

# **Relatório**

**João Carlos de Sousa Fé, Joanny Eva P. Monteiro**

Departamento de Sistemas de Informação  
Universidade Federal do Piauí (UFPI) – Picos, PI – Brasil

joaocarlos.losfe@gmail.com, joannypacheco@hotmail.com

## ***Resumo.***

*Este trabalho tem como objetivo demonstrar a execução de container docker simulando a execução de serviços de forma distribuída. É composto por um produtor de mensagens, uma sistema de filas, um consumidor e um banco de dados, onde cada é executado em seu container de forma independente.*

27/06/2023, Picos-PI.

## 1. Introdução

Docker é um conjunto de produtos de plataforma como serviço que usam virtualização de nível de sistema operacional para entregar software em pacotes chamados contêineres. Os contêineres são isolados uns dos outros e agrupam seus próprios softwares, bibliotecas e arquivos de configuração.

Com o Docker, é possível lidar com os containers como se fossem máquinas virtuais modulares e extremamente leves. Além disso, os containers oferecem maior flexibilidade para criar, implantar, copiar e migrar um container de um ambiente para outro.

A tecnologia Docker usa o kernel do Linux e funcionalidades do kernel, como **cGroups** e **namespaces**, para segregar processos. Assim, eles podem ser executados de maneira independente. O objetivo dos containers é criar independência: a habilidade de executar diversos processos e apps separadamente para utilizar melhor a infraestrutura e, ao mesmo tempo, manter a segurança que você teria em sistemas separados.

## 2. Sistema construído

O projeto resume-se na construção de serviços distribuídos utilizando containers docker. O sistema construído tem como finalidade a consulta de dados de uma cidade através do CEP, obtendo os seguintes dados de retorno: Nome da cidade, estado e rua e outros dados. A API de dados de CEP foi fornecida pelo site **Brasil API** (CEP).

Os seguintes containers foram utilizados:

1. Produtor
2. Consumidor
3. Servidor de Mensageria (RabbitMQ)
4. Banco de dados (MongoDB)

O **produtor** é responsável por receber a requisição de CEP do usuário. Em seguida, a informação é submetida ao **sistema de filas** e por fim, é enviado ao **consumidor** (caso esteja ativo), para processamento, onde é executado uma consulta a uma api externa para obter os dados da cidade, salvando as informações no **banco de dados**.

## 3. Execução

Clone o repositório do projeto disponível no Github no seguinte endereço:

[https://github.com/joaocarlos-losfe/topicos\\_em\\_sd\\_trabalho](https://github.com/joaocarlos-losfe/topicos_em_sd_trabalho)

Após ter clonado o repositório, execute os seguintes passos:

1. Rodar o docker compose com o comando: `$ docker-compose up;`
2. Abrir o navegador no endereço `127.0.0.1:5000/docs` para fazer as solicitações de CEP;
3. Instalar o **mongodb compass** para visualizar o banco de dados criado com a descrição `"city_data"`.

## 4. Resultados

### 1 - Iniciando os containers.

```
joaocarlos@joaocarlos-Inspiron-5458: /media/joaocarlos/Arquivos/Dev/SD/topicos_em_sd_trabalho
Arquivo Editar Ver Pesquisar Terminal Ajuda
rabbitmq 2023-06-26 00:12:37.961616+00:00 [info] <0.230.0> Running boot step networking defined by app rabbit
rabbitmq 2023-06-26 00:12:37.961704+00:00 [info] <0.230.0> Running boot step definition_import_worker_pool defined by app rabbit
rabbitmq 2023-06-26 00:12:37.961764+00:00 [info] <0.337.0> Starting worker pool 'definition_import_pool' with 4 processes in it
rabbitmq 2023-06-26 00:12:37.962689+00:00 [info] <0.230.0> Running boot step cluster_name defined by app rabbit
rabbitmq 2023-06-26 00:12:37.962820+00:00 [info] <0.230.0> Running boot step direct_client defined by app rabbit
rabbitmq 2023-06-26 00:12:37.963282+00:00 [info] <0.230.0> Running boot step rabbit_maintenance_mode_state defined by app rabbit
rabbitmq 2023-06-26 00:12:37.963384+00:00 [info] <0.230.0> Creating table rabbit_node_maintenance_states for maintenance mode status
rabbitmq 2023-06-26 00:12:37.963732+00:00 [info] <0.230.0> Running boot step rabbit_management_load_definitions defined by app rabbitmq_management
rabbitmq 2023-06-26 00:12:37.963914+00:00 [info] <0.472.0> Resetting node maintenance status
rabbitmq 2023-06-26 00:12:37.993141+00:00 [info] <0.531.0> Management plugin: HTTP (non-TLS) listener started on port 15672
rabbitmq 2023-06-26 00:12:37.993417+00:00 [info] <0.559.0> Statistics database started.
rabbitmq 2023-06-26 00:12:37.993576+00:00 [info] <0.558.0> Starting worker pool 'management_worker_pool' with 3 processes in it
rabbitmq 2023-06-26 00:12:38.015515+00:00 [info] <0.573.0> Prometheus metrics: HTTP (non-TLS) listener started on port 15692
rabbitmq 2023-06-26 00:12:38.015785+00:00 [info] <0.472.0> Ready to start client connection listeners
rabbitmq 2023-06-26 00:12:38.019191+00:00 [info] <0.617.0> started TCP listener on [::]:5672
rabbitmq completed with 4 plugins.
rabbitmq 2023-06-26 00:12:38.137502+00:00 [info] <0.472.0> Server startup complete; 4 plugins started.
rabbitmq 2023-06-26 00:12:38.137502+00:00 [info] <0.472.0> * rabbitmq_prometheus
rabbitmq 2023-06-26 00:12:38.137502+00:00 [info] <0.472.0> * rabbitmq_management
rabbitmq 2023-06-26 00:12:38.137502+00:00 [info] <0.472.0> * rabbitmq_web_dispatch
rabbitmq 2023-06-26 00:12:38.137502+00:00 [info] <0.472.0> * rabbitmq_management_agent
consumer-app_1 Requirement already satisfied: pika in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 1)) (1.3.2)
consumer-app_1 Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 2)) (2.31.0)
consumer-app_1 Requirement already satisfied: pymongo in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 3)) (4.4.0)
consumer-app_1 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->-r requirements.txt (line 2)) (3.4)
consumer-app_1 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->-r requirements.txt (line 2)) (3.1.0)
consumer-app_1 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->-r requirements.txt (line 2)) (2.0.3)
consumer-app_1 Requirement already satisfied: certifi<=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->-r requirements.txt (line 2)) (2023.5.7)
consumer-app_1 Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from pymongo->-r requirements.txt (line 3)) (2.3.0)
consumer-app_1 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended
consumer-app_1 to use a virtual environment instead: https://pip.pypa.io/warnings/venv
consumer-app_1 TERM environment variable not set.
rabbitmq 2023-06-26 00:12:39.054293+00:00 [info] <0.621.0> accepting AMQP connection <0.621.0> (172.18.0.1:40840 -> 172.18.0.2:5672)
rabbitmq 2023-06-26 00:12:39.057507+00:00 [info] <0.621.0> connection <0.621.0> (172.18.0.1:40840 -> 172.18.0.2:5672): user 'guest' authenticated and granted a
ccess to vhost '/'
domingo, junho 25, 21:12
```

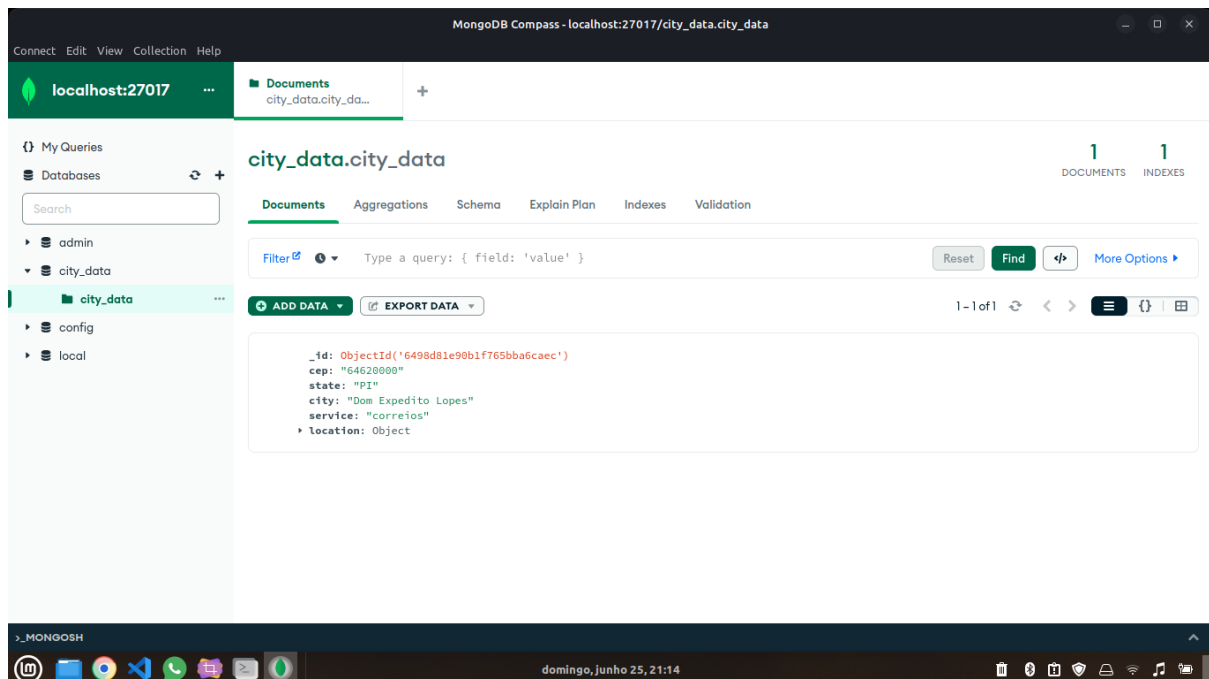
### 2 - Interface para envio do CEP.

```
(375) STMPD RCRDS MXT x FastAPI-Swagger UI +
127.0.0.1:5000/docs#/Producer/send_cep_post
Producer
POST /cep Send
Parameters
cep *required
string
(path) 64620-000
Execute Clear
Responses
Curl
curl -X 'POST' \
'http://127.0.0.1:5000/64620-000' \
-H 'accept: application/json' \
-d ''
Request URL
http://127.0.0.1:5000/64620-000
Server response
Code Details
200
Response body
{"enviado": "64620-000"}
Response headers
content-length: 28
content-type: application/json
date: Mon, 26 Jun 2023 00:11:17 GMT
domingo, junho 25, 21:13
```

### 3 - Iniciando os serviços.

```
joaoarclos@joaoarclos-Inspiron-5458: /media/joaoarclos/Arquivos/Dev/SD/topicos_em_sd_trabalho
Arquivo Editar Ver Pesquisar Terminal Ajuda
rabbitmq 2023-06-26 00:12:39.057507+00:00 [info] <0.621.0> connection <0.621.0> (172.18.0.1:40840 -> 172.18.0.2:5672): user 'guest' authenticated and granted a
ccess to vhost '/'
producer-app_1 INFO: 172.19.0.1:44608 - "GET /docs HTTP/1.1" 200 OK
producer-app_1 INFO: 172.19.0.1:44608 - "GET /favicon.ico HTTP/1.1" 405 Method Not Allowed
producer-app_1 INFO: 172.19.0.1:44608 - "GET /openapi.json HTTP/1.1" 200 OK
rabbitmq 2023-06-26 00:13:18.122174+00:00 [info] <0.641.0> accepting AMQP connection <0.641.0> (172.18.0.1:58456 -> 172.18.0.2:5672)
rabbitmq 2023-06-26 00:13:18.131771+00:00 [info] <0.641.0> connection <0.641.0> (172.18.0.1:58456 -> 172.18.0.2:5672): user 'guest' authenticated and granted a
ccess to vhost '/'
producer-app_1 [x] Sent body >> 64620-000
producer-app_1 INFO: 172.19.0.1:44616 - "POST /64620-000 HTTP/1.1" 200 OK
producer-app_1 INFO: 172.19.0.1:44616 - "POST /64620-000 HTTP/1.1" 200 OK
rabbitmq 2023-06-26 00:13:18.143019+00:00 [info] <0.641.0> closing AMQP connection <0.641.0> (172.18.0.1:58456 -> 172.18.0.2:5672, vhost: '/', user: 'guest')
producer-app_1 {"t":{"sdate":"2023-06-26T00:13:18.871+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"
172.19.0.1:60280","connectionId":1,"connectionCount":1}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.871+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"client metadata","attr":{"remote":"172.19.
0.1:60280","client":"conn1","doc":{"driver":{"name":"PyMongo","version":"4.4.0"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-75-generi
c"},"platform":"CPython 3.10.6.final.0"}}}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.874+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"
172.19.0.1:60290","connectionId":2,"connectionCount":2}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.874+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"
172.19.0.1:60302","connectionId":3,"connectionCount":3}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.874+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client metadata","attr":{"remote":"172.19.
0.1:60290","client":"conn2","doc":{"driver":{"name":"PyMongo","version":"4.4.0"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-75-generi
c"},"platform":"CPython 3.10.6.final.0"}}}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.875+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn3","msg":"client metadata","attr":{"remote":"172.19.
0.1:60302","client":"conn3","doc":{"driver":{"name":"PyMongo","version":"4.4.0"},"os":{"type":"Linux","name":"Linux","architecture":"x86_64","version":"5.15.0-75-generi
c"},"platform":"CPython 3.10.6.final.0"}}}}
mongodb {"t":{"sdate":"2023-06-26T00:13:18.880+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn2","msg":"Connection ended","attr":{"remote":"172.19.
0.1:60290","connectionId":2,"connectionCount":2}}
mongodb {"t":{"sdate":"2023-06-26T00:13:19.374+00:00"},"s":"I", "c":"-", "id":20883, "ctx":"conn1","msg":"Interrupted operation as its client discon
nected","attr":{"opId":668}}
mongodb {"t":{"sdate":"2023-06-26T00:13:19.374+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn3","msg":"Connection ended","attr":{"remote":"172.19.
0.1:60302","connectionId":3,"connectionCount":1}}
mongodb {"t":{"sdate":"2023-06-26T00:13:19.374+00:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn1","msg":"Connection ended","attr":{"remote":"172.19.
0.1:60280","connectionId":1,"connectionCount":0}}
mongodb {"t":{"sdate":"2023-06-26T00:13:26.435+00:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"WTCheckpointThread","msg":"WiredTiger message","attr":{"
message":"[1687738406:435171][1:0x7f1fb787700], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 4, snapshot max: 4 snapshot count:
0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0)}}
```

### 4 - Banco de dados.



## 5 - Containers ativos.

```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
87dc6e98ad94	topicos_em_sd_trabalho_consumer-app	"bash -c 'chmod +x r..."	21 minutes ago	Up 2 minutes	0.0.0.0:5001->5001/tcp, :::5001->5001/tcp
blcdf7de8aa6	topicos_em_sd_trabalho_producer-app	"bash -c 'chmod +x r..."	21 minutes ago	Up 2 minutes	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
e5ea525aa78d	rabbitmq:3-management-alpine	"docker-entrypoint.s..."	7 days ago	Up 2 minutes	4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, :::5672->5672/tcp
33776b50ed1f	mongo:4.4.6	"docker-entrypoint.s..."	7 days ago	Up 2 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp

```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$
```

## 6 - Parando o container do “consumidor”.

```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
87dc6e98ad94	topicos_em_sd_trabalho_consumer-app	"bash -c 'chmod +x r..."	21 minutes ago	Up 2 minutes	0.0.0.0:5001->5001/tcp, :::5001->5001/tcp
blcdf7de8aa6	topicos_em_sd_trabalho_producer-app	"bash -c 'chmod +x r..."	21 minutes ago	Up 2 minutes	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
e5ea525aa78d	rabbitmq:3-management-alpine	"docker-entrypoint.s..."	7 days ago	Up 2 minutes	4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, :::5672->5672/tcp
33776b50ed1f	mongo:4.4.6	"docker-entrypoint.s..."	7 days ago	Up 2 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp

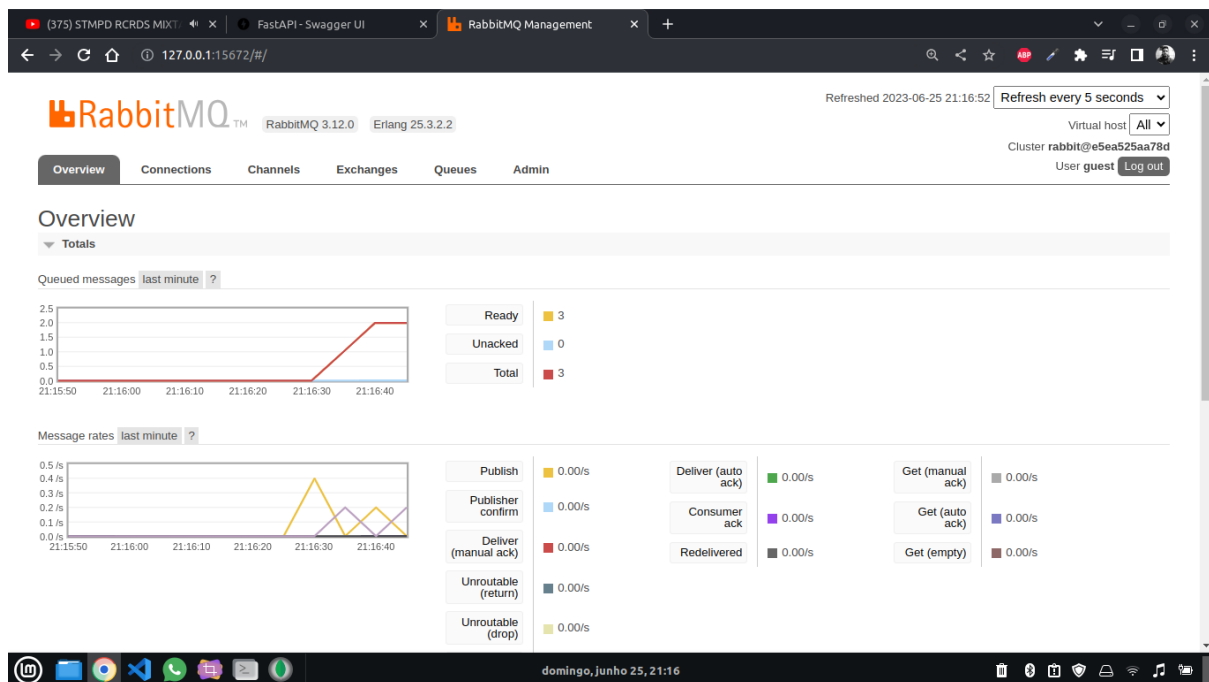
```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker stop 87d
```

```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
blcdf7de8aa6	topicos_em_sd_trabalho_producer-app	"bash -c 'chmod +x r..."	21 minutes ago	Up 3 minutes	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
e5ea525aa78d	rabbitmq:3-management-alpine	"docker-entrypoint.s..."	7 days ago	Up 3 minutes	4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, :::5672->5672/tcp
33776b50ed1f	mongo:4.4.6	"docker-entrypoint.s..."	7 days ago	Up 3 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp

```
joao@joao-Inspiron-5458: /media/joao/Arquivos/Dev/SD/topicos_em_sd_trabalho$
```

## 7 - Dashboard do RabbitMQ exibindo 3 mensagens na fila.



## 8 - Iniciando o container do consumidor novamente.

```
joaocarlos@joaocarlos-Inspiron-5458: /media/joaocarlos/Arquivos/Dev/SD/topicos_em_sd_trabalho
Arquivo Editar Ver Pesquisar Terminal Abas Ajuda
joaocarlos@joaocarlos-Inspiron-5458: /media/joaocarlos/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
topicos_em_sd_trabalho_consumer-app   latest             a85f56dc7d18       23 minutes ago     483MB
topicos_em_sd_trabalho_producer-app    latest             b860cc61fc6a       23 minutes ago     483MB
<none>                                <none>             b482f48dc72b       3 days ago         483MB
<none>                                <none>             e4e2d48dd370       7 days ago         483MB
<none>                                <none>             625261dea5ef       7 days ago         483MB
ubuntu               22.04              99284ca6cea0       2 weeks ago        77.8MB
rabbitmq              3-management-alpine f424289e38d5       3 weeks ago        179MB
hello-world           latest             9c7a54a9a43c       7 weeks ago        13.3kB
mongo                4.4.6              61ea24dc52c6       23 months ago      423MB
joaocarlos@joaocarlos-Inspiron-5458: /media/joaocarlos/Arquivos/Dev/SD/topicos_em_sd_trabalho$ docker run -it a85 bash -c "./run.sh"
```

## 9 - Container do consumidor iniciado recebendo mensagens da fila.

```
joaoCarlos@joaoCarlos-Inspiron-5458: /media/joaoCarlos/Arquivos/Dev/SD/topicos_em_sd_trabalho
Arquivo Editar Ver Pesquisar Terminal Abas Ajuda

joaoCarlos@joaoCarlos-Inspiron-5458: /media/joaoCarlos/Arquivos/Dev/SD/topicos_em_sd_... x joaoCarlos@joaoCarlos-Inspiron-5458: /media/joaoCarlos/Arquivos/Dev/SD/topicos_em_sd_... x +

[*] Aguardando mensagens. CTRL+C para sair
[x] Mensagem recebida ✓
Processando...
95020-360
inserted data on database
{'cep': '95020360', 'state': 'RS', 'city': 'Caxias do Sul', 'neighborhood': 'Nossa Senhora de Lourdes', 'street': 'Rua Ernesto Alves', 'service': 'correios', 'location': {'type': 'Point', 'coordinates': {'longitude': '-51.1774387', 'latitude': '-29.1637962'}}}
[*] Aguardando mensagens. CTRL+C para sair
[x] Mensagem recebida ✓
Processando...
64602-990
inserted data on database
{'cep': '64602990', 'state': 'PI', 'city': 'Picos', 'street': 'BR-316 Lado Impar - Altamira', 'service': 'correios', 'location': {'type': 'Point', 'coordinates': {}}}
[*] Aguardando mensagens. CTRL+C para sair
[x] Mensagem recebida ✓
Processando...
01153-000
inserted data on database
{'cep': '01153000', 'state': 'SP', 'city': 'São Paulo', 'neighborhood': 'Barra Funda', 'street': 'Rua Vitorino Carmilo', 'service': 'correios', 'location': {'type': 'Point', 'coordinates': {'longitude': '-46.6542596', 'latitude': '-23.5291132'}}}
[*] Aguardando mensagens. CTRL+C para sair
```

## 10 - Novos dados salvos.

MongoDB Compass - localhost:27017/city\_data.city\_data

Connect Edit View Collection Help

localhost:27017 Documents city\_data.city\_da...

My Queries Databases Search

city\_data

city\_data.city\_data 1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT DATA 1 - 4 of 4

	cep String	state String	city String	service String
1	95020360	RS	Caxias do Sul	correios
2	64602990	PI	Picos	correios
3	01153000	SP	São Paulo	correios

>\_MONGOSH

## 5. Erros encontrados

**Problema 1:** O produtor nunca saberá se o dado enviado para o consumidor será processado, devido a problemas de disponibilidade da API externa ou crashes no sistema.

Solução: uma possível solução para esse problema, é implementar um sistema de notificação para o produtor informando da disponibilidade do serviço do consumidor.

**Problema 2:** Usar somente filas não é uma boa abordagem, já que elas podem acumular muitas mensagens fazendo o sistema não ter escalabilidade.

Solução: Um balanceador de carga juntamente com as filas é o ideal na arquitetura do sistema proposto. Onde o balanceador de carga fica responsável por distribuir as solicitações de API, garantindo que a carga seja distribuída de maneira equilibrada e que cada servidor receba apenas uma quantidade razoável de solicitações.