

## Lista 3 de Exercício – React

Siga as orientações abaixo:

- 1- Crie um projeto com o nome projeto-componentes
- 2- Entre na pasta do projeto e instale os pacotes react-icons (de ícones) e react-toastify (de alerta)

**cd** projeto-componentes

**npm install react-icons --save**

**npm install react-toastify --save**

**\*\***Depois olhe o arquivo package.json e veja as dependências criadas, pois você as salvou neste arquivo na instalação dos pacotes

**\*\*** Veja os ícones em <https://react-icons.github.io/react-icons/>

**\*\*** Veja as informações do pacote de mensagens <https://fkhadra.github.io/react-toastify/introduction>

- 3- Inicie o projeto e altere o index.js para a nova versão do React, importando o createRoot e o usando para renderizar.

Também importe os elementos ToastContainer e estilo css do react-toastify

```
import './index.css';
import App from './App';
import { createRoot } from 'react-dom/client';
import 'react-toastify/dist/ReactToastify.css';
import { ToastContainer } from 'react-toastify';

const root = createRoot(document.getElementById('root'));
root.render(
  <>
    <ToastContainer autoClose={false} />
    <App tab="home" />
  </>
);
```

**\*\*** Não se esqueça que usa o fragment ( <> </> ) para renderizar mais de um componente

**\*\*** <https://reactjs.org/docs/rendering-elements.html> (React 18)

- 4- Crie um Componente (Header) que mostre uma mensagem de cabeçalho – informando por meio de uma mensagem o texto “Cadastro de clientes” e mostrando um ícone
  - a. Para usar o ícone o escolha no site do react e o importe, conforme exemplo abaixo:  
import { DiAndroid } from "react-icons/di"  
Para usar é só informar o nome  
<DiAndroid color="#00FF22" />

- 5- Altere seu APP:
- Apague o código de retorno
  - Faça aparecer um Componente que criou
  - Crie um formulário** os dados nome, endereço, cidade, estado e telefone e um submit ao final.
- 6- No início do formulário desvie a execução do envio dos dados para a função enviarValidar()
- `<form onSubmit={enviarValidar}>`  
Faça a função de forma que quando a pessoa pressionar o submeter apareça “Dados Cadastrados”

```
function enviarValidar(e){  
  e.preventDefault()  
  window.alert("Dados cadastrados")  
}
```

**\*\* Não se esqueça que pode ser qualquer nome a função inclusive a padrão handleSubmit**  
**\*\* O e se refere ao evento manipulado. Pode-se colocar e, event ou outra definição. Para que não sejam enviados os dados para outra página daremos o preventDefault, que cancela a ação do submit.**

- 7- Para utilizarmos os dados do componente em outro local, como os dados cadastrados, precisamos ter o estado. Para que isso ocorra:
- Importe o useState `import {useState} from 'react'`
  - acrescente no início da função App, antes do return constantes para cada entrada do formulário. Veja abaixo:

```
const [nome ,    setNome] = useState();  
    |              |  
Nome do campo   função de mudança de valor
```

Depois também crie as funções de alteração de cada campo, como abaixo

```
function nomeAlterar(e){  
  setNome(e.target.value); } }
```

**\*\* Não se esqueça que as funções estão definidas na entrada do formulário**  
`<input type = 'text' placeholder='Nome...' value= {nome} onChange={nomeAlterar} ></input>`

**\*\* {nome} refere-se ao valor atual do nome e {nomeAlterar} executará a função de atualização dos valores do ano.**

8- Agora vamos fazer a validação e usar o toast

a) Importe ele no App

```
import {toast} from 'react-toastify'
```

b) Altere a função enviarValidar. Acrescente a verificação do nome.


```
if(nome === undefined)
```

```
    toast.error("Erro. Nome não pode estar vazio")
```

```
else
```

```
    toast.success("Dados Cadastrados! ")
```

Nome:  Telefone:

 Erro. Nome não pode estar vazio

9- Desafio Relógio: Crie um novo componente Relogio.js

O relógio deve fazer aparecer a hora atual. Esse data e hora deve aparecer no Componente Header. Veja os slides do arquivo "Introdução NPM\_React.pdf"

10- Melhore a aplicação 😊