

Sistema de Computação na Cloud

**Mestrado em Engenharia Informática**

## Trabalho Prático - Parte 1

---

Grupo 2

Paulo José Capa Azevedo Meneses - 17611

André Filipe Silva Carvalho - 14062

Alexandre da Cunha Cepa - 10207

# Índice

[Índice](#)

[Introdução](#)

[Desenvolvimento](#)

[Conclusão](#)

[Bibliografia](#)

# Introdução

No âmbito da unidade curricular de Sistemas de Computação na *Cloud*, foi solicitado pelo docente a realização de um trabalho prático para aplicar os conteúdos leccionados na disciplina, cujo tema seria a criação de um container que contenha uma base de dados e uma aplicação web.

O container permite criar um package de uma determinada aplicação de forma padronizada, incluindo todo o código, configurações e dependências num único objecto.

Para gerir os containers foi utilizado o Docker que é o maior e mais popular sistema para criar e gerir containers.

Trata-se de uma aplicação aberta para programadores e administradores de sistema que permite o build, ship e execução de aplicações distribuídas.

Desta forma, o objetivo deste trabalho é estudar o deployment de uma aplicação utilizando containers.

Esperamos ao longo deste trabalho cumprir com todos os objetivos propostos, ultrapassando as dificuldades encontradas e, mais importante, esperamos aprender o deployment aplicacional utilizando containers, mais especificamente através do sistema docker.

# Desenvolvimento

## Conteúdo e configurações do *container*

O *container* desenvolvido tem uma base de dados PostgreSQL na versão 14.0-alpine e uma API em nodejs. A base de dados é acessível na porta 5432 e a API na porta 8080.

O **dockerfile** executa a instalação das dependências do nodejs 16 e uma cópia da aplicação para o diretório raiz do container.

```
1 FROM node:16
2 EXPOSE 8080
3
4 WORKDIR /home/app
5
6 COPY package.json /home/app/
7 COPY package-lock.json /home/app/
8
9 RUN npm ci
10
11 COPY . /home/app
12
13 #CMD [ "npm", "start" ]
```

Figura 1 - dockerfile

O **docker-compose** contém as configurações da base de dados e da aplicação.

Na parte das configurações que diz respeito à base de dados, é usada a imagem postgres:14.0-alpine e configurado o ambiente da base de dados, no qual é iniciado o nome de utilizador, palavra-passe e nome da base de dados.

A aplicação é configurada de modo a que só seja executada depois da base de dados, é dito qual as portas de acesso e o comando para a execução da aplicação, que é `command: ["npm", "start"]`.

```
1  # docker-compose.yml
2  version: "3"
3  services:
4    db:
5      image: postgres:14.0-alpine
6      # volumes:
7      #   - ./data/db:/var/lib/postgresql/data
8      environment:
9        - POSTGRES_DB=postgres
10       - POSTGRES_USER=postgres
11       - POSTGRES_PASSWORD=postgres
12      ports:
13        - "5432:5432"
14    app:
15      build: .
16      depends_on:
17        - db
18      ports:
19        - "8080:8080"
20      command: ["npm", "start"]
```

Figura 2 - docker-compose

## Conclusão

Uma vez concluída a realização desta primeira parte do trabalho prático o balanço que se pode fazer é bastante positivo.

Esperamos naturalmente ter cumprido todos os objetivos e requisitos propostos pelo docente.

## Bibliografia

<https://docs.docker.com/engine/reference/builder/>;

[https://hub.docker.com/\\_/postgres](https://hub.docker.com/_/postgres);

Toda a documentação fornecida pelo docente;

