**Examen Final Java Docker Developer**

**Nombre : Christian Eder Vladimir Gutiérrez Salas**

El Archivo comprimido contiene:

1. - La carpeta docker-compose: contiene el archvio docker-compose.yml

2. - La carpeta dockerFile-apiSeguridad: contiene el docker file del api de seguridad

3. - La carpeta dockerFile-apiMicro1: contiene el docker file de la api de consulta de médicos.

4. - La carpeta dockerFile-apiMicro2: contiene el docker file de la api de consulta de especialidades.

5. - La carpeta dockerFile–database: contiene el docker file de creación de la base de datos postgres

6. - La carpeta dockerFile-proxy: contiene el docker file archivo nginx del server proxy

7. - La carpeta front\_micro: contiene las fuentes y el docker file del frontend.

8. - La carpeta docker-swarn-compose: contiene el docker-compose.yml para swarm.

**EJECUTAR DOCKER COMPOSE**

En la carpeta docker-compose se encuentra el archivo docker-compose.yml

para probarlo se sigue los siguientes pasos:

1.- Abrir un terminal y navegar hasta el directorio del archivo docker-compose.yml

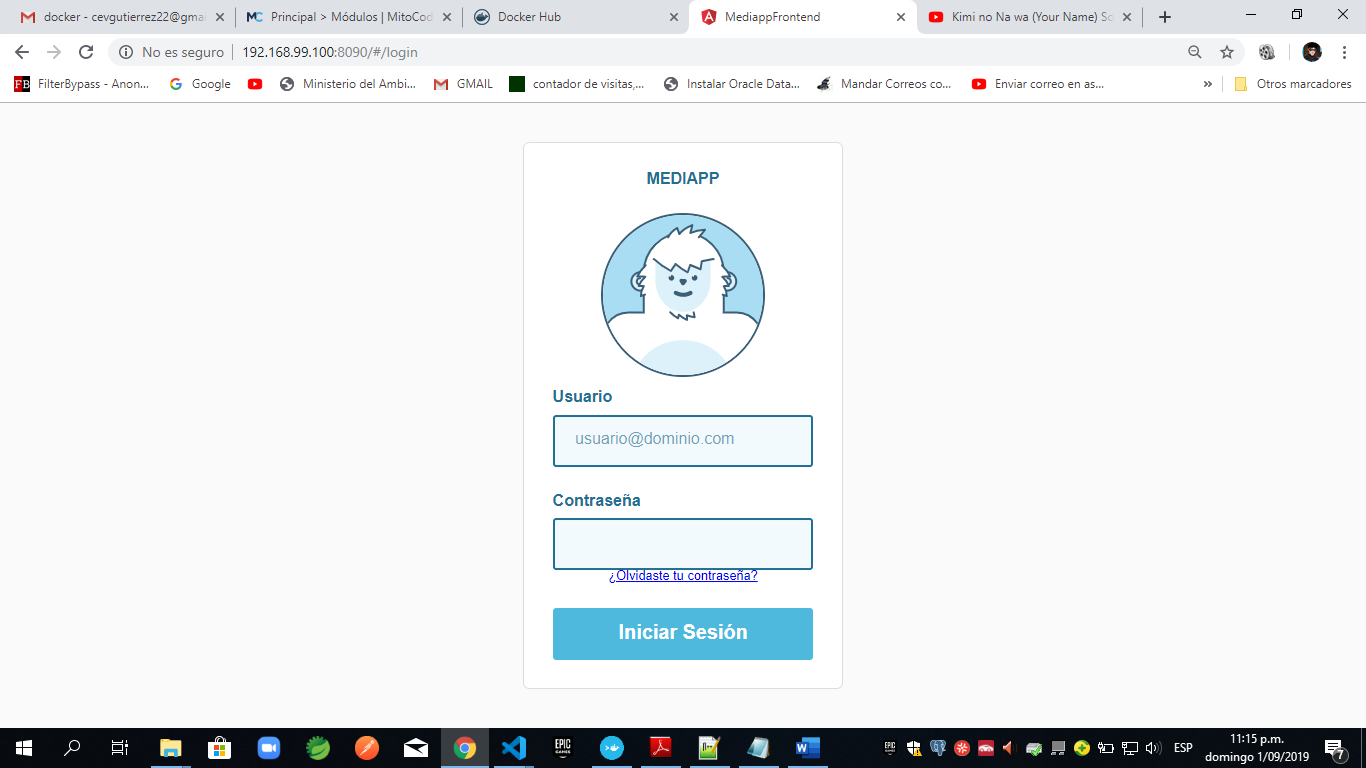
2.- Ejecutar el siguiente comando : docker-compose up -d

Para probar el aplicativo se ingresa en el navegador web <http://192.168.99.100:8090> (se ha utilizado Docker for Windows)

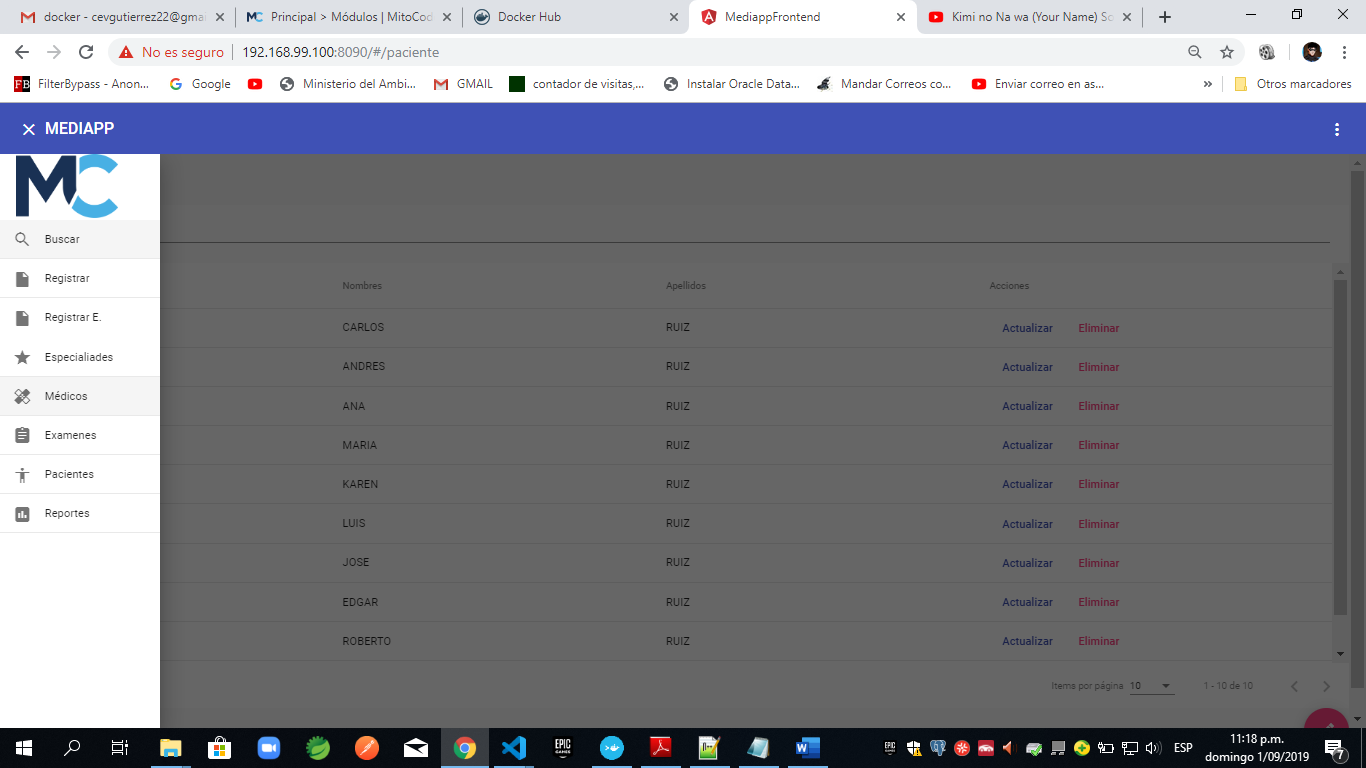
Se ingresa las credenciales de acceso:

usuario: cevgutierrez

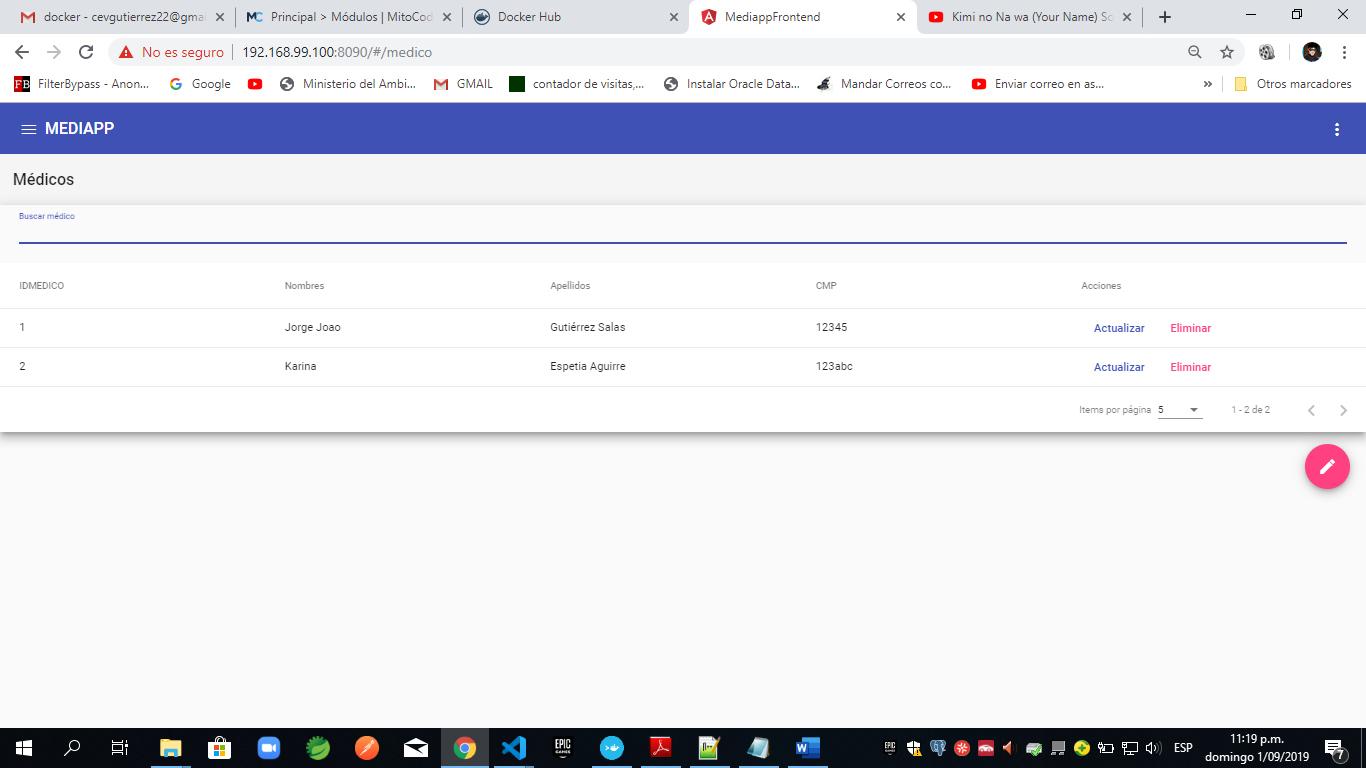
contraseña : 123456



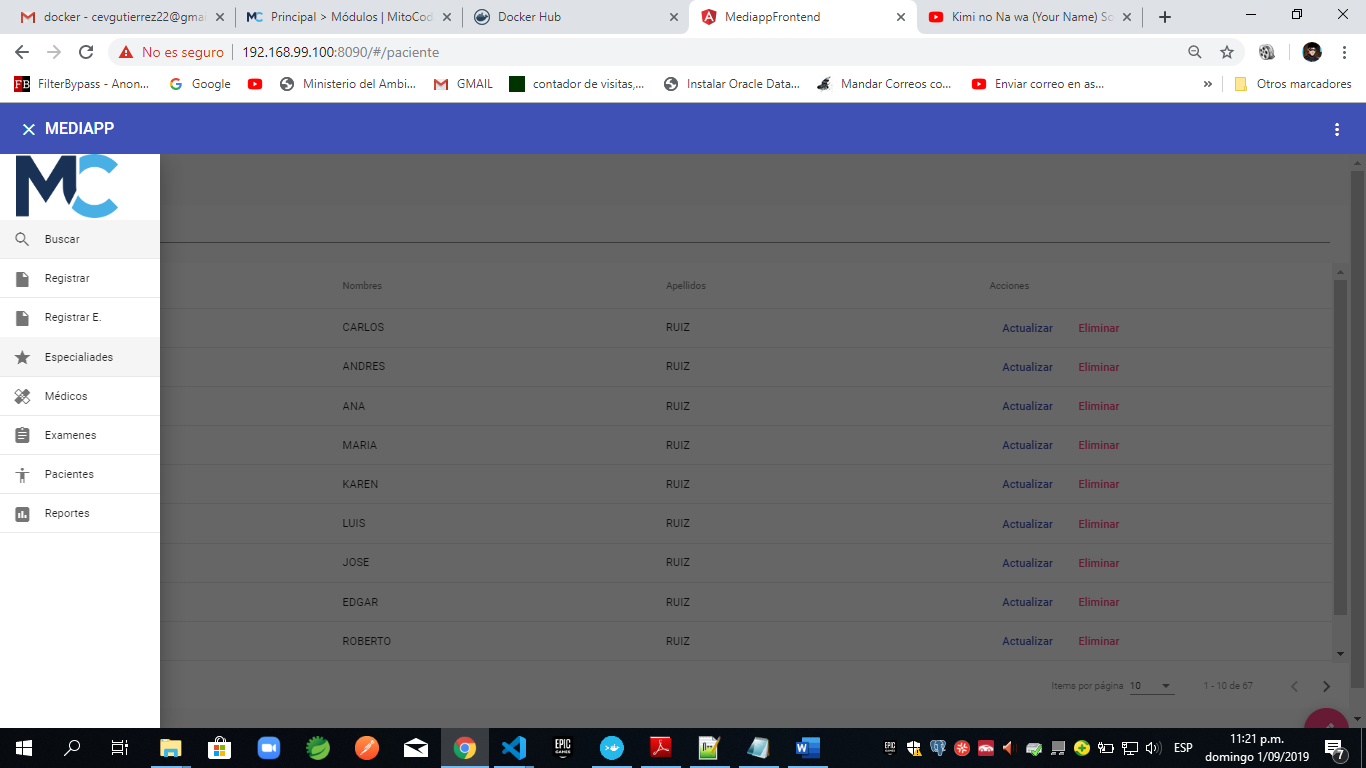
En el Menu lateral escoger la opción medicos :



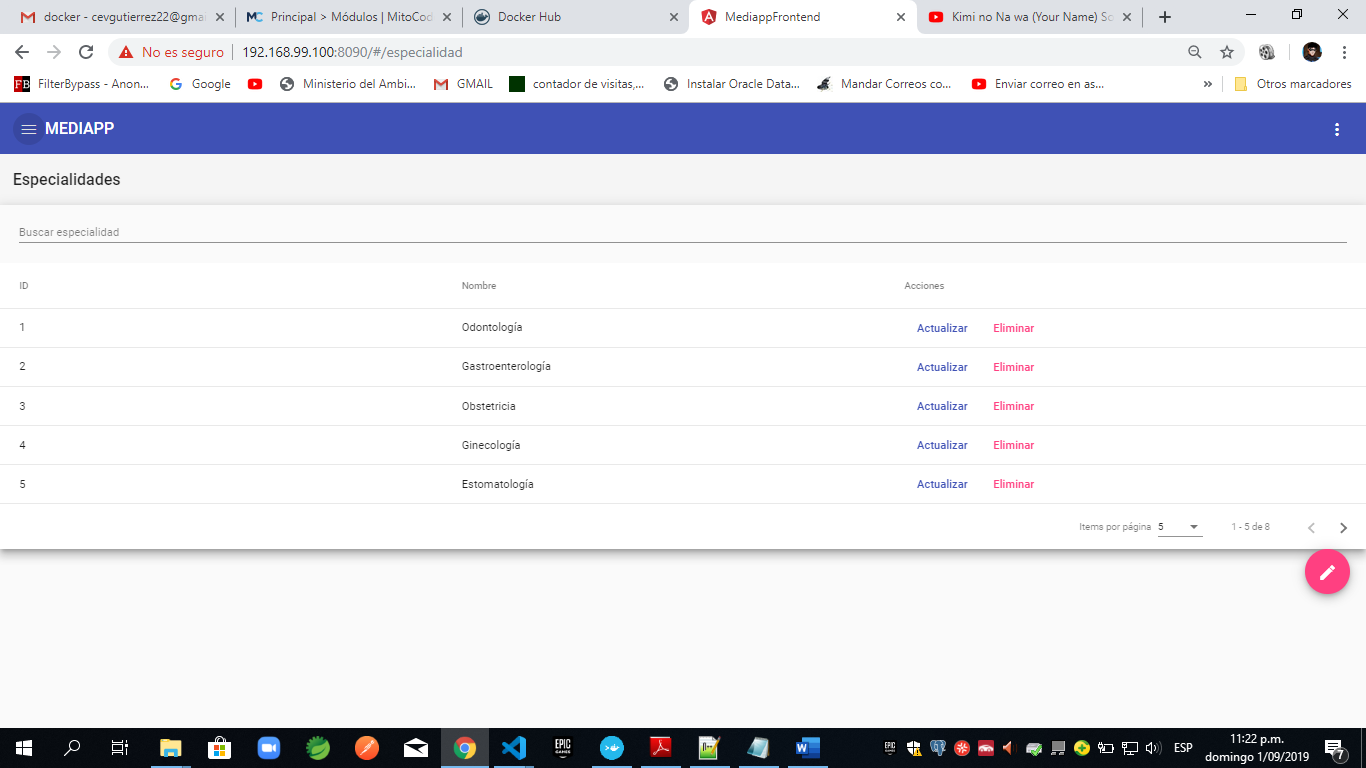
En la pantalla Medicos se listará la consulta de médicos (**apimicro1**)



En el Menu lateral escoger la opción **especialidades** :



En la pantalla especialidades se listará la consulta de especialidades (**apimicro2**)



**EJECUTAR CLUSTER DOCKER SWARM**

**1.- Crear maquinas virtuales :**

Se ejecutan los siguientes comandos:

Se crea el manager1 :

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso manager1

Se crea el manager2 :

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso manager2

Se crea el worker1 :

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso worker1

Se crea el worker2 :

docker-machine create --driver virtualbox --virtualbox-boot2docker-url=https://github.com/boot2docker/boot2docker/releases/download/v18.05.0-ce/boot2docker.iso worker2

------------------------------------------------------------------------------------------------------------------------

***Se inicia el docker swarn:***

docker-machine ssh manager1 "docker swarm init --advertise-addr $(docker-machine ip manager1)"

para agregar workers

docker swarm join --token SWMTKN-1-1o13zfz3anp4vm0knscrgpmm9lsjajqm7oo68mib4e9c97vj2l-4eqkchhlh1v42cxle5exmzv1v 192.168.99.101:2377

para agregar managers

docker swarm join --token SWMTKN-1-1o13zfz3anp4vm0knscrgpmm9lsjajqm7oo68mib4e9c97vj2l-2y48fdw8epwfiivp9wl8xza8l 192.168.99.101:2377

***agregando manager 2 a manager 1***

docker-machine ssh manager2 "docker swarm join --token SWMTKN-1-1o13zfz3anp4vm0knscrgpmm9lsjajqm7oo68mib4e9c97vj2l-2y48fdw8epwfiivp9wl8xza8l 192.168.99.101:2377"

***agregando a worker1***

docker-machine ssh worker1 "docker swarm join --token SWMTKN-1-1o13zfz3anp4vm0knscrgpmm9lsjajqm7oo68mib4e9c97vj2l-4eqkchhlh1v42cxle5exmzv1v 192.168.99.101:2377"

***agregando a worker2***

docker-machine ssh worker2 "docker swarm join --token SWMTKN-1-1o13zfz3anp4vm0knscrgpmm9lsjajqm7oo68mib4e9c97vj2l-4eqkchhlh1v42cxle5exmzv1v 192.168.99.101:2377"

**Verificando que el cluster esté activo:**

docker-machine env manager1

Ejecutar comando que nos indica :

eval $(docker-machine env manager1)

verificar el cluster:

docker node ls

**Ejecutar el archivo docker-compose.yml**

Navegar al directorio Docker Swarn compose

Luego ejecutar el siguiente comando desplegar el aplicativo en el cluster docker swarm:

docker stack deploy -c docker-compose.yml mediapp

Se debe verificar el despliegue con los siguientes comandos:

docker stack ps mediapp

docker service ls

Verificar los log de la base de datos:

docker service logs -f mediapp\_postgres\_server

Verificar los log del api de seguridad:

docker service logs -f mediapp\_apiseguridad

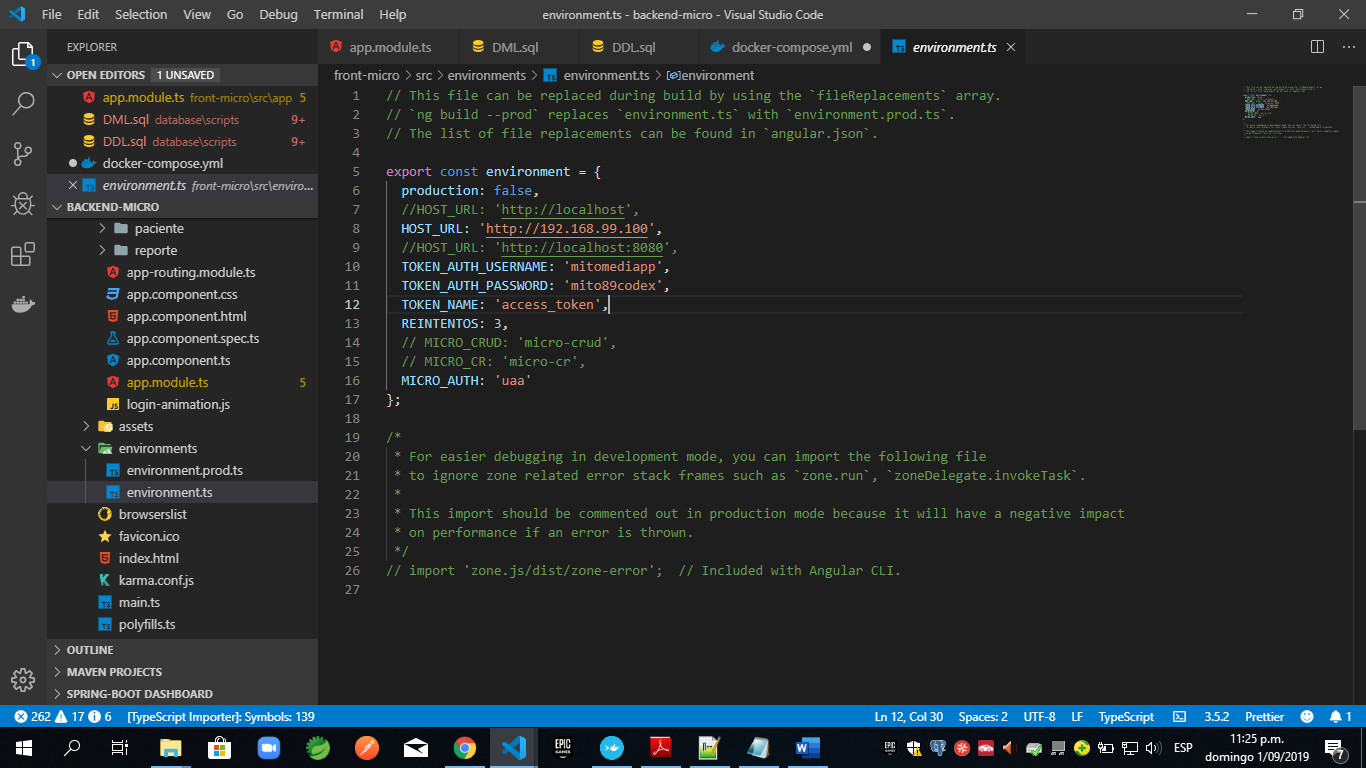
se puede verificar los log del api de apimicro1:

docker service logs -f mediapp\_apimicro1

se puede verificar los log del api de apimicro2:

docker service logs -f mediapp\_apimicro2

Se levanta el front-end del angular y cambiamos la constante HOST por el ip del manage1:



y se realizan las pruebas.