

React Native

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

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

Conteúdos Alura:

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

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=oCY5YEEjlwE>)
- **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 Native - Fundamentos:

- React Native é um framework de desenvolvimento de software de código aberto criado pelo Facebook. Ele permite desenvolver aplicativos para Android e iOS usando um único código-fonte.
- No React Native, as interfaces são construídas usando componentes. Esses componentes, como View, Text, Image, Button e TextInput, são responsáveis por definir tanto a aparência quanto o comportamento da sua aplicação. A ideia é simplesmente descrever como a interface deve se comportar de acordo com o estado da aplicação.

Conteúdos

- **Site** React Native - Documentação (<https://reactnative.dev/>)
- **Site** Expo - Documentação (<https://expo.dev/>)
- **Site** React Native - Guia de Introdução (<https://reactnative.dev/docs/getting-started>)
- **Site** React Navigation - Navegação no React Native (<https://reactnavigation.org/>)
- **Site** React Native Elements - Biblioteca de Componentes (<https://reactnativeelements.com/>)
- **Site** React Native Debugger - Ferramenta de Debug (<https://github.com/jhen0409/react-native-debugger>)
- **Site** React Native - Uso de Hooks (<https://reactnative.dev/docs/hooks>)

Conteúdos Alura:

- **Curso** Formação React Native: Desenvolva seu primeiro app (<https://cursos.alura.com.br/formacao-react-native-primeiro-app>)

React Native - Pacotes Essenciais:

- O React Native permite o uso de pacotes e bibliotecas de terceiros para agilizar o desenvolvimento e adicionar funcionalidades extras aos aplicativos.
- Conhecer e saber utilizar o NPM (Node Package Manager) para gerenciar pacotes.
- Conhecer os principais pacotes do React Native e suas utilidades.

Conteúdos

- **Site** Documentação Oficial do React Native - Usando Pacotes (<https://reactnative.dev/docs/linking-libraries>)
- **Site** React Native - Pacotes no NPM (<https://www.npmjs.com/package/react-native>)
- **Site** React Native - Pacotes Populares (<https://github.com/ReactNativeCommunity>)
- **Site** React Native Firebase - Biblioteca para Firebase (<https://rnfirebase.io/>)
- **Site** React Navigation - Navegação no React Native (<https://reactnavigation.org/>)
- **Site** React Native Paper - Componentes UI (<https://callstack.github.io/react-native-paper/>)
- **Site** Axios - Biblioteca para requisições HTTP (<https://axios-http.com/>)

- **YouTube** React Native: 10 Essential Libraries for Developers (<https://youtu.be/2EJkEzV0FJ8>)
- **Artigo** Bibliotecas essenciais para o desenvolvimento com React Native (<https://blog.codemagic.io/essential-libraries-for-react-native-developers/>)
- **Artigo** Principais Pacotes para React Native (<https://medium.com/@newtonstechblog/most-popular-react-native-libraries-9cc62b8f5e4b>)

Conteúdos Alura:

- **Curso** Formação React Native: Desenvolva seu primeiro app (<https://cursos.alura.com.br/formacao-react-native-primeiro-app>)

React Native - Testes:

- Aprender as melhores práticas de testes em React Native.
- Conhecer as ferramentas e bibliotecas de testes para React Native, como Jest e React Native Testing Library.
- Aprender a escrever testes unitários, de integração e de interface de usuário para garantir a qualidade do código.

Conteúdos

- **Site** Documentação Oficial do React Native - Testes (<https://reactnative.dev/docs/testing-overview>)
- **Site** Jest - Framework de Testes para JavaScript (<https://jestjs.io/>)
- **Site** React Native Testing Library - Testes de UI (<https://testing-library.com/docs/react-native-testing-library/intro/>)
- **Site** Enzyme - Testes para React Native (<https://airbnb.io/enzyme/>)
- **Site** Mocking em Jest - Documentação (<https://jestjs.io/docs/mock-functions>)
- **Site** Detox - Testes End-to-End para React Native (<https://wix.github.io/Detox/>)
- **YouTube** React Native Testing Tutorial | Jest and React Native Testing Library (<https://youtu.be/RX61r8u1fMc>)
- **Artigo** Testando Aplicações React Native com Jest (<https://www.toptal.com/react-native/react-native-testing-jest>)
- **Artigo** Testes End-to-End com Detox no React Native (<https://www.smashingmagazine.com/2020/04/detox-end-to-end-testing-react-native/>)

Nível 3

React Native - Nativo:

- Conhecer as especificidades do desenvolvimento nativo para Android.
- Conhecer as especificidades do desenvolvimento nativo para iOS

Conteúdos

- **Site** Documentação Oficial do React Native - Componentes e APIs (<https://reactnative.dev/docs/components-and-apis>)
- **Site** Documentação Oficial do React Native - Integração com Android (<https://reactnative.dev/docs/integration-with-existing-apps#android>)
- **Site** Documentação Oficial do React Native - Integração com iOS (<https://reactnative.dev/docs/integration-with-existing-apps#ios>)

- **Site** Documentação Oficial do React Native - Trabalhando com Native Modules (<https://reactnative.dev/docs/native-modules-android>)
- **Site** Documentação Oficial do Expo - Começando com o Expo (<https://docs.expo.dev/get-started/installation/>)
- **Site** Documentação Oficial do Expo - API do Expo (<https://docs.expo.dev/versions/latest/>)
- **Site** Documentação Oficial do Expo - Desenvolvimento para Android e iOS (<https://docs.expo.dev/get-started/installation/#android-and-ios>)
- **Site** Expo Managed Workflow - O que é e como funciona (<https://docs.expo.dev/workflow/managed/>)

Conteúdos Alura:

- **Curso** Curso React Native: integrando câmera nativa para capturar fotos (<https://cursos.alura.com.br/course/react-native-integrando-camera-nativa-capturar-fotos>)

React Native - Animações:

- Entender os conceitos básicos de animação no React Native.
- Aprender a utilizar as bibliotecas de animação mais comuns no React Native.
- Integrar animações em componentes e fluxos de navegação.

Conteúdos

- **Site** Documentação Oficial do React Native - Animações (<https://reactnative.dev/docs/animations>)
- **Site** React Native Animatable - Biblioteca de Animações (<https://github.com/oblador/react-native-animatable>)
- **Site** Lottie for React Native - Animações vetoriais (<https://docs.lottiefiles.com/guide/react-native>)

Conteúdos Alura:

- **Curso** ALURA+: Como utilizar ícones e animações em apps com React Native (<https://cursos.alura.com.br/extra/alura-mais/como-utilizar-icone-e-animacoes-em-apps-com-react-native-c1227>)

React Native - Arquitetura:

- Compreender as melhores práticas de arquitetura para criar aplicativos React Native.
- Conhecer os principais padrões de arquitetura no React Native.
- Aprender como organizar pastas e componentes.

Conteúdos

- **Site** Documentação oficial do React Native - Arquitetura (<https://reactnative.dev/docs/architecture-overview>)
- **Site** Arquitetura 0.76 do React Native - Novidades e Melhoria (<https://reactnative.dev/blog/2020/09/23/version-0.76>)
- **Site** React Navigation - Arquitetura de navegação no React Native (<https://reactnavigation.org/docs/getting-started>)
- **Site** React Native - Organizando componentes e pastas (<https://reactnative.dev/docs/structure>)
- **Site** Padrão MVVM no React Native (<https://reactnative.dev/docs/architecture#mvvm-pattern>)
- **Site** React Native - Melhorias de performance com arquitetura (<https://reactnative.dev/docs/optimizing-performance>)

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-ZyAOQ>)
- **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=VI_mT8EowuemptmU)
- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub (https://youtu.be/v9ZM2PVzctM?si=49ah_HRP1wpSEX4A)

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=BAvmakQkIQ>)
- **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>)

Design Patterns:

- Na engenharia de software, um "padrão de projeto" (Design Pattern em inglês) é uma solução geral e reutilizável para um problema que ocorre normalmente dentro de um determinado contexto de projeto de software.
- Conhecer e aplicar os principais Design Patterns

Conteúdos

- **Artigo** Design Patterns — O que são e quais os benefícios? (<https://djesusnet.medium.com/design-patterns-gof-o-que-s%C3%A3o-e-quais-os-benef%C3%ADcios-9cd0cfd6ebf>)
- **Artigo** Design Patterns (<https://medium.com/xp-inc/design-patterns-727494af001d>)
- **YouTube** Matheus Castiglioni: O que são design patterns (padrões de projetos) e para que servem? (<https://www.YOUTUBE.com/watch?v=KNWd1aWtsMw>)
- **YouTube** RinaldoDev: Design Patterns - o que são? Por que aprender a utilizar? (<https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Design Patterns – Hipsters Ponto Tech #347 (<https://www.hipsters.tech/techguide-design-patterns-hipsters-ponto-tech-347/>)
- **Artigo** Design patterns: Breve introdução aos padrões de projeto (<https://www.alura.com.br/artigos/design-patterns-introducao-padroes-projeto>)
- **Podcast** Hipsters.tech: Design Patterns - Hipsters 206 (<https://www.alura.com.br/podcast/design-patterns-hipsters-206-a345>)
- **Curso** Curso Design Patterns em Java I: boas práticas de programação (<https://www.alura.com.br/curso-online-introducao-design-patterns-java>)
- **Curso** Curso Design Patterns C# I: boas práticas de programação (<https://www.alura.com.br/curso-online-design-patterns-dotnet>)
- **Curso** Curso Design Patterns em PHP: padrões comportamental (<https://www.alura.com.br/curso-online-php-design-pattern-comportamental>)

- **Curso** Curso Design Patterns em PHP: padrões estruturais (<https://www.alura.com.br/curso-online-php-design-pattern-estrutural>)
- **Curso** Curso Design Patterns em PHP: padrões criacionais (<https://www.alura.com.br/curso-online-php-design-pattern-criacional>)

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=Z45BTNeZ1lQ>)

- **YouTube** Andre Iacono: O que é MICROSOFT AZURE? Qual Certificação começar em 2022? (<https://www.youtube.com/watch?v=f-oVzkyMwnE>)
- **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>)

SOLID:

- O Solid possui cinco princípios considerados como boas práticas no desenvolvimento de software que ajudam os programadores a escrever os códigos mais limpos, separando as responsabilidades, diminuindo acoplamentos, facilitando na refatoração e estimulando o reaproveitamento do código.

Conteúdos

- **Artigo** Princípios de S.O.L.I.D em C# — Guia prático (<https://medium.com/beelabacademy/princ%C3%ADpios-de-s-o-l-i-d-em-c-guia-pr%C3%A1tico-cbb1e6584284>)
- **YouTube** Filipe Deschamps: SOLID fica fácil com Essas Ilustrações (<https://www.youtube.com/watch?v=6SfrO3D4dHM>)
- **YouTube** Central dotNET: Princípios de SOLID com C# (<https://www.youtube.com/watch?v=iU4BMUcjg8g>)
- **YouTube** Felipe Pinheiro - Tech: Princípios SOLID, o que é DIP? (<https://www.youtube.com/watch?v=q0BGgQJcp7w>)
- **YouTube** Dev Eficiente: SOLID - Uma reflexão sobre o Princípio da responsabilidade única (https://www.youtube.com/watch?v=GGe0o_v5vjM)

Conteúdos Alura:

- **Artigo** SOLID: o que é e quais os 5 princípios da Programação Orientada a Objetos (POO) (<https://www.alura.com.br/artigos/solid>)
- **Podcast** SOLID: Código bom e bonito - Hipsters Ponto Tech 219 (<https://www.alura.com.br/podcast/hipsterstech-solid-codigo-bom-e-bonito-hipsters-ponto-tech-219-a649>)
- **YouTube** Alura: Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **Curso** Curso C#: aplique princípios SOLID (<https://cursos.alura.com.br/course/csharp-aplique-principios-solid>)
- **Curso** Curso Boas práticas de programação: melhore o código de uma API Java (<https://cursos.alura.com.br/course/boas-praticas-programacao-melhora-codigo-api-java>)
- **Curso** Curso SOLID com PHP: princípios da programação orientada a objetos (<https://cursos.alura.com.br/course/solid-php-principios-orientacao-a-objetos>)
- **Curso** Curso iOS: escrevendo código de qualidade com SOLID em Swift (<https://www.alura.com.br/curso-online-ios-escrevendo-codigo-qualidade-solid-swift>)
- **Curso** SOLID com TypeScript: aplicando boas práticas em orientação a objetos (<https://www.alura.com.br/curso-online-solid-typescript-boas-praticas-orientacao-objetos>)
- **Curso** Formação Boas práticas em C# (<https://cursos.alura.com.br/formacao-boas-praticas-c-sharp>)
- **Curso** Formação Boas práticas em PHP (<https://cursos.alura.com.br/formacao-boas-praticas-php>)
- **Curso** Formação Boas práticas em Java (<https://cursos.alura.com.br/formacao-boas-praticas-java>)
- **Site** Livro Casa do Código: Desbravando SOLID - Práticas avançadas para códigos de qualidade em Java moderno (<https://www.casadocodigo.com.br/products/livro-desbravando-solid>)

- **YouTube** CodandoTV(Rods): SOLID e o 'D' do Princípio da Inversão de Dependência usando a Injeção de Dependência para ajudar (<https://youtu.be/xpcCDCH5k4Y>)

Clean Architecture:

- A Clean Architecture (Arquitetura Limpa) é uma forma de desenvolver software, de tal forma que apenas olhando para o código fonte de um programa, você deve ser capaz de dizer o que o programa faz.

Conteúdos

- **Artigo** Descomplicando a Clean Architecture (<https://medium.com/luizalabs/descomplicando-a-clean-architecture-cf4dfc4a1ac6>)
- **YouTube** Como DEV ser!: Entenda CLEAN ARCHITECTURE de uma vez por todas! (<https://www.youtube.com/watch?v=HynTfTli4mw>)
- **YouTube** Full Cycle: O que é Clean Architecture (<https://www.youtube.com/watch?v=dQys-mnOtQg>)
- **YouTube** CodandoTV(Rods) - Simplificando o Clean Architecture +MVVM na sua aplicação mobile - Guia Completo (<https://youtu.be/8ehIZfyN1S0?si=l7-l5l4zsnYLSjD>)

Conteúdos Alura:

- **Site** Clean Architecture (Arquitetura Limpa) - O que é? (<https://cursos.alura.com.br/extra/alura-mais/clean-architecture-arquitetura-limpa-o-que-e--c204>)
- **Podcast** Hipsters.tech: Clean Architecture - Hipsters Ponto Tech 254 (<https://www.hipsters.tech/clean-architecture-hipsters-ponto-tech-254/>)
- **YouTube** Alura: Arquitetura de sistemas (<https://www.youtube.com/watch?v=oedWxgAZc2A>)
- **Site** Primeiras aulas do curso Java e Clean Architecture: descomplicando arquitetura de software (<https://www.alura.com.br/conteudo/java-clean-architecture>)
- **Curso** Curso Java e Clean Architecture: descomplicando arquitetura de software (<https://www.alura.com.br/curso-online-java-clean-architecture>)
- **Curso** Curso PHP e Clean Architecture: descomplicando arquitetura de software (<https://cursos.alura.com.br/course/php-introducao-clean-achitecture>)

Firebase:

- O Firebase é uma plataforma de desenvolvimento de aplicativos Backend-as-a-Service (BaaS) que fornece serviços de backend hospedados, tais como banco de dados em tempo real, armazenamento em nuvem, autenticação, relatórios de falhas, aprendizado de máquina, configuração remota e hospedagem para seus arquivos estáticos.
- Entender como Instalar o Firebase
- Conhecer a documentação do Firebase
- Conhecer as ferramentas do Firebase disponíveis

Conteúdos

- **Site** Produtos do Firebase | Firebase (<https://firebase.google.com/products-build?hl=pt-br>)
- **Site** Casos de Uso do Firebase | Firebase (<https://firebase.google.com/use-cases?hl=pt-br>)
- **Site** Adicionando o Firebase ao App Flutter | Firebase Docs (<https://firebase.google.com/docs/flutter/setup?platform=android>)
- **Site** Adicionando o Firebase ao seu projeto Android | Firebase Docs (<https://firebase.google.com/docs/android/setup>)

- **YouTube** Firebase: Começando no Firebase com Flutter - Firecasts (inglês) (https://www.youtube.com/watch?v=EXp0gg9kGxI&ab_channel=Firebase)
- **YouTube** Prof. Diego Antunes: Instalação do Firebase no Flutter em 2022 (FlutterFire e Realtime Database) (https://www.youtube.com/watch?v=OjdGSoDntZQ&ab_channel=Prof.DiegoAntunes)
- **Site** Precificação do Firebase | Firebase (<https://firebase.google.com/pricing?hl=pt-br>)

Conteúdos Alura:

- **Artigo** Integrando App Android com o Firebase Cloud Messaging (<https://www.alura.com.br/artigos/integrando-app-android-com-o-firebase-cloud-messaging>)
- **Artigo** Tratando notificações recebidas do Firebase no Android (<https://www.alura.com.br/artigos/tratando-notificacoes-recebidas-do-firebase-no-android>)
- **Artigo** Autenticando com a conta Google no Android utilizando o Firebase Authentication (<https://www.alura.com.br/artigos/autenticando-no-google-com-firebase-authentication>)
- **Artigo** Flutter: Tratamento de exceções com Firebase Crashlytics (<https://www.alura.com.br/artigos/tratamento-de-execucoes-com-firebase-crashlytics>)
- **YouTube** Alura: Primeiros passos em Cloud Firestore no Flutter (Parte 1) | #AluraMais (<https://youtu.be/6jsR5CXhHOs>)
- **YouTube** Alura: Primeiros passos em Cloud Firestore no Flutter (Parte 2) | #AluraMais (<https://youtu.be/KWWTTY6gSw4>)
- **YouTube** Alura: Registro de exceções no Firebase Crashlytics | #AluraMais (https://www.youtube.com/watch?v=LOLjsG_eNyY&ab_channel=AluraCursosOnline)
- **Curso** Curso Flutter: Push Notifications com Firebase Cloud Messaging (<https://www.alura.com.br/curso-online-flutter-push-notifications-firebase-cloud-messaging>)
- **Curso** Formação Firebase com Android (<https://www.alura.com.br/formacao-firebase-android>)

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=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>)
- **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_lAcQQ)
- **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-8b538a3aaaa5>)
- **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/acla/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_VhzEyqT6U)

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=MSKPUtIdadI>)
- **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-incrivei>)
- **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>)

Design Responsivo:

- Ajustar suas páginas para o tamanho da tela do usuário
- Media queries
- Conhecer o conceito de Mobile first

Conteúdos

- **Artigo** Design responsivo para leigos (<https://medium.com/neworder/design-responsivo-para-leigos-888c9d0dfdaf>)
- **Artigo** Entendendo as diferenças entre design responsivo, adaptativo e mobile-first (<https://medium.com/@fnandaleite/entendendo-as-diferen%C3%A7as-entre-design-responsivo-adaptativo-e-mobile-first-ea3c61fc9181>)
- **Artigo** Web Design Responsivo — O Que É e Como Usá-lo (<https://medium.com/@carlosverza/web-design-responsivo-o-que-%C3%A9-e-como-us%C3%A1-lo-12477b168fc7>)
- **YouTube** MobGeek: O que é Design Responsivo? (<https://www.youtube.com/watch?v=WXfYtneJZlI>)
- **YouTube** Ralf Lima: Mobile First e Desktop First (https://www.youtube.com/watch?v=oqggJTC3_vA)

Conteúdos Alura:

- **Artigo** Como fazer Grids e a Responsividade na Web (<https://www.alura.com.br/artigos/como-fazer-grids-e-a-responsividade-na-web>)
- **Curso** Curso Layouts Responsivos: trabalhando com layouts mobile (<https://www.alura.com.br/curso-online-mobile-first-layouts-responsivos>)
- **Curso** HTML e CSS: responsividade com mobile-first (<https://www.alura.com.br/curso-online-html-css-responsividade-mobile-first>)
- **Desafio** 7 Days of Code: Responsividade (<https://7daysofcode.io/matricula/responsividade>)

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