

Crie uma pasta e abra ela usando vs code execute os seguintes comandos

`npx create-react-app my-app` `create-react-app` (o nome que você quiser para a sua pasta)

Para ver se seu aplicativo está funcionando vá para a pasta do aplicativo com o comando `cd` (nome que você colocou para sua pasta) e depois para rodar o aplicativo escreva: `npm start`

Após testar e verificar que esta funcionando, insira o seguinte código que vai ser padrão no js

```
import React, { useState } from 'react';
```

```
import axios from 'axios';
```

```
function App() {
```

```
  const [message, setMessage] = useState(""); /* Estado para armazenar a mensagem recebida do backend */  
}
```

```
/*LOCAL PARA CRIAR FUNÇÕES OU FORMULÁRIOS COMO NOS EXEMPLOS SEGUINTE*/
```

```
export default App;
```

E rode `npm install axios` dentro da pasta do projeto react usando `cd <nome da pasta>` para funcionar o axios

Criação do projeto com C#:

Use `cd <nome da pasta do projeto>` sem entrar dentro do projeto react para criar uma API e digite : `dotnet new webapi -o {Nome da API}`

Coloque os seguintes codigos na api em `program.cs` para resolver erros https futuros

```
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Http;
using Microsoft.Extensions.Hosting;
using System.Text.Json;
using System.Threading.Tasks;
var builder = WebApplication.CreateBuilder(args);
// Configuração do CORS
builder.Services.AddCors();
var app = builder.Build();
if (app.Environment.IsDevelopment())
{
```

```
app.UseDeveloperExceptionPage();
}

// Habilitar o CORS

app.UseCors(builder => builder
.AllowAnyOrigin() // Permitir solicitações de qualquer
//origem

.AllowAnyMethod() // Permitir solicitações de qualquer
//método (GET, POST, PUT, etc.)

.AllowAnyHeader()); // Permitir solicitações com
//qualquer cabeçalho]

//AQUI SERA O LOCAL PARA VOCE COLOCAS AS
//FUNCOES DAS PARTES

// POSTERIORMENTES MOSTRADAS


app.Run();

//AQUI SERA O LOCAL PARA COLOCAR AS CLASSES
//POSTERIORMENTE USADAS

//

também vai em >Properties>launchSettings.json e
apague de profiles o primeiro http, deixe somente o que
tem https e http, ficara assim
```

```
Appjs M  Projeto-React.sln  Program.cs 3  Startup.cs  launchSettings.json X  MyAp.http
MyAp > Properties > {} launchSettings.json > {} profiles > {} IIS Express > {} launchBrowser
1
2 {
3   "$schema": "http://json.schemastore.org/launchSettings.json",
4   "iisSettings": {
5     "windowsAuthentication": false,
6     "anonymousAuthentication": true,
7     "iisExpress": {
8       "applicationUrl": "http://localhost:44662",
9       "sslPort": 44368
10    }
11  },
12  "profiles": {
13    "http": {
14      "commandName": "Project",
15      "dotnetRunMessages": true,
16      "launchBrowser": true,
17      "launchUrl": "swagger",
18      "applicationUrl": "https://localhost:7279;http://localhost:5171",
19      "environmentVariables": {
20        "ASPNETCORE_ENVIRONMENT": "Development"
21      }
22    },
23    "IIS Express": {
24      "commandName": "IISExpress",
25      "launchBrowser": true,
26      "launchUrl": "swagger",
27      "environmentVariables": {
28        "ASPNETCORE_ENVIRONMENT": "Development"
29      }
30    }
31  }
32 }
```