

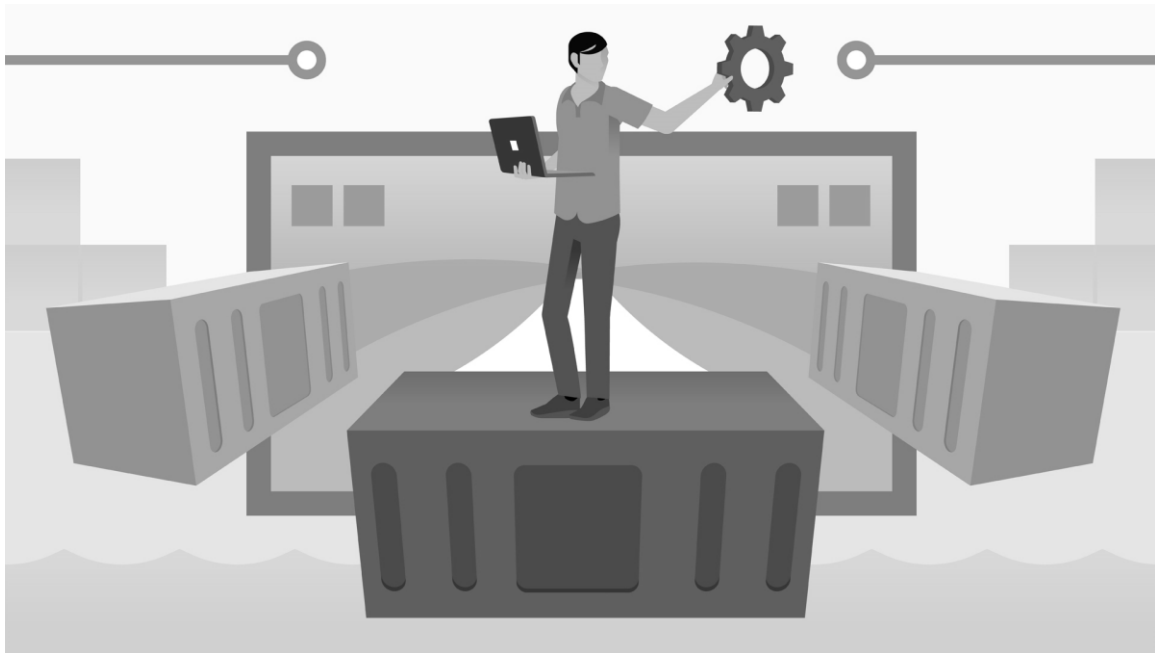
INTRODUCCION A LAS BASES DE DATOS

“¡Los datos! ¡Los datos! ¡Los datos!, gritó con impaciencia. ¡No puedo hacer ladrillos sin arcilla!”

(Sherlock Holmes)

BASES DE DATOS

Las Bases de Datos son un componente esencial de la vida cotidiana en la sociedad moderna. Actualmente, la mayoría de nosotros nos enfrentamos a diversas actividades que implican cierta interacción con una base de datos. Por ejemplo, ir a un cajero automático a depositar o extraer fondos, realizar una reserva de pasajes de avión, acceder al material informatizado de una biblioteca, visualizar mapas a través de un GPS, comprar online a través de una tienda virtual, compartir información en una red social, o simplemente buscar nuestras canciones preferidas en Spotify, son todas actividades que implican la interacción con una base de datos.



Nota

En el lenguaje cotidiano no diferenciamos datos e información, sin embargo, hay una diferencia muy importante. Los datos no tienen ningún significado. La información es algo más preciso, consiste en darle un significado a los datos bajo un contexto determinado. Es decir, si tengo el dato "100", la información sobre ese dato podría ser: "Vendimos 100 computadoras en el mes de Marzo".

De una manera simple, una base de datos es un repositorio que permite almacenar la información de forma ordenada con diferentes propósitos y usos. Las bases de datos tienen varias características importantes:

Independencia: Los datos son independientes de las aplicaciones que los usan, así como de los usuarios.

Disponibilidad: Se facilita el acceso a los datos desde contextos, aplicaciones y medios distintos, haciéndolos útiles para un mayor número de usuarios.

Mayor seguridad (protección de los datos): No todos los usuarios de un sistema deberían poder acceder a todos los datos se les debe asignar distintos permisos a los grupos de usuarios.

Menor redundancia: La repetición de datos no controlada produce un costo de almacenamiento y de acceso mayor a los necesarios. En segundo lugar, y más importante, la redundancia introduce la posibilidad de inconsistencia (Es decir, tener diversas copias de los mismos datos donde la información de los mismos puede no coincidir).

Junto con las bases de datos, el elemento fundamental para el aprovechamiento de estas son los Sistemas Gestores de Bases de Datos (SGDB o DBMS, del inglés DataBase Management System) o simplemente Motor de la base datos. Este software representa un elemento intermedio entre los propios datos y los programas que van a hacer uso de ellos, facilitando las operaciones a realizar sobre aquellos. El trabajo de un motor de base de datos no es sencillo piense en una base de datos puede tener millones de registros y ser utilizada simultáneamente por miles de usuarios, que a su vez pueden utilizar diversos programas, no todos ellos del mismo tipo. El SGBD debe proporcionar a todos ellos la metodología adecuada para extraer del conjunto de datos completo cuanto sea necesario en cada caso.

TIPOS DE BASES DE DATOS

Dentro del sector de las bases de datos nos encontraremos con bases de datos relacionales (las más usadas actualmente) y por otro lado las bases de datos NoSQL. Junto a estos dos tipos de bases de datos aparecen las bases de datos híbridas (SQL/NoSQL), bases de datos in-memory y bases de datos as a service.

BASES DE DATOS RELACIONALES

Constituyen el modelo de bases de datos más utilizado en la actualidad. Utiliza para ello un esquema basado en tablas. Las tablas (relaciones) contienen un número dado de registros (filas en la tabla), así como campos (columnas), lo que da lugar a una correcta estructuración y un acceso eficiente. Por ejemplo, una base de datos de un negocio incluiría una tabla que describiera a un cliente con columnas para el nombre, dirección, número de teléfono, y así sucesivamente. Así mismo el motor de la base de datos provee un lenguaje especial denominado SQL (Structured Query Language, Lenguaje Estructurado de Consultas) que permiten a los usuarios de las bases de datos realizar consultas. Actualmente, SQL es el lenguaje estándar de los SGBD relacionales. Los principales motores de bases de datos relacionales son Oracle, DB2 de IBM, Microsoft SQL Server, PostgreSQL y MySQL.

BASE DE DATOS NoSQL

Por otro lado tenemos las bases de datos NoSQL (Not Only SQL, No solo SQL), especialmente útiles cuando una empresa necesita acceder y analizar grandes cantidades de datos no estructurados o datos que se almacenan de forma remota en varios servidores en la nube donde existen millones de consultas diarias. En este tipo de bases de datos el sistema de almacenamiento de información no cumple con el esquema relacional ya que no imponen una estructura de datos en forma de tablas y relaciones entre ellas, en ese sentido son más flexibles, ya que suelen permitir almacenar información en otros formatos como clave-valor, Mapeo de Columnas, Documentos o Grafos. Las bases de datos NoSQL más populares son MongoDB, CouchDB, Apache Cassandra, Google BigTable y SimpleDB. Las empresas que utilizan NoSQL incluyen a Netflix, LinkedIn y Twitter.

MODELO HÍBRIDO

Algunas bases de datos SQL están adoptando características NoSQL y viceversa. Esto es lo que llamamos modelo híbrido. Cada vez más sistemas usan este tipo de modelo. Por ejemplo, IBM ha extendido su base de datos DB2 para ofrecer la posibilidad de utilizar bases de datos NoSQL.

BASES DE DATOS COMO SERVICIOS

En términos simples, las bases de datos como servicios (DBaaS, Data Base as a Service) son bases de datos que operan en Datacenters, con una gran cantidad de recursos, que pueden ser consumidos por personas o empresas desde cualquier parte del mundo, pagando sólo por lo que usan.



Nota

El concepto de SaaS (Software as a Service, Software como un servicio) ha existido desde hace mucho tiempo. Se trata de cualquier servicio basado en la Web. Tales servicios se acceden a través del navegador sin ocuparnos del desarrollo, mantenimiento, actualizaciones o copias de seguridad ya que todo ello es responsabilidad del proveedor. Típico ejemplo de este concepto es Gmail, Dropbox, Google Docs, etc.

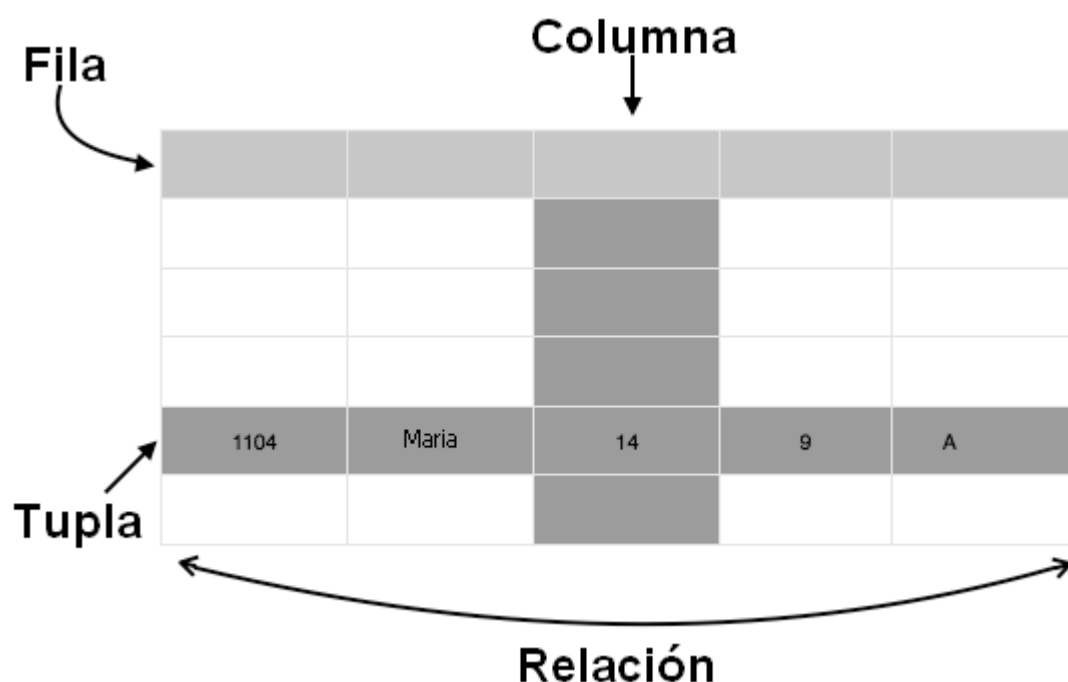
Las DbaaS son simples de usar con un pago por el servicio, ya que no requiere que nuestros administradores/desarrolladores realicen gestiones como configuraciones, copias de seguridad, parches etc. Simplemente nos podemos aprovechar de los administradores de sistemas que disponen las compañías detrás de la nube y preocuparnos solamente de mantener nuestra aplicación o servicio que queremos publicar y sus datos, inclusive nos despreocupamos de cuestiones como que sistemas operativos o que hardware tendrán nuestros usuarios para poder operar nuestras aplicaciones. Los DBaaS ofrecen tanto bases de datos relacionales (MySQL, PostgreSQL), como base de datos no relacionales o noSQL como CouchDB o MongoDB. Dentro de esta categoría podemos mencionar a Oracle DBaaS, Microsoft SQL Azure, Amazon RDS.

BIG DATA

Disponer de muchos datos para tomar una decisión no es nuevo. Lo que ha cambiado radicalmente en los últimos tiempos es la cantidad de datos de los que podemos disponer, la velocidad en que se generan esos nuevos datos y los distintos tipos de información que podemos procesar para obtener aprendizajes de negocio sobre ellos. Esto es lo que hoy conocemos como Big Data. Por ejemplo, gracias al Big Data en la ciudad de Nueva York se puede saber cuáles de los edificios de la ciudad pueden un siniestro de incendio en base a los llamados al 911, el año de construcción, los pisos y los materiales sobre el cual están contruidos. El Big Data se define como “las tres V”: volumen, velocidad y variedad. La primera “V”, Volumen, se refiere a la posibilidad de manejar grandes cantidades de información, y es la que da nombre a toda esta disciplina. La segunda “V” se refiere a la velocidad con la que se procesan estos datos, ya que una de las grandes diferencias de Big Data con los sistemas anteriores de procesamiento de información es que plantea tratar datos en tiempo real para obtener conclusiones en tiempo real, no procesa datos estáticos para obtener explicaciones a lo que ya pasó. Es un proceso vivo. Y la tercera “V” habla de la variedad de datos a tratar. En los proyectos de Big Data no sólo se procesan datos perfectamente organizados, lo que se conoce como información estructurada, sino que deben ser capaces de tratar datos no estructurados.

MODELO RELACIONAL

Aunque, como ya hemos visto, existen diversos modelos de bases de datos, las más utilizadas con diferencia en la actualidad son las relacionales, que han demostrado su idoneidad en la mayor parte de situaciones. En el modelo relacional los datos se organizan en entidades también denominada relaciones. Las entidades son las unidades de almacenamiento principal dentro de las bases de datos. Dentro de estas entidades los datos están organizados a su vez en filas y columnas. Las columnas representan los distintos atributos asociados a la entidad, mientras que las filas conforman los distintos registros. Una fila se forma con un conjunto de n atributos, constituyendo una tupla.



Un esquema contiene la definición de una estructura de la relación, es decir, el nombre de la relación, sus atributos y el dominio de cada atributo. El dominio especifica el tipo de dato a contener en cada columna. Por ejemplo, si se cargamos el nombre de una persona el atributo será de tipo alfanumérico, mientras que si el atributo es un código deberá ser de tipo entero. El número de filas de la tabla (el número de tuplas) se conoce como cardinalidad. Las relaciones son, por tanto, un conjunto de tuplas asociadas a un esquema. No debe existir dos tuplas iguales dentro de la misma relación y cada atributo puede solo tomar un único valor del dominio, es decir, no pueden contener listas de valores. En una relación, tanto el orden de las filas como el de las columnas son irrelevantes, pero es importante que cada atributo sea del tipo correspondiente a la columna a la que pertenece. Es decir, que sea coherente con el esquema. Una forma abreviada de definir las relaciones que forman parte de una base de datos es mediante su nombre y su esquema expresado como una lista de los atributos que lo constituyen. Por ejemplo, podemos definir una relación denominada TWEETS como:

TWEETS (Id, Texto, Fecha_Creacion, Id_Usuario)

Una base de datos contiene normalmente más de una tabla, ya que suelen ser muchos los tipos de datos a almacenar y resulta conveniente dividirlos en distintas tablas. Además de las relaciones que la tabla en sí implica, es necesario definir relaciones entre las distintas tablas, y para ello se emplean los denominados atributos clave. Un atributo clave es aquel que tiene valor único para cada tupla, pudiendo servir para representar a esta plenamente. Por ejemplo, en la tabla anterior se podría haber usado como clave el Id del tweet ya que en su carga no puede repetirse. Un esquema de relación puede contener varios atributos clave, que se conocen como claves candidatas. Normalmente, de estas se elige una como representante principal de las tuplas, y se conoce como clave primaria. Si una de las columnas de la interrelación es clave principal de la otra entidad a dicha columna se la denomina clave externa. Ambos conceptos de clave son extremadamente importantes. Por ejemplo, en la relación TWEETS el campo Id_Usuario puede ser clave externa en dicha entidad ya que es clave primaria en la Tabla USUARIOS (Los usuarios son los que generan los tweets). Por convención, las claves se escriben subrayadas al definir el esquema de la tabla, de tal modo que el de la tabla TWEETS quedaría de la siguiente forma:

TWEETS (Id, Texto, Fecha_Creacion, Id_Usuario)

Las interrelaciones entre tablas pueden ser de distintos tipos en función del número de elementos distintos que se vinculan de cada tabla. Por ejemplo, si tenemos la entidad USUARIOS y la queremos interrelacionar con TWEETS tendremos que un TWEET puede ser generado solo por un USUARIO de la tabla USUARIOS, mientras que un USUARIO de la tabla USUARIOS puede haber generados muchos TWEETS distintos. Es decir, cada tupla de la tabla TWEET se relaciona con una única de la tabla USUARIOS, y cada tupla de esta última se relaciona con una o varias de la primera. Este tipo de relación se conoce como de uno a muchos. Existen otros dos tipos de relaciones además de esta: las denominadas de uno a uno y las de muchos a muchos. Por ejemplo, puede haber una relación de uno a uno entre los empleados y los autos que conducen de dicha empresa.

DISEÑO Y CREACIÓN DE UNA BASE DE DATOS RELACIONAL

Concretamente, pueden distinguirse las siguientes fases en el proceso global de desarrollo de una base de datos:

- **Diseño lógico:** Independiente del SGBD empleado, es un diseño conceptual que pretende modelizar el contenido de la base de datos.
- **Diseño físico:** Es la adaptación del diseño conceptual a las particularidades del SGBD escogido.
- **Implementación:** Introducción de los datos en la base de datos.
- **Mantenimiento:** Monitorización de la actividad sobre la base de datos.

La primera fase en el diseño de una base de datos implica un análisis de los datos que se van a recoger. Como resultado de ese análisis debe surgir un modelo conceptual que exprese la estructura de la información dicho modelo debe intentar capturar del mejor modo posible la realidad que se pretende modelizar, por lo que el conjunto de tipos de datos y relaciones debe elaborarse de modo similar a dicha realidad para recoger toda la complejidad del sistema. Existen diversas metodologías para desarrollar un modelo conceptual. Una de las más extendidas por su sencillez y potencia es la del modelo entidad-relación (modelo E-R) que es una representación gráfica donde sus principales elementos son las entidades y sus interrelaciones.



Nota

El origen del modelo E-R se encuentra en trabajos efectuados por Peter Chen en 1976. Posteriormente, muchos otros autores han descrito variantes y/o extensiones de este modelo. Así pues, en la literatura se encuentran muchas formas diferentes del modelo ER que pueden variar simplemente en su notación. En nuestro curso utilizaremos un software para la creación de dichos diagramas.

LOS SGBD MÁS POPULARES

Ahora mencionaremos los motores de base de datos relacionales más populares en la actualidad:

- **MySQL:** El Motor de base de datos OPEN SOURCE más popular usado en Internet.
- **Oracle:** Uno de los más usados. Está escrito en lenguaje C++. Oracle también ha lanzado un Sistema de Base de datos.
- **IMB DB2:** Uno de los mejores productos actualmente muy usado para análisis de Big Data. Suele ser la segunda base de datos más utilizada en entornos Unix/Linux después de Oracle.
- **MS SQL Server:** Motor de Base de datos con compatibilidad únicamente para sistemas Windows. Existe una versión en la nube que forma parte de Microsoft Azure denominándose SQL Azure.
- **MariaDB:** Es un motor de Base de datos derivado de MySQL.
- **PostgreSQL:** Otro motor OPEN SOURCE, funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios.

INTRODUCCIÓN A MySQL

MySQL (www.mysql.com) es el SGBD o DBMS de código abierto más popular del mundo, gracias a su gran rendimiento, alta fiabilidad y facilidad de uso. Desde abril del año 2009 este SGBD está en manos de la empresa ORACLE CORPORATION, la empresa que tal vez más conoce sobre base de datos. MySQL es una idea originaria de la empresa MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad. MySQL está desarrollado para distintas plataformas de Hardware y Software es muy común encontrarlo en Servidores Windows/GNU Linux para desarrollos de Aplicaciones Web. Actualmente es usado por las siguientes organizaciones: GitHub, US Navy, NASA, Tesla, Netflix, Facebook, Twitter, YouTube, Spotify.

¿QUÉ ES MariaDB?

MariaDB es un derivado (fork) del proyecto MySQL creada en 2009 por Michael Widenius (también creador de MySQL) y amparada por la MariaDB Foundation, esta opción se creó debido a la disconformidad de los creadores de MySQL luego de que el producto fuera comprado por la empresa ORACLE CORPORATION. Muchas de las críticas en contra de ORACLE es que el producto no está evolucionado como los creadores lo esperaban. Esta nueva variante tiene el total apoyo de la comunidad Open Source además es el gestor de base de datos adoptado por Wikipedia, Mozilla y Google en el último año. También es importante recalcar que fue elegido el gestor por defecto de varios sistemas operativos GNU/Linux de gran relevancia como Fedora y OpenSuSe. Debido a las mejoras introducidas en este nuevo SGBD el rendimiento frente a MySQL mejoró. Por otro lado, MariaDB cuenta con una alta compatibilidad con MySQL, posee los mismos comandos, interfaces, APIs y bibliotecas, siendo su objetivo poder cambiar un servidor MySQL por MariaDB de manera directa.



Nota

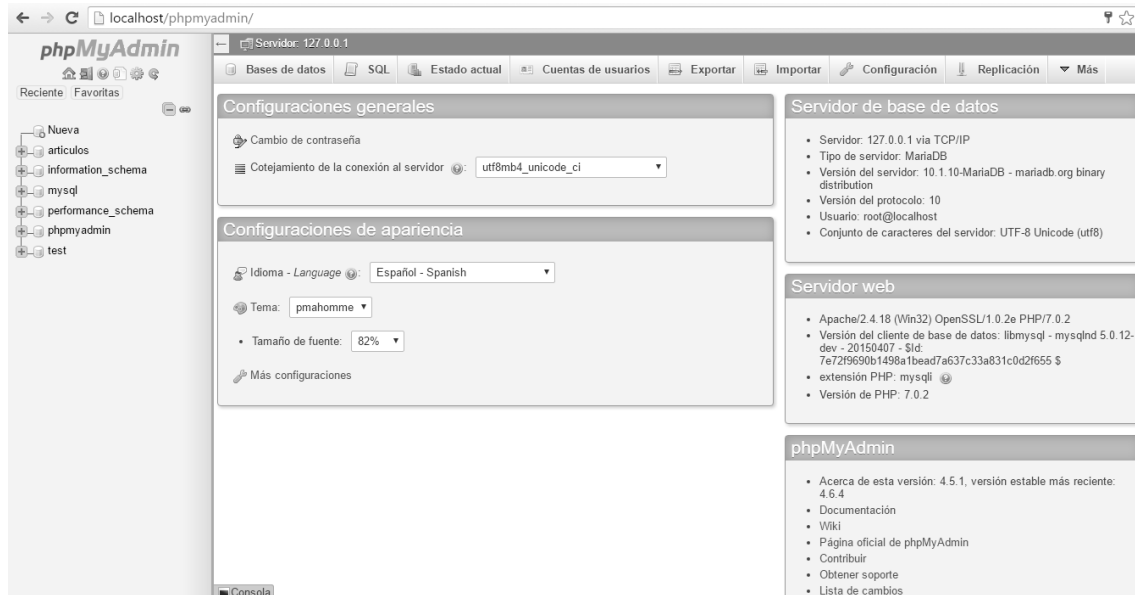
El sitio <http://db-engines.com/en/ranking> ofrece una lista de motores de base de datos rankeados según la cantidad de instalaciones en el mundo.

INTERFACES PARA OPERAR MySQL

Cuando el producto se instala no nos deja ningún entorno gráfico para trabajar lo que hace que debamos usar el CLI (Command Line Interface) Una consola donde podemos escribir comandos para interactuar con la base. Por suerte la mayoría de los programadores usan una aplicación gráfica para interactuar como: phpMyadmin, Squirrel SQL o MySQL Workbench. Sin lugar a dudas la más popular de estas herramientas es phpMyAdmin. (<http://www.phpmyadmin.net/>) es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva. Este programa está escrito en lenguaje PHP (de ahí su nombre) por una comunidad de programadores sin ánimo de lucro.

¿COMO ACCEDER A PHPMYADMIN DESDE XAMPP?

Para acceder a PHPMyAdmin deberá dirigirse a la página <http://localhost/phpmyadmin/> desde su Navegador. Al momento deberá aparecer una página de este estilo:



Esta página no es ni más ni menos que la página principal de phpMyAdmin. En la parte inferior del mismo veremos un cartel indicando que debemos tener cuidado ya que el usuario root no tiene contraseña esto se debe principalmente que XAMPP fue desarrollado para entornos de desarrollo y no de producción.

MODIFICANDO LA CONTRASEÑA DEL ROOT

Para empezar, accederemos a la línea de comandos de nuestro sistema operativo Windows. La forma más fácil es accediendo desde el menú Ejecutar y colocando el comando cmd.exe. Luego buscaremos la carpeta bin dentro del directorio mysql (que es donde se encontraran todos los binarios de esta aplicación) escribiendo el siguiente comando:

```
cd c:\xampp\mysql\bin
```

Luego dentro de este directorio ejecutaremos el siguiente comando:

```
mysql -h localhost -u root -p
```

Este comando se conecta a nuestro servidor mysql pasándole el usuario con que trabajaremos (en nuestro caso root) e indicándole que para iniciar sesión debemos escribir nuestro password. Al terminar de ejecutar este comando aparece el siguiente mensaje:

```
Enter Password:
```


Como el usuario root en nuestro servidor no posee contraseña presionaremos la tecla Enter. Ahora mediante otra instrucción procederemos a modificar la contraseña, para ello primero nos dirigiremos a la base de datos mysql con la instrucción use.

```
mysql> use mysql;
```

El punto y coma (;) al final de la instrucción indica que el comando puede ejecutarse. El resultado de la ejecución nos mostrara que hemos cambiado de base de datos. La próxima instrucción servirá para efectivamente modificar la contraseña:

```
update user set password=PASSWORD("p12345678") where user='root';
```

La instrucción update modifica el valor de un campo en este caso el campo password. Usaremos también la función PASSWORD para indicarle la nueva contraseña (en este caso a modo de ejemplo es p12345678). La cláusula where indicará que la modificación solo se hará sobre el usuario root. Finalmente aplicamos estos dos comandos:

```
mysql> flush privileges;
```

```
mysql> quit
```

El primer comando fuerza al servidor a recargar los privilegios sin necesidad de reiniciar el servidor. El segundo comando abandona la sesión iniciada anteriormente sobre el servidor. Para proseguir con nuestro ejercicio nos damos cuenta de que al iniciar nuevamente phpMyAdmin nos da un error de acceso. Esto sucede porque phpMyAdmin necesita cambiar la forma de autenticación. Para hacer este cambio debe ir con el Explorador de Windows a la carpeta donde está instalado XAMPP (C:\xampp) y buscar la carpeta que dice phpMyAdmin. A continuación, buscamos el archivo config.inc.php y lo abrimos con el block de Notas. Buscamos dentro del archivo la línea \$cfg['Servers'][\$i]['auth_type'] y modificamos el valor config por http. Luego guardamos y cerramos el archivo. El servidor debe reiniciarse después de este cambio, es decir debemos parar Apache y MySQL y ejecutarlos nuevamente. Ahora al iniciar phpMyAdmin con el navegador vera que es necesarios ingresar el usuario y la contraseña anteriormente creada para el root.

CONFIGURAR EL ACCESO REMOTO A UN SERVIDOR MySQL

Por norma general la mayoría de los servidores están preparados para recibir conexiones en forma local, es decir, desde el mismo equipo. Para ello hay que cambiar la configuración de varios archivos. Antes de ello mediante un comando configuraremos los privilegios para que los usuarios en forma remota puedan acceder a las bases de datos del servidor.

Para ello, desde el servidor, tipeamos el siguiente comando:

```
mysql> grant all privileges on *.* to root@'%' identified by 'p12345678';
```

Utilizando la sentencia `grant` podemos otorgarle uno o varios privilegios sobre los objetos de una base de datos, o la base de datos completa. En este caso le asignamos todos los permisos al usuario `root` de cualquier equipo sobre todos los objetos de todas las bases de datos.

Después de este comando aplicamos los cambios:

```
mysql> flush privileges;
```

Luego abrimos el archivo `my.ini` de la carpeta `mysql\bin` y modificamos la siguiente línea:

```
bind-address="MI_IP"
```

Donde `MI_IP` es el número de IP del equipo servidor obtenido con el comando:

```
ipconfig.exe
```

En el archivo `config.inc.php` modificamos la siguiente línea:

```
$cfg['Servers'][$i]['host'] = 'MI_IP';
```

Donde `MI_IP`, es otra vez el numero IP obtenido con el comando `ipconfig.exe`- Ahora está todo listo para que los usuarios puedan acceder a la aplicación `PHPMyAdmin` almacenado en el servidor `Mysql`. Lo único que quedaría es modificar un archivo de seguridad de `XAMPP` denominado `httpd-xampp.conf` ubicado en la carpeta `apache\conf\extra\` (puede variar entre versiones de `XAMPP`):

```
Alias /phpmyadmin "C:/xampp/phpMyAdmin/"
<Directory "C:/xampp/phpMyAdmin">
    AllowOverride AuthConfig
    Require local
    ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</Directory>
```

La línea que debe modificar es:

```
Require local
```

Por esta otra:

```
Require all granted
```