

React





TechGuide - Alura, FIAP e PM3

Nível 1





HTML - Fundamentos:

- HTML é uma linguagem de marcação que define a estrutura do seu conteúdo. HTML consiste em uma série de elementos que você usa para mostrar algo de uma determinada maneira ou agir de uma certo modo. As tags podem criar um hiperlink de uma palavra ou imagem para outro lugar, podem colocar palavras em itálico, podem aumentar ou diminuir a fonte e assim por diante.
- Aprender quais tags são necessárias para um HTML básico
- Criar um parágrafo de texto
- Exibir uma imagem
- Conhecer a diferença entre 'h1', 'h2', 'h3', etc
- Criar um texto com hyperlink
- Criar um formulário com campos relevantes
- Criar uma lista de itens ordenada ou não ordenada
- Criar uma lista de itens dentro de uma lista suspensa (dropdown list)
- Conectar com um arquivo de CSS
- Criar uma tabela
- Adicionar IDs e classes

Conteúdos

-  MDN Web Docs: HTML básico (https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics)
-  Rafaella Ballerini: 5 MINUTOS DE HTML PARA INICIAR EM PROGRAMAÇÃO! (<https://www.youtube.com/watch?v=3oSlqlqzN3M>)
-  Rafaella Ballerini: LANDING PAGE COM HTML e CSS! (<https://www.youtube.com/watch?v=llF6vD-RlJE>)
-  Rafaella Ballerini: FORMULÁRIOS COM HTML e CSS! (<https://www.youtube.com/watch?v=wwqOJ2o84S4>)

Conteúdos Alura:**

-  HTML: o que é, a importância para a Web, como aprender e um Guia para iniciantes (<https://www.alura.com.br/artigos/html>)
-  Alura+: Fazendo deploy de um projeto na Vercel (<https://cursos.alura.com.br/extra/alura-mais/fazendo-deploy-de-um-projeto-na-vercel-c9239>)
-  HTML, CSS e Javascript, quais as diferenças? (<https://www.alura.com.br/artigos/html-css-e-js-definicoes>)
-  Formação HTML e CSS (<https://www.alura.com.br/formacao-html-e-css>)

- **Curso** Apostila: Desenvolvimento Web com HTML, CSS e JavaScript (<https://www.alura.com.br/apostila-html-css-javascript>)
- **Desafio** 7 Days of Code: HTML e CSS (<https://7daysofcode.io/matricula/html-css>)

CSS - Fundamentos:

- Cascading Style Sheets (CSS) é uma linguagem usada para descrever a apresentação de um documento escrito em uma linguagem de marcação como HTML ou XML. CSS pode ser usado para estilos de texto de documentos muito básicos — por exemplo, para alterar a cor e o tamanho de títulos e links. Ele pode ser usado para criar um layout — por exemplo, transformar uma única coluna de texto em um layout com uma área de conteúdo principal e uma barra lateral para informações relacionadas. Pode até ser usado para efeitos como animações.
- Aprender a estrutura visual de uma página, com 'margin' e 'padding'
- Estabelecer o tamanho com 'width' e 'height'
- Aprender sobre a posição de um elemento ('static', 'relative' ou 'absolute')
- Aprender sobre o 'display' de exibição de um elemento ('block', 'inline', 'inline-block')
- Aprender a posicionar imagens em relação ao texto
- Aprender sobre alinhamento
- Aprender sobre estilo de fontes
- Aprender as diferenças e vantagens de usar as diferentes unidades de medida em CSS (% , relativas, etc)
- Conectar com os elementos (IDs, classes) de um arquivo HTML
- Alterar características de um elemento quando o mouse passar por cima dele ('hover')
- Aprender box-sizing
- Aprender Flexbox
- Aprender Grid

Conteúdos

- **Site** MDN Web Docs: Iniciando com CSS (https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps/Getting_started)
- **Site** MDN Web Docs: Seletores (https://developer.mozilla.org/pt-BR/docs/Learn/CSS/Building_blocks/Selectors)
- **Site** MDN Web Docs: Cascata e herança (https://developer.mozilla.org/pt-BR/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)
- **Site** MDN Web Docs: Propriedade 'display' (<https://developer.mozilla.org/pt-BR/docs/Web/CSS/display>)
- **Site** MDN Web Docs: Conceitos básicos de Flexbox (https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox)
- **Site** MDN Web Docs: Grid Layout (https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout)
- **Artigo** Entendendo CSS: Display Inline, Block e Inline-Block (<https://dev.to/sucodelarangela/entendendo-css-display-inline-block-e-inline-block-lic>)
- **Artigo** Entendendo CSS: Pseudo-Classes e Pseudo-Elementos (<https://dev.to/sucodelarangela/entendendo-css-pseudo-classes-e-pseudo-elementos-b83>)
- **Artigo** Como aplicar opacidade em background-image sem afetar textos (<https://dev.to/sucodelarangela/como-aplicar-opacidade-em-background-image-sem-afetar-textos-31fj>)

- **YouTube** Rafaella Ballerini: O QUE É CSS? (<https://www.youtube.com/watch?v=LWU2OR19ZG4>)
- **YouTube** Rafaella Ballerini: FLEXBOX CSS! Como posicionar elementos na página web - parte 1 (https://www.youtube.com/watch?v=KbjLtEgmZ_E)
- **YouTube** Rafaella Ballerini: FLEXBOX CSS! Como posicionar elementos na página web - parte 2 (https://www.youtube.com/watch?v=hjz6ezV9_uc)
- **YouTube** Matheus Castiglioni: Arquitetura CSS - BEM (<https://www.youtube.com/watch?v=yKPXW9aSxQI>)
- **YouTube** Matheus Castiglioni: CSS Grid Layout (<https://www.youtube.com/watch?v=HBIBNAtFcdw>)
- **YouTube** Mario Souto - Dev Soutinho: Como centralizar no CSS (<https://www.youtube.com/watch?v=Cu-HP-gvggg>)

Conteúdos Alura:**

- **Podcast** Hipster 09 - CSS: Cansei de Ser Simples (<https://www.hipsters.tech/css-cansei-de-ser-simples-hipsters-09/>)
- **Artigo** CSS: o que é, como usar no HTML e um Guia para iniciar (<https://www.alura.com.br/artigos/css>)
- **Artigo** HTML, CSS e Javascript, quais as diferenças? (<https://www.alura.com.br/artigos/html-css-e-js-definicoes>)
- **Artigo** Guia de unidades no CSS (<https://www.alura.com.br/artigos/guia-de-unidades-no-css>)
- **Artigo** CSS: Guia do Flexbox (<https://www.alura.com.br/artigos/css-guia-do-flexbox>)
- **Artigo** Como fazer Grids e a Responsividade na Web (<https://www.alura.com.br/artigos/como-fazer-grids-e-a-responsividade-na-web>)
- **Artigo** Como criar animações incríveis com CSS (<https://www.alura.com.br/artigos/animacoes-com-css>)
- **Artigo** Houdini CSS: um jeito mágico de criar estilos personalizados (<https://www.alura.com.br/artigos/houdini-css>)
- **YouTube** Alura: CSS FlexBox: Dicas para começar (<https://www.youtube.com/watch?v=326-B1avuYo>)
- **YouTube** Alura: Box Model e Sizing no CSS (<https://www.youtube.com/watch?v=qcNUxyOWXlw>)
- **YouTube** Alura: Curso de HTML5 e CSS3 (<https://www.youtube.com/watch?v=78AyiuxYceg>)
- **Curso** Apostila: Desenvolvimento Web com HTML, CSS e JavaScript (<https://www.alura.com.br/apostila-html-css-javascript>)
- **Curso** Formação HTML e CSS (<https://www.alura.com.br/formacao-html-e-css>)
- **Curso** Formação CSS: aprofunde em estilos (<https://cursos.alura.com.br/formacao-css-estilos>)
- **Curso** HTML e CSS: responsividade com mobile-first (<https://www.alura.com.br/curso-online-html-css-responsividade-mobile-first>)
- **Curso** CSS: cursos para transformar designs com Grid, Flexbox e Sass (<https://cursos.alura.com.br/formacao-css-cursos-transformar-designs-grid-flexbox-sass>)
- **Desafio** 7 Days of Code: HTML e CSS (<https://7daysofcode.io/matricula/html-css>)
- **Desafio** 7 Days of Code: Responsividade (<https://7daysofcode.io/matricula/responsividade>)

JavaScript - Fundamentos:

- JavaScript é a linguagem de programação mais popular do mundo e é uma das principais tecnologias da World Wide Web, juntamente com HTML e CSS. Ela possui tipagem dinâmica,

orientação a objetos baseada em protótipos e funções de primeira classe. Ela é multi-paradigma e suporta estilos de programação orientados a eventos, funcionais e imperativos.

- Conhecer os tipos primitivos
- Declarar variáveis, considerando a diferença entre 'var', 'let' e 'const'
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação ('=', '==', '===')
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular arrays e listas
- Aprender o conceito de Orientação a Objetos
- Fazer um CRUD
- Obter dados de uma API
- Fazer chamadas assíncronas usando 'Async/Await', 'Promise', etc

Conteúdos

- **Site** MDN Web Docs: Um primeiro mergulho no JavaScript (https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps)
- **Site** MDN Web Docs: Trabalhando com texto — strings em JavaScript (https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Strings)
- **Site** MDN Web Docs: if...else (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else>)
- **Site** MDN Web Docs: while (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/while>)
- **Site** MDN Web Docs: for (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for>)
- **Site** MDN Web Docs: Arrays (https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/Arrays)
- **Site** MDN Web Docs: Template strings (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Template_literals)
- **Artigo** Definindo Funções em Javascript (<https://blog.matheuscastiglioni.com.br/definindo-funcoes-em-javascript/>)
- **YouTube** Mario Souto - Dev Soutinho: Como manipular arrays e objetos em JavaScript (<https://www.youtube.com/watch?v=yS7AcF-xRUg>)
- **YouTube** Marco Bruno: O que é Json e como criar um objeto (<https://www.youtube.com/watch?v=oCY5YEEjIwE>)
- **YouTube** Mario Souto - Dev Soutinho: Como usar Async/Await? Promises no JavaScript? (<https://www.youtube.com/watch?v=q28IfkBd9F4>)
- **YouTube** Mario Souto - Dev Soutinho: Como pegar dados de uma API? Como fazer AJAX? | Pegando dados de serviços via JavaScript (<https://www.youtube.com/watch?v=85vJXFpXLQw>)

Conteúdos Alura:**

- **Artigo** JavaScript: o que é, como aprender e um Guia da linguagem mais popular do mundo (<https://www.alura.com.br/artigos/javascript/>)
- **Podcast** Hipster 38 - O Reino encantado do JavaScript (<https://www.hipsters.tech/o-reino-encantado-do-javascript-hipsters-38/>)

- **Podcast** Hipster 169 - JavaScript: manual de sobrevivência 2020 (<https://www.hipsters.tech/javascript-manual-de-sobrevivencia-2020-hipsters-169/>)
- **Podcast** Hipster 236 - Evolução do JavaScript (<https://www.hipsters.tech/evolucao-do-javascript-hipsters-ponto-tech-236/>)
- **Artigo** Strings com JavaScript: o que são e como manipulá-las (<https://www.alura.com.br/artigos/strings-com-javascript-o-que-sao-e-como-manipular>)
- **Artigo** Como utilizar operadores de comparação em Javascript (<https://www.alura.com.br/artigos/operadores-matematicos-em-javascript>)
- **Artigo** Para que serve um Array? (<https://www.alura.com.br/artigos/javascript-para-que-serve-array>)
- **Artigo** Quando usar forEach e map? (<https://www.alura.com.br/artigos/javascript-quando-devo-usar-foreach-e-map>)
- **Artigo** Manipulação de array com map, filter e reduce (<https://www.alura.com.br/artigos/manipulacao-de-array-com-map-filter-e-reduce>)
- **Curso** Formação Desenvolva aplicações Web com JavaScript (<https://cursos.alura.com.br/formacao-javascript-front-end>)
- **YouTube** Alura: O que é JavaScript? (<https://www.youtube.com/watch?v=NaVSbnnV75Q>)
- **YouTube** Alura: Destructuring em JavaScript (<https://www.youtube.com/watch?v=f8a-qwKC5yk>)
- **YouTube** Alura: Classes x funções no Javascript (<https://www.youtube.com/watch?v=i0hhj-k9L6s>)
- **Desafio** 7 Days of Code: JavaScript e DOM (<https://7daysofcode.io/matricula/javascript-e-dom>)
- **Desafio** 7 Days of Code: JavaScript e DOM com API (<https://7daysofcode.io/matricula/javascript-e-dom-api>)
- **Desafio** 7 Days of Code: JavaScript e DOM - Wordle (<https://7daysofcode.io/matricula/javascript-e-dom-wordle>)

DOM - Fundamentos:

- O Document Object Model (DOM) é uma interface de programação para documentos web. Ele representa a página para que os programas possam alterar a estrutura, o estilo e o conteúdo do documento. O DOM representa o documento como nós e objetos; dessa forma, linguagens de programação podem interagir com a página.
- Entender como funciona a árvore do DOM
- Acessar e manipular elementos do HTML e CSS
- Acessar os pais e filhos de um elemento
- Inserir um novo elemento na árvore
- Remover um elemento da árvore
- Esperar por um evento em certo elemento da página usando 'addEventListener()'

Conteúdos

- **Site** MDN Web Docs: Introdução ao DOM (https://developer.mozilla.org/pt-BR/docs/Web/API/Document_Object_Model/Introduction)
- **Site** MDN Web Docs: querySelector() (<https://developer.mozilla.org/pt-BR/docs/Web/API/Document/querySelector>)
- **Site** MDN Web Docs: addEventListener() (<https://developer.mozilla.org/pt-BR/docs/Web/API/EventTarget/addEventListener>)

- **YouTube** Felipe Aguiar - Laboratório de javascript: Javascript DOM 01 - DOM - A base da manipulação e criação de páginas dinâmicas (<https://www.youtube.com/watch?v=G31EuXNMDPk>)

Conteúdos Alura:**

- **Curso** Curso JavaScript: manipulando o DOM (<https://cursos.alura.com.br/course/javascript-manipulando-elementos-dom>)
- **Artigo** Guia para manipular eventos no DOM com JavaScript: técnicas e exemplos práticos (<https://www.alura.com.br/artigos/eventos-do-dom>)
- **Desafio** 7 Days of Code: JavaScript e DOM (<https://7daysofcode.io/matricula/javascript-e-dom>)
- **Desafio** 7 Days of Code: JavaScript e DOM com API (<https://7daysofcode.io/matricula/javascript-e-dom-api>)
- **Desafio** 7 Days of Code: JavaScript e DOM - Wordle (<https://7daysofcode.io/matricula/javascript-e-dom-wordle>)

Conceitos SPA:

- Um aplicativo de página única (SPA - single-page application) é uma aplicação web ou site que interage com o usuário reescrevendo dinamicamente a página web atual com novos dados do servidor web, em vez do método padrão de um navegador que carrega novas páginas inteiras. O objetivo são transições mais rápidas, que tornam o site mais parecido com uma aplicação nativa.
- Entender o que é uma SPA
- Estabelecer rotas para outras páginas
- Conhecer frameworks SPA
- Comunicação com APIs

Conteúdos

- **Site** Wikipédia: Aplicativo de página única (SPA) (https://pt.wikipedia.org/wiki/Aplicativo_de_p%C3%A1gina_%C3%BAnica)
- **Site** MDN Web Docs: SPA (Single-page application) (inglês) (<https://developer.mozilla.org/en-US/docs/Glossary/SPA>)
- **Artigo** What I wish I had known about SPAs (inglês) (<https://stackoverflow.blog/2021/12/28/what-i-wish-i-had-known-about-single-page-applications/>)

Conteúdos Alura:**

- **Podcast** Hipsters.tech: Single Page Applications (<https://www.hipsters.tech/single-page-applications-hipsters-16/>)
- **YouTube** Alura: O que é uma Single-Page Application? (SPA) (<https://www.youtube.com/watch?v=opxYpCKzILk>)
- **YouTube** Alura: Como configurar o React Router para sua SPA (<https://www.youtube.com/watch?v=4Tb8dp5GYql>)
- **Curso** Imersão React: AluraCord - Aula 02 (<https://cursos.alura.com.br/imersoes/aulas/aula-02-state-novas-paginas-e-navegacao-spa-vs-a-tradicional-c69>)
- **Site** Primeiras aulas do curso React Router: navegação em uma SPA (<https://www.alura.com.br/conteudo/react-router-navegacao-spa>)
- **Curso** Curso React: desenvolvendo em React Router com JavaScript (<https://cursos.alura.com.br/course/React-desenvolvendo-react-router-javascript>)

- **Curso** Curso React: conhecendo a biblioteca React Router (<https://www.alura.com.br/curso-online-react-biblioteca-react-router>)

React - Componentes:

- O React permite definir componentes como classes ou funções. Componentes definidos como classes fornecem mais recursos. Eles aceitam entradas arbitrárias (chamadas "props") e retornam elementos React descrevendo o que deve aparecer na tela.
- Para que servem e como funcionam componentes
- Conhecer a biblioteca Styled Components

Conteúdos

- **Artigo** Criando Styled Components Com React (<https://blog.matheuscastiglioni.com.br/criando-styled-components-com-react/>)
- **Site** React: Documentação - Componentes e Props (<https://pt-br.reactjs.org/docs/components-and-props.html#gatsby-focus-wrapper>)
- **Site** MDN Web Docs: Começando com React - Explorando seu primeiro componente React (https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#explorando_seu_primeiro_componente_react_%E2%80%99)
- **YouTube** Mario Souto - Dev Soutinho: CSS-in-JS na prática! Criando componentes com React e CSS no NextJS (<https://www.youtube.com/watch?v=aDKxJfJiM28>)

Conteúdos Alura:**

- **Artigo** React: o que é, como funciona e um Guia dessa popular ferramenta JS (<https://www.alura.com.br/artigos/react-js>)
- **Artigo** Utilizando o Vite para criar uma aplicação React com TypeScript (<https://www.alura.com.br/artigos/vite-criar-aplicacao-react-typescript>)
- **Artigo** Componentes React com TypeScript (<https://www.alura.com.br/artigos/componentes-react-typescript>)
- **Artigo** React: estratégias para componentes escaláveis (<https://www.alura.com.br/artigos/componentes-do-react>)
- **Site** Ciclo de vida de um componente React (<https://cursos.alura.com.br/extra/alura-mais/ciclo-de-vida-de-um-componente-react-c1333>)
- **Curso** Curso React: desenvolvendo com JavaScript (<https://cursos.alura.com.br/course/react-desenvolvendo-javascript>)
- **Curso** Curso React: como os componentes funcionam (<https://www.alura.com.br/curso-online-react-componentes-funcionam>)
- **Curso** Curso React: React: estilize componentes com Styled Components e manipule arquivos estáticos (<https://www.alura.com.br/curso-online-react-estilize-componentes-styled-components-manipule-arquivos-estaticos>)
- **Curso** Formação Explore React com JavaScript (<https://cursos.alura.com.br/formacao-react-javascript>)
- **Curso** Imersão React: AluraFlix (<https://cursos.alura.com.br/imersoes/aulas/aula-1-react-components-e-aluraflix-c18>)
- **Curso** Imersão React: AluraQuiz (<https://cursos.alura.com.br/imersoes/aulas/aula-1-react-styled-components-e-nextjs-c33>)
- **Curso** Imersão React: AluraKut (<https://cursos.alura.com.br/imersoes/aulas/aula-1-components-com-react-styled-components-e-nextjs-c53>)

- **Curso** Imersão React: AluraCord (<https://cursos.alura.com.br/imersoes/aulas/aula-01-criando-o-nosso-projeto-aluracord-c68>)

React - Props:

- Props são um objeto que é injetado em componentes e fornece alguns dados que podem ser compartilhados entre outros componentes em um fluxo de dados unidirecional, de um elemento pai para um elemento filho. Props são de somente leitura.
- Como passar props
- Como manipular props

Conteúdos

- **Site** React: Documentação - Componentes e Props (<https://react.dev/learn/passing-props-to-a-component>)
- **Site** React: Documentação - Render Props (<https://pt-br.reactjs.org/docs/render-props.html#gatsby-focus-wrapper>)
- **Site** MDN React Props (https://www.w3schools.com/react/react_props.asp)
- **Site** MDN Web Docs: Começando com React - Variáveis e props (https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#vari%C3%A1veis_e_props)

Conteúdos Alura:**

- **Artigo** React: o que é, como funciona e um Guia dessa popular ferramenta JS (<https://www.alura.com.br/artigos/react-js>)
- **Curso** Curso React: desenvolvendo com JavaScript (<https://cursos.alura.com.br/course/react-desenvolvendo-javascript>)
- **Curso** Formação Explore React com JavaScript (<https://cursos.alura.com.br/formacao-react-javascript>)
- **Curso** Curso React: escrevendo com Typescript (<https://www.alura.com.br/curso-online-react-modernizando-escrever-typescript>)
- **Curso** Imersão React: ALURATUBE (<https://cursos.alura.com.br/imersoes/aulas/aula-02-entendendo-state-e-fazendo-a-busca-do-aluratube-c94>)

React Hooks - State:

- O hook React **useState** nos permite rastrear o estado em um componente de função. O hook **useState** pode ser usado para acompanhar strings, números, booleanos, arrays, objetos, etc.
- Controlar o estado de componentes
- Manipular variáveis
- Atualizar o valor de elementos

Conteúdos

- **Site** React: Documentação - Introdução aos Hooks (<https://pt-br.react.dev/reference/react>)
- **Artigo** Qual o tipo de um React Hook? (<https://www.alura.com.br/artigos/qual-tipo-react-hook>)
- **Site** React: Documentação - Usando o State do Hook (<https://react.dev/learn/state-a-components-memory>)
- **Site** MDN Web Docs: React interactivity: Events and state (inglês) (https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_interactivity_events_state#state_and_the_ustate_hook)

- **Artigo** Primeiros passos com React Hooks — useState (<https://medium.com/@devjpnobrega/primeiros-passos-com-react-hooks-usestate-aa4eeec5f4e>)

Conteúdos Alura:**

- **Artigo** React Hooks: o que é e como eles funcionam (<https://www.alura.com.br/artigos/react-hooks>)
- **Artigo** React: o que é, como funciona e um Guia dessa popular ferramenta JS (<https://www.alura.com.br/artigos/react-js>)
- **YouTube** Alura: useState - aprenda o React Hook neste tutorial (<https://www.youtube.com/watch?v=KdQa4Rd6K1A>)
- **Curso** Imersão React: AluraCord - Aula 02 (<https://cursos.alura.com.br/imersoes/aulas/aula-02-state-novas-paginas-e-navegacao-spa-vs-a-tradicional-c69>)
- **Curso** Curso React: desenvolvendo com JavaScript (<https://cursos.alura.com.br/course/react-desenvolvendo-javascript>)
- **Curso** Formação Explore React com JavaScript (<https://cursos.alura.com.br/formacao-react-javascript>)
- **Curso** Curso React: escrevendo com Typescript (<https://www.alura.com.br/curso-online-react-modernizando-escrever-typescript>)

Criando uma aplicação React:

- Estruturar um novo projeto React
- Criar uma aplicação funcional do zero

Conteúdos

- **Site** React: Documentação - Crie um novo React App (<https://pt-br.react.dev/learn/start-a-new-react-project>)
- **Site** React: Documentação - Usando TypeScript com Create React App (<https://pt-br.reactjs.org/docs/static-type-checking.html#using-typescript-with-create-react-app>)
- **Site** MDN Web Docs: Começando com React - Configurando seu primeiro app React (https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started#configurando_seu_primeiro_app_react)
- **YouTube** Mario Souto - Dev Soutinho: Como começar a estruturar um projeto React? com NextJS, Styled Components (<https://www.youtube.com/watch?v=mJK5oGixSYo>)
- **YouTube** Mario Souto - Dev Soutinho: Do Zero ao React | Como fazer o React e aprender a ideia do Virtual DOM na prática (<https://www.youtube.com/watch?v=5MzOCxSWrrc>)

Conteúdos Alura:**

- **Artigo** React: o que é, como funciona e um Guia dessa popular ferramenta JS (<https://www.alura.com.br/artigos/react-js>)
- **Artigo** Utilizando o Vite para criar uma aplicação React com TypeScript (<https://www.alura.com.br/artigos/vite-criar-aplicacao-react-typescript>)
- **Podcast** Hipsters.tech: Guia do iniciante em React (<https://www.hipsters.tech/guia-do-iniciante-em-react-hipsters-209/>)
- **Podcast** Hipsters.tech: Do Front End ao React (<https://www.hipsters.tech/do-front-end-ao-react-hipsters-ponto-tech-258/>)
- **Curso** Curso React: desenvolvendo com JavaScript (<https://cursos.alura.com.br/course/react-desenvolvendo-javascript>)

- **Curso** Formação Explore React com JavaScript (<https://cursos.alura.com.br/formacao-react-javascript>)
- **Curso** Curso React: escrevendo com Typescript (<https://www.alura.com.br/curso-online-react-modernizando-escrever-typescript>)
- **Curso** Formação Gerencie aplicações React com Typescript (<https://www.alura.com.br/formacao-react-ts>)
- **Curso** Imersão React: AluraFlix (<https://cursos.alura.com.br/imersoes/aulas/aula-1-react-components-e-aluraflix-c18>)
- **Curso** Imersão React: AluraQuiz (<https://cursos.alura.com.br/imersoes/aulas/aula-1-react-styled-components-e-next-js-c33>)
- **Curso** Imersão React: AluraKut (<https://cursos.alura.com.br/imersoes/aulas/aula-1-components-com-react-styled-components-e-nextjs-c53>)
- **Curso** Imersão React: AluraCord (<https://cursos.alura.com.br/imersoes/aulas/aula-01-criando-o-nosso-projeto-aluracord-c68>)
- **Curso** Imersão React: AluraTube (<https://cursos.alura.com.br/imersoes/aulas/aula-01-a-web-os-componentes-e-o-deploy-c93>)
- **Desafio** 7 Days of Code: React (<https://7daysofcode.io/matricula/react>)

React Hooks - Effect:

- UseEffect() é um hook React que permite que você manipule efeitos colaterais em seus componentes funcionais do React.
- Executar um componente somente após a renderização
- Acessar as props de um elemento
- Fazer chamadas a APIs

Conteúdos

- **Site** React: Documentação - Introdução aos Hooks (<https://pt-br.react.dev/reference/react>)
- **Site** React: Documentação - Usando Effect Hook (Hook de Efeito) (<https://react.dev/reference/react/useEffect>)
- **Artigo** Entenda como funciona o useEffect (<https://medium.com/reactbrasil/entenda-como-funciona-o-useeffect-e743b86f104>)
- **Artigo** Um guia completo para useEffect (<https://overreacted.io/pt-br/a-complete-guide-to-useeffect/>)
- **Artigo** Entenda de vez o hook useEffect e Transforme Seu Código React (<https://medium.com/@marcosviniciosneves/entenda-de-vez-o-hook-useeffect-e-transforme-seu-c%C3%B3digo-react-39245296ef7e>)

Conteúdos Alura:**

- **Artigo** React Hooks: o que é e como eles funcionam (<https://www.alura.com.br/artigos/react-hooks>)
- **Artigo** React: o que é, como funciona e um Guia dessa popular ferramenta JS (<https://www.alura.com.br/artigos/react-js>)
- **Curso** Imersão React: AluraQuiz - Aula 03 (<https://cursos.alura.com.br/imersoes/aulas/aula-3-inputs-e-pagina-de-quiz-c35>)
- **Curso** Imersão React: AluraKut - Aula 02 (<https://cursos.alura.com.br/imersoes/aulas/aula-2-react-state-e-o-primeiro-formulario-c54>)

- **Curso** Curso React: desenvolvendo com JavaScript (<https://cursos.alura.com.br/course/react-desenvolvendo-javascript>)
- **Curso** Formação Explore React com JavaScript (<https://cursos.alura.com.br/formacao-react-javascript>)
- **Curso** Curso React: escrevendo com Typescript (<https://www.alura.com.br/curso-online-react-modernizando-escrever-typescript>)

Nível 2

React Hooks - Memo:

- O hook React **useMemo** retorna um valor memoizado. **useMemo** apenas recalculará o valor memoizado quando uma das dependências for alterada.
- Controlar o estado de variáveis externas
- Evitar cálculos caros em cada renderização

Conteúdos

- **Site** React: Documentação - useMemo (<https://pt-br.react.dev/reference/react/useMemo>)
- **Artigo** React.useMemo na prática (<https://medium.com/reactbrasil/react-usememo-na-pr%C3%A1tica-692110771c01>)
- **Artigo** Diminuindo Processamento com useMemo em React (<https://blog.matheuscastiglioni.com.br/diminuindo-processamento-com-use-memo-em-react/>)
- **Site** useHooks.com: useMemo (inglês) (<https://usehooks.com/useMemo/>)
- **YouTube** React Floripa: Os mitos sobre useMemo e useCallback (https://www.youtube.com/watch?v=ZntFz_DDakQ)

Conteúdos Alura:**

- **YouTube** Alura: useMemo - aplicação React JS com mais performance (<https://www.youtube.com/watch?v=CkGh3NM6NuQ>)
- **YouTube** Alura: Hook useCallback x useMemo - performance no React (<https://www.youtube.com/watch?v=VVmGrEzoByM>)
- **Curso** Curso React: otimizando a performance (<https://cursos.alura.com.br/course/react-otimizando-performance>)

React Hooks - Callback:

- O hook React **useCallback** retorna uma função de callback memoizada.
- Retornar uma versão memoizada do callback caso as entradas tenham sido alteradas

Conteúdos

- **Site** React: Documentação - useCallback (<https://pt-br.react.dev/reference/react/useCallback>)
- **Artigo** Primeiros passos com React Hooks — useMemo e useCallback (<https://medium.com/@devjpnobrega/primeiros-passos-com-react-hooks-usememo-e-usecallback-13d54da4f9ba>)
- **YouTube** React Floripa: Os mitos sobre useMemo e useCallback (https://www.youtube.com/watch?v=ZntFz_DDakQ)
- **Artigo** Your Guide to React.useCallback() (inglês) (<https://dmitripavlutin.com/dont-overuse-react-usecallback/>)

Conteúdos Alura:**

- **YouTube** Alura: Hook useCallback x useMemo - performance no React (<https://www.youtube.com/watch?v=VVmGrEZoByM>)
- **Curso** Curso React: otimizando a performance (<https://cursos.alura.com.br/course/react-otimizando-performance>)

React Hooks - Ref:

- O hook **useRef** permite que você persista valores entre renderizações.
- Criar objetos mutáveis

Conteúdos

- **Site** React: Documentação - useRef (<https://pt-br.react.dev/reference/react/useRef>)
- **Artigo** Conhecendo o useRef do React (https://medium.com/@guigaoliveira_/conhecendo-o-useref-do-react-9d67e66)
- **Artigo** Primeiros passos com React Hooks — useMemo e useCallback (<https://medium.com/@devjpnobrega/primeiros-passos-com-react-hooks-usememo-e-usecallback-13d54da4f9ba>)
- **Artigo** The Complete Guide to useRef() and Refs in React (inglês) (<https://dmitripavlutin.com/react-useref-guide/>)

Conteúdos Alura:**

- **YouTube** useRef: como funciona esse React Hook | #AluraMais (https://youtu.be/BwRxBGsT_f0?si=hxzvPvFoQywZBZ9M)
- **Curso** Curso React: otimizando a performance (<https://cursos.alura.com.br/course/react-otimizando-performance>)

React - Bibliotecas de Design System:

- Criar e manter bibliotecas que serão consumidas e usadas como padrão para a construção de um projeto

Conteúdos

- **Artigo** Criando uma biblioteca de Design System utilizando React (<https://medium.com/iclub/criando-uma-biblioteca-de-design-system-utilizando-react-214f31219a33>)
- **Artigo** Aplicando Design System com Styled Components em ReactJS (<https://medium.com/gbtech/aplicando-design-system-com-styled-components-em-reactjs-49e9f40e9bc>)
- **YouTube** Caio Gonzalez: O que é Design System? / Guia para começar o seu próprio design system (<https://www.youtube.com/watch?v=ajqbXpFVaAw>)

Conteúdos Alura:**

- **Artigo** O que é Design System? (<https://www.alura.com.br/artigos/o-que-e-design-system>)
- **Artigo** Design Systems: exemplos práticos (<https://www.alura.com.br/artigos/design-systems-exemplos-praticos>)
- **Artigo** Como publicar uma biblioteca de hooks React no npm usando Turborepo (<https://www.alura.com.br/artigos/lib-de-hooks-com-turborepo>)
- **Artigo** Atomic design no front-end: construa componentes escaláveis e modulares (<https://www.alura.com.br/artigos/atomic-design>)

- **YouTube** Alura: Design System & Style Guide (<https://www.youtube.com/watch?v=rSLvpQqC5wQ>)
- **YouTube** Alura: Design Systems (com Charles Assunção) (<https://www.youtube.com/watch?v=MLGFctEtmYQ>)
- **YouTube** Alura: O que é Design System? | Hipsters.Talks (<https://www.youtube.com/watch?v=LNQ4fZbjgWM>)
- **Podcast** Hipsters.tech: Design Systems - Hipsters 170 (<https://www.alura.com.br/podcast/hipsterstech-design-systems-hipsters-170-a399>)
- **Podcast** Layers.tech: Design System – Layers ponto tech 36 (<https://www.alura.com.br/podcast/layerstech-design-system-layers-ponto-tech-36-a1056>)
- **Curso** Curso React: criando um Design System com TailwindCSS (<https://cursos.alura.com.br/course/react-criando-design-system-tailwind>)
- **Curso** Curso React: desenvolvendo design systems modulares com Turborepo (<https://cursos.alura.com.br/course/react-desenvolvendo-design-systems-modulares-turborepo>)
- **Curso** Formação Next.js e Tailwind: construindo um design system (<https://cursos.alura.com.br/formacao-next-js-tailwind-design-system>)
- **Curso** Formação Construa um Design System com React, Turborepo e Storybook (<https://cursos.alura.com.br/formacao-design-system-react-turborepo-storybook>)

React Developer Tools:

- React Developer Tools é usado para inspecionar componentes do React, editar props e estado, e também para identificar problemas de desempenho.
- Executar o 'debugging' de aplicações

Conteúdos

- **Site** Extensão React Developer Tools para o Google Chrome (<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>)
- **Site** Documentação: Analisando componentes com o DevTools Profiler (<https://pt-br.reactjs.org/docs/optimizing-performance.html#profiling-components-with-the-devtools-profiler>)
- **Artigo** O que é React Developer Tool? (<https://acervolima.com/o-que-e-react-developer-tool/>)
- **Artigo** Debug React applications with React DevTools (inglês) (<https://blog.logrocket.com/debug-react-applications-with-the-new-react-devtools/>)
- **YouTube** Rubens Guimarães: Depuração no React com o React Developer Tools (<https://www.youtube.com/watch?v=GHoeoW88Jis>)

Conteúdos Alura:**

- **Curso** Curso React: otimizando a performance (<https://www.alura.com.br/curso-online-react-otimizando-performance>)
- **Curso** React Vite: monitorando aplicações com New Relic (<https://cursos.alura.com.br/course/react-vite-monitorando-aplicacoes-new-relic>)

Versionamento Semântico para Front-end:

- SemVer (Versionamento Semântico) é um conjunto simples de regras e requisitos que ditam como os números de versão são atribuídos e incrementados.
- Organizar as dependências de um projeto

Conteúdos

- **Site** Documentação: Versionamento Semântico (<https://semver.org/lang/pt-BR/>)
- **Artigo** Versionamento semântico - O que é e como usar (<https://imasters.com.br/codigo/versionamento-semantico-o-que-e-e-como-usar>)
- **Artigo** Criando e automatizando o versionamento semântico de projetos NodeJS (<https://dev.to/wnqueiroz/criando-e-automatizando-o-versionamento-semantico-de-projetos-nodejs-3ogj>)
- **YouTube** Mario Souto - Dev Soutinho: Como padronizar commits? Conventional Commits + Semantic Release (<https://www.youtube.com/watch?v=1eTofdmfq1g>)
- **YouTube** Filho da nuvem: O que é SEMVER e como entender versões de bibliotecas? (<https://www.youtube.com/watch?v=sgheXCK-Rfw>)

Conteúdos Alura:**

- **Artigo** Git Flow: entenda o que é, como e quando utilizar (<https://www.alura.com.br/artigos/git-flow-o-que-e-como-quando-utilizar>)
- **Artigo** Versionamento Semântico (SemVer): uma breve introdução (<https://www.alura.com.br/artigos/versionamento-semantico-breve-introducao>)
- **Artigo** Simplificando o controle de versão com Conventional Commits (<https://www.alura.com.br/artigos/simplificando-controle-versao-conventional-commits>)

CSS-in-JS:

- CSS-in-JS refere-se a um padrão onde o CSS é composto usando JavaScript em vez de ser definido em arquivos externos.
- Lidar com código CSS usando JavaScript
- Utilizar styled-components
- Conhecer Styled-JSX

Conteúdos

- **Site** React: Documentação - Estilização e CSS (<https://pt-br.reactjs.org/docs/faq-styling.html>)
- **Site** Styled Components Documentation (inglês) (<https://styled-components.com/docs/basics#motivation>)
- **Site** Microsoft Docs: Edição de estilo das estruturas CSS-in-JS (<https://docs.microsoft.com/pt-br/microsoft-edge/devtools-guide-chromium/css/css-in-js>)
- **Site** Criando Styled Components Com React - Blog do Matheus Castiglioni (<https://blog.matheuscastiglioni.com.br/criando-styled-components-com-react/>)
- **Site** Chrome Developers: CSS-in-JS support in DevTools (inglês) (<https://developer.chrome.com/blog/css-in-js/>)
- **YouTube** Mario Souto - Dev Soutinho: Os poderes do CSS in JS (<https://www.youtube.com/watch?v=QdfjWRc4ySA>)
- **YouTube** Mario Souto - Dev Soutinho: CSS-in-JS na prática! Criando componentes com React e CSS no NextJS (<https://www.youtube.com/watch?v=aDKxJfJiM28>)
- **Artigo** A Thorough Analysis of CSS-in-JS (inglês) (<https://css-tricks.com/a-thorough-analysis-of-css-in-js/>)

Conteúdos Alura:**

- **Artigo** CSS com NextJS (<https://www.alura.com.br/artigos/css-com-next-js>)

- **Curso** Imersão React: AluraCord - Aula 01 (<https://cursos.alura.com.br/imersoes/aulas/aula-01-criando-o-nosso-projeto-aluracord-c68>)
- **Curso** Curso React: estilize componentes com Styled Components e manipule arquivos estáticos (<https://www.alura.com.br/curso-online-react-estilize-componentes-styled-components-manipule-arquivos-estaticos>)
- **Curso** Curso React: desenvolvendo componentes customizados e acessíveis com Emotion (<https://cursos.alura.com.br/course/react-componentes-customizados-acessiveis-emotion>)

Styled Components:

- Lidar com código CSS em componentes

Conteúdos

- **Artigo** Criando Styled Components Com React (<https://blog.matheuscastiglioni.com.br/criando-styled-components-com-react/>)
- **Site** Documentação - Styled Components (inglês) (<https://styled-components.com/docs>)
- **YouTube** Mario Souto - Dev Soutinho: Styled Components, por onde começar? Os poderes do CSS in JS (<https://www.youtube.com/watch?v=QdfjWRc4ySA>)
- **YouTube** Mario Souto - Dev Soutinho: Como começar a estruturar um projeto React? com NextJS, Styled Components (<https://www.youtube.com/watch?v=mJK5oGixSYo>)
- **YouTube** Marco Bruno: Como usar styled-components - Introdução (<https://www.youtube.com/watch?v=B9z51O1ctzU>)
- **YouTube** BrasilJS: Efeito Stagger com React e Styled Components (<https://www.youtube.com/watch?v=HGrXz6vHngI>)

Conteúdos Alura:**

- **Artigo** React: componentes com Styled Components (<https://www.alura.com.br/artigos/react-componentes-com-styled-components>)
- **Site** Primeiras aulas do curso React: abstraindo seu CSS com Styled Components (<https://www.alura.com.br/conteudo/react-styled-components>)
- **Curso** Curso React: abstraindo seu CSS com Styled Components (<https://www.alura.com.br/curso-online-react-styled-components>)

React - Roteamento web:

- Manipular a navegação entre interfaces e componentes

Conteúdos

- **Site** Documentação: Divisão de Código Baseada em Rotas (<https://pt-br.reactjs.org/docs/code-splitting.html#route-based-code-splitting>)
- **Artigo** Roteamento no React com os poderes do React Router v4 (<https://medium.com/collabcode/roteamento-no-react-com-os-poderes-do-react-router-v4-fbc191b9937d>)
- **Artigo** ReactJS: Manipulando rotas com react-router (<https://ateliware.com/blog/react-router>)
- **Artigo** Tutorial de React Router (<https://www.freecodecamp.org/portuguese/news/tutorial-de-react-router-como-usar-render-redirect-switch-link-e-mais-com-exemplos-de-codigo/>)
- **YouTube** Kevin Beltrão: Rotas com React Router Dom (https://www.youtube.com/watch?v=9pB_lwmlc74)

Conteúdos Alura:**

- **YouTube** Alura: Como configurar o React Router para sua SPA (<https://www.youtube.com/watch?v=4Tb8dp5GYqI>)
- **Curso** Curso React: desenvolvendo em React Router com JavaScript (<https://cursos.alura.com.br/course/React-desenvolvendo-react-router-javascript>)
- **Curso** Curso React: conhecendo a biblioteca React Router (<https://www.alura.com.br/curso-online-react-biblioteca-react-router>)

React - Gerenciamento de Estados:

- O gerenciamento de estado em React é a prática de controlar e atualizar os dados de uma aplicação React para garantir que os componentes tenham acesso às informações necessárias. Isso é essencial para criar aplicativos interativos e dinâmicos.
- Garante que os dados exibidos nos componentes estejam sempre atualizados e em sincronia.
- Permite que componentes compartilhem dados e comuniquem-se uns com os outros.
- Otimiza a renderização, garantindo que apenas as partes relevantes da interface do usuário sejam atualizadas.
- Torna o código mais organizado e fácil de entender, especialmente em aplicativos complexos.

Conteúdos

- **Site** Passando dados profundamente com contexto (inglês) (<https://react.dev/learn/passing-data-deeply-with-context>)
- **Artigo** Redux-Saga: Gerenciando efeitos e ações (<https://oieduardorabelo.medium.com/redux-saga-gerenciando-efeitos-f518a31c744e>)
- **Artigo** Recoil.js — Gerenciamento de Estado de Alto Desempenho para React Simplificado (inglês) (<https://medium.com/weekly-webtips/recoil-js-high-performance-state-management-for-react-simplified-cb520464a667>)
- **YouTube** A Context API é um state management?! A forma CORRETA de utilizar a CONTEXT API do React.js (https://youtu.be/06_575UN55A?si=eMQQ1lz62BYa5Y-)
- **YouTube** Redux // Dicionário do Programador - Código Fonte TV (<https://youtu.be/kAOzxQp6AKE?si=bpYO9rlG6QbPhwO7>)

Conteúdos Alura:**

- **Site** Primeiras aulas do curso de React: gerenciamento de estados globais com ContextAPI (<https://www.alura.com.br/conteudo/react-context-estados-globais-contextapi>)
- **Artigo** Context API e TypeScript: como utilizar o superpoder desta dupla (<https://www.alura.com.br/artigos/context-api-typescript-utilizar-superpoder-dupla>)
- **Artigo** Estados globais: diferenças entre Redux e Context API (<https://www.alura.com.br/artigos/estados-globais-diferencas-redux-context-api>)
- **Artigo** React e Context API: testes de unidade (<https://www.alura.com.br/artigos/react-context-api-testes-de-unidade>)
- **Artigo** Guia de bibliotecas para lidar com gerenciamento de estados em projetos React (<https://www.alura.com.br/artigos/guia-de-bibliotecas-react>)
- **Artigo** Redux: desvendando a arquitetura com Flux (<https://www.alura.com.br/artigos/redux-desvendando-arquitetura-flux>)
- **Artigo** Ainda vale a pena aprender Redux? – vem ver nossa opinião e o motivo dela (<https://www.alura.com.br/artigos/vale-a-pena-aprender-redux>)

- **Artigo** Controlando múltiplas instâncias dinâmicas com atomFamily e selectorFamily (<https://www.alura.com.br/artigos/atom-selector-family/>)
- **Artigo** Observando os estados de uma aplicação React com a biblioteca MobX (<https://www.alura.com.br/artigos/mobx>)
- **Curso** Curso de React: gerenciamento de estados globais com ContextAPI (<https://www.alura.com.br/curso-online-react-context-estados-globais-contextapi>)
- **Curso** Curso de React: arquitetando soluções de roteamento e gestão de estado com React Router e Context API (<https://www.alura.com.br/curso-online-solucoes-roteamento-gestao-estado-react-router-context-api>)
- **Curso** Curso de React: combinando Context API e Apollo Client (<https://www.alura.com.br/curso-online-react-combinando-context-api-apollo-client>)
- **Curso** Curso de React: gerenciando estado com Recoil (<https://www.alura.com.br/curso-online-react-gerenciando-estado-recoil>)
- **Curso** Curso de React: gerenciamento de estados globais com Redux (<https://www.alura.com.br/curso-online-react-gerenciamento-estados-globais-redux>)
- **Curso** Curso de Redux Toolkit: explorando a arquitetura Flux (<https://cursos.alura.com.br/course/redux-toolkit-explorando-arquitetura-flux>)
- **Curso** Formação Gerencie estados em React com Redux (<https://www.alura.com.br/formacao-react-gestao-estados>)
- **Curso** Mobx: gerenciando estados em aplicações React (<https://cursos.alura.com.br/course/mobx-gerenciando-estados-aplicacoes-react>)
- **Curso** Formação React: domine o gerenciamento de estados (<https://cursos.alura.com.br/formacao-react-gerenciamento-estados>)
- **Podcast** Hipsters.tech: Guia do iniciante em React 209 (<https://www.alura.com.br/podcast/guia-do-iniciante-em-react-hipsters-209-a341>)

TypeScript - Fundamentos:

- TypeScript é uma linguagem de programação fortemente tipada que se baseia em JavaScript.
- Entender a fundo o que são tipos e a importância da tipagem
- Aprender o que é o TypeScript, por que foi criado, como ele funciona e sua relação com o JavaScript
- Conhecer as ferramentas do TypeScript (integração com o editor de código, verificador estático e compilador)
- Escrever código em TypeScript com suas ferramentas (interfaces, enum, decorators, etc)
- Desenvolver aplicações em TypeScript

Conteúdos

- **Site** Microsoft Docs: Compreendendo o TypeScript (<https://docs.microsoft.com/pt-br/archive/msdn-magazine/2015/january/typescript-understanding-typescript>)
- **Artigo** TypeScript: O guia definitivo (<https://oieduardorabelo.medium.com/typescript-o-guia-definitivo-1a63b04259cc>)
- **YouTube** Mario Souto - Dev Soutinho: Linguagens Sem Tipos, dá pra ficar bom programando nelas? JavaScript vs TypeScript (<https://www.youtube.com/watch?v=t-9-pkh5FDQ>)
- **Site** Documentação: TypeScript para o Novo Programador (<https://www.typescriptlang.org/pt/docs/handbook/typescript-from-scratch.html>)

Conteúdos Alura:**

- **Artigo** Javascript ou Typescript? (<https://www.alura.com.br/artigos/javascript-ou-typescript>)
- **Podcast** Hipsters.tech: O Hype do TypeScript - Hipsters Ponto Tech #207 (<https://www.alura.com.br/podcast/o-hype-do-typescript-hipsters-207-a344>)
- **Podcast** Hipsters.tech: TechGuide: Typescript - Hipsters Ponto Tech #378 (<https://www.alura.com.br/podcast/hipsterstech-techguide-typescript-hipsters-ponto-tech-378-a9120>)
- **YouTube** Alura: Quais são as principais vantagens do Typescript? (<https://www.youtube.com/watch?v=snwevp02slw>)
- **YouTube** Alura: Imersão React Live 02 - TypeScript no nosso projeto React (<https://www.youtube.com/watch?v=SpWFDTvgS9k>)
- **Curso** Formação TypeScript: desenvolva front-end com mais produtividade (<https://cursos.alura.com.br/formacao-typescript-desenvolva-front-end-productividade>)
- **Curso** Formação Typescript (<https://www.alura.com.br/formacao-typescript>)
- **Curso** Formação React com Typescript (<https://www.alura.com.br/formacao-react-ts>)
- **Curso** Curso Typescript: construção de uma API com tipagem segura (<https://cursos.alura.com.br/course/typescript-construcao-api-tipagem-segura>)
- **Curso** Curso TypeScript: desenvolvendo validações e tratando erros (<https://cursos.alura.com.br/course/typescript-desenvolvendo-validacoes-tratando-erros>)
- **Curso** Padrões de projeto com TypeScript: aprimorando uma API com arquitetura limpa (<https://www.alura.com.br/curso-online-padroes-projeto-typescript-api-arquitetura-limpa>)

React Testing Library:

- A React Testing Library se baseia na DOM Testing Library adicionando APIs para trabalhar com componentes React. É um conjunto de auxiliares que permitem testar componentes React sem depender de seus detalhes de implementação.
- Testar componentes React

Conteúdos

- **Site** React: Documentação - Visão Geral sobre Testes (<https://pt-br.reactjs.org/docs/testing.html#gatsby-focus-wrapper>)
- **YouTube** Washington Developer: React: Teste seus componentes | Jest | Testing Library (<https://www.youtube.com/watch?v=pbwXsjyEMqg>)
- **Site** React Testing Library: Documentação (inglês) (<https://testing-library.com/docs/react-testing-library/intro/>)
- **Site** Desmistificando testes no Frontend (<https://marcosviniciosneves.medium.com/desmistificando-testes-no-frontend-52afaeb6b35a>)

Conteúdos Alura:**

- **Artigo** Testes automatizados: como melhorar qualidade e confiabilidade do seu software (<https://www.alura.com.br/artigos/testes-automatizados>)
- **Curso** Curso React JS: crie testes com Jest e Testing Library e garanta o funcionamento do Front-end (<https://cursos.alura.com.br/course/react-js-testes-automatizados-front-end>)
- **Curso** Formação Melhore sua aplicação React com testes automatizados (<https://cursos.alura.com.br/formacao-react-testes-automatizados>)
- **Curso** Curso React: testando os seus componentes (<https://cursos.alura.com.br/course/react-testando-componentes>)

React - Jest:

- Jest é um framework de teste em JavaScript projetado para garantir a correção de qualquer código JavaScript. Ele permite que você escreva testes com uma API acessível, familiar e rica em recursos que lhe dá resultados rapidamente.
- Testar componentes

Conteúdos

- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software! (<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Site** Jest: Documentação (<https://jestjs.io/pt-BR/docs/getting-started>)
- **Site** Jest: Testando Apps React (<https://jestjs.io/pt-BR/docs/tutorial-react>)
- **YouTube** Washington Developer: React: Teste seus componentes | Jest | Testing Library (<https://www.youtube.com/watch?v=pbwXsjyEMqg>)
- **YouTube** Matheus Castiglioni: Jest - Iniciando com testes e TDD (<https://www.youtube.com/watch?v=sLx00kXPoTg>)

Conteúdos Alura:**

- **Artigo** Testes automatizados: como melhorar qualidade e confiabilidade do seu software (<https://www.alura.com.br/artigos/testes-automatizados>)
- **Curso** Curso React: testando os seus componentes (<https://cursos.alura.com.br/course/react-testando-componentes>)
- **Curso** Formação Melhore sua aplicação React com testes automatizados (<https://cursos.alura.com.br/formacao-react-testes-automatizados>)

Cypress:

- Criar e executar testes

Conteúdos

- **Site** Cypress: Documentação (inglês) (<https://docs.cypress.io/guides/getting-started/installing-cypress>)
- **Artigo** Um Overview sobre Cypress.io — Framework de Automação de Testes end-to-end (<https://medium.com/@faelbercam/um-overview-sobre-cypress-io-framework-de-automa%C3%A7%C3%A3o-de-testes-end-to-end-dc438b9ee7a1>)
- **YouTube** Otavio Lemos: Guia Rápido Sobre Cypress (<https://www.youtube.com/watch?v=e5SuUUFIDss>)

Conteúdos Alura:**

- **YouTube** Alura: Testes em Javascript - Alura Live 50 (<https://www.youtube.com/watch?v=r1rndQwFLMY>)
- **Curso** Curso React: escreva testes end to end com Cypress (<https://cursos.alura.com.br/course/react-escreva-testes-end-to-end-cypress>)
- **Curso** Curso React: avançando em testes e2e com Cypress (<https://cursos.alura.com.br/course/react-avancando-testes-e2e-cypress>)
- **Curso** Formação Melhore sua aplicação React com testes automatizados (<https://cursos.alura.com.br/formacao-react-testes-automatizados>)
- **Site** Primeiras aulas do curso Cypress: automação de testes E2E (<https://www.alura.com.br/conteudo/cypress-automatizando-testes-e2e>)

- **Curso** Curso Cypress: automação de testes E2E (<https://www.alura.com.br/curso-online-cypress-automacao-testes-e2e>)
- **Curso** Formação Carreira QA: processos e automação de testes (<https://www.alura.com.br/formacao-carreira-tester-qa>)

JavaScript - Callbacks e Promises:

- Uma promessa (Promise) é um proxy para um valor não necessariamente conhecido quando a promessa é criada. Isso permite que métodos assíncronos retornem valores como métodos síncronos - em vez de retornar imediatamente o valor final, o método assíncrono retorna uma promessa de fornecer o valor em algum momento no futuro.
- Uma função de Callback é uma função passada para outra função como um argumento, que é então invocado dentro da função externa para completar algum tipo de rotina ou ação.
- Uma função assíncrona (async) é uma função declarada com a palavra-chave `async`, e a palavra-chave `await` é permitida dentro dela. As palavras-chave `async` e `await` permitem que o comportamento assíncrono e baseado em promessas seja escrito em um estilo mais limpo, evitando a necessidade de configurar explicitamente as cadeias de promessas.
- Entender o conceito de assincronicidade em programação
- Escrever código assíncrono entendendo o conceito de promessas em JavaScript
- Utilizar os métodos, palavras-chaves e objetos do JavaScript para manipulação de promessas como 'Async/Await', '.then()', 'Promise', etc
- Aprender em quais situações é necessário o uso de programação assíncrona
- Fazer chamadas em APIs com `fetch()`

Conteúdos

- **Site** MDN Web Docs: Introdução ao JavaScript Async (<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Asynchronous/Introducing>)
- **Site** MDN Web Docs: Funções assíncronas (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/async_function)
- **Site** MDN Web Docs: Promise (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Promise)
- **Site** MDN Web Docs: Função Callback (https://developer.mozilla.org/pt-BR/docs/Glossary/Callback_function)
- **YouTube** Mario Souto - Dev Soutinho: Como usar Async/Await? Promises no JavaScript? (<https://www.youtube.com/watch?v=q28IfkBd9F4>)
- **YouTube** Mario Souto - Dev Soutinho: Callbacks, Síncrono, Assíncrono e Event Loop no JavaScript (<https://www.youtube.com/watch?v=6lbBaM18X3g>)
- **Artigo** Entendendo como fazer AJAX com a FetchAPI (<https://medium.com/@omariosouto/entendendo-como-fazer-ajax-com-a-fetchapi-977ff20da3c6>)
- **YouTube** Mario Souto - Dev Soutinho: Como pegar dados de uma API? Como fazer AJAX? | Pegando dados de serviços via JavaScript (<https://www.youtube.com/watch?v=85vJXFpXLQw>)
- **Artigo** Revolução no Node.js: adeus ao Axios e fetch API na versão 17.5.0 (<https://www.alura.com.br/artigos/revolucao-node-js-adeus-axios-fetch-api-versao-17-5-0>)

Conteúdos Alura:**

- **Artigo** Async/await no JavaScript: o que é e quando usar a programação assíncrona? (<https://www.alura.com.br/artigos/async-await-no-javascript-o-que-e-e-quando-usar>)

- **Site** Alura+: JavaScript assíncrono e Fetch (<https://cursos.alura.com.br/extra/alura-mais/javascript-assincrono-e-fetch-c93>)
- **Curso** Curso JavaScript: consumindo e tratando dados de uma API (<https://cursos.alura.com.br/course/javascript-consumindo-tratando-dados-uma-api>)

JavaScript - Manipulação de Erros:

- O tratamento de erros refere-se aos procedimentos de resposta e recuperação de condições de erro presentes em um aplicativo de software. Em outras palavras, é o processo composto de antecipação, detecção e resolução de erros de aplicação, de programação ou de comunicação.
- Conhecer e tratar as exceções mais comuns
- Saber quais os tipos de erros e em quais situações eles podem ocorrer
- Entender como o Node.js faz o manejo de erros
- Usar 'try' e 'catch' para tratamento de erros
- Em que ocasiões e de que forma utilizar o **throw**
- Criar exceções específicas de acordo com a necessidade de sua aplicação

Conteúdos

- **Site** MDN Web Docs: Controle de Fluxo e Manipulação de Erro (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Control_flow_and_error_handling#declara%C3%A7%C3%B5es_de_manipula%C3%A7%C3%A3o_de_erros)
- **Site** MDN Web Docs: Error (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Error)
- **Site** Microsoft TechNet: JavaScript: Tratamento de erros (<https://social.technet.microsoft.com/wiki/pt-br/contents/articles/32455.javascript-tratamento-de-erros.aspx>)
- **Artigo** Exceções e Erros em JavaScript (<http://gabrielprates.com/2017/06/02/excecoes-e-erros-em-js.html>)
- **Artigo** Criando Exceptions para impressionar no Teste Técnico (<https://dev.to/he4rt/criando-exceptions-para-impressionar-no-teste-tecnico-2nie>)

Conteúdos Alura:**

- **Artigo** Lidando com erros no Node.js (<https://www.alura.com.br/artigos/lidando-com-erros-node-js>)
- **Curso** Curso Node.js: Criando sua primeira biblioteca (<https://cursos.alura.com.br/course/javascript-node-js-criando-primeira-biblioteca>)

Babel - Fundamentos:

- Babel é um compilador JavaScript. É uma ferramenta usada principalmente para converter código ECMAScript 2015+ em uma versão compatível com versões anteriores do JavaScript em navegadores ou ambientes atuais e mais antigos.
- Compreender como Babel converte a sintaxe para JavaScript

Conteúdos

- **Site** Documentação: What is Babel? (inglês) (<https://babeljs.io/docs/en/>)
- **Artigo** O que é babel? (<https://coodesh.com/blog/dicionario/o-que-e-babel/>)
- **Artigo** React e Babel — Como melhorar o desempenho de suas aplicações com plugins, presets e benchmark! (<https://medium.com/nossa-coletividade/react-e-babel-como-melhorar-o-desempenho-de-suas-aplica%C3%A7%C3%B5es-com-plugins-presets-e-benchmark-fb35700a52e9>)

- **YouTube** Matheus Castiglioni: React, Babel e Webpack, o que é tudo isso? (<https://www.youtube.com/watch?v=y5B0MXmt428>)
- **YouTube** web3escola: Introdução ao React - Babel (<https://www.youtube.com/watch?v=jSGg2j2UNr4>)

Conteúdos Alura:**

- **Curso** Curso JavaScript: salvando dados localmente com IndexedDB (<https://www.alura.com.br/curso-online-javascript-es6-orientacao-a-objetos-parte-3>)

Nível 3

Lottie:

- Lottie é uma biblioteca que processa animações do Adobe After Effects exportadas como json com Bodymovin e as renderiza nativamente em dispositivos mobile e na web.

Conteúdos

- **Site** Documentação: Lottie (inglês) (<https://airbnb.io/lottie/#/README>)
- **Site** Componentes Lottie (inglês) (<https://lottiereact.com/components/Lottie>)
- **Artigo** Animações com Lottie no React Native (<https://medium.com/@victor96novais/anima%C3%A7%C3%B5es-com-lottie-no-react-native-d6f8ffa749a9>)
- **YouTube** Mario Souto - Dev Soutinho: Lottie: Animações com React (<https://www.youtube.com/watch?v=dZH9aDX8T-0>)
- **YouTube** Wendel Dev: Como usar animações Lottie em apenas 3 minutos com ReactJS (https://www.youtube.com/watch?v=bWo_wlqPfQY)

Conteúdos Alura:**

- **Podcast** Layers.tech: WEB Animation LottieFiles (<https://www.alura.com.br/podcast/web-animation-lottiefiles-layers-ponto-tech-42-a1107>)

Framer Motion:

- Criar animações

Conteúdos

- **Site** Documentação: Framer Motion (inglês) (<https://www.framer.com/motion/>)
- **Artigo** Framer Motion + React (Exemplo de Utilização) (<https://medium.com/@leonardord99/framer-motion-react-514f39a9d03e>)
- **Artigo** Galeria De Páginas Deslizantes Animadas Usando Framer-motion E React.js (<https://acervolima.com/galeria-de-paginas-deslizantes-animadas-usando-framer-motion-e-react-js/>)
- **YouTube** PedroTech: Animations In React - Framer-Motion Tutorial (inglês) (<https://www.youtube.com/watch?v=GOuwOI-WSkE>)

Conteúdos Alura:**

- **Artigo** TL Draw: o que é e como escrever códigos com desenhos (<https://www.alura.com.br/artigos/tl-draw>)
- **Curso** Framer: construindo um portal do colaborador com IA (<https://cursos.alura.com.br/course/framer-construindo-portal-colaborador-ia>)

- **Curso** Framer: criando seu e-commerce com IA (<https://cursos.alura.com.br/formacao-framer-criando-e-commerce-ia>)

Service Workers:

- Melhorar a performance e trabalhar offline;
- Permitem que aplicações web armazenem recursos no cache;
- Interceptam e gerenciam requisições de rede;
- Facilitam notificações push, mesmo quando o app não está aberto ou usuário não está na página;
- Funcionam como um intermediário entre o navegador e a rede/servidor;
- São assíncronos e não acessam diretamente o DOM, garantindo segurança e eficiência;

Conteúdos

- **Site** MDN Web Docs: Usando Service Worker (https://developer.mozilla.org/pt-BR/docs/Web/API/Service_Worker_API/Using_Service_Workers)
- **Artigo** Os primeiros passos para tornar uma aplicação web offline (<https://medium.com/aurum-tech/os-primeiros-passos-para-tornar-uma-aplica%C3%A7%C3%A3o-web-offline-8dee5958818b>)
- **Artigo** Como o JavaScript funciona: Service Workers, seu ciclo de vida e casos de uso (<https://medium.com/reactbrasil/como-o-javascript-funciona-service-workers-seu-ciclo-de-vida-e-casos-de-uso-8404870ee728>)
- **Artigo** Service Workers in React (inglês) (<https://levelup.gitconnected.com/service-workers-in-react-8bcaccfd0356>)
- **YouTube** Matheus da Guia de Campos: Configurando service-worker em sua aplicação Reactjs para cache de páginas estáticas (<https://www.youtube.com/watch?v=dBigQB4p1zY>)

Conteúdos Alura:**

- **Artigo** Offline Web (<https://www.alura.com.br/artigos/palestra-offline-web-com-service-workers>)
- **Artigo** Desvendando os service workers: ciclo de vida e atualização (<https://www.alura.com.br/artigos/service-worker>)
- **Artigo** Quais as boas práticas de segurança para PWAs? (<https://www.alura.com.br/artigos/praticas-de-seguranca-pwas>)
- **Artigo** Cuidados com CSRF/XSRF ao utilizar a Push API (<https://www.alura.com.br/artigos/cuidados-com-csrf>)
- **Podcast** Hipsters.tech: Progressive Web Apps - Hipsters 03 (<https://www.alura.com.br/podcast/progressive-web-apps-hipsters-03-a583>)
- **YouTube** O que é uma Progressive Web App (PWA)? (<https://www.youtube.com/watch?v=UeMkXvmk68Q>)
- **Curso** Formação React: construindo PWA's modernas e eficientes (<https://www.alura.com.br/formacao-react-construindo-pwas-modernas-eficientes>)

React - Validação de formulários:

- Em React, é super importante que, em formulários, os dados inseridos pelos usuários atendam a critérios específicos antes do envio ou processamento das informações.
- Conhecer o Formik e o React Hook Form, bibliotecas populares para lidar com formulários em React.

- Conhecer o Yup e o Zod, ambas bibliotecas de validação baseadas em esquema. Permitem a definição de esquemas para validar valores de um objeto.
- Compreender como as bibliotecas trabalham, facilitando a criação de formulários complexos com validação e gerenciamento de estado.

Conteúdos

- **Site** React Hook Form: Documentação (inglês) (<https://react-hook-form.com/get-started>)
- **Artigo** Dev.to: React Hook Form (<https://dev.to/stanley/react-hook-form-1hk>)
- **YouTube** Huriel: Como usar o React Hook Form! (<https://www.youtube.com/watch?v=QEEdm3EVrhY0>)

Conteúdos Alura:**

- **Artigo** React: validação de formulários com React Hook Form (<https://www.alura.com.br/artigos/react-hook-form>)
- **Artigo** Criando esquemas de validação com Yup (<https://www.alura.com.br/artigos/validacao-yup>)
- **Curso** React: bibliotecas para desenvolver formulários (<https://cursos.alura.com.br/formacao-react-bibliotecas-desenvolver-formularios>)

React - Gerenciamento de Performance:

- Em React, uma boa performance garante uma experiência de usuário fluida e responsiva.
- A otimização de performance envolve técnicas para minimizar a quantidade de renderizações e maximizar a eficiência da atualização da interface.
- Entender o que é otimização de performance em React.
- Conhecer e saber usar técnicas como Lazy Loading e Code Splitting.
- Conhecer as demais ferramentas e práticas para otimização de performance no React.

Conteúdos

- **Site** React: Documentação - Performance Hooks (<https://pt-br.react.dev/reference/react/hooks#performance-hooks>)
- **Artigo** Victor Rios: Performance em Aplicações React: Dicas e Técnicas de Otimização (<https://dev.to/vitorrios1001/performance-em-aplicacoes-react-dicas-e-tecnicas-de-otimizacao-414i>)
- **Artigo** TabNews: Performance tuning em ReactJS (<https://www.tabnews.com.br/gabrielOak/performance-tuning-em-reactjs>)
- **YouTube** Sujeito Programador: Como melhorar a performance do seu projeto com React JS (<https://youtu.be/ENcr7ssBaMg?si=haaDbHWg0Qbp9v-s>)

Conteúdos Alura:**

- **Artigo** Métricas de performance centradas no usuário (<https://www.alura.com.br/artigos/metricas-performance-centradas-no-usuario>)
- **Artigo** Como construir aplicações SSR com React, React Router e Vite (<https://www.alura.com.br/artigos/vite-ssr-react>)
- **Podcast** Hipsters.tech: Performance na Web 2: SSR, Core Web Vitals e Front-end (<https://cursos.alura.com.br/extra/hipsterstech/performance-na-web-2-ssr-core-web-vitals-e-front-end-hipsters-ponto-tech-416-a9348>)
- **Podcast** Hipsters.tech: Observabilidade no Itaú (<https://cursos.alura.com.br/extra/hipsterstech/observabilidade-no-itaui-hipsters-ponto-tech->)

[334-a1927\)](#)

- **Curso** React: maximizando a performance de Aplicações (<https://cursos.alura.com.br/formacao-react-maximizando-performance-aplicacoes>)

Lodash:

- Lodash é uma biblioteca de utilitários JavaScript moderna que oferece modularidade, desempenho e extras. Os métodos modulares do Lodash são ótimos para iterar arrays, objetos e strings; manipular e testar valores; e criar funções compostas.

Conteúdos

- **Site** Documentação: Lodash (inglês) (<https://lodash.com/>)
- **Artigo** Limpando seu código javascript com Lodash (<https://medium.com/codigorefinado/limpando-seu-c%C3%B3digo-javascript-com-lodash-93ca10f3f1fe>)
- **YouTube** Tiago Matos: Conhecendo o Lodash, uma biblioteca javascript (<https://www.youtube.com/watch?v=ly7D-YL-rdU>)
- **YouTube** Tiago Matos: 10 funções imperdíveis da biblioteca javascript lodash (<https://www.youtube.com/watch?v=2JKGdadhFxU>)

Conteúdos Alura:**

- **Curso** Livro Casa do Código: Primeiros passos com Node.js (<https://www.casadocodigo.com.br/products/livro-primeiros-passos-node>)

GraphQL:

- GraphQL é uma linguagem de consulta e manipulação de dados de código aberto para APIs. É considerada uma alternativa para arquiteturas REST.
- Aprender o que é GraphQL e por que foi criado
- Entender como o GraphQL é utilizado no desenvolvimento de APIs
- Criar APIs utilizando as bibliotecas e frameworks para GraphQL

Conteúdos

- **Site** RedHat: O que é GraphQL? (<https://www.redhat.com/pt-br/topics/api/what-is-graphql>)
- **Artigo** Dev.to: Entendendo um pouco sobre o que é GraphQL (<https://dev.to/amaurygoncalves/entendendo-um-pouco-sobre-o-que-e-graphql-9l7>)
- **YouTube** Mateus Silva: GraphQL: Pra que serve, Conceitos e Hello World (<https://youtu.be/7RoHxSGVAdU>)
- **Site** How to GraphQL: The Fullstack tutorial for GraphQL (em inglês) (<https://www.howtographql.com/>)

Conteúdos Alura:**

- **Artigo** Guia essencial do GraphQL para pessoas que desenvolvem frontend (<https://www.alura.com.br/artigos/guia-graphql-pessoas-desenvolvem-frontend>)
- **YouTube** Alura: O que é GraphQL? (https://www.youtube.com/watch?v=trf3ZR_K1nk)
- **Podcast** Hipsters.tech: Python, GraphQL e Serverless no Zé Delivery (Hipsters On The Road 53) (<https://www.alura.com.br/podcast/python-graphql-e-serverless-no-ze-delivery-hipsters-on-the-road-53-a804>)
- **Curso** React: executando consultas dinâmicas com Apollo Client (<https://cursos.alura.com.br/course/react-executando-consultas-dinamicas-apollo-client>)

- **Curso** Curso React: combinando Context API e Apollo Client (<https://cursos.alura.com.br/course/react-combinando-context-api-apollo-client>)
- **Curso** Curso de React Query: gerenciando HTTP State (<https://cursos.alura.com.br/course/react-query-gerenciando-http-state>)
- **Curso** React Query: manipulando interações com mutations (<https://www.alura.com.br/curso-online-react-query-manipulando-interacoes-mutations>)

Apollo Client:

- Apollo Client é uma biblioteca abrangente de gerenciamento de estado para JavaScript que permite gerenciar dados locais e remotos com o GraphQL.
- Utilizar o Apollo para criar um servidor GraphQL
- Conectar com uma API

Conteúdos

- **Site** Documentação: Apollo Client (inglês) (<https://www.apollographql.com/docs/react/>)
- **Artigo** GraphQL hoje usando Apollo em aplicações que utilizam APIs REST (<https://imasters.com.br/apis-microservicos/graphql-hoje-usando-apollo-em-aplicacoes-que-utilizam-apis-rest>)
- **Artigo** Consumindo API GraphQL usando React e Apollo Client (<https://dev.to/mensonones/consumindo-api-graphql-usando-react-e-apollo-client-4blo>)
- **Artigo** Apollo Client + Angular 7 : Configurando Apollo Client com o poderoso ng add (<https://medium.com/@gustavo93.lima/apollo-client-angular-7-configurando-apollo-client-com-o-poderoso-ng-add-a7788b63d4b3>)

Conteúdos Alura:**

- **Curso** Curso React: executando consultas dinâmicas com Apollo Client (<https://cursos.alura.com.br/course/react-executando-consultas-dinamicas-apollo-client>)
- **Curso** Curso React: combinando Context API e Apollo Client (<https://cursos.alura.com.br/course/react-combinando-context-api-apollo-client>)
- **Curso** Curso de React Query: gerenciando HTTP State (<https://cursos.alura.com.br/course/react-query-gerenciando-http-state>)

React - Arquitetura:

- Manter a arquitetura do nosso projeto bem estruturada, nos ajuda a organizar o código da melhor maneira possível para facilitar a manutenção e escalabilidade. Combinada aos princípios do SOLID, ela potencializa ainda mais toda essa organização minimizando o acoplamento e aprimorando a coesão dos componentes da nossa aplicação.
- Estruturar o projeto React com Arquitetura Limpa;
- Conhecer e aplicar os princípios SOLID para maior reusabilidade e coesão;
- Isolar responsabilidades com hooks e injeção de dependências.

Conteúdos

- **Artigo** Arquitetura para aplicações React (<https://vinniciusgomes.medium.com/arquitetura-para-aplicacoes-react-d77361bf2d49>)
- **Artigo** O que é SOLID: O guia completo para você entender os 5 princípios da POO (<https://medium.com/desenvolvendo-com-paixao/o-que-%C3%A9-solid-o-guia-completo-para-voc%C3%AA-entender-os-5-princ%C3%ADpios-da-poo-2b937b3fc530>)

Conteúdos Alura:**

- **YouTube** Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **YouTube** Arquitetura de sistemas (<https://www.youtube.com/watch?v=oedWxgAZc2A>)
- **Podcast** SOLID: Código bom e bonito – Hipsters Ponto Tech #219 (<https://www.hipsters.tech/solid-codigo-bom-e-bonito-hipsters-ponto-tech-219/>)
- **Podcast** Arquitetura no ecossistema de Front-End – Hipsters Ponto Tech #441 (<https://www.hipsters.tech/arquitetura-no-ecossistema-de-front-end-hipsters-ponto-tech-441/>)
- **Artigo** SOLID: o que é e quais os 5 princípios da Programação Orientada a Objetos (POO) (<https://www.alura.com.br/artigos/solid>)
- **Curso** Curso de SOLID aplicado ao React: melhorando a modularidade e flexibilidade do código (<https://cursos.alura.com.br/course/solid-aplicado-react-melhorando-modularidade-flexibilidade-codigo>)
- **Curso** Curso de React: integração com Supabase e conceitos de arquitetura limpa na prática (<https://cursos.alura.com.br/course/react-integracao-supabase-conceitos-arquitetura-limpa-pratica>)
- **Curso** Formação React: da teoria à prática em arquitetura e design de aplicações (<https://cursos.alura.com.br/formacao-react-arquitetura-design-aplicacoes>)

Redux Saga:

- Redux Saga é uma biblioteca que visa tornar os efeitos colaterais da aplicação (ou seja, coisas assíncronas, como busca de dados e coisas impuras, como acessar o cache do navegador) mais fáceis de gerenciar, mais eficientes de executar, fáceis de testar e melhores no tratamento de falhas.
- Criar e manter chamadas assíncronas e efeitos

Conteúdos

- **Site** Documentação: Redux Saga (inglês) (<https://redux-saga.js.org/docs/introduction/GettingStarted>)
- **Artigo** Redux-Saga: Gerenciando efeitos e ações (<https://oieduardorabelo.medium.com/redux-saga-gerenciando-efeitos-f518a31c744e>)
- **Artigo** Redux Saga: Você no controle das operações assíncronas (<https://medium.com/nossa-coletividade/redux-saga-voc%C3%AA-no-controle-das-opera%C3%A7%C3%B5es-ass%C3%ADncronas-71c9e6b3aabc>)
- **YouTube** Kuze: Entendendo o fluxo do Redux Saga - Criando um app (<https://www.youtube.com/watch?v=5NFIRaXwrOQ>)
- **YouTube** Programador de Sucesso: O guia definitivo para aprender REDUX e REDUX SAGA (<https://www.youtube.com/watch?v=7YoMyQ-jsBk>)

Conteúdos Alura:**

- **Artigo** Redux: desvendando a arquitetura com Flux (<https://www.alura.com.br/artigos/redux-desvendando-arquitetura-flux>)
- **YouTube** Alura: Como funciona o Redux? (<https://www.youtube.com/watch?v=DbEpBeZ6yic>)
- **Curso** Curso React: gerenciamento de estados globais com Redux (<https://cursos.alura.com.br/course/react-gerenciamento-estados-globais-redux>)
- **Curso** Formação Gerencie estados em React com Redux (<https://cursos.alura.com.br/formacao-react-gestao-estados>)

- **Podcast** Hipsters.tech: Guia do iniciante em React 209 (<https://www.alura.com.br/podcast/guia-do-iniciante-em-react-hipsters-209-a341>)

NextJS - Fundamentos:

- Construir interfaces Web
- Diminuir o tempo de carregamento das páginas
- Renderizar páginas no lado do servidor
- Melhorar a performance em React
- Construir rotas de API com funções serveless
- CSS-in-JS

Conteúdos

- **Site** Documentação: NextJS (inglês) (<https://nextjs.org/>)
- **Artigo** Introdução ao Next.js (<https://prensa.li/api-playbook/introducao-ao-nextjs/>)
- **Artigo** Next.js: O que é e porque utilizar este framework (<https://medium.com/codelblog/next-js-o-que-%C3%A9-e-porque-utilizar-este-framework-6180bad09133>)

Conteúdos Alura:**

- **Artigo** NextJS: aplicações fullstack com super poderes (<https://www.alura.com.br/artigos/next-js>)
- **Artigo** NextJS: por que usar? (<https://www.alura.com.br/artigos/next-js-vantagens>)
- **Artigo** Navegação com Next.js utilizando rotas dinâmicas (<https://www.alura.com.br/artigos/navegacao-next-js-utilizando-rotas-dinamicas>)
- **Artigo** Roteamento eficiente com Next.js: descobrindo o App Router (<https://www.alura.com.br/artigos/roteamento-eficiente-next-js-app-router>)
- **Artigo** Next: Server Actions aprendendo na prática (<https://www.alura.com.br/artigos/next-server-actions-aprendendo-pratica>)
- **Artigo** CSS com NextJS (<https://www.alura.com.br/artigos/css-com-next-js>)
- **Artigo** Como o Next.js ajuda no SEO das SPAs? (<https://www.alura.com.br/artigos/como-next-js-ajuda-no-seo-spas>)
- **Artigo** Next.js: criando aplicações fullstack (<https://www.alura.com.br/artigos/aplicacoes-fullstack>)
- **Artigo** Como proteger suas aplicações Next.js: práticas essenciais de segurança (<https://www.alura.com.br/artigos/nextjs-seguranca>)
- **Artigo** Autenticação moderna com OAuth 2.0 e OpenID Connect: dicas para desenvolvedores frontend (<https://www.alura.com.br/artigos/oauth-2-e-openid-connect>)
- **Artigo** Segurança do Next.js 14+ com Middlewares (<https://www.alura.com.br/artigos/next-js-middlewares>)
- **Podcast** Hipsters.tech: Nextjs - Hipsters 220 (<https://www.alura.com.br/podcast/nextjs-hipsters-ponto-tech-220-a652>)
- **Podcast** Next.js: Visão Atual - Hipsters 399 (<https://www.hipsters.tech/next-js-visao-atual-hipsters-ponto-tech-399/>)
- **YouTube** Alura: Primeiros passos Next.js (<https://www.youtube.com/watch?v=slmtdlWNwcE>)
- **YouTube** Alura: NextJs é o melhor framework (com Mario Souto) (https://www.youtube.com/watch?v=7x4_HNC4Uu4)

- **YouTube** Conhecendo o Next.js - Hipsters.Talks 24 (https://www.youtube.com/watch?v=8uoRx4_6mOE)
- **Curso** Formação Next.JS (<https://www.alura.com.br/formacao-next-js>)
- **Curso** Next.js: crie estratégias de componentes Server-Side (<https://cursos.alura.com.br/course/next-js-estrategias-componentes-server-side>)
- **Curso** Next.js 14: desenvolvendo aplicações robustas com alta produtividade (<https://cursos.alura.com.br/formacao-next-js-14-aplicacoes-robustas-alta-productividade>)
- **Curso** Vercel IA SDK: construindo um Chatbot com Next.js (<https://cursos.alura.com.br/course/vercel-ia-sdk-construindo-chatbot-next-js>)
- **Curso** Formação Next.js: autenticação e gestão de sessão (<https://www.alura.com.br/formacao-next-js-autenticacao-gestao-sessao>)

Habilidade Auxiliar: Infraestrutura e Back-end

Git e GitHub - Fundamentos:

- Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.
- GitHub é um serviço de hospedagem para desenvolvimento de software e controle de versão usando Git.
- Criar um repositório
- Clonar um repositório
- Fazer commit, push e pull de e para o repositório
- Reverter um commit
- Criar branches e pull requests
- Lidar com merge e conflitos

Conteúdos

- **Site** GitHub Documentação (<https://docs.github.com/pt>)
- **Site** GitHub Pages Documentação (<https://docs.github.com/pt/pages/getting-started-with-github-pages/about-github-pages>)
- **Site** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)
- **Site** Dangit, Git!?! (<https://dangitgit.com/>)
- **YouTube** Rafaella Ballerini: O que é Git e GitHub? - definição e conceitos importantes 1/2 (<https://www.youtube.com/watch?v=DqTITcMq68k>)
- **YouTube** Rafaella Ballerini: Como usar Git e GitHub na prática! - desde o primeiro commit até o pull request! 2/2 (<https://www.youtube.com/watch?v=UBAX-13g8OM>)
- **YouTube** Mario Souto - Dev Soutinho: Git: Entendendo de vez como funciona do melhor e mais visual jeito possível (<https://www.youtube.com/watch?v=4-tfJ-ZyA0Q>)
- **YouTube** Mario Souto - Dev Soutinho: Como colocar seu projeto no ar DE GRAÇA via GitHub! | Hospedagem com GitHub Pages (https://www.youtube.com/watch?v=BU-w2_Aae54)
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git (https://youtu.be/MqoqPzjQyCY?si=Vl_mT8EowuemptmU)
- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub (https://youtu.be/v9ZM2PVzctM?si=49ah_HRPlwpSEX4A)

Conteúdos Alura:**

- **Artigo** Git e Github: O que são, Como Configurar e Primeiros Passos (<https://www.alura.com.br/artigos/o-que-e-git-github>)
- **Artigo** Mais git com o hub: a linha de comando do Github (<https://www.alura.com.br/artigos/github-na-linha-de-comando>)
- **Podcast** Hipsters 184: Guia do Iniciante em Github (<https://cursos.alura.com.br/extra/hipsterstech/guia-do-iniciante-em-github-hipsters-184-a378>)
- **Site** GitHub: diferentes maneiras de compartilhar seu projeto (<https://cursos.alura.com.br/extra/alura-mais/github-diferentes-maneiras-de-compartilhar-seu-projeto-c2002>)
- **Site** Websérie: Git e Github para Sobrevivência (<https://www.alura.com.br/webseries/git-e-github-para-sobrevivencia>)
- **Podcast** Hipsters 109: Git e Github (<https://www.alura.com.br/podcast/hipsterstech-git-e-github-hipsters-109-a474>)
- **YouTube** Alura: Git e Github para Sobrevivência 01: Como o Git funciona? (<https://www.youtube.com/watch?v=BAmvmaKQkIQ>)
- **Curso** Curso Git e GitHub: compartilhando e colaborando em projetos (<https://cursos.alura.com.br/course/git-github-compartilhando-colaborando-projetos>)
- **Curso** Curso Git e GitHub: dominando controle de versão de código (<https://cursos.alura.com.br/course/git-github-dominando-controle-versao-codigo>)
- **Desafio** 7 Days of Code: GitHub (<https://7daysofcode.io/matricula/github>)

HTTP - Fundamentos:

- HTTP significa Hyper Text Transfer Protocol. A comunicação entre computadores cliente e servidores web é feita enviando solicitações HTTP e recebendo respostas HTTP.
- Entender a diferença dos verbos HTTP
- Testar os requests e ver os status codes no navegador
- Saber fazer uma requisição HTTP na linha de comando com WGET
- Baixar uma imagem com WGET
- Fazer um post

Conteúdos

- **Site** MDN Web Docs: Uma visão geral do HTTP (<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>)
- **Site** MDN Web Docs: Métodos de requisição HTTP (<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>)
- **Site** HTTP Cats (<https://http.cat/>)
- **Site** HTTP Dogs (<https://http.dog/>)
- **YouTube** Fabiano Gabardo Lemos: Requisições HTTP - GET, POST, PUT, PATCH DELETE (<https://www.youtube.com/watch?v=kncOJZrnkTg>)
- **YouTube** Programador a Bordo: Protocolo HTTP e TCP/IP - Introdução (<https://www.youtube.com/watch?v=V4XZ81vRGtM>)

Conteúdos Alura:**

- **Artigo** HTTP: Desmistificando o protocolo da Web (<https://www.alura.com.br/artigos/desmistificando-o-protocolo-http-parte-1>)

- **Artigo** Métodos de requisição do HTTP (<https://www.alura.com.br/artigos/metodos-de-requisicao-do-http>)
- **Artigo** Qual é a diferença entre HTTP e HTTPS? (<https://www.alura.com.br/artigos/qual-e-diferenca-entre-http-e-https>)
- **Podcast** Hipsters.tech: HTTP/2: magia com o novo protocolo - Hipsters 13 (<https://www.alura.com.br/podcast/http-2-magia-com-o-novo-protocolo-hipsters-13-a573>)
- **Curso** Curso HTTP: Entendendo a web por baixo dos panos (<https://www.alura.com.br/curso-online-http-entendendo-web-por-baixo-dos-panos>)

JSON:

- JSON significa JavaScript Object Notation (notação de objeto JavaScript). É um formato de texto para armazenar e transmitir dados.
- Criar um objeto
- Transformar um objeto em uma string
- Transformar uma string em objeto
- Manipular um objeto

Conteúdos

- **Site** MDN Web Docs: JSON (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- **Site** MDN Web Docs: Trabalhando com JSON (<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>)
- **Artigo** Introdução ao JSON (<https://juliocarneiro.medium.com/introdu%C3%A7%C3%A3o-ao-json-7825b1a550ff>)
- **YouTube** Marco Bruno: O que é JSON e como criar um objeto (<https://www.youtube.com/watch?v=oCY5YEEjIwE>)

Conteúdos Alura:**

- **Artigo** O que é JSON? (<https://www.alura.com.br/artigos/o-que-e-json>)
- **Curso** Curso MySQL e JSON: persistindo JSON de maneira eficiente (<https://www.alura.com.br/curso-online-mysql-json-persistencia>)

Linha de comando - Fundamentos:

- CLI é um programa de linha de comando que aceita entradas de texto para executar funções do sistema operacional.
- Conhecer os principais comandos

Conteúdos

- **Site** Microsoft Docs: Como usar argumentos de linha de comando para terminal do Windows (<https://docs.microsoft.com/pt-br/windows/terminal/command-line-arguments?tabs=windows>)
- **YouTube** Estevan Maito: Linha de comando básica - 8 comandos mais frequentes - WINDOWS e LINUX (<https://www.youtube.com/watch?v=rKBqXJZocYk>)
- **YouTube** Ninja do Linux: Comandos básicos da linha de comando do Linux (https://www.youtube.com/watch?v=rs_yshFGu8E)

Conteúdos Alura:**

- **Artigo** CMD: dicas para trabalhar no prompt do Windows (<https://www.alura.com.br/artigos/cmd-dicas-para-trabalhar-no-prompt-do-windows>)

- **Artigo** Como configurar variáveis de ambiente no Windows, Linux e macOS (<https://www.alura.com.br/artigos/configurar-variaveis-ambiente-windows-linux-macos>)
- **Site** Primeiras aulas do Curso Windows Prompt: Trabalhando na linha de comando (<https://www.alura.com.br/conteudo/windows-prompt-utilizando-cmd>)
- **Curso** Curso Windows Prompt: Trabalhando na linha de comando (<https://www.alura.com.br/curso-online-prompt>)
- **Curso** Curso Terminal: aprenda comandos para executar tarefas (<https://cursos.alura.com.br/course/terminal-comandos-executar-tarefas>)

Cloud - Fundamentos:

- Cloud, ou computação em nuvem é a distribuição de serviços de computação pela Internet usando um modelo de preço pago conforme o uso. Uma nuvem é composta de vários recursos de computação, que abrangem desde os próprios computadores (ou instâncias, na terminologia de nuvem) até redes, armazenamento, bancos de dados e o que estiver em torno deles. Ou seja, tudo o que normalmente é necessário para montar o equivalente a uma sala de servidores, ou mesmo um data center completo, estará pronto para ser utilizado, configurado e executado.
- Conhecer a diferença entre IaaS, PaaS e SaaS
- Conhecer os maiores provedores de cloud
- Especializar-se em algum provedor

Conteúdos

- **Artigo** Computação em nuvem (<https://medium.com/sysadminas/computa%C3%A7%C3%A3o-em-nuvem-515930304cf9>)
- **Artigo** O que é cloud? (<https://gabriel-faraday.medium.com/o-que-%C3%A9-cloud-991109e708c6>)
- **YouTube** Gabs Ferreira: Por que investir e estudar cloud? (<https://www.youtube.com/watch?v=Z45BTNeZ1l0>)
- **YouTube** Andre Iacono: O que é MICROSOFT AZURE? Qual Certificação começar em 2022? (<https://www.youtube.com/watch?v=f-oVzkvMwnE>)
- **YouTube** AWS: O que é a AWS? (<https://www.youtube.com/watch?v=8Jl9wQ8sUdQ>)
- **YouTube** O que é Google Cloud e por que aprender? (<https://www.youtube.com/shorts/Lzq3f1DHWcl>)
- **Artigo** AWS vs Google Cloud vs Azure: o que cada um tem de melhor? (<https://medium.com/data-hackers/aws-vs-google-cloud-vs-azure-o-que-cada-um-tem-de-melhor-52107174f7b7>)
- **YouTube** Código Fonte TV: Azure (A plataforma Cloud da Microsoft) (<https://www.youtube.com/watch?v=YgE-sZaCuJ0>)
- **YouTube** Mundo da Cloud: AWS do Zero ao Expert (<https://www.youtube.com/watch?v=HiBCv9DolxI&list=PLtL97Owd1gkQ0dfqGW8OtJ-155Gs67EcZ>)

Conteúdos Alura:**

- **Podcast** Hipsters.tech: TechGuide - Fundamentos Cloud – Hipsters Ponto Tech #348 (<https://www.hipsters.tech/techguide-fundamentos-cloud-hipsters-ponto-tech-348/>)
- **Artigo** Cloud: o que é, História e Guia da computação em nuvem (<https://www.alura.com.br/artigos/cloud>)
- **Podcast** Hipsters.tech: Uma jornada Para o Cloud – Hipsters Deep Dive 005 (<https://www.alura.com.br/podcast/uma-jornada-para-o-cloud-hipsters-deep-dive-005-a1100>)

- **Podcast** Hipsters.tech: Histórias do Cloud - Hipsters 04 (<https://www.alura.com.br/podcast/historias-do-cloud-hipsters-04-a582>)
- **Artigo** Heroku, Vercel e outras opções de cloud como plataforma (<https://www.alura.com.br/artigos/heroku-vercel-outras-opcoes-cloud-plataforma>)
- **Artigo** AWS: Guia sobre o que é Amazon Web Services, seus Serviços e Certificações (<https://www.alura.com.br/artigos/aws>)
- **Artigo** Terraform: criando máquinas na Azure (<https://www.alura.com.br/artigos/terraform-maquinas-na-azure>)
- **YouTube** Alura: O que é cloud? (<https://www.YOUTUBE.com/watch?v=wev9fMrg-TU>)
- **YouTube** Alura: AWS, Google Cloud e Azure: Por onde começar? | Hipsters.Talks (<https://www.YOUTUBE.com/watch?v=z9k6rsdmWc0&t=300s>)
- **YouTube** Alura: Certificação em Cloud: Azure, AWS, Google | Hipsters.Talks (https://www.YOUTUBE.com/watch?v=W4K82n_WK5g&t=290s)
- **Curso** Formação Começando em Cloud Computing (<https://cursos.alura.com.br/formacao-cloud-computing>)
- **Curso** Formação Amazon Web Services (<https://cursos.alura.com.br/formacao-amazon-web-services>)
- **Curso** Formação Google Certified Associate Cloud Engineer (<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Certificação AWS Certified Cloud Practitioner (<https://cursos.alura.com.br/formacao-aws-certified-cloud-practitioner>)
- **Curso** Formação Containers com AWS ECS e EKS (<https://cursos.alura.com.br/formacao-containers-aws>)
- **Curso** Formação Google Cloud Platform (<https://cursos.alura.com.br/formacao-google-cloud>)
- **Curso** Formação Certificação Google Certified Associate Cloud Engineer (<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Azure (<https://cursos.alura.com.br/formacao-conhecendo-azure>)
- **Curso** Formação Certificação AZ-900: Microsoft Azure Fundamentals (<https://cursos.alura.com.br/formacao-certificacao-az-900-microsoft-azure-fundamentals>)

YARN:

- Yarn é um gerenciador de pacotes para seu código. Ele permite que você use e compartilhe código com outros desenvolvedores. O código é compartilhado por meio de algo chamado pacote (às vezes chamado de módulo). Um pacote contém todo o código que está sendo compartilhado, bem como um arquivo package.json que descreve o pacote.
- Gerenciar pacotes
- Gerenciar dependências
- Instalação de pacotes offline
- Comandos
- Arquivo yarn.lock

Conteúdos

- **Site** YARN: Documentação (inglês) (<https://classic.yarnpkg.com/en/docs>)
- **Artigo** Porque você deveria dar uma chance ao Yarn? (<https://rafaell-lycan.com/2017/porque-voce-deveria-dar-uma-chance-yarn/>)

- **YouTube** Lucas Santos: Yarn VS NPM, qual é melhor? (<https://www.youtube.com/watch?v=vxES6rbrd-U>)

Conteúdos Alura:**

- **Artigo** NPM vs Yarn (<https://www.alura.com.br/artigos/npm-vs-yarn>)

Habilidade Auxiliar: UX e Design

Design System:

- Um Design System (sistema de design) é uma coleção de componentes reutilizáveis, guiados por padrões claros, que podem ser colocados juntos para construir aplicações.
- Criar e manter bibliotecas que serão consumidas e usadas como padrão para a construção de um projeto
- Design tokens
- Estilos fundamentais
- Construção de componentes
- Microinterações
- Documentação

Conteúdos

- **Artigo** Design System: o que é e quais os benefícios? (<https://medium.com/ipnet-growth-partner/design-system-o-que-e-438773dd811>)
- **Artigo** Afinal, o que é Design System? (<https://brasil.uxdesign.cc/afinal-o-que-%C3%A9-design-system-448c257b0021>)
- **Artigo** O que são Design Tokens (<https://medium.com/pretux/design-tokens-112b2ee11ddf>)
- **YouTube** Caio Gonzalez: O que é Design System? / Guia para começar o seu próprio Design System (<https://www.youtube.com/watch?v=ajgbXpFVaAw>)

Conteúdos Alura:**

- **Artigo** O que é Design System? (<https://www.alura.com.br/artigos/o-que-e-design-system>)
- **Artigo** Design Systems: exemplos práticos (<https://www.alura.com.br/artigos/design-systems-exemplos-praticos>)
- **YouTube** Alura: Design System & Style Guide (<https://www.youtube.com/watch?v=rSLvpOqC5wQ>)
- **YouTube** Alura: Design Systems (com Charles Assunção) (<https://www.youtube.com/watch?v=MLGFctEtmYQ>)
- **Podcast** Hipsters.tech: Design Systems - Hipsters 170 (<https://www.alura.com.br/podcast/hipsterstech-design-systems-hipsters-170-a399>)
- **Podcast** Layers.tech: Design System – Layers ponto tech 36 (<https://www.alura.com.br/podcast/layerstech-design-system-layers-ponto-tech-36-a1056>)
- **Curso** Design System: definindo estilos e tokens (<https://cursos.alura.com.br/course/design-system-definindo-estilos-tokens>)
- **Curso** Design System: projetando e construindo componentes (<https://cursos.alura.com.br/course/design-system-projetando-construindo-componentes>)
- **Curso** Design System: documentando um design system (<https://cursos.alura.com.br/course/design-system-documentando-design-system>)
- **Desafio** Experimente o que é ser um(a) profissional de Ux (<https://www.alura.com.br/challenges/ux-4>)

Figma - Fundamentos:

- Figma é uma aplicação web colaborativa para design de interfaces. O conjunto de recursos do Figma se concentra na interface do usuário e no design da experiência do usuário, com ênfase na colaboração em tempo real, utilizando uma variedade de editores de gráficos vetoriais e ferramentas de prototipagem.
- Criar layouts de páginas e componentes

Conteúdos

- **Site** Figma Básico - Primeiros passos com Figma (<https://www.figma.com/community/file/1190593276621265988>)
- **Artigo** O que é o Figma e por que usar ele? (<https://medium.com/nerdzaio/o-que-%C3%A9-o-figma-e-por-que-usar-ele-a71fbf1dbdd8>)
- **Artigo** Entenda o que é e como usar o Figma para criar designs (<https://blog.b2bstack.com.br/figma/>)
- **YouTube** Marco Bruno: Como usar o Figma! (<https://www.youtube.com/watch?v=qoE-2YFeW-Q>)
- **YouTube** 4 Exemplos de Componentes interativos no Figma (<https://www.youtube.com/watch?v=mGL3jrY-OBc>)

Conteúdos Alura:**

- **Artigo** Figma: o que é a ferramenta, Design e uso (<https://www.alura.com.br/artigos/figma>)
- **Site** Como Front-End utiliza o Figma - Alura+ (<https://cursos.alura.com.br/extra/alura-mais/como-front-end-utiliza-o-figma-c858>)
- **Artigo** Top 5 plugins no Figma para trabalhar com Design System (<https://www.alura.com.br/artigos/top-5-plugins-figma-trabalhar-com-design-system>)
- **Artigo** 5 plugins essenciais que você precisa ter no seu Figma (<https://www.alura.com.br/artigos/5-plugins-essenciais-do-figma>)
- **Artigo** 10 truques incríveis e pouco conhecidos no Figma (<https://www.alura.com.br/artigos/10-truques-incriveis-pouco-conhecidos-figma>)
- **Podcast** Layers.tech: Figma do Design ao Código 03 (<https://www.alura.com.br/podcast/layerstech-figma-do-design-ao-codigo-layers-ponto-tech-03-a726>)
- **YouTube** Como fazer estilos locais mais rapidamente no Figma? (https://www.youtube.com/watch?v=fCHd2_IaCQQ)
- **Curso** Formação Figma (<https://www.alura.com.br/formacao-figma>)

Componentes de design:

- Conhecer os componentes descrevem um layout ou interface

Conteúdos

- **Artigo** Componentes Principais UI (<https://medium.com/@prikuer/componentes-principais-ui-8b538a3aaa5>)
- **Artigo** Compartilhando CSS em projetos do Nx (<https://www.alura.com.br/artigos/compartilhando-css-projetos-nx>)
- **YouTube** danielmacedoco: UI Design e Design System - Visão geral e prática sobre alguns componentes (<https://www.youtube.com/watch?v=VpgQNeSltpA>)
- **YouTube** 4 Exemplos de Componentes interativos no Figma (<https://www.youtube.com/watch?v=mGL3jrY-OBc>)

Conteúdos Alura:**

- **Curso** Curso Design System: criando componentes e documentando (<https://www.alura.com.br/curso-online-design-system-componentes-documentando>)
- **Curso** Curso Figma: conhecendo componentes da interface (<https://www.alura.com.br/curso-online-figma-conhecendo-componentes-interface>)

Sistemas de cores:

- Definir uma paleta de cores que faça sentido para determinada interface

Conteúdos

- **Artigo** Cores em UI: Um Guia Rápido Para Usar em Seus Projetos (<https://medium.com/aela/cores-em-ui-um-guia-r%C3%A1pido-para-usar-em-seus-projetos-31ccffe3e16b>)
- **Artigo** A Psicologia das cores e sua relação com o UX Design (<https://brasil.uxdesign.cc/a-psicologia-das-cores-e-sua-rela%C3%A7%C3%A3o-com-o-ux-design-af02460639cd>)
- **YouTube** kacio.design: Aplicação de Cores - Princípios do UI (https://www.youtube.com/watch?v=C_VhzEygT6U)

Conteúdos Alura:**

- **Curso** Curso Cores para Designers: escolhendo e trabalhando com cores em um projeto (<https://www.alura.com.br/curso-online-cores-para-seu-projeto>)
- **Site** Primeiras aulas do curso Cores: sistemas básicos e paletas (<https://www.alura.com.br/conteudo/fundamentos-da-cor>)
- **Curso** Cores: sistemas básicos e paletas (<https://www.alura.com.br/curso-online-fundamentos-da-cor>)
- **Curso** Curso Design editorial: criação de materiais gráficos (<https://www.alura.com.br/curso-online-design-editorial>)

Como usar fontes:

- Escolher a fonte mais adequada para determinado projeto

Conteúdos

- **Artigo** Você sabe usar tipografia em UI Design? (<https://medium.com/ui-lab-school/voc%C3%AA-sabe-usar-tipografia-em-ui-design-9ce4ccdbab43>)
- **Artigo** As 20 fontes mais usadas pelos designers gráficos (<https://apenasumchico.medium.com/as-20-fontes-mais-populares-de-todos-os-tempos-38a6c121dfb>)
- **Artigo** Usando ícones SVG como fontes com o IcoMoon (<https://dev.to/sucodelarangela/usando-icone-sv-g-como-fontes-com-o-icomoon-563e>)
- **YouTube** Nadine Fronza: Como escolher a melhor fonte para seu projeto (<https://www.youtube.com/watch?v=MSKPUtldadI>)
- **YouTube** Thiago Abreu: Como Escolher Fontes para um Trabalho de Design (<https://www.youtube.com/watch?v=I-TrNbZi-aU>)

Conteúdos Alura:**

- **Artigo** 10 tipografias gratuitas para um design incrível (<https://www.alura.com.br/artigos/10-tipografias-gratuitas-design-incrivel>)
- **Site** Tipografia para Web (<https://tipografiaparaweb.netlify.app/>)

- **Site** Primeiras aulas do curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/conteudo/tipografia-conceito>)
- **Curso** Curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/curso-online-tipografia-conceito>)
- **Curso** Curso Design editorial: criação de materiais gráficos (<https://www.alura.com.br/curso-online-design-editorial>)

TechGuide - Alura
Alura, PM3 e FIAP
O Techguide.sh é um projeto open source