

Front-end

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-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=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>)

Acessibilidade em Javascript:

- Acessibilidade Digital é a eliminação de barreiras na Web. O conceito pressupõe que os sites e portais sejam projetados de modo que todas as pessoas possam perceber, entender, navegar e interagir de maneira efetiva com as páginas.
- Escrever código com acessibilidade em mente

Conteúdos

- **Site** MDN Web Docs: Acessibilidade (<https://developer.mozilla.org/pt-BR/docs/Learn/Accessibility>)
- **Site** MDN Web Docs: HTML - Boas práticas em acessibilidade (<https://developer.mozilla.org/pt-BR/docs/Learn/Accessibility/HTML>)
- **Artigo** Escrevendo JavaScript com acessibilidade em mente (<https://medium.com/emanuelg-blog/escrevendo-javascript-com-acessibilidade-em-mente-3720e9779969>)
- **Artigo** Garantindo acessibilidade com Javascript não obstrutivo (<https://klauslaube.com.br/2010/11/18/acessibilidade-javascript-nao-obstrutivo.html>)
- **YouTube** Matheus Castiglioni: Acessibilidade em páginas web (<https://www.youtube.com/watch?v=pOmdL9LzDJg>)
- **Curso** Web.dev: Aprenda acessibilidade! (<https://web.dev/learn/accessibility/welcome?hl=pt-br>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: Acessibilidade web - Hipsters 21 (<https://www.alura.com.br/podcast/hipsterstech-acessibilidade-web-hipsters-21-a565>)
- **YouTube** Alura: Acessibilidade Web | #HipstersPontoTube (https://www.youtube.com/watch?v=gnO_6NDfQHI)
- **YouTube** Alura: Acessibilidade na Web - Alura Live 47 (<https://www.youtube.com/watch?v=vrLx1yJjsXU>)
- **Curso** Formação Acessibilidade Web (<https://cursos.alura.com.br/formacao-acessibilidade-tecnologias-web-html-css-javascript-projetos-inclusivos>)

Estratégias de SEO:

- SEO significa otimização para motores de busca e que diz respeito as estratégias usadas para ranquear um site dentro de mecanismos de busca como o Yahoo, o Bing e, claro, o mais famoso de todos, o Google.

- Escolher palavras-chave
- Entender como o Google classifica páginas
- Conhecer os fatores de ranqueamento
- Fazer Link Building

Conteúdos

- **Artigo** Guia completo de SEO — Como fazer e quais são as boas práticas (<https://medium.com/@redacaogalata/guia-completo-de-seo-como-fazer-e-quais-s%C3%A3o-as-boas-pr%C3%A1ticas-529ea97fdc98>)
- **YouTube** Kenneth Corrêa: SEO - O que é SEO? (<https://www.youtube.com/watch?v=BJpf9VD7A5g>)
- **YouTube** Canal do Consultor: Introdução ao SEO: O que é? Para que serve? Diferenças entre SEO e SEM (<https://www.youtube.com/watch?v=hn0DDyCB5eg>)

Conteúdos Alura:

- **Artigo** SEO: o que é, boas práticas e ferramentas para a sua estratégia dar certo (<https://www.alura.com.br/artigos/seo-o-que-e-boas-praticas-ferramentas-estrategia>)
- **Podcast** Hipsters.tech: SEO moderno - Hipsters Ponto Tech 289 (<https://www.alura.com.br/podcast/hipsterstech-seo-moderno-hipsters-ponto-tech-289-a1394>)
- **Podcast** Layers.tech: SEO e Youtube - Layers ponto tech 38 (<https://www.alura.com.br/podcast/seo-e-youtube-layers-ponto-tech-38-a1081>)
- **YouTube** Alura: Como rankear bem no Google? (<https://www.youtube.com/watch?v=PfGyzrLGqZ8>)
- **Curso** Formação SEO (<https://www.alura.com.br/formacao-seo>)

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=WXfYtneJZzI>)
- **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>)

Nível 2

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 - Testes:

- O teste de software é o processo de avaliação e verificação de que um software realmente faz o que deveria fazer. Os benefícios dos testes incluem a prevenção de bugs, a redução dos custos de desenvolvimento e a melhoria do desempenho.
- Usar testes unitários
- Usar testes de integração
- Usar testes de comportamento (behavior)
- Usar mocks

Conteúdos

- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software! (<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Artigo** Uma visão geral de testes em Javascript (<https://medium.com/devzera/uma-vis%C3%A3o-geral-de-testes-em-javascript-em-2018-8484154caf63>)
- **Artigo** Criando testes unitários em javascript (<https://ryogawa.medium.com/criando-testes-unit%C3%A1rios-em-javascript-ce34c3079ab9>)
- **Artigo** O que é teste de software? Iniciando TDD com JavaScript (<https://medium.com/@murilo001os/o-que-%C3%A9-teste-de-software-iniciando-tdd-com-javascript-985e9dc7fc09>)
- **YouTube** Roger Melo: JavaScript puro: Introdução ao TDD (https://www.youtube.com/watch?v=hwy0I7_XRE)
- **YouTube** Erick Wendel: Como fazer testes automatizados em JavaScript (<https://www.youtube.com/watch?v=8hxxZkBzWhk>)
- **YouTube** DevPleno: Como criar teste unitários em Angular (https://www.youtube.com/watch?v=MQs8_Klj_PU)
- **YouTube** Hype IT Brasil: Testes Unitários com JASMINE (https://www.youtube.com/watch?v=tYe_5ufcka4)

Conteúdos Alura:

- **Artigo** Tipos de testes: quais os principais e por que utilizá-los? (<https://www.alura.com.br/artigos/tipos-de-testes-principais-por-que-utiliza-los>)
- **Artigo** Dicas para desenvolver testes unitários e de integração no Front-end (<https://www.alura.com.br/artigos/dicas-desenvolver-testes-unitarios-integracao-front-end>)
- **YouTube** Pirâmide de Testes (<https://www.youtube.com/watch?v=gVJ1Lap-JYA>)
- **Podcast** Testes de Software – Hipsters Ponto Tech #382 (<https://www.hipsters.tech/techguide-testes-de-software-hipsters-ponto-tech-382/>)
- **Podcast** Testes Automatizados – Hipsters #51 (<https://www.hipsters.tech/testes-automatizados-hipsters-51/>)

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

Conteúdos Alura:

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

JavaScript - ES6:

- Conhecer as diferenças dessa versão do JavaScript

Conteúdos

- [Site](https://acervolima.com/introducao-ao-es6/) Acervo Lima: Introdução ao ES6 (<https://acervolima.com/introducao-ao-es6/>)
- [Artigo](https://medium.com/@matheusml/o-guia-do-es6-tudo-que-voc%C3%AA-precisa-saber-8c287876325f) O Guia do ES6: TUDO que você precisa saber (<https://medium.com/@matheusml/o-guia-do-es6-tudo-que-voc%C3%AA-precisa-saber-8c287876325f>)
- [Artigo](https://medium.com/tech-grupozap/sim-outro-post-sobre-ecmascript-2015-ou-es6-6d50a7f33bd4) Sim, outro post sobre ECMAScript 2015 ou ES6 (<https://medium.com/tech-grupozap/sim-outro-post-sobre-ecmascript-2015-ou-es6-6d50a7f33bd4>)
- [Artigo](https://medium.com/trainingcenter/entendendo-m%C3%B3dulos-no-javascript-73bce1d64dbf) Entendendo módulos ES6 (<https://medium.com/trainingcenter/entendendo-m%C3%B3dulos-no-javascript-73bce1d64dbf>)
- [YouTube](https://www.youtube.com/watch?v=yRr8Wo4XfYY) CODECASTS: O que é ES6? (<https://www.youtube.com/watch?v=yRr8Wo4XfYY>)
- [YouTube](https://www.youtube.com/watch?v=vylVbb2PYOM) Tiago Matos: Reutilizando funções dinamicamente com javascript ES6 módulos (import/export) (<https://www.youtube.com/watch?v=vylVbb2PYOM>)

Conteúdos Alura:

- **YouTube** Alura: JavaScript avançado I: ES6, orientação a objetos e padrões de projetos (<https://www.youtube.com/watch?v=T-JVRDnykYg>)
- **Curso** JavaScript: evoluindo a sua aplicação com ES6+ (<https://www.alura.com.br/curso-online-javascript-evoluindo-sua-aplicacao-es6>)

JavaScript - Modularização:

- Isolar partes do código em módulos
- Usar import e export

Conteúdos

- **Site** MDN Web Docs: Módulos JavaScript (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Modules>)
- **Artigo** Utilizando Módulos Javascript De Forma Nativa (<https://blog.matheuscastiglioni.com.br/utilizando-modulos-javascript-de-forma-nativa/>)
- **YouTube** Evandro Falleiros: Introdução à POO com JS - Modularização e importações (<https://www.youtube.com/watch?v=KL0qEMbTYFE>)
- **YouTube** Professor Edson Maia: JS OO Modularizar o código import e export (<https://www.youtube.com/watch?v=RMK5WdPGWDU>)

Conteúdos Alura:

- **Artigo** Um guia para importação e exportação de módulos com JavaScript (<https://www.alura.com.br/artigos/guia-importacao-exportacao-modulos-javascript>)
- **Site** Primeiras aulas do curso JavaScript: de padrões a uma abordagem funcional (<https://www.alura.com.br/conteudo/javascript-padroes-abordagem-funcional>)
- **Curso** Curso JavaScript: de padrões a uma abordagem funcional (<https://www.alura.com.br/curso-online-javascript-padroes-abordagem-funcional>)

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-semantic-breve-introducao>)
- **Artigo** Simplificando o controle de versão com Conventional Commits (<https://www.alura.com.br/artigos/simplificando-controle-versao-conventional-commits>)

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.

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>)
- **YouTube** Matheus Castiglioni: Jest - Iniciando com testes e TDD (<https://www.youtube.com/watch?v=sLx00kXPoTg>)

Conteúdos Alura:

- **Curso** Node.js: testes unitários e de integração (<https://cursos.alura.com.br/course/nodejs-testes-unitarios-integracao>)

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** Embaixadora da Qualidade: Guia Rápido Sobre Cypress (<https://www.youtube.com/watch?v=ZWF7QT3ogk0>)
- **YouTube** Otavio Lemos: Guia Rápido Sobre Cypress (<https://www.youtube.com/watch?v=e5SuUUFIDss>)

Conteúdos Alura:

- **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>)

Nível 3

Estruturas de Dados:

- No contexto dos computadores, uma estrutura de dados é uma forma específica de armazenar e organizar os dados na memória do computador para que esses dados possam ser facilmente recuperados e utilizados de forma eficiente quando necessário posteriormente.

- Conhecer as principais estruturas de dados
- Implementar as principais estruturas de dados

Conteúdos

- **Artigo** Estrutura de dados (<https://medium.com/php-word/estrutura-de-dados-6afb386a6fc1>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 1 (<https://medium.com/@PaoloProdossimoLopes/01-estrutura-de-dados-e-algoritmo-9cb6a9b14aa>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 2 (<https://medium.com/@PaoloProdossimoLopes/02-estrutura-de-dados-e-algoritmo-deba0b384a0f>)
- **YouTube** Mario Souto - Dev Soutinho: Estruturas de dados com JavaScript (<https://www.youtube.com/watch?v=MweeZn1rR8s>)
- **YouTube** Fabio Akita: Árvores (<https://www.youtube.com/watch?v=9GdesxWtOgs>)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Lista Encadeada (https://www.youtube.com/watch?v=RW0oD2L_tSg)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Filas (https://www.youtube.com/watch?v=RdoLTgJSV_c)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Pilhas (<https://www.youtube.com/watch?v=ZdU4wMyiTSs>)

Conteúdos Alura:

- **Site** Apostila Python: Estruturas de dados (<https://www.alura.com.br/apostila-python-orientacao-a-objetos/estrutura-de-dados>)
- **Artigo** Estruturas de dados: uma introdução (<https://www.alura.com.br/artigos/estruturas-de-dados-introducao>)
- **Artigo** Curso Estrutura de Dados: computação na prática com Java (<https://www.alura.com.br/artigos/estrutura-dados-computacao-na-pratica-com-java>)
- **Podcast** Hipsters.tech: Algoritmos e estrutura de dados - Hipsters 186 (<https://www.alura.com.br/podcast/hipsterstech-algoritmos-e-estrutura-de-dados-hipsters-186-a375>)
- **Curso** Curso PHP: dominando as Collections (<https://cursos.alura.com.br/course/php-dominando-collections>)
- **Curso** Curso Python: crie a sua primeira aplicação (<https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao>)
- **Desafio** 7 Days of Code: Estruturas de Dados (<https://7daysofcode.io/matricula/estruturas-de-dados>)

Conceitos de Orientação a Objetos:

- A Programação Orientada a Objetos é um paradigma de programação de software baseado na composição e interação entre diversas unidades chamadas de 'objetos' e as classes, que contêm uma identidade, propriedades e métodos). Ela é baseada em quatro componentes da programação: abstração digital, encapsulamento, herança e polimorfismo.
- Como funcionam objetos
- Criar e utilizar construtores
- O que são classes
- Criar e utilizar métodos

- Como funciona encapsulamento
- O que é herança
- O que é polimorfismo
- Como funcionam interfaces
- O que são abstrações

Conteúdos

- **Artigo** Princípios básicos da Programação Orientação a Objetos (POO) (<https://yanborowski.medium.com/princ%C3%ADpios-b%C3%A1sicos-da-programa%C3%A7%C3%A3o-orienta%C3%A7%C3%A3o-a-objetos-poo-62da3998b7ce>)
- **YouTube** FernandaDev: O que é Programação Orientada a Objetos?(POO) (<https://www.youtube.com/watch?v=QJY2TNYl-8>)
- **YouTube** Giuliana Bezerra: Desmistificando os tipos de referência - Ep. 8 (<https://youtu.be/D9hANYbPYCc>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Orientação a Objetos – Hipsters Ponto Tech #350 (<https://www.hipsters.tech/techguide-orientacao-a-objetos-hipsters-ponto-tech-350/>)
- **Artigo** POO: o que é programação orientada a objetos? (<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>)
- **Site** Apostila: Java e Orientação a Objetos (<https://www.alura.com.br/apostila-java-orientacao-objetos>)
- **Site** Livro: Orientação a Objetos (<https://www.casadocodigo.com.br/products/livro-oo-conceitos>)
- **YouTube** Alura: Orientação a objetos (com Roberta Arcoverde) (<https://www.youtube.com/watch?v=jpuJ1grluoU>)
- **YouTube** Alura: Imersão Java: Orientação a Objetos, APIs e além (e live coding!) (<https://www.youtube.com/watch?v=WdT90ffB-0Q>)
- **Curso** Formação Java e Orientação a Objetos (<https://cursos.alura.com.br/formacao-java>)
- **Curso** Formação C# e Orientação a Objetos (<https://cursos.alura.com.br/formacao-c-sharp-orientacao-objetos>)
- **Curso** Formação Python e orientação a objetos (<https://cursos.alura.com.br/formacao-Python-linguagem>)
- **Curso** Curso Orientação a Objetos com PHP: Classes, métodos e atributos (<https://www.alura.com.br/curso-online-php-oo-classes-metodos-atributos>)
- **Curso** Curso JavaScript: programação orientada a objetos (<https://www.alura.com.br/curso-online-javascript-passos-programacao-orientada-objetos>)

JavaScript - Armazenamento:

- Armazenar dados no front-end com localStorage
- Manipular dados armazenados
- Persistir dados armazenados

Conteúdos

- **Site** MDN Web Docs: Armazenamento do lado do cliente (https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage)

- **Site** MDN Web Docs: Usando a API Web Storage (https://developer.mozilla.org/pt-BR/docs/Web/API/Web_Storage_API/Using_the_Web_Storage_API)
- **Artigo** Armazenando e manipulando dados no localStorage (<https://medium.com/@lameckanao/armazenando-e-manipulando-dados-no-localstorage-7bcc901ba12b>)
- **Artigo** Funcionalidades do Local Storage (<https://medium.com/jaguaribetech/dlskaddaldkslkdlskdlk-333dae8ef9b8>)
- **YouTube** Fabiano Gabardo Lemos: Armazenando dados do formulário em LocalStorage (<https://www.youtube.com/watch?v=i04nMOowrII>)
- **YouTube** Roger Melo: Como armazenar e persistir dados no front-end com o localStorage (<https://www.youtube.com/watch?v=hNTozXI-qJA>)
- **YouTube** Luis Tavares: Como armazenar vários dados (array) no localStorage (<https://www.youtube.com/watch?v=YlAs11EilJc>)

Conteúdos Alura:

- **Site** Primeiras aulas do curso JavaScript na Web: armazenando dados no navegador (<https://www.alura.com.br/conteudo/javascript-manipulacao-elementos-localstorage>)
- **Curso** Curso JavaScript na Web: armazenando dados no navegador (<https://www.alura.com.br/curso-online-javascript-web-armazenando-dados-navegador>)

JavaScript - Concorrência:

- Programação concorrente é um paradigma de programação para a construção de programas que fazem uso da execução simultânea de várias tarefas computacionais interativas, que podem ser implementadas como programas separados ou como um conjunto de threads criadas por um único programa.
- Executar tarefas paralelamente

Conteúdos

- **Site** MDN Web Docs: Thread (inglês) (<https://developer.mozilla.org/en-US/docs/Glossary/Thread>)
- **Artigo** Multithreading using Javascript (inglês) (<https://medium.com/@sarthakvit/multithreading-using-javascript-cf9ad7cf9cfe>)
- **Artigo** Multithreading Javascript (inglês) (<https://medium.com/techtrument/multithreading-javascript-46156179cf9a>)
- **YouTube** DevPleno: Mono ou multi thread? POO para JS (<https://www.youtube.com/watch?v=iGV8gjzx3to>)
- **YouTube** Canal dotNET: Desenvolvimento Multithread em Aplicações JavaScript (<https://www.youtube.com/watch?v=jroozZAWeTg>)

Conteúdos Alura:

- **Artigo** Concorrência ou paralelismo: Threads, Processes, Fibers e Actors (<https://www.alura.com.br/artigos/concorrenca-ou-paralelismo-threads-processes-fibers-e-actors>)
- **Curso** Formação: Desenvolva aplicações Web em JavaScript com tarefas concorrentes e orientadas a objetos (<https://cursos.alura.com.br/formacao-aplicacoes-javascript-tarefas-concorrentes-orientadas-objetos>)

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-produtividade>)
- **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>)

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>)

Micro Frontends:

- Micro Frontends são uma abordagem arquitetural na qual uma aplicação web é dividida em partes independentes, cada uma representando uma parte específica da interface do usuário.
- Divide uma aplicação web em partes independentes, facilitando o desenvolvimento e a manutenção.
- Possibilita a implantação e atualização contínua de partes específicas da aplicação sem afetar o aplicativo como um todo.
- Permite que equipes trabalhem de forma independente em diferentes partes da aplicação.
- Integra diferentes tecnologias e frameworks para desenvolver microfrontends, como React, Angular, Vue.js, entre outros.

Conteúdos

- **Artigo** Micro Frontends (<https://medium.com/@lorenamelor/micro-frontends-ac0e5d87582a>)
- **Artigo** O poder dos Micro-Frontends: Uma abordagem moderna para a Arquitetura Frontend (<https://aws.amazon.com/pt/blogs/aws-brasil/o-poder-dos-micro-frontends-uma-abordagem-moderna-para-a-arquitetura-frontend/>)
- **YouTube** Implementando micro front-end com Single SPA e React (<https://youtu.be/68LaXOWwxZI?si=PM2te6z9l1Qymshl>)

Conteúdos Alura:

- **Podcast** Microfrontends – Hipsters #164 (<https://www.hipsters.tech/microfrontends-hipsters-164/>)
- **Podcast** Micro Frontends na Conta Azul – Hipsters On The Road #26 (<https://www.hipsters.tech/micro-frontends-no-counta-azul-hipsters-on-the-road-26/>)
- **Site** Plurall e a Evolução da Arquitetura Front-End (<https://cursos.alura.com.br/extra/cases/plurall-e-a-evolucao-da-arquitetura-front-end-z1064>)
- **Site** Klarna e a Evolução da Arquitetura Front-End (<https://cursos.alura.com.br/extra/cases/klarna-e-a-evolucao-da-arquitetura-front-end-z1283>)
- **Site** Idwall e a Evolução da stack de Javascript (<https://cursos.alura.com.br/extra/cases/idwall-e-a-evolucao-da-stack-de-javascript-z1290>)
- **Artigo** Monorepos descomplicados: explorando o NX (<https://www.alura.com.br/artigos/monorepos-nx>)
- **Curso** Single SPA: roteando micro front-ends de forma avançada (<https://www.alura.com.br/curso-online-single-spa-roteando-micro-frontends-forma-avancada>)
- **Curso** Cursos de micro front-ends: implementando a arquitetura com Single SPA (<https://cursos.alura.com.br/formacao-cursos-micro-front-ends-implementando-arquitetura-single-spa>)

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** Rafaela Ballerini: O que é Git e GitHub? - definição e conceitos importantes 1/2 (<https://www.youtube.com/watch?v=DqTITcMq68k>)
- **YouTube** Rafaela 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/MqogPzjQyCY?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_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://julicarneiro.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=oCY5YEEjlwE>)

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=vxES6brd-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=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-svg-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-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>)