Atividade 04

Desenvolvimento de Software para WEB
Projeto de Interface Web

Conteúdo: Promises, Async/Await, Context

Obs.: Faça os exercícios abaixo usando o CodeSandBox ou o ambiente de trabalho instalado em sua máquina (VSCode + Node, por exemplo).

01 - Implemente uma **Promise** que demora 3 segundos para concluir (use o setTimeout).

Dentro do setTimeout compute randomicamente (Math.random) um número aleatório (chamado de **NUM**) de 1 a 10 (incluindo estes). Caso o NUM seja:

1 : ERRO DE CONEXÃO (chame o reject)
2 : ERRO DE DADOS INVÁLIDOS (chame o reject)
3 a 10 : OK (chame o resolve)

No caso de **resolve** e **reject,** veja abaixo o que deve ser retornado:

O **resolve** da Promise deve retornar um array (vetor) de objetos JSON que representem alunos de um curso, por exemplo:

```
[
        {id:1, nome: “Beltrano”, ira: 6.7},
        {id:2, nome: “Fulano”, ira: 8.3},
        {id:3, nome: “Sicrano”, ira: 5.2}
]
```

No exemplo acima temos apenas 3 alunos mas sinta-se livre para retornar mais, caso queira. O mínimo são 3.

O **reject** deve retornar um único objeto JSON com o número **NUM** computado e a mensagem de “ERRO”.

Implemente a chamada da Promise descrita acima dentro do useEffect de um componente React chamado Questao01.jsx

No componente Questao01, leia os valores usando o then/catch e armazene os dados (no caso de um resolve), dentro de uma variável de estado. Use um Array.map para ler os dados vindos do resolve e os imprima em tela, no JSX.

No caso do reject (catch), apenas imprima a mensagem de erro em Alert.

02 - Implemente uma componente Questao02.jsx com a mesma lógica de Questao01.jsx. No entanto, não use then/catch e sim Async/Await para tratar os dados (não esqueça do try/catch para poder pegar a mensagem de erro).

03 - Ao usar o comando no console:

```
fetch("https://restcountries.com/v3.1/name/brazil")
.then((res)=>res.json())
.then((json)=>console.log(json[0].population))
```

é impresso a população do Brasil de acordo com https://restcountries.com/#api-endpoints-using-this-project. Implemente um componente JSX que armazene o nome de um país no Context e imprima sua população e nome ao ser carregado (useEffect). Veja na API quais são os nomes válidos para se usar na URL (no caso do Brasil é “brazil”). Teste o seu componente com outros países. Use o FETCH!