ	7.0.2	January 10, 2023	May 14, 2024
.NET 6.0	Long Term Support ⓘ	Active ⓘ	6.0.13	January 10, 2023	November 12, 2024



Vá em computador, clique com o direito do mouse e em propriedades, para verificar se seu Windows é 32 ou 64 bits e faça o download compatível com seu computador

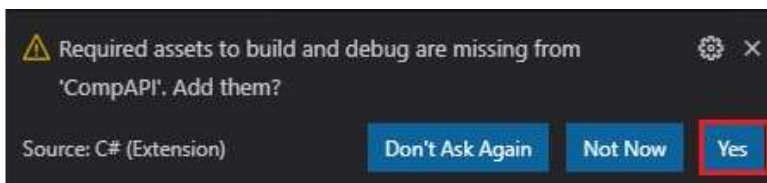
OS	Installers	Binaries
Linux	Package manager instructions	ARM32 ARM64 RHEL 6 x64 x64 x64 Alpine
macOS	x64	x64
Windows	x64 x86 — Windows 64 bits Windows 32 bits	ARM32 x64 x86
All	dotnet-install scripts	

Projetos WebApi

Crie uma pasta na sua organização de arquivos chamada **RpgApi** e abra ela no VS Code. Depois disso abra uma o terminal e digite o comando para criação de uma API, conforme abaixo

dotnet new WebApi

Após a criação aparecerá uma a mensagem para ativar o modo de depuração para o C#. Escolha sim conforme abaixo

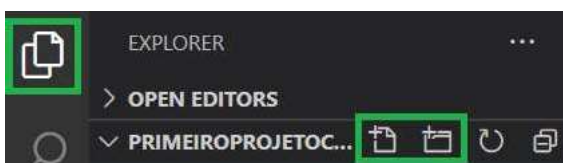


- Para compilar o projeto navegue até o menu View → Terminal e execute `dotnet build`
- O resultado esperado é sempre 0 erros
- Para rodar o projeto execute o comando a seguir no terminal `dotnet run`

Extensões importantes para o VS Code: É possível adicionar extensões ao VS Code. Vá até até até o menu View → Extensions e na caixa de busca, digite as extensões listada abaixo e clique no botão install

- C# Extensions – Autor: JosKreativ
- Material Icon Theme – Autor: Philipp Kief
- C# For Visual Studio Code – Autor: Microsoft (provavelmente já estará adicionada)

Podemos observar os arquivos abertos conforme a imagem abaixo e os ícones em que podemos criar arquivos e pastas, sendo possível criar arquivos e pastas clicando com o botão direito.



O arquivo `Program.cs` é uma classe e é o ponto de partida para a execução do projeto.



Identificação dos arquivos no Projeto

Classe Program: Será o ponto de partida ao rodar o projeto, como mencionado acima, nela está apontada a classe Startup.

Arquivo .csproject: Arquivo em que ficará registrado dos os pacotes baixados para utilização no projeto. Framework do banco de dados por exemplo.

Appsettings.json: Arquivo em que pode ser guardado informações de configurações, por exemplo o IP e dados de acesso de um banco de dados por exemplo.

Launchsettings.json (pasta properties): Arquivo em que estarão informações sobre a execução do projeto, por exemplo qual o endereço que constará no navegador ao rodar a aplicação ou se utilizará o protocolo http ou https por exemplo. Neste arquivo remova o endereço *https* que aparece na propriedade *applicationUrl* para que o navegador não exiba mensagem de bloqueio ao executar o aplicativo

```
"CompDS": {  
  "commandName": "Project",  
  "launchBrowser": true,  
  "launchUrl": "weatherforecast",  
  "applicationUrl": "https://localhost:5001;http://localhost:5000",  
  "environmentVariables": {  
    "ASPNETCORE_ENVIRONMENT": "Development"  
  }  
}
```

- Isso é necessário para que ao rodar localmente o projeto, não ocorram problemas por não existir certificado de conexão segura.

Abra o terminal através do menu View → Terminal, digite a linha de comando `dotnet run` para rodar o projeto. Abra o navegador, digite o endereço e porta da API e o nome Controller que temos até então:

`localhost:5000/WeatherForecast`

O Navegador deverá exibir dados aleatórias em C° e F° que se trata da avaliação de temperaturas. Execute o comando CTRL + C no Visual Studio Code para interromper a aplicação assim que desejar.

Nas próximas etapas entenderemos melhor o que é uma Controller, mas como uma breve introdução, durante a criação do projeto foi criada uma Controller chamada WeatherForecast automaticamente na pasta correspondente



Testando API com Postman

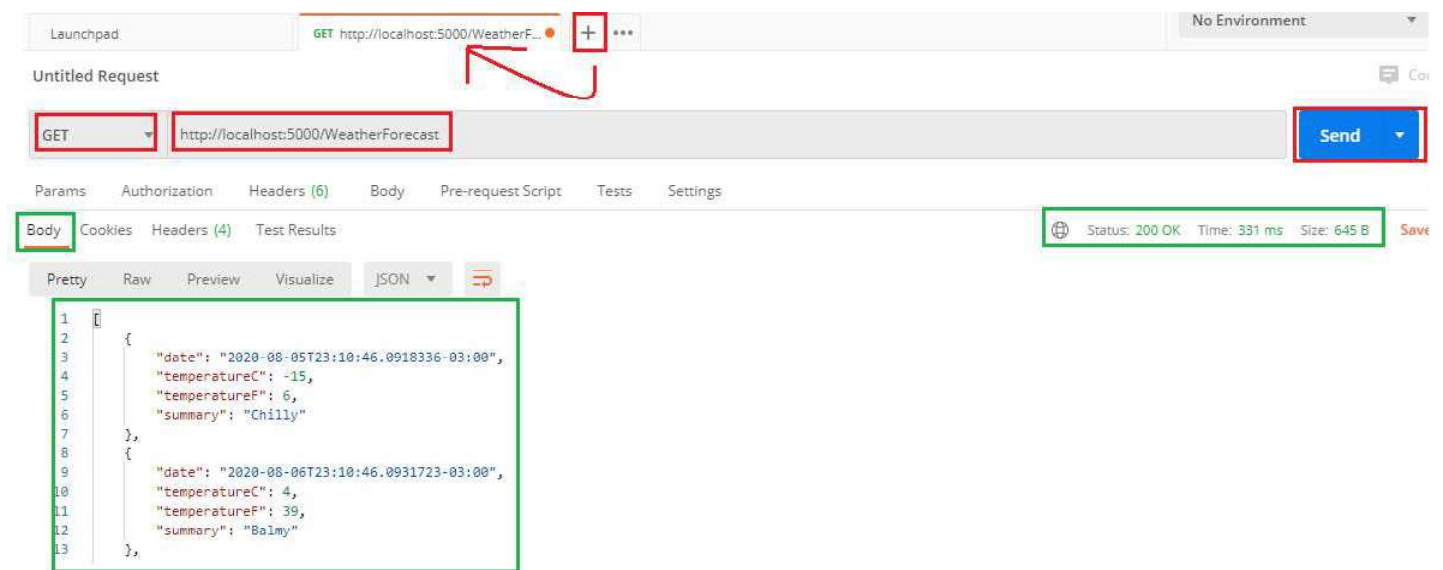
Como ainda não temos front-end (interface) ainda, precisaremos de uma ferramenta para executar testes no nosso back-end (Programação) e se mostra uma alternativa mais completa para testar todos os recursos que uma API oferece em comparação como o navegador. O Postman pode ser baixado através do endereço abaixo:

<https://www.postman.com/downloads/>

Você pode realizar o login através do gmail e manter o histórico de todos os seus testes dentro da ferramenta.

Em linhas gerais, o Postman é um API Client que podemos utilizar para realizar as requisições na API através dos principais métodos: Get, Post, Put e Delete

Execute a aplicação e realize as seguintes configurações no Postman para poder testar o método Get da API



- Em vermelho temos as configurações que devem ser feitas e em verde o resultado da requisição na API.