## SEGURIDAD

Se enumeran aspectos a tener en cuenta para mantener la seguridad e integridad de los datos en la BD; se toma como modelo que las consultas a la misma serán solicitadas por un servicio web local a la BD.

* Las consultas realizadas por formulario web deberán ser cifradas antes de ser enviadas al servidor; de manera similar serán tratadas la información de respuesta a consultas (Encriptación MD5).
* La BD debe estar alojada en un servidor en una zona desmilitarizada; el firewall debe bloquear el puerto 3306. Se puede comprobar haciendo telnet al servidor con ese puerto.
* No almacenar claves sin cifrar en la BD (tabla usuarios).
* Verificar que los permisos de usuarios sean los mínimos requeridos (Comando SHOW GRANTS).
* Utilizar comillas simples en las constantes numéricas para evitar ejecución de comandos por parte de los usuarios (Comprobar errores ingresando comillas simples y dobles; comprobar todos los valores que se pasan).
* No utilizar usuario de administración para la aplicación.
* Caracteres de escape para caracteres especiales (Pendiente).

Además, si la base es consultada por diversos usuarios, además de los administradores, se debe tener en cuenta los siguientes aspectos:

* Todos los usuarios deben tener clave
* No ejecutar servidor de BD con usuario de root, generar un usuario propio para la BD.
* Privilegios PROCESS, SUPER, FILE sólo para usuarios administrativos.
* Establecer límite de conexiones por cuenta.
* Activar –safe-user-create para evitar que los usuarios sin privilegios de INSERT en la tabla de Usuarios pueda crear nuevos usuarios con GRANT.
* Activar –skip-name-resolver para evitar resoluciones erróneas o maliciosas de nombres de servidores; se guardarán las IPs directamente en las tablas.
* “--skip-networking” deshabilita las conexiones vía TCP/IP a la BD, la misma se hace a través de ficheros sockets en UNIX (Investigar más a fondo para ver si aplica).
* Restringir el permiso SHOW DATABASE a usuarios administrativos.

## CONTROL DE ACCESO A USUARIOS

Para el realizar un control de acceso seguro a la web se utiliza Spring Security basado en el proyecto Acegi que luego se integraría a Spring. Utilizando esto se pretende controlas dos aspectos principales: autenticación y autorización.

La autenticación se puede realizar mediante algún mecanismo HTTP, LDAP, basado en formulario u OpenID. La autorización se controla mediante las URLs que se pueden requerir, y los métodos, clases e instancias a utilizar.

Todo el sistema de autenticación trabaja a través de filtros para poder crear capas de seguridad. Estos filtros procesan o reenvían la petición para finalmente autenticarla. La autorización se basa en un sistema de votos el cual tiene diferentes implementaciones de acuerdo a lo que se necesite; para esto se realizan los votos sobre los roles permitidos o no, según la petición. Otra implementación de esto puede ser a través de un sistema de votos para distinguir entre acceso anónimo o autenticado.

## FUENTE

* <http://dev.mysql.com/doc/refman/5.0/es/security-guidelines.html>
* <http://dev.mysql.com/doc/refman/5.0/es/security-against-attack.html>
* <http://dev.mysql.com/doc/refman/5.0/es/privileges-options.html>
* <http://dev.mysql.com/doc/refman/5.0/es/load-data-local.html>
* <http://www.slideshare.net/SantiagoSolis1/spring-security-6890070>