

INDICE

1. El SGBD MySQL Server.....	1
1.1. Funciones de MySQL	1
1.2. Arquitectura de MySQL.....	2

1. El SGBD MySQL Server

Este SGBD surgió como alternativa libre a los grandes SGBD. Actualmente es propiedad de Oracle tras su compra en 2009, junto con Sun Microsystems. Muy extendido en entornos web con aplicaciones de tipo XAMPP¹. MySQL posee una licencia dual, con la posibilidad de comprar una licencia comercial y una licencia GNU.

El SGBD responde a una serie de necesidades de las empresas privadas y públicas, gobiernos y todo tipo de organizaciones. Estas necesidades se cubren aportando diferentes funciones, y en base a lo numerosas y a la calidad de cada una de estas funciones, el precio de la licencia de uso de un SGBD es más o menos cara.

1.1. Funciones de MySQL

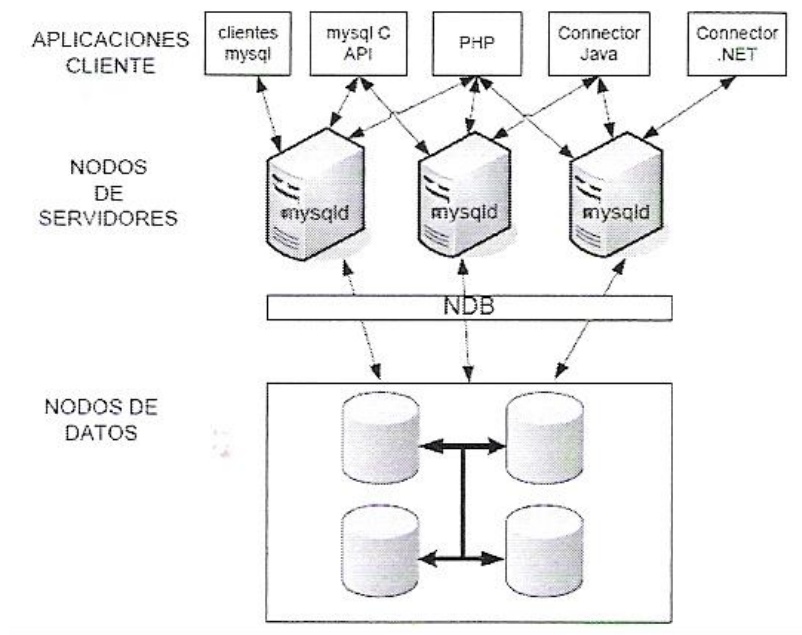
Hay que tener en cuenta que MySQL Server es un producto de Oracle Corporation desde que esta adquirió Sun Microsystems. Desde entonces, continuando con la filosofía *open source* del producto, actualmente se ofrecen las versiones: **MySQL Standard Edition**, **MySQL Enterprise Edition** y **MySQL Cluster para Web y aplicaciones de comercio electrónico**. Cada versión incorpora nuevas herramientas para la gestión avanzada del gestor.

- **MySQL Replication.** Disponible en todas las versiones, permite dotar al servidor de base de datos de la escalabilidad y fiabilidad necesaria para una gran aplicación con grandes requisitos de rendimiento. Una herramienta específica de la versión de MySQL para Cluster es la MySQL Geo Replication, para replicar bases de datos en entornos cluster. Grandes aplicaciones como Facebook o Ebay implementan replicación mediante MySQL Geo Replication para poder dar servicio a millones de usuarios.
- **MySQL Partitioning.** Disponible en las versiones Enterprise y Cluster. Permite realizar divisiones sobre las tablas y distribuirlas a través de un sistema de ficheros en base a unas reglas. Incorpora nuevas características de SQL para su gestión.
- **MySQL Enterprise Backup.** Permite realizar todo tipo de backups en caliente sin bloquear las tablas de las bases de datos. Implementa también backups incrementales y múltiples opciones de restauración de forma completa o parcial. Esta herramienta solo se puede encontrar en las versiones Enterprise y Cluster.
- **MySQL Enterprise Monitor.** Herramienta para monitorizar constantemente el rendimiento del servidor y resolver potenciales problemas que puedan ocurrirle al gestor. Consta de complementos como *Query*

¹ **XAMPP** es una arquitectura libre para el desarrollo y la publicación de sitios web con licencia GNU: X=Cualquier sistema operativo, A=Apache, M=MySQL, P=Perl,P=PHP. Sistemas: WAMP,LAMP,MAMP

Analyzer, herramienta visual para analizar las consultas enviadas al SGBD, y permite buscar problemas en el rendimiento de las mismas. Disponible únicamente en las versiones Enterprise y Cluster.

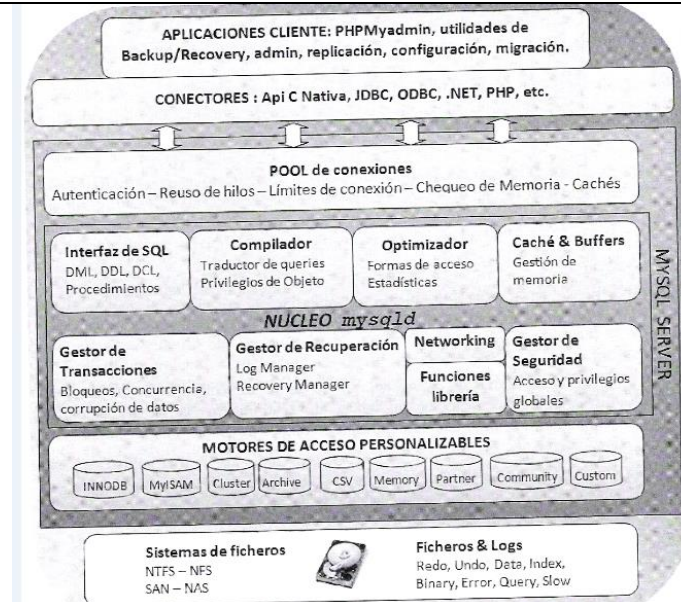
- **MySQL Cluster Manager.** Herramienta que simplifica la creación y el mantenimiento de BBDD en cluster automatizando muchas tareas de mantenimiento.
- **NDB.** Motor de almacenamiento para clusters. Se conecta a un cluster de nodos ofreciendo alta disponibilidad a través de redundancia, alto rendimiento a través del partitioning y escalabilidad mediante combinación de ambos métodos. Solo disponible en la versión Cluster.



Cluster de MySQL con NDB

1.2. Arquitectura de MySQL

En comparación con otros gestores de bases de datos, la arquitectura de MySQL es muy sencilla. En realidad, parte de su éxito consiste en conservar una arquitectura sencilla que sirve de base para aplicaciones de tamaño pequeño/mediano.

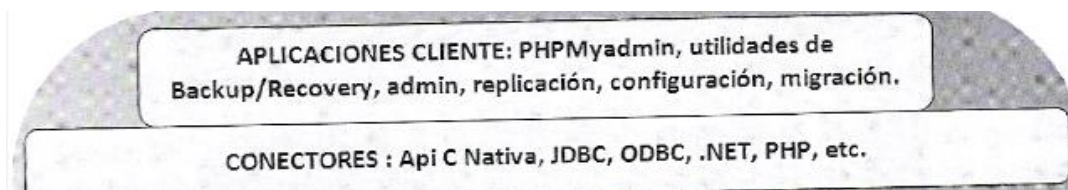


Cuando se habla de **arquitectura** de un SGBD se debe pensar en el conjunto de aplicaciones y servicios que forman parte del funcionamiento global de la BD. La arquitectura abarca desde la forma en que las herramientas o aplicaciones se conectan a la BD, hasta la forma en que se almacena los datos físicamente.

Cada capa del SGBD está pensada para flexibilizar el funcionamiento del mismo y permitir, de esta manera, que el SGBD pueda ser ejecutado en distintas arquitecturas. Así por ejemplo, el SGBD MySQL Server, se puede ejecutar en un ordenador con procesador MIPS, o con procesador x86, combinado con un SO Windows o Unix, y almacenando los datos en un servidor NFS, en un sistema de archivos NTFS o incluso en la nube.

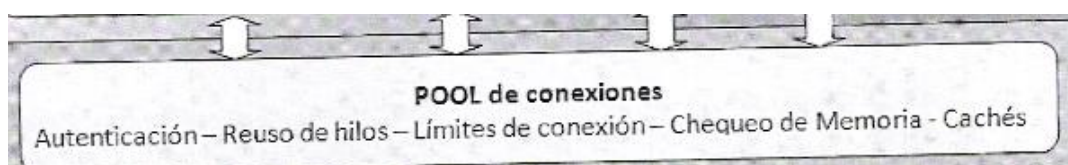
Los distintos elementos en funcionamiento de MySQL, son los siguientes:

- **Las conexiones generadas por las aplicaciones clientes**, para poder generar una conexión al SGBD, un proceso se tiene que autenticar para poder determinar a qué partes de la información puede acceder, y cuales le están prohibidas.



Arquitectura de MySQL. Las conexiones cliente y los conectores

- **MySQL** recibe las conexiones a través de una cola de conexiones o *POOL*.



Arquitectura de MySQL. El POOL de conexiones

- **El núcleo**, con todo el software que controla tanto la ejecución de consultas, como el acceso a datos, trabajo en red, etc....



Arquitectura de MySQL. El núcleo del SGBD

El núcleo del SGBD consta de:

- **Interfaz SQL:** Las aplicaciones solicitan la información al gestor a través del lenguaje SQL.

La función para conectarse a la base de datos es **mysqli_connect()**:

```
$link = mysqli_connect("localhost", "usuario", "contraseña", "nombre_base_de_datos");
```

Esta función regresa un vínculo identificador que usamos para manejar todas las peticiones subsecuentes a la base de datos. Después de esto revisamos si la conexión a la base de datos tuvo éxito, o si no mostramos un mensaje de error y salimos del programa:

```
if (!$link)
{ die("No se puede conectar a la base de datos: " . mysqli_connect_error());
}
```

Todas las peticiones pueden ser hechas después estas líneas. Una vez que terminamos con todas las peticiones de MySQL, debemos cerrar la conexión a la base de datos usando:

```
$mysqli_close($link);
```

- **Compilador:** El gestor dispone de un **Parser** que procesa la sintaxis y la semántica de la instrucción en SQL que la aplicación cliente envía al gestor.
- **Optimizador:** El gestor examina la query y, gracias a un conjunto de estadísticas que recopila constantemente, se interpreta cuál es la mejor manera de ejecutar la query.
- **Cachés y Buffers:** Son conjuntos de datos en memoria resultado de haber ejecutado consultas con anterioridad. Estos datos se almacenan temporalmente en memoria RAM para que su acceso posterior sea más rápido.
- **Gestor de Transacciones:** En este elemento se gestionan los bloqueos que son necesarios aplicar cuando un usuario o aplicación cliente accede a una tabla. Gestiona el acceso concurrente a los objetos que componen la base de datos y controla la corrupción de los datos.
- **Gestor de Recuperación:** Esta parte del núcleo controla y gestiona un elemento llamado el **Log manager**, encargado del volcado de las operaciones que se realizan guardando los comandos SQL ejecutados. El otro elemento controlado por el gestor de recuperación, el

Recovery Manager, capaz de volver a establecer la base de datos, en caso de mal funcionamiento, el último estado estable y sin incoherencias.

- MySQL está creado para ser flexible en cuanto al almacenamiento de la información, de manera que se puede personalizar cómo se almacena la información de cada tabla mediante el uso de un **Motor de Almacenamiento**.



Arquitectura de MySQL. Los motores de almacenamiento

Cada gestor de almacenamiento tiene sus características, por ejemplo **MyISAM** está pensado para funcionar en servidores web de consultas para proporcionar alta velocidad en el registro y consulta de datos. Esto se consigue gracias a que MyISAM no efectúa ningún control de integridad referencial, obteniendo una gran velocidad al no tener que realizar estas comprobaciones.

El motor **Memory**, no almacena las tablas en memoria directamente, es decir, no se almacenan de forma persistente y si, el servidor se reinicia, los datos se pierden. Esto puede tener su utilidad cuando se trabaja con procedimientos almacenados que requieren almacenar datos temporales para hacer cálculos y procesos, o por ejemplo, cuando se quiere volcar información resultante de una query a caché de forma temporal. Otro ejemplo de un motor muy usado es el motor **Federated** que permite consultar una tabla que está alojada en otro servidor remoto y operar con tablas locales (mediante joins, por ejemplo). Hay otros motores muy útiles, por ejemplo, el motor **CSV** que almacena los datos de una tabla directamente en un fichero cvs (comma separated values), el motor **INNODB** dispone de soporte para transacciones, bloqueo de registros, respeta las características ACID garantizando las integridad de nuestras tablas, es un motor muy aconsejable para un uso elevado de INSERT Y UPDATE o el motor **Archive**, que permite almacenar gran cantidad de datos en ficheros comprimidos.

- Sistemas de Ficheros, por último, tanto los datos de los motores de almacenamiento (excepto Memory) y los ficheros log que reportan la actividad del gestor, se almacenan en un sistema de ficheros bajo el control del sistema operativo.



Arquitectura de MySQL. Los sistemas de ficheros