RECINTOS DO ZOO

COMO BAIXAR O CÓDIGO E SUBMETER MINHA SOLUÇÃO?

Para completar a etapa do desafio você terá que baixar a estrutura do código aqui na Azure, resolver o desafio usando Javascript e entregá-lo no repositório no seu github.

BAIXANDO A ESTRUTURA

Para baixar a estrutura no formato zip, basta clicar neste link.

ENTREGANDO O DESAFIO

Após resolver o desafio e validá-lo com os testes (mais detalhes nos tópicos abaixo), você terá que criar um repositório **público** no Github com o **nome** de desafio-seuUsername-2024 (substitua "seuUsername" pelo seu usuário do GitHub) e colocar o código na **branch** main.

Se você ainda não teve contato com essa ferramenta, não tem problema. Separamos um material para lhe ajudar nessa etapa: Como usar Git e Github na prática.

O DESAFIO

Olá! Você foi contratado para ajudar na organização de um zoológico. Sua missão será construir a lógica para indicar os recintos onde novos animais se sintam confortáveis.

RECINTOS EXISTENTES

O zoológico possui os seguintes recintos disponíveis.

número	bioma	tamanho total	animais existentes
1	savana	10	3 macacos
2	floresta	5	vazio
3	savana e rio	7	1 gazela
4	rio	8	vazio
5	savana	9	1 leão

ANIMAIS

O zoológico só está habilitado a tratar dos animais abaixo. A tabela mostra o espaço que cada indivíduo ocupa e em quais biomas se adapta.

espécie	tamanho	bioma
LEAO	3	savana
LEOPARDO	2	savana

espécie	tamanho	bioma
CROCODILO	3	rio
MACACO	1	savana ou floresta
GAZELA	2	savana
HIPOPOTAMO	4	savana ou rio

REGRAS PARA ENCONTRAR UM RECINTO

1. Um animal se sente confortável se está num bioma adequado e com espaço suficiente para cada indivíduo
2. Animais carnívoros devem habitar somente com a própria espécie
3. Animais já presentes no recinto devem continuar confortáveis com a inclusão do(s) novo(s)
4. Hipopótamo(s) só tolera(m) outras espécies estando num recinto com savana e rio
5. Um macaco não se sente confortável sem outro animal no recinto, seja da mesma ou outra espécie
6. Quando há mais de uma espécie no mesmo recinto, é preciso considerar 1 espaço extra ocupado
7. Não é possível separar os lotes de animais nem trocar os animais que já existem de recinto (eles são muito apegados!). Por exemplo, se chegar um lote de 12 macacos, não é possível colocar 6 em 2 recintos.

ENTRADAS E SAÍDAS

1. O programa deve receber tipo e quantidade de animal (nessa ordem)
2. O programa deve retornar uma estrutura contendo a lista de todos os recintos viáveis ordenada pelo número do recinto (caso existam) e a mensagem de erro (caso exista)
3. A lista de recintos viáveis deve indicar o espaço livre que restaria após a inclusão do(s) animal(is) e o espaço total, no formato "Recinto nro (espaço livre: valorlivre total: valortotal)"
4. Caso animal informado seja inválido, apresentar erro "Animal inválido"
5. Caso quantidade informada seja inválida, apresentar erro "Quantidade inválida"
6. Caso não haja recinto possível, apresentar erro "Não há recinto viável"

EXEMPLOS

Entrada para um caso válido

"MACACO", 2

Saída

{

recintosViaveis: ["Recinto 1 (espaço livre: 5 total: 10)",

"Recinto 2 (espaço livre: 3 total: 5)",

"Recinto 3 (espaço livre: 2 total: 7)"]

}

Entrada para um caso inválido

"UNICORNIO", 1

Saída

{

erro: "Animal inválido"

}

O CÓDIGO

Você está recebendo uma estrutura básica para desenvolver a lógica do desafio. O arquivo principal está localizado dentro da pasta src e se chama recintos-zoo.js. Você pode desenvolver a sua lógica criando outros arquivos, métodos e até mesmo outras classes, porém o resultado deve poder ser obtido através do método analisaRecintos.

ALERTA: É importante que essa estrutura básica não seja alterada, pois as etapas automáticas da nossa validação dependem disso. Conseguir executar os passos descritos mais adiante na seção VALIDANDO A SOLUÇÃO também ajudará você a verificar que seu código segue a estrutura definida.

Exemplo de chamada

new RecintosZoo().analisaRecintos('MACACO', 2);

INSTALANDO E RODANDO NA SUA MÁQUINA

1. Instalar o Node
2. Instalar dependencias do projeto com o seguinte comando:

npm install

VALIDANDO A SOLUÇÃO

Junto com a estrutura básica você está recebendo alguns cenários de testes no arquivo recintos-zoo.test.js para auxiliar na validação da sua solução. Recomendamos que você crie mais casos de teste para aumentar a confiabilidade da sua solução. Para testar sua solução com os cenários existentes ou novos, rode o seguinte comando:

npm test

Para saber mais consulte a Documentação do Jest.

VALIDANDO A ENTREGA

Para garantir que seu desafio vai ser considerado entregue, revise os seguintes pontos:

GIT

O repositório deve ser **público** e ter o **nome** e **branch** indicados na seção ENTREGANDO O DESAFIO.

Para verificar que o repositório é público, deslogue-se do github e tente ver o código. Se conseguir, nós também conseguimos! Lembrando que vamos usar o link para o usuário informado durante o cadastro na Gupy. Veja como alterar a visibilidade.

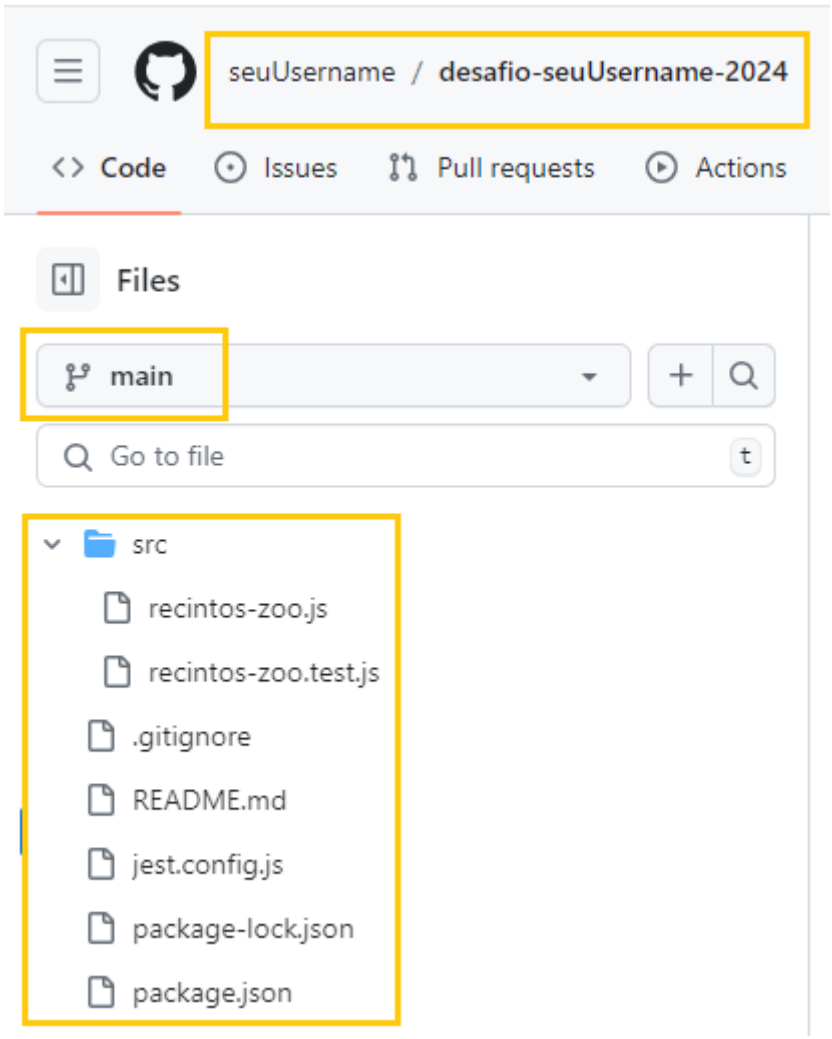
CÓDIGO

A solução deve ser entregue em **javascript** e a **estrutura de pastas e arquivos** deve seguir o indicado na seção O CÓDIGO.

O **export** da classe deve ser mantido da seguinte maneira para compatibilidade com o arquivo de testes:

export { RecintosZoo as RecintosZoo };

Se todos os passos forem seguidos corretamente, você terá um repositório como o da figura abaixo (lembrando que é permitido criar mais arquivos), onde seuUsername é o seu usuário do GitHub, que você informou no questionário da Gupy.


seuUsername / desafio-seuUsername-2024
Code
Issues
Pull requests
Actions
Files
main
Go to file
t
src
recintos-zoo.js
recintos-zoo.test.js
.gitignore
README.md
jest.config.js
package-lock.json
package.json