

Node.js

TechGuide - Alura, FIAP e PM3

Nível 1

Lógica de Programação:

- Aprender lógica de programação, fundamental para o desenvolvimento de software
- Conhecer as bases para se criar, analisar e resolver problemas computacionais de forma estruturada e eficiente
- Entender o que são tipos de dados
- Declarar variáveis, considerando os diferentes tipos
- Conhecer os operadores de atribuição e comparação
- Usar estruturas condicionais
- Usar estruturas de repetição e laços
- Usar funções, passando parâmetros e argumentos

Conteúdos

- **Artigo** Lógica de programação: o que é e para que serve? (<https://www.locaweb.com.br/blog/temas/codigo-aberto/logica-de-programacao-o-que-e/>)
- **Artigo** Lógica de programação: guia completo para iniciantes (<https://www.hostgator.com.br/blog/logica-de-programacao/>)
- **Site** Livro - Lógica de Programação (<https://www.casadocodigo.com.br/products/livro-programacao>)
- **YouTube** O MÍNIMO QUE VOCÊ PRECISA SABER ANTES DE PROGRAMAR! (<https://www.youtube.com/watch?v=BTENKdRVS2U>)
- **Podcast** Se você não aprender lógica de programação, você NÃO VAI CONSEGUIR VAGA (<https://open.spotify.com/episode/2XxEhO0cPflGp3oGRgGJLX?si=FtgYSOvaSZGo-B31ex5ANQ>)

Conteúdos Alura:**

- **Curso** Formação A partir do zero: iniciante em programação (<https://cursos.alura.com.br/formacao-programacao>)

JavaScript - Fundamentos:

- JavaScript é a linguagem de programação mais popular do mundo e é uma das principais tecnologias da World Wide Web, juntamente com HTML e CSS. Ela possui tipagem dinâmica, orientação a objetos baseada em protótipos e funções de primeira classe. Ela é multi-paradigma e suporta estilos de programação orientados a eventos, funcionais e imperativos.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando a diferença entre 'var', 'let' e 'const'
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação ('=', '==', '===')
- Usar estruturas de repetição e laços ('while', 'for')

- Usar funções, passando parâmetros e argumentos
- Manipular arrays e listas
- Aprender o conceito de Orientação a Objetos
- Fazer um CRUD
- Obter dados de uma API
- Fazer chamadas assíncronas usando 'Async/Await', 'Promise', etc

Conteúdos

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

Conteúdos Alura:**

- **Artigo** JavaScript: o que é, como aprender e um Guia da linguagem mais popular do mundo (<https://www.alura.com.br/artigos/javascript/>)
- **Podcast** Hipster 38 - O Reino encantado do JavaScript (<https://www.hipsters.tech/o-reino-encantado-do-javascript-hipsters-38/>)
- **Podcast** Hipster 169 - JavaScript: manual de sobrevivência 2020 (<https://www.hipsters.tech/javascript-manual-de-sobrevivencia-2020-hipsters-169/>)
- **Podcast** Hipster 236 - Evolução do JavaScript (<https://www.hipsters.tech/evolucao-do-javascript-hipsters-ponto-tech-236/>)
- **Artigo** Strings com JavaScript: o que são e como manipulá-las (<https://www.alura.com.br/artigos/strings-com-javascript-o-que-sao-e-como-manipular>)
- **Artigo** Como utilizar operadores de comparação em Javascript (<https://www.alura.com.br/artigos/operadores-matematicos-em-javascript>)

- **Artigo** Para que serve um Array? (<https://www.alura.com.br/artigos/javascript-para-que-serve-array>)
- **Artigo** Quando usar forEach e map? (<https://www.alura.com.br/artigos/javascript-quando-devo-usar-foreach-e-map>)
- **Artigo** Manipulação de array com map, filter e reduce (<https://www.alura.com.br/artigos/manipulacao-de-array-com-map-filter-e-reduce>)
- **Curso** Formação Desenvolva aplicações Web com JavaScript (<https://cursos.alura.com.br/formacao-javascript-front-end>)
- **YouTube** Alura: O que é JavaScript? (<https://www.youtube.com/watch?v=NaVSbnnV75Q>)
- **YouTube** Alura: Destructuring em JavaScript (<https://www.youtube.com/watch?v=f8a-qwKC5yk>)
- **YouTube** Alura: Classes x funções no Javascript (<https://www.youtube.com/watch?v=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>)

Node.js - Fundamentos:

- Node.js é um ambiente de execução JavaScript que permite executar aplicações desenvolvidas com a linguagem de forma autônoma, sem depender de um navegador.
- Conhecer operações bloqueantes e não-bloqueantes
- Aprender o conceito de laço de eventos (event loop)
- Aprender a usar as bibliotecas do Node.js, como 'net', 'fs', 'http', 'path', entre outras
- Entender como Timers funcionam

Conteúdos

- **Site** Introdução ao Node.js - documentação oficial (<https://nodejs.dev/pt/learn/>)
- **Site** Sobre Node.js (<https://nodejs.org/pt-br/about/>)
- **Artigo** O que é Node.js? Guia para iniciantes (<https://tecnoblog.net/responde/o-que-e-node-js-guia-para-iniciantes/>)
- **Artigo** Introdução ao Node.js (Single-Thread, Event-Loop e mercado) (<https://fabiojanio.medium.com/introdu%C3%A7%C3%A3o-ao-node-js-single-thread-event-loop-e-mercado-46edd82c1faf>)
- **Artigo** net.js - Um pacote Node.JS para trabalhar com TCP (<https://blog.zerokol.com/2012/09/netjs-um-pacote-nodejs-para-trabalhar.html>)
- **Artigo** Node — Como manipular arquivos usando FS (<https://thihenos.medium.com/node-crud-arquivos-142e3fb1892d>)
- **Artigo** Como criar um servidor Web em Node.js com o módulo HTTP (<https://www.digitalocean.com/community/tutorials/how-to-create-a-web-server-in-node-js-with-the-http-module-pt>)
- **Artigo** O módulo de path do Node.js (<https://nodejs.reativa.dev/0047-node-module-path/index>)

Conteúdos Alura:**

- **Artigo** Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um Guia para iniciar (<https://www.alura.com.br/artigos/node-js>)

- **Artigo** Node.js: definição, características, vantagens e usos possíveis (<https://www.alura.com.br/artigos/node-js-definicao-caracteristicas-vantagens-usos>)
- **Podcast** Hipsters.tech: Node.js - Hipsters 19 (<https://cursos.alura.com.br/extra/hipsterstech/node-js-hipsters-199-a356>)
- **YouTube** Alura: O que é Node.js? | AluraMais (<https://www.youtube.com/watch?v=8VSTrZY8vwI>)
- **YouTube** Alura: Ecossistema Node.js (<https://www.youtube.com/watch?v=2hrlwQWx9w4>)
- **Curso** Formação JavaScript para back-end (<https://cursos.alura.com.br/formacao-js-backend>)
- **Curso** Curso Node.js: criando sua primeira biblioteca (<https://www.alura.com.br/curso-online-nodejs-criando-primeira-biblioteca>)

JavaScript - Callbacks e Promises:

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

Conteúdos

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

- **Artigo** Revolução no Node.js: adeus ao Axios e fetch API na versão 17.5.0 (<https://www.alura.com.br/artigos/revolucao-node-js-adeus-axios-fetch-api-versao-17-5-0>)

Conteúdos Alura:**

- **Artigo** Async/await no JavaScript: o que é e quando usar a programação assíncrona? (<https://www.alura.com.br/artigos/async-await-no-javascript-o-que-e-e-quando-usar>)
- **Site** Alura+: JavaScript assíncrono e Fetch (<https://cursos.alura.com.br/extra/alura-mais/javascript-assincrono-e-fetch-c93>)
- **Curso** Curso JavaScript: consumindo e tratando dados de uma API (<https://cursos.alura.com.br/course/javascript-consumindo-tratando-dados-uma-api>)

JavaScript - Manipulação de Erros:

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

Conteúdos

- **Site** MDN Web Docs: Controle de Fluxo e Manipulação de Erro (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Control_flow_and_error_handling#declara%C3%A7%C3%B5es_de_manipula%C3%A7%C3%A3o_de_fluxo_e_manipula%C3%A7%C3%A3o_de_erros)
- **Site** MDN Web Docs: Error (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Error)
- **Site** Microsoft TechNet: JavaScript: Tratamento de erros (<https://social.technet.microsoft.com/wiki/pt-br/contents/articles/32455.javascript-tratamento-de-erros.aspx>)
- **Artigo** Exceções e Erros em JavaScript (<http://gabrielprates.com/2017/06/02/excecoes-e-erros-em-js.html>)
- **Artigo** Criando Exceptions para impressionar no Teste Técnico (<https://dev.to/he4rt/criando-exceptions-para-impressionar-no-teste-tecnico-2nie>)

Conteúdos Alura:**

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

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=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>)

Node.js - 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.
- Aprender o que são testes unitários
- Aprender o que são testes de integração
- Trabalhar com frameworks de testes e suas ferramentas

- Criar mocks para seus testes
- Aprender os conceitos de BDD (Behavior-Driven Development) e TDD (Test-Driven Development)

Conteúdos

- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software! (<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Artigo** Anatomia de um teste em JavaScript (<https://gabrieluizramos.com.br/anatomia-de-um-teste-em-javascript>)
- **Artigo** Fundamentando mocks em JavaScript (<https://gabrieluizramos.com.br/fundamentando-mocks-em-javascript>)
- **Artigo** Testes assíncronos em JavaScript (<https://gabrieluizramos.com.br/testes-assincronos-em-javascript>)
- **Artigo** Testes Unitários em Node.js usando Chai e Mocha (https://medium.com/@matheuslves_45120/testes-unit%C3%A1rios-em-node-js-usando-chai-e-mocha-c415c9f84764)
- **Site** Jest - documentação oficial (<https://jestjs.io/pt-BR/>)

Conteúdos Alura:**

- **Site** Livro: JavaScript Assertivo (<https://www.casadocodigo.com.br/products/livro-javascript-assertivo>)
- **Curso** Curso Node.js: testes unitários e de integração (<https://cursos.alura.com.br/course/nodejs-testes-unitarios-integracao>)
- **Curso** Curso Node.js: implementando testes em uma API Rest (<https://cursos.alura.com.br/course/node-js-implementando-testes-api-rest>)

Nível 2

SOLID:

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

Conteúdos

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

Conteúdos Alura:**

- **Artigo** SOLID: o que é e quais os 5 princípios da Programação Orientada a Objetos (POO) (<https://www.alura.com.br/artigos/solid>)
- **Podcast** SOLID: Código bom e bonito - Hipsters Ponto Tech 219 (<https://www.alura.com.br/podcast/hipsterstech-solid-codigo-bom-e-bonito-hipsters-ponto-tech-219-a649>)
- **YouTube** Alura: Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **Curso** Curso C#: aplique princípios SOLID (<https://cursos.alura.com.br/course/csharp-aplique-principios-solid>)
- **Curso** Curso Boas práticas de programação: melhore o código de uma API Java (<https://cursos.alura.com.br/course/boas-praticas-programacao-melhora-codigo-api-java>)
- **Curso** Curso SOLID com PHP: princípios da programação orientada a objetos (<https://cursos.alura.com.br/course/solid-php-principios-orientacao-a-objetos>)
- **Curso** Curso iOS: escrevendo código de qualidade com SOLID em Swift (<https://www.alura.com.br/curso-online-ios-escrevendo-codigo-qualidade-solid-swift>)
- **Curso** SOLID com TypeScript: aplicando boas práticas em orientação a objetos (<https://www.alura.com.br/curso-online-solid-typescript-boas-praticas-orientacao-objetos>)
- **Curso** Formação Boas práticas em C# (<https://cursos.alura.com.br/formacao-boas-praticas-c-sharp>)
- **Curso** Formação Boas práticas em PHP (<https://cursos.alura.com.br/formacao-boas-praticas-php>)
- **Curso** Formação Boas práticas em Java (<https://cursos.alura.com.br/formacao-boas-praticas-java>)
- **Site** Livro Casa do Código: Desbravando SOLID - Práticas avançadas para códigos de qualidade em Java moderno (<https://www.casadocodigo.com.br/products/livro-desbravando-solid>)
- **YouTube** CodandoTV(Rods): SOLID e o 'D' do Principio da Inversão de Dependência usando a Injeção de Dependência para ajudar (<https://youtu.be/xpcCDCH5k4Y>)

Node.js - Express:

- O Express é um framework de aplicações web Node.js flexível que fornece um conjunto robusto de recursos para aplicativos da web e mobile.
- Utilizar o framework Express para criação de APIs REST com Node.js
- Gerenciar requisições de diferentes verbos HTTP em diferentes URLs
- Definir a porta a ser usada para conexão e a localização dos modelos que são usados para renderizar a resposta
- Criar manipuladores de rotas usando o método 'router'
- Conhecer a bibliotecas 'Router' e seus verbos HTTP, como 'get', 'post', 'put', etc.
- Definir endpoints com route paths

Conteúdos

- **Site** Express.js - documentação oficial (<https://expressjs.com/pt-br/>)
- **Site** Express.js - documentação oficial - Roteamento (<https://expressjs.com/pt-br/guide/routing.html>)
- **Artigo** MDN Web Docs: Introdução Express/Node.js (https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction)

- **Artigo** Tutorial Node.js — Como usar o Express.js (<https://pedrompinto.medium.com/tutorial-node-js-como-usar-o-express-js-7d3027f4f57b>)
- **Artigo** Criando uma API Node em 10 passos com Express.js (<https://medium.com/xp-inc/criando-uma-api-node-em-10-passos-com-express-js-52b2d612a8a9>)
- **Artigo** Entendendo o express.Router e usando módulos de forma simples (<https://medium.com/owinteractive/entendendo-o-express-router-e-usando-m%C3%B3dulos-de-forma-simples-24ca6723fe91>)

Conteúdos Alura:**

- **Curso** Formação Node.js com Express (<https://www.alura.com.br/formacao-node-js-express>)

Node.js - ORM:

- Object-Relational Mapping (ORM), em português, mapeamento objeto-relacional, é uma técnica utilizada para fazer o mapeamento entre sistemas orientados a objetos e bancos de dados relacionais, onde as tabelas do banco de dados são representadas em classes e os registros das tabelas seriam instâncias dessas classes.
- Entender o que são ORMs e para que são utilizados
- Conhecer o SQL e seus gerenciadores de bancos de dados
- Trabalhar com o Sequelize, um ORM para uso com Node.js
- Conhecer outros ORMs Node.js, como o Prisma

Conteúdos

- **Site** Sequelize - documentação oficial (<https://sequelize.org/>)
- **Artigo** O que é ORM - Mapeamento objeto-relacional (<https://dev.to/biosbug/o-que-e-orm-mapeamento-objeto-relacional-2m8n>)
- **YouTube** Josias Pereira: Iniciando API com NodeJs e Sequelize ORM (<https://www.youtube.com/watch?v=mpKXSe08ygA>)
- **YouTube** Nicolas Teófilo: O melhor ORM para Node.js: Prisma (<https://www.youtube.com/watch?v=tF-ly2sB3hU>)

Conteúdos Alura:**

- **YouTube** Alura+: O que é ORM (<https://www.youtube.com/watch?v=x39vqeBTUmE&t=118s>)
- **Curso** Curso ORM com Node.js: desenvolvendo uma API com Sequelize e SQLite (<https://cursos.alura.com.br/course/orm-node-js-desenvolvendo-api-sequelize-sqlite>)
- **Curso** Curso ORM com Node.js: avançando nas funcionalidades do Sequelize (<https://cursos.alura.com.br/course/orm-node-js-funcionalidades-sequelize>)

Node.js - Autenticação e Tokens:

- O JWT é um método criado pelo padrão RFC 7519 que representa a comunicação segura entre duas partes. Esse token é composto por três partes: cabeçalho, carga útil e assinatura.
- Construir um sistema de autenticação usando tokens
- Entender o funcionamento do JSON Web Token (JWT)
- Construir uma allowlist para guardar tokens opacos
- Implementar métodos de atualização de tokens

Conteúdos

- **Artigo** Autenticação, autorização e gerenciamento de identidade e acesso (<https://gabrieluizramos.com.br/autenticacao-autorizacao-e-gerenciamento-de-identidade-e->

[acesso](#))

- **Artigo** NodeJS: Autenticação com JWT (<https://medium.com/@filipefilipe/nodejs-autentica%C3%A7%C3%A3o-com-jwt-6e274fb205dc>)
- **Artigo** JWT — Usando tokens para comunicação eficiente (<https://medium.com/trainingcenter/jwt-usando-tokens-para-comunica%C3%A7%C3%A3o-eficiente-cf0551c0dd99>)

Conteúdos Alura:**

- **Curso** Curso Node.js: Criptografia e tokens JWT (<https://www.alura.com.br/curso-online-nodejs-criptografia-tokens-jwt>)

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=SpWFDVgS9k>)
- **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>)

Nível 3

Nest.js - Fundamentos:

- NestJS é um framework Node com total suporte a TypeScript e que roda sobre frameworks HTTP como expressJS ou Fastify. Ele utiliza diversos elementos de programação orientada a objetos e uma série de funcionalidades do TypeScript.
- Aprender o que é o NestJS e por que é utilizado
- Utilizar recursos específicos do NestJS, como providers, módulos e controllers
- Desenvolver APIs usando o NestJS
- Usar a Interface de Linha de Comando (CLI) do Nest.js

Conteúdos

- **Site** NestJS - Documentação oficial (em inglês) (<https://docs.nestjs.com/>)
- **Artigo** Nest.js—backend com cara de Angular sobre Node.js (<https://medium.com/codigorefinado/nest-js-typescript-pra-node-js-e-express-e5042b77cae0>)
- **Artigo** O que é NestJS (<https://eduardo-diogo.medium.com/o-que-%C3%A9-nestjs-9287d9a8dcd3>)
- **Artigo** Construindo uma API com NestJS, PostgreSQL e Docker — Parte 1 (<https://medium.com/@iago.maiasilva/construindo-uma-api-com-nestjs-postgresql-e-docker-parte-1-criando-nosso-primeiro-endpoint-248d4b8ecc9c>)
- **YouTube** Código Falado: Por que Nest.js? (<https://www.youtube.com/watch?v=ledmemIH9jA>)
- **YouTube** Paulo Salvatore: NestJS, o melhor framework backend de Node.js (<https://www.youtube.com/watch?v=bAH4nBb1Nfc>)

Conteúdos Alura:**

- **YouTube** Alura: Conhecendo NestJS | AluraMais (<https://www.youtube.com/watch?v=YHpG6t9IoW8>)
- **Curso** Nest.js: criando uma API Restful (<https://cursos.alura.com.br/course/nestjs-criando-api-resftul>)
- **Curso** Nest.js: adicionando funcionalidades com Redis, JWT e logging (<https://cursos.alura.com.br/course/nest-js-funcionalidades-redis-jwt-logging>)
- **Curso** Formação Next.js: autenticação e gestão de sessão (<https://www.alura.com.br/formacao-next-js-autenticacao-gestao-sessao>)
- **Artigo** Segurança do Next.js 14+ com Middlewares (<https://www.alura.com.br/artigos/next-js-middlewares>)

Arquitetura de Microserviços:

- Microserviços são uma abordagem de arquitetura na qual o software consiste de pequenos serviços independentes que se comunicam entre si e são organizados de acordo com seus domínios de negócio.

- Aprender o conceito de arquitetura planejada para microsserviços
- Realizar a comunicação usando APIs
- Melhorar a escalabilidade de um sistema

Conteúdos

- **Artigo** Google Cloud: O que é a arquitetura de microsserviços? (<https://cloud.google.com/learn/what-is-microservices-architecture?hl=pt-br>)
- **Artigo** Amazon: O que são microsserviços? (<https://aws.amazon.com/pt/microservices/>)
- **YouTube** Otavio Lemos: Microsserviços | O mínimo que você precisa saber (<https://www.youtube.com/watch?v=eN8dFfTrEtQ>)
- **YouTube** Douglas Mugnos: O que é arquitetura de microsserviços? Como ela funciona? (https://www.youtube.com/watch?v=_Oyy5PFOlcU)

Conteúdos Alura:**

- **Podcast** Hipsters.tech: TechGuide - Microsserviços - Hipsters Ponto Tech #349 (<https://www.hipsters.tech/techguide-microservicos-hipsters-ponto-tech-349/>)
- **YouTube** Alura: O que são Microsserviços? (<https://www.youtube.com/watch?v=jSnL0oGjQ80>)
- **Podcast** Hipsters.tech: Microserviços - Hipsters 17 (<https://www.alura.com.br/podcast/microservicos-hipsters-17-a569>)
- **Podcast** Hipsters.tech: Microsserviços na Caelum - Hipsters On The Road 6 (<https://www.alura.com.br/podcast/microservicos-na-caelum-hipsters-on-the-road-6-a436>)
- **Podcast** Hipsters.tech: Microsserviços autônomos na Usabilla - Hipsters On The Road 10 (<https://www.alura.com.br/podcast/microservicos-autonomos-na-usabilla-hipsters-on-the-road-10-a424>)
- **Site** Alura+: O que são Microsserviços? (<https://cursos.alura.com.br/extra/alura-mais/o-que-sao-microservicos--c699>)
- **Site** Alura+: Tipos de Microservices (<https://cursos.alura.com.br/extra/alura-mais/tipos-de-microservices-c698>)
- **Curso** Curso Microsserviços: padrões de projeto (<https://cursos.alura.com.br/course/microservicos-padroes-projeto>)
- **Curso** Curso Microsserviços: explorando os conceitos (<https://cursos.alura.com.br/course/fundamentos-microservicos-aprofundando-conceitos>)
- **Curso** Curso Microsserviços na prática: entendendo a tomada de decisões (<https://cursos.alura.com.br/course/Microservicos-pratica-tomada-decisoes>)

Contêineres:

- Os contêineres são pacotes de software que contêm todos os elementos necessários para serem executados em qualquer ambiente. Gerenciamento de contêineres é uma área crucial na computação em nuvem e DevOps, que envolve o uso de tecnologias para automatizar o processo de criação, implantação, escalonamento e monitoramento de contêineres. Contêineres são unidades de software padronizadas que permitem aos desenvolvedores empacotar todas as dependências de um aplicativo (código, bibliotecas, configurações, etc.) em um único pacote. Isso permite que o aplicativo seja executado de forma consistente em qualquer ambiente de infraestrutura.
- A tecnologia de contêineres, como exemplificada pelo Docker, fornece um ambiente consistente e portátil para desenvolvimento, teste e implantação de aplicativos, o que é vital para o trabalho eficiente de engenharia de dados. Além disso, o Kubernetes, um sistema de orquestração de contêineres, permite o gerenciamento, a automação e a escalabilidade de aplicações baseadas

em contêineres em ambientes de produção. Dominar esses conceitos e tecnologias possibilita a engenheiros de dados construir e manter pipelines de dados eficientes e confiáveis.

- O Kubernetes (também conhecido como k8s ou kube) é uma plataforma de orquestração de containers open source que automatiza grande parte dos processos manuais necessários para implantar, gerenciar e escalar aplicações em containers.
- Isolar seu software para funcionar independentemente
- Implantar software em clusters
- Modularizar seu sistema em pacotes menores
- Conhecer a plataforma Docker
- Conhecer Kubernetes

Conteúdos

- **Site** IBM: Contêineres (<https://www.ibm.com/br-pt/cloud/learn/containers>)
- **Site** HPE: O que são contêineres? (<https://www.hpe.com/br/pt/what-is/containers.html>)
- **Site** Google Cloud: O que são contêineres? (<https://cloud.google.com/learn/what-are-containers?hl=pt-br>)
- **Site** Azure: O que é um contêiner? (<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-container/>)
- **Artigo** Red Hat: O que é Kubernetes (<https://www.redhat.com/pt-br/topics/containers/what-is-kubernetes>)
- **Artigo** Azure: O que é Kubernetes? (<https://azure.microsoft.com/pt-br/topic/what-is-kubernetes/>)
- **Artigo** Kubernetes - Guia de Referência Gratuito (<https://renatogroffe.medium.com/kubernetes-guia-de-refer%C3%A2ncia-gratuito-a2c1be8e7593>)
- **YouTube** Código Fonte TV: Containers // Dicionário do Programador (<https://www.youtube.com/watch?v=-pUZBovqRcU>)
- **YouTube** Código Fonte TV: Kubernetes // Dicionário do Programador (<https://www.youtube.com/watch?v=mVL0nOM3AGo>)
- **YouTube** Insight Lab: Revisão Docker - Curso de Introdução ao Kubernetes (<https://www.youtube.com/watch?v=bcRArpK00OU>)
- **YouTube** Pisani da Arch: Kubernetes - entenda e comece a arquitetar agora mesmo (<https://www.youtube.com/watch?v=XjaqTqm7g58>)

Conteúdos Alura:**

- **Artigo** O que é Kubernetes (<https://www.alura.com.br/artigos/o-que-e-kubernetes>)
- **Artigo** Criando volumes com Docker (<https://www.alura.com.br/artigos/criando-volumes-com-docker>)
- **Artigo** Docker Compose para compor uma aplicação (<https://www.alura.com.br/artigos/compondo-uma-aplicacao-com-o-docker-compose>)
- **YouTube** Alura: O que são Containers? (https://www.youtube.com/watch?v=jv4_sLIGOS0)
- **YouTube** Alura: Containers, Docker e Kubernetes (com Giovanni Bassi) (<https://www.youtube.com/watch?v=wxLvMxzc1Q>)
- **Podcast** Hipsters.tech: Containers e Docker - Hipsters 75 (<https://www.alura.com.br/podcast/hipsterstech-containers-e-docker-hipsters-75-a511>)
- **Podcast** Hipsters.tech: Kubernetes - Hipsters 177 (<https://www.alura.com.br/podcast/kubernetes-hipsters-177-a389>)

- **Podcast** Hipsters.tech: Microsserviços e Kubernetes na Americanas.com - Hipsters On The Road 56 (<https://www.alura.com.br/podcast/microsservicos-e-kubernetes-na-americanas-com-hipsters-on-the-road-56-a857>)
- **Curso** Formação Orquestração de containers com Kubernetes (<https://www.alura.com.br/formacao-kubernetes>)
- **Curso** Curso Docker: criando e gerenciando containers (<https://www.alura.com.br/curso-online-docker-criando-gerenciando-containers>)
- **Curso** Curso Kubernetes: Pods, Services e ConfigMaps (<https://www.alura.com.br/curso-online-kubernetes-pods-services-configmap>)
- **Curso** Curso Kubernetes: Deployments, Volumes e Escalabilidade (<https://www.alura.com.br/curso-online-kubernetes-deployments-volumes-escalabilidade>)
- **Curso** Curso Google Kubernetes Engine (<https://www.alura.com.br/curso-online-google-kubernetes-engine-primeiros-passos>)
- **Curso** Curso Kubernetes na Digital Ocean: gerenciando aplicações containerizadas (<https://www.alura.com.br/curso-online-kubernetes-digital-ocean>)
- **Curso** Curso Amazon EKS: gerenciando aplicações containerizadas com Kubernetes (<https://www.alura.com.br/curso-online-amazon-eks-kubernetes>)
- **Curso** Curso Azure: gerencie Kubernetes com AKS e ACR (<https://www.alura.com.br/curso-online-azure-gerencie-kubernetes-com-aks-acr>)

WebSockets:

- WebSocket é uma tecnologia que permite a comunicação bidirecional por canais full-duplex sobre um único soquete Transmission Control Protocol (TCP). Ele é projetado para ser executado em browsers e servidores web que suportem o HTML5, mas pode ser usado por qualquer cliente ou servidor de aplicativos.
- Conhecer o protocolo WebSocket e seu uso na comunicação cliente-servidor
- Aprender sobre os diversos usos do WebSocket na web
- Criar aplicações que utilizam WebSockets com as APIs e bibliotecas do Node.js

Conteúdos

- **Site** Documentação: Escrevendo aplicações cliente WebSocket (https://developer.mozilla.org/pt-BR/docs/Web/API/WebSockets_API/Writing_WebSocket_client_applications)
- **Artigo** A necessidade de criar um servidor WebSocket (<https://medium.com/tableless/a-necessidade-de-criar-um-servidor-web-socket-63a7c56a1698>)
- **Artigo** Como o JavaScript funciona: Aprofundando em WebSockets e HTTP/2 com SSE (<https://medium.com/reactbrasil/como-o-javascript-funciona-aprofundando-em-websockets-e-http-2-com-sse-como-escolher-o-caminho-d4639995ef85>)
- **YouTube** Filipe Deschamps: WebSockets na prática é MUITO poderoso (<https://www.youtube.com/watch?v=bQ7NNSyGV2U>)

Conteúdos Alura:**

- **Artigo** Entendendo o socket no Python criando um bot de IRC (<https://www.alura.com.br/artigos/entendendo-socket-no-python-criando-um-bot-de-irc>)
- **YouTube** Alura: Entendendo WebSockets (<https://www.youtube.com/watch?v=RwUbUnPdWqs>)
- **Curso** Curso WebSockets: implemente comunicações em tempo real com Socket.IO e MongoDB (<https://cursos.alura.com.br/course/websockets-comunicacoes-tempo-real-socket-io-mongodb>)

- **Curso** Curso WebSockets: implemente autenticação e avance no Socket.IO (<https://cursos.alura.com.br/course/websockets-implemente-autenticacao-avance-socket-io>)
- **Curso** Curso PHP Reativo: programação assíncrona em tempo real (<https://cursos.alura.com.br/course/php-reativo-programacao-assincrona-tempo-real>)

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>)

Habilidade Auxiliar: Infraestrutura e boas práticas

Git e GitHub - Fundamentos:

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

Conteúdos

- **Site** GitHub Documentação (<https://docs.github.com/pt>)
- **Site** GitHub Pages Documentação (<https://docs.github.com/pt/pages/getting-started-with-github-pages/about-github-pages>)
- **Site** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)
- **Site** Dangit, Git!?! (<https://dangitgit.com/>)
- **YouTube** Rafaella Ballerini: O que é Git e GitHub? - definição e conceitos importantes 1/2 (<https://www.youtube.com/watch?v=DqTITcMq68k>)
- **YouTube** Rafaella Ballerini: Como usar Git e GitHub na prática! - desde o primeiro commit até o pull request! 2/2 (<https://www.youtube.com/watch?v=UBAX-13g8OM>)
- **YouTube** Mario Souto - Dev Soutinho: Git: Entendendo de vez como funciona do melhor e mais visual jeito possível (<https://www.youtube.com/watch?v=4-tfJ-ZyAOQ>)
- **YouTube** Mario Souto - Dev Soutinho: Como colocar seu projeto no ar DE GRAÇA via GitHub! | Hospedagem com GitHub Pages (https://www.youtube.com/watch?v=BU-w2_Aae54)
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git (https://youtu.be/MqoqPzjQyCY?si=VI_mT8EowuemptmU)
- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub (https://youtu.be/v9ZM2PVzctM?si=49ah_HRPiwpSEX4A)

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>)

Design Patterns:

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

Conteúdos

- **Artigo** Design Patterns — O que são e quais os benefícios? (<https://djesusnet.medium.com/design-patterns-gof-o-que-s%C3%A3o-e-quais-os-benef%C3%ADcios-9cd0cfdd6ebf>)
- **Artigo** Design Patterns (<https://medium.com/xp-inc/design-patterns-727494af001d>)
- **YouTube** Matheus Castiglioni: O que são design patterns (padrões de projetos) e para que servem? (<https://www.YOUTUBE.com/watch?v=KNWd1aWtsMw>)
- **YouTube** RinaldoDev: Design Patterns - o que são? Por que aprender a utilizar? (<https://www.YOUTUBE.com/watch?v=AWOf6Wo6gtg>)

Conteúdos Alura:**

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

Linha de comando - Fundamentos:

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

Conteúdos

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

Conteúdos Alura:**

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

Cloud - Fundamentos:

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

Conteúdos

- **Artigo** Computação em nuvem (<https://medium.com/sysadminas/computa%C3%A7%C3%A3o-em-nuvem-515930304cf9>)
- **Artigo** O que é cloud? (<https://gabriel-faraday.medium.com/o-que-%C3%A9-cloud-991109e708c6>)
- **YouTube** Gabs Ferreira: Por que investir e estudar cloud? (<https://www.youtube.com/watch?v=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>)

Habilidade Auxiliar: Front-end

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=IIF6vD-RjJE>)
- **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>)

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=gnQ_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=B.Jpf9VD7A5g>)
- **YouTube** Canal do Consultor: Introdução ao SEO: O que é? Para que serve? Diferenças entre SEO e SEM (<https://www.youtube.com/watch?v=hn0DDvCB5eg>)

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>)