

Site para diagramação: [Draw.io](https://draw.io)

**Observação:** Os comandos de terminal a seguir foram feitos para o terminal **Git Bash**.

---

## Descritivo

- ✓ Realizar deploy do Mysql, PHPMyadmin e aplicação PHP.
  - ✓ Diagrama Mysql, PHP
- ✓ Realizar o deploy da aplicação Java.
  - ✓ Diagrama Java
- ✓ Realizar deploy do Mongo DB.
  - ✓ Diagrama Mongo

## Regras

Individual, réplicas os dois serão considerados 0.

- Evidências de criação, passa a passo.
  - Diagrama da arquitetura, de cada aplicação.
  - Deverá conter data e hora nos prints.
- 

## Criando uma Rede

```
docker network create [NOME]
```

Exemplo:

```
docker network create saes-net
```

```
MINGW64/c/Users/Saes
Saes@DESKTOP-CMTLIMA MINGW64 ~
$ docker network create saes-net
de2fa167a6c560df8541c15e026a1b048989a8add20d845c9b790b0653b4db58

Saes@DESKTOP-CMTLIMA MINGW64 ~
$
```

# PHP - MySQL - PhpMyAdmin

## MySQL Server

Comando que vai iniciar nosso container em maneira ( `-d 'detached'`) desacoplada (para não travar o terminal) e definir as seguintes configurações:

- `--name saes-mysql` -> define o nome do container.
- `--network saes-net` -> define uma rede a qual o container vai pertencer.
- `-e` -> flag AMBIENTE Ambiente. nos ajuda a configurar variáveis de ambiente que ajudam a configurar nosso banco de dados.
- `-e MYSQL_ROOT_PASSWORD=123` -> Define a senha de acesso root ao nosso mysql-server.
- `-e MYSQL_ROOT_DATABASE=saes-db` -> Cria um banco de dados.
- `-e MYSQL_USER=viniuser` -> cria um usuário para o banco de dados saes-db.
- `-e MYSQL_PASSWORD=vini123` -> Define a senha do usuário viniuser.
- `mysql/mysql-server:latest` -> Informa que a imagem é o mysql-server do pacote mysql e :latest busca a última versão

## Comando para iniciar o MySQL Container

```
docker run -d \
  --name saes-mysql \
  --network saes-net \
  -e MYSQL_ROOT_PASSWORD=123 \
```

```
-e MYSQL_DATABASE=saes-db \  
-e MYSQL_USER=viniuser \  
-e MYSQL_PASSWORD=vini123 \  
mysql/mysql-server:latest
```

```
Saes@DESKTOP-CMTLIMA MINGW64 ~  
$ docker run -d \  
> --name saes-mysql \  
> --network saes-net \  
> -e MYSQL_ROOT_PASSWORD=123 \  
> -e MYSQL_DATABASE=saes-db \  
> -e MYSQL_USER=viniuser \  
> -e MYSQL_PASSWORD=vini123 \  
> mysql/mysql-server:latest  
Unable to find image 'mysql/mysql-server:latest' locally  
latest: Pulling from mysql/mysql-server  
c7e93886e496: Pulling fs layer  
6a4a3ef82cdc: Pulling fs layer  
abe8d2406c31: Pulling fs layer  
349b52643cc3: Pulling fs layer  
5518b09b1089: Pulling fs layer  
b6b576315b62: Pulling fs layer  
c7668948e14a: Pulling fs layer  
c7e93886e496: Download complete  
abe8d2406c31: Download complete  
349b52643cc3: Download complete  
c7668948e14a: Download complete  
5518b09b1089: Download complete  
6a4a3ef82cdc: Download complete  
6a4a3ef82cdc: Pull complete  
5518b09b1089: Pull complete  
b6b576315b62: Download complete  
abe8d2406c31: Pull complete  
349b52643cc3: Pull complete  
b6b576315b62: Pull complete  
c7e93886e496: Pull complete  
c7668948e14a: Pull complete  
Digest: sha256:d6c8301b7834c5b9c2b733b10b7e630f441af7bc917c74dba379f24eeeb6a313  
Status: Downloaded newer image for mysql/mysql-server:latest  
846a323b6511fa6c6d672350c0dc761bcb57e65a07c37808d616593569b1e8e  
  
Saes@DESKTOP-CMTLIMA MINGW64 ~  
$ |
```

^ [ ] [ ] POR 14:57  
PTB 08/05/2025 [ ]

## Verificar Logs

```
docker logs -f saes-mysql
```

A flag `-f` (follow) mantém os logs sendo exibidos em tempo real.

```
MINGW64/c/Users/Saes
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.

[Entrypoint] ignoring /docker-entrypoint-initdb.d/*

2025-05-08T17:42:57.615498Z 14 [System] [MY-013172] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 8.0.32).
2025-05-08T17:42:59.075877Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: shutdown complete (mysqld 8.0.32) MySQL Community Server - GPL.
[Entrypoint] Server shut down

[Entrypoint] MySQL init process done. Ready for start up.

[Entrypoint] Starting MySQL 8.0.32-1.2.11-server
2025-05-08T17:42:59.829846Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2025-05-08T17:42:59.831309Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.32) starting as process 1
2025-05-08T17:42:59.858438Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2025-05-08T17:43:00.114413Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2025-05-08T17:43:00.437758Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2025-05-08T17:43:00.437818Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2025-05-08T17:43:00.473215Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
2025-05-08T17:43:00.473357Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.32' socket: '/var/lib/mysql/mysql.sock' port: 3306 MySQL Community Server - GPL.

Saes@DESKTOP-CMTLIMA MINGW64 -
$
```

## phpMyAdmin

Comando para criar um container e exibir suas portas.

- `docker container run -d \` -> Cria e executa o container de maneira - d detached(desacoplada) afim de não travar o terminal.
- `--name saes-phpadmin \` -> Define o nome do container.
- `--network saes-net \` -> Define a qual rede o container irá pertencer.
- `-p 8080:80 \` -> mapeia a porta 80 do container para a porta 8080 do local host (localhost:container).
- `-e PMA_HOST=saes-mysql \` -> diz para o phpMyAdmin se conectar com um banco de dados chamado saes-mysql .
- `-e PMA_USER=viniuser \` -> Usa o usuário viniuser como usuário de acesso ao banco de dados.
- `-e PMA_PASSWORD=vini123 \` -> Usa a senha vini123 do usuário viniuser como senha de acesso ao banco de dados.
- `phpmyadmin` -> imagem do phpMyAdmin.

```
docker container run -d \
  --name saes-phpadmin \
  --network saes-net \
  -p 8080:80 \
  -e PMA_HOST=saes-mysql \
```

```
-e PMA_USER=viniuser \  
-e PMA_PASSWORD=vini123 \  
phpmyadmin
```



A terminal window titled 'MINGW64: c/Users/Saes/Documents/Fiap/2025/DevOps/cp2/docker' showing the execution of Docker commands. The user runs 'docker container run' to start a 'phpmyadmin' container with specific environment variables. Then, they run 'docker container ls' to list all containers. The output shows two containers: 'saes-phpadmin' (phpmyadmin image) and 'saes-mysql' (mysql/mysql-server:latest image). The 'saes-mysql' container is up for 45 minutes and is healthy.

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker (main)  
$ docker container run -d --name saes-phpadmin --network saes-net -e PMA_HOST=saes  
-db -p 8080:80 -e PMA_USER=viniuser -e PMA_PASSWORD=vini123 phpmyadmin  
50eecd5df4415fd929dc664243815b9f9d9a08011a60520d33d95dd236d78e55  
  
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker (main)  
$ docker container ls  
CONTAINER ID   IMAGE                                COMMAND                  CREATED  
STATUS        PORTS              NAMES  
50eecd5df441   phpmyadmin             "/docker-entrypoint...." 8 seconds ago  
Up 7 seconds   0.0.0.0:8080->80/tcp  saes-phpadmin  
846a323b6511   mysql/mysql-server:latest "/entrypoint.sh mysql..." 45 minutes ago  
Up 45 minutes (healthy) 3306/tcp, 33060-33061/tcp saes-mysql  
  
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker (main)  
$
```

## Servidor PHP

Primeiro, vamos criar um arquivo `index.php` que servirá como nossa home na web para identificar se obtivemos sucesso ao conectar com o banco de dados criado previamente. Iremos inserir o seguinte código dentro do arquivo `index.php` dentro de uma pasta agosto(á gosto do cliente).

Criar o arquivo `index.php` :

```
<?php  
$mysqli = new mysqli("saes-mysql", "viniuser", "vini123", "saes-db");  
  
if ($mysqli->connect_error) {  
    die("Erro na conexão: " . $mysqli->connect_error);  
}  
  
echo "Conectado com sucesso ao Banco de Dados saes-db!";  
?>
```

```
$ cat index.php
<?php
$mysqli = new mysqli("saes-sql", "viniuser", "vini123", "saes-db");
if ($mysqli->connect_error) {
    die("Erro na conexão: " . $mysqli->connect_error);
}
echo "Conectado com sucesso ao Banco de Dados saes-db!";
?>

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker/saes-backend
(main)
$ |
```

Criar a pasta agosto (nome à escolha do cliente), e dentro dela executar:

Dentro da nossa pasta agosto, executamos o nosso comando para instanciar nosso container php e definimos um volume que irá compartilhar uma pasta local(não se perde caso delete o container) entre nosso container e nosso host.

```
docker run -d \
  --name saes-php \
  --network saes-net \
  -p 80:80 \
  -v "$(pwd -W)":/var/www/html \
  php:8.2-apache \
  bash -c "apt update \
  && apt install -y libpng-dev libjpeg-dev libonig-dev libxml2-dev libzip-dev \
  unzip zip libmcrypt-dev \
  && docker-php-ext-install mysqli \
  && docker-php-ext-enable mysqli \
  && apachectl -D FOREGROUND"
```

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker/saes-backend
(main)
$ docker container run -d --name saes-php --network saes-net -p 80:80 -v "$PWD":/
ar/www/html php:8.2-apache bash -c "apt update \
  && apt install -y libpng-dev libjpeg-dev libonig-dev libxml2-dev libzip-dev unzi
zip
libmcrypt-dev \
  && docker-php-ext-install mysqli \
  && docker-php-ext-enable mysqli \
  && apachectl -D FOREGROUND"
Unable to find image 'php:8.2-apache' locally
8.2-apache: Pulling from library/php
Digest: sha256:e2408924aac97ed8dce0ba54adff30443fe7a940a87d7b0d083b36941d8aa431
Status: Downloaded newer image for php:8.2-apache
19e0f3876b0db049f2fce79e7a9fd1662fff586cedbdd160aefae7624bd76a7f

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2/docker/saes-backend
(main)
$
```

## Explicação

1. Serve os arquivos do diretório atual via Apache na porta 80.
2. Instala dependências e extensões PHP necessárias (como MySQLi).

## Logs

```
docker logs -f saes-php
```



```
Build complete.
Don't forget to run 'make test'.

+ strip --strip-all modules/mysql.so
Installing shared extensions:      /usr/local/lib/php/extensions/no-debug-non-zts-20220829/
Installing header files:           /usr/local/include/php/
find . -name \*.gcno -o -name \*.gcda | xargs rm -f
find . -name \*.lo -o -name \*.o -o -name \*.dep | xargs rm -f
find . -name \*.la -o -name \*.a | xargs rm -f
find . -name \*.so | xargs rm -f
find . -name .libs -a -type d|xargs rm -rf
rm -f libphp.la      modules/* libs/*
rm -f ext/opcache/jit/zend_jit_x86.c
rm -f ext/opcache/jit/zend_jit_arm64.c
rm -f ext/opcache/minilua

warning: mysqli (mysqli) is already loaded!

AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 172.18.0.4. Set the 'ServerName' directive globally to suppress this
message
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 172.18.0.4. Set the 'ServerName' directive globally to suppress this
message
[Thu May 08 19:12:07.290177 2025] [mpm_prefork:notice] [pid 3645:tid 3645] AH00163
: Apache/2.4.62 (Debian) PHP/8.2.28 configured -- resuming normal operations
[Thu May 08 19:12:07.290234 2025] [core:notice] [pid 3645:tid 3645] AH00094: Comma
nd line: '/usr/sbin/apache2 -D FOREGROUND'
```

## DIAGRAMA:

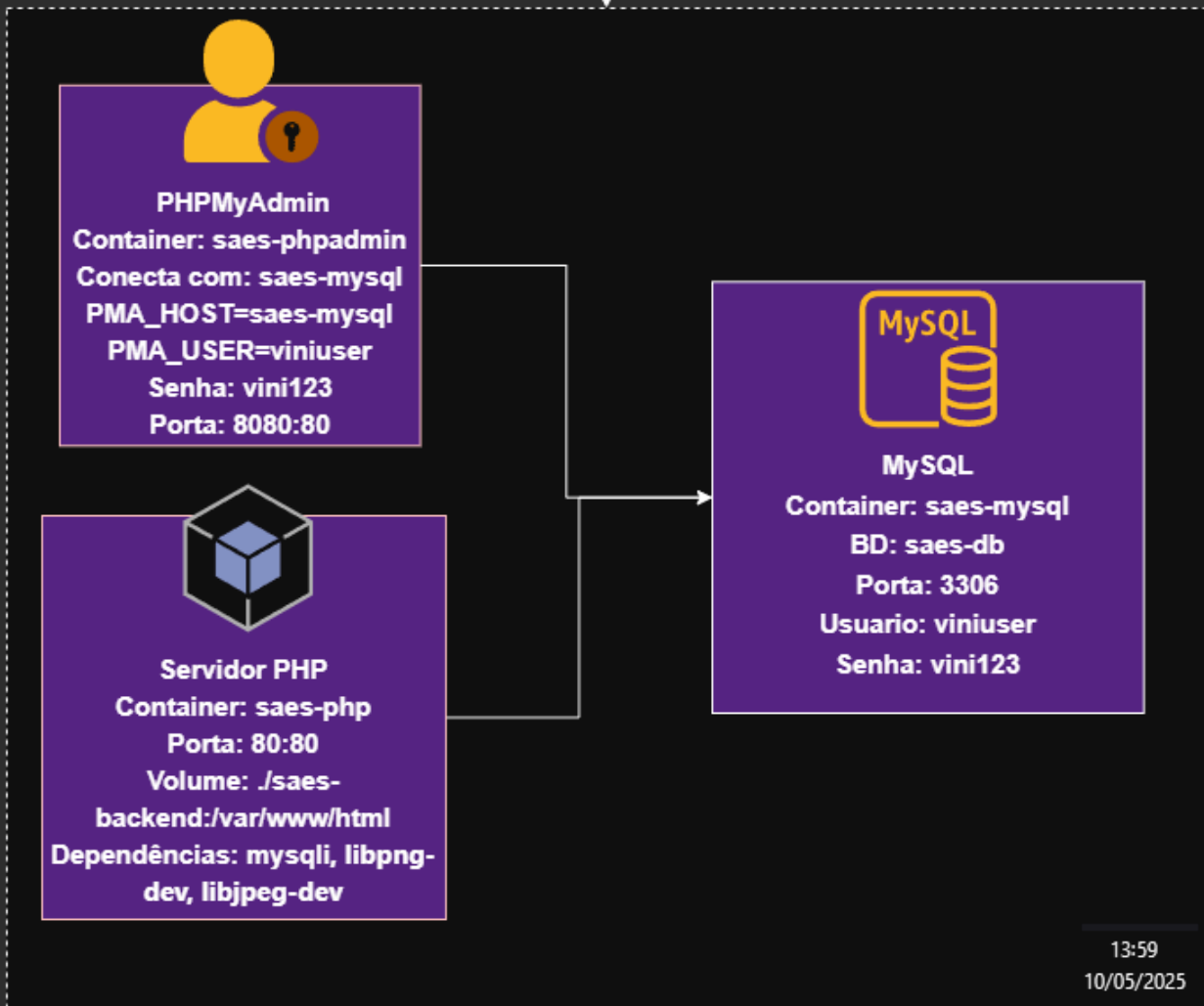




## CP2 - PHP / MySQL

Vinícius Saes de Souza - RM554456

saes-net



## Java:

### Aplicação Java com Tomcat

Iremos usar a imagem do Tomcat que irá servir nossa página web de exemplo vinda do arquivo petclinic.war <sup>[1]</sup>, essa imagem será executada dentro de um container com as seguintes características:

- **Imagem:** tomcat:10

- **Nome:** tomcat10
- **Expor a Porta:** 8080
- **Rodar em segundo plano (desacoplado)**

## Criar container:

```
docker container run -d \
  --name tomcat10 \
  -p 80:8080 \
  tomcat:10
```

- `docker container run -d` -> roda o container de maneira desacoplada (não trava o terminal)
- `--name tomcat` -> Define o nome do container
- `-p 80:8080` -> Mapeia a porta 80 do localhost para a porta 8080 do container
- `tomcat:10` -> Define qual imagem será utilizada no container

```
MINGW64/c/Users/Saes/Documents/Fiap/2025/DevOps/cp2
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker container run -d --name tomcat10 -p 80:8080 tomcat:10
59abc5277e35a6f6bc62172f5177c1fcf62535c8c16c1a51bb21f595f05c72f1

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
59abc5277e35   tomcat:10 "catalina.sh run"        4 seconds ago Up 3 seconds   0.0.0.0:80->8080/tcp              tomcat10

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$
```

## Copiar WAR para o container:

Comando para copiar a aplicação `petclinic.war` para o container `tomcat10` criado anteriormente:

```
docker cp petclinic.war tomcat10:/usr/local/tomcat/webapps
```

Formato: `docker cp [arquivo] [container]:[local de destino]`

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker cp petclinic.war tomcat10:/usr/local/tomcat/webapps

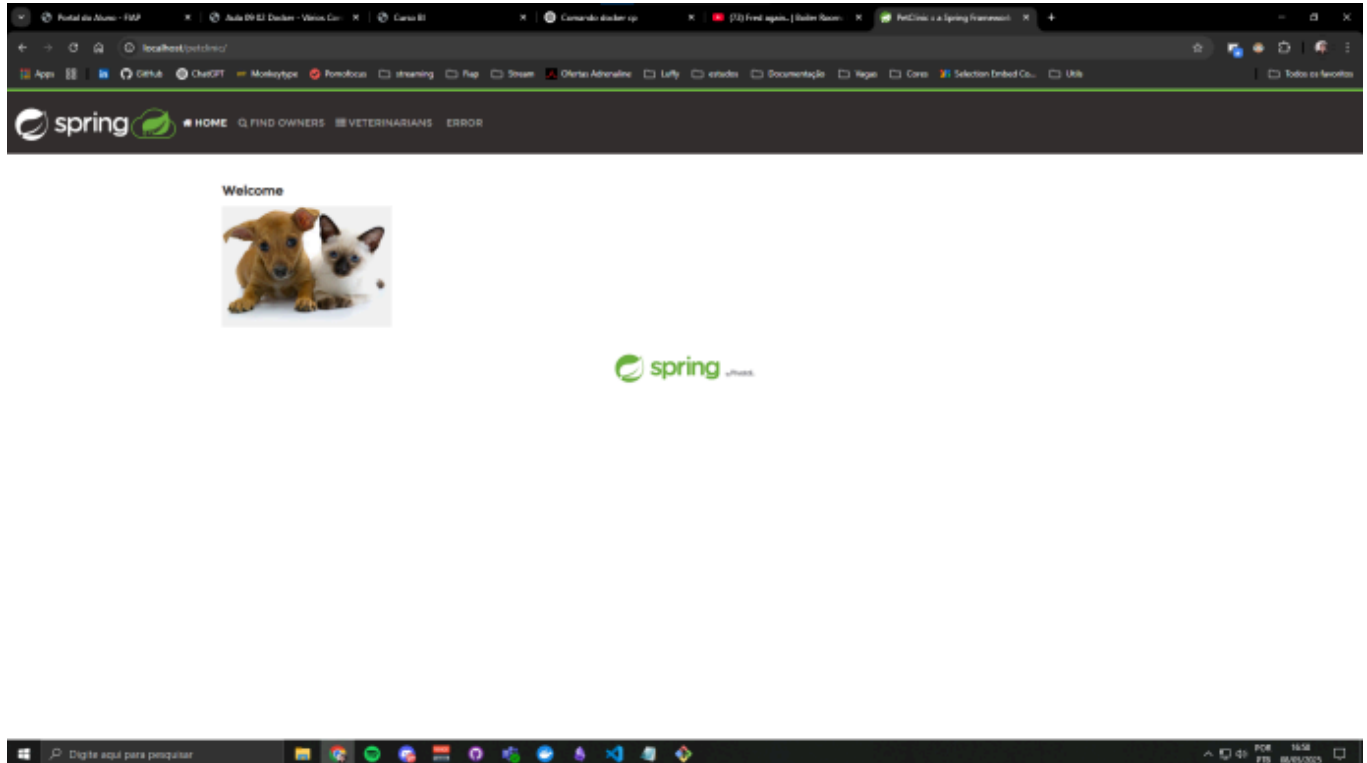
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ |
```

# Teste

Para testar a conexão com a porta mapeada 8080, vamos utilizar o navegador da nossa máquina entrando no endereço : `http://localhost:80/petclinic`

Acessar via navegador:

```
http://localhost:80/petclinic
```



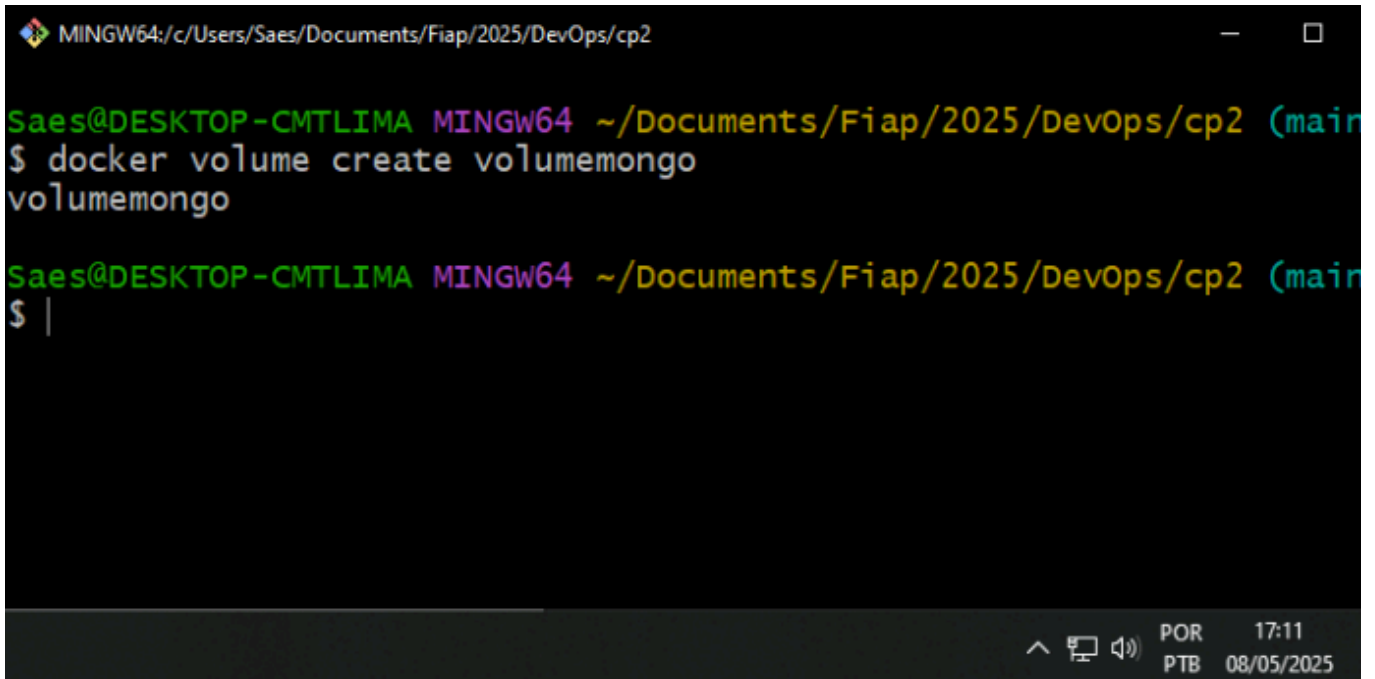
**DIAGRAMA:**



## MongoDB

## Criar um volume para persistir nossas coleções do Banco de dados:

```
docker volume create volumemongo
```

A screenshot of a Windows terminal window. The title bar shows 'MINGW64: c:/Users/Saes/Documents/Fiap/2025/DevOps/cp2'. The prompt is 'Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)'. The user enters the command 'docker volume create volumemongo'. The prompt changes to 'Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)' and the user enters a second prompt '\$ |'. The system tray at the bottom right shows 'POR 17:11', 'PTB 08/05/2025', and icons for network, volume, and sound.

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker volume create volumemongo
volumemongo

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ |
```

## Criar o container:

- **Nome:** mongo.
- **Network:** saes-net.
- **Host:** mongo
- **Porta:** 27017:27017
- **Volume:** volumemongo:/data/db
- **Imagem:** mongo

**Comando:**

```
docker container run \
--name mongo \
--network saes-net \
-h mongo \
-p 27017:27017 \
-v volumemongo:/data/db \
mongo
```

-h / --hostname : define o **nome do host (hostname)** dentro do contêiner. Outros contêineres na mesma rede podem conseguir se comunicar com ele usando esse hostname, dependendo da configuração da rede.

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker container run -d --name mongo --network saes-net -h mongo -p
 27017:27017 -v volumemongo:/data/db \
> mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
82a30a244416: Pulling fs layer
c1aa6a7c13c0: Pulling fs layer
f6ae99607c30: Pulling fs layer
54db8e3d8732: Pulling fs layer
d2c82d0518f8: Pulling fs layer
ab6be4e21c66: Pulling fs layer
ca015e545994: Pulling fs layer
82a30a244416: Download complete
ab6be4e21c66: Download complete
f6ae99607c30: Download complete
54db8e3d8732: Download complete
c1aa6a7c13c0: Download complete
c1aa6a7c13c0: Pull complete
d2c82d0518f8: Download complete
82a30a244416: Pull complete
d2c82d0518f8: Pull complete
ab6be4e21c66: Pull complete
f6ae99607c30: Pull complete
ca015e545994: Download complete
ca015e545994: Pull complete
54db8e3d8732: Pull complete
Digest: sha256:2e018e386e891d2e4239aca6035fb7701dac51b72891247ecd2f95
ff8a167859
Status: Downloaded newer image for mongo:latest
4348844a80d944e209b30a40eeb842674e552fadb9e9ed64652c8714ddeb5ec4

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$
```

## Verificando as informações do Docker para o Mongo:

```
docker ps
docker image ls
docker volume ls
docker network ls
```

```
Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
4348844a80d9   mongo    "docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  0.0.0.0:27017->27017/t

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mongo         latest    2e018e386e89   6 days ago    1.19GB
tomcat        10        76edc30a01a4   4 weeks ago    711MB
php           8.2-apache e2408924aac9   7 weeks ago    709MB
phpmyadmin    latest    68d7f9dc247b   3 months ago   813MB
mysql/mysql-server latest    d6c8301b7834   2 years ago    698MB

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker volume ls
DRIVER    VOLUME NAME
local     57889bd7e7091efd161b76a321144ebcc830f874f542c168e3861277ec4a8f0a
local     volumemongo

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ docker network ls
NETWORK ID    NAME        DRIVER    SCOPE
df2152b56410  bridge     bridge    local
db579cb4034a  host       host      local
8d008d74a8e1  none       null      local
de2fa167a6c5  saes-net   bridge    local

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$
```

## Acessando o terminal do Container:

```
docker container exec -it mongo bash
```

## Acessando o shell do MongoDB (mongosh):

```
mongosh
```



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Saes@DESKTOP-CMTLIMA MINGW64 ~/Documents/Fiap/2025/DevOps/cp2 (main)
$ winpty docker container exec -it mongo bash
root@mongo:/# mongosh
Current Mongosh Log ID: 681d1717aef02b858cd861df
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelection
TimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB:      8.0.9
Using Mongosh:      2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2025-05-08T20:23:43.476+00:00: Using the XFS filesystem is strongly recommended with
the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2025-05-08T20:23:44.563+00:00: Access control is not enabled for the database. Read a
nd write access to data and configuration is unrestricted
  2025-05-08T20:23:44.564+00:00: For customers running the current memory allocator, we
suggest changing the contents of the following sysfsFile
  2025-05-08T20:23:44.564+00:00: We suggest setting the contents of sysfsFile to 0.
  2025-05-08T20:23:44.564+00:00: vm.max_map_count is too low
  2025-05-08T20:23:44.564+00:00: We suggest setting swappiness to 0 or 1, as swapping c
an cause performance problems.
-----
test> |
```

## Inserindo itens no BD:

```
db.eletronicos.insertMany(
  [
    {
      produto: "Smartphone",
      tipo_produto: "Celular",
      quantidade_vendida: 100,
      valor_venda: 599.99
    },
    {
      produto: "TV LED",
      tipo_produto: "Televisor",
      quantidade_vendida: 50,
      valor_venda: 799.99
    },
    ...
    ...
  ]
);
```

```
...   "valor_venda": 499.99
... },
... {
...   "produto": "Caixa de Som",
...   "tipo_produto": "Áudio",
...   "quantidade_vendida": 60,
...   "valor_venda": 129.99
... },
... {
...   "produto": "Monitor",
...   "tipo_produto": "Computador",
...   insertedIds: {
...     '0': ObjectId('681f576a54bd7c81d5d861e0'),
...     '1': ObjectId('681f576a54bd7c81d5d861e1'),
...     '2': ObjectId('681f576a54bd7c81d5d861e2'),
...     '3': ObjectId('681f576a54bd7c81d5d861e3'),
...     '4': ObjectId('681f576a54bd7c81d5d861e4'),
...     '5': ObjectId('681f576a54bd7c81d5d861e5'),
...     '6': ObjectId('681f576a54bd7c81d5d861e6'),
...     '7': ObjectId('681f576a54bd7c81d5d861e7'),
...     '8': ObjectId('681f576a54bd7c81d5d861e8'),
...     '9': ObjectId('681f576a54bd7c81d5d861e9')
...   }
... }
produtos>
```

RAM 1.61 GB CPU 0.00% Disk: 2.44 GB used (limit 1006.85 GB) v4.40.0

Git Bash X

Git Bash X

10:41 10/05/2025

```
produtos> show dbs;
admin      40.00 KiB
config     60.00 KiB
local      40.00 KiB
produtos   40.00 KiB
produtos>
```

```
produtos> show collections
eletronicos
produtos>
```

DIAGRAMA:

## CP2 - MongoDB

Vinicius Saes de Souza

- RM554456



saes-net



**MongoDB 8.0**

**Container: mongo**

**DB: produtos**

**Volume: mongo:/data/db**

**Porta: 27017:27017 (host:container)**

**Hostname: mongo**

15:03

10/05/2025

1. A extensão .war (Web Application Archive) é um arquivo empacotado usado para distribuir e implantar aplicações web Java. Ele segue o padrão definido pela Jakarta EE (antiga Java EE) e contém todos os componentes necessários para executar uma aplicação web em um servidor compatível, como o Apache Tomcat, Jetty ou servidores de aplicação como WildFly e Payara. ↩