

## Plan de Seguridad de la Base de Datos para el Módulo de Radiología e Imagen



### ***INTEGRANTES***

Alexis Gomez Gaona

Juan Cruz Ortiz

Janeth Ahuacatitla Amixtlan

Armando Carrazco Vargas

# **Plan de Seguridad de la Base de Datos para el Módulo de Radiología e Imagen**

## ***Introducción:***

El siguiente informe detalla el plan de seguridad propuesto para la base de datos del módulo de radiología e imagen de un hospital, utilizando una combinación de bases de datos relacional MySQL y no relacional MongoDB. La elección de esta combinación de bases de datos se basa en la necesidad de manejar una variedad de datos complejos que van desde registros clínicos detallados. La integración de sistemas relacional y no relacional permite una gestión eficiente y escalable de esta diversidad de datos, garantizando al mismo tiempo su integridad, confidencialidad y disponibilidad de los datos, así como establecer roles, usuarios, privilegios y recomendaciones para el respaldo de la base de datos. Este plan de seguridad busca proporcionar una estructura sólida y adaptable para proteger la información crítica almacenada en la base de datos del módulo de radiología e imagen del hospital, garantizando así la calidad y seguridad de la atención médica ofrecida a los pacientes.

## ***Objetivo general***

Implementar un Plan de Seguridad de la Base de Datos para el Módulo de Radiología e Imagen en el sitio web del hospital, garantizando la confidencialidad, integridad y disponibilidad de la información clínica y sensible de los pacientes, así como el cumplimiento de regulaciones y normativas de seguridad de datos.

## ***Objetivos específicos***

1. Realizar un análisis general de los requisitos de seguridad de datos específicos para el Módulo de Radiología e Imagen, considerando la sensibilidad e importancia de la seguridad de la información médica almacenada.
2. Diseñar e implementar políticas de acceso y control de usuarios para restringir el acceso no autorizado a la base de datos de radiología e imagen, asegurando que solo personal autorizado tenga permisos adecuados.
3. Establecer procedimientos de cifrado de datos para proteger la confidencialidad de la información médica durante la transmisión y almacenamiento en la base de datos.
4. Configurar y mantener sistemas de copias de seguridad regulares y redundantes para garantizar la disponibilidad y recuperación de datos en caso de fallos o desastres.
5. Realizar pruebas periódicas de la base de datos y evaluaciones de vulnerabilidad para identificar posibles puntos débiles en el sistema de seguridad y aplicar medidas correctivas.
6. Identificar los roles de usuarios necesarios: Realizar un análisis detallado de los diferentes tipos de usuarios que interactuarán con la base de datos de radiología e imagen, como radiólogos,

técnicos médicos, administradores del sistema y personal administrativo. Identificar las funciones y responsabilidades de cada tipo de usuario.

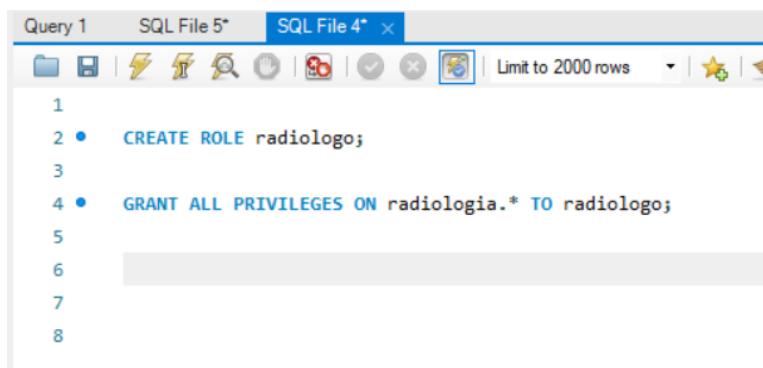
7. Definir los privilegios de acceso: Establecer los privilegios de acceso requeridos para cada rol de usuario, determinando qué acciones específicas están permitidas o restringidas para cada tipo de usuario en la base de datos. Esto incluye permisos para leer, escribir, modificar y eliminar datos, así como para ejecutar procedimientos almacenados y funciones.
8. Crear y asignar roles de usuario: Utilizando las funcionalidades de gestión de usuarios de la base de datos, crear roles de usuario personalizados que reflejen las diferentes funciones dentro del entorno de radiología e imagen. Asignar usuarios individuales a los roles correspondientes según sus responsabilidades y autorizaciones.
9. Revisar y actualizar regularmente los roles y privilegios: Realizar revisiones periódicas de los roles de usuario y los privilegios de acceso para garantizar que sigan siendo apropiados y necesarios en función de los cambios en las responsabilidades laborales y los requisitos de seguridad. Actualizar y ajustar los roles y privilegios según sea necesario.

### ***Roles, Usuarios y Privilegios:***

**Roles radiología e imagen:** Se definieron los roles específicos para diferentes usuarios, como Personal Médico, Personal Administrativo y Pacientes, cada uno con acceso y privilegios adecuados según sus responsabilidades y necesidades.

- **Personal Médico:** Acceso a historiales médicos, resultados de radiología, informes de imágenes, etc.
- **Personal Administrativo:** Acceso limitado a información no clínica, como horarios, programación de citas, etc.
- **Paciente:** Acceso a su historial médico y resultados de radiología de manera controlada.

### ***Creación de rol en MYSQL:***



```
1
2 • CREATE ROLE radiologo;
3
4 • GRANT ALL PRIVILEGES ON radiologia.* TO radiologo;
5
6
7
8
```

“CREATE ROLE”, es la funcionalidad para crear el rol, seguido del nombre, en este caso es “radiologo”.

“GRANT ALL PRIVILEGES ON” es para otorgar todos los privilegios en la base de datos "radiologia".

### Respaldo de rol en MYSQL:

```
5
6 SHOW GRANTS FOR nombre_del_rol INTO OUTFILE '/ruta/del/archivo.txt';
7
```

Los roles no se respaldan directamente como las bases de datos o las tablas, pero es posible guardar los resultados de la consulta anterior en un archivo de texto para crear un respaldo de los privilegios del rol.

Para restaurar los privilegios de un rol desde el archivo de respaldo, puedes ejecutar las declaraciones de GRANT contenidas en el archivo. Puedes hacerlo utilizando el comando SOURCE en la línea de comandos de MySQL. Por ejemplo:

```
8 SOURCE /ruta/del/archivo.txt;
```

### Creación de rol en MongoDB:

```
>_MONGOSH
test> db.createRole({
  role: "nombre_del_rol",
  privileges: [
    { resource: { db: "nombre_de_base_de_datos", collection: "" }, actions: ["accion1", "accion2", ...] },
    // Puedes especificar múltiples privilegios aquí
  ],
  roles: []
})
```

Reemplaza "nombre\_del\_rol" con el nombre que desees para el rol, "nombre\_de\_base\_de\_datos" con el nombre de la base de datos en la que desees otorgar privilegios, "accion1", "accion2", etc., con las acciones que desees permitir (por ejemplo, "read", "write", "dbAdmin", etc.). Por ejemplo, para crear un rol llamado "radiologo" con privilegios de lectura en una base de datos llamada "radiologia":

### Respaldo de rol:

```
>_MONGOSH
test> var roles = db.getRoles({ showBuiltinRoles: false });
      printjson(roles);
```

Puedes guardar la lista de roles en un archivo de texto o en un formato que prefieras. Por ejemplo, puedes utilizar `printjson()` para imprimir los roles en formato JSON y luego redirigir la salida a un archivo.

```
>_MONGOSH
test> mongo < nombre_del_script.js > backup_rols.json
```

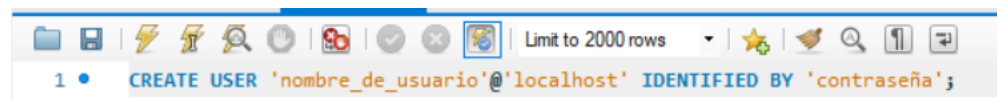
Puedes ejecutar este script en la línea de comandos de MongoDB o guardar el código en un archivo JavaScript y ejecutarlo con el comando `load()`.

## Usuarios en la Base de Datos:

Se definieron usuarios con roles definidos para garantizar la seguridad y el acceso controlado a la información. Esto asegura que cada usuario tenga acceso solo a la información relevante para su función dentro del sistema.

- **Usuario Médico:** Acceso completo a las tablas relacionadas con historiales médicos, resultados de radiología, informes de imágenes y otros datos clínicos relevantes.
- **Usuario Administrativo:** Acceso limitado a tablas no clínicas, como horarios, programación de citas, inventario de equipos médicos, etc.
- **Usuario Paciente:** Acceso controlado a su propio historial médico, resultados de radiología y otros datos personales de manera segura y protegida.

### *creación de usuario en mysql*



'nombre\_de\_usuario' es el nombre que deseas asignar al usuario.

'localhost' es el host desde el cual el usuario puede conectarse. Puedes cambiar esto según tus necesidades (por ejemplo, puedes usar '%' para permitir conexiones desde cualquier host).

'contraseña' es la contraseña que deseas asignar al usuario.

### *creación de usuario en mongodb:*

```
test> db.createUser({
  user: "nombre_de_usuario",
  pwd: "contraseña",
  roles: [ { role: "rol", db: "nombre_de_la_base_de_datos" } ]
})
```

"nombre\_de\_usuario" es el nombre que deseas asignar al usuario.

"contraseña" es la contraseña que deseas asignar al usuario.

"rol" es el rol que deseas asignar al usuario. Puedes especificar múltiples roles si es necesario.

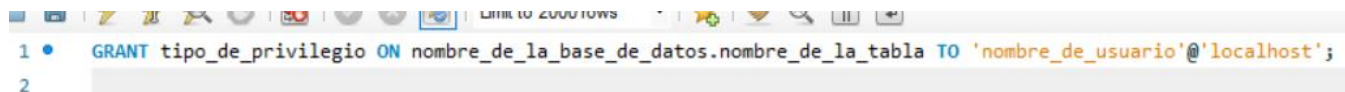
"nombre\_de\_la\_base\_de\_datos" es el nombre de la base de datos en la que deseas otorgar permisos al usuario.

## Privilegios en la Base de Datos:

Los privilegios se asignarán de manera específica para cada tipo de usuario, asegurando un acceso apropiado y seguro a los datos. Lo que significa que se otorgan solo los permisos necesarios para realizar las tareas específicas de cada usuario. Esto minimiza el riesgo de acceso no autorizado a datos sensibles.

- **Privilegios de Usuario Médico:** SELECT, INSERT, UPDATE, DELETE en tablas clínicas, acceso a procedimientos almacenados relacionados con diagnósticos, tratamientos, etc.
- **Privilegios de Usuario Administrativo:** SELECT, INSERT, UPDATE en tablas administrativas, posiblemente con privilegios de creación de informes o consultas ad-hoc.
- **Privilegios de Usuario Paciente:** SELECT en tablas de historial médico personal, acceso limitado a su propia información médica y resultados de radiología.

### Creación de privilegios en MYSQL:



```
1 • GRANT tipo_de_privilegio ON nombre_de_la_base_de_datos.nombre_de_la_tabla TO 'nombre_de_usuario'@'localhost';
2
```

'tipo\_de\_privilegio' es el tipo de privilegio que deseas otorgar, como SELECT, INSERT, UPDATE, DELETE, etc.

'nombre\_de\_la\_base\_de\_datos' es el nombre de la base de datos en la que deseas otorgar privilegios.

'nombre\_de\_la\_tabla' es el nombre de la tabla en la que deseas otorgar privilegios. Puedes usar \* para indicar todas las tablas en la base de datos.

'nombre\_de\_usuario'@'localhost' es el usuario al que deseas otorgar privilegios y el host desde el que puede conectarse. Puedes especificar '%' para permitir conexiones desde cualquier host.

### Creación de privilegios en NoSQL:

```
test> db.grantRolesToUser(
    "nombre_de_usuario",
    [
        { role: "rol", db: "nombre_de_la_base_de_datos" },
        // Puedes especificar múltiples roles aquí
    ]
)
```

"nombre\_de\_usuario" es el nombre del usuario al que deseas otorgar roles.

"rol" es el nombre del rol que deseas otorgar.

"nombre\_de\_la\_base\_de\_datos" es el nombre de la base de datos en la que deseas otorgar el rol.

### Recomendaciones para el Respaldo de la Base de Datos:

1. **Frecuencia de Respaldo:**

- Se recomienda realizar respaldos completos de la base de datos relacional (MySQL) al menos una vez al día, preferiblemente fuera del horario de mayor actividad.
- Para la base de datos no relacional (MongoDB), se sugiere un respaldo incremental cada 4 horas y un respaldo completo diario.

## 2. Métodos de Respaldo:

- Para MySQL, se utilizará el comando mysqldump para realizar respaldos completos y el registro binario para respaldos incrementales.
- En MongoDB, se empleará la función de replicación y copias de seguridad automáticas integradas para garantizar la integridad de los datos.

### **Respaldo de la base de datos en MYSQL:**

```
1 mysqldump -u nombre_de_usuario -p nombre_de_la_base_de_datos > nombre_del_archivo.sql
```

nombre de usuario es el nombre de usuario de MySQL.

`nombre_de_la_base_de_datos` es el nombre de la base de datos que deseas respaldar.

`nombre_del_archivo.sql` es el nombre que deseas asignar al archivo de respaldo SQL. Puedes elegir el nombre que prefieras.

### ***Respaldo de la base de datos en MONGODB***

```
test> mongodump --host nombre_del_host --port numero_de_puerto --username nombre_de_usuario --password contraseña --authenticationDatabase nombre_de_la_base_de_datos --db nombre_de_la_base_de_datos --out ruta_del_directorio_de_respaldo
```

- **nombre\_del\_host** es la dirección del host de MongoDB.
- **numero\_de\_puerto** es el puerto en el que MongoDB está escuchando.
- **nombre\_de\_usuario** es el nombre de usuario de MongoDB.
- **contraseña** es la contraseña del usuario de MongoDB.
- **nombre\_de\_la\_base\_de\_datos** es el nombre de la base de datos que deseas respaldar.
- **ruta\_del\_directorio\_de\_respaldo** es la ruta del directorio donde deseas guardar el respaldo.

### **Recomendaciones Adicionales:**

- Realizar pruebas regulares de restauración de respaldos para verificar su integridad.
- Mantener actualizados los sistemas de gestión de bases de datos y aplicar parches de seguridad.
- Capacitar al personal en prácticas de seguridad de la información y protocolos de acceso seguro a la base de datos.