Carta de recomendaciones para prevenir SQLi

❏ Validar el formato de las entradas.

- ❏ Realizar las correspondientes validaciones en la entrada de datos. Se aconseja usar sentencias preparadas o parametrizadas.
- ❏ También se puede realizar una validación de datos manual o utilizando funciones provistas por el lenguaje. Por ejemplo, la función `mysqli_real_escape_string` escapa los caracteres especiales en PHP de una cadena para usarla en una sentencia SQL, tomando en cuenta el conjunto de caracteres actual de la conexión.

❏ Respetar el mínimo privilegio.

- ❏ Creación de usuarios con permisos mínimos. Esto quiere decir, la aplicación debe usar un usuario de base de datos que tenga los permisos mínimos de acceso o modificación en las tablas.

❏ Almacenar la información sensible en un formato secreto.

- ❏ Ejemplo: claves de usuarios.
- ❏ En principio hay dos maneras de almacenar la información sensible.
  - ❏ Cifrarla a través de algoritmos criptográficos como el AES. MySQL brinda la función AES_ENCRYPT y AES_DECRYPT.
  - ❏ Hashearla a tráves de algortimos de generación de hashes como el SHA-1 o SHA-2. MySQL brinda las funciones SHA1 y SHA2. No usar las funciones MD5 ya que se consideran débiles en la actualidad.

❏ Realizar backups periódicamente.

- ❏ Los backups deben poder ser restaurados.
- ❏ Los backups deben realizarse de manera automática para evitar olvidos. MySQL brinda la herramienta mysqldump que permite generar backups.

❏ Registrar las acciones de los usuarios mediante logs.

- ❏ Los logs deben poder ser procesados para obtener información. Por esa razón, se recomienda que todos tengan el mismo formato.
- ❏ Cada log debe ser clasificado de acuerdo a algún nivel de criticidad y aquellos que pertenezcan a ataques reiterados deben poder ser fácilmente detectables. Para lograr esto último se recomienda, al menos, guardar IP de origen.

❏ Ocultar los mensajes de error que brinden información relevante.

- ❏ No mostrar los mensajes de error de la aplicación cuando la misma esté en producción.
- ❏ ¡Atención con esto!. No quiere decir que no mostrar los mensajes de errores implique que la aplicación esté libre de vulnerabilidades. Existen técnicas de SQLi a ciegas que permiten explotarlas.