

# Resumen 1

José David Mamani Vilca

## I. VISTA AL CURSO E INTRODUCCIÓN

Más allá de una breve introducción y algunos de los beneficios al ser administrador de una base de datos, este primer capítulo da a entender algunas de las herramientas que nos serán útiles en esta labor. La primera de ellas es PSequel, una interfaz de comando muy tradicional en Postgres, simple de usar y sumamente poderosa al momento de saber cual son las posibles ocurrencias en un servidor. También se destaca la importancia de los Scripts y como, a mayor tiempo se tenga trabajando con estos, mayor será tu eficiencia al administrar los recursos de un sistema.

## II. OPTIMIZANDO EL HARDWARE

Primero; si eres nuevo, ubica físicamente dónde está tu base de datos. Ya no es muy común que las empresas tengan alojados sus equipos en un rincón todo descuidado.

Hecho eso, es mejor saber que existen empresas que prestan como servicio el alojamiento de servidores. Una de las más importantes es *Compose*, empresa propiedad de IBM y que da a hogar a múltiples tipos de base de datos. Posee una excelente velocidad además de una interfaz WEB muy amigable para el administrador. Otra posible opción es Heroku, con un precio ligeramente mayor ofrece las mismas funcionalidades y capacidades.

Respecto al área de *Hardware* y Sistemas Operativos; bueno, habrá que descartar Windows y habrá que inclinarse por el uso de *SSD*. Respecto al porque descartamos Windows y vamos por Linux pues bien: Cuestiones de Licencia, Eficiencia respecto a los recursos utilizados por el mismo *OS* y facilidad para hacer modificaciones grandes en lo extenso que es PostgreSQL.

Finalmente, respecto a la instalación de Postgres, solo destacaremos la importancia de la cuenta *root*. Esta cuenta sólo deberá ser utilizada para la instalación de algunos servicios y más allá de eso no la volveremos a tocar.

## III. BLOQUEANDO CIERTAS COSAS

Dos son las cosas más importantes en una base de datos: La Seguridad y el Rendimiento. De ellas es posible tener ciertos problemas respecto a los tiempos de tu servidor pero indudablemente serás despedido si presentas fallas en asuntos de seguridad.

Como encargado de la base de datos, la configuración de cuentas será una de las labores más importantes que tendrás que desarrollar. La característica *SUDO*, por ejemplo, permite acceder como superusuario a todos los datos del sistema. Dicho privilegio es sumamente peligroso pues en primera instancia permite borrar datos, agregar tablas e incluso eliminar a otros usuarios. Dicha jerarquía es muy peligrosa y solo deberá ser otorgada a aquellos con los derechos necesarios. Otras características relevantes en cuestiones de seguridad son las llaves *SSH*. Dichos *hash* de números permiten el acceso remoto de forma mucho más simplificada pero garantizan al mismo tiempo un alto grado de seguridad.

Posteriormente extenderemos un poco más el ambiente de los superusuarios y el cómo crearlos. Básicamente usaremos los comandos *SUPERUSER*, *CREATE* y *CREATEROLE*. Hechos estos pasos, habremos definido quién crea las tablas, quién las administra, etc.

#### IV. BACK UPS Y RESTAURACIÓN DEL SISTEMA

Visto los aspectos de seguridad ahora es importante garantizar que los datos estén cuando realmente se los necesite y en caso ocurra un incidente este afecte lo más minimamente posible.

Una de las primeras pruebas como administrador de una base de datos ocurre al mover toda la información de un lugar a otro (sin importar el contexto por detrás). Para realizar dicha acción se requiere del comando *pg dump* el cuál en un instante toma una captura del estado actual de la base de datos, su estructura, etc y la almacena en un archivo. Otro comando bastante útil es *scp, secure copy* del cual por su nombre se deduce que es una utilidad que permite copiar datos de un servidor a otros.

Otras tarea importante dentro de una base de datos son los backups. Si bien ya comentamos la utilidad del comando *pg dump* para realizar estas tareas, también podemos destacar la importancia de guardar espacio en disco puesto que estas tareas suelen realizarse con suma frecuencia. Una opción a este problema deriva en la aplicación de programas comprimidores. Mediante el uso de *pipes* en conjunto con *gzip* podemos reducir en gran medida el tamaño de los backups diarios. Otra alternativa, un poco más complicada, implica en la customización de la propia herramienta *pg dump* de forma tal que se puedan omitir resultados irrelevantes para un backup diario (Personalmente lo veo muy poco recomendable.)

Con los backups explicados viene ahora la importancia de saber cuando, como y hasta donde hacerlos. Por lo general se recomienda hacer backups diarios y especialmente al anochecer. Estos backups deben conservarse durante un tiempo y finalmente descartarse cuando se vea que ya no son útiles. Sin embargo, también es importante conservar algunos de ellos, especialmente aquellos de intervalos semanales o mensuales (Para poder mantener un historial del desenvolvimiento de nuestra base de datos). Para realizar esta tarea se recomienda trabajar con un herramienta propia de Linux, CRON la cuál permite programar tareas que deben realizarse con cierta frecuencia y que resulta muy útil si se busca automatizar gran parte de la labor de mantener una base de datos.