**Anotações durante a NLW/Together – ReactJS**

**AULA-01**

**► SPA – Single Page Application -** *Aplicação de uma só página* • Conceito de como a aplicação se comporta quando precisamos de mais rotas, páginas dentro do projeto.  
 • Distribuição de responsabilidades, back-end separado do front-end.  
 • Diferente do método tradicional, no SPA o back-end retorna os dados em formato JSON.

► **Firebase** – produto construído pela Google, com uma série de ferramentas para auxiliar desenvolvimento de aplicações web e mobile. Por exemplo: autenticações, banco de dados e outras funcionalidades. A maioria das ferramentas são gratuitas. Vamos utilizar os seguintes serviços:  
 • Autenticação  
 • Banco de Dados

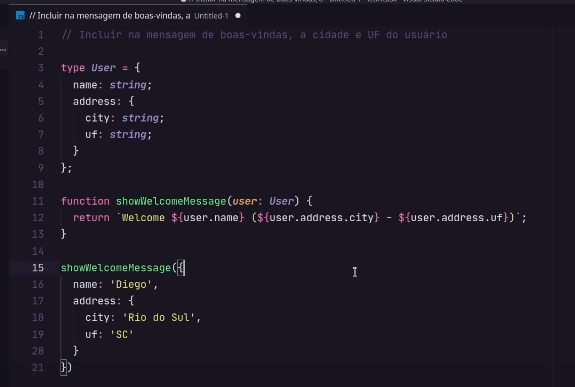
► **React** – totalmente front-end, biblioteca de criação de interfaces.  
 • Componente  
 • Propriedade  
 • Estado  
  
► Integrar React e Firebase.

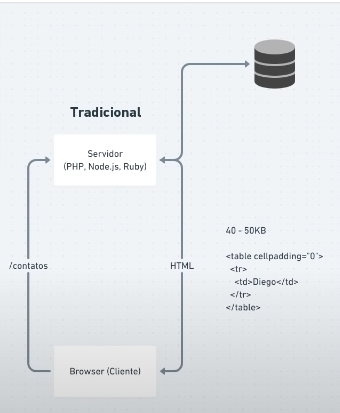
► Aprender variáveis ambientes.

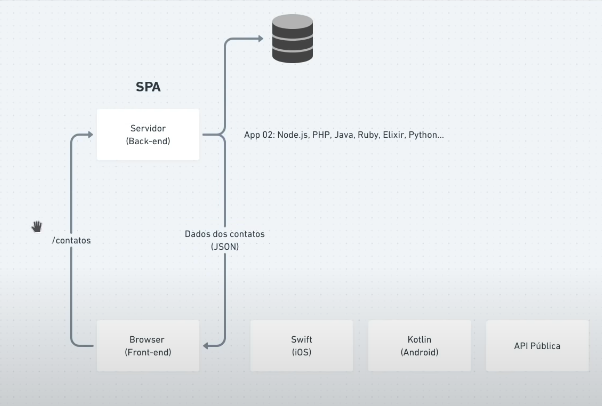
► **Node** – permite executar JS fora do ambiente do navegador. Por padrão, o JS é uma linguagem construída para ser executada dentro do browser, porém, com o node permite executar JS em outros ambientes sem ser o browser.

► **Yarn** – gerenciador de pacotes. Instalar códigos dentro do projeto

► **TypeScript -** tipagem







**Create React App** – ferramenta para iniciar um projeto REACT

***$$ yarn create react-app letmeask --template typescript***

package.json – arquivo onde guarda nome, versão do projeto e as dependências.

Tsconfig.json - Arquivo de configuração do TypeScript

Fast-refresh – permite que sua aplicação tenha sua parte visual atualizada sem precisar ficar dando reload no browser.

Index.html é a única página da nossa aplicação e, ÚNICO HTML (no caso do React, por causa do SPA). O conteúdo da página vai ser construída a partir do JS.

O Create React app – executa o index.html e o index.tsx é o primeiro arquivo JS executado na aplicação.

Index.tsx – importa o React, ReactDOM

DOM = Document Object Model -> representação do HTML através de um objeto dentro do JS.

HTML dentro do JS => JSX (JavaScript Xml), XML é a linguagem que da base para o HTML.

• TSX no caso de TypeScript

O **ReactDOM** usa um método chamado render uma única vez, tem como propósito renderizar, exibir alguma coisa dentro de um elemento HTML. Ele recebe um document.getElementById(‘root’) e um componente App, esse App é importado, no caso um arquivo App.tsx com uma função dentro App() retornando o JSX

**Componentes** – são pedaços isolados de códigos, quando juntos formam nossa aplicação. Um componente dentro do React é uma **FUNÇÃO** que devolve **HTML.** Tudo é COMPONENTE!

**Propriedades** – informações que podemos passar para um componente se comportar de maneira diferente (como atributos no HTML).  
exemplo no caso de HTML:  
<a href=”” target=”\_blank”></a> o atributo target muda o funcionamento da ancora (tag <a>)

Em alguns casos, evitar dar um EXPORT DEFAULT para evitar problemas com nomenclaturas de funções, importações etc.

**Propriedades podem receber qualquer tipo de informação do JS.**

**Children ->** todo conteúdo de um componente, ele automaticamente se chama children dentro do React

**componente Button,** exemplo:

**<Button> ... conteúdo do componente (children) ...</Button>**

**Com COMPONENTE E PROPRIEDADE, construímos qualquer interface componentizada dentro do React.**

***Variável let*** = let it change

A gente não consegue dar vida ao componente, manipular ele. Por isso temos o conceito mais importante do React – **Estado**.

**Estado** é uma informação mantida por um componente dentro do React. Sempre que tiver uma informação que vai ter o valor alterado pelo uso do usuário (user clica no botão, a informação tem um novo valor) isso é armazenado no Estado. Sempre que uma informação não mantém o mesmo valor durante todo o uso da aplicação, guardamos a informação no Estado.

**ESTADO É SEMPRE CONST**

**Imutabilidade =>** a partir do momento da criação de uma variável dentro do Estado de um Componente, ela não sofre alteração, sempre criamos uma nova informação baseada na informação anterior.

const [counter, setCounter] = useState(0);

  function increment() {

    setCounter(counter + 1);

    console.log(counter);

  }

Não foi alterado o valor do counter, e sim, **SETADO** uma nova informação no counter baseada na informação anterior.

**\*\*\* PASTA SERVICES:** pasta onde coloca arquivo de configuração com um serviço externo.  
 • ***Exemplo:*** Firebase, API de cálculo de cep etc.

**\*\*\* APENAS NO CREATE REACT APP:**

.env.local – arquivo para configurar variáveis do ambiente (variáveis sensíveis), não sobe no git.

**\*\*\***

**Named Export** - isso permite importar mais de uma coisa dentro do arquivo, exemplo:  
{ useEffect, useState, ... }

**\*\*\*** Sempre que quiser colocar um código em JS, o código deve estar entre chaves ***(curly braces).***

**\*\*\* || (or) ‘Default’**

Por conta que o type text ele é opcional (text?), então pode dar erro, por que os outros elementos não foram setados.

**\*\*\***