



# **Curso completo de Python!**





**Instrutor: Vitor Mazuco**

**<http://facebook.com/vitormazuco>**

**Email: [vitor.mazuco@gmail.com](mailto:vitor.mazuco@gmail.com)**

**WebSite:**

**<http://vmzsolutions.com.br>**



## Introdução a seção – Construindo uma API com Flask



# Flask

web development,  
one drop at a time

mongo  
engine



## Entendendo o Model

O Model é uma aplicação semelhante ao modelo de seu banco de dados, e que pode ser tanto em SQL ou em noSQL. Um BD é usado para dar persistência aos seus dados para que sejam usados posteriormente. Assim como a nossa API, podemos a princípio controlar apenas usuários e seus grupos. Logo precisamos de 2 entidades, Usuários e Grupos.



## Entendendo o Model

Quando o assunto é BD, temos que lembrar de cardinalidade e relacionamento entre os dados. Ex: 1 Grupo para N Usuários. Já vimos isso em ORM SQLAlchemy e essa lógica tem apenas em BD que são relacionais. Ou seja apenas fazem associações dos dados entre as tabelas pelas chaves primárias e estrangeiras.



## Entendendo o Model

O noSQL trabalha da mesma maneira que o SQL, mas temos que trabalhar fazendo isso com o uso de subdocumentos colocando dentro de uma lista em um documento principal.

Ex:

Grupos:

Nome:Analistas

Integrantes:[Vitor,Leandro,Carlos e Gaby]



## Entendendo o Model

No mongoDB ficaria assim:

```
{  
  
  "_id": 1,  
  "nome": "Analistas",  
  "integrantes": [{"nome": "Vitor"}, {"nome": "Gaby"},  
                  {"nome": "Carlos"},  
                  {"nome": "Leandro"}]  
  
}
```



## Entendendo o Model

Lembrando que o MongoDB trabalha com o armazenamento de dados no formato do tipo Json. Agora já sabemos que é preciso de um documento chamado de Grupos e um subdocumento chamado de Integrantes.





## Entendendo o Model

O Grupos necessita apenas de duas informações: nome e integrantes. E o Subdocumento Integrantes, apenas de: nome e e-mail. Agora, é necessário ser refletido pelo Model.