

### T2.3 – Ejecución de servicios. Puertos y variables de entorno

Una de las cosas que **más interesantes de usar docker** no es ya el hecho de que puedo **probar todas versiones de los distintos sistemas** que van apareciendo, es el hecho de que **PARA PROBAR Y USAR CUALQUIER SERVICIO Y CUALQUIER APLICACIÓN NO TENGO QUE INSTALAR NADA EN MI SISTEMA**, sea cual sea el servicio o la aplicación que se me ocurra, siempre la tengo en Docker Hub. Solo tengo que buscarla, averiguar cuál es la versión que quiero y lanzar el contenedor o contenedores necesarios. Y si por casualidad no está lo que yo quiero siempre podré construirlo desde cero, cosa que veremos en el módulo 6.

No es ya que al evitar la instalación evite la sobrecarga que eso supone y la cantidad de ficheros y "basurilla" que van dejando este proceso, es que el uso de contenedores de este tipo impacta de manera muy favorable en el trabajo diario en al menos dos aspectos:

- Permite **utilizar servicios y aplicaciones** que a veces, dadas la **limitada potencia de los equipos de aula** que tenemos, es difícil que podamos usar con soltura.. Recordad que una de las características de los contenedores es que eran entornos de ejecución aislados (no dejan basura ninguna si los elimino) y que la sobrecarga sobre el sistema era mínima si lo comparáramos con una máquina virtual arrancando, consiguientemente, de una manera mucho más rápida.
- **Elimino** de una vez el problema que tantos quebraderos de cabeza nos da: **"EN MI EQUIPO NO FUNCIONA"**. En cualquier equipo donde se haya instalado docker de manera previa el contenedor que quiero que los demás usen va a funcionar con total seguridad.



*Juan Diego Pérez Jiménez. Funciona en mi ordenador (Dominio público)*

De momento en este apartado **nos vamos a centrar en servicios de un solo contenedor**, estamos hablando de **servidores de bases de datos, servidores web, servidores de aplicaciones etc...** servicios que por otra parte son de uso casi diario en nuestro trabajo. Veremos en módulos posteriores aplicaciones que requieren la interacción de más de un contenedor.

Para la ejecución de contenedores de este tipo vamos a tener en cuenta varias cosas:

- **flag -d**: Se usa para que el servicio se ejecute en **modo background o dettach**. Si no lo hacemos se bloqueará el terminal mostrando el log del servicio (en ciertas ocasiones puede interesarnos) y tendremos que salir del mismo con Ctrl+C. Esto detiene el contenedor aunque podremos arrancarlo posteriormente.

- **flag -p:** se usa si queremos que el servicio que vamos a lanzar sea accesible desde el exterior de la siguiente manera:

**-p PUERTO\_EN\_HOST:PUERTO\_EN\_CONTENEDOR**

que normalmente sería el puerto por defecto del servicio. Esto es una **REDIRECCIÓN DE PUERTOS**. Podemos tener varias reglas -p al arrancar (dependiendo del servicio será necesario) y es muy importante recordar que no podemos tener dos servicios escuchando en el mismo puerto. Si lo intentamos se nos mostrará un mensaje de error.

- **flag -e: Comprobar y definir** si es necesario las **variables de entorno** que puede tener el contenedor. Las variables de entorno importantes se describen **en la página de las imágenes en DockerHub** y para usarlas tenemos que escribir:

**-e NOMBRE\_VARIABLE=VALOR.**

Para ilustrar todo esto vamos a poner varios ejemplos:

```
# Ejecuto un servidor Apache sin el flag -d ni redirección de puertos. Se bloquea el terminal mostrando los logs y tendré que salir con Ctrl+C
```

```
> docker run httpd
```

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.22. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.22. Set the 'ServerName' directive globally to suppress this message
[Mon Dec 07 18:27:28.561909 2020] [mpm_event:notice] [pid 1:tid 140253864719488] AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations
[Mon Dec 07 18:27:28.562072 2020] [core:notice] [pid 1:tid 140253864719488] AH00094: Command line: 'httpd -D FOREGROUND'
```

```
# Ejecuto un servidor Apache en background y accediendo desde el exterior a través del puerto 8888 de mi máquina.
```

```
> docker run -d -p 8888:80 httpd
```

```
# Creación de un servidor de base de datos mariadb accediendo desde el exterior a través del puerto 3306 y estableciendo una contraseña de root mediante una variable de
```

```
# entorno
```

```
> docker run -it -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root mariadb
```