

# Pesquisa Tema Css e React

- **O que é Bootstrap?**

Bootstrap é um framework (esta palavra vem de um termo em inglês que significa estrutura, e um framework é um conjunto de bibliotecas que unificam códigos comuns entre vários projetos). O Bootstrap fornece componentes de prateleira gratuitos, que reúne todas as estruturas CSS, HTML e JS (JavaScript) necessárias. Ele foi criado para facilitar a vida dos profissionais de programação. A ferramenta de bootstrap foi criada para ser usada no Twitter, mas sua comodidade é que você não precisa se preocupar com comandos básicos e funções adicionais. Ele foi originalmente criado para desenvolver estruturas de front-end mais rapidamente, mas é open source. O objetivo original da criação era adotar uma única estrutura, reduzindo assim os erros do programador entre diferentes métodos de codificação.

Fontes:

<https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap>

<https://www.homehost.com.br/blog/tutoriais/o-que-e-bootstrap/>

<https://tableless.github.io/iniciantes/manual/js/o-que-framework.html>

- **O que é Flexbox?**

Flexbox é uma solução de layout que pode alinhar e organizar elementos na interface. Ele cria contêineres e itens flexíveis que podem ser adaptados a vários tipos de telas sem a necessidade de escrever CSS específico para cada resolução. Ele se encaixa na web e pode ser expandido para preencher o espaço disponível ou selecionado para evitar travar no site. Existem dois eixos no Flexbox, a saber: eixo principal e eixo horizontal. O eixo principal é orientado pela direção flexível que define a direção e seu valor:

1. Row (linha)
2. Row-reverse (linha reversa)
3. Column (coluna)
4. Column-reverse (coluna reversa)

Se definirmos no eixo principal que ele deve seguir a direção da linha, o eixo horizontal é paralelo ao eixo principal, de forma que o eixo horizontal ocupará a posição da linha. Se o valor selecionado for linha ou reverso de linha, ele se moverá na linha, se estiver em coluna ou reverso de coluna, ele se moverá na coluna. Também temos linhas inicial e final para guiar a borda inicial. Como existem diferenças entre o Oriente e o Ocidente, não indicamos esse problema,

usamos o início e o fim. Essa abordagem se tornou natural depois de lidar com o método de layout por um período de tempo. Outro termo que usamos no Flexbox é Flex container. Para usar essa estrutura, o valor é definido pela propriedade display (ela define o elemento flex container e o torna um flex child-item).

Fontes:

[https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Basic\\_Concepts\\_of\\_Flexbox](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)

<https://origamid.com/projetos/flexbox-guia-completo/>

[https://www.youtube.com/watch?v=ssc-cJZJrII&ab\\_channel=DevMedia](https://www.youtube.com/watch?v=ssc-cJZJrII&ab_channel=DevMedia)

<https://www.zup.com.br/blog/flexbox-beagle>

<https://blog.betrybe.com/css/css-flexbox/>

- **O que é Grid CSS?**

Grid apresenta um sistema de grade bidimensional (literalmente significa grade), que divide a página inteira em linhas e colunas e coloca os elementos onde você quiser nessa partição. A diferença entre grade e Flexbox é que a grade possui duas dimensões (colunas e linhas), enquanto o Flexbox possui apenas uma dimensão (colunas ou linhas), mas uma completa a outra. O que é a grade final nada mais é do que uma coleção de interseções de linhas e colunas, e grades de diferentes dimensões podem ser criadas. O tamanho pode ser fixo (quando definimos o tamanho em pixels). Ou flexível (quando definido como uma porcentagem). No entanto, existem alguns tipos de grades, são eles: Use display-grid ou display-inline-grid como o contêiner de grade para os elementos marcados. E definir a faixa de grade das linhas e colunas na grade nas propriedades.

Fontes: [https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS\\_Grid\\_Layout/Basic\\_Concepts\\_of\\_Grid\\_Layout](https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)

[https://www.youtube.com/watch?v=HN1UjzRSdBk&ab\\_channel=Rocketseat](https://www.youtube.com/watch?v=HN1UjzRSdBk&ab_channel=Rocketseat)

<https://www.treinaweb.com.br/blog/css-grid-um-guia-interativo-parte-1-containers>

- **O que é React?**

O React surgiu em 2011 com o time do Facebook, pois eles tinham dificuldades em sincronizar diferentes códigos que acontecia na tela ao mesmo tempo, mas o React nada mais é que uma biblioteca JavaScript (Js) de código aberto com o foco em criação de UI – Interface de usuários. Ele é excelente para o usuário adicional comandos usando um método novo para carregar a página.

- **E as vantagens de usá-lo para o desenvolvimento de Single Page Application?**

Antes de explicar as vantagens de usar o React, precisando entender o que é Single Page Application ou SPA, basicamente é uma aplicação que codifica no serve-side (lado do servidor) e mais no cliente-side (lado do cliente, também conhecido como front-end).

Algumas das Vantagens são:

1. Fácil de Usar:

React é uma biblioteca de código aberto, de software livre que se concentra em uma única coisa: tornar a experiência do usuário mais eficiente.

2. Performance:

Permite que a página tenha uma desenvoltura melhor, pois carrega o sistema mais rápido.

3. Suporte Componentes em Java:

Permite que reutilize componentes de outras linguagens e que use as mesmas funções.

4. Componente Fácil de escrever:

É fácil de escrever pois usa o JSX que é uma extensão de sintaxe do Js. Onde permite a combinação de HTML e Js.

Fontes:

<https://kenzie.com.br/blog/react/>

[https://www.youtube.com/watch?v=6luQUgeDPg0&ab\\_channel=AluraCursosOnline](https://www.youtube.com/watch?v=6luQUgeDPg0&ab_channel=AluraCursosOnline)

<https://blog.locaweb.com.br/desenvolvimento/o-que-e-single-page-application/>