

LABORATORIO 7

Alumno: Cesar Fabricio Tirado Vera

Código: 20222297

Ejercicio 2: Despliegue Local con Docker

2.1. Contenerizar la Base de Datos MySQL Localmente

Dentro del escritorio creé una carpeta llamada “mysql-init”, la cual contiene el script .sql que proporcionaron en el pdf del laboratorio 7.

```
Windows PowerShell
PS C:\Users\FABRICIO\Desktop> docker run -d --name mysql-trabajadores `
>> -e MYSQL_ROOT_PASSWORD=MyRootPass123! `
>> -e MYSQL_DATABASE=trabajadores_db `
>> -p 3307:3306 `
>> -v "${PWD}\mysql-init:/docker-entrypoint-initdb.d:ro" `
>> -v mysql_trabajadores_data:/var/lib/mysql `
>> mysql:8
5bf22ce8577e54ce2209b3d708b972c7269584541ad498da5dc694b0605073b2
```

Comando Docker run empleado:

```
docker run -d --name mysql-trabajadores `
-e MYSQL_ROOT_PASSWORD=MyRootPass123! `
-e MYSQL_DATABASE=trabajadores_db `
-p 3307:3306 `
-v "${PWD}\mysql-init:/docker-entrypoint-initdb.d:ro" `
-v mysql_trabajadores_data:/var/lib/mysql `
mysql:8
```

NOTA: Lo que realiza la línea “-v "\${PWD}\mysql-init:/docker-entrypoint-initdb.d:ro"” es que monta la carpeta mysql-init dentro del contenedor en /docker-entrypoint-initdb.d modo solo lectura (:ro). Asimismo, la línea “-v mysql_trabajadores_data:/var/lib/mysql” guarda todos los datos del contenedor MySQL en el volumen “mysql_trabajadores_data”, para que no se pierdan, aunque borre el contenedor.

```
PS C:\Users\FABRICIO\Desktop> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
5bf22ce8577e   mysql:8   "docker-entrypoint.s..." 41 seconds ago Up 40 seconds 0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp  mysql-trabajadores
PS C:\Users\FABRICIO\Desktop>
```

Verificamos que esté corriendo con “docker ps”. Corroboramos que efectivamente está corriendo “mysql:8”. Cabe recalcar que se utiliza el puerto 3307 que lo redirige al 3306, esto se debe a que localmente el puerto 3306 ya se estaba empleando, por lo que opté por usar el puerto 3307.

Verificamos que exista la tabla “trabajadores_db” y verificamos que también exista la tabla “trabajadores” y por último verificamos que existan datos en la tabla mencionada.

```
Windows PowerShell
PS C:\Users\FABRICIO\Desktop> docker exec -it mysql-trabajadores mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.7 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| trabajadores_db |
+-----+
5 rows in set (0.04 sec)

mysql> USE trabajadores_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_trabajadores_db |
+-----+
| trabajadores |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM trabajadores;
+----+-----+-----+
| id | nombre | cargo |
+----+-----+-----+
| 1 | Juan Perez | Desarrollador |
| 2 | Maria Garcia | Diseñadora |
| 3 | Carlos Lopez | Gerente de Proyecto |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

2.2. Construir y Ejecutar la Aplicación Web Localmente

```
PS C:\Users\FABRICIO\Desktop>
PS C:\Users\FABRICIO\Desktop> docker run -d --name miwebapp `
>> -p 8080:8080 `
>> -e SPRING_DATASOURCE_URL=jdbc:mysql://host.docker.internal:3307/trabajadores_db `
>> -e SPRING_DATASOURCE_USERNAME=root `
>> -e SPRING_DATASOURCE_PASSWORD=MyRootPass123! `
>> miwebapp:local
abc5914bfe1c8a954786c8c2d52e550414c16d0dd2f449ad50c8dfb83a9ae0c2
PS C:\Users\FABRICIO\Desktop>
```

Comando Docker run utilizado:

docker run -d --name miwebapp `

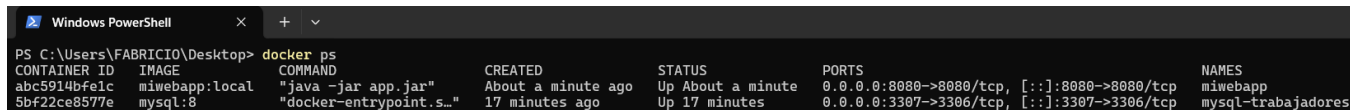
-p 8080:8080 `

-e SPRING_DATASOURCE_URL=jdbc:mysql://host.docker.internal:3307/trabajadores_db `

-e SPRING_DATASOURCE_USERNAME=root `

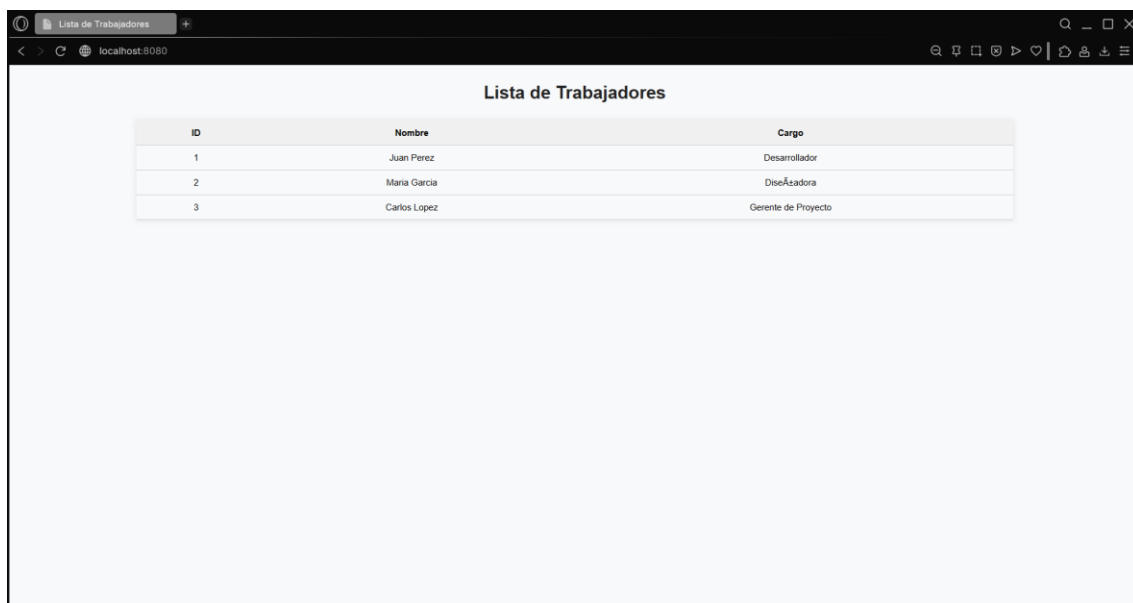
-e SPRING_DATASOURCE_PASSWORD=MyRootPass123! `

miwebapp:local



CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
abc5914bfe1c	miwebapp:local	"java -jar app.jar"	About a minute ago	Up About a minute	0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp	miwebapp
5bf22ce8577e	mysql:8	"docker-entrypoint.s..."	17 minutes ago	Up 17 minutes	0.0.0.0:3307->3306/tcp, [::]:3307->3306/tcp	mysql-trabajadores

Verificamos que la aplicación web esté corriendo con “docker ps”. Corroboramos que efectivamente está desplegada “miwebapp:local”. Por lo que procederemos a verificar ingresando a la url local “http://localhost:8080”.



ID	Nombre	Cargo
1	Juan Perez	Desarrollador
2	Maria Garcia	Diseñadora
3	Carlos Lopez	Gerente de Proyecto

Ejercicio 3: Despliegue Remoto en la Nube con Docker

3.1. Preparar la Instancia Remota

Instalamos Docker en la instancia remota

```
[ec2-user@ip-172-31-22-42 ~]$ sudo dnf install docker -y
Last metadata expiration check: 0:02:36 ago on Sat Oct 25 20:13:42 2025.
Dependencies resolved.
=====
Package                Arch      Version              Repository           Size
=====
Installing:
docker                 x86_64    25.0.8-1.amzn2023.0.6  amazonlinux          46 M
Installing dependencies:
container-selinux      noarch    3:2.233.0-1.amzn2023  amazonlinux          55 k
containerd             x86_64    2.0.6-1.amzn2023.0.1  amazonlinux          26 M
iptables-libs          x86_64    1.8.8-3.amzn2023.0.2  amazonlinux          401 k
iptables-nft           x86_64    1.8.8-3.amzn2023.0.2  amazonlinux          183 k
libcgroup              x86_64    3.0-1.amzn2023.0.1    amazonlinux           75 k
libnetfilter_conntrack x86_64    1.0.8-2.amzn2023.0.2  amazonlinux           58 k
libnftnl               x86_64    1.0.1-19.amzn2023.0.2  amazonlinux           30 k
libnftnl               x86_64    1.2.2-2.amzn2023.0.2  amazonlinux           84 k
pigz                   x86_64    2.5-1.amzn2023.0.3    amazonlinux           83 k
runc                   x86_64    1.2.6-1.amzn2023.0.1  amazonlinux          3.7 M
=====
Transaction Summary
=====
Install 11 Packages

Total download size: 77 M
Installed size: 292 M
Downloading Packages:
(1/11): container-selinux-2.233.0-1.amzn202 1.8 MB/s | 55 kB    00:00
(2/11): iptables-libs-1.8.8-3.amzn2023.0.2. 8.6 MB/s | 401 kB  00:00
```

Verificamos que esté habilitada y corriendo (running)

```
[ec2-user@ip-172-31-22-42 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-10-25 20:22:29 UTC; 1h 10min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 28282 (dockerd)
      Tasks: 24
     Memory: 55.9M
        CPU: 15.837s
    CGroup: /system.slice/docker.service
            └─28282 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nfile=32768:65536
              └─28868 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3306 -container-ip 172.17.0.2 -container-port 3306
                └─28880 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 3306 -container-ip 172.17.0.2 -container-port 3306

Oct 25 20:22:29 ip-172-31-22-42.ec2.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.101627353Z" level=info msg="Starting up"
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.150247321Z" level=info msg="Loading containers: start."
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.569318799Z" level=info msg="Loading containers: done."
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.589017045Z" level=info msg="Docker daemon" commit=71907ca containerd=
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.589207741Z" level=info msg="Daemon has completed initialization"
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal dockerd[28282]: time="2025-10-25T20:22:29.620434063Z" level=info msg="API listen on /run/docker.sock"
Oct 25 20:22:29 ip-172-31-22-42.ec2.internal systemd[1]: Started docker.service - Docker Application Container Engine.
[lines 1-2/22 (END)]
```

3.2. Desplegar la Base de Datos MySQL Remotamente

Procedemos a inicializar la Base de Datos MySQL remotamente

```
[ec2-user@ip-172-31-22-42 ~]$ sudo docker run -d --name mysql-trabajadores \
-e MYSQL_ROOT_PASSWORD=MyRootPass123! \
-e MYSQL_DATABASE=trabajadores.db \
-p 3306:3306 \
-v $(pwd)/mysql-init:/docker-entrypoint-initdb.d:ro \
-v mysql-trabajadores_data:/var/lib/mysql \
mysql:8
Unable to find image 'mysql:8' locally
8: Pulling from library/mysql
823a182c62a0: Pull complete
f5f78fcd9ccb: Pull complete
494c372d15c3: Pull complete
0d0f66273639: Pull complete
6d7d6108636e: Pull complete
3763684269ad: Pull complete
f7c524da3882: Pull complete
824d865d5643: Pull complete
270bbb610640: Pull complete
fe8b6e0a15a: Pull complete
Digest: sha256:b306273dd4d6b1a7f265138c3dd493c0462253af7634203e569add0403a7b273
Status: Downloaded newer image for mysql:8
cfd6bb05f86485c6b1ef02c17a45aeb3b73fa2d8c7332belf4d7394elaaebf1
[ec2-user@ip-172-31-22-42 ~]$
[ec2-user@ip-172-31-22-42 ~]$
[ec2-user@ip-172-31-22-42 ~]$ docker ps
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/container
s/json": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-22-42 ~]$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS
cfd6bb05f864   mysql:8   "docker-entrypoint.s..." 5 minutes ago    Up 5 minutes    0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql-trabajadores
[ec2-user@ip-172-31-22-42 ~]$
```

Comando docker utilizado:

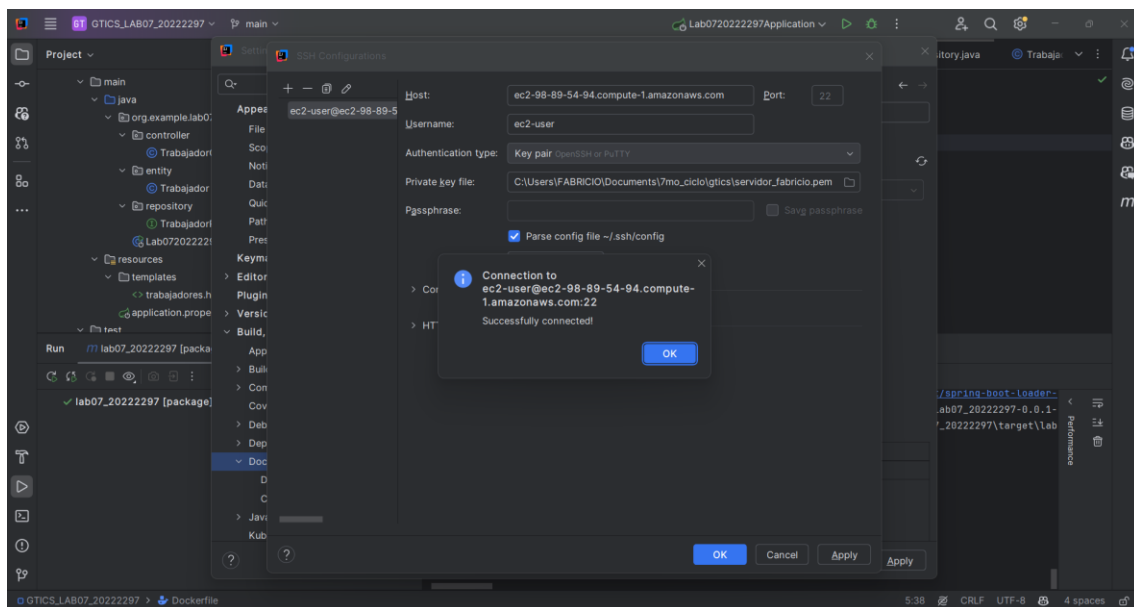
```
sudo docker run -d --name mysql-trabajadores \
-e MYSQL_ROOT_PASSWORD=MyRootPass123! \
-e MYSQL_DATABASE=trabajadores_db \
-p 3306:3306 \
-v $(pwd)/mysql-init:/docker-entrypoint-initdb.d:ro \
-v mysql_trabajadores_data:/var/lib/mysql \
mysql:8
```

Verificamos que esté corriendo con “docker ps”. Corroboramos que efectivamente está corriendo “mysql:8”.

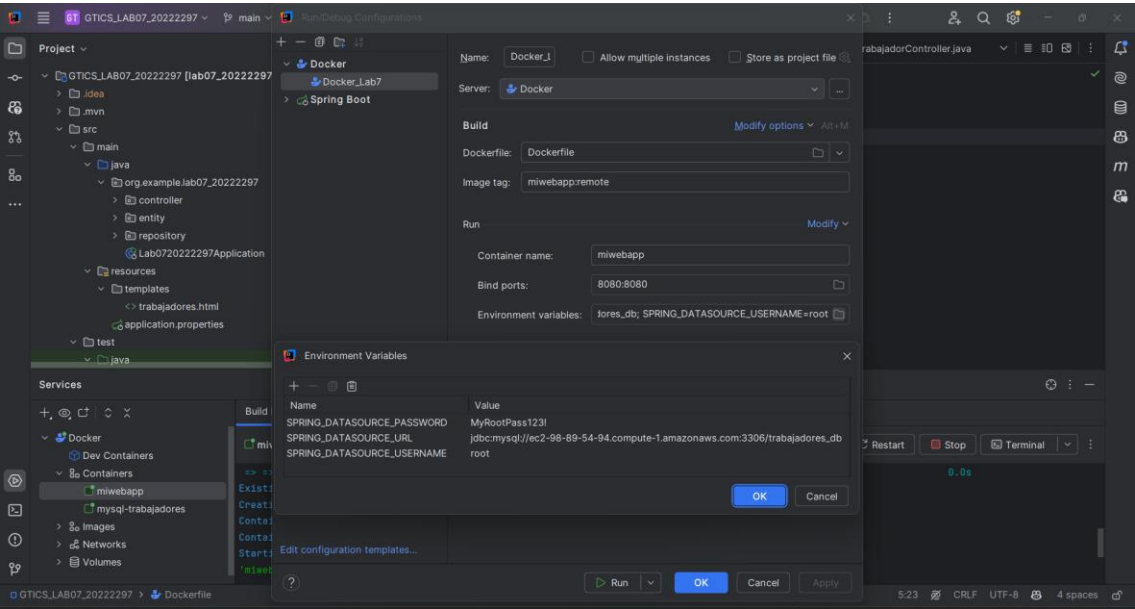
```
[ec2-user@ip-172-31-22-42 ~]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
cfd6bb05f864   mysql:8   "docker-entrypoint.s..." About an hour ago Up About an hour   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql-trabajadores
[ec2-user@ip-172-31-22-42 ~]$
```

3.3. Desplegar la Aplicación Web Remotamente con IntelliJ

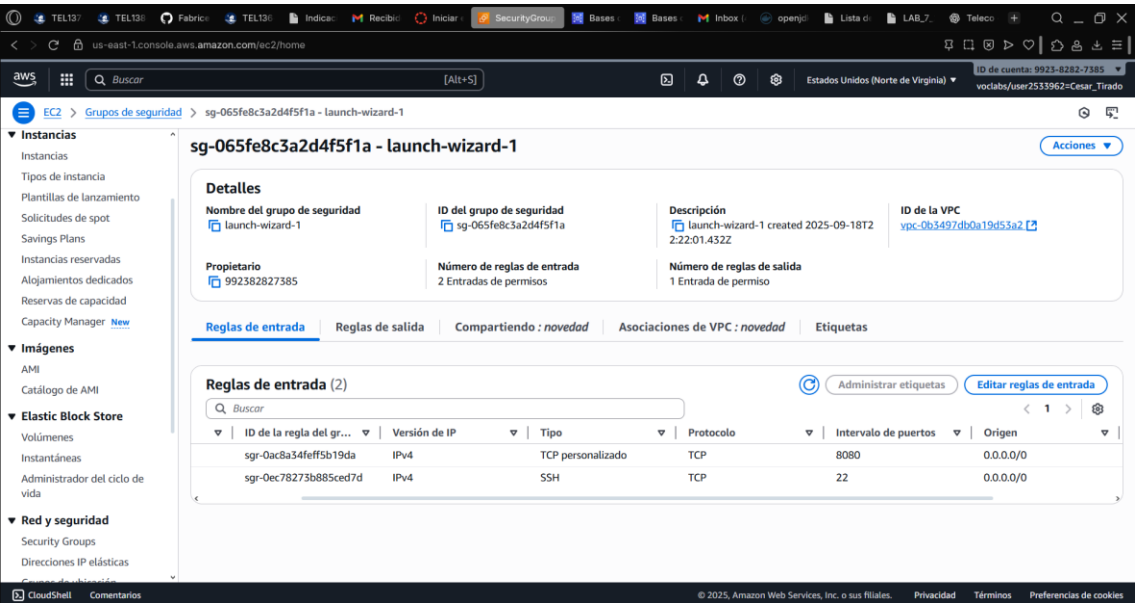
Configuramos un nuevo Docker daemon remoto para poder conectarme a mi instancia de nube a través de SSH.



Configuramos las variables de entorno `SPRING_DATASOURCE_PASSWORD`, `SPRING_DATASOURCE_URL` y `SPRING_DATASOURCE_USERNAME`.

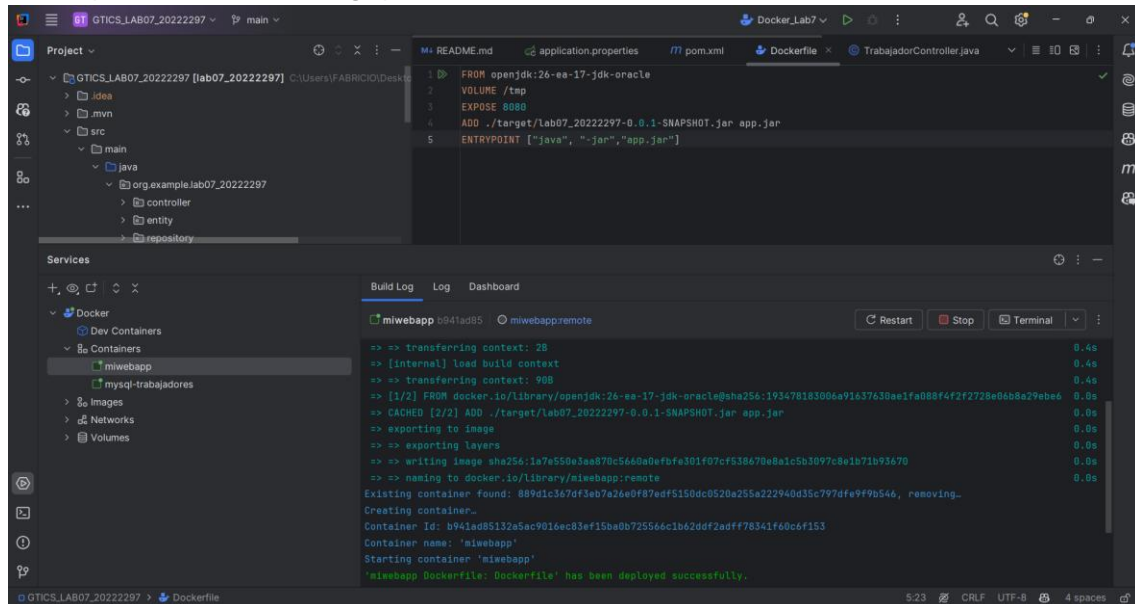


Configuramos las reglas de seguridad de nuestro proveedor de nube para permitir el tráfico entrante al puerto 8080.

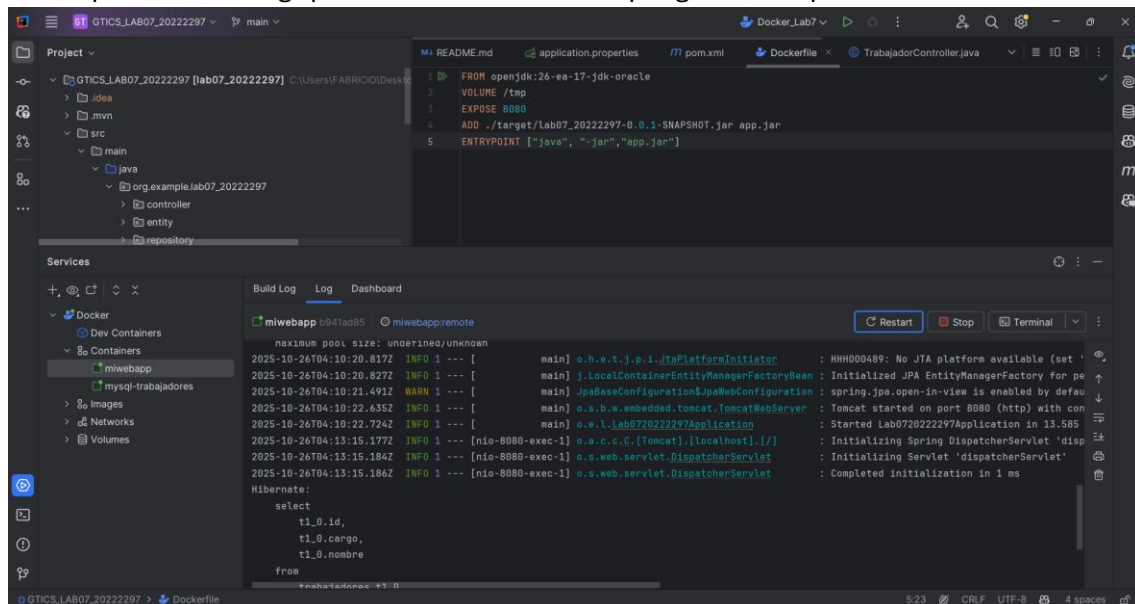


Desplegando la aplicación web utilizando la configuración remota de IntelliJ.

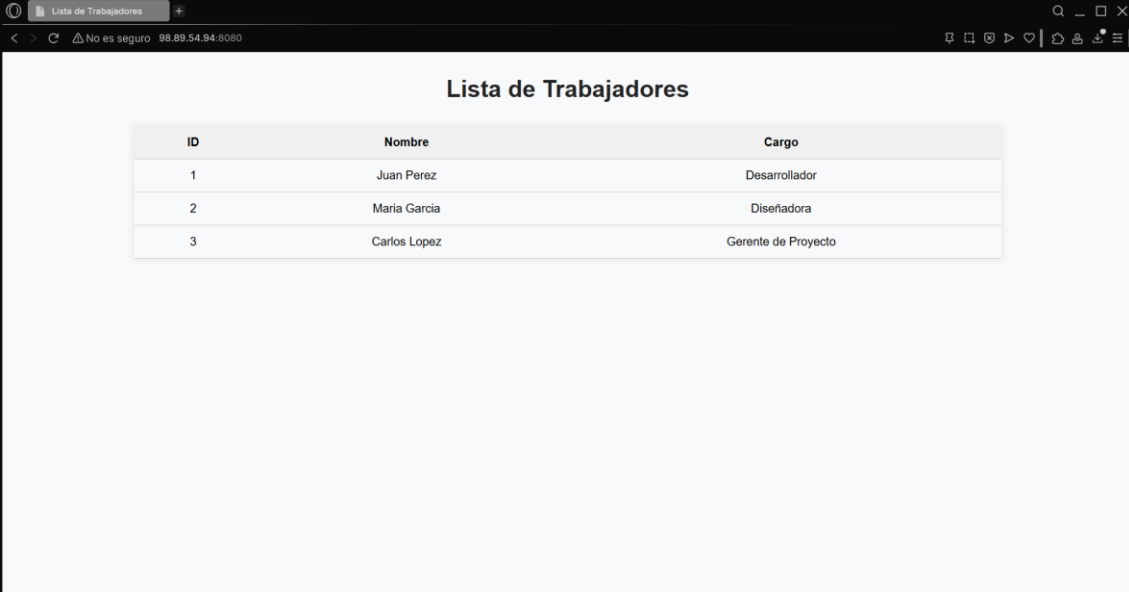
En el apartado de “Build Log” podemos observar la construcción del contenedor.



En el apartado de “Log” podemos observar el despliegue de la aplicación web.



Finalmente, para corroborar el despliegue, nos dirigimos a nuestro navegador y colocamos la IP pública de nuestra instancia creada: <http://98.89.54.94:8080>.



The screenshot shows a web browser window with the address bar displaying 'No es seguro 98.89.54.94:8080'. The page title is 'Lista de Trabajadores'. The main content area displays a table with the following data:

ID	Nombre	Cargo
1	Juan Perez	Desarrollador
2	Maria Garcia	Diseñadora
3	Carlos Lopez	Gerente de Proyecto

Como puede observarse, la aplicación funciona y muestra la lista de trabajadores, obteniendo los datos del contenedor MySQL remoto.