1. PARA BORRAR TODOS LOS CONTENEDORES INCLUYENDO SU USO DE VOLUMEN.

docker rm -vf $(Docker ps -a -q)

1. PARA BORRAR TODAS LAS IMAGENES

Docker rmi -f $(docker images -a -q)

1. PARAR ELIMINAR TODO.

Docker system Prune -a –volumes

1. PARA VER LAS IMAGENES DISPONIBLES

Docker images

1. MUESTRA QUE CONTENEDORES ESTAN FUNCIONANDO

Docker ps -a

1. MUESTRA INFORMACION DE LAS IMAGENES Y TAMAÑO FECHA DE CREACION

Docker info

1. INICIAR EL CONTENDOR

Docker start

1. PARAR EL CONTENEDOR

Docker stop

1. Para levantar Docker compose

docker-compose up -d

levantar con dockerfile

docker build -t “ingnolasco/edwinnolasc:desa1”

docker build -t

docker-compose -f docker-compose.yml -p prueba up -d –build

/\*desde el docker file \*/

docker build -t aspnetapp .

docker run -it --rm -p 5000:80 --name aspnetcore\_sample aspnetapp



1. COMANDO DESCARGAR IMAGEN RABBIT

docker pull rabbitmq:3-management

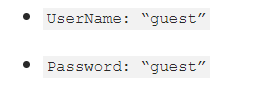
1. CONFIGURAMOS EL PUERTO DE SALIDA DE DOCKER

docker run -d -p 15672:15672 -p 5672:5672 --name rabbit-test-for-medium rabbitmq:3-management

1. ENTRAMOS EN LA SIGUIENTE RUTA

[http://localhost:15672](http://localhost:15672/#/)

1. CREDENCIALES DE USUARIO





1. DECARGAR IMAGEN SONAR

docker pull sonarqube

1. PARA INICIAR LA APLICACION DE SONAR

Docker run -d –name sonarqube -p 9000:9000 <imagen\_name>

1. ENTRAMOS EN LA SIGUIENTE URL

<http://localhost:9000/>

1. CREDENCIALES PARA SONAR

