

DEPARTAMENTO DE TECNOLOGIAS Y SERVICIOS INFORMATICOS  
COORDINACION DE OPERACIONES  
ADMINISTRACION DE PLATAFORMAS DE CÓMPUTO  
GUÍA DE ESTUDIO PARA EL PUESTO DE ANALISTA DE BASES DE DATOS Y APLICACIONES ACADEMICAS  
ADMINISTRATIVAS



GUÍA DE ESTUDIO PARA EL PUESTO DE ANALISTA DE BASES DE DATOS Y APLICACIONES  
ACADEMICAS ADMINISTRATIVAS



## TEMAS

### 1. Bases de datos

- 1.1 Definición
- 1.2 Sistema de gestión de bases de datos (SGBD)
- 1.3 Tipos de bases de datos
- 1.4 Tipos de motores de bases de datos
- 1.5 Tipos de datos
- 1.6 Lenguaje SQL
- 1.7 Tareas de Mantenimiento

### 2. Open Source

- 2.1 Sistema Operativo GNU/Linux
- 2.2 Software de Aplicación

### 3. Virtualización de servidores

- 3.1 Virtualización de equipo servidor
- 3.2 Características y beneficios de la virtualización de servidores
- 3.3 Alta disponibilidad en servidores
- 3.4 Virtualización de interfaces de comunicación en servidores
- 3.5 Uso de fibra canal en servidores

### 1. Bases de datos

#### 1.1 Definición

Es una colección de datos persistentes que están lógicamente relacionados entre sí mediante el diseño de una estructura según un modelo que refleja las relaciones y restricciones que tienen estos datos en el mundo real. En la siguiente tabla se muestran los objetos más comunes de los que se componen bases de datos tanto en Microsoft SQL Server como en MySQL.

Objeto	Función
Tablas	Es el objeto que contiene la información a archivar organizada por campos o columnas de distintos tipos.
Campo	Es la unidad básica de entrada de datos con un tipo definido en función de la información contenido en la tabla.

DEPARTAMENTO DE TECNOLOGIAS Y SERVICIOS INFORMATICOS  
COORDINACION DE OPERACIONES  
ADMINISTRACION DE PLATAFORMAS DE CÓMPUTO  
GUÍA DE ESTUDIO PARA EL PUESTO DE ANALISTA DE BASES DE DATOS Y APLICACIONES ACADEMICAS  
ADMINISTRATIVAS

Índices	Es una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla.
Vistas	Es una consulta que se presenta como una tabla (virtual) a partir de un conjunto de tablas en una base de datos. Las vistas tienen la misma estructura que una tabla pero sólo se almacena la definición y no los datos.
Stored Procedure	Es un programa escrito en lenguaje SQL que al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos y envía el resultado al usuario.
Usuarios	Son usuarios que tienen un nivel de acceso a la base de datos en base a un rol o a permisos definidos sobre los objetos de la base de datos.

## 1.2 Sistema de gestión de base de datos (SGBD)

Los Sistemas de Gestión de Bases de datos (en inglés DataBase Management System) son un tipo de software muy específico de fabricante, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Un ejemplo de una herramienta para administrar bases de datos con interfaz gráfica es Microsoft SQL Server Management Studio de la compañía Microsoft y para administrar bases de datos de software libre como MySQL se realiza a través de una conexión remota vía SSH utilizando una herramienta como Putty.

## 1.3 Tipos de Bases de Datos

### Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible.

### Bases de datos relacionales

Este modelo se basa en el principio de la organización de la información en trozos pequeños, que se relacionan entre ellos mediante la relación de identificadores. La información puede ser recuperada o almacenada mediante consultas a través del lenguaje SQL que ofrecen una amplia flexibilidad y poder para administrar la información.

### Bases de datos No Relacionales (NoSQL)

Es un modelo de base de datos diferente que no se basa en relaciones de las entidades sino en el uso de formatos específicos para el almacenamiento de la información. No usan el lenguaje SQL como lenguaje principal de consultas. Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente la atomicidad, consistencia, aislamiento y durabilidad dado que habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan "no sólo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL.

### **Bases de datos multidimensionales**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de cubos OLAP. En las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean aprender.

## **1.4 Tipos de motores de base de datos**

**Entre los diferentes tipos de motores de bases de datos, podemos encontrar los siguientes:**

- **MySQL:** Es un motor de bases de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez, aunque no se recomienda su uso para grandes volúmenes de datos.
- **Postgresql – Oracle:** Son sistemas de base de datos poderosos, administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets.
- **Microsoft SQL Server:** Es una base de datos más potente que Microsoft Access desarrollada por Microsoft y es empleada para manejar grandes volúmenes de información.
- **NoSQL:** Son motores que utilizan diferentes estructuras para la organización y almacenamiento de bases de datos. Algunos motores usados son MondoDB, Cassandra, CouchDB, Elastic, DynamoDB.

## **1.5 Tipos de datos**

Un tipo de datos define qué valor puede contener cada columna de una tabla y al crear una se requiere asignarle a cada columna un nombre y un tipo de datos. A continuación, se presentan los más comunes tanto para Microsoft SQL Server como para MySQL:



### Tipo de datos Texto

char (n)	Cadena de caracteres de ancho fijo	8,000 caracteres
varchar (n)	Cadena de caracteres de ancho variable	8,000 caracteres
text	Cadena de caracteres de ancho variable	2 GB de datos de texto
binary (n)	Cadena binaria de ancho fijo	8,000 bytes
varbinary	Cadena binaria de ancho variable	8,000 bytes

### Tipos de datos numéricos

bit	Entero que puede ser 0, 1 o NULL	
tinyint	Permite números enteros de 0 a 255	1 byte
smallint	Permite números enteros entre -32,768 y 32,767	2 bytes
int	Permite números enteros entre -2,147,483,648 y 2,147,483,647	4 bytes
decimal (p, s)	Números de escala y precisión fijos. Permite números de $-10^{38} + 1$ a $10^{38} - 1$ .	5-17 bytes
float (n)	Datos del número de precisión flotante desde $-1.79E + 308$ a $1.79E + 308$ .	4 u 8 bytes
real	Datos numéricos de precisión flotante desde $-3.40E + 38$ a $3.40E + 38$	4 bytes



### Tipos de datos de fecha

datetime	Desde el 1 de enero de 0001 hasta el 31 de diciembre de 1999	6-8 bytes
smalldatetime	Del 1 de enero de 1900 al 6 de junio de 2079	4 bytes
date	Almacenar una fecha solamente. Del 1 de enero de 0001 al 31 de diciembre de 9999	3 bytes
time	Almacenar un tiempo solo con una precisión de 100 nanosegundos	3-5 bytes

### 1.6 Lenguaje SQL

El lenguaje SQL (Structured Query Language - Lenguaje de consulta estructurada) es utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

#### Sentencias DDL (Data Definition Language – Lenguaje de Definición de Datos)

El lenguaje de definición de datos es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir tablas en las que se almacenan los datos de la base de datos. A continuación, se presentan ejemplos de sintaxis de sentencias para operaciones básicas utilizadas comúnmente en Microsoft SQL Server y en MySQL:

Ejemplo de sentencia de Microsoft SQL Server y MySQL para crear una base de datos:

**CREATE DATABASE NOMBREBD;**

Ejemplo de sentencia de *Microsoft SQL Server* y *MSQL* para definir la base de datos de trabajo:

**USE NOMBREBD;**

Ejemplo de sentencia de *Microsoft SQL Server* y *MSQL* para crear una tabla con tres campos:

**CREATE TABLE *NOMBRETABLA* (**

**CAMPO1 int,  
CAMPO2 varchar(30),  
CAMPO3 time);**

Ejemplo de sentencia de *Microsoft SQL Server* y *MySQL* para agregar un campo a una tabla:

**ALTER TABLE ADD *NOMBRETABLA CAMPO4 time;***

Ejemplo de sentencia de *Microsoft SQL Server* y *MySQL* para eliminar una tabla:

**DROP TABLE *NOMBRETABLA;***

Ejemplo de sentencia de *Microsoft SQL Server* y *MySQL* para eliminar una base de datos:

**DROP DATABASE *NOMBREBD;***

### **Sentencias DML (Data Manipulation Language – Lenguaje de Manipulación de Datos)**

Un lenguaje de manipulación de datos es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado. A continuación, se presentan ejemplos de sintaxis de sentencias para cuatro operaciones básicas utilizadas comúnmente en *Microsoft SQL Server* y en *MySQL*:

Ejemplo de sentencia de *Microsoft SQL Server* y *MySQL* para consultar todos registros de una tabla:

**SELECT \* FROM *NOMBRETABLA;***

Ejemplo de sentencia de *Microsoft SQL Server* y MySQL para actualizar/modificar un campo de una tabla de acuerdo a una condición:

**UPDATE NOMBRETABLA SET CAMPO1='VALOR' WHERE CAMPO3='13:30';**

Ejemplo de sentencia de *Microsoft SQL Server* y MySQL para eliminar registros de una tabla de acuerdo a una condición:

**DELETE FROM NOMBRETABLA WHERE CAMPO2='13:30';**

Ejemplo de sentencias de *Microsoft SQL Server* y MySQL para agregar un registro a una tabla con dos campos:

**INSERT INTO NOMBRETABLA COLUMNA VALUES ("VALOR1", "VALOR2");**

## 1.7 Tareas de Mantenimiento

### Actualización de Estadísticas

Las estadísticas son objetos que mantienen información sobre el contenido de las columnas de las tablas y cuya utilidad consiste en llevar control sobre la selectividad, la cardinalidad y la densidad de los datos. Mantener esos datos en constante actualización impacta de forma positiva en la velocidad en la que la información puede ser consultada. A continuación, se presentan ejemplos de sintaxis de sentencias para actualizar estadísticas de una tabla para bases de datos en Microsoft SQL Server y MySQL:

Sentencia de *Microsoft SQL Server* para actualizar estadísticas de una tabla:

**UPDATE STATISTICS NOMBRETABLA WITH FULL SCAN;**

Sentencia de *MySQL* para actualizar estadísticas de una tabla:

**ANALYZE TABLE NOMBRETABLA;**

### Reconstrucción de Índices



Las bases de datos mantienen índices automáticamente cada vez que inserta, actualiza o elimina operaciones realizadas en los datos subyacentes. Con el tiempo, estas modificaciones pueden hacer que la información del índice se disperse por la base de datos (se fragmente). La fragmentación ocurre cuando los índices tienen páginas en las que la ordenación lógica, basada en el valor de clave, no coincide con la ordenación física dentro del archivo de datos. Los índices muy fragmentados pueden reducir el rendimiento de la consulta y ralentizar la respuesta de la aplicación. A continuación, se presentan ejemplos de sintaxis de sentencias para reconstruir índices de una tabla para bases de datos en Microsoft SQL Server y MySQL:

Sentencia de *Microsoft SQL Server* para reconstruir índices de una tabla:

**ALTER INDEX NOMBREINDICE ON NOMBRETABLA REBUILD**

Sentencia de *MySQL* para reconstruir índices de una tabla:

**ANALIZE TABLE NOMBRETABLA;**

### Respaldos y restauración de bases de datos

La copia de seguridad, también llamada respaldo o backup, se refiere a la copia de archivos físicos o virtuales o bases de datos a un sitio secundario para su preservación en caso de falla del equipo u otra catástrofe. En la siguiente tabla se presentan los tipos de respaldos para bases de datos.

Tipo	Descripción
Respaldos completos	Es el tipo de respaldo más básico completo porque consiste en la copia de la totalidad de los datos en otro medio o ubicación para garantizar la continuidad de la operación e integridad de la información.
Respaldos incrementales	El respaldo incremental sólo copia los datos que se han incrementado desde la última operación de respaldo de cualquier tipo. La ventaja de un respaldo incremental es que copia una menor cantidad de datos que un respaldo completo por lo que requiere menos tiempo para generarse y consume menos espacio de almacenamiento.
Respaldos diferenciales	Una operación de respaldo diferencial copia todos los datos que hayan cambiado desde el respaldo anterior. Sin embargo, cada vez que se vuelva

	a ejecutar, seguirá copiando todos los datos que hayan cambiado desde el anterior completo.
--	---

En Microsoft SQL Server los respaldos y restauración de bases de datos se realizan generalmente utilizando asistentes que incorpora la herramienta Microsoft Management Studio. En MySQL se utiliza más comúnmente línea de comandos. A continuación, se muestran ejemplos de sintaxis de sentencias para para realizar un respaldo y restauración de una base de datos tanto para Microsoft SQL Server como para MySQL:

Sentencia para realizar respaldo de una base de datos en *Microsoft SQL server*:

```
SqlCmd -E -S NOMBRESEVIDOR -Q "BACKUP DATABASE NOMBREBD TO DISK='D:\NOMBREARCHIVO.BAK'"
```

Ejemplo de sentencia para realizar respaldo de una base de datos en *MySQL*, del cual se especifica el usuario, nombre de la base de datos a respaldar y el nombre del archivo donde se almacenará el respaldo.

```
mysqldump --single-transaction -u USUARIOBD -p NOMBREBD > NOMBREARCHIVO.sql;
```

Sentencia para realizar restauración de una base de datos en *Microsoft SQL server*:

```
SqlCmd -E -S NOMBRESEVIDOR -Q "RESTORE DATABASE NOMBREBD FROM DISK='D:\NOMBREARCHIVO.BAK'"
```

Sentencia para realizar restauración de una base de datos en *MySQL*:

```
mysql -u USUARIOBD -p NOMBREBD < NOMBREARCHIVO.sql;
```

Para realizar la conexión a MySQL se debe realizar vía consola SSH utilizando el comando: **mysql -u USUARIOBD -p**

### Características para Microsoft SQL Server

GUÍA DE ESTUDIO PARA EL PUESTO DE ANALISTA DE BASES DE DATOS Y APLICACIONES ACADEMICAS ADMINISTRATIVAS

Para realizar la instalación de Microsoft SQL Server se deben seleccionar las siguientes características que permitan crear y administrar bases de datos:

- Servicios de Motor de Base de datos (*Database Engine Services*)
- Herramientas de cliente con compatibilidad con versiones anteriores (*Client Tools Backwards Compatibility*).

Las características citadas anteriormente se presentan en versiones de SQL Server 2012 y posteriores.

### **Intercalación de una base de datos**

La intercalación en una base de datos (Collate) permite realizar una operación de conversión a una expresión de cadena de caracteres que permita procesar diferentes fuentes de información con cadenas de datos distintas. Dicha intercalación puede aplicar a nivel de base de datos, tabla o de campo. A continuación, se muestran ejemplos de sintaxis de sentencias para asignar un collate al momento de crear una base de datos en *Microsoft SQL server*:

#### **CREATE DATABASE BASEDATOS**

***COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS;***

## **2. Opensource**

OpenSource se refiere a la Libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el Software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

### **2.1 Sistema Operativo GNU/Linux**

Linux es un sistema operativo de software libre multitarea, multiusuario, compatible con UNIX, y proporciona una interfaz de comandos y una interfaz gráfica, que lo convierte en un sistema muy atractivo y estable. Dado que Linux se distribuye de forma libre existen muchísimas versiones distintas de un mismo sistema operativo que se llaman comúnmente

Distribuciones. Una distribución incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios. Las distribuciones más utilizadas son Ubuntu (y sus derivados), Debian, Fedora, RedHat, Arch Linux y Gentoo.

### **Núcleo**

El kernel de un sistema es el componente central que controla las acciones del sistema operativo y su interacción con el hardware. Es la capa responsable de asegurar que todos los programas y procesos tengan acceso a los recursos que necesitan de la máquina (memoria RAM, acceso a disco y el control de la CPU) al mismo tiempo, de modo que haya un recurso compartido de estos. En otras palabras, es el cerebro del sistema operativo; el responsable de coordinar el acceso al hardware y los datos entre los diferentes componentes del sistema. Además, es el encargado de gestionar los recursos a través de servicios de llamada al sistema.

### **Gestor de Arranque**

Un gestor de arranque, también llamado boot o bootloader, es el programa cuya tarea es cargar el sistema operativo de un ordenador en la memoria. Cuando un equipo se conecta, la BIOS o UEFI del equipo realiza algunas pruebas iniciales de las actividades básicas para el correcto funcionamiento de todos los recursos de la máquina; y entonces transfiere el control al Registro de Arranque Maestro (MBR – Master Boot Record), donde el bootloader se encuentra. En sistemas Linux, por lo general, el gestor de arranque que se utiliza es GRUB.

### **Demonios y servicios**

Los demonios (daemons) o servicios son programas que se ejecutan en segundo plano del sistema operativo para ejecutar tareas específicas que son necesarias para la ejecución de procesos como detectar dispositivos USB, permitir iniciar sesión, administrar el sistema de archivo, etc. Muchas de las distribuciones de Linux existentes utilizan el nuevo controlador de servicios llamado systemd para el control total del sistema operativo, aunque aún persiste el uso del controlador anterior llamado sysvinit en algunas de las distribuciones de Linux.

## **Sistema de Archivos**

El sistema de archivos controla como la información es almacenada y recuperada. La información es separada en piezas llamadas archivos para su identificación y facilitar su uso por el usuario final. Linux soporta el uso de distintos sistemas de archivos, aunque el uso del tipo ext4 es de uso generalizado en la mayoría de las implementaciones. Otros ejemplos de tipos de archivos son: ext2, ext3, jfs, ReiserFS, XFS, BtrFS.

En general la estructura de directorios en Linux generalmente posee los siguientes elementos:

- /bin Archivos binarios ejecutables para distintas tareas del sistema operativo.
- /dev Archivos especiales que representan dispositivos de Hardware como USB.
- /etc Archivos de configuración de los servicios y software del sistema operativo.
- /home Directorio privado para la información de los usuarios.
- /mnt Punto de montaje para dispositivos externos como CD o USB.
- /tmp Directorio para almacenar archivos temporales.
- /usr Archivos de configuración de software de aplicación.
- /var Información generada por el Software instalado como logs o informe de errores.

## **Gestor de paquetes**

Son una colección de herramientas de software que automatizan el proceso de instalar, actualizar, configurar o eliminar software del sistema operativo de una forma consistente y segura. Los gestores de paquetes están intrínsecamente ligados a las distribuciones de Linux para las que han sido diseñados. Algunos ejemplos de ellos son: aptitude, apt, pacman, yum, yaourt, zipper.

## **Administrador de entorno gráfico**

El modo gráfico en Linux es generado por el servidor gráfico X (X Window System), que no es parte del núcleo de Linux. Entre otras funciones, es responsable de la activación de la tarjeta de vídeo, ratón y teclado, lo que permite al usuario el uso de interfaces gráficas que son llamadas de Gestores de Ventanas y Entornos de Escritorio. Los Entornos de Escritorio



disponen de interfaz completa para el usuario, así como: barra de herramientas, botones, menús, iconos y bibliotecas gráficas. El administrador de entornos gráficos más usado es X.org aunque ya existen otras implementaciones como Unity, Wayland y Mir.

### Entornos de Escritorio

Son colecciones de aplicaciones y librerías diseñadas para presentar un interfaz de usuario consistente para las tareas más comunes en forma gráfica. Son altamente parametrizables para satisfacer las necesidades del usuario y pueden existir múltiples entornos en una misma instalación de Linux. Existe una gran variedad de entornos de escritorio que varían en popularidad con el usuario final debido a criterios como practicidad, consumo de recursos y nivel de parametrización, entre los más conocidos están: GNOME, KDE, XFCE, LXDE, Plasma, Pantheon, Budgie, i3, Lumina, Cinnamon, Mate, etc.

### Interfaz de línea de comandos

Es un módulo, comúnmente llamado Shell o CLI (Command Line Interface – Interfaz de Línea de comando), que actúa como capa externa entre el usuario y el sistema operativo que permite a los usuarios ejecutar comandos que cambian la configuración del sistema o modifican información almacenada en él. Actualmente, la mayoría de los sistemas Linux usan, por defecto, una evolución de Bourne Shell llamado Bash (Bourne Again Shell).

### Comandos básicos de línea de comando

Los siguientes son ejemplos de comandos básicos de Linux.

pwd	Muestra el directorio actual.	# pwd
ls	Lista el contenido del directorio actual.	# ls -la
cat	Muestra el contenido de un archivo de texto	# cat /etc/apache/ports.conf
cp	Copia archivos de una ubicación a otra.	# cp archivo.doc /usr/archivo.doc
mv	Renombra archivos.	# mv archivo.doc archivo2.doc

DEPARTAMENTO DE TECNOLOGIAS Y SERVICIOS INFORMATICOS  
 COORDINACION DE OPERACIONES  
 ADMINISTRACION DE PLATAFORMAS DE CÓMPUTO  
 GUÍA DE ESTUDIO PARA EL PUESTO DE ANALISTA DE BASES DE DATOS Y APLICACIONES ACADEMICAS  
 ADMINISTRATIVAS

mkdir	Crear un directorio nuevo.	# mkdir documentos
rmdir	Elimina un directorio.	# rmdir -R documentos
sudo	Forza la ejecución de un comando en modo de supervisor.	# sudo rm -R documentos
head	Despliega las primeras líneas de un archivo de texto.	# head /var/logs/syslog
tail	Despliega las líneas finales de un archivo de texto.	# tail /var/logs/syslog
chmod	Cambia los atributos de un archivo	# chmod +x script.sh
service	Inicia o detiene servicios o demonios	# sudo service apache2 stop
locate	Localiza archivos y directorios.	# locate -i *firefox*
df	Muestra información de consumo de disco duro.	# df -h
free	Muestra el consumo de memoria RAM.	# free -m
uname	Muestra información referente a la versión de Linux en uso.	# uname -a
kill	Detiene un proceso en ejecución	# kill 12345
ps	Muestra información de los procesos en ejecución	# ps aux
passwd	Permite cambiar la contraseña del usuario	# passwd usuario
hostname	Muestra o cambia el nombre del equipo	# hostname
ssh	Permite establecer una conexión a un servidor remoto	# ssh 192.168.0.1 -l usuario
chown	Permite cambiar el propietario de un archivo o directorio	# chown -R root:root Directorio
grep	Busca una cadena de texto	# grep Linux

top	Muestra los procesos en ejecución	# top
crontab	Permite crear tareas programadas	# crontab -e
apt-get	Permite instalación de aplicaciones.	apt-get install NombreAplicacion

## 2.2 Software de Aplicación

Un servidor de aplicaciones gestiona la mayor parte de las funciones de lógica de negociación y acceso a los datos de las aplicaciones. Los principales beneficios son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones, mejora la alta disponibilidad, la escalabilidad y el mantenimiento de aplicaciones. Algunos ejemplos de servidores de aplicación son: WebLogic Application Server, WebSphere Application Server, JBoss Enterprise Application Platform, Tomcat.

### Servidores WEB

Son programas que implementan el protocolo HTTP y HTTPS para servir los archivos que conforman los sitios web a los usuarios en respuesta a sus peticiones a través de clientes http como navegadores. Los servidores HTTP suelen almacenar por lo general páginas web, que incluyen archivos HTML, PHP, ASP, etc, así como imágenes, vídeos y audio. Algunos ejemplos de servidores de aplicación son: Apache, nginx, lighttpd, caddy, IIS, Litespeed, Cherokee, NodeJS.

### Servidores de Bases de Datos

Es un sistema que realiza en un modelo cliente/servidor tareas de análisis de datos, almacenamiento, manipulación y explotación de la información contenida en las bases de datos por medio de una interfaz gráfica o con el uso del lenguaje SQL. Ejemplos de servidores de bases de datos disponibles para Linux son: PostgreSQL, MySQL. Oracle 12g, DB2, Informix y recientemente Microsoft SQL Sever sobre Ubuntu.

## 3. Virtualización de servidores

### 3.1 Virtualización de equipo servidor

La virtualización de un servidor es la capacidad de abstraer los recursos de un servidor físico y luego presentar los recursos de hardware a cada servidor virtual que se ejecuta en el host físico. A los servidores físicos se conocen como hosts, y los servidores virtuales son los invitados del host y se puede usar la virtualización de servidores para crear y ejecutar múltiples servidores virtuales en un solo host físico. Aunque todos los servidores virtuales se ejecutan en un servidor físico, cada una de ellos puede ejecutar su propio sistema operativo.

El hipervisor es una capa de software que es la tecnología subyacente que existe por encima del hardware y por debajo de uno o más sistemas operativos. Crea particiones para proporcionar entornos de ejecución aislados y administra recursos físicos compartidos. En la virtualización del servidor x86, los hipervisores se pueden clasificar en dos tipos:

Tipo 1: Es un sistema operativo que se instala directamente en el hardware que se está virtualizando y no depende de un sistema operativo host, por ejemplos Windows Server Hyper-V y Windows Client Hyper-V.

Tipo 2: Es un hipervisor que requiere y depende del sistema operativo del host por ejemplo, Windows Virtual PC, Windows Virtual Server y Oracle VirtualBox.

Para realizar la instalación del sistema operativo o de una aplicación en un servidor virtual, se debe definir en el hipervisor si se utiliza la unidad de DVD del servidor físico o se inserta un archivo ISO en el menú *Dispositivos* del servidor virtual en Virtual Box y/o *Medios* en el caso de Hyper-V.

### **3.2 Características y beneficios de la virtualización de servidores**

Además de ejecutar múltiples sistemas operativos en un único servidor físico, la virtualización de servidores proporciona muchas características y beneficios que se pueden utilizar para crear y mantener soluciones altamente escalables, resistentes y eficientes para las cargas de trabajo de sus servidores virtuales. Algunas de las características y beneficios que brinda la virtualización son:

**Memoria dinámica:** La memoria dinámica permite administrar eficientemente el total de la memoria del servidor físico de acuerdo a la cantidad de memoria utilizada por los servidores virtuales.

**Procesador:** Es posible asignar y modificar la cantidad de procesadores lógicos (cores) a todos los servidores virtuales contenidos en un host de acuerdo a su carga de trabajo.

**Almacenamiento:** Para cada servidor virtual contenido en el host se pueden crear y asignar varios discos duros virtuales, los cuales pueden redimensionarse individualmente para modificar su capacidad.

**Consumo de energía:** Mediante virtualización es posible convertir servidores físicos a virtuales y consolidarlos en un servidor físico de alto desempeño lo que brinda un ahorro en el consumo de energía eléctrica.

### 3.3 Alta disponibilidad en servidores

El objetivo de la alta disponibilidad es hacer que los sistemas y servicios estén disponibles, y eliminar cada posible punto único de falla. La disponibilidad a menudo se expresa numéricamente, como el porcentaje de tiempo que un servicio está disponible. Por ejemplo, un requisito de 99.9 por ciento de disponibilidad permite 8.75 horas de tiempo de inactividad por año, o aproximadamente 40 minutos de tiempo de inactividad cada cuatro semanas. Sin embargo, con un 99,999% de tiempo de actividad, el tiempo de inactividad del servicio permitido se reduce a solo cinco minutos por año.

### 3.4 Virtualización de interfaces de comunicación en servidores

Para algunas cargas de trabajo, una sola tarjeta de interfaz de red (NIC) puede ser suficiente; sin embargo, los servidores físicos generalmente cuentan múltiples tarjetas de red (NIC). Aunque un host puede ser completamente funcional con una sola NIC, esto generalmente no se recomienda. Una sola NIC no proporciona redundancia, y si la NIC falla, el host de y todas las máquinas virtuales que se ejecutan en ese host perderán la conectividad de la red. Es posible que un host de deba tener más NIC por varios motivos, incluido un mayor ancho de banda para varios servidores virtuales que comparten el mismo canal, mejor rendimiento, administración y redundancia.

### 3.5 Uso de Fibra Canal en servidores

Para conectar un almacenamiento compartido (SAN) a un servidor virtual, el hipervisor proporciona puertos de fibra canal dentro del sistema operativo invitado, lo que le permite conectarse directamente por fibra canal como un servidor físico. Con la función de fibra canal del hipervisor es posible conectarse al almacenamiento de fibra canal desde un servidor virtual permitiendo almacenar discos duros virtuales y proveer funciones como SAN virtuales, migración en vivo, MPIO, así como también operar en clúster de conmutación por error.

### **3.6 Réplica de servidores virtuales**

La réplica de un servidor virtual es una función de recuperación de desastres integrada en algunos hipervisores y se puede utilizar para replicar un servidor virtual en ejecución hacia una ubicación secundaria, y en algunos casos se puede extender la replicación hacia una tercera ubicación.

#### **Actividades generales para preparar un servidor virtual de base de datos con Windows Server 2016 en Hyper-V**

- Realizar instalación sistema Operativo Windows Server 2016 Standard a 64 bits con interfaz gráfica configurando idioma, zona horaria, idioma teclado, parámetros de red, configuración de firewall, usuario administrador y nombre del servidor.
- Realizar instalación de Microsoft SQL Server configurando características, collate, usuarios y servicios.
- Realizar instalación Microsoft Management Studio para administrar bases de datos.

#### **Actividades generales para preparar un servidor virtual de base de datos con UBUNTU Server 16.04 y MySQL en Hyper-V**

- Realizar instalación sistema Operativo UBUNTU Server 16.04 a 64 bits configurando idioma, ubicación, idioma del teclado, zona horaria, parámetros de red, usuarios (P/servidor y MySQL), nombre del servidor, LAMP y protocolo SSH.