

Semana 4

Objetivo: Aprofundando na linguagem Python e em conceitos de Data.

Conteúdo: Python:

1. Orientação a Objetos com Python
2. ETL
3. ETL em Python
4. Cursos

Desafio: Realizar os cursos e responder todos os exercícios propostos neste material



Orientação a Objetos com Python

O que é do Orientação a Objetos?

De maneira introdutória, orientação a objetos consiste em programação orientada a objetos e como o próprio nome já expõem são linguagens e/ou técnicas de programação que possibilitam a utilização objetos. Breve e **resumidamente** serão introduzidos alguns dos diversos conceitos de OO:



Polimorfismo:

Se dá quando em uma hierarquia de classes existe a implementação métodos com a mesma assinatura, ou seja, o código funciona para qualquer classe desta hierarquia removendo a necessidade de implementações específicas para cada uma das classes.

Herança:



Existe quando uma classe herda os mesmos atributos e mesmos métodos, por exemplo, Cachorro e Gato são objetos da classe de mamíferos.

Encapsulamento:

O Encapsulamento se baseia no conceito de “esconder” os atributos de uma classe, ou seja, para quem for usar a classe não a utilizar de forma incorreta.

Benefícios da OO

- Diminui a complexidade para entendimento e manutenibilidade
- Orientação a Objetos facilita a reusabilidade de códigos, evitando a duplicidade.
- Facilita e acelera o desenvolvimento paralelo de classes e programas.
- Programas Orientados a Objetos são mais fáceis de testar, gerenciar e “*debuggar*”.

Desvantagens da OO

- Possível superficialidade entre classes.
- Planejamento prévio e conhecimentos sobre programação são necessários.
- Muitas vezes classes acabam sendo generalizadas.

Curso Indicado

- Este curso irá introduzir conceitos que irão auxiliar na resolução dos exercícios adiante:
 - [Python 3: introdução a Orientação a objetos](#)

Exercícios

Exercício 1

Implemente duas classes "Pato" e "Pardal" que herdem de uma classe "Passaro" a habilidade de "voar" e "emitir som", porém, tanto "Pato" quanto "Pardal" devem emitir sons diferentes (de maneira escrita) no console.

Exemplo: Pato voa e faz “*quack quack*” e Pardal voa e faz “*piu piu*”.



Exercício 2

Crie uma classe "Pessoa" que tenha um atributo privado "nome" e um atributo publico "id".

Crie uma função que atribua um valor a "nome" e uma função que retorne o valor de "nome".

Importante: Para atributos privados utilizamos “__” Ex: __atributo

Exercício 3

Crie uma classe "Calculo" que contenha um metodo que aceita dois parametros, X e Y, e retorne a soma dos dois. Nessa mesma classe, implemente um metodo de subtração, que aceita dois parametros, X e Y, e retorne a subtração dos dois (resultados negativos são permitidos).

Exercício 4

Crie uma classe "Ordenadora" que cotenha um atributo "listaBaguncada" e que contenha os metodos "ordenacaoCrescente" e "ordenacaoDecrescente".

Instancie um objeto chamado 'crescente' dessa classe "Ordenadora" que tenha como "listaBaguncada" a lista [3,4,2,1,5] e instancie um outro objeto, 'decrescente' dessa mesma classe com uma outra "listaBaguncada" sendo [9,7,6,8].

Para o primeiro objeto citado, use o metodo "ordenacaoCrescente" e para o segundo objeto, use o metodo "ordenacaoDecrescente".

Exercício 5

Crie uma classe "Avião" que possua os atributos “modelo”, "velocidade_maxima", “cor” e “capacidade”.

Defina o atributo de sua classe cor, de maneira que todos as instâncias de sua classe avião sejam da cor “azul”.

Após isso, a partir de entradas do teclado (inputs), instancie e armazene em uma lista 3 objetos da classe avião.

Ao final, itere pela lista imprimindo cada um dos objetos no formato:

“O avião de modelo “x” possui uma velocidade máxima de “y”, capacidade para “z” passageiros e é da cor “w”.

Sendo x, y, z e w cada um dos atributos da classe “Avião”.



Extração, Transformação, Carregamento (ETL)

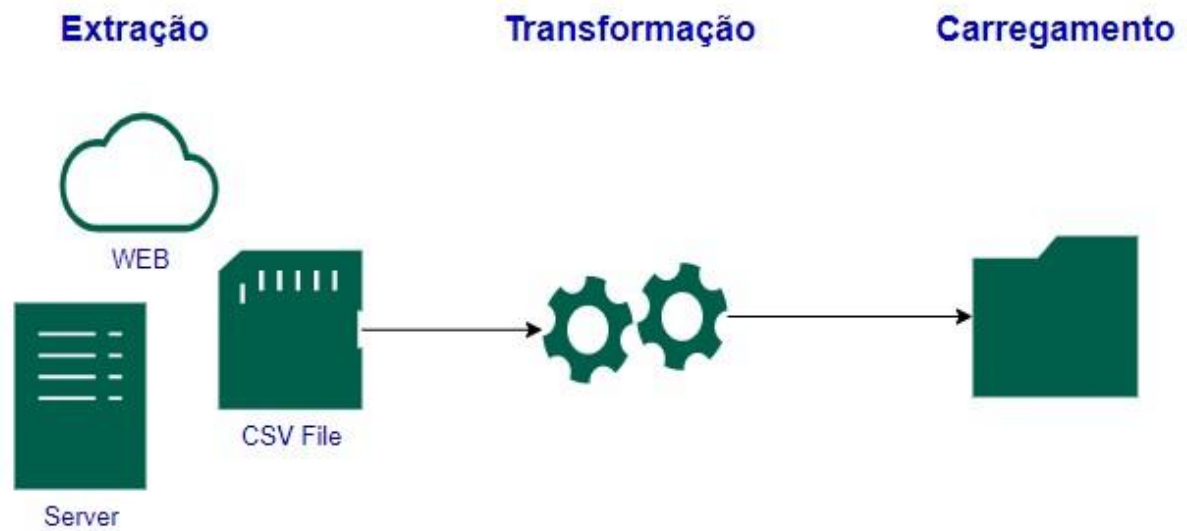
Como pode ser percebido pela primeira letra da sigla ETL, o primeiro processo a ser conduzido é a extração dos dados, ou seja, buscar os dados que devem ser armazenados na *staging area* (ou *landing zone*) de maneira que eles possam passar pelas transformações necessárias.

Dentre essas manipulações, diversas são usuais dentro do cenário da área de dados como por exemplo remoção de duplicatas, combinação de dados de fontes diversas para formação de novos *datasets* (*Conjuntos de dados*) e correção erros ortográficos por exemplo. Todas essas manipulações de dados citadas acima fazem parte dos processos de transformações que são os passos anteriores ao *Load* do ETL.

No último passo deste processo, ocorre o *Loading* dos dados aferidos. Neste estágio é de extrema importância assegurar a qualidade e integridade dos dados a serem publicados, de maneira que o usuário receba essa nova gama de informação com o resultado desejado.

O processo de ETL possui teorias mais extensas e essa introdução foi apenas uma forma de iniciar o assunto, no decorrer do programa de bolsas, os demais conceitos e exemplos serão apresentados com cursos e exercícios.





Desafio Python

Armazene o arquivo actors.csv dentro de uma nova pasta, após isso crie 5 arquivos do tipo “txt” vazios (1 para cada exercício do desafio).

Em seguida para cada um dos enunciados abaixo, leia o arquivo actors.csv utilizando Python como linguagem de programação e depois de obter as repostas necessárias armazene cada um dos resultados em um dos arquivos “txt” criados:

1. O ator/atriz com maior número de filmes e o respectivo número de filmes.
2. A média do número de filmes por autor.
3. O ator/atriz com a maior média por filme.
4. O nome do(s) filme(s) mais frequente(s) e sua respectiva frequência.
5. O Autor que foi mais bem pago.



Pontos de Atenção:

- Para desenvolvimento deste exercício, não deve ser utilizado as bibliotecas Pandas e NumPy e/ou outras bibliotecas e engines que utilizam de dataframes.
- Todas as transformações que julgarem necessárias, devem ser feitas utilizando os scripts Python e nenhuma modificação deve ser feita no arquivo actors.csv
- Para leitura do arquivo actors.csv, não deve ser utilizado o módulo csv nativo do Python.

