

# Laboratorio de administración y gestión de redes y sistemas

## Examen práctico

20 de enero de 2022.

Grado en Ingeniería Telemática, Universidad Rey Juan Carlos

---

### Ejercicio 1 (6 puntos)

En este ejercicio prepararás 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. En el contenedor, habrá un script de shell llamado `/opt/examen/bin/maquina`
3. Este script `maquina` escribirá 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.

Naturalmente, *TU-HOST* será siempre el nombre de la máquina donde se ejecuta el script `maquina`. Cuando se ejecute en tu puesto del laboratorio, devolverá el nombre del puesto del laboratorio. Cuando se ejecute en el servidor de contenedores, devolverá el nombre del servidor de contenedores. Cuando se ejecute en un contenedor, devolverá el nombre de ese contenedor.

4. Dentro del contenedor, con la sesión de shell de *root* abierta, el usuario podrá ejecutar la orden `maquina`, *a secas*, sin necesidad de escribir el trayecto completo (`/opt/examen/bin/maquina`).

Como seguramente sabes, para poder conseguir esto, tendrás que añadir cierta línea al fichero `.bashrc` del usuario *root*.

Hazlo de forma similar a lo que hiciste en la práctica 1.23 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.
  6. La imagen del contenedor se llamará `ex2022`. 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 `jperex2022-1`. 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 serán:

```
~/lagrs.ene.22/ex2022/construye.sh
~/lagrs.ene.22/ex2022/lanza_jperex2022-1.sh
~/lagrs.ene.22/ex2022/context/Dockerfile
~/lagrs.ene.22/ex2022/context/entrypoint.sh
~/lagrs.ene.22/ex2022/context/maquina
~/lagrs.ene.22/ex2022/context/delta_bashrc
```

(Reemplazando *jper* por las primeras 4 letras de tu login).

Cuando acabes, enseña el resultado al profesor.

## Ejercicio 2 (4 puntos)

Escribe un programa en python llamado `~/lagrs.ene.22/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.

Cuando acabes, enseña el resultado al profesor.