

# Laboratorio de administración y gestión de redes y sistemas

## Examen práctico

30 de junio de 2021.

Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

---

- Entra en tu cuenta del laboratorio y ejecuta

```
~mortuno/prepara
```

- Esto creará los ficheros

```
~/lagrs.julio.21/final.txt
```

```
~/lagrs.julio.21/conexiones.py
```

Contesta las preguntas en estos dos ficheros, en ningún otro sitio.

## Ejercicio 1 (6 puntos)

En este ejercicio indicarás todo lo necesario para preparar un contenedor docker según la siguiente especificación:

1. Deseamos que al ponerse en marcha el contenedor, el usuario *root* tenga una sesión de shell abierta.
2. El usuario podrá ejecutar la orden *maquina*. Será un script de shell que simplemente escriba en la salida estándar el mensaje *Estás en TU-HOST*, donde *TU-HOST* será el nombre del *host*, obtenido de la variable de entorno correspondiente <sup>1</sup>.
3. El *path* completo del script *maquina* será */opt/examen/bin/maquina*.
4. Como seguramente sabes, para que el usuario pueda ejecutar la orden tecleando *maquina* ( y no un trayecto completo como */opt/examen/bin/maquina*), tendrás que añadir cierta línea al fichero *.bashrc* del usuario *root*.

Haz esto de forma similar a lo que hiciste en la práctica 1.20 para añadir una línea a */etc/host*. Esto es:

- a) Prepara, en el directorio contexto, un fichero *delta\_bashrc* que contenga la línea adecuada.
  - b) Haz que este fichero aparezca en el directorio */tmp/* de la imagen.
  - c) Haz que cuando se prepare la imagen del contenedor, se añada el contenido de *delta\_bashrc* al fichero *.bashrc* del usuario *root*.
5. El contenedor estará construido a partir de *ubuntu 20.04*, con los paquetes actualizados a la última versión disponible. No tendrá ningún otro paquete ni ninguna configuración adicional.

---

<sup>1</sup>En el caso particular de este examen, el nombre del host lo conoces a priori, será *jperex2021-1*. Pero no lo escribas como una cadena constante en el script. Obtenlo de la variable de entorno

6. La imagen del contenedor se llamará `ex2021`. Los contenedores basados en esta imagen (en este caso solo uno), tendrán como prefijo parte de tu nombre de usuario, y como sufijo, un número. Si tu nombre de usuario fuera `jperez`, el prefijo sería `jper`. El sufijo, `-1`. Por tanto el contenedor se llamaría `jperex2021-1`. El nombre de `host` será este mismo. Sustituye `jper` por los primeros 4 caracteres de tu nombre de usuario.
7. Siguiendo un convenio como el que hemos venido usando durante toda la asignatura, los nombres de los ficheros serían:

```
~/lagrs.jul.21/ex2021/construye.sh
~/lagrs.jul.21/ex2021/lanza_jperex2021-1.sh
~/lagrs.jul.21/ex2021/context/Dockerfile
~/lagrs.jul.21/ex2021/context/entrypoint.sh
~/lagrs.jul.21/ex2021/context/maquina
~/lagrs.jul.21/ex2021/context/delta_bashrc
```

(Reemplazando `jper` por las primeras 4 letras de tu login). Pero no escribas en esos ficheros, **escribe solamente en `final.txt`** indicando qué habría en cada fichero. P.e. *Contenido del fichero X: blabla. Contenido del fichero Y: blabla, etc.*

## Ejercicio 2 (4 puntos)

Escribe un programa en python llamado `~/lagrs.jul.21/conexiones.py` según la siguiente especificación:

1. El programa usará la librería `optparse` para aceptar un número de puerto, con la opción `-p`. Ejemplo: `conexiones.py -p 631`
2. Partiendo de la información proporcionada por la orden de shell `netstat -tupan`, el programa mostrará en la salida estándar todas las conexiones tcp v4 y todas las *pseudo-conexiones* udp v4 que usen dicho puerto en la dirección local. El programa ignorará las conexiones tcp v6 y las udp v6.
3. También indicará el número de conexiones totales que muestra.

En otras palabras, el programa invocará la orden `netstat -tupan` (con estas opciones y no otras), ignorará las líneas correspondientes a `tcp6` y `udp6`, mostrará todas aquellas líneas cuyo puerto de la columna *Dirección local* sea el indicado, e indicará cuántas líneas son.