

## Avaliação vaga desenvolvedor Back-end

**Prazo de entrega:** 19/02/2021 até 12:00.

Sua tarefa é construir um simples repositório para gerenciar desenvolvedores com seus respectivos aplicativos desenvolvidos.

Utilize um repositório Git (público, de sua preferência) para versionamento e disponibilização do código.

A aplicação deve ser construída em .NET Core, utilizando EF Core com banco de dados PostgreSQL, pode utilizar qualquer lib disponível no NuGET.

Utilizar o Swagger para documentar a API criada.

### *O que será avaliado?*

Queremos avaliar sua capacidade de desenvolver e documentar um back-end para uma aplicação.

Serão avaliados:

- Código bem escrito e limpo;
- Quais libs foram usadas, como e por que, além do seu conhecimento das mesmas;
- Seu conhecimento em banco de dados, requisições HTTP, APIs REST, etc;
- Sua capacidade de documentação da sua parte da aplicação.
- Organização do repositório

### *O mínimo necessário*

Uma aplicação contendo uma API real simples, sem autenticação, que atenda os requisitos descritos abaixo, fazendo requisições a um banco de dados para persistência, possuindo uma documentação da API utilizando o Swagger.

### *Bônus*

Os seguintes itens não são obrigatórios, mas darão mais valor ao seu trabalho:

- Cuidados especiais com otimização, padrões, entre outros;
- Utilização de migrations para configuração do banco de dados utilizado;
- Testes Automatizado unitário/integração;
- Autenticação e autorização JWT;
- Utilizar o padrão CQRS, MediatR, FluentValidator.
- Utilizar conceitos do DDD (Domain Driven Design)
- Docker compose que suba a API criada no Docker.

### *Requisitos da aplicação:*

**0.** A Api deve responder na porta 3000

**1.** Deve haver uma rota para listar todos os desenvolvedores cadastrados:

GET /desenvolvedor

**2.** Deve haver uma rota para listar todos os aplicativos cadastrados:

GET /aplicativo

**3.** Deve haver uma rota para cadastrar um novo desenvolvedor

POST /desenvolvedor

O corpo da requisição deve conter as informações do desenvolvedor a ser cadastrado, sem o Id que deve ser gerado automaticamente pelo servidor. A resposta, em caso de sucesso, deve ser o novo Id gerado.

Deve realizar as seguintes validações:

- Nome não pode ser vazio ou nulo;
- Nome não pode ser maior que 255 caracteres;
- CPF não pode ser vazio ou nulo;
- CPF deve ser válido;
- E-mail não pode ser vazio ou nulo
- E-mail não pode ser maior que 100 caracteres;
- E-mail deve ser válido;

Em caso de falha em qualquer uma das validações a cima deve retornar com o status 400 Bad Request e as respectivas mensagens.

Status: 400 Bad Request

```
{
  erros: [
    {
      campo: "nome",
      erro: "nome é obrigatorio"
    },
    {
      campo: "cpf",
      erro: "cpf inválido"
    }
  ]
}
```

4. Deve existir uma rota para cadastrar o aplicativo  
POST /aplicativo

O corpo da requisição deve conter as informações do aplicativo a ser cadastrado, sem o Id que deve ser gerado automaticamente pelo servidor. A resposta, em caso de sucesso, deve ser o novo Id gerado.

Deve realizar as seguintes validações:

- Nome não pode ser vazio ou nulo;
- Nome não pode ser maior que 255 caracteres;
- Nome não pode repetir em mais de 1 aplicativo.
- Data do lançamento não pode ser vazia nem nula;
- Data do lançamento deve ser válida;
- Plataforma de programação deve ser 1-Desktop, 2-Web ou 3-Mobile.
- Desenvolvedor responsável deve ser válido.

Tratamentos de sucesso e falha devem ser semelhantes ao da rota de desenvolvedor.

5. O usuário deve poder alterar um desenvolvedor  
PUT /desenvolvedor/id

6. O usuário deve poder alterar um aplicativo  
PUT /aplicativo/id

7. O usuário deve poder remover um desenvolvedor por id

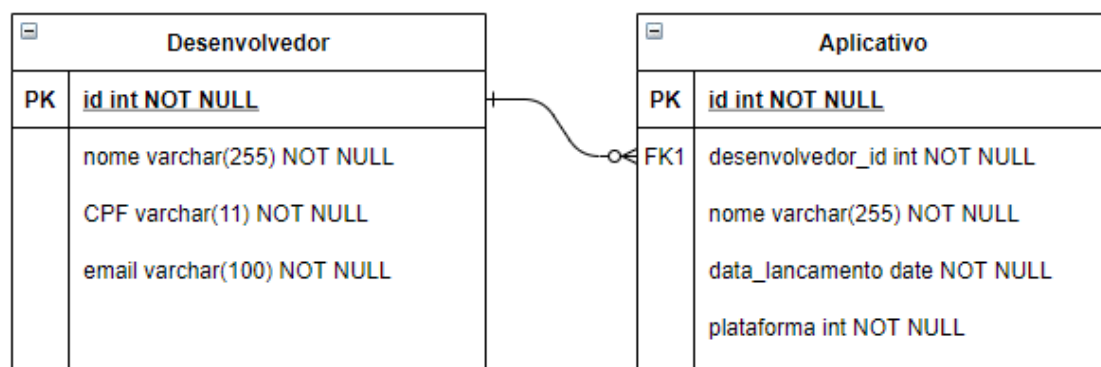
DELETE /desenvolvedor/id

Não pode ser removido desenvolvedor com aplicativo vinculado.

8. O usuário deve poder remover um aplicativo por id

DELETE /aplicativo/id

### *Estrutura de dados*



### *Cr terios de Aceita  o*

A API deve ser real e escrita por voc . Ferramentas que criam APIs automaticamente (Loopback, json-server, etc) n o s o aceitas;

Todos os requisitos acima devem ser cumpridos, seguindo o padr o de rotas estabelecido;

Se voc  julgar necess rio, adequado ou quiser deixar a aplica  o mais completa (b nus!) voc  pode adicionar outras rotas, m todos, meios de autentica  o com usu rios, etc;

Organiza  o do reposit rio, branch's e commit's;

### *V deo de explica  o do projeto:*

Quando terminar de realizar o desafio voc  dever  gravar um v deo de no m ximo 30 minutos, contando pra gente esses principais pontos:

- Compartilhe sua tela no v deo e mostre o que voc  construiu no desafio, passando sobre o c digo e o resultado final, e os pontos que voc  julga importante para o funcionamento do mesmo;
- As decis es que teve que tomar para constr  -lo;

Para conseguir mostrar o seu desafio para n s, voc  deve baixar uma ferramenta de v deo chamada Loom. Ela   uma extens o do navegador totalmente gratuita, na qual voc  poder  gravar o seu desafio seguindo o roteiro que indicamos acima.

N o se esque a de nos enviar o link do seu v deo, junto com a sua avalia  o.

### *O que deve ser nos enviado:*

- Link do reposit rio do GIT.
- Link do v deo do Loom explicando o que foi realizado.

Enviar os links para o e-mail [fabiano.rates@inovamobil.com.br](mailto:fabiano.rates@inovamobil.com.br) at  a data e hor rio limite.