

# Manual da Instituição

*por*

**Gabriel Couto Assis**  
**Oscar Praga de Sousa**



**CEFET-MG Campus-V - Divinópolis**

10 de agosto de 2023

# Conteúdo

<b>1</b>	<b>Manual da Instituição</b>	<b>1</b>
1.1	Obtenção . . . . .	1
1.2	Requisitos do sistema . . . . .	1
1.3	Compilar e rodar . . . . .	2
1.4	Utilização . . . . .	2

# Capítulo 1

## Manual da Instituição

Neste documento será apresentado como a instituição que utilizar o RAEG (Registro e acompanhamento de estudantes egressos), deve proceder ao decidir utilizar a plataforma internamente.

Este documento possui um linguagem técnica, é aconselhável que o responsável do setor de tecnologia da informação da instituição leia-o e aconselhe a diretoria da mesma.

### 1.1 Obtenção

O RAEG é uma aplicação feita em duas partes, o "backend" e o "frontend", ambos estão disponíveis no link: <https://github.com/Couto1411/RAEG>, nas pastas "backendsharp" e "frontend", respectivamente.

### 1.2 Requisitos do sistema

O RAEG foi desenvolvido e testado sob os ambientes do Firebase Hosting para o "frontend" e do Amazon EC2 para o "backend", com banco de dados no sistema Amazon RDS. O "backend" foi feito em ASP.NET Core 6, e testado no IIS, portanto foi feito apenas em ambientes que possuem Windows Server.

O "backend" funcionou com respostas em menos de 1s às requisições mais comuns do "frontend" no sistema EC2 t2.micro, com sistema operacional Microsoft Windows Server 2022 Base, com 1 GiB de memória RAM, 30 GiB de Armazenamento em gp2(SSD de uso geral) e alocação compartilhada.

Lembra-se também que o ambiente que irá hospedar o "backend" deve possuir os seguintes pacotes:

- .NET 6.0.20 - Windows Server Hosting
- .NET Runtime - 6.0.20
- .NET SDK - 6.0.412

- ASP.NET Core - 6.0.20

E o ambiente que irá hospedar o "frontend" deve possuir o Node.js com a versão testada 18.16.0.

## 1.3 Compilar e rodar

Antes da compilação é necessário criar um arquivo na pasta `backendcsharp\backendcsharp` chamado `"appsettings.json"` com a seguinte estrutura:

```
1 {
2   "TokenConfigurations": {
3     "Audience": "Frontend APP",
4     "Issuer": "Backend API",
5     "Seconds": 3600,
6     "SecretJwtKey": "SUA-CHAVE-DE-ENCRIPTAÇÃO"
7   },
8   "ConnectionStrings": {
9     "DefaultConnection": "SUA-CONNECTION-STRING"
10  },
11  "AppAdminInfo": {
12    "email": "SEU-EMAIL-DE-PROPRIETARIO",
13    "senha": "SUA-SENHA-DE-PROPRIETARIO"
14  }
15 }
```

Também é importante que no arquivo `frontend\src\config\api.jsx`, `"sua-url-backend"` deve ser substituída pela url em que seu "backend" está hospedado.

Para obter os arquivos de produção, basta compilar ambos os projetos nas pastas `backendcsharp\backendcsharp` e `frontend`, para isso deve-se rodar os comandos `dotnet build`, os arquivos de compilação estarão na pasta `backendcsharp\backendcsharp\bin\Debug\net6.0\publish` e `npm run build`, os arquivos de compilação estarão na pasta `frontend\build`, respectivamente.

Para rodar localmente basta utilizar o comando `dotnet run` na pasta `backendcsharp\backendcsharp` e o comando `npm serve` na pasta `frontend`. Para rodá-los em outros serviços basta utilizar os arquivos de compilação nos mesmos seguindo os devidos passos para produção.

## 1.4 Utilização

A utilização do proprietário (instituição) é semelhante a de qualquer outro administrador do sistema, retirando o fato de que o cadastro do proprietário não pode ser excluído

## *1.4 Utilização*

---

através do sistema. Utilize o email e senha presentes no "appsettings.json" citado na seção anterior.