

PRIMEIRA AVALIAÇÃO PARCIAL - 2022.2
DESENVOLVIMENTO PARA WEB E PROJETO DE INTERFACES MÓVEIS
JEFFERSON DE CARVALHO SILVA

(2,5 pts) Questão 1 -) Implemente um componente chamado “**Questao01.jsx**” que internamente use dois outros componentes: “**Questao01A.jsx**” e “**Questao01B.jsx**”. O código de “**Questao01.jsx**” ficaria mais ou menos assim:

```
<Questao01>
  <Questao01A />
  <Questao01B />
</Questao01>
```

<Questao01> deve ser chamado dentro de **App.js** e usar “**props.children**” para exibir <Questao01A> e <Questao01B>.

Em <Questao01A> você deverá exibir seus dados pessoais como **nome, sobrenome e curso**.

Em <Questao01B> você deverá exibir a **lista dos nomes das disciplinas** (pelo menos 3) que você está matriculado neste semestre (pode inventar mais disciplinas, caso queira).

Atenção: dentro do componente <Questao01B> você terá uma variável parecida com essa:

let disciplinas = ['FUP', 'Estruturas de Dados', 'Introdução à Computação'].

De alguma forma (usando o map, por exemplo), você deverá ler essa lista e exibir os nomes das disciplinas em <Questao01B>.

(2,5 pts) Questão 02 -) Implemente um componente chamado “**Questao02.jsx**” que implemente a seguinte interface:

Número 1 Número 2

+ - * /

Resultado:

Onde “**Número 1**” e “**Número 2**” são campos de texto as operações de soma, subtração, multiplicação e divisão são botões que ao serem clicados mostram o resultado logo abaixo depois de “Resultado: ”. <Questao02> deve ser chamado dentro de App.js (observação: USE ESTADOS).

(2,5 pts) Questão 03 -) A URL <https://pokeapi.co/api/v2/pokemon?limit=10&offset=0> exibe os 10 primeiros pokemons no seguinte formato:

```
count: 1154
next: "https://pokeapi.co/api/v2/pokemon?offset=10&limit=10"
previous: null
▼ results: [] 10 items
  ▼ 0: {} 2 keys
    name: "bulbasaur"
    url: "https://pokeapi.co/api/v2/pokemon/1/"
  ▼ 1: {} 2 keys
    name: "ivysaur"
    url: "https://pokeapi.co/api/v2/pokemon/2/"
  ▼ 2: {} 2 keys
    name: "venusaur"
    url: "https://pokeapi.co/api/v2/pokemon/3/"
  ▼ 3: {} 2 keys
    name: "charmander"
    url: "https://pokeapi.co/api/v2/pokemon/4/"
  ▼ 4: {} 2 keys
```

Implemente um componente chamado Questao03.jsx que deverá ser chamado em App.js e que irá exibir os **nomes dos pokemons** em tela.

A lista de pokemons deve ser uma variável de ESTADO (useState).

Para finalizar a questão, implemente um BOTÃO que ao ser clicado/pressionado, exiba os próximos 10 pokemons.

Dica: veja que para isso, basta modificar a variável “offset” da URL. Implemente “offset” como uma variável de estado a qual é somado 10 todas vez que for clicado no BOTÃO. Coloque “offset” como uma variável que dispara um EFEITO (useEffect) para recarregar novamente a lista de pokemons (chamada da API).

(1,5 pts) Questão 04 -) A URL <https://restcountries.com/v2/region/africa?fields=name,population> retorna todos os países da África com os dados “name” e “population”. Faça um programa que leia essa lista e indique na interface o país mais populoso da África.

(1,0 pts) Questão 05 -) Modifique a **Questão 4**: Crie dois botões que mudam o valor “africa” da URL. O primeiro botão deve mudar o valor “africa” para “americas” e recalcular o país **MAIS** populoso da América. O segundo botão deve modificar o valor para “asia” e recalcular o país **MENOS** populoso.

OBSERVAÇÕES:

Pessoal de WEB: usar a biblioteca do Axios e também o BOOTSTRAP;
Pessoal de Mobile: usar o Fetch e Estilos.

ENVIE APENAS A URL DO SEU PROJETO DO GITHUB OU SNACK (PARA O PESSOAL DE MOBILE) OU CODESANDBOX (PARA O PESSOAL DE WEB). ENVIE A URL EM UM ARQUIVO TXT PARA O SIGAA.