

Java

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>)

Java - Fundamentos:

- Java é uma linguagem de programação amplamente usada para codificar aplicações Web. Java é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si. É uma linguagem de programação rápida, segura e confiável para codificar tudo, desde aplicações móveis e software empresarial até aplicações de big data e tecnologias do servidor.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando os diferentes tipos
- 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 métodos
- Manipular arrays e listas
- Obter dados de uma API
- Criar construtores

Conteúdos

- **Site** O que é Java? (<https://aws.amazon.com/pt/what-is/java/>)

- **Artigo** Aprenda Comigo: Java — Parte 1
(<https://medium.com/clebertech/aprenda-comigo-java-parte-1-42ef8fddd8b6>)
- **YouTube** FernandaDev: Java (Aula 1 - Classes, Objetos, Atributos e Métodos)
(<https://www.youtube.com/watch?v=ohmHbdUhAGc>)
- **YouTube** Loiane Groner: Primeiro Programa em Java
(<https://youtu.be/mu2ti43cgwc>)

Conteúdos Alura:

- **Artigo** Java: o que é, linguagem e um Guia para iniciar na tecnologia
(<https://www.alura.com.br/artigos/java>)
- **Podcast** Hipster 313 - Ecossistema Java revisitado
(<https://www.hipsters.tech/ecossistema-java-revisitado-hipsters-ponto-tech-313/>)
- **Artigo** Como começar a desenvolver em Java?
(<https://www.alura.com.br/artigos/comecando-com-o-desenvolvimento-java>)
- **Artigo** Meu primeiro programa em Java
(<https://www.alura.com.br/artigos/meu-primeiro-programa-em-java>)
- **Artigo** Java: Conheça o método main
(<https://www.alura.com.br/artigos/metodo-main-em-java>)
- **Artigo** Desenvolvendo aplicações Java com o VS Code
(<https://www.alura.com.br/artigos/desenvolvendo-aplicacoes-java-vs-code>)
- **Artigo** Diferença entre int e Integer em Java
(<https://www.alura.com.br/artigos/diferenca-entre-int-e-integer-em-java>)
- **Artigo** Como converter String para Date em Java
(<https://www.alura.com.br/artigos/como-converter-string-para-date-em-java>)
- **Artigo** Iterando uma lista em Java
(<https://www.alura.com.br/artigos/iterando-uma-lista-em-java>)
- **Artigo** Importando classes no Java
(<https://www.alura.com.br/artigos/importando-classes-em-java>)

- **Site** Apostila: Java e Orientação a Objetos (<https://www.alura.com.br/apostila-java-orientacao-objetos>)
- **Site** Apostila: Java para Desenvolvimento Web (<https://www.alura.com.br/apostila-java-web>)
- **YouTube** Alura: O que é o Java? (<https://www.youtube.com/watch?v=90NcVNsKGik>)
- **Curso** Formação Java e Orientação a Objetos (<https://cursos.alura.com.br/formacao-java>)
- **Desafio** 7 Days of Code: Java (<https://7daysofcode.io/matricula/java>)

Conceitos de Orientação a Objetos:

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

Conteúdos

- **Artigo** Princípios básicos da Programação Orientação a Objetos (POO) (<https://yanborowski.medium.com/princ%C3%ADpios-b%C3%A1sicos-da-programa%C3%A7%C3%A3o-orienta%C3%A7%C3%A3o-a-objetos-poo-62da3998b7ce>)

- **YouTube** FernandaDev: O que é Programação Orientada a Objetos?(POO) (<https://www.youtube.com/watch?v=QJjY2TNyl-8>)
- **YouTube** Giuliana Bezerra: Desmistificando os tipos de referência - Ep. 8 (<https://youtu.be/D9hANYbPYCc>)

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Orientação a Objetos – Hipsters Ponto Tech #350 (<https://www.hipsters.tech/techguide-orientacao-a-objetos-hipsters-ponto-tech-350/>)
- **Artigo** POO: o que é programação orientada a objetos? (<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>)
- **Site** Apostila: Java e Orientação a Objetos (<https://www.alura.com.br/apostila-java-orientacao-objetos>)
- **Site** Livro: Orientação a Objetos (<https://www.casadocodigo.com.br/products/livro-oo-conceitos>)
- **YouTube** Alura: Orientação a objetos (com Roberta Arcoverde) (<https://www.youtube.com/watch?v=jpuJ1qrluoU>)
- **YouTube** Alura: Imersão Java: Orientação a Objetos, APIs e além (e live coding!) (<https://www.youtube.com/watch?v=WdT90ffB-0Q>)
- **Curso** Formação Java e Orientação a Objetos (<https://cursos.alura.com.br/formacao-java>)
- **Curso** Formação C# e Orientação a Objetos (<https://cursos.alura.com.br/formacao-c-sharp-orientacao-objetos>)
- **Curso** Formação Python e orientação a objetos (<https://cursos.alura.com.br/formacao-Python-linguagem>)
- **Curso** Curso Orientação a Objetos com PHP: Classes, métodos e atributos (<https://www.alura.com.br/curso-online-php-oo-classes-metodos-atributos>)
- **Curso** Curso JavaScript: programação orientada a objetos (<https://www.alura.com.br/curso-online-javascript-passos-programacao-orientada-objetos>)

Java - 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** Wikibooks: Java - Exceções
(<https://pt.wikibooks.org/wiki/Java/Exce%C3%A7%C3%B5es>)
- **Artigo** Exceções em Java: Uma Primeira Conversa
(<https://www.linkedin.com/pulse/exce%C3%A7%C3%B5es-em-java-uma-primeira-conversa-vanessa/?originalSubdomain=pt>)
- **Artigo** Criando Exceptions para impressionar no Teste Técnico
(<https://dev.to/he4rt/criando-exceptions-para-impressionar-no-teste-tecnico-2nie>)
- **YouTube** Loiane Groner: Exceptions: try, catch
(<https://www.youtube.com/watch?v=ld2C4GcAtsg>)

Conteúdos Alura:

- **Artigo** Evite o NullPointerException no Java
(<https://www.alura.com.br/artigos/evite-o-nullpointerexception-no-java>)
- **Artigo** Be the exception! Dicas para dominar exceções em seus códigos Java
(<https://dev.to/dearrudam/be-the-exception-dicas-para-dominar-excecoes-em-seus-codigos-java-4nlo>)
- **Site** Java Exceções e Controle de Erros (<https://www.alura.com.br/apostila-java-orientacao-objetos/excecoes-e-controle-de-erros>)
- **Curso** Curso Java exceções: aprenda a criar, lançar e controlar exceções
(<https://cursos.alura.com.br/course/java-excecoes-aprenda-criar-lancar->

[controlar-excecoes\)](#)

Java - Coleções:

- 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 Equals e hashCode
- Saiba trabalhar com ArrayList, LinkedList ou Vector
- Classes Wrappers

Conteúdos

- **Site** Acervo Lima: Coleções em Java (<https://acervolima.com/colecoes-em-java/>)
- **Artigo** Dev.to: Básico efetivo sobre Coleções e Mapas com Java (<https://dev.to/gabrielsantosba/basico-efetivo-sobre-colecoes-e-mapas-com-java-4bnc>)
- **Artigo** Equals e hashCode: O que é, porque usar e como implementar (<https://angeliski.com.br/equals-e-hashcode?x-host=angeliski.com.br>)
- **YouTube** Caio Rodrigues: O Poder das Collections do Java (<https://www.youtube.com/watch?v=eLiHH9Bu04E>)
- **YouTube** Giuliana Bezerra: Aprenda a usar arrays no Java - Ep. 11 (<https://youtu.be/-wqLBOr3aqc>)
- **Artigo** Sassine.dev: Domine as Coleções e Mapas com Java Stream (<https://sassineasmar.hashnode.dev/java-stream-domine-as-listascolecoes-de-uma-vez-por-todas>)

Conteúdos Alura:

- **YouTube** Alura: Aprofundando no Java: domine as Collections (<https://www.youtube.com/watch?v=PeFDQtLBIt0>)

- **Curso** Curso Java: trabalhando com listas e coleções de dados (<https://cursos.alura.com.br/course/java-listas-colecoes-dados>)

Java - Testes:

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

Conteúdos

- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software! (<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Artigo** Testes 101 — Testando aplicações Java (<https://vepo.medium.com/testes-101-testando-aplica%C3%A7%C3%B5es-java-a12465319e61>)
- **Artigo** Iniciando em testes automatizados com Java - Parte 1 (<https://medium.com/@pauloloboneto/iniciando-seus-estudos-em-testes-automatizados-configure-sua-m%C3%A1quina-windows-87e72e6fe1d1>)
- **Artigo** Iniciando em testes automatizados com Java - Parte 2 (<https://medium.com/@pauloloboneto/iniciando-em-testes-automatizados-com-java-parte-2-ddf43a63b422>)
- **Artigo** Qualidade e JUnit: introduzindo automatização de testes unitários do seu software Java no dia-a-dia (<https://mari-azevedo.medium.com/qualidade-e-junit-introduzindo-automatiza%C3%A7%C3%A3o-de-testes-unit%C3%A1rios-do-seu-software-java-no-dia-a-dia-849611de5574>)

- **Artigo** Criando mocks e escrevendo testes unitários com Junit 5
(<https://mmarcosab.medium.com/criando-mocks-e-escrevendo-testes-unit%C3%A1rios-com-junit-5-f54e6407bd7c>)
- **Artigo** Turbinando sua aplicação Java com Testes Parametrizados
(<https://medium.com/@vandernobrel/turbinando-sua-aplica%C3%A7%C3%A3o-java-com-testes-parametrizados-70517f885f81>)
- **Artigo** Aplicando automação de testes em Java com BDD utilizando Cucumber
(<https://medium.com/vhsys-engineering/aplicando-automa%C3%A7%C3%A3o-de-testes-em-java-com-bdd-utilizando-cucumber-b0907cdd273b>)
- **YouTube** Leonardo Barauna: Teste Unitário em Java - Conceito
(<https://www.youtube.com/watch?v=PPagtVUrQr8>)
- **YouTube** Michelli Brito: TDD na prática com Java usando MockBean
(<https://www.youtube.com/watch?v=4VmbETu-dcA>)
- **YouTube** Michelli Brito: TDD na prática com Java usando MockMvc
(<https://www.youtube.com/watch?v=AGDcPgXj8BU>)
- **YouTube** Jean: Testes unitários em @Service com Mockito
(<https://www.youtube.com/watch?v=c8PFBxiVO8o>)

Conteúdos Alura:

- **Artigo** Tipos de testes: quais os principais e por que utilizá-los?
(<https://www.alura.com.br/artigos/tipos-de-testes-principais-por-que-utiliza-los>)
- **Podcast** Hipsters.tech: Testes Automatizados
(<https://www.alura.com.br/podcast/testes-automatizados-hipsters-51-a535>)
- **Curso** Curso Java e refatoração: melhorando códigos com boas práticas
(<https://cursos.alura.com.br/course/java-refatoracao-melhorando-codigos-boas-praticas>)
- **Curso** Curso Boas práticas de programação: automatizando testes com Java
(<https://cursos.alura.com.br/course/boas-praticas-programacao-testes-java>)

- **Curso** Curso Selenium: testes automatizados de aceitação em Java (<https://cursos.alura.com.br/course/selenium-testes-automatizados-aceitacao-java>)
- **Curso** Curso BDD e Java: Behavior Driven Development com Cucumber (<https://cursos.alura.com.br/course/bdd-cucumber-java>)

Java - Pacotes:

- Um pacote (package) em Java é usado para agrupar classes relacionadas, de forma semelhante a uma pasta em um diretório de arquivos. Os pacotes são usados para evitar conflitos de nomes e para escrever um código de melhor manutenção.
- Use imports e organize o seu código através de packages
- Conhecer a java.lang
- Entender a imutabilidade e a classe String
- Entender a classe java.lang.Object
- Conhecer a java.io

Conteúdos

- **Site** Acervo Lima: Como criar um pacote em Java (<https://acervolima.com/como-criar-um-pacote-em-java/>)
- **Site** Acervo Lima: Pacotes em Java (<https://acervolima.com/pacotes-em-java/>)
- **Artigo** Padronizando pacotes em uma arquitetura REST (<https://medium.com/@thaleslima19/padronizando-pacotes-em-uma-arquitetura-rest-280e7a6c38bb>)
- **YouTube** Loiane Groner: Orientação a Objetos - Pacotes e import (<https://www.youtube.com/watch?v=aRQHjfYBpM8>)
- **YouTube** Pedro Cavaleiro: Criando Pacotes no Eclipse (<https://www.youtube.com/watch?v=wFPXkNCOuf4>)

Conteúdos Alura:

- **YouTube** Alura: Aprofundando no Java: domine as Collections (<https://www.youtube.com/watch?v=PeFDQtLBIt0>)
- **Site** Pacotes - Organizando suas classes e bibliotecas (<https://www.alura.com.br/apostila-java-orientacao-objetos/pacotes-organizando-suas-classes-e-bibliotecas>)
- **Curso** Curso Java: trabalhando com listas e coleções de dados (<https://cursos.alura.com.br/course/java-listas-colecoes-dados>)

Estruturas de Dados:

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

Conteúdos

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

- **YouTube** Loiane Groner: Estrutura de Dados com Java | Filas
(https://www.youtube.com/watch?v=RdoLTgJSV_c)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Pilhas
(<https://www.youtube.com/watch?v=ZdU4wMyiTSs>)

Conteúdos Alura:

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

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)
- **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/U6Gh6HYyl6c>)

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)

Java - Gerenciamento da Memória:

- Em Java, o gerenciamento de memória é o processo de alocação e desalocação de objetos, chamado de gerenciamento de memória. Java faz

o gerenciamento de memória automaticamente. Java 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 Java
- Entender como funciona a memória o Garbage Collector

Conteúdos

- **Artigo** Gerenciamento de Memória Java
(<https://medium.com/@danilomoraiis/gerenciamento-de-mem%C3%B3ria-java-3969e7b8423c>)
- **Artigo** Java Garbage Collector — Por que precisamos conhecê-lo?
(<https://medium.com/luizalabs/java-garbage-collector-porque-precisamos-conhec%C3%AA-lo-9d26ebb0a6d8>)
- **YouTube** Loiane Groner: Garbage Collector (Coletor de Lixo)
(https://www.youtube.com/watch?v=_11TPkgZARk)
- **YouTube** Dias de Dev: Estratégias de gestão de memória - O que é Garbage Collector
(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)
- **Curso** Formação Java e Microserviços com Spring e RabbitMQ
(<https://cursos.alura.com.br/formacao-java-microservicos>)
- **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)
- **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:

- **Curso** Formação Java e Spring Boot (<https://www.alura.com.br/formacao-spring-boot-3>)
- **Curso** Formação Java Web: crie aplicações usando Spring Boot (<https://cursos.alura.com.br/formacao-java-web-spring-boot>)

Build tools Java:

- Uma build tool é um sistema que permite automatizar todas as tarefas rotineiras de um projeto de uma forma organizada e que evite que o desenvolvedor tenha que perder tempo. Em outras palavras, adicionar uma nova biblioteca, realização de testes, empacotamento e deploy, ou até mesmo, a compatibilidade entre as diversas IDEs são tarefas facilmente resolvidas com uma build tool.
- Conheça as principais ferramentas de build do Java, como Maven, Jenkins, Apache Ant, Gradle, etc., e como usá-las

Conteúdos

- **Site** Microsoft Docs: Maven e outras ferramentas de build para java (com legenda em português) (<https://docs.microsoft.com/pt-br/shows/java-for-beginners/maven-and-other-build-tools-for-java>)
- **Artigo** Entenda o que são build tools do mundo Java (<https://medium.com/@alex.felipe/entenda-o-que-s%C3%A3o-build-tools-do-mundo-java-682a26e38ff2>)
- **Artigo** Ant e o deploy e build de Projetos Java (<http://www.linhadecodigo.com.br/artigo/657/ant-e-o-deploy-e-build-de-projetos-java.aspx>)
- **YouTube** by { iorio }: Introdução ao Maven e Instalando a ferramenta (<https://www.youtube.com/watch?v=T-1Lgkqs6SM>)
- **YouTube** Pedro Cavaleiro: Entendendo Maven no Spring Boot (<https://www.youtube.com/watch?v=-KGAckgJKQA>)

Conteúdos Alura:

- **YouTube** Alura: Conhecendo as build tools no mundo Java (<https://www.youtube.com/watch?v=6lIRWbdnmuk>)
- **YouTube** Alura: Começando com o Gradle: Tasks e comandos básicos (<https://www.youtube.com/watch?v=uX6Ezf73OEY>)
- **Curso** Curso Maven: gerencie dependências e faça o build de aplicações Java (<https://cursos.alura.com.br/course/maven-gerencie-dependencias-build-aplicacoes-java>)

Java - Persistência:

- O conceito de "persistência de dados" refere-se a garantir que as informações inseridas na aplicação serão armazenadas em um meio em que possam ser recuperadas de forma consistente. Ou seja, são registros permanentes e que não são perdidos quando há o encerramento da sessão.
- Entender sobre JDBC e JPA
- Usar frameworks como Spring Data e Hibernate
- Comunicar-se com um banco de dados relacional
- Entender a diferença entre relacionamentos EAGER e LAZY
- Planejar queries com join fetch
- Encapsular o acesso em um DAO
- Entender como a memória funciona nessa situação

Conteúdos

- **Site** IBM Docs: API de Persistência Java (JPA) (<https://www.ibm.com/docs/pt-br/was-liberty/zos?topic=overview-java-persistence-api-jpa>)
- **Artigo** Do JDBC ao Spring Data (ou: é possível reduzir código?) (<https://dev.to/rodrigovp/do-jdbc-ao-spring-data-ou-e-possivel-reduzir-codigo-361f>)
- **YouTube** Rafael Mesquita: Persistência de Dados - Java JDBC (<https://www.youtube.com/watch?v=D-By3tUHcys>)

- **YouTube** Rafael Mesquita: Persistência de Dados - Java JDBC Códigos (<https://www.youtube.com/watch?v=cjFSgYv4RjI>)
- **YouTube** Michelli Brito: JPA - Java Persistence API (<https://www.youtube.com/watch?v=yvXCxWkLpfA>)
- **YouTube** Michelli Brito: JPA e Hibernate - configurando a transação manualmente (<https://www.youtube.com/watch?v=TQQki9yQaLg>)

Conteúdos Alura:

- **Site** Apostila Java para Desenvolvimento Web: Bancos de Dados e JDBC (<https://www.alura.com.br/apostila-java-web/bancos-de-dados-e-jdbc>)
- **Curso** Curso Java e JDBC: trabalhando com um banco de dados (<https://cursos.alura.com.br/course/java-jdbc-banco-dados>)
- **Curso** Curso Java e JPA: consultas avançadas, performance e modelos complexos (<https://cursos.alura.com.br/course/java-jpa-consultas-avancadas-performance-modelos-complexos>)
- **Curso** Curso Persistência com JPA: Hibernate (<https://cursos.alura.com.br/course/persistencia-jpa-introducao-hibernate>)

Nível 3

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 microserviços
- Realizar a comunicação usando APIs
- Melhorar a escalabilidade de um sistema

Conteúdos

- **Artigo** Google Cloud: O que é a arquitetura de microserviços? (<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=jSnLOoGjQ80>)
- **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 Microserviços na prática: entendendo a tomada de decisões (<https://cursos.alura.com.br/course/Microservicos-pratica-tomada-decisoes>)

Java - Concorrência:

- Programação concorrente é um paradigma de programação para a construção de programas que fazem uso da execução simultânea de várias tarefas computacionais interativas, que podem ser implementadas como programas separados ou como um conjunto de threads criadas por um único programa.
- Executar tarefas simultaneamente
- Colocar tarefas para aguardar até que um determinado evento ocorra
- Entender como a memória funciona nessa situação

Conteúdos

- **Artigo** Concorrência e Paralelismo — Threads, múltiplos processos e AsyncIO — Parte 1 (<https://medium.com/diogommartins/concorr%C3%Aancia-e-paralelismo-threads-m%C3%BAltiplos-processos-e-asyncio-parte-1-fbbdd2309d74>)
- **Artigo** Concorrência e Paralelismo — Threads, múltiplos processos e AsyncIO — Parte 2 (<https://medium.com/diogommartins/concorr%C3%Aancia-e-paralelismo-threads-m%C3%BAltiplos-processos-e-asyncio-parte-2-8e539c59a16a>)
- **Artigo** Understanding Java threads once and for all (inglês) (<https://medium.com/swlh/understanding-java-threads-once-and-for-all-711f71e0ec1e>)
- **YouTube** Loiane Groner: Criando Threads + métodos start, run e sleep (<https://www.youtube.com/watch?v=v5l30QMKv6c>)
- **YouTube** Michelli Brito: Threads em java (<https://www.youtube.com/watch?v=rVsYXGwbmag>)

Conteúdos Alura:

- **Site** Programação Concorrente E Threads
(<https://www.alura.com.br/apostila-java-orientacao-objetos/apendice-programacao-concorrente-e-threads>)
- **Curso** Curso Java threads: aprenda a criar, gerenciar e aplicar com o Spring
(<https://cursos.alura.com.br/course/java-threads-criar-gerenciar-aplicar-spring>)

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=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>)

Kafka:

- O Apache Kafka é uma plataforma distribuída de transmissão de dados que é capaz de publicar, subscrever, armazenar e processar fluxos de registro em tempo real. Essa plataforma foi desenvolvida para processar fluxos de dados provenientes de diversas fontes e entregá-los a vários clientes.
- Utilizar o Kafka para comunicação assíncrona
- Criar microsserviços com Kafka
- Criar produtores e consumidores
- Entender como usar o Kafka para paralelismo e execução serializada
- Obter garantias relativas ao envio ou entrega das mensagens

Conteúdos

- **Site** Red Hat: O que é Apache Kafka? (<https://www.redhat.com/pt-br/topics/integration/what-is-apache-kafka>)
- **Artigo** Kafka — Produzindo mensagens com Java (<https://medium.com/experiencecode/kafka-produzindo-mensagens-com-java-13d326223507>)
- **Artigo** Criando consumers e producers em Kafka com Java (<https://ivanqueiroz.dev/2020/09/2020-09-26-consumer-producer-kafka.html>)
- **Artigo** Apache Kafka — Aprendendo na prática (<https://medium.com/trainingcenter/apache-kafka-codifica%C3%A7%C3%A3o-na-pratica-9c6a4142a08f>)
- **YouTube** Rodrigo Freitas: Java - Kafka para Iniciantes (https://www.youtube.com/watch?v=3xxb_QyrFnw)

- **YouTube** Expertos Tech: Como conectar o JAVA com o APACHE KAFKA (<https://www.youtube.com/watch?v=Vc6kvqaU1Jg>)

Conteúdos Alura:

- **YouTube** Alura: O que é Kafka? (<https://www.youtube.com/watch?v=G6Tcy7hNdA8>)
- **YouTube** Alura: Começando com Kafka - Hipsters Ponto Talks 12 (<https://www.youtube.com/watch?v=aaCqcX30pzc>)
- **Podcast** Hipsters.Tech: Tudo sobre Kafka - Hipsters On The Road 14 (<https://www.alura.com.br/podcast/hipsterstech-tudo-sobre-kafka-hipsters-on-the-road-14-a411>)
- **Curso** Formação Apache Kafka (<https://www.alura.com.br/formacao-kafka>)
- **Site** Livro Apache Kafka e Spring Boot da Casa do Código (<https://www.casadocodigo.com.br/products/livro-apache-kafka>)

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)
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git (https://youtu.be/MqoqPzjQyCY?si=Vl_mT8EowuemptmU)
- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub (https://youtu.be/v9ZM2PVzctM?si=49ah_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-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>)

JSON:

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

Conteúdos

- **Site** MDN Web Docs: JSON (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON)
- **Site** MDN Web Docs: Trabalhando com JSON (<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>)
- **Artigo** Introdução ao JSON (<https://juliocarneiro.medium.com/introdu%C3%A7%C3%A3o-ao-json-7825b1a550ff>)
- **YouTube** Marco Bruno: O que é JSON e como criar um objeto (<https://www.youtube.com/watch?v=oCY5YEEjlwE>)

Conteúdos Alura:

- **Artigo** O que é JSON? (<https://www.alura.com.br/artigos/o-que-e-json>)
- **Curso** Curso MySQL e JSON: persistindo JSON de maneira eficiente (<https://www.alura.com.br/curso-online-mysql-json-persistencia>)

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=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)
- **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>)

SQL - Fundamentos:

- SQL (Structured Query Language, traduzindo, Linguagem de Consulta Estruturada) é uma linguagem de programação padronizada que é usada para gerenciar bancos de dados relacionais e realizar várias operações sobre os dados neles contidos.
- Conhecer os comandos mais comuns do SQL
- Usar SELECT para consultar uma tabela
- Usar INSERT para inserir dados em uma tabela
- Usar UPDATE para atualizar uma tabela
- Usar DELETE para remover dados de uma tabela
- Usar JOIN para conectar os dados de múltiplas tabelas
- Conhecer as cláusulas (FROM, ORDER BY, etc)

Conteúdos

- **Artigo** SQL — O que é e como funciona na prática? (<https://medium.com/pravaler-digital-team/sql-o-que-%C3%A9-e-como-funciona-na-pr%C3%A1tica-6ae7a322554a>)
- **Artigo** SQL: comandos básicos (<https://medium.com/nerdzao/sql-comandos-b%C3%A1sicos-483a8dc12909>)
- **YouTube** CreateSe: Comandos básicos do SQL (https://www.youtube.com/watch?v=_rZxrqn0X-4)

Conteúdos Alura:

- **Artigo** O que é SQL? (<https://www.alura.com.br/artigos/o-que-e-sql>)
- **Artigo** SQL JOIN: Aprenda INNER, LEFT, RIGHT, FULL e CROSS (<https://www.alura.com.br/artigos/join-em-sql>)
- **YouTube** Alura: O que é SQL e NoSQL? (<https://www.youtube.com/watch?v=aure5d3B88g>)

- **Curso** Formação Conhecendo SQL (<https://www.alura.com.br/formacao-conhecendo-sql>)
- **Curso** Curso SQL com MySQL: manipule e consulte dados (<https://cursos.alura.com.br/course/mysql-manipule-dados-com-sql>)
- **Curso** Formação SQL com MySQL Server da Oracle (<https://cursos.alura.com.br/formacao-oracle-mysql>)
- **Artigo** SQL e NoSQL: trabalhando com bancos relacionais e não relacionais (<https://www.alura.com.br/artigos/sql-nosql-bancos-relacionais-nao-relacionais>)
- **Curso** Microsoft SQL Server 2022: conhecendo SQL (<https://cursos.alura.com.br/course/microsoft-sql-server-conhecendo-sql>)
- **Curso** Curso de PostgreSQL (<https://cursos.alura.com.br/course/introducao-postgresql-primeiros-passos>)
- **Curso** SQL com Oracle Database: manipule e consulte dados (<https://cursos.alura.com.br/course/introducao-sql-oracle-manipule-consulte-dados>)
- **Podcast** SQL e MySQL (<https://cursos.alura.com.br/extra/hipsterstech/sql-e-mysql-hipsters-ponto-tech-404-a9270>)

Habilidade Auxiliar: Boas práticas

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-melhore-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>)

Clean Architecture:

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

Conteúdos

- **Artigo** Descomplicando a Clean Architecture
(<https://medium.com/luizalabs/descomplicando-a-clean-architecture-cf4dfc4a1ac6>)
- **YouTube** Como DEV ser!: Entenda CLEAN ARCHITECTURE de uma vez por todas! (<https://www.youtube.com/watch?v=HynTfTli4mw>)
- **YouTube** Full Cycle: O que é Clean Architecture
(<https://www.youtube.com/watch?v=dQys-mnOtQg>)
- **YouTube** CodandoTV(Rods) - Simplificando o Clean Architecture +MVVM na sua aplicação mobile - Guia Completo (<https://youtu.be/8ehIZfyN1S0?si=I7-I5l4zsnYLI5jD>)

Conteúdos Alura:

- **Site** Clean Architecture (Arquitetura Limpa) - O que é?
(<https://cursos.alura.com.br/extra/alura-mais/clean-architecture->

[arquitetura-limpa-o-que-e--c204\)](#)

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

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>)

Clean Code:

- Aplicar técnicas simples que visam facilitar a escrita e leitura de um código
- Refatorar seu código para que fique mais claro

Conteúdos

- **Artigo** Clean Code: O que é? Porque usar? (<https://medium.com/desenvolvendo-com-paixao/1-clean-code-o-que-%C3%A9-porque-usar-1e4f9f4454c6>)
- **YouTube** Filipe Deschamps: CLEAN CODE 1 - Introdução (<https://www.youtube.com/watch?v=9w3o9NHXqu0>)

- **YouTube** Filipe Deschamps: CLEAN CODE 2 - O que é código limpo? (https://www.youtube.com/watch?v=ZVtGbO_CnfA)
- **Livro** Código limpo: habilidades práticas do Agile software (<https://www.amazon.com.br/C%C3%B3digo-limpo-Robert-C-Martin/dp/8576082675>)

Conteúdos Alura:

- **Artigo** Clean Code: O que é, Casos de Uso, Exemplo de Código Limpo (<https://www.alura.com.br/artigos/o-que-e-clean-code>)
- **YouTube** Alura: Clean Code e Solid (https://www.youtube.com/watch?v=XV9B4LX_re8)
- **Livro** Orientação a Objetos e SOLID para Ninjas (https://www.casadocodigo.com.br/products/livro-oo-solid?_pos=2&_sid=fe732c657&_ss=r)

Conceitos de Design Orientado a Domínio (Domain-Driven Design - DDD):

- O Design Orientado a Domínio (DDD) é uma abordagem ao projeto e desenvolvimento de software que é primeiramente informado pelos requisitos de negócios. Os componentes do programa (objetos, classes, matrizes, etc.) indicam a indústria, setor ou domínio empresarial em que o negócio opera.
- Modelar domínios de forma efetiva
- Basear projetos complexos em modelos do domínio
- Conhecer os blocos de construção de DDD

Conteúdos

- **Artigo** Domain-Driven Design, do início ao código (<https://medium.com/cwi-software/domain-driven-design-do-in%C3%ADcio-ao-c%C3%B3digo-569b23cb3d47>)
- **Artigo** Introdução ao DDD (<https://medium.com/xp-inc/introdu%C3%A7%C3%A3o-ao-ddd-2cad38e4234a>)

- **YouTube** Dias de Dev: O que é Domain-Driven Design (<https://www.youtube.com/watch?v=SNdGV0c40yo>)
- **YouTube** Otavio Lemos: DDD + Clean Architecture (<https://www.youtube.com/watch?v=cZuyPPrRGAw>)

Conteúdos Alura:

- **YouTube** Alura: Domain Driven Design (<https://www.youtube.com/watch?v=n40Z1c9Ryog>)
- **Site** Domain-Driven Design (DDD) - O que é? (<https://cursos.alura.com.br/extra/alura-mais/domain-driven-design-ddd-o-que-e--c283>)
- **Site** Primeiras aulas do curso Java e Domain Driven Design: apresentando os conceitos (<https://www.alura.com.br/conteudo/java-domain-driven-design-conceitos>)
- **Curso** Curso Java e Domain Driven Design: apresentando os conceitos (<https://www.alura.com.br/curso-online-java-domain-driven-design-conceitos>)
- **Site** Primeiras aulas do curso PHP e Domain Driven Design: apresentando os conceitos (<https://www.alura.com.br/conteudo/domain-driven-design-php>)
- **Curso** Curso PHP e Domain Driven Design: apresentando os conceitos (<https://www.alura.com.br/curso-online-domain-driven-design-php>)