**React**

1. **Etapas do curso**

- Introdução

- Fundamentos de React

- Avançando em React

- React e CSS

- Formulários com React

- Projeto: Secret Word

- Requisições HTTP com React (API)

- React Router (biblioteca)

- Context API

- Projeto: MiniBlog(bd)

- Entenda os hooks de React

- Projeto: ReactGram

**Conteúdos Adicionais**

- Javascript Moderno

-Introdução a APIs

- Setup de TypeScript

- React com TypeScript

- Projeto de React com TS

**- Atualizações**

* + 1. **O que é React?**

React é uma biblioteca JS para desenvolvimento de aplicações front-end;

Framework: segue maneira de criar aplicação para ter sucesso, já têm-se a estrutura pronta.

Biblioteca: nós mesmos criamos a base e o fluxo, não tem uma estrutura rígida a ser seguida.

Aplicações são chamadas **SPA** (Single Page Application), um carregamento, reload para diversas páginas;

Arquitetura do React é baseada em componentes;

Com o React pode-se criar a aplicação, ou inserir ele a uma;

É mantido pelo Facebook.

* + 1. **O que é Node.js?**

- O Node.js é uma runtime de JS, ou seja, o js é escrito, um compilador é adicionado e utiliza uma linguagem de mais baixo nível para fazer as execuções. Deixando assim os programas mais rápidos, performáticos;

- Uma biblioteca utilizada por um compilador durante a execução do programa;

- Biblioteca que está contruída na V8 engine, que faz com que JS vira C++, por ser uma linguagem de mais baixo nível;

- Possibilita a criação de softwares em JS no lado do servidor;

- Código JS rodando em C++ garantindo uma alta performance.

* + 1. **O que é npm?**

- Gerenciador de pacotes do Node, serve para que seja possível baixar bibliotecas de terceiros;

- A maioria dos projetos que serão criados em Node, estarão sendo iniciados, gerenciados pelo npm (Node Package Manager) Gerenciador de Pacotes do Node;

- Serve para executar determinados scripts no programa, com isso, podem ser criados setups mais complexos;

- Dificilmente um software em Node.js não utiliza o npm;

- Os módulos externos ficam numa pasta chamada **node\_modules**(padrão de todo projeto Node);

- Ela deve ser descartável e isso não poderá interferir no projeto, ou seja, a cada instalação do projeto são baixados todos os pacotes novamente.

* + 1. **Instalação Node Windows**

**-** O dowload do Node.js é feito no site oficial: nodejs.org;

**-** será baixado um arquivo .msi, o instalador;

**- npm** vem junto com o Node;

**-** Após a instalação, o Node e o npm serão testados em um terminal, para assim a instalação ser validada;

**-** Os binários precisam estar no **PATH**, passo a ser seguido na instalação do Node.

* + 1. **Hello World em React**

- Executor de scripts do Node: npx(x vem de executar);

- Dificilmente cria-se uma aplicação do zero, mas, para gerar uma, basta usar o comando **npx create-react-app<nome>**

- npm start – inicia a aplicação.

npm start

npm run build

npm test

npm run eject

‘cd hello world – abrindo uma pasta

npm start – iniciando a aplicação'

***outra forma de criação de uma aplicação(mais ágil)***

npm create vite@latest

cd <nome>

npm i

npm run dev

* + 1. **Estrutura base do react**

Há algumas pastas e arquivos chaves para o desenvolvimento do react;

**- node\_modules:** onde as dependências do projeto ficam-pode ser removida a qualquer momento, logo, dentro dela não devem ser colocados códigos;

**- public:** assets estáticos e html de inicialização;

**- src:** onde serão programados os apps, principal pasta – components, estruturas responsáveis por fazer requisições assíncrona, configurações de adaptação;

**- src / index.js:** arquivo de inicialização do react;

**- src / App.js:** componente principal da aplicação.

* + 1. **Extensão para React**

Extensão no vs code que ajuda a programar em react: **marketplace**;

Nome: **ES7 + React/Redux/React-Native snippets**;

Pode ser criado muito código com poucos atalhos.

**Preparando o Emmet para o React**

Extensão nativa do VS Code: ajuda a escrever html mais rápido;

Não vem configurada para o React;

File>Settings>Extensions e procurar Emmet;

Incluir: Javascript-javascriptreact;

**Livros que podem auxiliar:**

JS

<https://www.amazon.com.br/gp/product/B016N7G8EK?&linkCode=sl1&tag=horadecodar04-20&linkId=62464e9456c337e90e1471bd13b54ebd&language=pt_BR&ref_=as_li_ss_tl>

React

<https://www.amazon.com.br/React-Praticando-Desenvolva-Aplica%C3%A7%C3%B5es-Biblioteca/dp/6586057396?__mk_pt_BR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=21F52A2MJJQPA&keywords=react&qid=1681470148&sprefix=react,aps,329&sr=8-1&linkCode=sl1&tag=horadecodar04-20&linkId=442f338d162184060295952eee0c6af6&language=pt_BR&ref_=as_li_ss_tl>

1. **Fundamentos do React**
   * 1. **Criando componentes**

Na maioria dos projetos os componentes ficam em uma pasta chamada **componentes**, que devemos criar;

Geralmente nomeados com a **camel case: FirstComponent.js**;

No arquivo **criamos** **uma** **função**, que contém o código desse componente (lógica e template);

Exportar função para reutilizá-lo.

* + 1. **Importando componente**

Importação – maneira de reutilizar componentes;

Sintaxe: **import nomecomponente from ‘./components/nomecomponente’**;

Para colocar o componente importado de outro, coloca-se em forma de tag: **<FirstComponent />**

Finaliza importação, importando FirstComponent em App.

import FirstComponent from './components/FirstComponent';

//importação de FirstComponent para o arquivo JS FirstComponent

Exibindo o elemento no browser

<FirstComponent />