

Apostila AV2 – CODIFICAÇÃO FRONT-END

Instalação do React

1. Abra o terminal e digite o seguinte comando para instalar o Create React App, uma ferramenta que configura um novo projeto React para você:

```
npm install -g create-react-app
```

2. Crie um novo projeto React com o seguinte comando: `npx create-react-app nome-do-projeto`

4. Entre na pasta do projeto e inicie o servidor de desenvolvimento:

```
cd nome-do-projeto
```

```
npm start
```

Inserindo Imagens

5. Para inserir uma imagem em seu componente React, você pode usar a tag `` e o módulo `require` para importar a imagem. Aqui está um exemplo:

```
<img src={require('./caminho/para/sua/imagem.png')} alt="Descrição da imagem" />
```

Criação de Links com React Router

6. Para criar links entre suas páginas, você pode usar o pacote `react-router-dom`. Primeiro, instale o pacote com o seguinte comando:

```
npm install react-router-dom
```

7. Depois, você pode usar o componente `<BrowserRouter>` e `<Routes>` para definir suas rotas. Aqui está um exemplo básico:

```
import React from 'react'; import { BrowserRouter, Routes, Route
} from 'react-router-dom'; import Home from './Home'; function
App() {

  return (

    <div className="App">

      <Menu />

      <BrowserRouter>

        <Routes>

          <Route path="/" element={<Home />} />

          <Route path="/contato" element={<FormularioContato />} />

        </Routes>

      </BrowserRouter>

      <Footer />

    </div>    );}

export default App;
```

CSS Básico

8. Aqui estão alguns comandos básicos de CSS que seus alunos podem achar úteis:

- `float: left;` **OU** `float: right;`: Estes comandos são usados para alinhar elementos à esquerda ou à direita, respectivamente.
- `background-color: #cor;`: Este comando é usado para definir a cor de fundo de um elemento.
- `display: flex;`: Este comando é usado para criar um container flexível que torna fácil alinhar elementos lado a lado.
- `padding: 10px;`: O comando padding é usado para criar espaço ao redor do conteúdo do elemento, dentro de qualquer borda definida. Você pode especificar diferentes valores para cada lado (topo, direita, inferior, esquerda) ou usar um único valor para todos os lados.
- `max-width: 100%;`: O comando max-width é usado para definir a largura máxima de um elemento. Isso pode ser útil para garantir que o elemento não fique muito grande em telas grandes.

- `line-height: 1.5;`: O comando `line-height` é usado para definir a altura da linha de texto. Isso pode ser útil para ajustar a legibilidade do texto.
- `align-items: center;`: O comando `align-items` é usado com um container flexível para alinhar os itens ao longo do eixo transversal do container. Por exemplo, `align-items: center;` irá centralizar os itens verticalmente se a direção do flexível for linha (a padrão).
- `border-bottom: 1px solid black;`: O comando `border-bottom` é usado para definir uma borda na parte inferior de um elemento. Você pode especificar a espessura, o estilo e a cor da borda.
- `margin-bottom: 20px; /* Espaçamento inferior */`
- `justify-content: space-between; /* Para distribuir o espaço entre os elementos */`
- `width: calc(25% - 10px); /* Ajustado para 25% para que haja espaço para todas as imagens */`
- `object-fit: cover; /* Mantém a proporção da imagem e corta o excesso para preencher o espaço */`
- `display: flex; /* exibe os elementos em linha */`
- `flex-direction: column; /* alinha os elementos verticalmente */`

Formulário exemplo:

```
return (
  <div className="formulario-container">
    <form onSubmit={handleSubmit} className="formulario">
      <fieldset>
        <legend>Quem você deseja contactar?</legend>
        <div className="input-group">
          <input type="text" name="nome" placeholder="Digite o seu
nome" />
        </div>
        <div className="input-group">
          <input type="number" name="numPessoas"
placeholder="Quantidade de pessoas" />
        </div>
        <div className="input-group">
          <input type="email" name="email" placeholder="Digite o seu
email" />
        </div>
      </fieldset>
    </form>
  </div>
)
```

```

        <div className="input-group">
            <label htmlFor="dtaNascimento">Data que você quer ver a
            pessoa</label>
            <input type="date" id="dtaNascimento" name="dtaNascimento" />
        </div>
    </fieldset>

    <fieldset>
        <legend>Complete a frase: "Filho de cabaço, cabaço
    ____"</legend>
        <div className="checkbox-group">
            <input type="checkbox" id="fixacao" name="frase" value="é" />
            <label htmlFor="fixacao">é</label>
        </div>
        <div className="checkbox-group">
            <input type="checkbox" id="projecao" name="frase" value="tem"
    />

            <label htmlFor="projecao">Jamais será cabaço</label>
        </div>
        <div className="checkbox-group">
            <input type="checkbox" id="cheiro" name="frase" value="verde"
    />

            <label htmlFor="cheiro">Alienígena</label>
        </div>
        <div className="checkbox-group">
            <input type="checkbox" id="cor" name="frase" value="bonito"
    />

            <label htmlFor="cor">Bonito</label>
        </div>
    </fieldset>

    <fieldset>
        <legend>Dê uma nota para seu professor:</legend>
        <div className="input-range">
            <label>O professor é top?</label>
            <input type="range" min="1" max="100" defaultValue="50"
    className="SatisfacaoGeral" id="SatisfacaoGeral" />
        </div>
    </fieldset>

    <button type="submit" className="botao-enviar">Enviar</button>
</form>

    {enviado && <p>Formulário enviado com sucesso!</p>}
</div>
    );
}

```

Alerta exemplo:

```
const handleSubmit = () => {      alert('MENSAGEM QUE  
SERÁ APRESENTADA AO USUARIO!');  
}
```