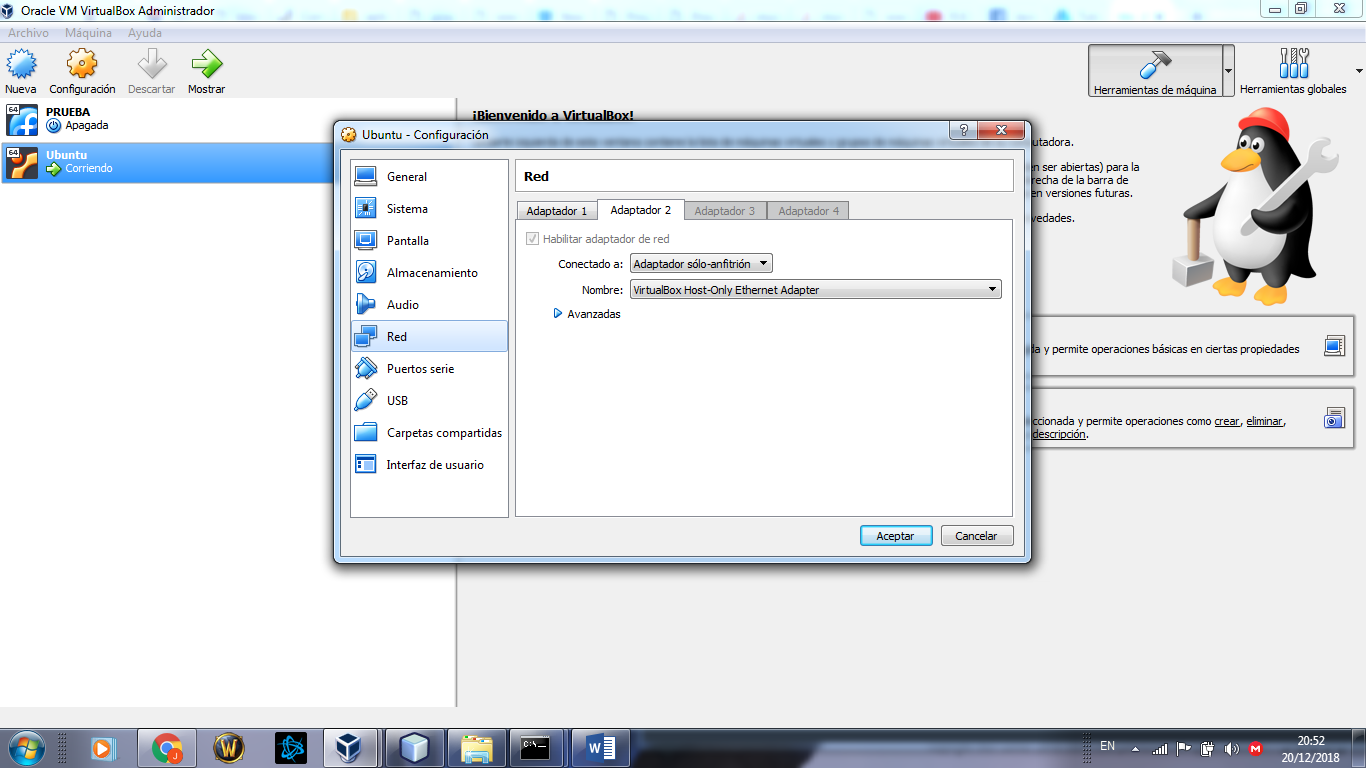
**Comandos utilizados para el levantamiento de la Base de Datos en un contenedor de Docker**

La forma en la que realizamos la conexión fue a través de Virtualbox, en el cual simulamos un servidor de Ubuntu, por lo general la configuración inicial de Ubuntu solo tiene habilitado un adaptador (NAT), el cual es el que permite únicamente la conexión desde esa máquina virtual hacia el Internet, para poder lograr que también funcione como un servidor, hay que habilitar el segundo adaptador y ponerlo con la opción de “Adaptador sólo-anfitrión”.



Una vez realizado esto, mandamos a correr la máquina virtual. Es importante destacar que se debe tener instalado mysql-server en la máquina virtual.

Una vez dentro le configuramos la ip tanto del VirtualBox Host-OnlyNetwork como de la Conexión de Área Local del servidor que utiliza como puerta la ip del Host-OnlyNetwork, nosotros les pusimos las ips fijas de 192.168.5.1 y 192.168.5.2 respectivamente, incluyendo otra configuración de las redes como máscaras de red, etc.

Aparte de eso hay que instalar Docker, para eso vamos a hacer lo siguiente:

1. *Sudo apt update*
2. *Sudo apt install apt-transport-https ca-certificates curl software-properties-common*
3. *Curl –fsSL* [*https://download.docker.com/linux/ubuntu/gpg*](https://download.docker.com/linux/ubuntu/gpg) *|sudo apt-key add -*
4. *Sudo add-apt-repository “deb [arch=amd64]* [*https://download.docker.com/linux/ubuntu*](https://download.docker.com/linux/ubuntu) *bionic stable”*
5. *Sudo apt update*
6. *Apt-cache policy docker-ce*
7. *Sudo apt install docker-ce*
8. *Sudo systemctl status docker*

Ahora una vez que tengamos instalado Docker, lo que se hizo fue crear un archivo DockerFile, del cual se va a generar una imagen en la computadora y apartir de la cual vamos a crear nuestro contenedor para poder ejecutar el servidor que contiene la Base de Datos. El DockerFile se lo realizo así:

*FROM mysql:latest*

*ENV MYSQL\_ROOT\_PASSWORD guerrero*

*ENV MYSQL\_DATABASE TecnoImportDB*

*ENV MYSQL\_USER guerrero*

*ENV MYSQL\_PASSWORD guerrero10*

*ADD tables.sql / docker-entrypoint-initdb.d*

*EXPOSE 3366*

La imagen que va a generar el DockerFile utiliza una imagen ya existente en el repositorio de Docker, que es mysql en su última versión, se utilizan comandos especiales que tienen el efecto de que cuando se genere el mysql en la nueva imagen, va a generar un usuario “guerrero” con contraseña “guerrero10”, creara una base de datos con el nombre de TecnoImportDB, cambiara la contraseña del usuario Root a “guerrero” y añadirá el archivo .sql a las bases de datos que tiene registrado el mysql en la misma base que se especificó con el nombre del comando ENV MYSQL\_DATABASE.

Luego para poder crear una imagen de ese DockerFile, se usa el comando:

*docker build –t jaanrosa/mysql03 .*

El punto final indica la ubicación de mi archive DockerFile que indica el propio directorio, y jaanrosa/mysql03 es el nombre de la imagen que se creara localmente.

Luego se procede a utilizar el siguiente comando para poder crear el contenedor:

*Docker run –name dbf –p3306:3306 –d jaanrosa/mysql03*

En donde el nombre del contenedor sera dbf, habilitamos el Puerto 3306 (si ya estan siendo utilizados debo matar a los procesos que lo usan) y lo ejecutamos como un daemon.

Se pone el comando: *docker ps* para comprobar que ha sido creado satisfactoriamente

Una vez que el contenedor ya ha sido creado, lo que se hace es entrar al bash del contenedor usando esto:

*docker exec –it dbf /bin/bash*

Luego te botara del contenedor y para volver a ingresar primero debes levantarlo con:

*Docker start <ID del contenedor>*

Luego vuelvas a usar esto:

*docker exec –it dbf /bin/bash*

Ahora lo que haces es ingresar al mysql con los usuarios y contrasenas creados en el DockerFile y ya estaría levantado y listo el servidor de la base de datos dentro del contenedor.

Nota: Puedes matar a un proceso de docker que use esos puertos con: *docker stop <Nombre del contenedor>*. Para matar procesos ajenos a docker que usen el puerto 3306 lo que se hace es *sudo service mysql stop*