

# DUPLA: Pedro Julius e Clara Livia.

## ATIVIDADE 07:

### Passo 01: Clonando repositório da aula;

```
Pedro Julius@GOLD-GAMER MINGW64 /d/ARQUIVOS/DOCUMENTOS/atividade-07
$ git clone https://github.com/programadorabordo/docker-introducao.git
Cloning into 'docker-introducao'...
remote: Enumerating objects: 45, done.
remote: Total 45 (delta 0), reused 0 (delta 0), pack-reused 45
Receiving objects: 100% (45/45), 456.52 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (14/14), done.
```

### Passo 02: Instalando NPM;

```
Pedro Julius@GOLD-GAMER MINGW64 /d/ARQUIVOS/DOCUMENTOS/atividade-07/docker-introducao (master)
$ cd api

Pedro Julius@GOLD-GAMER MINGW64 /d/ARQUIVOS/DOCUMENTOS/atividade-07/docker-introducao/api (master)
$ npm install
```

### Passo 03: Criando arquivo docker-compose.yml;

```
1 version: "3.7"
2 services:
3   db:
4     image: mysql
5     container_name: mysql-container
6     command: --default-authentication-plugin=mysql_native_password
7     environment:
8       MYSQL_ROOT_PASSWORD: programadorabordo
9     volumes:
10      - ./api/db/data:/var/lib/mysql
11     restart: always
12   api:
13     build: "./api"
14     container_name: node-container
15     restart: always
16     volumes:
17      - ./api:/home/node/app
18     ports:
19      - "9001:9001"
20     depends_on:
21      - db
22   web:
23     image: "php:7.2-apache"
24     container_name: php-container
25     restart: always
26     volumes:
27      - ./website:/var/www/html
28     ports:
29      - "8888:80"
30     depends_on:
31      - db
```

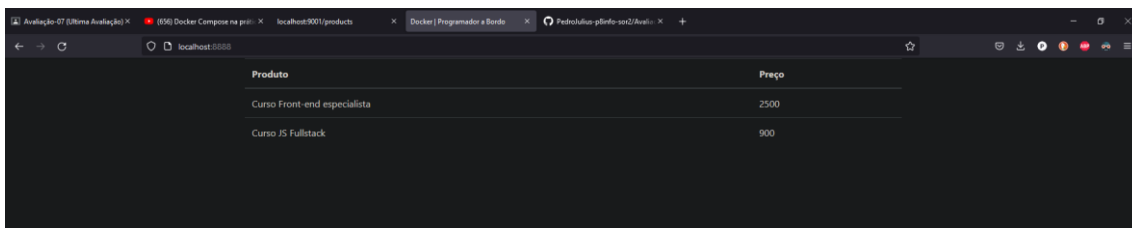
## Passo 04: Evidenciação dos containers rodando;

```
Pedro Julius@GOLD-GAMER MINGW64 ~/atividade-07/docker-introducao (master)
$ docker-compose up -d
Container mysql-container Created
Container node-container Created
Container php-container Created
Container mysql-container Starting
Container mysql-container Started
Container node-container Starting
Container node-container Started
Container php-container Starting
Container php-container Started
```

## Passo 05: Alimentando o Banco de Dados;

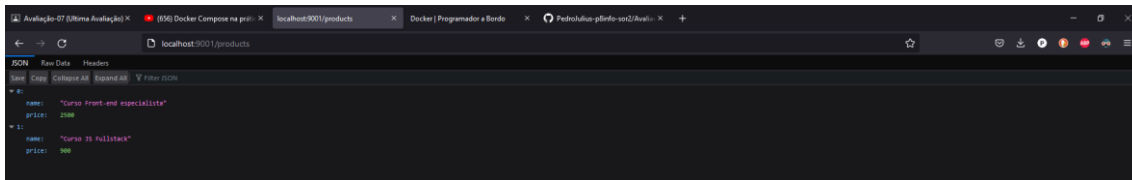
```
MINGW64:/c/Users/Pedro Julius/atividade-07/docker-introducao
Pedro Julius@GOLD-GAMER MINGW64 ~/atividade-07/docker-introducao (master)
$ docker exec -i mysql-container mysql -uroot -pprogramadorabordo < api/db/script.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
```

## Passo 06: Evidenciação dos arquivos rodando no localhost;



The screenshot shows a web browser window with the address bar set to localhost:8080. The page displays a table with two columns: 'Produto' and 'Preço'. The table contains two rows of data.

Produto	Preço
Curso Front-end especialista	2500
Curso JS Fullstack	900



The screenshot shows the same web browser window, but with the 'Raw Data' tab selected. It displays the JSON data returned by the API.

```
{
  "products": [
    {
      "name": "Curso front-end especialista",
      "price": 2500
    },
    {
      "name": "Curso JS Fullstack",
      "price": 900
    }
  ]
}
```