

Angular

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](https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics) 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=IIF6vD-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=iohhj-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>)

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>)

Angular - Fundamentos:

- Angular é uma framework de construção de aplicações e plataforma de desenvolvimento construído em TypeScript para criar aplicações eficientes e sofisticadas de página única (SPA).
- Construir interfaces utilizando HTML, CSS e TypeScript
- Criar aplicações SPA
- Construir aplicações web, mobile ou desktop
- Integrar dados com API's REST
- Utilizar a composição para criar componentes reutilizáveis
- Utilizar serviços do tipo Resolver
- Manipular requisições criando serviços do tipo Interceptor

Conteúdos

- **Site** Documentação Angular (inglês) (<https://angular.dev/overview>)
- **Artigo** Bê-á-bá do Angular (<https://belmirofss.medium.com/b%C3%AA-%C3%A1-b%C3%A1-do-angular-8f23a109b4fc>)
- **Artigo** Angular I — Iniciando um projeto do zero (<https://medium.com/@iagomachado/angular-i-iniciando-um-projeto-do-zero-5290da9fa90e>)
- **YouTube** Mario Souto - Dev Soutinho: Como APRENDER um FRAMEWORK Front End? Como eu aprendi Angular? (<https://www.youtube.com/watch?v=mMMqMk2buzs>)

- **YouTube** Loiane Groner: Curso Angular 2 (<https://www.youtube.com/watch?v=tPOMG0D57S0&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G>)

Conteúdos Alura:

- **Artigo** Angular: o que é, para que serve e um Guia para iniciar no framework JavaScript (<https://www.alura.com.br/artigos/angular-js>)
- **Artigo** Como começar com Angular (<https://www.alura.com.br/artigos/como-comecar-com-angular>)
- **Artigo** Fundamentos de Angular e TypeScript (<https://www.alura.com.br/artigos/fundamentos-de-angular-typescript>)
- **Artigo** Angular: como funciona um componente? (<https://www.alura.com.br/artigos/angular-como-funciona-um-componente>)
- **Podcast** Hipsters.tech: Evoluções no Angular e Outros Frameworks Front-End – Hipsters Ponto Tech #383 (<https://www.hipsters.tech/evolucoes-no-angular-e-outros-frameworks-front-end-hipsters-ponto-tech-383/>)
- **YouTube** Alura: O que é Angular e AngularJS? (<https://www.youtube.com/watch?v=LFINU30u7d8>)
- **Curso** Formação Angular: crie aplicações web ágeis (<https://www.alura.com.br/formacao-angular-crie-aplicacoes-web-ageis>)
- **Curso** Formação: Explore o Framework Angular (<https://www.alura.com.br/formacao-angular>)

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=QJjY2TNyl-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=jpuJ1qrluoU>)
- **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>)

RxJS - Fundamentos:

- A RxJS é uma biblioteca para a composição de programas assíncronos e baseados em eventos, utilizando sequências observáveis. Ela fornece o tipo núcleo, o Observable, tipos satélite (Observer, Schedulers, Subjects) e operadores inspirados nos métodos Array (map, filter, reduce, every, etc.) para permitir a manipulação de eventos assíncronos como coleções.
- Criar programas assíncronos
- Criar programas baseados em eventos
- Entender o conceito de observables e sequências de observables
- Entender como usar Observers, Subscription, Subject

Conteúdos

- **Site** Documentação RxJS (inglês) (<https://rxjs.dev/guide/overview>)
- **Artigo** O que é RxJS? (<https://blog.flavioflima.com/2018/08/o-que-e-rxjs.html>)
- **Artigo** RxJS – Entendendo o conceito na prática
(<https://bognarjunior.wordpress.com/2018/10/07/rxjs-entendendo-o-conceito-na-pratica/>)
- **YouTube** Paulo Eduardo Melo: Observable (<https://www.youtube.com/watch?v=3n-WCiSxu2k>)
- **YouTube** Paulo Eduardo Melo: Observable a fundo
(https://www.youtube.com/watch?v=dm_9xWPxLUg)

Conteúdos Alura:

- **YouTube** Alura: Programação Reativa - Alura Live 54
(<https://www.youtube.com/watch?v=N0ie4xe48Ng>)
- **Site** Primeiras aulas do curso Angular e RxJS: programação reativa
(<https://www.alura.com.br/conteudo/angular-rxjs-introducao-programacao-reativa>)
- **Curso** Curso Angular e RxJS: programação reativa
(<https://www.alura.com.br/curso-online-angular-rxjs-introducao-programacao-reativa>)

O pattern Observer:

- Em engenharia de software, o padrão (pattern) chamado Observer é um padrão de projeto de software no qual um objeto, chamado de sujeito (subject), mantém uma lista de seus dependentes, chamados de observadores (observers), e os notifica automaticamente de qualquer mudança de estado, geralmente chamando um de seus métodos.
- Entender o que são Design Patterns
- Atualizar diversos elementos simultaneamente usando Observers
- Declarar os Subjects
- Criar programas baseados em eventos

Conteúdos

- **Artigo** Padrões em JS: Observer Pattern
(<https://oieduardorabelo.medium.com/padr%C3%B5es-em-js-observer-pattern-bff0ecc55d01>)
- **Artigo** Observer Pattern - Design Patterns com Typescript
(<https://meneguete.com/2019/06/23/design-patterns-com-typescript-observer/>)
- **Artigo** Entendendo RxJS Observable com Angular
(<https://medium.com/tableless/entendendo-rxjs-observable-com-angular-6f607a9a6a00>)
- **YouTube** Filipe Deschamps: Observer Pattern
(https://www.youtube.com/watch?v=4OLCrCib_So)

- **YouTube** Paulo Eduardo Melo: Observable (<https://www.youtube.com/watch?v=3n-WCiSxu2k>)
- **YouTube** Paulo Eduardo Melo: Observable a fundo (https://www.youtube.com/watch?v=dm_9xWPxLUg)

Conteúdos Alura:

- **YouTube** Alura: Programação Reativa - Alura Live 54 (<https://www.youtube.com/watch?v=N0ie4xe48Ng>)
- **Podcast** Design Patterns - Hipsters 206 (<https://www.alura.com.br/podcast/design-patterns-hipsters-206-a345>)
- **Site** Primeiras aulas do curso Angular e RxJS: programação reativa (<https://www.alura.com.br/conteudo/angular-rxjs-introducao-programacao-reativa>)
- **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 em PHP: padrões comportamentais (<https://www.alura.com.br/curso-online-php-design-pattern-comportamental>)
- **Curso** Curso Design Patterns C# I: boas práticas de programação (<https://www.alura.com.br/curso-online-design-patterns-dotnet>)

Nível 2

Angular - Templates:

- Em Angular, templates são um modelo para um fragmento de uma interface de usuário (IU). Templates são escritos em HTML, e uma sintaxe especial pode ser usada dentro de um template para construir com base em muitas das características do Angular.

Conteúdos

- **Artigo** Angular: NgTemplate (<https://medium.com/criciumadev/angular-ngtemplate-d59b32d47d01>)
- **Artigo** ngTemplateOutlet: O segredo da personalização (<https://andrewrosario.medium.com/ngtemplateoutlet-o-segredo-da->

[personaliza%C3%A7%C3%A3o-6f3203787a56\)](#)

- **YouTube** Loiane Groner: Introdução aos Templates
(<https://www.youtube.com/watch?v=bBFScNfW904>)

Conteúdos Alura:

- **Site** Primeiras aulas do curso Angular: explorando o framework
(<https://www.alura.com.br/conteudo/angular-comecando-framework>)
- **Curso** Curso Angular: explorando o framework
(<https://www.alura.com.br/curso-online-angular-comecando-framework>)
- **Curso** Acessibilidade no Angular: aprimorando formulários, modais e rotas
(<https://www.alura.com.br/curso-online-acessibilidade-angular-formularios-modais-rotas>)

Angular - Renderização:

- Uma aplicação Angular normal é executada no navegador, renderizando páginas no DOM em resposta às ações do usuário. A Angular Universal executa no servidor, gerando páginas de aplicação estática que mais tarde são inicializadas no cliente.
- Exibir uma página Angular no navegador
- Realizar renderização no lado do servidor

Conteúdos

- **Site** Documentação Angular Server-Side Rendering (inglês)
(<https://angular.dev/guide/ssr#enable-server-side-rendering>)
- **Artigo** Angular Universal — Renderizando sua aplicação no Servidor
(<https://medium.com/@arturquirino/angular-universal-renderizando-sua-aplica%C3%A7%C3%A3o-no-servidor-f9ac617237d2>)
- **Artigo** Performance em Angular — Dicas e truques
(<https://medium.com/codigorefinado/performance-em-angular-dicas-e-truques-deeb6cb3860c>)
- **Artigo** Introdução a Renderização Dinâmica
(<https://reulison.com.br/renderizacao-dinamica/>)

Conteúdos Alura:

- **Artigo** Desbravando o SSR no Angular 17
(<https://www.alura.com.br/artigos/ssr-angular-17>)
- **Artigo** CSR, SSR e SSG: entenda as diferenças e aplicações
(<https://www.alura.com.br/artigos/csr-ssr-ssg>)
- **Site** Primeiras aulas do curso Angular: explorando o framework
(<https://www.alura.com.br/conteudo/angular-comecando-framework>)
- **Curso** Curso Angular: explorando o framework
(<https://www.alura.com.br/curso-online-angular-comecando-framework>)
- **Curso** Formação Angular: avance com service workers, signals e SSR
(<https://cursos.alura.com.br/formacao-angular-avance-service-workers-signals-ssr>)

Angular - Services:

- Um serviço é uma categoria ampla que engloba qualquer valor, função ou funcionalidade que uma aplicação necessite. Um serviço é tipicamente uma classe com um propósito conciso e bem definido. O Angular distingue componentes de serviços, para aumentar a modularidade e a reusabilidade.
- Criar dados que estarão sempre disponíveis
- Dividir a aplicação web em diversas partes

Conteúdos

- **Artigo** O que é um serviço no Angular e porquê usá-lo?
(<https://dumildematos.medium.com/o-que-%C3%A9-um-servi%C3%A7o-no-angular-e-por-que-us%C3%A1-lo-2c8b441618cf>)
- **Artigo** Herança de services em Angular
(<https://vanderloureiro.medium.com/heran%C3%A7a-de-services-em-angular-fca7a97b5f0c>)
- **YouTube** Loiane Groner: Criando um serviço (Service)
(<https://www.youtube.com/watch?v=8fE1e9j6fJo>)

Conteúdos Alura:

- **Artigo** Services e injeção de dependência no Angular (<https://www.alura.com.br/artigos/services-injecao-dependencia-angular-o-que-sao-como-funcionam>)
- **Site** Primeiras aulas do curso Angular: explorando o framework (<https://www.alura.com.br/conteudo/angular-comecando-framework>)
- **Curso** Curso Angular: explorando o framework (<https://www.alura.com.br/curso-online-angular-comecando-framework>)

Angular - Roteamento:

- Navegar até um componente
- Incluir um parâmetro de rota
- Controlar o fluxo de navegação do seu usuário com guarda de rotas

Conteúdos

- **Site** Angular Routing (inglês) (<https://angular.dev/guide/routing>)
- **Artigo** Angular 5: Trabalhando com Rotas (<https://medium.com/angularbr/angular-5-trabalhando-com-rotas-8335617fcdbc>)
- **Artigo** Layout, rotas e parâmetros (<https://kosmicke.medium.com/aula-01-layout-rotas-e-par%C3%A2metros-80f44dafb43a>)
- **YouTube** Loiane Groner: Rotas - Introdução (<https://www.youtube.com/watch?v=wzdKc0AFY6k&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=51>)
- **YouTube** Loiane Groner: Rotas - Configurando rotas simples (<https://www.youtube.com/watch?v=8OHOAZ6j0Rg&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=52>)

Conteúdos Alura:

- **Artigo** Como usar a Guarda de Rotas Funcional em Angular? (<https://www.alura.com.br/artigos/guarda-de-rotas-funcional-angular>)
- **Site** Primeiras aulas do curso Angular: controle o fluxo de navegação (<https://www.alura.com.br/conteudo/angular-comecando-framework>)

- **Curso** Curso Angular: controle o fluxo de navegação (<https://www.alura.com.br/curso-online-angular-controle-fluxo-navegacao>)

Angular - CLI (Interface de Linha de Comando):

- A Interface de Linha de Comando (CLI) do Angular é uma ferramenta de interface de linha de comando que você utiliza para inicializar, desenvolver, estruturar e manter aplicações Angular diretamente de um shell de comando.
- Aprender a sintaxe 'ng [argumento-opcional] [opções]'
- Conhecer os comandos mais importantes, como 'ng add', 'ng build', 'ng update', 'ng deploy', 'ng new', 'ng test', entre outros

Conteúdos

- **Site** Documentação Angular CLI (inglês) (<https://angular.dev/tools/cli/setup-local>)
- **YouTube** Cristian William Dev: Angular CLI - Comandos que você deve saber (<https://www.youtube.com/watch?v=llbBYa07u3Q>)
- **YouTube** Loiane Groner: Angular CLI - Instalação e criação de projetos: ng new e ng serve (https://www.youtube.com/watch?v=RlfOhrhPh_Y)
- **YouTube** Loiane Groner: Angular CLI - Criando components, services: ng generate (<https://www.youtube.com/watch?v=8x-5Q2YzvuE>)
- **YouTube** Loiane Groner: Angular CLI - Estrutura do projeto (<https://www.youtube.com/watch?v=ICvq9YeDCh0>)
- **YouTube** Loiane Groner: Angular CLI - Gerando build de produção (<https://www.youtube.com/watch?v=U0zHj14mNrl>)

Conteúdos Alura:

- **Artigo** Criando aplicações Angular com Angular CLI (<https://www.alura.com.br/artigos/criando-aplicacoes-angular-com-angular-cli>)

Nível 3

Angular - Gerenciamento de Estado:

- Um 'estado' (state) é qualquer dado necessário para reconstruir UI (Interface de Usuário) qualquer momento. A alteração desses dados desencadeará um redesenho da interface do usuário. O gerenciamento do estado é o conceito de adicionar, atualizar, remover e ler esses dados e seus 'estados' em uma aplicação.
- Atualizar componentes em tempo real
- Esperar por alterações em algum componente e executar alterações
- Usar Redux, NGXS e outros

Conteúdos

- **Artigo** Pensando Reativamente - Teoria e Prática com Angular e RxJS (<https://medium.com/@richardleecba/pensando-reativamente-91361cfdccc5>)
- **Artigo** Gerenciamento de estado com Redux (<https://medium.com/@marcossantosdev/gerenciamento-de-estado-com-redux-d048c0c9d0ad>)
- **Artigo** Como Estruturar o seu Front-End com Angular (<https://ricardo-mello.medium.com/como-estruturar-o-seu-front-end-com-angular-7fd0e7ef31a2>)
- **Artigo** Padrões e Boas Práticas em Angular (<https://andrewrosario.medium.com/padr%C3%B5es-e-boas-pr%C3%A1ticas-em-angular-que-te-ajudar%C3%A3o-a-escalar-5001e544e7de>)
- **YouTube** GDG Maringá: Gerenciamento de estado no Angular com NgRx e programação reativa (<https://www.youtube.com/watch?v=9taz2c-l2Yk>)
- **YouTube** Matheus Castiglioni: Angular - Gerenciando estado com NGRX (<https://www.youtube.com/watch?v=BobNwRgz6Uw>)

Conteúdos Alura:

- **Curso** Curso Angular: design de componentes com acessibilidade (<https://www.alura.com.br/curso-online-angular-introducao-componentes->

[acessibilidade](#))

Angular - Formulários:

- Um formulário web é uma página online que aceita injeção de dados (inputs) do usuário. É uma página interativa que imita um documento ou formulário em papel, onde os usuários preenchem determinados campos.
- Criar formulários com Template Forms
- Criar formulários reativos com Reactive Forms

Conteúdos

- **Artigo** Tutorial — Reactive Forms com Angular
(<https://medium.com/mobicareoficial/tutorial-reactive-forms-com-angular-d4f6c050a794>)
- **Artigo** Criando formulários com etapas no Angular
(<https://medium.com/inside-picpay/criando-formul%C3%A1rios-com-etapas-no-angular-135e9fb48d99>)
- **Artigo** Formulários Dinâmicos com Angular
(<https://medium.com/@paulogabrielvieira/formul%C3%A1rios-din%C3%A2micos-com-angular-d538aee17834>)
- **YouTube** Loiane Groner: Formulários (template vs data / reativo) Introdução
(<https://www.youtube.com/watch?v=TevqIDM7aY8&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=74>)
- **YouTube** Loiane Groner: Formulários - Criando o projeto inicial com Bootstrap 3
(<https://www.youtube.com/watch?v=j7OZGu3hlQ0&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=75>)
- **YouTube** Loiane Groner: Formulários reativos (data driven) Introdução
(<https://www.youtube.com/watch?v=aJz-BhRxN38&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=90>)
- **YouTube** Loiane Groner: Formulários reativos: Criando um form com código Angular
(<https://www.youtube.com/watch?v=931fynCdTKw&list=PLGxZ4Rq3BOBoSRcKWEdQACbUCNWLczg2G&index=92>)

Conteúdos Alura:

- **Curso** Curso Angular: formulários orientados a templates (<https://cursos.alura.com.br/course/angular-formularios-orientados-templates>)
- **Curso** Formação Angular: crie aplicações web ágeis (<https://cursos.alura.com.br/formacao-angular-crie-aplicacoes-web-ageis>)

Angular - Módulos:

- Um módulo (Module), diferentemente de um componente, não controla nenhuma view. Um módulo é constituído de um ou mais componentes. Uma aplicação Angular tem que ter no mínimo um módulo que contenha mínimo um componente (Component).
- Como usar a classe NgModule
- Declarar quais componentes podem ser usados por componentes de outros módulos
- Declarar quais services serão injetados
- Aprender a modularizar uma aplicação
- Carregar módulos como Lazy

Conteúdos

- **Artigo** Qual a diferença entre Module e Component no Angular? (<https://fabiodemiranda.com.br/diferenca-module-component-angular-curtinha-angular/>)
- **Artigo** Angular: Core vs Shared Modules (<https://medium.com/lfdev-blog/angular-core-vs-shared-modules-c6efb13fa1f1>)
- **Artigo** Angular - Módulos Lazy Loading (<https://medium.com/@iagomachado/angular-cap-ii-angular-lazy-loading-modules-88c175b81503>)
- **YouTube** Loiane Groner: Módulos (NgModule) (<https://www.youtube.com/watch?v=36kd3uR-hG8>)
- **YouTube** Loiane Groner: Criando um módulo de funcionalidade (<https://www.youtube.com/watch?v=P4GYpzNNEw4>)

- **YouTube** Loiane Groner: Forms (template driven) Módulos e FormsModule (<https://www.youtube.com/watch?v=9z47wkaaqlmc>)
- **YouTube** Loiane Groner: Formulários reativos: Configuração (Módulo e Componente) (<https://www.youtube.com/watch?v=2DoQX5CVvAk>)

Conteúdos Alura:

- **Curso** Formação: Desenvolva Aplicações Escaláveis com Angular (<https://cursos.alura.com.br/formacao-aplicacoes-escalaveis-angular>)

Angular - Injeção de Dependências:

- A injeção de dependência permite declarar as dependências de suas classes TypeScript sem cuidar de sua instanciação. Em vez disso, o Angular trata da instanciação para você. Este design pattern permite escrever um código mais testável e flexível.
- Declarar as dependências de suas classes
- Injetar dependências em um componente
- Conhecer os Injection Tokens
- Configurar providers
- Entender a Injeção de Dependência Hierárquica

Conteúdos

- **Site** Documentação - Dependency injection (inglês) (<https://angular.dev/guide/di/dependency-injection>)
- **Site** Injeção de dependência (<https://ajuda.gitlab.io/guia-rapido/angular/injecao-de-dependencia/>)
- **Artigo** Fabricando Injeções de Dependência no Angular (<https://andrewrosario.medium.com/fabricando-inje%C3%A7%C3%B5es-de-depend%C3%Aancia-no-angular-7b2e2bc98f61>)
- **YouTube** Loiane Groner: Injeção de Dependência (DI) + como usar um serviço em um componente (<https://www.youtube.com/watch?v=KqsFxBMIM9s>)
- **YouTube** Loiane Groner: Injetando um serviço em outro serviço (<https://www.youtube.com/watch?v=hMhCDO75Aus>)

Conteúdos Alura:

- **Artigo** Services e injeção de dependência no Angular: o que são e como funcionam? (<https://www.alura.com.br/artigos/services-injecao-dependencia-angular-o-que-sao-como-funcionam>)
- **Curso** Formação Angular: crie aplicações web ágeis (<https://cursos.alura.com.br/formacao-angular-crie-aplicacoes-web-ageis>)
- **Curso** Curso Angular: explorando o framework (<https://www.alura.com.br/curso-online-angular-comecando-framework>)
- **Curso** Formação: Desenvolva Aplicações Escaláveis com Angular (<https://cursos.alura.com.br/formacao-aplicacoes-escalaveis-angular>)

Angular - 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
- Familiarizar-se com Jasmine e/ou Karma

Conteúdos

- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software! (<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Artigo** Testes de unidade usando Angular (<https://medium.com/totvsdevelopers/testes-de-unidade-usando-angular-b83ea6accf7c>)
- **Artigo** Escrevendo testes eficientes de verdade no Angular (<https://andrewrosario.medium.com/escrevendo-testes-eficientes-de-verdade-no-angular-263a9d03cecf>)

- **Artigo** Melhore seus testes e2e com Playwright usando o modelo Page Object (<https://www.alura.com.br/artigos/page-object-model>)
- **YouTube** Hype IT Brasil: Testes Unitários no Angular (<https://www.youtube.com/watch?v=RCi2EEwNZ6M>)
- **YouTube** Hype IT Brasil: Como testar um componente no Angular (<https://www.youtube.com/watch?v=RFHDWc-apLQ>)
- **YouTube** Liara Programmer: Como criar teste unitários em Angular (<https://www.youtube.com/watch?v=S6M7xWQmirs>)

Conteúdos Alura:

- **Artigo** Testes automatizados: como melhorar qualidade e confiabilidade do seu software (<https://www.alura.com.br/artigos/testes-automatizados>)
- **Curso** Curso Angular: testes automatizados com Jasmine e Karma (<https://www.alura.com.br/curso-online-angular-testes-automatizados-jasmin-karma>)
- **Curso** Curso Angular: avançando com testes automatizados (<https://www.alura.com.br/curso-online-angular-avancando-testes-automatizados>)
- **Curso** Curso Angular: explorando testes de unidade e integração (<https://cursos.alura.com.br/course/angular-explorando-testes-unidade-integracao>)
- **Curso** Curso Angular: escreva testes end-to-end com Playwright (<https://cursos.alura.com.br/course/angular-escreva-testes-end-to-end-playwright>)

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>)

Angular - Design System:

- Um Design System é uma coleção organizada de componentes, padrões e diretrizes que ajudam equipes a criar interfaces consistentes e escaláveis. Ele atua como um guia para o design e desenvolvimento de produtos digitais,

garantindo uniformidade e eficiência. No Angular, o Design System ganha ainda mais poder quando implementado em um monorepo e documentado com o Storybook.

- Criar um Design System com Angular;
- Organizar o projeto em um Monorepo;
- Usar o Storybook para documentar e testar os componentes criados.

Conteúdos

- **Site** Workspace and project file structure (<https://angular.dev/reference/configs/file-structure#multiple-projects>)
- **Site** Building Angular Apps in an Nx Monorepo (<https://nx.dev/getting-started/tutorials/angular-monorepo-tutorial#building-angular-apps-in-an-nx-monorepo>)
- **Site** Storybook para o Angular tutorial (<https://storybook.js.org/tutorials/intro-to-storybook/angular/pt/get-started/>)
- **Site** NX: Gerencie o seu monorepo (<https://medium.com/@lorenamelor/nx-gerencie-o-seu-monorepo-6ef731f316f0>)
- **YouTube** Monorepo (Por Que Essa Estratégia Funciona em Grandes Empresas?) (https://www.youtube.com/watch?v=BbNluUy_F0w)

Conteúdos Alura:

- **YouTube** Design Systems (<https://www.youtube.com/watch?v=MLGFctEtmYQ>)
- **Curso** Formação Angular: cursos para construir um Design System com Nx, Monorepo e Storybook (<https://cursos.alura.com.br/formacao-angular-cursos-construir-design-system-nx-monorepo-storybook>)

Habilidade Auxiliar: Infraestrutura e Back-end

Git e GitHub - Fundamentos:

- Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.

- GitHub é um serviço de hospedagem para desenvolvimento de software e controle de versão usando Git.
- Criar um repositório
- Clonar um repositório
- Fazer commit, push e pull de e para o repositório
- Reverter um commit
- Criar branches e pull requests
- Lidar com merge e conflitos

Conteúdos

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

Conteúdos Alura:

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

HTTP - Fundamentos:

- HTTP significa Hyper Text Transfer Protocol. A comunicação entre computadores cliente e servidores web é feita enviando solicitações HTTP e recebendo respostas HTTP.
- Entender a diferença dos verbos HTTP
- Testar os requests e ver os status codes no navegador

- Saber fazer uma requisição HTTP na linha de comando com WGET
- Baixar uma imagem com WGET
- Fazer um post

Conteúdos

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

Conteúdos Alura:

- **Artigo** HTTP: Desmistificando o protocolo da Web (<https://www.alura.com.br/artigos/desmistificando-o-protocolo-http-parte-1>)
- **Artigo** Métodos de requisição do HTTP (<https://www.alura.com.br/artigos/metodos-de-requisicao-do-http>)
- **Artigo** Qual é a diferença entre HTTP e HTTPS? (<https://www.alura.com.br/artigos/qual-e-diferenca-entre-http-e-https>)
- **Podcast** Hipsters.tech: HTTP/2: magia com o novo protocolo - Hipsters 13 (<https://www.alura.com.br/podcast/http-2-magia-com-o-novo-protocolo-hipsters-13-a573>)
- **Curso** Curso HTTP: Entendendo a web por baixo dos panos (<https://www.alura.com.br/curso-online-http-entendendo-web-por-baixo-dos-panos>)

JSON:

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

Conteúdos

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

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=Z45BTNeZ1I0>)
- **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>)

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_IAcQQ)
- **Curso** Formação Figma (<https://www.alura.com.br/formacao-figma>)

Componentes de design:

- Conhecer os componentes descrevem um layout ou interface

Conteúdos

- **Artigo** Componentes Principais UI
(<https://medium.com/@prikuer/componentes-principais-ui-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_VhzEygT6U)

Conteúdos Alura:

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

Como usar fontes:

- Escolher a fonte mais adequada para determinado projeto

Conteúdos

- **Artigo** Você sabe usar tipografia em UI Design? (<https://medium.com/ui-lab-school/voc%C3%AA-sabe-usar-tipografia-em-ui-design-9ce4ccdbab43>)
- **Artigo** As 20 fontes mais usadas pelos designers gráficos (<https://apenasumchico.medium.com/as-20-fontes-mais-populares-de-todos-os-tempos-38a6c121dfb>)
- **Artigo** Usando ícones SVG como fontes com o IcoMoon (<https://dev.to/sucodelarangela/usando-icone-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-incrivel>)
- **Site** Tipografia para Web (<https://tipografiaparaweb.netlify.app/>)
- **Site** Primeiras aulas do curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/conteudo/tipografia-conceito>)
- **Curso** Curso Tipografia: conhecendo o que há por trás dos tipos (<https://www.alura.com.br/curso-online-tipografia-conceito>)
- **Curso** Curso Design editorial: criação de materiais gráficos (<https://www.alura.com.br/curso-online-design-editorial>)

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