



# growdev

---

## PROGRAMA STARTER FULL STACK WEB DEVELOPER

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal. É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

# Desafio final

## *Módulo Back-end I*

Neste módulo vamos aprender sobre lógica de programação, javascript, API Rest, NodeJS, Express, libs e como subir uma aplicação back-end em produção.

O objetivo desta avaliação é verificar se vamos atingir os conhecimentos propostos para este módulo.



## Desafio final

*O que vamos fazer?*

### Aplicação Lista de Recados

- Caso você não tenha tido a aula de express, mas já teve a base de JavaScript faça as funções e códigos ;
- Assim que tiver a aula de express, refatore os códigos criados anteriormente para o express.





# Desafio final

*O que vamos fazer?*

## Aplicação Lista de Recados

- Vamos criar um back-end que contém as seguintes funcionalidades:
  - Criação de conta
  - Login
  - CRUD\* de recados

\*CRUD (create, read, update e delete)



**Importante !!!**

**As rotas e respostas devem seguir o enunciado, caso contrário não será aprovado .**





## Parte 1 – Configurando



## Desafio final

*O que vamos fazer?*

- **Faça as configurações iniciais do express ;**
- **Crie um recurso com a seguinte rota ' / '**

**Em caso de sucesso, retorna um status 200**

**Com a seguinte mensagem  
( 'Bem vindo à aplicação' )**





## Parte 2 – Pessoa Usuária





## Desafio final

*O que vamos fazer?*

### Regras Gerais

- Não pode ter mais de uma pessoa usuária com o mesmo e-mail
- O login deve ser feito com e-mail e senha
- Cada recado deve ser uma pessoa usuária . Ou seja , uma pessoa pode ter vários recados. Porém um recado só pode ter uma pessoa.



**Sign Up**





## Desafio final

*O que vamos fazer?*

- **Sign Up – Criar pessoa usuária**

**Endereço :** /signup

**Dados :**

- Id (identificador único)
- Name
- Email
- Password





## Desafio final

*O que vamos fazer?*

- **Sign Up – Criar pessoa usuária**

Cada pessoa usuário deve ser um **newUser**

- Name, email, password devem ser inseridos pela pessoa usuária e todos esses itens precisam ser validados .
- Caso não passe um desses itens, informe um status (400) .



## Desafio final

*O que vamos fazer?*

- **Sign Up – Criar pessoa usuária**

Caso a pessoa seja criada com sucesso, retornar um status (201) e a seguinte mensagem :

" Seja bem vindo { NOME } ! Pessoa usuária registrada com sucesso! "





## Desafio final

*O que vamos fazer?*

- **Sign Up – Criar pessoa usuária**

Mensagens de verificação :

- Por favor, verifique se passou o nome .
- Por favor, verifique se passou o email .
- Email já cadastrado, insira outro .
- Por favor, verifique se passou a senha .





## Desafio final

*O que vamos fazer?*

- **Sign Up – Criar pessoa usuária**

O identificador único, não deve ser inserido pela a pessoa usuária ao criar .

Este item deve ser feito de forma **automatizada**

- Você deve criar uma lista estática, que armazene as pessoas usuárias , cujo o nome seja **users**



Login





## Desafio final

*O que vamos fazer?*

- **Login – Acessar a pessoa usuária**

**Endereço :** /login

**Dados :**

- Email
- Password





## Desafio final

*O que vamos fazer?*

- **Login – Acessar a pessoa usuária**

Deve-se verificar se a pessoa usuária inseriu os seguintes dados para fazer o login :

- Email
- Password



## Desafio final

*O que vamos fazer?*

- **Login – Acessar a pessoa usuária**

Caso o login seja realizado com sucesso, retornar um status (200) e a seguinte mensagem :

" Seja bem vindo { NOME } ! Pessoa usuária logada com sucesso! "





## Desafio final

*O que vamos fazer?*

- **Login – Acessar a usuária**

Caso a pessoa não tenha enviado email ou password você deve retornar as seguintes mensagens :

- Insira um e-mail válido
- Insira uma senha válida





## Desafio final

*O que vamos fazer?*

- **Login – Acessar a usuária**

Além disso, só será possível fazer o login caso o email esteja cadastrado no sistema.

Caso contrário, retornar a seguinte mensagem :

" Email não encontrado no sistema, verifique ou crie uma conta"



## Parte 3 – Mensagens





## Criar Mensagens





## Desafio final

*O que vamos fazer?*

- **Criar mensagem – Create**

**Endereço :** /message

**Dados :**

- Id ( identificador único)
- Title ( título da mensagem )
- Description ( descrição da mensagem )



## Desafio final

*O que vamos fazer?*

- **Criar mensagem – Create**

Caso a operação seja realizada com sucesso, retornar um status (201) e a seguinte mensagem :

" Mensagem criada com sucesso!  
{MENSAGEM}"





## Desafio final

*O que vamos fazer?*

- **Criar mensagem – Create**

Cada nova mensagem deve ser um **newMessage**

- Title, description devem ser inseridos pela pessoa usuária e todos esses itens precisam ser validados .
- Caso não passe um desses itens, informe um status (400) .





## Desafio final

*O que vamos fazer?*

- **Criar mensagem – Create**

Além disso, só é possível criar uma nova mensagem se o e-mail estiver registrado no nosso sistema.

Caso o e-mail não esteja no nosso sistema, deve retornar um status 404 e a seguinte mensagem : " Email não encontrado, verifique ou crie uma conta "



## Desafio final

*O que vamos fazer?*

- **Criar mensagem – Create**

O id de cada mensagem deve ser inserido automaticamente.





# Ler Mensagens





## Desafio final

*O que vamos fazer?*

- **Ler mensagem – Create**

**Endereço** :/message/:email

**Dados :**

– Email

A pessoa só conseguirá ler todas as mensagens se o e-mail estiver cadastrado em nosso sistema.



## Desafio final

*O que vamos fazer?*

- **Ler mensagem – Create**

Caso a operação seja realizada com sucesso, retornar um status (200) e a seguinte mensagem :

" Seja bem-vinde! {LISTA DE MENSAGENS}"





## Desafio final

*O que vamos fazer?*

- **Ler mensagem – Create**

Caso o e-mail não esteja no nosso sistema, deve retornar um status 404 e a seguinte mensagem : " Email não encontrado, verifique ou crie uma conta "



## Atualizar Mensagens





## Desafio final

*O que vamos fazer?*

- **Atualizar mensagem – Update**

**Endereço** /message/:id

**Dados :**

- Title
- Description



## Desafio final

*O que vamos fazer?*

- **Atualizar mensagem – Update**

Caso o operação seja realizado com sucesso, retornar um status (200) e a seguinte mensagem :

" Mensagem atualizada com sucesso  
{ MENSAGENS ATUALIZADA}"





## Desafio final

*O que vamos fazer?*

- **Atualizar mensagem – Update**

Caso o id da mensagem não seja encontrado, retornar a seguinte mensagem :

" Por favor, informe um id válido da mensagem"



## Desafio final

*O que vamos fazer?*

- **Atualizar mensagem – Update**

Caso a operação seja realizada com sucesso, retornar um status (200) e a seguinte mensagem :

" Mensagem atualizada com sucesso !  
{MENSAGEM ATUALIZADA}"





## Deletar Mensagens



## Desafio final

*O que vamos fazer?*

- **Deletar mensagem – Delete**

**Endereço** /message/:id

Caso o id da mensagem não seja encontrado no nosso banco, retorne a seguinte mensagem :

" Mensagem não encontrada, verifique o identificador em nosso banco"





## Desafio final

*O que vamos fazer?*

- **Deletar mensagem – Delete**

Caso a operação seja realizada com sucesso, retornar um status (200) e a seguinte mensagem :

" Mensagem apagada com sucesso"



## Critérios de Correção





O que será avaliado?

**MAS AINDA NÃO SEI NADA  
DE TAL COISA AINDA!!**

**Calma calma pessoal, nada de  
desespero...**



O que será avaliado?

**O projeto não deve ser incrementado com nada mais do que está sendo pedido.**





## O que será avaliado?

**Será avaliado tudo que foi ensinado em aula...**

Além disso, prestem atenção nos seguintes pontos:

1. Boas práticas
2. Código limpo
3. Utilização dos recursos da linguagem



# A entrega

## *Postagem e entrega*

1. Repositório no Github para o back-end
2. Link da aplicação no rodando no servidor – Deploy
3. Documentação no postman com o link do deploy

Data limite: **1 semana depois que o módulo acabar**





**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

**#BORAGROWDEVERS**

