

## LABORATORIO 6

Con esta práctica se observa que los contenedores son unidades ejecutables de software en las cuales el código de una aplicación, sus librerías y dependencias se empaquetan para que se pueda ejecutar en cualquier lugar. Esto les genera una mayor movilidad, es decir, las aplicaciones que se ejecutan en contenedores se pueden poner en marcha en plataformas de hardware y sistemas operativos diferentes, ya que estos llevan consigo todas sus dependencias. Estos “empaquetan” los recursos necesarios para el funcionamiento de una aplicación, para facilitar su uso a administradores y desarrolladores, simplificando tanto la instalación como la puesta en marcha de programas muy complejos. Por tanto, Docker es una herramienta para **iniciar, detener y gestionar contenedores** mediante una serie de comandos. Una característica destacable de este contenedor es el **Docker Hub**, un repositorio donde los usuarios de Docker pueden compartir las imágenes (contenedores) que han creado con otros usuarios. En esta práctica también se aprende a manejar los comandos necesarios para el uso de contenedores, con su previa instalación de Docker, desde un sistema local y a crear propios contenedores partiendo de una imagen.

Una imagen Docker es un archivo, compuesto por múltiples “capas” (conjuntos de cambios), que se utiliza para ejecutar código en un contenedor Docker. Estas imágenes son las plantillas base desde las que partimos ya sea para crear una nueva imagen o crear nuevos contenedores para ejecutar las aplicaciones. Las imágenes se utilizan para crear contenedores debido a que éstas nunca van a cambiar. Esto permite crear contenedores con diferentes “capas” de imágenes, las cuales se van a superponer sobre otras. En este laboratorio también se puede observar que Docker, para poder optimizar el espacio, lo que hace es que si muchos cambios realizados están basados en imágenes que ya tienes descargadas y son similares, no volverá a descargar esa imagen, optimizando el espacio en el disco.

Por tanto, algunos de los beneficios del uso de contenedores que se ha visto en esta práctica son:

- **Instalación más sencilla:** los contenedores de software se inician a partir de imágenes o representaciones portables de un contenedor, incluyendo un programa y todos los componentes requeridos (como librerías, soporte o archivos de configuración). De esta forma se compensan las diferencias entre sistemas operativos.
- **Independiente de la plataforma:** las imágenes se pueden transportar cómodamente de un sistema a otro y se caracterizan por una considerable independencia de la plataforma.
- **Aplicaciones aisladas:** cada programa funciona independientemente de otros contenedores, de forma que aplicaciones con requerimientos opuestos pueden funcionar en paralelo en el mismo sistema.
- **Reproducibilidad de los experimentos.**
- **La facilidad de colaboración entre personas del mismo o diferente grupo.**