

# Kotlin

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>)

### **Kotlin - Fundamentos:**

- Kotlin é uma linguagem de programação multiplataforma, orientada a objetos, funcional e estaticamente tipada. Ela compila para a máquina virtual Java e também pode ser traduzida para a linguagem JavaScript e compilada para código nativo (via LLVM). É a linguagem oficial do sistema Android da Google.
- Entender a sua sintaxe
- Conhecer os tipos primitivos
- Declarar e usar variáveis e constantes
- Usar estruturas condicionais (if, else)
- Usar estruturas de repetição e laços (while, for)
- Usar funções, passando parâmetros e argumentos
- Implementando métodos e reutilizando eles
- Null Safety (Elimine o perigo de referências nulas)
- Exceptions e Throwables
- Convenções de código
- Manipular Coleções, arrays e listas
- Orientação a Objetos com Kotlin (Properties, Data class, Companion Objects, Delegation)
- Receber dados de uma API

## Conteúdos

- **Site** Documentação Kotlin: começando com Kotlin (inglês) (<https://kotlinlang.org/docs/getting-started.html>)
- **Site** Documentação Kotlin: Tipos Básicos e Kotlin (inglês) (<https://kotlinlang.org/docs/basic-types.html>)
- **Site** Documentação Kotlin: Sintaxe básica do Kotlin(inglês) (<https://kotlinlang.org/docs/basic-syntax.html>)
- **Site** Documentação Kotlin: Convenções de código do Kotlin (inglês) (<https://kotlinlang.org/docs/coding-conventions.html>)
- **Site** Explorando a documentação: Números em Kotlin (<https://dev.to/kotlinautas/explorando-a-documentacao-numeros-em-kotlin-15go>)
- **YouTube** Kotlin // Dicionário do Programador (<https://youtu.be/BfjRYBN7Ur8>)

## Conteúdos Alura:

- **Artigo** Linguagem Kotlin: o que é, para que serve e um Guia para aprender (<https://www.alura.com.br/artigos/kotlin>)
- **Podcast** Case Contabilizei: Kotlin - Hipsters Ponto Tech (<https://www.hipsters.tech/case-contabilizei-kotlin-hipsters-ponto-tech-310/>)
- **Curso** Formação Kotlin (<https://cursos.alura.com.br/formacao-kotlin>)
- **Desafio** 7 Days of Code: Kotlin (<https://7daysofcode.io/matricula/kotlin>)

## 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=jpuJ1qrIuoU>)

- **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>)

## Estruturas de Dados:

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

## Conteúdos

- **Artigo** Estrutura de dados (<https://medium.com/php-word/estrutura-de-dados-6afb386a6fc1>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 1 (<https://medium.com/@PaoloProdossimoLopes/01-estrutura-de-dados-e-algoritmo-9cb6a9b14aa>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 2 (<https://medium.com/@PaoloProdossimoLopes/02-estrutura-de-dados-e-algoritmo-deba0b384a0f>)

- **YouTube** Mario Souto - Dev Soutinho: Estruturas de dados com JavaScript (<https://www.youtube.com/watch?v=MweeZn1rR8s>)
- **YouTube** Fabio Akita: Árvores (<https://www.youtube.com/watch?v=9GdesxWtOgs>)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Lista Encadeada ([https://www.youtube.com/watch?v=RW0oD2L\\_tSg](https://www.youtube.com/watch?v=RW0oD2L_tSg))
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Filas ([https://www.youtube.com/watch?v=RdoLTgJSV\\_c](https://www.youtube.com/watch?v=RdoLTgJSV_c))
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Pilhas (<https://www.youtube.com/watch?v=ZdU4wMyiTSs>)

### Conteúdos Alura:

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

## Kotlin - 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.
- Tratar exceções pré-definidas
- Uso de 'try' e 'catch'
- Criar exceções específicas
- Fazer o processo de Debug

## Conteúdos

- **Site** Documentação Kotlin: Exceções (inglês)  
(<https://kotlinlang.org/docs/exceptions.html>)
- **Artigo** Documentação Kotlin: Kotlin try catch (inglês)  
(<https://www.javatpoint.com/kotlin-try-catch>)
- **Artigo** Tratamento de exceções em Kotlin (inglês)  
(<https://www.baeldung.com/kotlin/exception-handling>)
- **Artigo** Diferença entre runCatching e try/finally em Kotlin (inglês)  
(<https://www.baeldung.com/kotlin/runcatching-vs-try-finally>)
- **Artigo** Tratamento de exceções em Kotlin com exemplos  
(<https://acervolima.com/tratamento-de-excecoes-em-kotlin-com-exemplos/>)

## Kotlin - Collections:

- Uma coleção representa um grupo de objetos, conhecidos como seus elementos. Eles são como recipientes que agrupam vários itens em uma única unidade. Algumas coleções permitem a duplicação de elementos e outras não. Algumas são ordenadas e outras não ordenadas.
- Aprender os usos e diferenças entre List, Set e Map
- Aprender os usos e diferenças entre read only e mutable

## Conteúdos

- **Site** Documentação Kotlin: Visão geral das coleções (inglês)  
(<https://kotlinlang.org/docs/collections-overview.html>)
- **Site** Documentação Kotlin: Coleções (inglês)  
(<https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.collections/-collection/>)
- **Site** Coleções em Kotlin  
(<https://www.codinggame.com/playgrounds/70522/fundamentos-de-kotlin/colecoes>)
- **Artigo** Guia de coleções Kotlin (inglês)  
(<https://www.baeldung.com/kotlin/kotlin-collection-guide>)

## Kotlin - Extensões:

- Como criar e usar extensões simples para melhorar a compreensão dos fundamentos da linguagem.

## Conteúdos

- **Site** Documentação Kotlin: Extensions (inglês)  
(<https://kotlinlang.org/docs/extensions.html>)

# Nível 2

## JVM:

- Máquina virtual Java (em inglês, Java Virtual Machine, JVM) é um programa que carrega e executa os aplicativos Java, convertendo os bytecodes em código executável de máquina. A JVM é responsável pelo gerenciamento dos aplicativos, à medida que são executados. Graças à máquina virtual Java, os programas escritos em Java podem funcionar em qualquer plataforma de hardware e software que possua uma versão da JVM, tornando assim essas aplicações independentes da plataforma onde funcionam.
- Entender como funciona a máquina virtual do Java



## Conteúdos

- **Site** Java Virtual Machine (JVM)  
(<https://sites.google.com/site/jvmtotal/home/conceitos/java-virtual-machine>)
- **Site** Wikipédia: Máquina virtual Java  
([https://pt.wikipedia.org/wiki/M%C3%A1quina\\_virtual\\_Java](https://pt.wikipedia.org/wiki/M%C3%A1quina_virtual_Java))
- **Artigo** JVM por dentro (<https://vepo.medium.com/jvm-por-dentro-553bab89a027>)
- **YouTube** Descompila: Entendendo a JVM na prática  
(<https://www.youtube.com/watch?v=4Qd16jfJDaY>)
- **YouTube** Lucas Silva: O que é JVM, JRE e JDK?  
(<https://www.youtube.com/watch?v=BNrf3yT8lcU>)
- **YouTube** Giuliana Bezerra: Entenda o que é WORA, JVM, JDK e JRE - Ep. 1  
(<https://youtu.be/U6Gh6HYyI6c>)

## Conteúdos Alura:

- **Artigo** JVM: conhecendo o processo de execução de código  
(<https://www.alura.com.br/artigos/jvm-conhecendo-processo-execucao-de-codigo>)
- **Podcast** Hipsters.tech: E o futuro do Java! – Hipsters 27  
(<https://www.hipsters.tech/e-o-futuro-do-java-hipsters-27/>)
- **YouTube** Alura: Java é lento? JIT e Otimizações na Java Virtual Machine  
([https://www.youtube.com/watch?v=rrhO\\_P0IM3s](https://www.youtube.com/watch?v=rrhO_P0IM3s))

## Kotlin - Gerenciamento de memória:

- O gerenciamento de memória é o processo de alocação e desalocação de objetos, chamado de gerenciamento de memória. Kotlin faz o gerenciamento de memória automaticamente. Kotlin usa um sistema de gerenciamento automático de memória chamado de Garbage Collector (coletor de lixo). Assim, não somos obrigados a implementar a lógica de gerenciamento de memória em nossa aplicação.
- Entender como funciona a memória e seu gerenciamento em Kotlin

- Entender como funciona a memória o Garbage Collector

## Conteúdos

- **Site** Documentação Kotlin: Gerenciamento de memória nativa no Kotlin (inglês) (<https://kotlinlang.org/docs/native-memory-manager.html>)
- **Artigo** Gerenciamento de Memória em Kotlin: Compreendendo o Garbage Collection (<https://www.linkedin.com/pulse/gerenciamento-de-mem%C3%B3ria-em-kotlin-compreendendo-o-de-oliveira/?originalSubdomain=pt>)
- **YouTube** Dias de Dev: Estratégias de gestão de memória - O que é Garbage Collector ([https://www.youtube.com/watch?v=SZqp3\\_Mw9zk](https://www.youtube.com/watch?v=SZqp3_Mw9zk))

## Spring Framework:

- O Spring é um framework open source para a plataforma Java. Trata-se de um framework não intrusivo, baseado nos padrões de projeto (design patterns) de inversão de controle (IoC) e injeção de dependência. No Spring o contêiner se encarrega de "instanciar" classes de uma aplicação Java e definir as dependências entre elas através de um arquivo de configuração em formato XML, inferências do framework, o que é chamado de auto-wiring ou ainda anotações nas classes, métodos e propriedades. Dessa forma, o Spring permite o baixo acoplamento entre classes de uma aplicação orientada a objetos.
- Entender o conceito de Injeção de Dependências
- Entender o padrão MVC
- Usar o Spring Data para manipular dados

## Conteúdos

- **Site** Documentação Spring Framework (inglês) (<https://spring.io/projects/spring-framework>)
- **Artigo** Spring Framework: existem muitos, mas para que cada um serve? (<https://arthur-almeida.medium.com/spring-framework-existem-muitos-mas-para-que-cada-um-serve-b1773bcb69f0>)

- **Artigo** MVC não é Arquitetura! (<https://arthur-almeida.medium.com/mvc-n%C3%A3o-%C3%A9-arquitetura-1f5cca077f18>)
- **YouTube** Michelli Brito: Spring MVC (<https://www.youtube.com/watch?v=2RSrcKDqWyo>)
- **YouTube** Michelli Brito: O que é um bean em Spring Framework (<https://www.youtube.com/watch?v=-PT-pXe-7UM>)
- **YouTube** Dev Eficiente: O que é e como funciona o Spring MVC? (<https://youtu.be/9Jf88iqKPPU>)

### Conteúdos Alura:

- **Artigo** Spring: Conheça esse framework Java (<https://www.alura.com.br/artigos/spring-conheca-esse-framework-java>)
- **Artigo** Primeiros passos com o Spring Framework (<https://www.alura.com.br/artigos/primeiros-passos-spring>)
- **Podcast** Hipsters.tech: Spring Framework – Hipsters 87 (<https://www.hipsters.tech/spring-framework-hipsters-87/>)
- **YouTube** Alura: O que é Spring Framework? (<https://www.youtube.com/watch?v=5XPojnx9bb8>)
- **YouTube** Alura: Melhores práticas com Spring Framework ([https://www.youtube.com/watch?v=xDuPI3pA\\_R8](https://www.youtube.com/watch?v=xDuPI3pA_R8))
- **Curso** Formação Java e Microsserviços com Spring e RabbitMQ (<https://cursos.alura.com.br/formacao-java-microsservicos>)
- **Curso** Formação Java Web: crie aplicações usando Spring Boot (<https://www.alura.com.br/formacao-java-web-spring-boot>)
- **Site** Livro da Casa do Código: Vire o jogo com Spring Framework (<https://www.casadocodigo.com.br/products/livro-spring-framework>)
- **Desafio** 7 Days of Code: Spring (<https://7daysofcode.io/matricula/spring>)

### Spring Boot:

- O Spring Boot é um framework de código aberto baseado em Java usado para criar microserviços com o Spring Framework. Ele é usado para construir aplicações Spring independentes e prontas para produção.

- Criar aplicações Spring standalone
- Usar os servidores HTTP embutidos

## Conteúdos

- **Site** Documentação Spring Boot (inglês) (<https://spring.io/projects/spring-boot>)
- **Artigo** EngDB: Como o Spring Boot está ajudando os desenvolvedores? (<https://blog.engdb.com.br/spring-boot/>)
- **Artigo** Porque você deve usar Spring Boot (<https://medium.com/@duduxss3/porque-voc%C3%AA-deve-usar-spring-boot-a22f75da0955>)
- **Artigo** Resumo Curso Spring Boot API REST: Construa uma API - Alura ([https://www.linkedin.com/pulse/resumo-curso-spring-boot-api-rest-construa-uma-alura-maikel-souza-1f/?trk=articles\\_directory&originalSubdomain=pt](https://www.linkedin.com/pulse/resumo-curso-spring-boot-api-rest-construa-uma-alura-maikel-souza-1f/?trk=articles_directory&originalSubdomain=pt))
- **YouTube** Michelli Brito: Iniciando uma aplicação Spring Boot (<https://www.youtube.com/watch?v=OHn1jLHGptw>)
- **YouTube** Michelli Brito: API REST com Spring Boot (<https://www.youtube.com/watch?v=bpBRFNKg8k4&list=PL8ilphQOyG-D2FP9wkg12AavzmVRWEcnJ>)
- **YouTube** Dev Multitask: 5 coisas para saber de Spring ANTES de cair no Spring Boot! (<https://youtu.be/4pj1EPgacg8>)

## Conteúdos Alura:

- **YouTube** Alura: Criando um projeto com Spring Boot em 5 minutos (<https://www.youtube.com/watch?v=-U9QpXcCh10>)
- **Curso** Formação Kotlin e Spring Boot (<https://cursos.alura.com.br/formacao-kotlin-spring-boot>)

## Kotlin - Coroutines:

- A programação assíncrona ou sem bloqueio é uma parte importante do cenário de desenvolvimento. Ao criar aplicativos do lado do servidor,

desktop ou dispositivos móveis, é importante fornecer uma experiência que não seja apenas fluida do ponto de vista do usuário, mas também escalável quando necessário.

## Conteúdos

- **Site** Documentação Kotlin: Coroutines (inglês)  
(<https://kotlinlang.org/docs/coroutines-overview.html>)
- **Site** Documentação Kotlin: Coroutines guide  
(<https://kotlinlang.org/docs/coroutines-guide.html>)
- **Site** Introdução a coroutines em Kotlin (inglês)  
(<https://www.baeldung.com/kotlin/coroutines>)

## Conteúdos Alura:

- **Site** Tratamento de erros e cancelamento de Coroutines do Kotlin  
(<https://cursos.alura.com.br/extra/alura-mais/tratamento-de-erros-e-cancelamento-de-coroutines-do-kotlin-c86>)
- **Site** Websérie - Operações Assíncronas com Coroutines no Kotlin  
(<https://cursos.alura.com.br/extra/webseries-coroutines-kotlin/operacoes-assincronas-no-kotlin-1-apresentacao-w1225>)
- **Curso** Kotlin: aplicações resilientes e assíncronas  
(<https://cursos.alura.com.br/course/kotlin-aplicacoes-resilientes-assincronas>)

## Kotlin - Multiplatform:

- Como o Kotlin pode ser usado em várias plataformas, como JVM, JavaScript e código nativo.

## Conteúdos

- **Site** Documentação Kotlin: Kotlin Multiplataforma (inglês)  
(<https://kotlinlang.org/docs/multiplatform.html>)

## Nível 3

### Microserviços:

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

### Conteúdos

- **Artigo** Amazon: O que são microserviços?  
(<https://aws.amazon.com/pt/microservices/>)
- **Artigo** Dev.to: Microserviços com Spring Cloud: Introdução  
(<https://dev.to/womakerscode/microservicos-com-spring-cloud-introducao-3cn6>)
- **Site** Vídeo aula sobre microserviços em Java com Spring Boot  
(<https://gasparbarancelli.com/post/video-aula-sobre-microservicos-em-java-com-spring-boot?lang=pt>)
- **Artigo** Oracle: Desenvolva um Aplicativo Java RESTful baseado em Microserviços (<https://docs.oracle.com/pt-br/solutions/develop-microservice-java-app/index.html>)
- **YouTube** Otavio Lemos: Microserviços | O mínimo que você precisa saber  
(<https://www.youtube.com/watch?v=eN8dFfTrEtQ>)
- **YouTube** Douglas Mugnos: O que é arquitetura de microserviços? Como ela funciona? ([https://www.youtube.com/watch?v=\\_Oyy5PFOlcU](https://www.youtube.com/watch?v=_Oyy5PFOlcU))

### Conteúdos Alura:

- **YouTube** Alura: O que são Microserviços?  
(<https://www.youtube.com/watch?v=jSnLOoGjQ80>)
- **YouTube** Alura: Fundamentos de Microserviços  
(<https://www.youtube.com/watch?v=KMzE6y8ACnU>)
- **YouTube** Alura: Spring Boot acelere o desenvolvimento de microserviços  
(<https://www.youtube.com/watch?v=TX1NztPbvMw>)

- **Podcast** Hipsters.tech: Microserviços - Hipsters 17 (<https://cursos.alura.com.br/extra/hipsterstech/microservicos-hipsters-17-a569>)
- **Podcast** Hipsters.tech: Microserviços na Caelum - Hipsters On The Road 6 (<https://cursos.alura.com.br/extra/hipsterstech/microservicos-na-caelum-hipsters-on-the-road-6-a436>)
- **Podcast** Hipsters.tech: Microserviços autônomos na Usabilla - Hipsters On The Road 10 (<https://cursos.alura.com.br/extra/hipsterstech/microservicos-autonomos-na-usabilla-hipsters-on-the-road-10-a424>)
- **Site** Alura+: O que são Microserviços? (<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 Microserviços: padrões de projeto (<https://cursos.alura.com.br/course/microservicos-padroes-projeto>)
- **Curso** Curso Microserviços: explorando os conceitos (<https://cursos.alura.com.br/course/fundamentos-microservicos-aprofundando-conceitos>)
- **Curso** Curso Microserviços na prática: entendendo a tomada de decisões (<https://cursos.alura.com.br/course/Microservicos-pratica-tomada-decisoes>)
- **Curso** Curso Microserviços na prática: implementando com Java e Spring (<https://cursos.alura.com.br/course/microservicos-implementando-java-spring>)
- **Curso** Formação Java e Microserviços com Spring e RabbitMQ (<https://cursos.alura.com.br/formacao-java-microservicos>)
- **Site** Livro Casa do Código: Back-end Java - Microserviços, Spring Boot e Kubernetes (<https://www.casadocodigo.com.br/products/livro-backend-java>)

## 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](https://www.youtube.com/watch?v=jv4_sLIGOS0))
- **YouTube** Alura: Containers, Docker e Kubernetes (com Giovanni Bassi) (<https://www.youtube.com/watch?v=wxLvvMxzc1Q>)

- **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/microservicos-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>)

## Kotlin - Type Safe Builders:

- Criar construtores type-safe usando recursos avançados de Kotlin, para construir DSLs (Domain-Specific Languages) personalizadas.

### Conteúdos

- **Site** Documentação Kotlin: Type-safe builders (inglês) (<https://kotlinlang.org/docs/type-safe-builders.html>)
- **Site** Medium: Type-Safe Builders in Kotlin: The Secret Sauce of Happy Developers (<https://adventures92.medium.com/type-safe-builders-in-kotlin-the-secret-sauce-of-happy-developers-8ad077acbf51>)

## Kotlin - Scope Functions:

- Funções de escopo (let, run, with, apply, also) e como elas podem ser usadas para simplificar o código e melhorar a legibilidade.

### Conteúdos

- **Site** Documentação Kotlin: Scope functions (<https://kotlinlang.org/docs/scope-functions.html>)

## Kotlin - Date Serialization Library:

- Como trabalhar com bibliotecas de data e serialização em Kotlin para manipular datas e objetos serializados.

### Conteúdos

- **Site** Documentação Kotlin: Serialization (inglês) (<https://kotlinlang.org/docs/serialization.html>)

## KDoc:

- Mostrar a documentação KDoc e como escrever documentação de código em Kotlin.

## Conteúdos

- **Site** Documentação Kotlin: Document Kotlin code: KDoc (inglês) (<https://kotlinlang.org/docs/kotlin-doc.html>)

## Habilidade Auxiliar: Infraestrutura

### 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](https://www.youtube.com/watch?v=BU-w2_Aae54))
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git ([https://youtu.be/MqogPzjQyCY?si=VL\\_mT8EowuemptmU](https://youtu.be/MqogPzjQyCY?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](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-maneras-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>)

## 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](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>)

## 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](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=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=8JI9wQ8sUdQ>)
- **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](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 hyperlink 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](https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics))
- **YouTube** Rafaella Ballerini: 5 MINUTOS DE HTML PARA INICIAR EM PROGRAMAÇÃO! (<https://www.youtube.com/watch?v=3oSlqlqzN3M>)

- **YouTube** Rafaella Ballerini: LANDING PAGE COM HTML e CSS!  
(<https://www.youtube.com/watch?v=llF6vD-RljE>)
- **YouTube** Rafaella Ballerini: FORMULÁRIOS COM HTML e CSS!  
(<https://www.youtube.com/watch?v=wwqOJ2o84S4>)

### Conteúdos Alura:

- **Artigo** HTML: o que é, a importância para a Web, como aprender e um Guia para iniciantes (<https://www.alura.com.br/artigos/html>)
- **Alura+** Alura+: Fazendo deploy de um projeto na Vercel  
(<https://cursos.alura.com.br/extra/alura-mais/fazendo-deploy-de-um-projeto-na-vercel-c9239>)
- **Artigo** HTML, CSS e Javascript, quais as diferenças?  
(<https://www.alura.com.br/artigos/html-css-e-js-definicoes>)
- **Curso** Formação HTML e CSS (<https://www.alura.com.br/formacao-html-e-css>)
- **Curso** Apostila: Desenvolvimento Web com HTML, CSS e JavaScript  
(<https://www.alura.com.br/apostila-html-css-javascript>)
- **Desafio** 7 Days of Code: HTML e CSS  
(<https://7daysofcode.io/matricula/html-css>)

### 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](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](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](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](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=q28lfkBd9F4>)

- **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>)

## 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](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](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](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](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](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** Rafaela 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](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](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>)