INSTITUTO INFNET
ESCOLA SUPERIOR DE TECNOLOGIA DA INFORMAÇÃO
GRADUAÇÃO EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

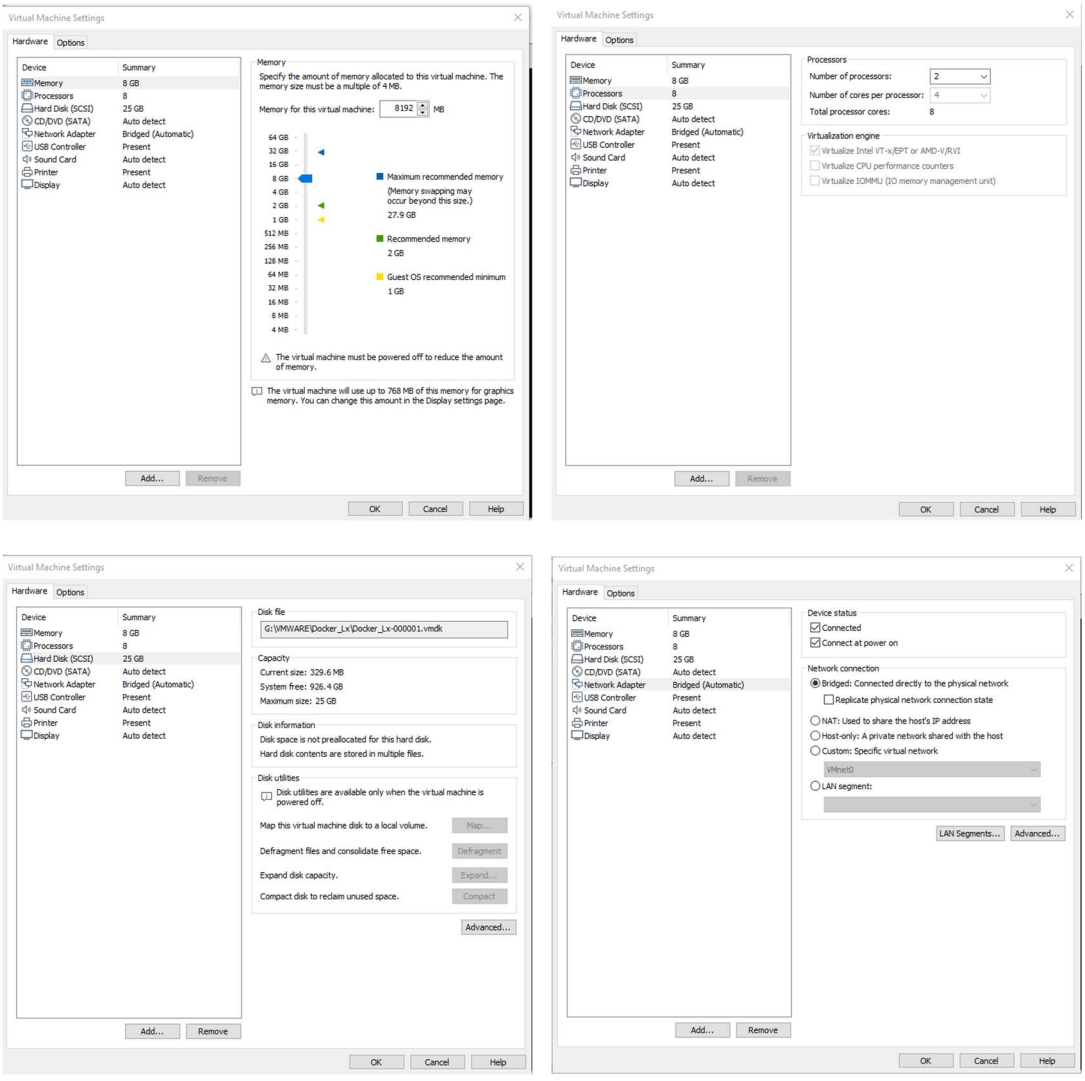
Projeto em Arquitetura de Infraestrutura de Aplicações

Aluno: Alexandre Garcia Barbeito Ferreira
Matrícula: 69165041100
E-MAIL: alexandre.pferreira@al.infnet.edu.br

TESTE DE PERFORMACE 04

Documente a execução de uma aplicação distribuída, composta por dois ou mais containers Docker. Seu trabalho deve incluir um texto de pelo menos uma página explicando que aplicação vai ser executada e quais containers serão usados para isto. O restante da entrega deve incluir uma documentação completa da execução dos containers, e uma captura de tela mostrando o resultado alcançado.

A aplicação escolhida será o WordPress. Ele rodará em um sistema Linux Ubuntu 18.04.03 em uma máquina virtual com as seguintes configurações:



O VMware workstation 15.5.1 foi hypervisor escolhido para criar a máquina virtual seguras e totalmente isoladas que irá encapsular o Ubuntu onde rodará Docker com seus conteiners. A camada de virtualização do VMware mapeia os recursos físicos de hardware para os recursos da máquina virtual, de forma que cada máquina virtual tenha seus próprios componentes, como CPU, memória, discos e dispositivos de entrada/saída.

Poderíamos rodar um sistema operacional ESXi e nele instalar o Ubuntu para execução dos containers, no entanto com essa configuração acredito que os objetivos do TP poderam ser atingindos.

Para execução da aplicação distribuída iremos instalar os containers através do Docker. Os containers instalados serão os seguintes:

- WordPress;
- Mysql.

A aplicação será acessada e configurada através de um cliente remoto Windows.

Segue as imagens de instalação e configuração do Docker e os containers Mysql e WordPress:

```
root@ubuntu:~# curl -sSl https://get.docker.com | sh
# Executing docker install script, commit: f45d7c11389849ff46a6b4d94e0dd1ffebca32c1
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/
dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
Warning: apt-key output should not be parsed (stdout is not a terminal)
+ sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" > /etc/apt/sources
.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ [ -n  ]
+ sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
```

Executando o script de instalação onde adiciona um repositório e fazer a instalação do pacote do Docker.

```
root@ubuntu:~# /etc/init.d/docker start
[ ok ] Starting docker (via systemctl): docker.service.
root@ubuntu:~# ps -ef | grep docker
root       4375      1  0 05:19 ?        00:00:00 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/
containerd.sock
root       6531   2115  0 05:24 pts/0    00:00:00 grep --color=auto docker
root@ubuntu:~#
```

Inicialização e check de execução do Docker.

```
root@ubuntu:~# sudo docker run --name banco -e MYSQL_ROOT_PASSWORD=dbsenha1234 -d mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
6d28e14ab8c8: Pull complete
dda15103a86a: Pull complete
55971d75ab8c: Pull complete
f1d4ea32020b: Pull complete
61420072af91: Pull complete
05c10e6ccca5: Pull complete
7e0306b13322: Pull complete
b2e2caa5b6cf: Pull complete
38d3c3421b5f: Pull complete
b8d532e0ef21: Pull complete
8be83ada689f: Pull complete
Digest: sha256:e4fd089298d0e81f7979a093328a4187a807244ff2ac485db46f857a2d037aa9
Status: Downloaded newer image for mysql:5.7
6b69f9a68c39ec7b1d0661e46f20d37de4c1697f857541d1f1d86316f719af42
root@ubuntu:~#
```

Instalação do container do banco de dados MySQL 5.7. e envio das variáveis de configuração para definir a senha root do banco.

```
root@ubuntu:~# sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
6b69f9a68c39        mysql:5.7           "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        3306/tcp, 33060/tcp      banco
root@ubuntu:~#
```

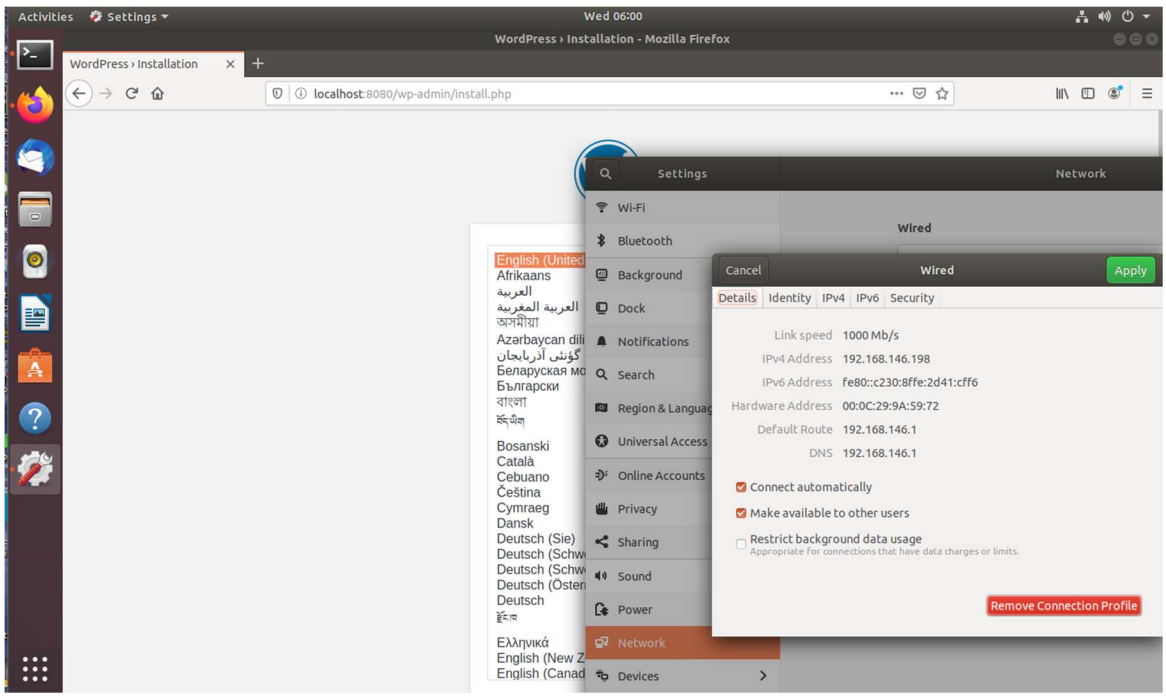
Verificação da execução do contêiner do banco de dados.

```
root@ubuntu:~# sudo docker run --name meusite --link banco:mysql -p 8080:80 -d wordpress
Unable to find image 'wordpress:latest' locally
latest: Pulling from library/wordpress
68ced04f60ab: Pull complete
1d2a5d8fa585: Pull complete
5d59ec4ae241: Pull complete
d42331ef4d44: Pull complete
408b7b7ee112: Pull complete
570cd47896d5: Pull complete
2419413b2a16: Pull complete
8c722e1dceb9: Pull complete
34fb68439fc4: Pull complete
e775bf0f756d: Pull complete
b1949a1e9661: Pull complete
6ed8bcec42ae: Pull complete
f6247da7d55f: Pull complete
a090bafe99ea: Pull complete
1499724c614a: Pull complete
838e071223d3: Pull complete
4f3f081f645a: Pull complete
5727cb8d10d6: Pull complete
77e0ad51ba4d: Pull complete
00c188d7a522: Pull complete
0421cc6f1038: Pull complete
Digest: sha256:6e17ef2ddd5ec3a0d4c8e86df409dc702db205330823df70518ce3f192e9b6c7
Status: Downloaded newer image for wordpress:latest
24b03c3fb95a54e55fe868d95480eae4b718129f505ed7210e940d54cc46546b
root@ubuntu:~#
```

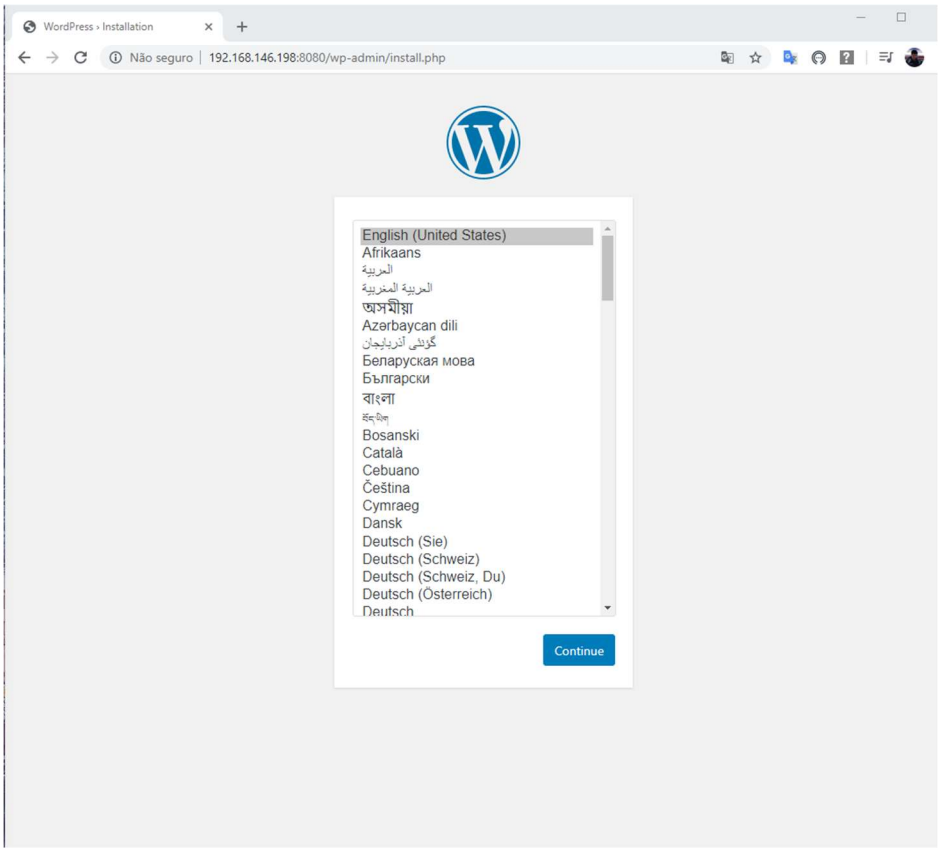
Agora vamos executar o segundo container com o WordPress com referência do container do banco de dados fazendo com que o WordPress recupere a senha do root do banco MySQL realizando a conexão sem a necessidade de configurações extra. O comando *"-p 8080:80"* fará com quando ocorra uma solicitação na porta 8080 do host está conexão seja encaminha para porta 80 do container. O comando *"-d wordpress"* carrega o container do WordPress realizando o seu download na primeira vez que for executado e executando-o em segundo plano.

```
root@ubuntu:~# sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
24b03c3fb95a        wordpress           "docker-entrypoint.s…"   38 seconds ago      Up 27 seconds       0.0.0.0:8080->80/tcp     meusite
6b69f9a68c39        mysql:5.7           "docker-entrypoint.s…"   21 minutes ago      Up 21 minutes       3306/tcp, 33060/tcp      banco
root@ubuntu:~#
```

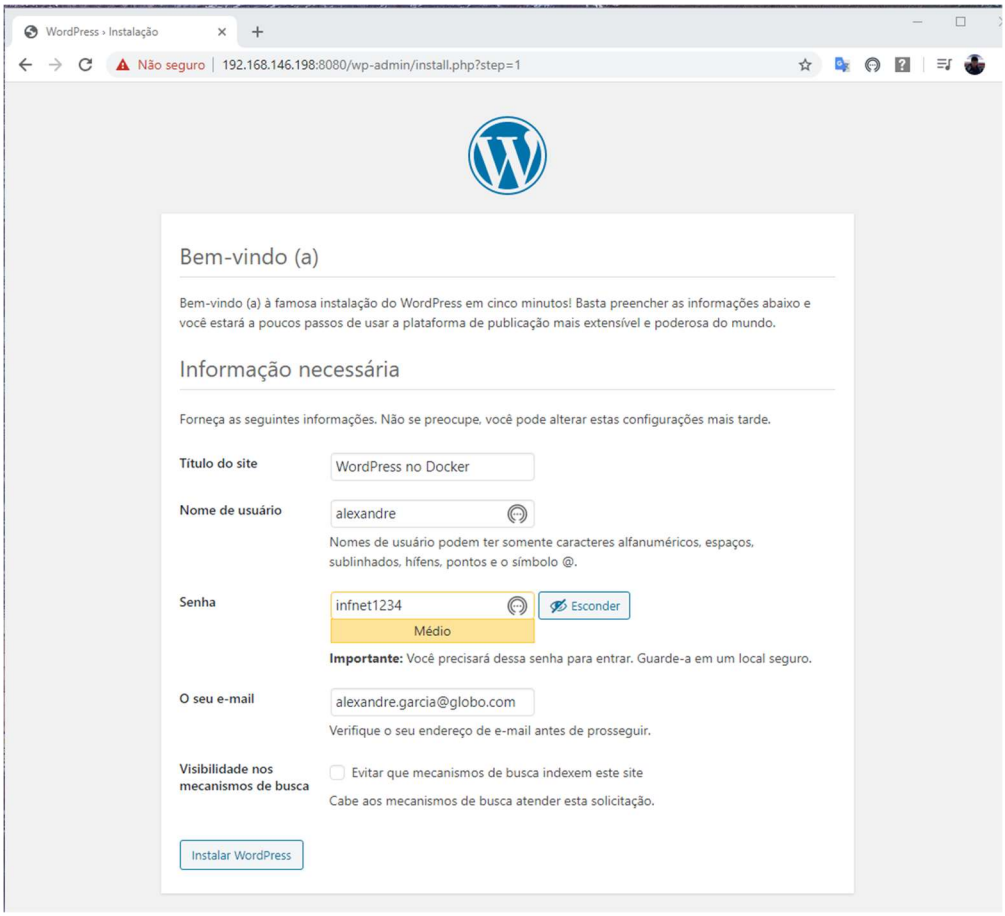
Agora podemos ver a execução dos dois containers e seus detalhes.



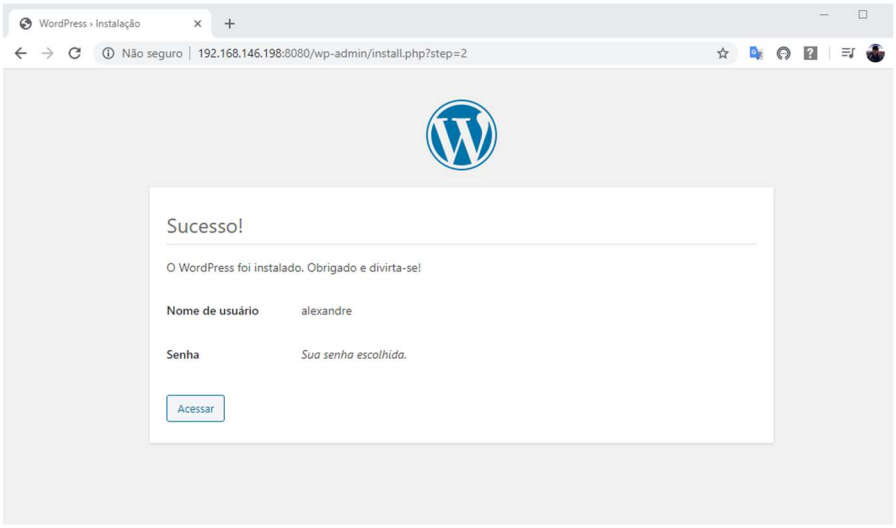
WordPress sendo executado no Mozila localmente.



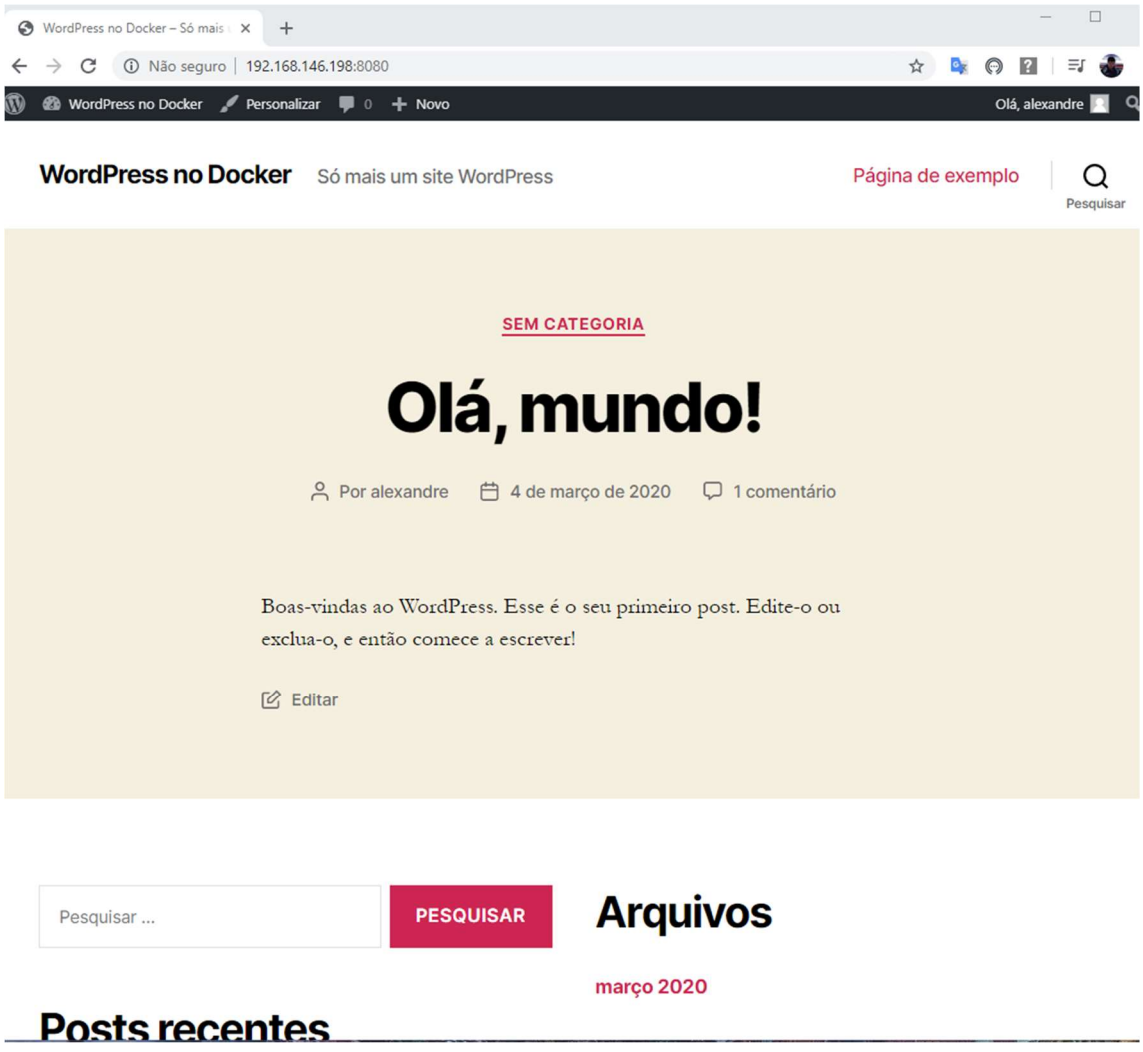
WordPress sendo executado no Google Chrome em um cliente remoto Windows 10.



Página do WordPress sendo configurada no cliente remoto.



Página no WordPress configurada.



Demonstração do WordPress já instalado e configurado no container.