

# Pacotes do NPM

Primeiro, vamos falar sobre o NPM. Ele é um registro de software Javascript onde desenvolvedores podem publicar seu trabalho e também usar projetos que outras pessoas desenvolveram a fim de resolver um problema. O maior uso que teremos do NPM é no terminal, onde utilizaremos comandos para manipular os pacotes dos nossos projetos.

## Como usar Pacotes do NPM

Nesse curso vamos focar em como podemos utilizar o NPM para re-utilizar projetos de bibliotecas em nossos projetos, mas vale mencionar que o NPM vai além disso, com planos *Enterprise* que fornecem CDNs para que você possa criar projetos privados e utiliza-los dentro de um escopo organizacional.

### Instalação

O NPM vem instalado junto com o NodeJS, já que toda a sua lógica e funcionamento se da pelo próprio. Então após instalar o NodeJS (coisa que fizemos no começo do curso), podemos rodar `npm --version` para verificar a versão instalada.

Após isso temos alguns comandos para listar:

- `npm init` → Esse comando iniciará na pasta atual o `package.json` - comando que não precisamos usar quando rodamos o `create-react-app` pois ele já configura os pacotes mas necessário se quisermos utilizar **Webpack** ou outro *task runner* para *buildar* nosso projeto.
- `npm install` ou `npm i` → Comando para instalar algum pacote e deve ser rodado sempre acompanhado de argumentos para mudar o comportamento da instalação:
  - `--save` ou `-s` → O pacote será instalado e salvo no `package.json` na propriedade `dependencies`
  - `--save-dev` → O pacote será instalado e salvo no `package.json` na propriedade `devDependencies`
  - `--global` ou `-g` → O pacote será instalado e estará disponível para o uso diretamente no terminal

# React Router

Nessa aula vimos como as versões que instalamos e as documentações que consultamos são casos que podem fazer o seu projeto simplesmente não funcionar. Dito isso, seguem as documentações do `react-router-dom` :

- Versão 5: <https://v5.reactrouter.com/web/guides/quick-start>
- Versão 6: <https://reactrouter.com/en/main>

E de bônus uma documentação especial sobre Versionamento Semântico, dessa forma você conseguirão entender mais facilmente as mudanças de versões nos pacotes que utilizam:

- <https://semver.org/lang/pt-BR/>

# React Slick

Complicado como não ler as documentações podem dar uma dor de cabeça, então depois dessa aula espero ter deixado claro como entender o que você está prestes a colocar no seu projeto é importante.

<https://react-slick.neostack.com/docs/get-started>

# Pacotes de Design System

Falaremos mais sobre Design System, então deixe esse conceito vivo na sua mente. Por enquanto basta saber que existem esses e muitos outros pacotes públicos que podemos utilizar em nossos projetos para facilitar a vida útil do projeto:

- React Bootstrap: <https://react-bootstrap.github.io/>
- Material UI: <https://mui.com/pt/>
- Chakra UI: <https://chakra-ui.com/>

E se utilizar um Design System publico não é uma opção, podemos utilizar bibliotecas CSS para agilizar o desenvolvimento com React. As bibliotecas **atômicas** são melhores utilizadas em React pois ao mesmo tempo em que escrevemos a estrutura HTML dos nossos componentes, podemos incrementar com regras CSS de posicionamento, tamanho, cores e etc e deixar ajustes finos para depois:

- Tailwind CSS: <https://tailwindcss.com/>

- Tachyons: <https://tachyons.io/>

© Curso Online de React do Zero ao Pro  
Desenvolvido por Gustavo Vasconcellos e EBAC Online