

# Python

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>)

### Python - Fundamentos:

- Python é uma linguagem de programação de alto nível, de uso geral, amplamente utilizada em aplicações web, desenvolvimento de software, ciência de dados e Machine Learning. Sua filosofia de projeto enfatiza a legibilidade do código com o uso de indentação significativa. Python é dinamicamente tipada e tem um garbage collector.
- 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
- Funções anônimas

### Conteúdos

- **Site** Documentação Python (<https://docs.python.org/pt-br/3/tutorial/>)

- **Artigo** Programação | Python — Parte 1 (<https://medium.com/turing-talks/turing-talks-4-python-parte-1-29b8d9efd0a5>)
- **Artigo** Introdução ao Python (<https://medium.com/@goularteduarda.a/introdu%C3%A7%C3%A3o-ao-python-d30c29eba0d6>)
- **YouTube** Eduardo Mendes: Curso introdutório de Python (<https://www.youtube.com/watch?v=yTQDbqmv8Ho>)

### Conteúdos Alura:

- **Artigo** O que é Python? História, Sintaxe e um Guia para iniciar na Linguagem (<https://www.alura.com.br/artigos/python>)
- **Podcast** Hipsters.tech: Python – Hipsters 122 (<https://www.hipsters.tech/python-hipsters-122/>)
- **Podcast** Hipsters.tech: Python Fluente – Hipsters Ponto Tech 179 (<https://www.hipsters.tech/python-fluente-hipsters-ponto-tech-179/>)
- **YouTube** Alura: A linguagem Python - Alura Live 94 (<https://www.youtube.com/watch?v=Nbt0eQHChol>)
- **YouTube** Alura: O que é Python? (<https://www.youtube.com/watch?v=-LATVnPcvHI>)
- **Artigo** Python - Uma Introdução à Linguagem (<https://www.alura.com.br/artigos/python-uma-introducao-a-linguagem>)
- **Artigo** Python: Trabalhando com precisão em números decimais (<https://www.alura.com.br/artigos/precisao-numeros-decimais-python>)
- **Artigo** Listas em Python: operações básicas (<https://www.alura.com.br/artigos/listas-no-python>)
- **Artigo** Como comparar objetos no Python? (<https://www.alura.com.br/artigos/como-comparar-objetos-no-python>)
- **Artigo** Python datetime: Lidando com datas e horários (<https://www.alura.com.br/artigos/lidando-com-datas-e-horarios-no-python>)
- **Curso** Curso Python: crie a sua primeira aplicação (<https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao>)

- **Curso** Formação Python e orientação a objetos  
(<https://cursos.alura.com.br/formacao-linguagem-python>)

## 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/](https://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>)

## Estruturas de Dados:

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

## Conteúdos

- **Artigo** Estrutura de dados (<https://medium.com/php-word/estrutura-de-dados-6afb386a6fc1>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 1 (<https://medium.com/@PaoloProdossimoLopes/01-estrutura-de-dados-e-algoritmo-9cb6a9b14aa>)
- **Artigo** Estrutura de Dados e Algoritmo - parte 2 (<https://medium.com/@PaoloProdossimoLopes/02-estrutura-de-dados-e-algoritmo-deba0b384a0f>)
- **YouTube** Mario Souto - Dev Soutinho: Estruturas de dados com JavaScript (<https://www.youtube.com/watch?v=MweeZn1rR8s>)
- **YouTube** Fabio Akita: Árvores (<https://www.youtube.com/watch?v=9GdesxWtOgs>)
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Lista Encadeada ([https://www.youtube.com/watch?v=RW0oD2L\\_tSg](https://www.youtube.com/watch?v=RW0oD2L_tSg))
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Filas ([https://www.youtube.com/watch?v=RdoLTgJSV\\_c](https://www.youtube.com/watch?v=RdoLTgJSV_c))
- **YouTube** Loiane Groner: Estrutura de Dados com Java | Pilhas (<https://www.youtube.com/watch?v=ZdU4wMyiTSs>)

## Conteúdos Alura:

- **Site** Apostila Python: Estruturas de dados (<https://www.alura.com.br/apostila-python-orientacao-a-objetos/estrutura-de-dados>)
- **Artigo** Estruturas de dados: uma introdução (<https://www.alura.com.br/artigos/estruturas-de-dados-introducao>)
- **Artigo** Curso Estrutura de Dados: computação na prática com Java (<https://www.alura.com.br/artigos/estrutura-dados-computacao-na-pratica-com-java>)
- **Podcast** Hipsters.tech: Algoritmos e estrutura de dados - Hipsters 186 (<https://www.alura.com.br/podcast/hipsterstech-algoritmos-e-estrutura-de-dados-hipsters-186-a375>)

- **Curso** Curso PHP: dominando as Collections  
(<https://cursos.alura.com.br/course/php-dominando-collections>)
- **Curso** Curso Python: crie a sua primeira aplicação  
(<https://cursos.alura.com.br/course/python-crie-sua-primeira-aplicacao>)
- **Desafio** 7 Days of Code: Estruturas de Dados  
(<https://7daysofcode.io/matricula/estruturas-de-dados>)

## Python - 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.
- Utilizar listas e tuplas
- Utilizar polimorfismo nas coleções
- Utilizar conjuntos e dicionários

## Conteúdos

- **Artigo** Coleções em Python — Tuplas  
(<https://medium.com/@desk467/cole%C3%A7%C3%B5es-em-python-tuplas-4444d683963a>)
- **Artigo** Entenda o que é, e quais as diferenças entre Listas, Sets, Tuplas e Dicionários no Python (<https://medium.com/ensina-ai/entenda-o-que-e-e-quais-as-diferencas-entre-listas-sets-tuplas-e-dicionarios-no-python-aa927ed64796>)
- **YouTube** Refatorando: Collections, Lists e Tuples  
(<https://www.youtube.com/watch?v=5gqpc-nCq7w>)
- **YouTube** Refatorando: Coleções - set e dictionary  
(<https://www.youtube.com/watch?v=7uVwpmuw4Yg>)
- **YouTube** Python 42: Coleções no Python | Listas, tuplas, conjuntos e dicionários (<https://www.youtube.com/watch?v=7uVwpmuw4Yg>)

## Conteúdos Alura:

- **Artigo** Compreensão de listas no Python  
(<https://www.alura.com.br/artigos/simplicando-o-processamento-com-compreensao-de-lista-do-python>)
- **Artigo** Ordenando listas no Python  
(<https://www.alura.com.br/artigos/ordenando-listas-no-python>)
- **Artigo** Conhecendo as tuplas no Python  
(<https://www.alura.com.br/artigos/conhecendo-as-tuplas-no-python>)
- **Artigo** Python: Trabalhando com dicionários  
(<https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python>)
- **Curso** Curso Python Collections parte 1: listas e tuplas  
(<https://cursos.alura.com.br/course/python-collections-listas-e-tuplas>)
- **Curso** Curso Python Collections parte 2: conjuntos e dicionários  
(<https://cursos.alura.com.br/course/python-collections-conjuntos-e-dicionarios>)

## Python - 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

- **Site** Documentação Python - Framework de Testes Unitários  
(<https://docs.python.org/pt-br/dev/library/unittest.html>)



- **Artigo** Testes Unitários: Fundamentos e Qualidade de Software!  
(<https://dev.to/wnqueiroz/testes-unitarios-fundamentos-e-qualidade-de-software-5af0>)
- **Artigo** Tutorial: Testes Unitários e Python — Parte I  
(<https://medium.com/mercos-engineering/tutorial-testes-unit%C3%A1rios-e-python-parte-i-bb77182db93f>)
- **Artigo** Testes Unitários e Python — Parte II: Mocks, APIs & Star Wars  
(<https://medium.com/mercos-engineering/testes-unit%C3%A1rios-e-python-parte-ii-mocks-apis-star-wars-ou-como-consumir-uma-api-rest-3a24d153db59>)
- **Artigo** Testes em Python - Parte 1: Introdução  
(<https://dev.to/womakerscode/testes-em-python-parte-1-introducao-43ei>)
- **YouTube** pythonando: Introdução a testes unitários com Python e Django  
(<https://www.youtube.com/watch?v=cEXt8hDyKQw>)
- **YouTube** Programando Com Roger: Criando testes automatizados com Python usando unittest (<https://www.youtube.com/watch?v=F3qKoXicglo>)
- **YouTube** Programador Lhama: Python Conceitos Rápidos - Testes Unitários  
(<https://www.youtube.com/watch?v=CwaHM-KLKC8>)

## 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 - Hipsters 51  
(<https://cursos.alura.com.br/extra/hipsterstech/testes-automatizados-hipsters-51-a535>)
- **YouTube** Alura: Pirâmide de Testes (<https://www.youtube.com/watch?v=gVJ1Lap-JYA>)
- **Curso** Curso Python e TDD: explorando testes unitários  
(<https://cursos.alura.com.br/course/python-tdd-explorando-testes-unitarios>)

## Python - Comunicação com APIs:

- Uma API é uma interface que desenvolvedores de software utilizam para programar a interação com componentes ou recursos de software fora de seu próprio código. Uma definição ainda mais simples é que uma API é a parte de um componente de software que é acessível a outros componentes.
- Entender o que é uma API REST
- Conhecer os comandos básicos de comunicação HTTP
- Entender o que é uma API REST
- Saber fazer requisições autenticadas
- Converter objetos para JSON e vice-versa
- Saber usar as ferramentas do pacote Requests

### Conteúdos

- **Site** Documentação Requests - HTTP para Humanos ([https://requests.readthedocs.io/projects/pt/pt\\_BR/latest/](https://requests.readthedocs.io/projects/pt/pt_BR/latest/))
- **Artigo** Como começar a usar a biblioteca Requests em Python (<https://www.digitalocean.com/community/tutorials/how-to-get-started-with-the-requests-library-in-python-pt>)
- **Artigo** Python Requests & Spotify Web API (<https://medium.com/@pedron.ketlin/python-requests-spotify-web-api-63cce3641c66>)
- **YouTube** Otávio Miranda: Requests em Python - Consumindo uma API Rest usando o módulo Requests ([https://www.youtube.com/watch?v=Qd8JT0bnJGs&ab\\_channel=Ot%C3%A1vioMiranda](https://www.youtube.com/watch?v=Qd8JT0bnJGs&ab_channel=Ot%C3%A1vioMiranda))
- **YouTube** Walisson Silva: Requisições HTTP com o módulo requests | Web Scraping com Python ([https://www.youtube.com/watch?v=XRUdG2msEQ&ab\\_channel=WalissonSilva](https://www.youtube.com/watch?v=XRUdG2msEQ&ab_channel=WalissonSilva))

### Conteúdos Alura:

- **Artigo** Consumindo a API do Github em Python (<https://www.alura.com.br/artigos/consumindo-a-api-do-github-em-python>)

- **Artigo** Como criar APIs em Python usando FastAPI (<https://www.alura.com.br/artigos/como-criar-apis-python-usando-fastapi>)
- **Curso** Curso API com Django 3 (<https://www.alura.com.br/curso-online-api-django-3-testes-documentacao>)

## Nível 2

### Flask:

- Flask é um pequeno framework web escrito em Python. É classificado como um microframework porque não requer ferramentas ou bibliotecas particulares, mantendo um núcleo simples, porém, extensível. Não possui camada de abstração de banco de dados, validação de formulário ou quaisquer outros componentes onde bibliotecas de terceiros pré-existentes fornecem funções comuns. No entanto, o Flask oferece suporte a extensões que podem adicionar recursos do aplicativo como se fossem implementados no próprio Flask.
- Criar aplicações web
- Definir rotas, redirecionamentos e templates
- Validar formulários

### Conteúdos

- **Site** Documentação Flask (<https://flask-ptbr.readthedocs.io/en/latest/>)
- **Artigo** PythOn Flask — Iniciando ao Framework (<https://medium.com/@cfmedeiros/python-flask-iniciando-ao-framework-efd66250add>)
- **Artigo** Introdução ao desenvolvimento web com Python e Flask (<https://medium.com/@urapython.community/introdu%C3%A7%C3%A3o-ao-desenvolvimento-web-com-python-e-flask-636e81b0b884>)
- **YouTube** Eduardo Mendes: Introdução ao Flask ([https://www.youtube.com/watch?v=yxUm\\_QnP4Qo](https://www.youtube.com/watch?v=yxUm_QnP4Qo))
- **YouTube** Eduardo Mendes: Início da construção da API em Python #1 [venv + Flask + Dev Container] (<https://www.youtube.com/watch?v=veYF2g9M6IA>)

## Conteúdos Alura:

- **Artigo** Flask: Instalação (<https://www.alura.com.br/artigos/flask-instalacao>)
- **Artigo** Django ou Flask: Características, semelhanças e diferenças (<https://www.alura.com.br/artigos/django-ou-flask>)
- **YouTube** Alura: O que é Flask? (<https://www.youtube.com/watch?v=I2SdqHcKmnU>)
- **Curso** Formação Começando com Flask (<https://cursos.alura.com.br/formacao-flask>)

## Python - Orientação a Objetos Avançada:

- Mixin é uma classe que fornece implementações de métodos para reutilização por múltiplas classes filhas relacionadas.
- Sobrecarga do operador significa dar significado estendido além de seu significado operacional predefinido.

## Conteúdos

- **Artigo** Python e Herança Múltipla: Como funciona? (<https://medium.com/rafaeltardivo/python-e-heran%C3%A7a-m%C3%BAltipla-como-funciona-876db0449efe>)
- **Artigo** Orientação a objetos de outra forma: Herança múltiplas e mixins (<https://dev.to/acaverna/orientacao-a-objetos-de-outra-forma-heranca-multiplas-e-mixins-31eb>)
- **Site** Acervo Lima: Sobrecarga de operadores em Python (<https://acervolima.com/sobrecarga-de-operador-em-python/>)
- **Artigo** Sobrecarga de operadores e funções no Python 3 (<https://medium.com/@pauloeddias/sobrecarga-de-operadores-e-fun%C3%A7%C3%B5es-no-python-3-2e66b80361b6>)
- **YouTube** Eduardo Mendes: Programação orientada a objetos - Sobrecarga de operadores ([https://www.youtube.com/watch?v=XSvVXso\\_Ukc](https://www.youtube.com/watch?v=XSvVXso_Ukc))
- **YouTube** PETEE UFMG: Orientação a Objetos - Sobrecarga de Operadores (<https://www.youtube.com/watch?v=2FitezEqOH8>)

## Conteúdos Alura:

- **Curso** Curso Python: avance na Orientação a Objetos e consuma API (<https://cursos.alura.com.br/course/python-avance-orientacao-objetos-consuma-api>)

## Django:

- Django é um framework web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção.
- Criar aplicações web
- Entender a arquitetura de uma aplicação feita com Django
- Criar o admin de uma página
- Utilizar templates e rotas
- Criar formulários

## Conteúdos

- **Site** Documentação Django (<https://docs.djangoproject.com/pt-br/4.1/>)
- **Site** MDN Web Docs: Introdução ao Django (<https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introduction>)
- **Artigo** Introdução ao Django — Começando um projeto do zero (<https://medium.com/@thais.ribeiro/introdu%C3%A7%C3%A3o-ao-django-come%C3%A7ando-um-projeto-do-zero-57a953e0614b>)
- **YouTube** Samuel Gonçalves: O que é o Django Framework - Para iniciantes (<https://www.youtube.com/watch?v=NnxhXfkuJIY>)
- **YouTube** Thiago Kaiser: Como funciona o Django - Modelo MTV (<https://www.youtube.com/watch?v=2XaZXJ5P81k>)

## Conteúdos Alura:

- **Artigo** Django: instalação, configuração e escrevendo seu primeiro app (<https://www.alura.com.br/artigos/django-instalacao-configuracao-e-escrevendo-seu-primeiro-app>)

- **Artigo** Django ou Flask: Características, semelhanças e diferenças (<https://www.alura.com.br/artigos/django-ou-flask>)
- **Artigo** Fazendo o deploy de uma aplicação Django (<https://www.alura.com.br/artigos/fazendo-o-deploy-de-uma-aplicacao-django>)
- **Artigo** Django: QuerySets e ORM (<https://www.alura.com.br/artigos/django-query-sets-e-orm>)
- **Artigo** Django: Diferença entre null e blank (<https://www.alura.com.br/artigos/django-diferenca-entre-null-e-blank>)
- **Artigo** Django e Django Rest: Diferenças e aplicações (<https://www.alura.com.br/artigos/django-django-rest-diferencas>)
- **YouTube** Alura: Criando uma API com Django - Hipsters Ponto Talks 11 (<https://www.youtube.com/watch?v=BKChTO8GADk>)
- **YouTube** Alura: Para o que serve o Django? (<https://www.youtube.com/watch?v=vhv82UCIt3k>)
- **Curso** Formação Django (<https://cursos.alura.com.br/formacao-django>)

## Django Rest Framework:

- O Django REST Framework é um conjunto de ferramentas poderosas e flexíveis para a construção de APIs.
- Desenvolver APIs
- Trabalhar com modelos, serializers e views
- Incluir filtros, buscas e ordenação
- Limitar o número de requisições

## Conteúdos

- **Site** Documentação Django Rest Framework (inglês) (<https://www.django-rest-framework.org/>)
- **Artigo** Introdução ao Django Rest Framework (<https://nelziosito.medium.com/introducao-ao-django-rest-framework-47ecb9ae0e6b>)

- **Artigo** API com Python e Django Rest Framework  
(<https://medium.com/@guilhermegoesgarcia/api-com-python-e-django-rest-framework-f01fcb37aea5>)
- **YouTube** Fernando Rodrigues: Django Rest Framework 01 - Introdução  
(<https://www.youtube.com/watch?v=uly58gcUGv8&list=PLsTx8TSx2fHY01FnuxBdwiBiOdZdPGik7>)
- **YouTube** Gregory Pacheco: Django Rest Framework em 30 minutos  
(<https://www.youtube.com/watch?v=gFslGJR5R8I>)

### Conteúdos Alura:

- **Artigo** Django e Django Rest: Diferenças e aplicações  
(<https://www.alura.com.br/artigos/django-django-rest-diferencas>)
- **Artigo** Django Rest Framework: Versionamento  
(<https://www.alura.com.br/artigos/django-rest-framework-versionamento>)
- **Artigo** Django: Upload de imagens em uma API Rest  
(<https://www.alura.com.br/artigos/django-upload-de-imagens-em-uma-api-rest>)
- **Curso** Formação Django REST Framework: crie APIs com Python  
(<https://www.alura.com.br/formacao-django-rest>)

### Python - MVC e MTV:

- MVC e MTV são dois padrões de projeto (design patterns) utilizados para implementar interfaces e aplicações web.
- Entender o padrão MVC
- Entender o padrão MTV
- Compreender a diferença entre os padrões MVC e MTV

### Conteúdos

- **Site** Acervo Lima: Diferença entre os padrões de design MVC e MVT  
(<https://acervolima.com/diferenca-entre-os-padroes-de-design-mvc-e-mvt/>)

- **Artigo** Como funciona a arquitetura MTV (Django)  
(<https://diandrasilva.medium.com/como-funciona-a-arquitetura-mtv-django-86af916f1f63>)
- **YouTube** Thiago Kaiser: Como funciona o Django - Modelo MTV  
(<https://www.youtube.com/watch?v=2XaZXJ5P81k>)
- **YouTube** Rafael Zottesso: Python e Django - Arquitetura Model View Template (MVT) ([https://www.youtube.com/watch?v=Zpl1w\\_Hn0CE](https://www.youtube.com/watch?v=Zpl1w_Hn0CE))

### Conteúdos Alura:

- **Curso** Formação Django (<https://cursos.alura.com.br/formacao-django>)

### Python - Lambdas e Closures:

- Funções lambda nada mais são do que funções anônimas. Enquanto funções normais podem ser criada utilizando def como prefixo, as funções lambda são criadas utilizando lambda.
- Uma closure em Python é um objeto de função interna, uma função que se comporta como um objeto, que se lembra e tem acesso a variáveis no escopo local em que foi criado, mesmo depois que a função externa tenha terminado de ser executada. Também pode ser definida como um meio de ligar dados a uma função sem passá-los como parâmetro.

### Conteúdos

- **Artigo** Entendendo as funções lambda no Python  
(<https://medium.com/@otaviobn/entendendo-as-fun%C3%A7%C3%B5es-lambda-no-python-cbe3c5abb179>)
- **Artigo** Como utilizar a função lambda em Python  
(<https://www.covildodev.com.br/artigo/funcao-lambda-python>)
- **Artigo** Expressões lambda em Python com map, reduce e filter  
(<https://medium.com/horadecodar/express%C3%B5es-lambda-em-python-com-map-reduce-e-filter-a391368ae886>)
- **Artigo** Afinal, o que são Closures?  
(<https://klauslaube.com.br/2011/05/29/afinal-o-que-sao-closures.html>)



- **Artigo** Closures e Decorators em Python (<https://imasters.com.br/back-end/closures-e-decorators-em-python-parte-01>)
- **YouTube** Otávio Miranda: Expressões lambda (funções anônimas) em Python (<https://www.youtube.com/watch?v=EQRsJ2bKvUU>)
- **YouTube** Walisson Silva: Funções Lambda no Python ([https://www.youtube.com/watch?v=Zpl1w\\_Hn0CE](https://www.youtube.com/watch?v=Zpl1w_Hn0CE))
- **YouTube** Eduardo Mendes: Funções, closures e decoradores em Python ([https://www.youtube.com/watch?v=Zpl1w\\_Hn0CE](https://www.youtube.com/watch?v=Zpl1w_Hn0CE))

## Nível 3

### 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](https://www.youtube.com/watch?v=_Oyy5PFOlcU))

## Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Microsserviços – Hipsters Ponto Tech #349 (<https://www.hipsters.tech/techguide-microsservicos-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: Microsserviços autônomos na Usabilla - Hipsters On The Road 10 (<https://www.alura.com.br/podcast/microsservicos-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-microsservicos--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/microsservicos-padroes-projeto>)
- **Curso** Curso Microsserviços: explorando os conceitos (<https://cursos.alura.com.br/course/fundamentos-microsservicos-aprofundando-conceitos>)
- **Curso** Curso Microsserviços na prática: entendendo a tomada de decisões (<https://cursos.alura.com.br/course/Microsservicos-pratica-tomada-decisoes>)

## Contêineres:

- Os contêineres são pacotes de software que contêm todos os elementos necessários para serem executados em qualquer ambiente. Gerenciamento de contêineres é uma área crucial na computação em nuvem e DevOps, que

envolve o uso de tecnologias para automatizar o processo de criação, implantação, escalonamento e monitoramento de contêineres. Contêineres são unidades de software padronizadas que permitem aos desenvolvedores empacotar todas as dependências de um aplicativo (código, bibliotecas, configurações, etc.) em um único pacote. Isso permite que o aplicativo seja executado de forma consistente em qualquer ambiente de infraestrutura.

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

## Conteúdos

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

- **Artigo** Red Hat: O que é Kubernetes (<https://www.redhat.com/pt-br/topics/containers/what-is-kubernetes>)
- **Artigo** Azure: O que é Kubernetes? (<https://azure.microsoft.com/pt-br/topic/what-is-kubernetes/>)
- **Artigo** Kubernetes - Guia de Referência Gratuito (<https://renatogroffe.medium.com/kubernetes-guia-de-refer%C3%A2ncia-gratuito-a2c1be8e7593>)
- **YouTube** Código Fonte TV: Containers // Dicionário do Programador (<https://www.youtube.com/watch?v=-pUZBovqRcU>)
- **YouTube** Código Fonte TV: Kubernetes // Dicionário do Programador (<https://www.youtube.com/watch?v=mVL0nOM3AGo>)
- **YouTube** Insight Lab: Revisão Docker - Curso de Introdução ao Kubernetes (<https://www.youtube.com/watch?v=bcRArpK00OU>)
- **YouTube** Pisani da Arch: Kubernetes - entenda e comece a arquitetar agora mesmo (<https://www.youtube.com/watch?v=XjaqTqm7g58>)

### Conteúdos Alura:

- **Artigo** O que é Kubernetes (<https://www.alura.com.br/artigos/o-que-e-kubernetes>)
- **Artigo** Criando volumes com Docker (<https://www.alura.com.br/artigos/criando-volumes-com-docker>)
- **Artigo** Docker Compose para compor uma aplicação (<https://www.alura.com.br/artigos/compondo-uma-aplicacao-com-o-docker-compose>)
- **YouTube** Alura: O que são Containers? ([https://www.youtube.com/watch?v=jv4\\_sLIGOS0](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/microsservicos-e-kubernetes-na-americanas-com-hipsters-on-the-road-56-a857>)
- **Curso** Formação Orquestração de containers com Kubernetes  
(<https://www.alura.com.br/formacao-kubernetes>)
- **Curso** Curso Docker: criando e gerenciando containers  
(<https://www.alura.com.br/curso-online-docker-criando-gerenciando-containers>)
- **Curso** Curso Kubernetes: Pods, Services e ConfigMaps  
(<https://www.alura.com.br/curso-online-kubernetes-pods-services-configmap>)
- **Curso** Curso Kubernetes: Deployments, Volumes e Escalabilidade  
(<https://www.alura.com.br/curso-online-kubernetes-deployments-volumes-escalabilidade>)
- **Curso** Curso Google Kubernetes Engine (<https://www.alura.com.br/curso-online-google-kubernetes-engine-primeiros-passos>)
- **Curso** Curso Kubernetes na Digital Ocean: gerenciando aplicações containerizadas (<https://www.alura.com.br/curso-online-kubernetes-digital-ocean>)
- **Curso** Curso Amazon EKS: gerenciando aplicações containerizadas com Kubernetes (<https://www.alura.com.br/curso-online-amazon-eks-kubernetes>)
- **Curso** Curso Azure: gerencie Kubernetes com AKS e ACR  
(<https://www.alura.com.br/curso-online-azure-gerencie-kubernetes-com-aks-acr>)

## Python - Tipagem estática:

- Python é uma linguagem dinamicamente tipada, ou seja, não há necessidade de pensar em tipos de dados. Linguagens estaticamente

tipadas (tais como C ou Java) realizam verificações de tipo durante a compilação. Pode parecer mais seguro, pois você pode dizer imediatamente o tipo de parâmetro de cada função.

- Conhecer type hinting

## Conteúdos

- **Artigo** Python 3 e Type Hints (<https://diogommartins.medium.com/python-3-e-type-hints-40e80a9e8214>)
- **YouTube** pythonando: Tipagem estática com Python? (<https://www.youtube.com/watch?v=jvLjANgC-wk>)
- **YouTube** pythonbrasil: Python e Tipagem Estática ([https://www.youtube.com/watch?v=oeacyCi3u\\_o](https://www.youtube.com/watch?v=oeacyCi3u_o))
- **YouTube** Vinta Software: Medalhões, Patos e Protocolos: tipos em Python 3.8 (<https://www.youtube.com/watch?v=UVhGUINj1e0>)

## Conteúdos Alura:

- **Artigo** O que são as tipagens estática e dinâmica em programação (<https://www.alura.com.br/artigos/o-que-sao-as-tipagens-estatica-e-dinamica-em-programacao>)

## Python - Geradores:

- Geradores permitem declarar uma função que se comporta como um iterador, por exemplo, ela pode ser usada em um loop 'for'.
- Criar objetos iteradores
- Usar avaliação preguiçosa
- Executar tarefas simultâneas
- Uso da palavra reservada 'yield'

## Conteúdos

- **Site** Wiki Python Brasil: Utilizando geradores (generators) (<https://wiki.python.org.br/UsandoGenerators>)

- **Artigo** Python Generators (<http://pythonclub.com.br/python-generators.html>)
- **Artigo** Introdução ao Python Generators (<http://blog.dunderlabs.com/introducao-ao-python-generators.html>)
- **YouTube** Otávio Miranda: Geradores, Iteradores e Iteráveis em Python (<https://www.youtube.com/watch?v=gHrc6FLBh20>)
- **YouTube** Eduardo Mendes: Iteradores e Geradores (<https://www.youtube.com/watch?v=Xj5LICeW0m0>)
- **YouTube** Eduardo Mendes: Desvendando o yield e as funções geradoras (<https://www.youtube.com/watch?v=ZjwZ9nfhsk4>)

### **Conteúdos Alura:**

- **Artigo** Conhecendo os geradores no Python (<https://www.alura.com.br/artigos/conhecendo-os-geradores-no-python>)

### **Python - Assíncrono:**

- Na programação assíncrona as funções não são executadas em ordem. Com o assincronismo, podemos interromper do código para conseguirmos alguma outra informação necessária para a continuar a execução. Isso significa que o código espera por uma outra parte do código e, enquanto espera, executa as demais partes.
- Aprender sobre corrotinas
- Corrotinas são generalizações de sub-rotinas. Elas são utilizadas para multitarefas cooperativas onde um processo cede controle voluntariamente, periodicamente ou quando ocioso, a fim de permitir que múltiplas aplicações sejam executadas simultaneamente.
- Lidar com concorrência
- Conhecer o conceito de objetos aguardáveis
- Criar tarefas concorrentemente
- Conhecer a biblioteca 'asyncio'

## Conteúdos

- **Artigo** Async/await: Introdução à programação assíncrona em Python (<https://medium.com/@edytarcio/async-await-introdu%C3%A7%C3%A3o-%C3%A0-programa%C3%A7%C3%A3o-ass%C3%ADncrona-em-python-fa30d077018e>)
- **Artigo** Desvendando programação assíncrona com Python (<https://medium.com/@wisner.oliveira/desvendando-programa%C3%A7%C3%A3o-ass%C3%ADncrona-com-python-87f68fb2e482>)
- **Site** Documentação - Corrotinas (<https://docs.python.org/pt-br/3.8/library/asyncio-task.html>)
- **Site** Acervo Lima: Corrotina em Python (<https://acervolima.com/corrotina-em-python/>)
- **YouTube** JLCP: Programação assíncrona com Python (<https://www.youtube.com/watch?v=ihkihYPI60>)
- **YouTube** OpenLearningB: Função Async com Python (<https://www.youtube.com/watch?v=9rWaaiJaWQs>)
- **YouTube** Eduardo Mendes: Uma introdução histórica à corrotinas ([https://www.youtube.com/watch?v=b6LT6O1s3\\_8](https://www.youtube.com/watch?v=b6LT6O1s3_8))
- **YouTube** Thiago Silva: AsyncIO - Corrotinas, Tarefas e Loop de Eventos (<https://www.youtube.com/watch?v=LKSp0SGWRTk>)
- **YouTube** pythonbrasil: Python Assíncrono - tudo ao mesmo tempo agora (<https://www.youtube.com/watch?v=djj20SpxO4s>)
- **YouTube** pythonbrasil: Corrotinas e Concorrência (<https://www.youtube.com/watch?v=djj20SpxO4s>)

## Python - args & kwargs:

- As variáveis mágicas `*args` e `**kwargs` são normalmente usadas na definições de uma função, e são usadas para passar um número desconhecido de argumentos para uma função.
- Entender a diferença entre `*args` e `**kwargs`



## Conteúdos

- **Site** Acervo Lima: args e kwargs em Python (<https://acervolima.com/args-e-kwargs-em-python/>)
- **Artigo** Python: entendendo o uso de \*args e \*\*kwargs em funções e métodos (<https://medium.com/rafaeltardivo/python-entendendo-o-uso-de-args-e-kwargs-em-fun%C3%A7%C3%B5es-e-m%C3%A9todos-c8c2810e9dc8>)
- **YouTube** Otávio Miranda: Funções (def) em Python - args e kwargs ([https://www.youtube.com/watch?v=\\_Rkr8LsH9BA&ab\\_channel=Ot%C3%A1vioMiranda](https://www.youtube.com/watch?v=_Rkr8LsH9BA&ab_channel=Ot%C3%A1vioMiranda))

## Python - Métodos especiais (dunder):

- Métodos especiais, ou métodos mágicos, em Python são métodos predefinidos em todos os objetos, com invocação automática sob circunstâncias especiais. Eles normalmente não são chamados diretamente pelo usuário mas podem ser overloaded (sobrescritos e alterados). Seus nomes começam e terminam com sublinhados duplos chamados de dunder (uma expressão derivada de double underscore).
- Entender o conceito de métodos especiais (ou mágicos)
- Conhecer os principais métodos mágicos e como usá-los

## Conteúdos

- **Site** Acervo Lima: Métodos mágicos ou Dunder (<https://acervolima.com/personalize-sua-classe-python-com-metodos-magic-ou-dunder/>)
- **YouTube** Eduardo Mendes: Objetos e métodos especiais (dunders) (<https://www.youtube.com/watch?v=syctPjStwQU>)
- **YouTube** pythonando: Faça **mágica** com Python ([https://www.youtube.com/watch?v=yxUm\\_QnP4Qo](https://www.youtube.com/watch?v=yxUm_QnP4Qo))
- **YouTube** pythonando: Orientação a Objetos - Métodos especiais (<https://www.youtube.com/watch?v=H1ZNau44xAl>)

## Conteúdos Alura:

- **Curso** Curso String em Python: extraindo informações de uma URL (<https://cursos.alura.com.br/course/string-python-extraindo-informacoes-url>)
- **Curso** Curso Python: avance na Orientação a Objetos e consuma API (<https://cursos.alura.com.br/course/python-avance-orientacao-objetos-consuma-api>)

## Python - Metaprogramação:

- Metaprogramação é uma técnica de programação com a qual programas têm a capacidade de tratar outros programas como seus dados. Isso significa que um programa pode ser projetado para ler, gerar, analisar ou transformar outros programas, e até mesmo modificar-se mesmo durante a execução.
- Escrever um programa que manipula outros programas
- Usar metaclasses

## Conteúdos

- **Site** Acervo Lima: Metaprogramação com Metaclasses em Python (<https://acervolima.com/metaprogramacao-com-metaclasses-em-python/>)
- **Artigo** Introdução à Metaprogramação em Python 3 (<https://alcarsil.medium.com/introdu%C3%A7%C3%A3o-%C3%A0-metaprograma%C3%A7%C3%A3o-em-python-3-fa6688b08d67>)
- **Artigo** Metaprogramação 101 (<https://guimorg.github.io/metaprogramming-101-pt/>)
- **YouTube** Victor Vernalha: Introdução à Metaprogramação em Python (<https://www.youtube.com/watch?v=S0xdG5Ctluo>)
- **YouTube** pythonbrasil: Atributos, propriedades e descritores - encapsulamento e metaprogramação (<https://www.youtube.com/watch?v=tF4DIra3uNI>)

## Python - Multiprocessamento:

- Em Python, o módulo de multiprocessamento inclui uma API muito simples e intuitiva para dividir o trabalho entre vários processos.
- Executar processos em paralelo
- Conhecer a classe Pool

### Conteúdos

- **Site** Documentação Python - Process Pools (inglês)  
(<https://docs.python.org/pt-br/3/library/multiprocessing.html#module-multiprocessing.pool>)
- **Site** Acervo Lima: Multiprocessamento em Python  
(<https://acervolima.com/multiprocessamento-em-python-conjunto-1-introducao/>)
- **Site** Acervo Lima: Sincronização e pooling de processos em Python  
(<https://acervolima.com/sincronizacao-e-pooling-de-processos-em-python/>)
- **Artigo** Uma pequena introdução a programação paralela e multiprocessamento com Python  
(<https://medium.com/@adson.develop/uma-pequena-introdu%C3%A7%C3%A3o-a-programa%C3%A7%C3%A3o-paralela-e-multiprocessamento-com-python-232bbf72a8f7>)
- **YouTube** Eduardo Mendes: Usando Processos - Multiprocessamento  
(<https://www.youtube.com/watch?v=ANGM6Tjf9zM>)
- **YouTube** Ignorância Zero: Execução Paralela V - Multiprocessing  
(<https://www.youtube.com/watch?v=EGJipyZBjbU>)

### Reflection e atributos:

- Os objetos de Reflection (reflexão) são usados para obter informações do tipo em tempo de execução. As classes que dão acesso aos metadados de um programa em execução estão no namespace System.Reflection.
- Escrever código que lê as informações e metadados de objetos em tempo de execução

- Obter nomes de classes em tempo de execução e criar objetos de uma classe

## Conteúdos

- **Site** Microsoft Docs: Reflection (C#) (<https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/reflection>)
- **Site** Acervo Lima: Reflexão em Python (<https://acervolima.com/reflexao-em-python/>)
- **Artigo** Reflection in C# (inglês) (<https://medium.com/@ghadeer.kenawi/reflection-in-c-fa27d8f827f8>)
- **Artigo** Reflection em Javascript (<https://medium.com/trainingcenter/reflection-em-javascript-73fc0e702e2>)
- **YouTube** Marco Diniz: C# em 5 minutos - Reflection ([https://www.youtube.com/watch?v=EX\\_0lf5zyrU](https://www.youtube.com/watch?v=EX_0lf5zyrU))
- **YouTube** Kerplunk: C# Orientação a Objeto - Reflection (<https://www.youtube.com/watch?v=fl1ChSv2yql>)
- **YouTube** Dev Coffee BR: C# Orientação a Objeto - Reflection (<https://www.youtube.com/watch?v=XkgeNMvZyzo>)

## Conteúdos Alura:

- **Curso** Curso Java Reflection: simplifique a conversão de objetos (<https://cursos.alura.com.br/course/java-reflection-simplifique-conversao-objetos>)
- **Curso** Curso C# Reflection: manipule dinamicamente tipos e assemblies (<https://cursos.alura.com.br/course/csharp-reflection-manipule-dinamicamente-tipos-assemblies>)

# Habilidade Auxiliar: Infraestrutura

## Git e GitHub - Fundamentos:

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

grandes com velocidade e eficiência.

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

## Conteúdos

- **Site** GitHub Documentação (<https://docs.github.com/pt>)
- **Site** GitHub Pages Documentação (<https://docs.github.com/pt/pages/getting-started-with-github-pages/about-github-pages>)
- **Site** Git School - Visualizing Git (<https://git-school.github.io/visualizing-git/>)
- **Site** Dangit, Git!?! (<https://dangitgit.com/>)
- **YouTube** Rafaella Ballerini: O que é Git e GitHub? - definição e conceitos importantes 1/2 (<https://www.youtube.com/watch?v=DqTITcMq68k>)
- **YouTube** Rafaella Ballerini: Como usar Git e GitHub na prática! - desde o primeiro commit até o pull request! 2/2 (<https://www.youtube.com/watch?v=UBAX-13g8OM>)
- **YouTube** Mario Souto - Dev Soutinho: Git: Entendendo de vez como funciona do melhor e mais visual jeito possível (<https://www.youtube.com/watch?v=4-tfJ-ZyA0Q>)
- **YouTube** Mario Souto - Dev Soutinho: Como colocar seu projeto no ar DE GRAÇA via GitHub! | Hospedagem com GitHub Pages ([https://www.youtube.com/watch?v=BU-w2\\_Aae54](https://www.youtube.com/watch?v=BU-w2_Aae54))
- **YouTube** CodandoTV(Rods) - 5 coisas que você precisa saber sobre Git ([https://youtu.be/MqoqPzjQyCY?si=VL\\_mT8EowuemptmU](https://youtu.be/MqoqPzjQyCY?si=VL_mT8EowuemptmU))

- **YouTube** CodandoTV(Rods) - README de Sucesso: Transforme seu Projeto ou seu Perfil em Destaque no GitHub ([https://youtu.be/v9ZM2PVzctM?si=49ah\\_HRPlwpSEX4A](https://youtu.be/v9ZM2PVzctM?si=49ah_HRPlwpSEX4A))

## Conteúdos Alura:

- **Artigo** Git e Github: O que são, Como Configurar e Primeiros Passos (<https://www.alura.com.br/artigos/o-que-e-git-github>)
- **Artigo** Mais git com o hub: a linha de comando do Github (<https://www.alura.com.br/artigos/github-na-linha-de-comando>)
- **Podcast** Hipsters 184: Guia do Iniciante em Github (<https://cursos.alura.com.br/extra/hipsterstech/guia-do-iniciante-em-github-hipsters-184-a378>)
- **Site** GitHub: diferentes maneiras de compartilhar seu projeto (<https://cursos.alura.com.br/extra/alura-mais/github-diferentes-maneras-de-compartilhar-seu-projeto-c2002>)
- **Site** Websérie: Git e Github para Sobrevivência (<https://www.alura.com.br/webseries/git-e-github-para-sobrevivencia>)
- **Podcast** Hipsters 109: Git e Github (<https://www.alura.com.br/podcast/hipsterstech-git-e-github-hipsters-109-a474>)
- **YouTube** Alura: Git e Github para Sobrevivência 01: Como o Git funciona? (<https://www.youtube.com/watch?v=BAmvmaKQklQ>)
- **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](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON))
- **Site** MDN Web Docs: Trabalhando com JSON (<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects/JSON>)
- **Artigo** Introdução ao JSON (<https://juliocarneiro.medium.com/introdu%C3%A7%C3%A3o-ao-json-7825b1a550ff>)
- **YouTube** Marco Bruno: O que é JSON e como criar um objeto (<https://www.youtube.com/watch?v=oCY5YEEjlwE>)

## Conteúdos Alura:

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



## Linha de comando - Fundamentos:

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

### Conteúdos

- **Site** Microsoft Docs: Como usar argumentos de linha de comando para terminal do Windows (<https://docs.microsoft.com/pt-br/windows/terminal/command-line-arguments?tabs=windows>)
- **YouTube** Estevan Maito: Linha de comando básica - 8 comandos mais frequentes - WINDOWS e LINUX (<https://www.youtube.com/watch?v=rKBqXJZocYk>)
- **YouTube** Ninja do Linux: Comandos básicos da linha de comando do Linux ([https://www.youtube.com/watch?v=rs\\_yshFGu8E](https://www.youtube.com/watch?v=rs_yshFGu8E))

### Conteúdos Alura:

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

## Cloud - Fundamentos:

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

## Conteúdos

- **Artigo** Computação em nuvem  
(<https://medium.com/sysadminas/computa%C3%A7%C3%A3o-em-nuvem-515930304cf9>)
- **Artigo** O que é cloud? (<https://gabriel-faraday.medium.com/o-que-%C3%A9-cloud-991109e708c6>)
- **YouTube** Gabs Ferreira: Por que investir e estudar cloud?  
(<https://www.youtube.com/watch?v=Z45BTNeZ1I0>)
- **YouTube** Andre Iacono: O que é MICROSOFT AZURE? Qual Certificação começar em 2022? (<https://www.youtube.com/watch?v=f-oVzkvMwnE>)
- **YouTube** AWS: O que é a AWS? (<https://www.youtube.com/watch?v=8JI9wQ8sUdQ>)
- **YouTube** O que é Google Cloud e por que aprender?  
(<https://www.youtube.com/shorts/Lzq3f1DHWcl>)
- **Artigo** AWS vs Google Cloud vs Azure: o que cada um tem de melhor?  
(<https://medium.com/data-hackers/aws-vs-google-cloud-vs-azure-o-que-cada-um-tem-de-melhor-52107174f7b7>)
- **YouTube** Código Fonte TV: Azure (A plataforma Cloud da Microsoft)  
(<https://www.youtube.com/watch?v=YgE-sZaCuJ0>)

- **YouTube** Mundo da Cloud: AWS do Zero ao Expert  
(<https://www.youtube.com/watch?v=HiBCv9DolxI&list=PLtL97Owd1gkQ0dfqGW8OtJ-155Gs67Ecz>)

## Conteúdos Alura:

- **Podcast** Hipsters.tech: TechGuide - Fundamentos Cloud – Hipsters Ponto Tech #348 (<https://www.hipsters.tech/techguide-fundamentos-cloud-hipsters-ponto-tech-348/>)
- **Artigo** Cloud: o que é, História e Guia da computação em nuvem  
(<https://www.alura.com.br/artigos/cloud>)
- **Podcast** Hipsters.tech: Uma jornada Para o Cloud - Hipsters Deep Dive 005  
(<https://www.alura.com.br/podcast/uma-jornada-para-o-cloud-hipsters-deep-dive-005-a1100>)
- **Podcast** Hipsters.tech: Histórias do Cloud - Hipsters 04  
(<https://www.alura.com.br/podcast/historias-do-cloud-hipsters-04-a582>)
- **Artigo** Heroku, Vercel e outras opções de cloud como plataforma  
(<https://www.alura.com.br/artigos/heroku-vercel-outras-opcoes-cloud-plataforma>)
- **Artigo** AWS: Guia sobre o que é Amazon Web Services, seus Serviços e Certificações (<https://www.alura.com.br/artigos/aws>)
- **Artigo** Terraform: criando máquinas na Azure  
(<https://www.alura.com.br/artigos/terraform-maquinas-na-azure>)
- **YouTube** Alura: O que é cloud? (<https://www.YOUTUBE.com/watch?v=wev9fMrg-TU>)
- **YouTube** Alura: AWS, Google Cloud e Azure: Por onde começar? | Hipsters.Talks (<https://www.YOUTUBE.com/watch?v=z9k6rsdmWc0&t=300s>)
- **YouTube** Alura: Certificação em Cloud: Azure, AWS, Google | Hipsters.Talks  
([https://www.YOUTUBE.com/watch?v=W4K82n\\_WK5g&t=290s](https://www.YOUTUBE.com/watch?v=W4K82n_WK5g&t=290s))
- **Curso** Formação Começando em Cloud Computing  
(<https://cursos.alura.com.br/formacao-cloud-computing>)

- **Curso** Formação Amazon Web Services  
(<https://cursos.alura.com.br/formacao-amazon-web-services>)
- **Curso** Formação Google Certified Associate Cloud Engineer  
(<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Certificação AWS Certified Cloud Practitioner  
(<https://cursos.alura.com.br/formacao-aws-certified-cloud-practitioner>)
- **Curso** Formação Containers com AWS ECS e EKS  
(<https://cursos.alura.com.br/formacao-containers-aws>)
- **Curso** Formação Google Cloud Platform  
(<https://cursos.alura.com.br/formacao-google-cloud>)
- **Curso** Formação Certificação Google Certified Associate Cloud Engineer  
(<https://cursos.alura.com.br/formacao-google-certified-associate-cloud-engineer>)
- **Curso** Formação Azure (<https://cursos.alura.com.br/formacao-conhecendo-azure>)
- **Curso** Formação Certificação AZ-900: Microsoft Azure Fundamentals  
(<https://cursos.alura.com.br/formacao-certificacao-az-900-microsoft-azure-fundamentals>)

## 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](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 e ferramentas

### 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](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](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-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-architecture>)



## 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>)

## Jupyter & Colab notebooks:

- Jupyter Notebook e Google Colaboratory são Notebooks que permitem a criação de blocos de texto e blocos de código
- Os Notebooks facilitam a elaboração de projetos de Data Science por ser possível visualizar o resultado da execução logo após o trecho de código
- O Google Colaboratory permite escrever e executar códigos Python diretamente no navegador, sem nenhuma ou poucas configurações necessárias
- Essas ferramentas facilitam o compartilhamento de projetos entre o time

## Conteúdos

- **Site** Jupyter Notebook (<https://jupyter.org/>)
- **Site** Google Colaboratory (<https://colab.research.google.com/>)
- **Artigo** Maximize sua eficiência: 10 passos para organizar seu Jupyter Notebook (<https://medium.com/data-hackers/maximize-sua-eficiencia-10-passos-para-organizar-seu-jupyter-notebook-522b4bdf8261>)

## Conteúdos Alura:

- **Artigo** Jupyter Notebook: Exemplos de Códigos e Como Usar (<https://www.alura.com.br/artigos/conhecendo-o-jupyter-notebook>)
- **Artigo** Google Colab: o que é, tutorial de como usar e criar códigos (<https://www.alura.com.br/artigos/google-colab-o-que-e-e-como-usar>)
- **YouTube** Trabalhando com arquivos no Google Colab (<https://www.youtube.com/watch?v=ojgvdqexJ0Q&t=3s>)

## Extração e Tratamento de Dados:

- A extração de dados é o processo de coleta ou recuperação de tipos diferentes de dados de uma variedade de fontes, muitos dos quais podem estar mal organizados ou completamente desestruturados.
- Obter os dados que serão analisados
- Tratar os dados obtidos, transformando-os, alterando sua estrutura e valores a fim de deixar a base de dados mais coerente e garantir que os dados que serão trabalhados estejam nas melhores condições para serem analisados
- Utilizar as bibliotecas Pandas e Scikit-learn para tratar os dados

## Conteúdos

- **Artigo** Tratamento e Transformação de Dados NaN: Uma visão geral e prática (<https://medium.com/data-hackers/tratamento-e-transforma%C3%A7%C3%A3o-de-dados-nan-uma-vis%C3%A3o-geral-e-pr%C3%A1tica-54efa9fc7a98>)
- **YouTube** Awari: Importação e Tratamento de Dados em Data Science (<https://www.youtube.com/watch?v=i1TK8pWRQas>)
- **YouTube** Celso Cestaro: Iniciando em coleta e tratamento de dados com Python (<https://www.youtube.com/watch?v=3DF54MnH9KM>)
- **YouTube** Celso Cestaro: Python para data science - limpando dados (<https://www.youtube.com/watch?v=t18Ogc8rNyo>)
- **Artigo** Como fazer uma limpeza de dados completa em Python (<https://medium.com/turing-talks/como-fazer-uma-limpeza-de-dados-completa-em-python-7abc9dfc19b8>)
- **YouTube** xavercoding: Manipulação de Dados em Python/Pandas (<https://www.youtube.com/watch?v=qKeM93NTKQM>)
- **YouTube** Bix Tecnologia: Tratamento de Dados com Python (Webinar) (<https://www.youtube.com/watch?v=slv1f2YpMJE>)

## Conteúdos Alura:

- **Artigo** Pandas: o que é, para que serve e como instalar  
(<https://www.alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar>)
- **Curso** Curso Pandas: conhecendo a biblioteca  
(<https://cursos.alura.com.br/course/pandas-conhecendo-biblioteca>)
- **Curso** Curso Pandas I/O: trabalhando com diferentes formatos de arquivos  
(<https://cursos.alura.com.br/course/pandas-io-trabalhando-diferentes-formatos-arquivos>)
- **Curso** Curso Pandas: seleção e agrupamento de dados  
(<https://cursos.alura.com.br/course/pandas-selecao-agrupamento-dados>)
- **Curso** Curso Pandas: transformação e manipulação de dados  
(<https://cursos.alura.com.br/course/pandas-transformacao-manipulacao-dados>)
- **Desafio** 7 Days of Code: Python Pandas  
(<https://7daysofcode.io/matricula/pandas>)