



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>)

Go (GoLang) - Fundamentos:

- Go é uma linguagem de programação de alto nível compilada e estaticamente tipada, desenvolvida no Google. Ela é sintaticamente semelhante ao C, mas também tem memory safety, coletor de lixo, tipagem estrutural e concorrência.
- 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
- Tratamento de erros
- Convenções de código
- Manipular Coleções, arrays e listas

Conteúdos

- **Site** Documentação Go: Começando com Go (inglês)
(<https://go.dev/doc/tutorial/getting-started>)

- **Site** Documentação Go: Criando um módulo Go (inglês)
(<https://go.dev/doc/tutorial/create-module>)
- **Artigo** GoLang — Simplificando a complexidade
(<https://medium.com/trainingcenter/golang-d94e16d4b383>)
- **Artigo** Trabalhando com concorrência em Go — com benchmark
(<https://vinnynordi.com/trabalhando-com-concorr%C3%Aancia-em-go-com-benchmark-aecefdc7ab7c>)
- **YouTube** Código Fonte TV: Golang (A Linguagem do Futuro?) // Dicionário do Programador (https://www.youtube.com/watch?v=Q0sKAMal4WQ&ab_channel=edureka%21)
- **YouTube** Léo Miranda Dev: Primeiros passos em Go (Golang)
(https://www.youtube.com/watch?v=Q0sKAMal4WQ&ab_channel=edureka%21)
- **YouTube** edureka!: Golang Tutorial For Beginners
(https://www.youtube.com/watch?v=Q0sKAMal4WQ&ab_channel=edureka%21)

Conteúdos Alura:

- **Artigo** Variáveis em Go: como elas funcionam?
(<https://www.alura.com.br/artigos/variaves-com-go-lang>)
- **Artigo** Estruturas básicas de controle com GO
(<https://www.alura.com.br/artigos/estruturas-basicas-de-controle-com-go>)
- **Artigo** Conversão de tipos com Go
(<https://www.alura.com.br/artigos/conversao-de-tipos-com-go>)
- **Artigo** Criando uma simples aplicação web com GO
(<https://www.alura.com.br/artigos/criando-uma-simples-aplicacao-web-com-go>)
- **Artigo** Golang: trabalhando com datas
(<https://www.alura.com.br/artigos/golang-trabalhando-com-datas>)
- **Podcast** A linguagem Go – Hipsters #67
(<https://www.hipsters.tech/linguagem-go-hipsters-67/>)

- **Podcast** Ecossistemas Golang- Hipsters Ponto Tech #296 (<https://www.hipsters.tech/ecossistemas-golang-hipsters-ponto-tech-296/>)
- **YouTube** Conhecendo a linguagem Go | Hipsters.Talks (https://www.youtube.com/watch?v=y8FeZMv37WU&ab_channel=Alura)
- **Curso** Formação Linguagem Go (<https://www.alura.com.br/formacao-go>)

Go - Orientação a Objetos:

- Entender como métodos são definidos e associados a tipos em Go.
- Aprender a diferenciar métodos e funções dentro do contexto de Go.
- Explorar a aplicação prática de métodos para manipulação de estruturas e interfaces.

Conteúdos

- **Artigo** Go Methods (inglês) (<https://go.dev/tour/methods/1>)
- **Artigo** Methods in Golang (inglês) (<https://www.geeksforgeeks.org/methods-in-golang/>)

Conteúdos Alura:

- **Curso** Go: Orientação a Objetos (<https://cursos.alura.com.br/course/go-lang-oo>)

Go - Estruturas:

- Aprender a definir e utilizar estruturas em Go.
- Entender como estruturas podem ser utilizadas para modelar dados e objetos.
- Explorar a relação entre estruturas e métodos para simular a orientação a objetos.

Conteúdos

- **Artigo** Definindo structs no Go (<https://www.digitalocean.com/community/tutorials/defining-structs-in-go-pt>)

- **Artigo** Structs (inglês) (<https://go.dev/tour/moretypes/2>)

Conteúdos Alura:

- **Curso** Go: Orientação a Objetos (<https://cursos.alura.com.br/course/go-lang-oo>)

Go - Tratamento de Erros:

- Compreender como Go trata erros através de seus mecanismos integrados.
- Aprender a utilizar o tipo **error** para reportar e tratar falhas.
- Explorar padrões de tratamento de erro para manter o código limpo e eficiente.

Conteúdos

- **Artigo** Errors (inglês) (<https://pkg.go.dev/errors>)
- **Artigo** Go por Exemplo: Erros (<https://goporexemplo.golangbr.org/errors.html>)

Nível 2

Go - Ponteiros:

- Entender o conceito de ponteiros e sua sintaxe em Go.
- Aprender a usar ponteiros para manipulação direta de memória e passagem por referência.
- Explorar técnicas avançadas de ponteiros, incluindo ponteiros para estruturas.

Conteúdos

- **Artigo** Pointers in Go (inglês) (<https://go.dev/tour/moretypes/1>)
- **Artigo** A Comprehensive Guide to Pointers in Go (<https://medium.com/@jamal.kaksouri/a-comprehensive-guide-to-pointers-in-go-4acc58eb1f4d>)

- **Artigo** Pointers in Golang (<https://www.geeksforgeeks.org/pointers-in-golang/>)

Go - Interfaces:

- Compreender o papel das interfaces em Go e como elas contribuem para o polimorfismo.
- Aprender a definir e implementar interfaces, facilitando a interação entre diferentes tipos.
- Explorar o uso de interfaces para desacoplamento e testabilidade do código.

Conteúdos

- **Artigo** Como usar interfaces em Go (<https://www.digitalocean.com/community/tutorials/how-to-use-interfaces-in-go-pt>)
- **Artigo** Interfaces (inglês) (<https://go.dev/tour/methods/9>)

Conteúdos Alura:

- **Curso** Curso Go: Orientação a Objetos (<https://cursos.alura.com.br/course/go-lang-oo>)

Go - Concorrência:

- Entender o modelo de concorrência em Go e como ele difere de outros modelos de programação paralela.
- Aprender a criar e gerenciar goroutines para tarefas concorrentes.
- Utilizar canais para a comunicação segura entre goroutines.

Conteúdos

- **Artigo** Concurrency in Go - Goroutines (inglês) (<https://go.dev/tour/concurrency/1>)

Go - Pacotes:

- Compreender a estrutura e o propósito dos pacotes em Go.

- Aprender a criar, importar e exportar pacotes para modularizar o código.
- Explorar o gerenciamento de dependências utilizando o Go Modules.

Conteúdos

- **Artigo** Packages (inglês) (<https://pkg.go.dev/golang.org/x/tools/go/packages>)
- **Artigo** How To Write Packages in Go(inglês)" (<https://www.digitalocean.com/community/tutorials/how-to-write-packages-in-go>)

Conteúdos Alura:

- **Curso** Curso Go: Orientação a Objetos (<https://cursos.alura.com.br/course/go-lang-oo>)

Go - Testes:

- Aprender a escrever testes unitários e de integração em Go.
- Entender a importância dos testes para manter a qualidade do código.
- Utilizar ferramentas e frameworks de teste disponíveis na comunidade Go.

Conteúdos

- **Artigo** Go Add a test (inglês) (<https://go.dev/doc/tutorial/add-a-test>)
- **Artigo** Testing in Go (inglês) (<https://pkg.go.dev/testing>)
- **YouTube** Comprehensive Guide to Testing in Go (inglês) (<https://blog.jetbrains.com/go/2022/11/22/comprehensive-guide-to-testing-in-go/>)

Conteúdos Alura:

- **Curso** Curso Go: validações, testes e páginas HTML (<https://cursos.alura.com.br/course/go-validacoes-testes-paginas-html>)

Go - Persistência de Dados:

- Aprender os fundamentos da persistência de dados em Go.

- Explorar a integração de Go com bancos de dados SQL e NoSQL.
- Entender melhores práticas para modelagem e acesso a dados em aplicações Go.

Conteúdos

- **Artigo** Persistence (inglês) (<https://pkg.go.dev/github.com/keep-network/keep-common/pkg/persistence>)
- **Artigo** Simple (And Easy) Data Persistence in Go (<https://medium.com/@shanev/simple-and-easy-data-persistence-in-go-aa019a6f3106>)

Conteúdos Alura:

- **Curso** Curso Go: crie uma aplicação web (<https://cursos.alura.com.br/course/go-lang-web>)

Go - Goroutines e Canais:

- Entender o modelo de concorrência baseado em goroutines e canais em Go.
- Aprender a criar e sincronizar goroutines para tarefas concorrentes.
- Explorar padrões de uso de canais para comunicação segura entre goroutines.

Conteúdos

- **Artigo** Goroutines (inglês) (<https://go.dev/tour/concurrency/1>)
- **Artigo** Goroutines e go channels (<https://medium.com/trainingcenter/goroutines-e-go-channels-f019784d6855>)
- **Artigo** Um Pouco Sobre Goroutines em Go! (<https://dev.to/jeffotoni/um-pouco-sobre-goroutines-em-go-3lbk>)

Go - Programação Web:

- Aprender a construir servidores web e APIs RESTful com Go.

- Explorar o ecossistema de frameworks e bibliotecas web disponíveis para Go.
- Entender práticas recomendadas para o desenvolvimento web seguro e eficiente com Go.

Conteúdos

- **Artigo** Writing Web Applications (inglês) (<https://go.dev/doc/articles/wiki/>)

Conteúdos Alura:

- **Curso** Go: crie uma aplicação web (<https://cursos.alura.com.br/course/go-lang-web>)

Nível 3

Go - Construindo APIs REST:

- Compreender os princípios básicos de APIs RESTful e sua implementação em Go.
- Aprender a estruturar uma aplicação Go para suportar o desenvolvimento de APIs.
- Explorar ferramentas e bibliotecas úteis para facilitar o desenvolvimento de APIs em Go.

Conteúdos

- **Artigo** Tutorial: Developing a RESTful API with Go and Gin(inglês) (<https://go.dev/doc/tutorial/web-service-gin>)
- **Artigo** Criando uma API REST em Go (Golang) com Erro Personalizado (<https://iamwilliamkoller.medium.com/criando-uma-api-rest-em-go-golang-com-erro-personalizado-668db4967f37>)

Conteúdos Alura:

- **Curso** Curso Go: Desenvolvendo uma API REST (<https://cursos.alura.com.br/course/go-desenvolvendo-api-rest>)

- **Curso** Curso Go e Gin: criando API rest com simplicidade (<https://cursos.alura.com.br/course/go-gin-api-rest-simplicidade>)

Arquitetura de Microsserviços:

- Microsserviços são uma abordagem de arquitetura na qual o software consiste de pequenos serviços independentes que se comunicam entre si e são organizados de acordo com seus domínios de negócio.
- Aprender o conceito de arquitetura planejada para microsserviços
- Realizar a comunicação usando APIs
- Melhorar a escalabilidade de um sistema

Conteúdos

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

Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Microsserviços – Hipsters Ponto Tech #349 (<https://www.hipsters.tech/techguide-microservicos-hipsters-ponto-tech-349/>)
- **YouTube** Alura: O que são Microsserviços? (<https://www.youtube.com/watch?v=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/microsservicos-na-caelum-hipsters-on->

[the-road-6-a436\)](#)

- **Podcast** Hipsters.tech: Microserviços 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 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>)

Go - Debugging:

- Aprender a utilizar as ferramentas de debugging disponíveis para Go.
- Compreender técnicas eficazes para identificar e resolver bugs em Go.
- Explorar melhores práticas para prevenir erros comuns em desenvolvimento Go.

Conteúdos

- **Artigo** How To Debug Go Code with Visual Studio Code (<https://www.digitalocean.com/community/tutorials/debugging-go-code-with-visual-studio-code>)

Streaming:

- Streaming é o processamento e transmissão contínua de dados em tempo real, à medida que são gerados. Diferentemente do armazenamento

tradicional, onde os dados são coletados e processados posteriormente, o streaming permite a análise e o uso imediato dos dados à medida que são recebidos. Essa abordagem é ideal para lidar com grandes volumes de dados em alta velocidade, permitindo a detecção de padrões em tempo real, tomada de decisões instantâneas e resposta rápida a eventos em andamento.

- O streaming desempenha um papel essencial na engenharia de dados, capacitando as organizações a processar, analisar e tomar decisões com base em dados em tempo real. Isso resulta em maior agilidade, escalabilidade e reatividade, possibilitando uma tomada de decisões mais informada e orientada por insights atualizados.

Conteúdos

- **Site** O que são fluxos de dados dados em tempo real? (<https://aws.amazon.com/pt/what-is/real-time-data-streaming/>)
- **YouTube** Kafka (Plataforma de Mensageria e Streaming) (https://www.youtube.com/watch?v=qOqXz5Qv_-8)
- **YouTube** Processando 1bi de mensagens com open source (<https://www.youtube.com/watch?v=SEBv4N595eY>)
- **YouTube** Streaming de Dados (<https://www.youtube.com/watch?v=PaUeW5a-PTs>)

Conteúdos Alura:

- **Artigo** Spark Streaming (<https://www.alura.com.br/artigos/spark-streaming>)
- **Curso** Kafka: Produtores, Consumidores e streams (<https://cursos.alura.com.br/course/kafka-introducao-a-streams-em-microservicos>)
- **Curso** Formação Mensageria com Apache Kafka (<https://cursos.alura.com.br/formacao-kafka>)

Go - gRPC:

- Compreender os fundamentos e a arquitetura do gRPC e como ele se integra ao ecossistema Go.

- Aprender a definir serviços gRPC usando protocol buffers, facilitando a interoperabilidade e a eficiência na comunicação entre serviços.
- Desenvolver servidores e clientes gRPC em Go, explorando recursos avançados como streaming, autenticação e interceptores.
- Adotar melhores práticas para o desenvolvimento de APIs gRPC em Go, garantindo segurança, performance e escalabilidade.

Conteúdos

- **Artigo** GRPC (inglês) (<https://pkg.go.dev/google.golang.org/grpc>)
- **Artigo** Escrevendo um micro serviço gRPC em Go: estrutura, definição e deploy no Google Kubernetes Engine
(<https://medium.com/unicoidtech/escrevendo-um-micro-servi%C3%A7o-grpc-em-go-estrutura-defini%C3%A7%C3%A3o-e-deploy-no-google-kubernetes-engine-82351e23a2f7>)

Conteúdos Alura:

- **Podcast** Hipsters On The Road #54: Go, gRPC e investimentos na Guru
(<https://cursos.alura.com.br/extra/hipsterstech/go-grpc-e-investimentos-na-guru-hipsters-on-the-road-54-a828>)

Go - Melhores Práticas de Segurança:

- Compreender as vulnerabilidades comuns em aplicações Go e como mitigá-las.
- Aprender a implementar técnicas de segurança, como autenticação e criptografia.
- Explorar ferramentas e bibliotecas de segurança disponíveis para Go.

Conteúdos

- **Artigo** Security Best Practices for Go Developers
(<https://go.dev/doc/security/best-practices>)
- **Artigo** Security (inglês) (<https://go.dev/doc/security/>)

Conteúdos Alura:

- **Curso** Formação Desenvolvimento Seguro
(<https://cursos.alura.com.br/formacao-desenvolvimento-seguro>)

Go - Otimização de Performance:

- Aprender a medir e analisar a performance de aplicações Go.
- Identificar gargalos de performance e aplicar técnicas de otimização.
- Explorar ferramentas e práticas para escrever código Go de alta performance.

Conteúdos

- **Artigo** Performance (inglês) (<https://go.dev/wiki/Performance>)
- **Artigo** Ultimate Golang Performance Optimization (inglês)
(<https://medium.com/@santoshcitpl/ultimate-golang-performance-optimization-guide-connect-infosoft-9c4c2b75b9f2>)

Habilidade Auxiliar: DevOps

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>)

Entrega e integração contínuas (CI/CD):

- CI/CD é a abreviação de Continuous Integration/Continuous Delivery, traduzindo para o português "entrega e integração contínuas". Trata-se de uma prática de desenvolvimento de software que visa tornar a integração de código mais eficiente por meio de builds e testes automatizados.
- Automatizar a integração de código entre varias partes da equipe se tornou cada vez mais importante, ja que assim é possível acelerar o desenvolvimento e diminuir o tempo de entrega de software.

- Executar testes automatizados da aplicação para verificar seu funcionamento.
- Realizar a entrega de atualizações de forma automática e com segurança.
- Realizar testes de conexão e testes de carga para evitar que a aplicação apresente problemas ao ser atualizada.

Conteúdos

- **Artigo** O que é CI/CD? (<https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>)
- **Artigo** Sobre integração contínua (<https://docs.github.com/pt/actions/automating-builds-and-tests/about-continuous-integration>)
- **YouTube** Código Fonte TV: Pipeline (CD/CI) | Dicionário do Programador (<https://www.youtube.com/watch?v=AZtTd3pFVTY>)

Conteúdos Alura:

- **Artigo** Entrega e integração contínua de aplicações Vue (<https://www.alura.com.br/artigos/entrega-e-integracao-continua-de-aplicacoes-vue>)
- **Artigo** GitOps: o que é? (<https://www.alura.com.br/artigos/o-que-e-git-ops>)
- **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** Integração Contínua, Deploy Contínuo e Github Actions (<https://cursos.alura.com.br/extra/hipsterstech/integracao-continua-deploy-continuo-e-github-actions-hipsters-213-a335>)
- **Podcast** Testes Automatizados (<https://cursos.alura.com.br/extra/hipsterstech/testes-automatizados-hipsters-51-a535>)
- **Curso** Curso Integração Contínua: mais qualidade e menos risco no desenvolvimento (<https://cursos.alura.com.br/course/desenvolvimento-software-integracao-continua>)

- **Curso** Curso Entrega Contínua: confiabilidade e qualidade na implantação de software (<https://cursos.alura.com.br/course/entrega-continua-confiabilidade-qualidade>)
- **Curso** Formação: Integração Contínua e Entrega Contínua (<https://cursos.alura.com.br/formacao-integracao-continua-entrega-continua>)

Habilidade Auxiliar: Best Practices

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>)

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>)

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>)

Contêineres:

- Os contêineres são pacotes de software que contêm todos os elementos necessários para serem executados em qualquer ambiente. Gerenciamento de contêineres é uma área crucial na computação em nuvem e DevOps, que envolve o uso de tecnologias para automatizar o processo de criação, implantação, escalonamento e monitoramento de contêineres. Contêineres são unidades de software padronizadas que permitem aos desenvolvedores empacotar todas as dependências de um aplicativo (código, bibliotecas,

configurações, etc.) em um único pacote. Isso permite que o aplicativo seja executado de forma consistente em qualquer ambiente de infraestrutura.

- A tecnologia de contêineres, como exemplificada pelo Docker, fornece um ambiente consistente e portátil para desenvolvimento, teste e implantação de aplicativos, o que é vital para o trabalho eficiente de engenharia de dados. Além disso, o Kubernetes, um sistema de orquestração de contêineres, permite o gerenciamento, a automação e a escalabilidade de aplicações baseadas em contêineres em ambientes de produção. Dominar esses conceitos e tecnologias possibilita a engenheiros de dados construir e manter pipelines de dados eficientes e confiáveis.
- O Kubernetes (também conhecido como k8s ou kube) é uma plataforma de orquestração de containers open source que automatiza grande parte dos processos manuais necessários para implantar, gerenciar e escalar aplicações em containers.
- Isolar seu software para funcionar independentemente
- Implantar software em clusters
- Modularizar seu sistema em pacotes menores
- Conhecer a plataforma Docker
- Conhecer Kubernetes

Conteúdos

- **Site** IBM: Contêineres (<https://www.ibm.com/br-pt/cloud/learn/containers>)
- **Site** HPE: O que são contêineres? (<https://www.hpe.com/br/pt/what-is/containers.html>)
- **Site** Google Cloud: O que são contêineres? (<https://cloud.google.com/learn/what-are-containers?hl=pt-br>)
- **Site** Azure: O que é um contêiner? (<https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-container/>)
- **Artigo** Red Hat: O que é Kubernetes (<https://www.redhat.com/pt-br/topics/containers/what-is-kubernetes>)
- **Artigo** Azure: O que é Kubernetes? (<https://azure.microsoft.com/pt-br/topic/what-is-kubernetes/>)

- **Artigo** Kubernetes - Guia de Referência Gratuito (<https://renatogroffe.medium.com/kubernetes-guia-de-refer%C3%A2ncia-gratuito-a2c1be8e7593>)
- **YouTube** Código Fonte TV: Containers // Dicionário do Programador (<https://www.youtube.com/watch?v=-pUZBovqRcU>)
- **YouTube** Código Fonte TV: Kubernetes // Dicionário do Programador (<https://www.youtube.com/watch?v=mVL0nOM3AGo>)
- **YouTube** Insight Lab: Revisão Docker - Curso de Introdução ao Kubernetes (<https://www.youtube.com/watch?v=bcRARpK00OU>)
- **YouTube** Pisani da Arch: Kubernetes - entenda e comece a arquitetar agora mesmo (<https://www.youtube.com/watch?v=XjaqTqm7g58>)

Conteúdos Alura:

- **Artigo** O que é Kubernetes (<https://www.alura.com.br/artigos/o-que-e-kubernetes>)
- **Artigo** Criando volumes com Docker (<https://www.alura.com.br/artigos/criando-volumes-com-docker>)
- **Artigo** Docker Compose para compor uma aplicação (<https://www.alura.com.br/artigos/compondo-uma-aplicacao-com-o-docker-compose>)
- **YouTube** Alura: O que são Containers? (https://www.youtube.com/watch?v=jv4_sLIGOS0)
- **YouTube** Alura: Containers, Docker e Kubernetes (com Giovanni Bassi) (<https://www.youtube.com/watch?v=wxLvMxzc1Q>)
- **Podcast** Hipsters.tech: Containers e Docker - Hipsters 75 (<https://www.alura.com.br/podcast/hipsterstech-containers-e-docker-hipsters-75-a511>)
- **Podcast** Hipsters.tech: Kubernetes - Hipsters 177 (<https://www.alura.com.br/podcast/kubernetes-hipsters-177-a389>)
- **Podcast** Hipsters.tech: Microsserviços e Kubernetes na Americanas.com - Hipsters On The Road 56

(<https://www.alura.com.br/podcast/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>)

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>)

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-I5l4zsnYLIjD>)

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>)

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>)

Infraestrutura como código (IaC):

- Infraestrutura como código (IaC) é uma abordagem em que a infraestrutura de TI, incluindo servidores, redes e recursos relacionados, é gerenciada e provisionada por meio de código. Em vez de configurar manualmente a infraestrutura, o IaC utiliza arquivos de configuração ou scripts que descrevem a infraestrutura desejada de forma automatizada e reproduzível.
- A infraestrutura como código é de extrema importância na engenharia de dados, pois permite uma gestão eficiente e escalável dos recursos de infraestrutura necessários para processamento e armazenamento de dados. Com o IaC, é possível definir, provisionar e configurar ambientes de maneira consistente, rápida e controlada. Isso facilita a implantação e manutenção de pipelines de dados, ambientes de desenvolvimento e testes, clusters de processamento distribuído e outros componentes necessários para a engenharia de dados. Além disso, o IaC permite a automação e o versionamento da infraestrutura, proporcionando maior agilidade,

rastreabilidade e garantia de qualidade no gerenciamento dos recursos de TI.

Conteúdos

- **Site** Red Hat: O que é infraestrutura como código (IaC)? (<https://www.redhat.com/pt-br/topics/automation/what-is-infrastructure-as-code-iac>)
- **YouTube** O que é Infrastructure as code? (IaC) Infraestrutura como código (https://youtu.be/_iho2OK_LDg)
- **Artigo** IaC — Infraestrutura como código (<https://medium.com/sysvle/iac-infraestrutura-como-c%C3%B3digo-c514a869b88d>)
- **YouTube** Código Fonte TV: Terraform em 10 Minutos // Dicionário do Programador (<https://www.youtube.com/watch?v=0EAjJe8aPkc>)
- **YouTube** Terraform: infraestrutura como código (<https://www.youtube.com/watch?v=tE1WZg9ib8k>)

Conteúdos Alura:

- **Curso** Curso Infraestrutura como código: preparando máquinas na AWS com Ansible e Terraform (<https://cursos.alura.com.br/course/infraestrutura-codigo-maquinas-aws-ansible-terraform>)
- **Curso** Formação DevOps (<https://cursos.alura.com.br/formacao-devops>)
- **Curso** Formação Infraestrutura como código (<https://cursos.alura.com.br/formacao-infraestrutura-codigo>)