

MANUAL 4: GUÍA DE SEGURIDAD Y ADMINISTRACIÓN - SISTEMA CONA

INFORMACIÓN DEL DOCUMENTO

Fecha de Creación: 21 de Julio de 2025

Proyecto: Sistema CONA (Gestión CONAVEG)

Audiencia: Administradores de Sistema, DevOps, Oficiales de Seguridad

Nivel: Avanzado

Tiempo Estimado: 3-5 horas (estudio y configuración)

Última Actualización: 21 de Julio de 2025

OBJETIVOS DE APRENDIZAJE

Al finalizar este manual, serás capaz de:

- ☒ Administrar el sistema completo de autenticación y autorización.
 - ☒ Gestionar usuarios, roles y permisos de manera segura.
 - ☒ Entender y aplicar las mejores prácticas de cifrado de contraseñas.
 - ☒ Configurar y monitorear el sistema de rate limiting.
 - ☒ Realizar auditorías de seguridad utilizando los logs del sistema.
 - ☒ Implementar una configuración de producción segura.
 - ☒ Ejecutar procedimientos de backup y recuperación de la base de datos.
-

REQUISITOS PREVIOS

Conocimientos Necesarios:

- Comprensión de conceptos de seguridad web (autenticación, autorización, JWT).
- Experiencia en administración de sistemas Linux/Windows.
- Conocimientos de administración de bases de datos MariaDB/MySQL.
- Familiaridad con la línea de comandos y scripting básico.

Acceso Requerido:

- Acceso de administrador al servidor donde se ejecuta CONA.
 - Permisos de superusuario (root) en la base de datos.
 - Acceso a los archivos de configuración y logs del sistema.
-

SISTEMA DE AUTENTICACIÓN Y AUTORIZACIÓN

El Sistema CONA utiliza un robusto mecanismo de seguridad basado en JSON Web Tokens (JWT) y Spring Security.

Flujo de Autenticación:

1. Login: El usuario envía credenciales (email y contraseña) al endpoint `POST /api /auth/login`.
2. Validación: El sistema verifica la contraseña contra el hash BCrypt almacenado en la base de datos.
3. Generación de JWT: Si las credenciales son válidas, se genera un JWT que contiene el ID de usuario, email y roles.
4. Acceso a Recursos: El cliente debe incluir el JWT en el header `Authorization: Bearer <token>` para todas las solicitudes a endpoints protegidos.
5. Validación de Token: Un filtro de seguridad intercepta cada solicitud, valida la firma y expiración del JWT, y establece el contexto de seguridad.

Autorización Basada en Roles:

- La autorización se controla mediante anotaciones `@PreAuthorize` en los endpoints del controller.
- Esto permite una gestión granular de permisos basada en los roles del usuario (ADMIN, GERENTE, EMPLEADO, USER).

GESTIÓN DE USUARIOS, ROLES Y PERMISOS

La gestión de usuarios y roles es una tarea exclusiva del rol ADMIN.

Roles del Sistema:

- ADMIN: Acceso total. Puede gestionar usuarios, roles y todas las entidades del sistema.
- GERENTE: Acceso de gestión a proyectos e inventario. No puede gestionar usuarios.
- EMPLEADO: Acceso de lectura a proyectos e inventario.
- USER: Rol base con acceso de solo lectura.

Matriz de Permisos Detallada:

Para una visión completa de qué rol puede acceder a qué endpoint, consulte la [docs/MATRIZ_PERMISOS_ACTUALIZADA.md](#).

Ejemplo de la Matriz:

Endpoint	Método	ADMIN	GERENTE	EMPLEADO	USER
<code>/api /users</code>	GET	✓	✗	✗	✗
<code>/api /proyectos</code>	GET	✓	✓	👁️?	👁️?

/api/proyectos	POST	✓	✓	✗	✗
----------------	------	---	---	---	---

Procedimientos Administrativos:

Crear un Nuevo Usuario (vía API):

```
# Requiere token de ADMIN
curl -X POST http://localhost:8080/conaveg/api/users \
-H "Authorization: Bearer <ADMIN_TOKEN>" \
-H "Content-Type: application/json" \
-d '{
  "userName": "nuevo.gerente",
  "email": "nuevo.gerente@conaveg.com",
  "password": "PasswordSeguro123!",
  "roleId": 2 // ID del rol GERENTE
}'
```

Cambiar el Rol de un Usuario:

```
# Requiere token de ADMIN
curl -X PUT http://localhost:8080/conaveg/api/users/5 \
-H "Authorization: Bearer <ADMIN_TOKEN>" \
-H "Content-Type: application/json" \
-d '{
  "roleId": 3 // Cambiar a rol EMPLEADO
}'
```



CIFRADO DE CONTRASEÑAS Y MEJORES PRÁCTICAS

La seguridad de las contraseñas es fundamental. CONA utiliza BCrypt con un factor de costo de 12.

Política de Contraseñas:

- Longitud: Mínimo 8 caracteres.
- Complejidad: Requiere mayúsculas, minúsculas, números y caracteres especiales.
- Almacenamiento: Las contraseñas NUNCA se almacenan en texto plano. Solo se guarda el hash BCrypt.

Mejores Prácticas para Administradores:

- Nunca pida la contraseña a un usuario. Utilice el flujo de recuperación de contraseña.
- Asegure que el costo de BCrypt (`app. security.bcrypt.strength=12`) no se reduzca en producción.
- Eduque a los usuarios sobre la creación de contraseñas seguras.
- Monitoree los logs de auditoría para detectar intentos de cambio de contraseña sospechosos.

Para más detalles técnicos, consulte la [docs/BCrypt_Usage_Guide.md](#) y [docs/Security_Best_Practices.md](#).

RATE LIMITING Y PROTECCIÓN CONTRA ATAQUES

Para prevenir ataques de fuerza bruta, el sistema implementa un mecanismo de rate limiting.

Configuración (application.properties):

```
# Habilitar/deshabilitar rate limiting
app.security.rate-limit.enabled=true

# Límite de intentos fallidos por IP en una hora
app.security.rate-limit.max-attempts-per-ip=10

# Límite de intentos fallidos por email en una hora
app.security.rate-limit.max-attempts-per-email=20

# Duración del bloqueo en minutos
app.security.rate-limit.block-duration-minutes=15
```

Cómo Funciona:

- El sistema rastrea los intentos de login fallidos por IP y por email.
- Si se superan los umbrales, la IP o el email son bloqueados temporalmente.
- Durante el bloqueo, la API responderá con un **HTTP 429 Too Many Requests**.

Monitoreo del Rate Limiting:

Los eventos de rate limiting se registran en los logs de auditoría.

Comando para buscar bloqueos recientes:

```
grep "RATE_LIMIT_EXCEEDED" logs/security.log | tail -n 20
```

Consulta SQL para ver IPs con más intentos fallidos:

```
SELECT ip_address, COUNT(*) as failed_attempts
FROM security_audit_logs
WHERE event_type = 'LOGIN_FAILED' AND timestamp > NOW() - INTERVAL 1 HOUR
GROUP BY ip_address
ORDER BY failed_attempts DESC
LIMIT 10;
```

AUDITORÍA DE SEGURIDAD Y LOGS

El sistema genera logs de auditoría detallados para todos los eventos de seguridad importantes.

Ubicación de Logs:

- Logs de aplicación: **logs/spring.log**
- Logs de seguridad: **logs/security.log**

Eventos Auditados:

- LOGIN_SUCCESS / LOGIN_FAILED
- PASSWORD_RESET_REQUESTED / PASSWORD_CHANGED
- TOKEN_REFRESH_SUCCESS / TOKEN_REFRESH_FAILED
- RATE_LIMIT_EXCEEDED
- PERMISSION_DENIED
- SYSTEM_ERROR

Análisis de Logs:

Un administrador debe revisar periódicamente los logs de seguridad para detectar actividades anómalas.

Ejemplo: Buscar todos los logins fallidos de las últimas 24 horas:

```
grep "LOGIN_FAILED" logs/security.log | grep "$(date -d '24 hours ago' +%Y-%m-%d)"
```

Ejemplo: Buscar actividad desde una IP específica:

```
grep "192.168.1.100" logs/security.log
```



CONFIGURACIÓN DE PRODUCCIÓN SEGURA

La configuración por defecto es para desarrollo. Para producción, es crucial aplicar una configuración más estricta.

Perfil de Producción:

Active el perfil `prod` para cargar la configuración de `application-prod.properties`.

```
# Ejecutar la aplicación con el perfil de producción
java -jar cona-1.0.0.jar --spring.profiles.active=prod
```

Checklist de Configuración de Producción:

- [] `app.dev.skip-authentication=false`: La autenticación NUNCA debe saltarse.
- [] `spring.jpa.hibernate.ddl-auto=validate`: Hibernate no debe modificar el esquema de la base de datos.
- [] `server.error.include-stacktrace=never`: No exponer stack traces en las respuestas de error.
- [] Variables de Entorno: Utilizar variables de entorno para todas las credenciales (base de datos, JWT secret, etc.) en lugar de archivos de propiedades.
- [] Firewall: Asegurar que solo los puertos necesarios (ej. 8080) estén abiertos al exterior.
- [] HTTPS: Configurar un proxy inverso (como Nginx o Apache) para manejar terminación SSL/TLS.

PROCEDIMIENTOS DE BACKUP Y RECUPERACIÓN

La integridad de los datos es vital. Se deben realizar backups regulares de la base de datos.

Procedimiento de Backup (MariaDB/MySQL):

Se recomienda un script que se ejecute diariamente a través de un **cron job**.

Script de Backup (**backup.sh**):

```
#!/bin/bash

# Variables de configuración
DB_USER="cona_user"
DB_PASS="TU_PASSWORD_SEGURO"
DB_NAME="conaveg_db"
BACKUP_DIR="/opt/cona/backups"
DATE=$(date +"%Y%m%d_%H%M%S")
BACKUP_FILE="$BACKUP_DIR/conaveg_db_backup_$DATE.sql.gz"

# Crear directorio de backup si no existe
mkdir -p $BACKUP_DIR

# Comando de backup
mysql dump -u $DB_USER -p$DB_PASS $DB_NAME | gzip > $BACKUP_FILE

# (Opcional) Eliminar backups antiguos (ej. más de 7 días)
find $BACKUP_DIR -type f -name "*.sql.gz" -mtime +7 -delete

echo "Backup completado: $BACKUP_FILE"
```

Configurar Cron Job (ejecutar cada día a las 2 AM):

```
# Abrir crontab
crontab -e

# Añadir la siguiente línea
0 2 * * * /path/to/your/backup.sh
```

Procedimiento de Recuperación:

En caso de desastre, se puede restaurar la base de datos desde un backup.

Script de Restauración (**restore.sh**):

```
#!/bin/bash

# Variables
DB_USER="cona_user"
DB_PASS="TU_PASSWORD_SEGURO"
DB_NAME="conaveg_db"
BACKUP_FILE=$1 # Pasar la ruta del archivo de backup como argumento

if [ -z "$BACKUP_FILE" ]; then
    echo "Uso: $0 /ruta/al/backup.sql.gz"
```

```
exit 1
fi

# Comando de restauración
gunzip < $BACKUP_FILE | mysql -u $DB_USER -p$DB_PASS $DB_NAME

echo "Restauración desde $BACKUP_FILE completada."
```

Ejecución:

```
./restore.sh /opt/cona/backups/conaveg_db_backup_20250721_020000.sql.gz
```



¡ADVERTENCIA! La restauración sobrescribirá todos los datos existentes en la base de datos. Realice este procedimiento con extrema precaución.


SOPORTE Y RECURSOS ADICIONALES


Documentación Relevante:

-  [Matriz de Permisos](#)
-  [Guía de Uso de BCrypt](#)
-  [Mejores Prácticas de Seguridad](#)
-  [Modo Desarrollo sin Autenticación](#)


Canales de Soporte:

-  Email: admin-support@conaveg.com
-  Slack: #cona-admin

 Fecha de Creación: 21 de Julio de 2025

 Responsable: Equipo de Infraestructura y Seguridad CONA

 Estado: Manual Completo y Validado

 Próxima Revisión: 21 de Agosto de 2025