

# Ingeniera de Software 3

## TP 3: Sistemas distribuidos

### 4.1: Sistema distribuido simple

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI>docker network create -d bridge mybridge
eef07802d5ee29ec8858fd63a73ef00c2dc08b5176bef44debbba7d6a6c9971f9
```

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI> docker run -d --net mybridge --name db redis:alpine
Unable to find image 'redis:alpine' locally
alpine: Pulling from library/redis
7264a8db6415: Pull complete
a28817da73be: Pull complete
536ccaebaffb: Pull complete
f54d1871dea6: Pull complete
4d190b4b6472: Pull complete
33fcc95c965f: Pull complete
Digest: sha256:fd5de2340bc46cbc2241975ab027797c350dec6fd86349e3ac384e3a41be6fee
Status: Downloaded newer image for redis:alpine
c9c74aeaec5af1eb5296a14c338234fd69ea220274192526655b4785327c6c2f
```

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI> docker run -d --net mybridge -e REDIS_HOST=db -e REDIS_PORT=6379 -p
5000:5000 --name web alexisfr/flask-app:latest
Unable to find image 'alexisfr/flask-app:latest' locally
latest: Pulling from alexisfr/flask-app
f49cf87b52c1: Pull complete
7b491c575b06: Pull complete
b313b08bab3b: Pull complete
51d6678c3f0e: Pull complete
09f35bd58db2: Pull complete
1bda3d37eead: Pull complete
9f47966d4de2: Pull complete
9fd775bf531: Pull complete
2446eec18066: Pull complete
b98b851b2dad: Pull complete
e119cb75d84f: Pull complete
Digest: sha256:250221bea53e4e8f99a7ce79023c978ba0df69bdf620401756da46e34b7c80b
Status: Downloaded newer image for alexisfr/flask-app:latest
20846243cc626714e266db08a4d4aa131abed5b13617501da99491efb8e1d88c
```

Mostramos los puertos utilizados.

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI>docker container ls --format "table {{.ID}}\t{{.Names}}\t{{.Ports}}"
-a
CONTAINER ID   NAMES      PORTS
20846243cc62   web        0.0.0.0:5000->5000/tcp
c9c74aeaec5a   db         6379/tcp
bfa698a7f769   heuristic_mestorf
1b6f2927697c   dreamy_borg
```

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI>docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
20846243cc62   alexisfr/flask-app:latest   "python /app.py"        18 seconds ago   Up 16 seconds   0.0.0.0:5000->5000/tcp
c9c74aeaec5a   redis:alpine            "docker-entrypoint.s..."   5 minutes ago   Up 5 minutes   6379/tcp
```

Mostramos los detalles de la red

```
C:\Users\Nico\Documents\Ing de Soft 3\SimpleWebAPI>docker network inspect mybridge
```

```
[
  {
    "Name": "mybridge",
    "Id": "eef07802d5ee29ec8858fd63a73ef00c2dc08b5176bef44debba7d6a6c9971f9",
    "Created": "2023-09-05T19:09:04.524238046Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "20846243cc626714e266db08a4d4aa131abed5b13617501da99491efb8e1d88c": {
        "Name": "web",
        "EndpointID": "4da9737364a0b92d46fc1f092ce2b829bdf3876496d602e7149db0203c2e9965",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      },
      "c9c74aeaec5af1eb5296a14c338234fd69ea220274192526655b4785327c6c2f": {
        "Name": "db",
        "EndpointID": "db0002cd818a56c3f000051939ab4f49162febd6d8179670ef9fca160a5ac503",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

← → ↻ ⓘ localhost:5000

Hello from Redis! I have been seen 1 times.

```
import os

from flask import Flask
from redis import Redis

app = Flask(__name__)
redis = Redis(host=os.environ['REDIS_HOST'],
port=os.environ['REDIS_PORT'])
bind_port = int(os.environ['BIND_PORT'])
```

```
@app.route('/')
def hello():
    redis.incr('hits')
    total_hits = redis.get('hits').decode()
    return f'Hello from Redis! I have been seen {total_hits} times.'

if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True, port=bind_port)
```

Explicacion:

Primero hacemos una importacion de los modulos os, Flask y Redis.

Despues creamos instancias para Flask y Redis.

Definimos una funcion 'hello()' que incrementa un contador para mostrar cuanta es la cantidad de veces que fue vista.

Por ultimo un condicional para garantizar que Flask se inicio solo si el script se ejecuta directamente. Ejecutando la aplicación en el puerto '0.0.0.0' con depuracion activada.

El parametro -e se utiliza para proporcionar información de configuración dinámica a las aplicaciones que se ejecutan en contenedores.

Cuando ejecuto: `docker rm -f web` y Corro `docker run -d --net mybridge -e REDIS_HOST=db -e REDIS_PORT=6379 -p 5000:5000 --name web alexisfr/flask-app:latest`

redis.exceptions.ConnectionError

redis.exceptions.ConnectionError: Error -2 connecting to db:6379. Name or service not known.

Traceback (most recent call last):

```
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 484, in connect
    sock = self._connect()
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 517, in _connect
    socket.SOCK_STREAM):
File "/usr/local/lib/python3.6/socket.py", line 745, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
```

During handling of the above exception, another exception occurred:

```
File "/usr/local/lib/python3.6/site-packages/redis/client.py", line 667, in execute_command
    connection.send_command(*args)
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 670, in send_command
    self.send_packed_command(self.pack_command(*args))
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 585, in send_packed_command
    self.connect()
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 489, in connect
    raise ConnectionError(self._error_message(e))
```

During handling of the above exception, another exception occurred:

```
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 484, in connect
    sock = self._connect()
File "/usr/local/lib/python3.6/site-packages/redis/connection.py", line 517, in _connect
    socket.SOCK_STREAM):
File "/usr/local/lib/python3.6/socket.py", line 745, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
```

During handling of the above exception, another exception occurred:

```
File "/usr/local/lib/python3.6/site-packages/flask/app.py", line 2309, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python3.6/site-packages/flask/app.py", line 2295, in wsgi_app
    response = self.handle_exception(e)
```

```
PS C:\Users\Nico\Documents\IngSoft3> docker rm -f db
db
```

## redis.exceptions.ConnectionError

redis.exceptions.ConnectionError: Error -2 connecting to db:6379. Name or service not known.

Al levantarlo nuevamente con `docker run -d --net mybridge --name db redis:alpine` funciona.



← → ↻ ⓘ localhost:5000

Hello from Redis! I have been seen 1 times.

```
PS C:\Users\Nico\Documents\IngSoft3> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d58ced5802e4	redis:alpine	"docker-entrypoint.s..."	5 minutes ago	Up 5 minutes	6379/tcp
de2cac113276	alexisfr/flask-app:latest	"python /app.py"	9 minutes ago	Up 9 minutes	0.0.0.0:5000->5000/tcp

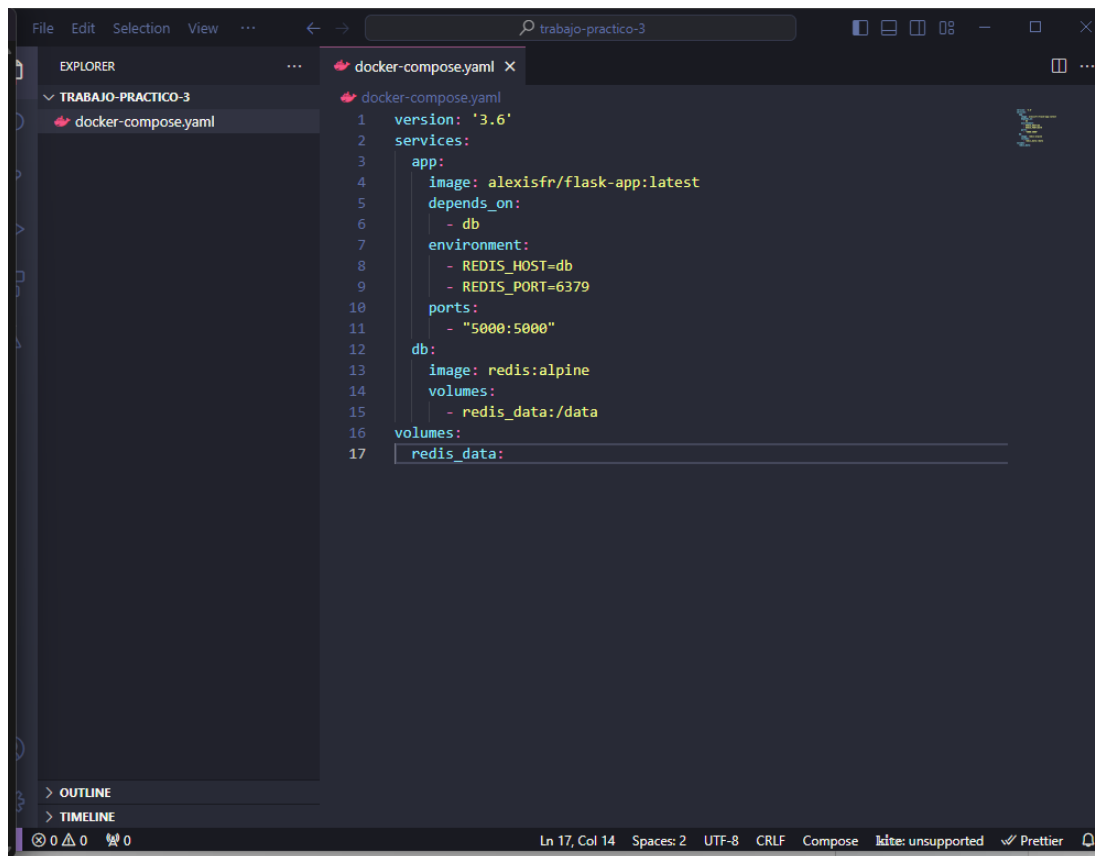
Para no perder la cuenta de las visitas, docker crea volúmenes que se utilizan para almacenar datos que necesitan sobrevivir al ciclo de vida de un contenedor y estos se almacenan en una ubicación fuera del contenedor.

Y por último eliminamos los elementos creados.

```
p web
PS C:\Users\Nico\Documents\IngSoft3> docker rm -f db
db
PS C:\Users\Nico\Documents\IngSoft3> docker rm -f web
web
PS C:\Users\Nico\Documents\IngSoft3> docker network rm mybridge
mybridge
```

### 4.3: Utilizando docker compose

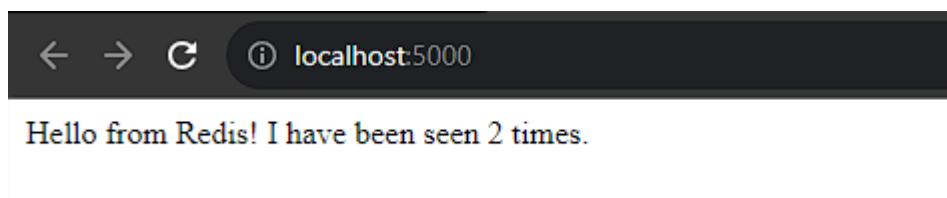
Creamos un archivo "docker-compose.yaml" y pegamos el código



```
Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ docker-compose up -d
Network trabajo-practico-3_default Creating
Network trabajo-practico-3_default Created
Volume "trabajo-practico-3_redis_data" Creating
Volume "trabajo-practico-3_redis_data" Created
Container trabajo-practico-3-db-1 Creating
Container trabajo-practico-3-db-1 Created
Container trabajo-practico-3-app-1 Creating
Container trabajo-practico-3-app-1 Created
Container trabajo-practico-3-db-1 Starting
Container trabajo-practico-3-db-1 Started
Container trabajo-practico-3-app-1 Starting
Container trabajo-practico-3-app-1 Started

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$
```

Accediendo al localhost

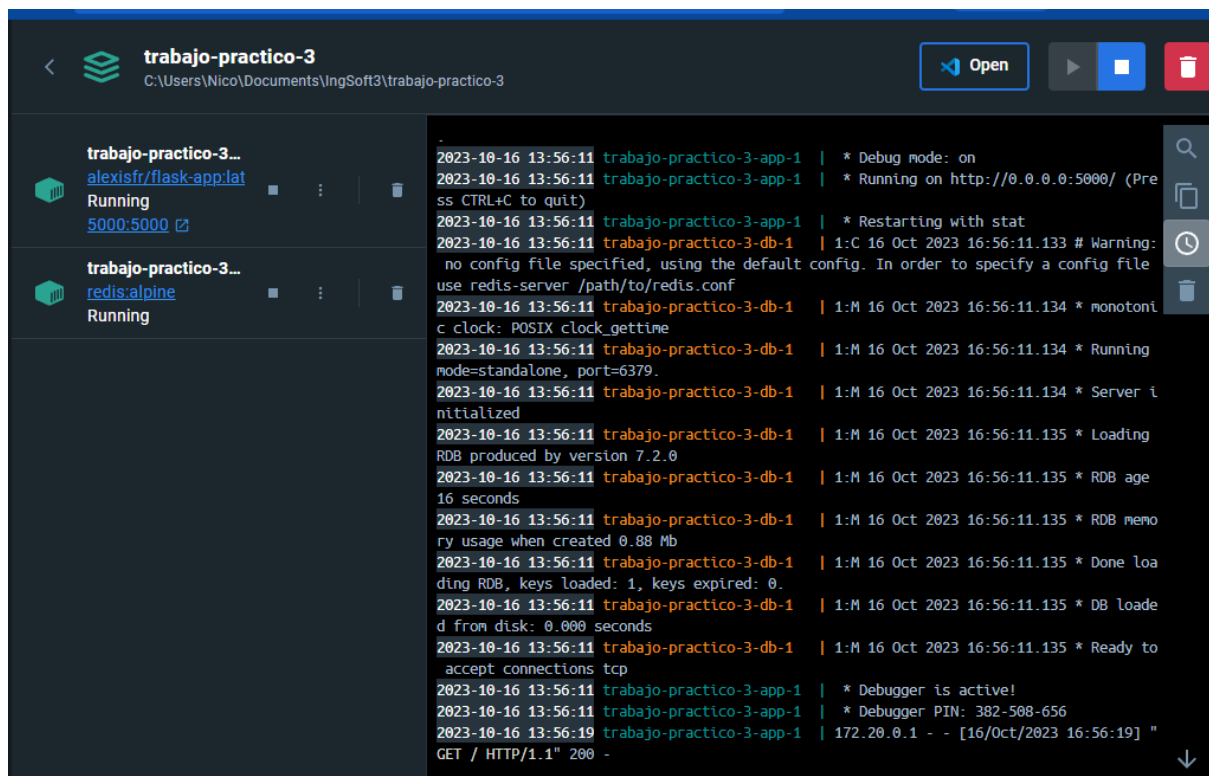


```
Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED
STATUS        PORTS          NAMES
4e7cc5c3013e   alexisfr/flask-app:latest           "python /app.py"        About a minu
te ago        Up 42 seconds  0.0.0.0:5000->5000/tcp   trabajo-practico-3-app-1
f153906aa6e0   redis:alpine                        "docker-entrypoint.s..." About a minu
te ago        Up 42 seconds  6379/tcp                trabajo-practico-3-db-1

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ docker network ls
NETWORK ID     NAME      DRIVER  SCOPE
e68dc903122d   bridge    bridge  local
fb33c1091221   host      host    local
0f2c1132b8c2   mybridge1 bridge    local
68917603368d   none      null    local
b77640521f68   trabajo-practico-3_default bridge    local

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ docker volume ls
DRIVER  VOLUME NAME
local   e24c06ff5e100d29fc1282e92d1ada0a87e4d872f0a4c280db8e21b603257e0
local   ff34bee96b0b6116d8d25ee9367897e33559e169eb0b6946087c18cc0979ee1e
local   trabajo-practico-3_redis_data
```

Docker compose nos permitio describir la configuracion de los servicios, redes y volúmenes, en este caso de Redis y Flask, en un solo archivo.  
Los agrupa.



4.4: Clonamos y hacemos un compose.

```

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ git clone https://github.com/dockerexamples/example-voting-app
Cloning into 'example-voting-app'...
remote: Enumerating objects: 1099, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 1099 (delta 0), reused 1 (delta 0), pack-reused 1091
Receiving objects: 100% (1099/1099), 1.16 MiB | 3.68 MiB/s, done.
Resolving deltas: 100% (411/411), done.

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3 (main)
$ cd example-voting-app
docker-compose -f docker-compose.yml up -d
db Pulling
96526aa774ef Pulling fs layer
c34fb5e28db2 Pulling fs layer
70e8a68669fa Pulling fs layer
08ae27ed4de0 Pulling fs layer
c0bee3658992 Pulling fs layer
efd5b655acb8 Pulling fs layer
15bff2d1d13a Pulling fs layer
615923b9e50e Pulling fs layer
08ae27ed4de0 Waiting
c0bee3658992 Waiting
efd5b655acb8 Waiting
15bff2d1d13a Waiting
615923b9e50e Waiting
c34fb5e28db2 Downloading [=====]
720B/1.264kB

```

Me da error y nose como solucionarlo.

```

Nico@DESKTOP-MDI8QUJ MINGW64 ~/Documents/IngSoft3/trabajo-practico-3/example-voting-app (main)
$ docker-compose -f docker-compose.yml up -d
Container example-voting-app-redis-1 Running
Container example-voting-app-db-1 Running
Container example-voting-app-redis-1 Waiting
Container example-voting-app-redis-1 Waiting
Container example-voting-app-db-1 Waiting
Container example-voting-app-db-1 Waiting
Container example-voting-app-redis-1 Error
Container example-voting-app-db-1 Error
Container example-voting-app-db-1 Error
Container example-voting-app-redis-1 Error
dependency failed to start: container example-voting-app-redis-1 is unhealthy

```