

# Documentação Fila Kafka

---

**Faculdade:** Faculdade UMFG

**Curso:** Analise e Desenvolvimento de Sistemas

**Período:** 5º

**Ano:** 2024

---

**Disciplina:** Comunicação de Dados

**Docente:** Pedro Matias Araujo

**Discente:** Gustavo Alfredo Correa Da Silva

**R.A:** 1379

---

**Repositório:** <https://github.com/Gustavocorrea0/fila-kafka>

---

## Sumário

1.0	Função.....	2
2.0	Requisitos .....	2
3.0	Estrutura do projeto .....	2
4.0	Como fazer Download do projeto .....	3
5.0	Como iniciar projeto .....	3
6.0	Como utilizar projeto com CURL .....	4
7.0	Como utilizar projeto com POSTMAN.....	4

## 1.0 Função

Receber e enviar mensagens sobre dados de veículos.

## 2.0 Requisitos

### 2.1 - Ferramentas:

- Visual Studio Code
- Terminal Linux
- Postman (Opcional)

### 2.2 - Tecnologias:

- Kafka
- Node.js
- Python
- Docker

## 3.0 Estrutura do projeto

A estrutura do projeto é composta pela seguinte formatação:



**Docker-compose.yml** – Faz a controle e organização dos contêineres criados nas subpastas, tornando mais simples o processo de gerenciamento e configurações de processos.

### Pasta 'consumer':

app.js - Responsável por criar um cliente Kafka e consumir mensagens de um tópico específico, além de criar, um tópico Kafka de forma automatizada.

Dockerfile – Responsável por criar a imagem Docker, que configura e prepara o ambiente necessário para a execução da parte do consumidor em um container.

package.json – Guarda as dependências do node.js

## Pasta 'producer':

app.py - Responsável por implementar o produtor Kafka que recebe requisições REST e produz mensagens na fila do Kafka. Configura e executa o servidor Flask, que define as rotas necessárias para requisições REST. Configura o produtor Kafka, utilizando a biblioteca kafka-python, necessária para criar um cliente Kafka. Além de Definir rotas para a produção e envio de mensagens.

Dockerfile – Responsável por criar a imagem Docker, que configura e prepara o ambiente necessário para a execução da parte do produtor em um container.

requirements.txt – Define as instalações necessárias para o container.

## 4.0 Como fazer Download do projeto

1 – Acesse o repositório do projeto em: <https://github.com/Gustavocorrea0/fila-kafka.git>

2 – Clone o repositório com o comando de terminal: **git clone https://github.com/Gustavocorrea0/fila-kafka.git**

## 5.0 Como iniciar projeto

1 – Na pasta principal(projeto-kafka) abra o terminal e digite: **sudo docker-compose up --build**

2 – Após a realização do build do projeto, este deve ser o resultado:

```
projeto-kafka-consumer-1 | Tópico criado com sucesso: []
```

3 – Para confirmar que o projeto foi iniciado, liste os containers ativos, abra outro terminal e digite: **sudo docker ps**

\*Resultado sem iniciar o projeto\*

```
gustavo@gustavo-VirtualBox:~/zEstudos/Kafka/projeto-kafka$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
```

\*Resultado com o projeto iniciado\*

```
gustavo@gustavo-VirtualBox:~/zEstudos/Kafka/projeto-kafka$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
1a04a53d3ef6   projeto-kafka-producer   "python app.py"        4 minutes ago   Up 55 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   projeto-kafka-producer-1
6816aa1e2629   projeto-kafka-consumer   "docker-entrypoint.s..." 4 minutes ago   Up 55 seconds   0.0.0.0:9092->9092/tcp, :::9092->9092/tcp   projeto-kafka-consumer-1
7e56eea8d914   wurstmeister/kafka:2.13-2.7.0 "start-kafka.sh"        4 minutes ago   Up 56 seconds   0.0.0.0:2181->2181/tcp, :::2181->2181/tcp   projeto-kafka-kafka-1
2e721586f31b   wurstmeister/zookeeper:3.4.6 "/bin/sh -c '/usr/sb..." 5 minutes ago   Up 57 seconds   22/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp   projeto-kafka-zookeeper-1
```

## 6.0 Como utilizar projeto com CURL

1 - Abra um novo terminal ou utilize o anterior (de listagem de container) e digite (altere os campos para os dados desejado): **curl -X POST http://localhost:5000/produce -H "Content-Type: application/json" -d '{"make":"<marca>", "model":"<modelo>", "year":<ano>, "color":"<cor>"}**

2 - Este será o retorno no terminal atual:

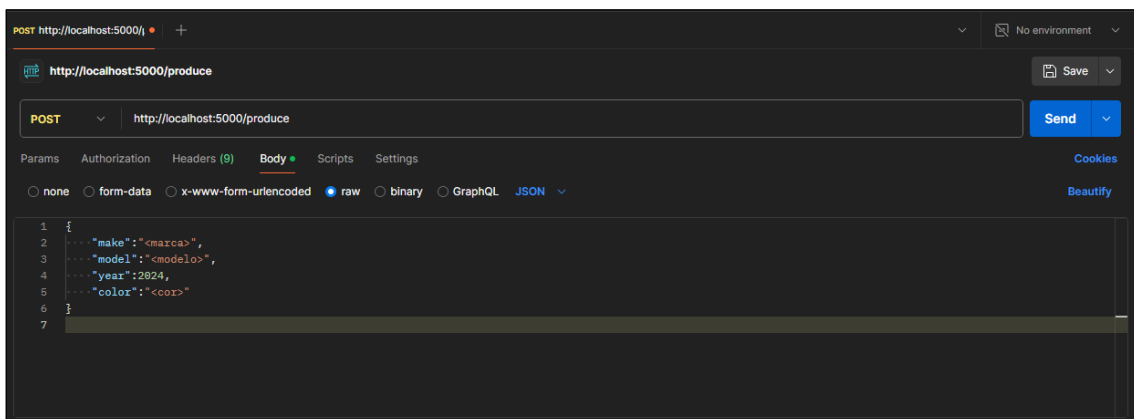
```
gustavo@gustavo-VirtualBox:~/zEstudos/Kafka/projeto-kafka$ curl -X POST http://localhost:5000/produce -H "Content-Type: application/json" -d '{"make":"BMW", "model":"M3", "year":2020, "color":"PRETO"}'
{"status":"Message sent"} RETORNO
```

No terminal o build foi realizado este será o retorno:

```
projeto-kafka-producer-1 | 172.20.0.1 - - [03/Jun/2024 00:08:47] "POST /produce HTTP/1.1" 200 -
projeto-kafka-consumer-1 | Consumed message: {"make": "BMW", "model": "M3", "year": 2020, "color": "PRETO"}
```

## 7.0 Como utilizar projeto com POSTMAN

- 1 – Com o projeto iniciado, inicie o postman e crie uma requisição POST
- 2 – Adicione a URL(<http://localhost:5000/produce>)
- 3 – Clique em Body
- 4 – Selecione “raw”
- 5 – O formato de arquivo será “JSON”
- 6 – Clique em “Send”.



7 - Os resultados serão iguais aos do processo com curl:

```
projeto-kafka-producer-1 | 172.20.0.1 - - [03/Jun/2024 00:08:47] "POST /produce HTTP/1.1" 200 -
projeto-kafka-consumer-1 | Consumed message: {"make": "BMW", "model": "M3", "year": 2020, "color": "PRETO"}
```