O que é o node

Node.js é um runtime de Javascript, criado pelo desenvolvedor Ryan dahl em 2009. O node .js, foi desenvolvido em cima do moto Javascript V8 – engine criada pelo google e utilizada nos navegadores chrome.

O que é o NPM

O NPM(Node Package Manager), é o gerenciador de pacote padrão do node. Ele é utilizado como gerenciamento de pacotes, fluxo de trabalho em linguagens de programação e ferramenta para construção de front-ends em aplicativos e websites.

<https://www.npmjs.com/>

O que é o Yarn

O yarn é um gerenciador de pacotes para aplicar comandos prontos ao código de uma aplicação.

Yarnpkg.com

<https://yarnpkg.com/getting-started/install>

para instalar no terminal (npm install -g yarn )

O que é o React DevTools

É uma extensão disponível para o Chorme, Firefox e também como um aplicativo independente que permite inspecionar a hierarquia de componentes do React nas ferramentas do desenvolvedor do navegador.

<https://github.com/facebook/react/tree/main/packages/react-devtools-extensions>

O porquê utilizar extensões

Costumam ser leves, não comprometem o funcionamento do software e são de fácil instalação. Cada dev tem a sua maneira de programar e é por isso que as extensão podem ser ótimas aliadas! Com elas, conseguimos adaptar o VSCode as nossas necessidades, criando na rotina de trabalho e estudo um ambiente mais funcional.

O que é o REACT

React JS é uma biblioteca JavaScript para criação de interfaces de usuários. Criado em 2011 pelo time do Facebook, o React surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no feed de notícias da rede social, entre eles chat, status, listagem de contatos e outros.

A história do React

A principio, todas essas atividades, chamadas de estados, tinham uma descrição muito complexa. Com o React, esta descrição torna-se mais simples, bem como também é simplificada a conexão entre HTML, CSS e JavaScript e todos os componentes de uma página.

Por ter demonstrado grande eficiência, nos anos que se seguiram o react foi incorporado à interface de outras redes sociais do grupo, como o instagram e, em 2013, seu código foi aberto à comunidade, dando início a sua popularização.

Framework e Biblioteca

Biblioteca – Esse é o recurso mais utilizado no mundo da programação e muitas pessoas nem se dão conta do quanto utilizam. A ideia da biblioteca é compartilhar soluções por meio de funções ou métodos.

Por exemplo: se você tiver que fazer um trabalho de matemática, por exemplo, poderá ir até uma biblioteca física, pegar um livro e utilizar equações desenvolvidas no livro. Então, não será preciso desenvolver as equações desde o início.

Exemplo de Biblioteca

-Moment.js

-Chart.js

-Voca

-mo.j

-React

Frameworks

O framework nada mais é do que uma ferramenta que vai te ajudar a ter como único objetivo focar em desenvolver o projeto, não em detalhes de configurações.

Exemplo de Frameworks

-Angular

-Vuex

-Ionic

-next

-Express

-LoopBack

<https://pt-br.reactjs.org/>

Ciclo de Vida de um componente

Diagrama

Descrição gerada automaticamente

O que é o React CLI

Create React App é um ambiente confortável para aprender React, e é a melhor maneira de começar um single-page application em React.

Além de configurar seu ambiente de desenvolvimento para utilizar as funcionalidades mais recentes do JavaScript, ele fornece uma experiencia de desenvolvimento agradável, e otimiza o seu tempo de produção.

O que é o Style componentes

Basicamente, é uma biblioteca que possibilita escrever códigos CSS dentro do JavaScript. Com ele, se pode criar websites bonitos e funcionais. Além disso ganhar mais agilidade e precisão no desenvolvimento web.

O que São HOOKS

Até a versão 16.7 do React algumas funcionalidades só eram possíveis de ser acessadas através de classes (como, por exemplo, o lifecycle). Apesar da possibilidade de criar componentes a partir de função, até essa versão, eles só recebiam propriedades, não podendo acessar todas as funcionalidades do React, como as classes.

Na versão 16.8 do React, foram lançados os hooks, que permitem o uso de vários recursos de forma simples através de funções. Eles também ajudam a organizar a logica utilizada dentro dos componentes.

useState

Serve para alteração de estado dentro do componente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

useEffect

Ele vai trabalhar com o ciclo de vida do componente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

useMemo

Memoriza alguma ação, um cálculo, algum valor por exemplo

Salvar referência em variáveis (const, var, let)

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

useCallback

Memoriza uma função

Salvar referência de função

Texto

Descrição gerada automaticamente

O que é TypeScript

É um superset de JavaScript, ou seja, um conjunto de ferramentas e formas mais eficientes de escrever código Javascript adicionando recursos que não estão presentes de maneira nativa na linguagem.

Porque Utilizar

Trazendo diversos benefícios, mas podemos destacar o potencial de detecção de erros durante o desenvolvimento de projetos e a possibilidade de incluir a inteligência da IDE que está sendo usada. Isso reflete num ambiente muito mais ágil e seguro enquanto o código está sendo digitado pelo usuário.

Conceitos Básicos

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, chat ou mensagem de texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente