

Julho, 2022

Discover .

Learn .

Grow .

Repeat .

vnt/school

powered by  venturus

Onde tem conhecimento tem vnt/school



# Recapitulação

# O que foi visto?

- Frontend X Backend
- HTTP
- HTML
- React
  - Preparação do ambiente
  - SPA

# Programação do dia

1 React

2 CSS

3 Next.js - Overview



# Ferramentas



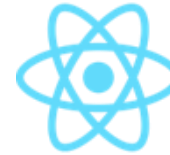
<https://code.visualstudio.com/>



<https://nodejs.org/>



<https://chakra-ui.com/>



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



<https://nextjs.org/>



<https://github.com/>

# React



# Typescript

# Typescript

## O que é o typescript?

Typescript é uma linguagem open source criada pela Microsoft em cima do Javascript para adicionar recursos de tipagem estáticas à linguagem original





# Typescript: Alguns exemplos

Alguns exemplos de tipos básicos e como criar e usar um novo tipo

```
1 let year: number = 2022;
2 var programName: string = 'Venturus 4tech react';
3 const isInProgress: boolean = true;
4
5 const arrayOfThings: any[] = [2022, 'Venturus 4tech', true];
```

```
1 type ProgramType = {
2   name: string;
3   year: number;
4   isInProgress: boolean;
5 };
6
7 const actualProgram: ProgramType = {
8   name: 'Venturus 4tech react',
9   year: 2022,
10  isInProgress: true,
11 };
12
```

# React + Typescript

O React tem total suporte ao typescript e atualmente a maioria dos grandes projetos no Venturus e em outras empresas são iniciados usando typescript.

Algumas razões para usar o typescript em aplicações react:

- Componentes mais fáceis de ler e entender
- Melhor IntelliSense (autocomplete)
- Validação do código em tempo de compilação



# Estrutura

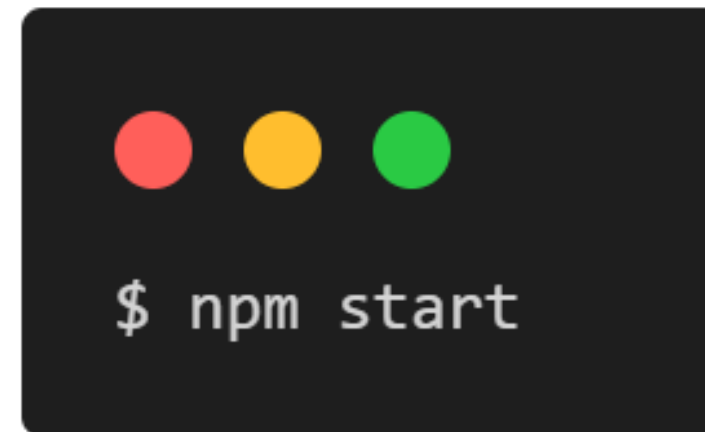
# Estrutura de pastas e arquivos

- node\_modules
- package.json
- package-lock.json
- .gitignore
- tsconfig.json



# Start

# Primeira execução de scripts

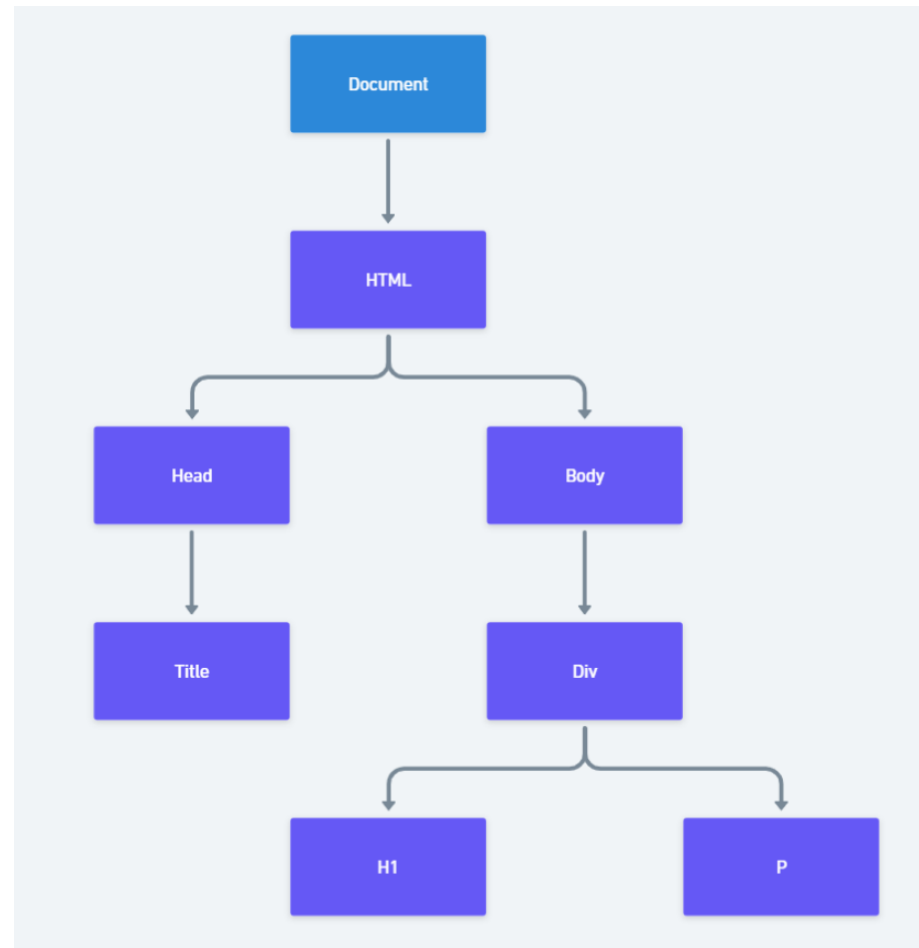




# **DOM – Document Object Model**

# DOM

O DOM (Document Object Model) é usado para representar os dados de um documento na Web.







# Componentes

# Componentes

Componente é uma parte de código que pode ser reaproveitada que retornam um elemento React para ser mostrado em tela.

Podemos ter diversos componentes dentro de um componente. Uma parte do código que se repete muitas vezes pode ser colocada dentro de um componente e apenas o componente será copiado.

Existem diversas bibliotecas de componentes prontos.

# Componentes: Função x Classe

- Um componente pode ser escrito como uma função (método mais simples) ou uma classe.
- Ambos componentes abaixo são equivalentes, porém possuem recursos e peculiaridades diferentes.
- O foco será no uso de funções por serem mais simples.



```
1 class Welcome extends React.Component {  
2   render() {  
3     return <h1>Olá, {this.props.name}</h1>;  
4   }  
5 }
```



```
1 interface WelcomeProps {  
2   name: string  
3 }  
4  
5 function Welcome(props : WelcomeProps) {  
6   return <h1>Olá, {props.name}</h1>  
7 }
```

# Componentes: Bibliotecas de componentes

Existem diversas bibliotecas com diversos componentes prontos que facilitam e agilizam o processo de criação da aplicação



Ant Design



MATERIAL-UI



chakra



# Propriedades

# Propriedades

As propriedades são as entradas de um componente React. Onde esses dados são passados de um componente Pai para um Filho.

Exemplo:

Componente: Box (ChakraUI)

Parâmetros:

- bg (cor do fundo)
- px (padding nas laterais)



```
1 <Box bg={'purple.800'} px={'4'}>
```



# Estado

# Estado

O estado é um dado armazenado relacionado a um componente. Por exemplo, a informação se um botão está habilitado ou não.

Para trabalhar com estado, é necessário usar o Hook nativo do React. Esse Hook é chamado de `useState`.

```
1 import { useState } from 'react';
2
3 function Counter() {
4   const [count, setCount] = useState(0);
5
6   return (
7     <button onClick={() => setCount((count) => count + 1)}>
8       count is {count}
9     </button>
10  )
11 }
12
13 export default Counter;
```





# Navegação

# React Router DOM

O React Router é uma biblioteca de navegação. Podendo ser executada onde quer que o React seja executado.

```
1 import {
2   BrowserRouter,
3   Routes,
4   Route,
5 } from "react-router-dom";
6 // import your route components too
7
8 export default function Routes() {
9   return (
10     <BrowserRouter>
11       <Routes>
12         <Route path="/" element={<App />} />
13         <Route index element={<Home />} />
14         <Route path="feed" element={<Feed />} />
15         <Route path=":postId" element={<Post />} />
16         <Route path="new" element={<AddPost />} />
17       </Route>
18     </Route>
19   </Routes>
20 </BrowserRouter>
21 )
22 }
```

# CSS

Cascading Style Sheets

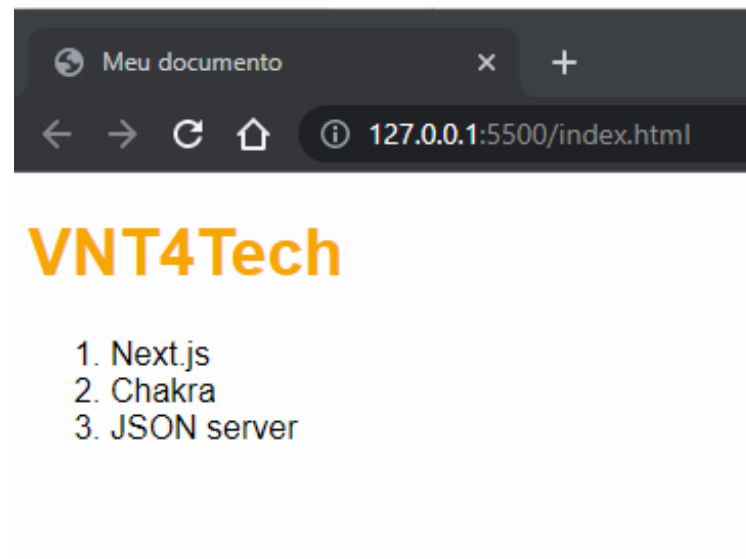


# Introdução ao CSS

# O que é CSS?

CSS (*Cascading Style Sheets*) é usado para descrever o estilo de um documento.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Meu documento</title>
6   <style>
7     body {
8       font-family: Arial, Helvetica, sans-serif;
9       background-color: #FDFDFD;
10    }
11    h1 {
12      color: #F5A700;
13    }
14  </style>
15 </head>
16 <body>
17   <h1>VNT4Tech</h1>
18   <ol>
19     <li>Next.js</li>
20     <li>Chakra</li>
21     <li>JSON server</li>
22   </ol>
23 </body>
24 </html>
```



# Next.js



# Overview

# O que é e pra quê serve?

O Next.js é um framework React que auxilia a criar uma aplicação Web, provendo um ambiente configurado e com funcionalidades, estruturas e otimizações adicionais.

Tipos de renderização possíveis:

- Client-Side Rendering (CSR)
- Server-Side Rendering (SSR)
- Static Site Generation (SSG)