



Curso completo de Python!





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite:

<http://vmzsolutions.com.br>



Classes

Nossas classes, teremos 4 arquivos:

A blue icon representing a file, with a folded top-right corner and a folded bottom-left corner.

Client.py

A blue icon representing a file, with a folded top-right corner and a folded bottom-left corner.

Product.py

A blue icon representing a file, with a folded top-right corner and a folded bottom-left corner.

Service.py

A blue icon representing a file, with a folded top-right corner and a folded bottom-left corner.

ServiceDao.py



Classes

Na classe **Client.py**, necessitamos apenas dos atributos que representam um cliente.

```
class Client:
    def __init__(self, name="", cpf="", segment=""):
        self._name = name
        self._cpf = cpf
        self._segment = segment
```



Classes

Na classe **Product.py**, necessitamos apenas dos atributos que representam um produto.

```
class Product:
    def __init__(self, name="", description="", image=""):
        self._name = name
        self._description = description
        self._image = image
```



Classes

Na classe **Service.py**, necessitamos apenas dos atributos que representam um serviço.

```
class Service:
    def __init__(self, client="", product="", status=""):
        self._client = client
        self._product = product
        self._status = status
```



Classes

Na classe **ServiceDao.py**, necessitamos efetuar as operações do Banco de Dados referente aos serviços:

```
class ServiceDao:
```

```
    def __init__(self,service=""):
        self._service = service
```

A classe **ServiceDao** deve ter como parâmetro do contrutor um objeto da classe Service.



Classes

Nele, também precisamos fazer um método `get()` que irá executar a busca desse serviço em nosso BD, nos permitindo assim passar os devidos dados como parâmetro para criar esse serviço.

```
def get(self):  
    try:  
        service =  
session.query(ServiceTable, Client, Product).join(Client, Product).filter(  
ServiceTable.id==self._service._id).first()
```




Classes

Já o nosso Model.py, iremos reutilizar muito do código feito nas aulas de SQLAlchemy.