

CONCEPTOS FUNDAMENTALES DE DOCKER Y BASES DE DATOS

ASIGNATURA LINEA DE INVESTIGACIÓN III

JUAN MANUEL NAVARRO CORDOBA

561220150

**UNIVERSIDAD DE CUNDINAMARCA EXTENSIÓN CHIA PROGRAMA DE
INGENIERÍA DE SISTEMAS FACULTAD DE INGENIERÍA**

2025

Contenido	
Introducción	3
Objetivo	3
Conceptos fundamentales de Docker	4
¿Qué es una imagen en Docker y cómo se relaciona con un contenedor?	4
¿Qué es Docker Hub y cómo se utiliza para descargar imágenes?	4
¿Qué son los volúmenes en Docker?	4
¿Qué es Docker Compose y qué ventajas tiene frente a ejecutar un contenedor con docker run?	5
Repaso de Bases de Datos	5
Entidades	5
Atributos	5
Relaciones	5
Cardinalidad	5
Conclusión	6
Referencias	6

Introducción

Docker facilita la creación de entornos ligeros y portables a través de contenedores que incluyen aplicaciones, dependencias y configuraciones necesarias, mientras que las bases de datos relacionales permiten organizar, almacenar y consultar información de manera estructurada y consistente. Comprender conceptos como imágenes, volúmenes, redes y Docker Compose, junto con entidades, atributos, relaciones, cardinalidad y formas normales, resulta esencial para implementar soluciones eficientes y escalables en entornos académicos y profesionales.

Objetivo

Fortalecer la comprensión de los conceptos fundamentales de Docker y de las bases de datos relacionales, analizando sus componentes, diferencias con otros entornos y la aplicación de principios de modelado de datos.

Conceptos fundamentales de Docker

¿Qué es Docker y en qué se diferencia de una máquina virtual?

- Docker es una plataforma que permite crear, desplegar y ejecutar aplicaciones dentro de contenedores.
- Un contenedor es una unidad ligera y portable que incluye el código de la aplicación, sus librerías y dependencias necesarias, garantizando que se ejecute de forma idéntica en cualquier entorno.
- **Diferencias con una máquina virtual (VM):**
 - Una VM incluye un sistema operativo completo sobre un hipervisor, lo cual consume más recursos.
 - Docker comparte el kernel del sistema operativo del host, por lo que es más ligero, rápido y eficiente.

¿Qué es una imagen en Docker y cómo se relaciona con un contenedor?

- Una imagen en Docker es una plantilla inmutable que contiene todo lo necesario para ejecutar una aplicación: sistema base, dependencias, configuraciones y binarios.
- Un contenedor es una instancia en ejecución de una imagen.
Ejemplo: una imagen de MySQL sirve como plantilla, y al ejecutar un contenedor basado en ella obtienes tu base de datos funcionando.

¿Qué es Docker Hub y cómo se utiliza para descargar imágenes?

- Docker Hub es el repositorio oficial de imágenes Docker.
- Permite buscar, descargar y publicar imágenes.
- Se utiliza con comandos como:

```
docker pull mysql:8.0
docker pull nginx
```

¿Qué son los volúmenes en Docker?

- Los volúmenes son mecanismos de almacenamiento persistente para los datos de los contenedores.
- Permiten que la información no se pierda al eliminar o recrear un contenedor.
- Ejemplo:
`docker run -v mysql-data:/var/lib/mysql mysql:8.0`

¿Qué es una red en Docker?

- Una red en Docker es un canal de comunicación que conecta contenedores entre sí y con el host.

- Permite que contenedores de diferentes servicios (por ejemplo, una aplicación web y una base de datos) se comuniquen de forma segura.
- Ejemplo:


```
docker network create mi_red
docker run --network mi_red ...
```

¿Qué es Docker Compose y qué ventajas tiene frente a ejecutar un contenedor con docker run?

- Docker Compose es una herramienta que permite definir y ejecutar múltiples contenedores usando un archivo docker-compose.yml.
- Ventajas frente a ejecutar manualmente con docker run:
 - Gestiona varios servicios al mismo tiempo (ej. base de datos, backend y frontend).
 - Simplifica la configuración al centralizarla en un archivo YAML.
 - Permite escalar y levantar entornos completos con un solo comando:


```
docker-compose up -d
```

Repaso de Bases de Datos

Entidades

- Son objetos del mundo real o conceptos sobre los que se almacena información.
- Ejemplo: Cliente, Pedido, Comercial.

Atributos

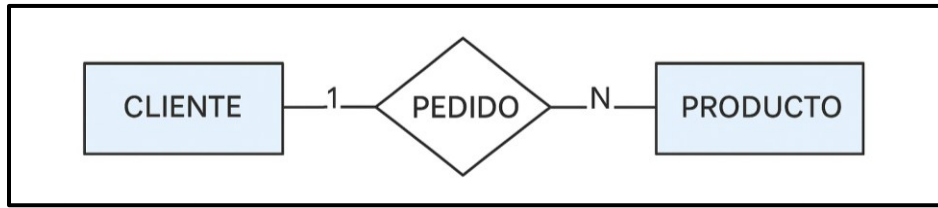
- Son las propiedades o características de una entidad.
- Ejemplo: Cliente → nombre, apellido, ciudad, categoría.

Relaciones

- Representan los vínculos entre entidades.
- Ejemplo: un cliente realiza pedidos; un comercial gestiona pedidos.

Cardinalidad

- Indica la cantidad mínima y máxima de ocurrencias que una entidad puede tener en una relación.
 - 1:1 → un cliente tiene un único documento de identidad.
 - 1:N → un cliente puede hacer muchos pedidos.
 - N:M → un estudiante puede inscribirse en muchos cursos y un curso puede tener muchos estudiantes.



Formas normales de una base de datos

Las formas normales son reglas para organizar los datos y evitar redundancia:

1. Primera Forma Normal (1FN): todos los atributos deben ser atómicos (sin listas o valores repetidos en una celda).
2. Segunda Forma Normal (2FN): cumplir 1FN + todos los atributos dependen de la clave primaria (no de una parte de ella).
3. Tercera Forma Normal (3FN): cumplir 2FN + no debe haber dependencias transitivas (atributos que dependan de otros atributos no clave).

Conclusión

El estudio permitió identificar cómo la virtualización mediante contenedores agiliza el despliegue de aplicaciones frente al uso tradicional de máquinas virtuales, destacando la relevancia de imágenes, volúmenes, redes y Docker Compose en la gestión de entornos portables. A su vez, el repaso de los fundamentos de bases de datos evidenció la importancia del correcto diseño de entidades, atributos y relaciones, así como la aplicación de cardinalidad y formas normales para garantizar integridad, evitar redundancia y optimizar el acceso a la información. En conjunto, ambos temas refuerzan la necesidad de integrar tecnologías de contenedores con principios de modelado de datos en proyectos reales.

Referencias

Documentación Docker Disponible en: <https://docs.docker.com/>

MySQL 8.0 Reference Manual. Disponible en: <https://dev.mysql.com/doc/>