

# Docker

APRENDA DE FORMA PRÁTICA E OBJETIVA A  
TRABALHAR COM CONTAINERS DOCKER E  
PUBLICAR SUAS APLICAÇÕES COM EFICIÊNCIA

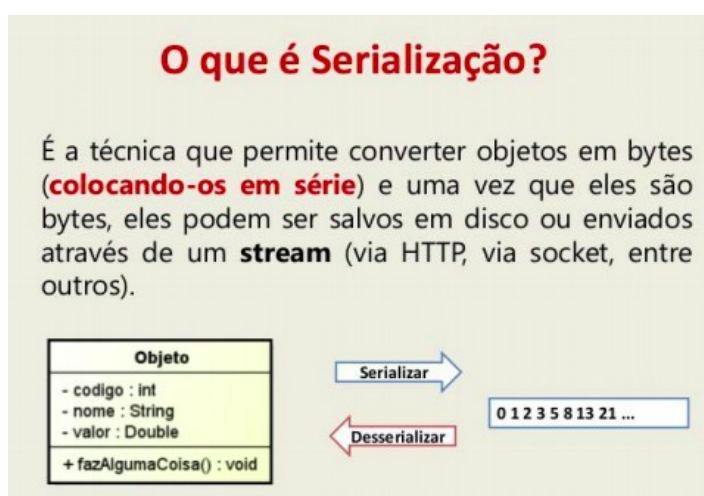
01

Raíssa Azevedo

## YAML - Yet Another Markup Language

No Docker são escritos arquivos de configuração para serem executados, e estes arquivos são escritos nessa linguagem de marcação YAML.

Yet Another Markup Language, criada em 2001 é uma linguagem não apenas de marcação, como o nome sugere, mas sim uma linguagem de **Serialização de Dados**.



A utilização do YALM é muito parecida com o JSON do JavaScript.

## A linguagem YAML:

- Se integra com outras linguagens, como Python, Ruby, Java, etc;
- Possui tipos de dados comuns como escalares, listas, arrays, etc;
- Comumente utilizada como arquivo de configuração ou armazenamento de dados;

## Tem como objetivos:

- Ser lida facilmente por humanos;
- Portátil;
- Integrar facilmente com outras linguagens
- Fácil de implementar e usar;

## Sintaxe:

Identar os códigos em outras linguagens é feito com o uso de **tag** no YAML é feito através do uso de **espaço**.

```
host: geek
datacenter:
  location: Vila Velha
  router: 42
roles:
  - web
  - dns
```

## Estilos:

Em YAML existem 2 estilos para escrever, estes são: block e flow.

O estilo **block** é mais fácil de ler, menos compacto e foi utilizado no exemplo acima, de sintaxe.

O estilo de fluxo ou **flow** é uma extensão do formato Json. Quebra longas linhas de conteúdo, e utiliza tags e âncoras.

```
host2: "geek"
datacenter2: {location:
  Vila Velha, router: 42 }
roles: [ web, dns ]
```

## Mapeamentos:

Mapeamentos em YAML são realizados através de chave e valor. É feito através de dois pontos e espaço.

```
host: geek
datacenter:
  location: São Paulo
  router: 42
```

## Sequências:

São feitos através de listas, coleções ou arrays. As sequências não podem ser vazias ou com linhas em branco.

```
roles:  
- webservice  
- database
```

## Escalares:

Valores escalares são valores únicos (string, números...). Diferente de sequencia de valores (arrays, coleções...).

```
hosts: geek  
datacenter:  
  location: São Paulo  
  router: "42"  
  switch: "34"
```

**Quando se tem um valor inteiro que não é usado em nenhum momento para fazer computação. Declare como *string*.**

Existem tipos que preserva e que não preserva a estrutura das linhas:

```
funcionamento: |  
segunda - das 08:00 as 17:00  
terça   - das 09:00 as 18:00  
quarta  - das 10:00 as 19:00  
quinta  - das 11:00 as 20:00  
sexta   - das 12:00 as 21:00  
sábado  - das 13:00 as 22:00  
domingo - das 14:00 as 23:00
```

Este é o valor escalar de múltiplas linhas, preservando a estrutura das linhas.

```
comentarios: >  
Temos alto uso de CPU  
desde 04/10/2020.  
Estamos investigando.
```

Este é o valor escalar onde a estrutura das linhas não será respeitada. Tudo será apresentado em uma linha só.

**Obs:** Ou seja, | ocorre a preservação da estrutura de linhas e o > não preserva.

### Estruturas:

Em YAML pode haver múltiplas diretivas ou configurações em um único arquivo.

```
---
host: geek
datacenter:
  location: Vila Velha
  router: 42
roles:
  - web
  - dns
---
host: udemy
datacenter:
  location: Colorado
  router: 23
roles:
  - web
  - dns
```

**Sem o ---** o arquivo indicaria erro de chave duplicada.

### Comentários:

Os comentários são feitos através de # assim como o Python.

```
# Configurações do Host Geek:
---
host: geek
datacenter:
  location: Vila Velha # Cidade do ES
  router: 42
roles:
  - web
  - dns
```

Para comentar ou descomentar várias linhas: Selecionar as linhas e clicar **ctrl** + /

### Tags:

São utilizadas para configurar um **uri** ou **url** própria. Assim como marcadores locais e tipos de dados.

```
%TAG ! tag:host:geek
---
host: geek
datacenter:
  location: !GEEK São Paulo
  router: "42"
  switch: '34'
```

Na definição da **tag** geek o processo é feito em letras minúsculas, quando a tag vai ser utilizada no processo ela é usada em letras maiúsculas.

Outra forma (mais comum) de configurar tags:

```
host: udemy
datacenter:
  location: Colorado
  router: !!str 42
  switch: !!str 34 # Ou !!int ou !!float
```

### Âncoras:

Ancôras são utilizadas para armazenar ou reutilizar dados. É definido por **&** e utilizada com **\***

```
---
host: geek
datacenter:
  location: &SP São Paulo
---
host: Udemy
datacenter:
  location: *SP
---
roles: &ghost
- webserver
- database
---
roles: *ghost
```

Dessa forma, é possível declarar uma vez, e reutilizar quantas vezes forem necessárias: