Imersão Desenvolvimento Web

Modulo II - Dia 3

Taylane Brandão Renan Verissimo Laion Luiz

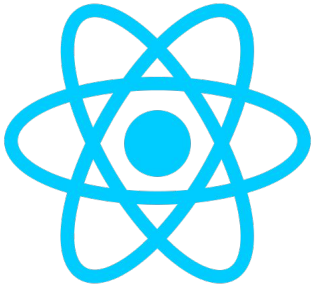
Indice:

- Hooks I
- Hooks II
- Pratica

Hooks I

Parte I

Afinal, o que são Hooks ???



- Eles foram introduzidos no **React** para facilitar nossa vida.
- Permitem o gerenciamento de **estado**.
- Além de ajudar na comunicação entre os **componentes** e com o **DOM**.
- São uma espécie de **função** especial dentro de um **componente**.

Lembrete

Precisamos sempre importar os Hooks que vão ser utilizados.

UseState

- Permite o gerenciamento do **estado** de um componente.
- Guardando o **estado**:

```
const [variavel, setVariavel] = useState(valorInicial);
```

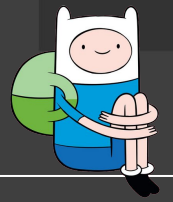
- Obtendo o **estado**:

```
<elemento>{variavel}</elemento>
```

- Gravando o **estado**:

```
setVariavel(valor);
```

É hora de codar!



- Crie uma **variavel** com useState();
- Atribua um valor a ela (**qualquer tipo**);
- Mostre em tela essa **variavel**;

useEffect

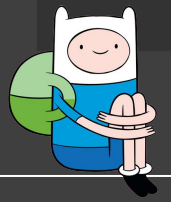
- Permite capturar acontecimentos (**efeitos**) da aplicação.
- Executar **efeitos** secundários;
- Rodando toda vez que o componente faz **Render**:

```
useEffect(() => {
  //Codigo executado em todo render
});
```

- Rodando na primeira vez que o componente faz **Render**:

```
useEffect(() => {
  //Codigo executado no primeiro render
},[]);
```

É hora de codar!



- Vamos fazer um **Contador**.
- Crie um botão que incremente o estado do contador.
- Mostre em tela esse **Contador**.

Hooks II

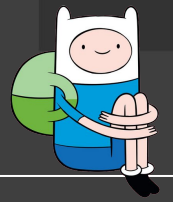
Parte III

Voltando aos formularios

- Imagine que temos um input e queremos obter o valor dele.
- Como fazer isso se não podemos acessar o DOM diretamente ?
- Utilizando os hooks para atribuir e modificar o valor do input.

```
<input
    type="text"
    value={variavelValor}
    onChange={e => setVariavelValor(e.target.value)}
/>
```

É hora de codar!



- Criar um **abaixo-assinado** com as seguintes funções:
- Deve salvar **nome** e **cpf** da pessoa que assinar.
- Deve ser possível encontrar certo assinante pelo **cpf**

Bonus Stage

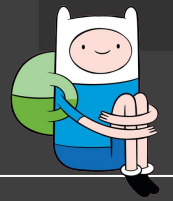
Parte III

Carregando...

- Telas de carregamento, apesar de irritantes, são extremamente **úteis**.
- Enquanto mostramos o **Loading** podemos terminar de processar informações, popular a tela, fazer uma requisição ao backend e etc...
- Como até então nós não fazemos tarefas tão lentas, vamos usar a função de c**ontagem regressiva** para atrasar um pouco as coisas:

```
const meuTimeout = setTimeout(minhaFuncao, tempoEmMilisegundos);
```

É hora de codar!



- Crie uma **variável** para guardar se a tela está carregando ou não;
- Crie um ***timeout*** que altere essa **variável** ao fim de **5 segundos;**
- Exiba um **p** indicando o **carregamento** da página;

useContext

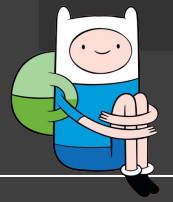
- Usado para manter um “**Contexto**”;
- Facilita a transmissão de dados entre os componentes;
- Dipara um **render** sempre que o valor for alterado;
- Ex:

```
const { variavelContext } = React.createContext();
const variavel = useContext(variavelContext);
```

Pratica

Parte III

É hora de codar!



- Vamos recriar nossa calculadora no React;


VICTORY