



Semana React e Python - Front End

Checklist do projeto Front-end da Semana React e Python da TreinaWeb

[!\[\]\(c3d993ca47bfe2a953c700506ce31fa0_img.jpg\) Repositório - GitHub](#)

Checklist do Ambiente

- ☐ Instalar Node.js
 -  [Download](#)
 - [!\[\]\(2176a4ba510fa27404d783166e891577_img.jpg\) Guia de Instalação](#)
- ☐ Instalar VS Code
 -  [Download](#)
 - [!\[\]\(428d7e9195be7f8a26074c24b6c91839_img.jpg\) Guia de Instalação](#)

Criação do Projeto

- ☐ Habilitar a execução de scripts no Windows:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

- ☐ Criar o projeto

```
npx create-next-app . --ts
```

- ☐ Iniciar o servidor

```
npm run dev
```

Conhecendo a estrutura do projeto

- ☐ Estrutura do projeto
- ☐ Modificar a página principal com Hello World!
- ☐ O que são e como criar o primeiro componente

Criando componentes estilizados

- ☐ Instalar Material UI

```
npm i @mui/material@5.10.2 @emotion/react@11.10.0  
@emotion/styled@11.10.0
```

- ☐ Remover estilização no arquivo `globals.css` da pasta `styles`
- ☐ Criar pasta `src` na raiz do projeto
- ☐ Criar pasta `components` em `src`
- ☐ Criar o componente `Cabecalho`
- ☐ Inserir a imagem de logo na pasta `public`



- ☐ Instalar a extensão `vscode-styled-components`

Configurando o tema

- ☐ Criar uma pasta chamada `themes` dentro de `src`
- ☐ Criar o arquivo chamado `theme.ts` com o seguinte conteúdo.
- ▼ 🎨 Criar o arquivo chamado `theme.ts` com o seguinte conteúdo (clique aqui para abrir)

```
import { createTheme } from '@mui/material'; const tema =  
createTheme({ palette: { primary: { main: '#F52273', },  
secondary: { main: '#F5F5F5', }, text: { primary: '#F52273',  
secondary: '#6c6767', }, }, components: { MuiButton: {  
styleOverrides: { root: { color: '#FFF', borderRadius: '0',  
backgroundColor: '#F52273', textTransform: 'none', fontWeight:  
'bold', padding: '12px', ':hover': {  
backgroundColor: '#F52273', } }, }, }, MuiTextField: {  
defaultProps: { InputLabelProps: { required: false, }, required:  
true, }, }, }); export default tema;
```

- ☐ Definir o tema em `_app.tsx`

Criar a listagem de professores

- ☐ Criar o componente `Lista` de professores
- ☐ Criar os itens dentro da lista de professores
- ☐ Criar a pasta `@types` dentro de `src`
- ☐ Criar o arquivo `professor.ts` dentro da pasta `@types`
- ☐ Criar a lista de professores para exibir

Listar os professores via API

- ☐ O que são os React Hooks
- ☐ Conhecendo o hook `useState`
- ☐ Criar a pasta `hooks` em `src`
- ☐ Criar a pasta `pages` em `hooks`
- ☐ Criar o arquivo `useIndex.ts` na pasta `pages` e definir a lista de professores
- ☐ Instalar o Axios

```
npm i axios@0.27.2
```

- ☐ Criar a pasta `services` dentro de `src`
- ☐ Criar o arquivo `ApiService.ts` na pasta `services`
- ☐ Iniciar o serviço de back-end

- Windows

```
.\.venv\Scripts\activate python manage.py runserver
```

- Linux/MacOS

```
source .venv/bin/activate python manage.py runserver
```

- ☐ Buscar os dados dinamicamente na API back-end
- ☐ Criar o arquivo `FormatadorService.ts` na pasta `services`

- ☐ Formatar o valor monetário
- ☐ Limitar o tamanho do texto da descrição

Marcar aula com professor

- ☐ Mostrar o diálogo com os campos e botões
- ☐ Passar o valor dos campos para o `useIndex`
- ☐ Definir o professor selecionado
- ☐ Lógica para fechar o diálogo
- ☐ Enviar o pedido a solicitação de marcação de aula
- ☐ Mostrar feedback para o usuário usando a `snackbar`
- ☐ Limpar o formulário

Problemas comuns

- Windows diz que não tenho permissão para executar o comando

- Execute:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

- Os comandos não funcionam

- Se tiver algum anti-vírus, desative-o
- Verifique se digitou corretamente
- Verifique se está dentro da pasta correta
- Tente atualizar o NPM executando:

```
npm install -g npm@latest --force
```

- Limpe o cache com o comando:

```
npm cache clean --force
```

- Feche o terminal e abra novamente

- Ao executar a criação do projeto recebe a mensagem: `O termo 'npm' não é reconhecido como nome de cmdlet...`
 - Certifique que baixou o nome e instalou corretamente, você pode baixar o node nesse link <https://nodejs.org/dist/v16.17.1/node-v16.17.1-x86.msi>
 - Fecha e abre o Visual Studio Code para ter certeza que o terminal está reconhecendo o comando após instalar o node.
- Meu VS Code está diferente
 - Isso não é um problema. A aparência do VS Code é customizável. Isso não irá influenciar na aula.
- Minhas pastas no VS Code são criadas todas juntas, por exemplo: "src/ui/components"
 - O VS Code junta o nome das pastas vazias. Você pode mudar esse comportamento indo em `File > Preferences > Settings`
Pesquise por "Compact folders"
Desmarque o checkbox da configuração `Explorer: Compact Folders`
- O comando de criar projeto inicia, mas não finaliza (ou termina com erro)
 - Pode ser que o processo de instalação tenha travado. Você pode forçar a finalização desse processo seguindo este guia:
<https://www.theopenindia.com/2021/04/found-0-vulnerabilities-how-to-fix-this-create-react-app-problem-.html>
- PORT 3000 is already in use
 - Sua porta 3000 já está sendo usada. No arquivo `package.json` altere o comando `next dev` para `next dev --port 3001` e tente acessar agora por `http://localhost:3001` ou reinicie o computador
- Mensagem de erro no navegador, dizendo "Module not found"
 - Um arquivo não foi encontrado. Verifique se digitou tudo certo como está na aula, se todos os arquivos que você alterou foram salvos e se todas as pastas e arquivos estão na estrutura correta
- Meu arquivo `tsconfig.json` não foi criado automaticamente
 - Veja se você criou o projeto com `--ts`
- Mensagem de Token não reconhecido
 - O PowerShell pode dar erro ao executar um comando com `@`. Coloque aspas em volta do comando, como:
`npm i axios "@mui/material"`

- Minha pasta foi criada mas está vazia
 - O projeto não foi criado corretamente e nem teve sua instalação concluída. Apague a pasta e execute `npx create-next-app . --ts` novamente