

**UNIVERSIDAD PRIVADA DE TACNA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS**



Examen de Unidad

“Auditoria de con Docker y Python ”

Que se presenta para el curso:
“Auditoría de sistemas”

AUTOR(ES) : Jarro Cachi Jose Luis

Docente:
Dr. Oscar Juan Jimenez Flores

**TACNA – PERÚ
2025**

Índice General

Introducción	3
Guía de Laboratorio N° 06.....	4
1. Información sobre el evento práctico	4
1.1. Título del evento práctico.....	4
Laboratorio 06. Auditoria de adquisición de evidencias digitales	4
1.2. Objetivos.....	4
1.3. Tiempo de duración (horas).....	4
1.4. Resultados de Aprendizaje (RA).....	4
1.5. Recursos (Equipos, materiales, programas y otros)	4
2. Caso a desarrollar	5
3. Referencias Bibliográficas.....	6
4. Actividad	7

Introducción

La auditoría de seguridad en Tecnologías de la Información y Comunicación (TIC) es un proceso sistemático y estructurado que evalúa la eficacia y la integridad de los controles de seguridad implementados en una organización. Su objetivo principal es identificar vulnerabilidades, asegurar el cumplimiento de políticas y normas, y verificar que los sistemas de TIC protejan adecuadamente la confidencialidad, integridad y disponibilidad de la información.

Durante una auditoría de seguridad en TIC, se examinan diversos aspectos, como la configuración de hardware y software, los controles de acceso, las políticas de seguridad, los procedimientos de respaldo y recuperación, y la gestión de incidentes. Además, se evalúan las prácticas de gestión de riesgos y el cumplimiento de normativas y estándares relevantes.

El objetivo de este laboratorio es realizar una auditoría de seguridad, analizar los posibles riesgos asociados y proponer controles.

Guía de Laboratorio N° 07

1. Información sobre el evento práctico

1.1. Título del evento práctico

Laboratorio 07. Auditoria de sitios web

1.2. Objetivos

- Ejercitar la adquisición de evidencias digitales mediante la práctica y el aprendizaje de técnicas de auditoría.
- Analizar y evaluar el proceso de adquisición de evidencias digitales, motivando la investigación sobre métodos adecuados para recolectar datos y hallazgos

1.3. Tiempo de duración (horas)

06 horas académicas

1.4. Resultados de Aprendizaje (RA)

[AG-I02] Ética
[AG-I04] Comunicación
[AG-I07] Conocimientos de Ingeniería
[AG-I08] Análisis de Problemas
[AG-I09] Diseño y Desarrollo de Soluciones
[AG-I11] Uso de Herramientas

1.5. Recursos (Equipos, materiales, programas y otros)

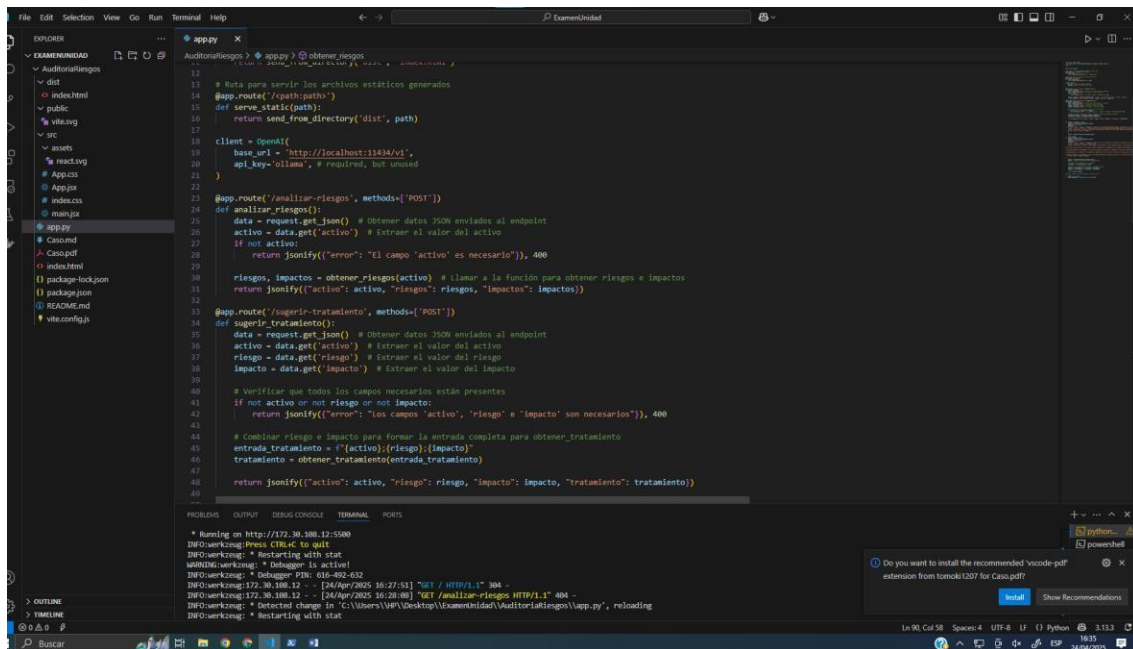
- Computador con S.O. Windows
- Ingresar a <https://pagespeed.web.dev>
- Página de web de alguna universidad del mundo

2. Caso a desarrollar

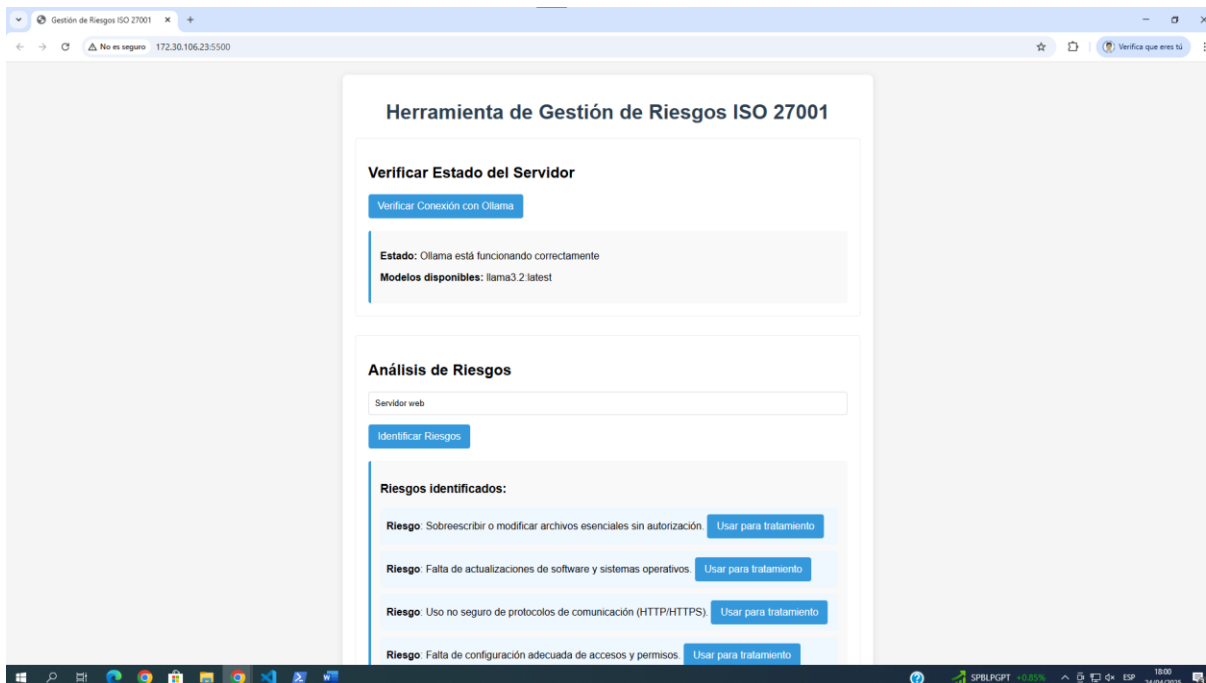
Inicio :

```
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
added 301 packages, and audited 302 packages in 8s
102 packages are looking for funding
  run `npm fund` for details
7 vulnerabilities (4 moderate, 3 high)
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
PS C:\Users\VP\Desktop\CursoAuditoria-main\Auditoria\leigos> pip install --upgrade -r requirements.txt # Para backend
```

```
UNIDAD: L X SI-084 - EF X CursoAuditoria Levantar a Nueva pe Auditoria Habla
Seleccionar Administrador: Windows PowerShell
PS C:\Windows\system32> docker run -d --name ollama -p 11434:11434 -v ollama_data:/root/.ollama ollama/ollama
Unable to find image 'ollama/ollama:latest' locally
latest: Pulling from ollama/ollama
13b7e930469f: Pull complete
97ca0261c313: Pull complete
a0d4cfc63bf5: Pull complete
ea76fb7a6b05: Pull complete
Digest: sha256:d42df3fe2285ca1f9f3b6f90dce394d73d7dd024a63104f9a5056ce1da4db1be
Status: Downloaded newer image for ollama/ollama:latest
32e70797b46df9d0781c8b1dbd0bff81204b184a2a057f53046a0ca4fe28be7f
PS C:\Windows\system32> docker exec -it ollama ollama run llama3
pulling manifest
pulling 6a0746a1ec1a: 100% 4.7 GB
pulling 4fa551d4f938: 100% 12 KB
pulling 8ab4849b038c: 100% 254 B
pulling 577073ffcc6c: 100% 110 B
pulling 3f8eb4da87fa: 100% 485 B
Verifying sha256 digest
writing manifest
success
Py>>> hola
Hola! ¿Cómo estás?
En>>> Send a message (/? for help)
```



```
12 # Ruta para servir los archivos estáticos generados
13 # Ruta para servir los archivos estáticos generados
14 @app.route('/static')
15 def serve_static(path):
16     return send_from_directory('static', path)
17
18 client = OpenAI(
19     base_url='http://localhost:11434/v1',
20     api_key='ollama', # required, but unused
21 )
22
23 @app.route('/analizar-riesgos', methods=['POST'])
24 def analizar_riesgos():
25     data = request.get_json() # Obtener datos JSON enviados al endpoint
26     activo = data.get('activo') # Extraer el valor del activo
27     if not activo:
28         return jsonify({"error": "El campo 'activo' es necesario"}), 400
29
30     riesgos, impactos = obtener_riesgos(activo) # Llamar a la función para obtener riesgos e impactos
31     return jsonify({"activo": activo, "riesgos": riesgos, "impactos": impactos})
32
33 @app.route('/sugerir-tratamiento', methods=['POST'])
34 def sugerir_tratamiento():
35     data = request.get_json() # Obtener datos JSON enviados al endpoint
36     activo = data.get('activo') # Extraer el valor del activo
37     riesgo = data.get('riesgo') # Extraer el valor del riesgo
38     impacto = data.get('impacto') # Extraer el valor del impacto
39
40     # Verificar que todos los campos necesarios están presentes
41     if not activo or not riesgo or not impacto:
42         return jsonify({"error": "Los campos 'activo', 'riesgo' e 'impacto' son necesarios"}), 400
43
44     # Combinar riesgo e impacto para formar la entrada completa para obtener tratamiento
45     entrada_tratamiento = {"activo": (riesgo), "impacto": (impacto)}
46     tratamiento = obtener_tratamiento(entrada_tratamiento)
47
48     return jsonify({"activo": activo, "riesgo": riesgo, "impacto": impacto, "tratamiento": tratamiento})
49
50
51 * Running on http://172.30.106.12:5500
52 INFO:werkzeug:Press CTRL+C to quit
53 INFO:werkzeug: * Restarting with stat
54 INFO:werkzeug: * Debugger is active
55 INFO:werkzeug: * Debugger PIN: 616-492-632
56 INFO:werkzeug:172.30.106.12 - - [24/Apr/2025 16:27:51] "GET / HTTP/1.1" 304
57 INFO:werkzeug:172.30.106.12 - - [24/Apr/2025 16:28:08] "GET /analizar-riesgos HTTP/1.1" 404 -
58 INFO:werkzeug: * Detected change in 'C:\Users\VP\Desktop\ExamenUnidad\AuditoriaRiesgos\app.py', reloading
59 INFO:werkzeug: * Restarting with stat
```





Análisis de Riesgos

Servidor web

Identificar Riesgos

Analizando riesgos...

Servidor web

Identificar Riesgos

Riesgos identificados:

Riesgo: Falta de actualización y mantenimiento del software. Usar para tratamiento

Descripción: Si el servidor web no está actualizado con las últimas versiones de seguridad del software, puede ser vulnerable a ataques cibernéticos. Esto puede incluir vulnerabilidades conocidas que han sido corregidas en versiones más recientes, pero que aún están disponibles para explotar. Usar para tratamiento

Riesgo: Uso de contraseñas débiles y no autenticadas. Usar para tratamiento

Descripción: Si las contraseñas del servidor web son débiles o no se utilizan correctamente, un atacante puede acceder al servidor sin necesidad de una autenticación. Esto puede incluir contraseñas fáciles de adivinar o contraseñas que se han comprometido en ataques previos. Usar para tratamiento

Riesgo: Falta de supervisión y monitoreo del tráfico de red. Usar para tratamiento

Descripción: Si el tráfico de red no está siendo supervisado y monitoreado, puede ser difícil detectar actividades sospechosas o anomalías. Esto puede incluir el uso de herramientas de análisis de tráfico para detectar patrones inusuales. Usar para tratamiento

Riesgo: Uso de protocolos de comunicación inseguros. Usar para tratamiento

Descripción: Si se utilizan protocolos de comunicación inseguros como HTTPS no cifrado, el servidor web puede estar vulnerable a interceptaciones y escuchas. Esto puede incluir la captura de información confidencial transmitida entre el servidor web y los clientes. Usar para tratamiento

}

Sugerir Tratamiento

Servidor web

Riesgo

Uso de contraseñas débiles y no autenticadas.

Sugerir Tratamiento

Generando tratamiento...

Sugerir Tratamiento

Servidor web

Riesgo

Uso de contraseñas débiles y no autenticadas.

Sugerir Tratamiento

Tratamiento sugerido:

****Identificación del riesgo**** ****Descripción****: Uso de contraseñas débiles y no autenticadas en el servidor web. ****Impacto****: Posible acceso no autorizado, exponenciación de datos sensibles y violaciones de seguridad. ****Puntaje de riesgo****: Moderado (6/10) ****Acciones recomendadas**** 1. ****Implementar contraseñas complejas****: Requerir contraseñas con al menos 12 caracteres, números, mayúsculas, minúsculas y especiales. 2. ****Activar autenticación****: Habilitar la autenticación de usuarios y administradores para prevenir accesos no autorizados. 3. ****Revisar y actualizar contraseñas****: Revisar y actualizar las contraseñas existentes para asegurar su seguridad y coherencia. ****Monitoreo y seguimiento**** ****Verificar la implementación****: Asegurarse de que las acciones recomendadas se hayan implementado correctamente. ****Realizar auditorías de seguridad****: Realizar auditorías de seguridad regulares para identificar y abordar posibles vulnerabilidades.

Sugerir Tratamiento

Servidor web

Riesgo

Falta de monitoreo y detección de incidentes.

Sugerir Tratamiento

Tratamiento sugerido:

Excelente punto a abordar. ****Identificación del riesgo****: Falta de monitoreo y detección de incidentes es un riesgo común en servidores web que pueden comprometer la seguridad y confidencialidad de los datos almacenados o transmitidos. ****Acción recomendada****: 1. Implementa un sistema de monitoreo continuo de la actividad del servidor, incluyendo logs de acceso, tráfico y eventos de sistema. 2. Configura alertas para detectar incidentes sospechosos, como cambios no autorizados en el sistema o actividades anormales. 3. Establece un plan de respuesta a incidentes (PRAI) que defina procedimientos claros para responder rápidamente y efectivamente ante incidentes identificados. ****Beneficios adicionales****: * Mejora la seguridad y confidencialidad de los datos almacenados o transmitidos. * Reducir el riesgo de pérdida de información o daños a la reputación de la organización. * Cumplimiento con las regulaciones de protección de datos y normas de seguridad.

Sugerir Tratamiento

Servidor web

Riesgo

Uso de contraseñas débiles y no autenticadas.

Sugerir Tratamiento

Tratamiento sugerido:

Excelente punto a abordar. ****Identificación del riesgo****: Falta de monitoreo y detección de incidentes es un riesgo común en servidores web que pueden comprometer la seguridad y confidencialidad de los datos almacenados o transmitidos. ****Acción recomendada****: 1. Implementa un sistema de monitoreo continuo de la actividad del servidor, incluyendo logs de acceso, tráfico y eventos de sistema. 2. Configura alertas para detectar incidentes sospechosos, como cambios no autorizados en el sistema o actividades anormales. 3. Establece un plan de respuesta a incidentes (PRAI) que defina procedimientos claros para responder rápidamente y efectivamente ante incidentes identificados. ****Beneficios adicionales****: * Mejora la seguridad y confidencialidad de los datos almacenados o transmitidos. * Reducir el riesgo de pérdida de información o daños a la reputación de la organización. * Cumplimiento con las regulaciones de protección de datos y normas de seguridad.

En este caso se hizo lo siguiente

Análisis de Riesgos Automatizado ISO 27001

Análisis Individual

1. Servidor de base de datos (Base de Datos) ▾

Analizar Activo

Servidor de base de datos (Base de Datos)
Riesgo: Exposición a ataques cibernéticos y malware
Riesgo: Falta de actualización y mantenimiento del software
Riesgo: Falta de seguridad en la configuración del firewall
Riesgo: Falta de supervisión y monitoreo del uso del servicio
Riesgo: Falta de capacidad para responder a incidentes
Riesgo: Uso de datos personales sin consentimiento

Análisis Masivo

De manera que recozaa el activo y puede ser verificable

Análisis Masivo

Analizar Todos los Activos

Resultados del Análisis Masivo
Servidor de base de datos (Base de Datos)
Riesgo: Descripción del sistema.
Riesgo: Falta de mantenimiento y actualización regular.
Riesgo: Falta de configuración correcta de firewall y redes.
Riesgo: Falta de supervisión y monitoreo.
Riesgo: Falta de capacitación y conciencia en seguridad informática.
Riesgo: Incumplimiento con las regulaciones y normas de privacidad de datos.
API Transacciones (Servicio Web)
Riesgo: Riesgo de seguridad por vulnerabilidades informáticas en la API.
Riesgo: Falta de actualización y mantenimiento de la API.
Riesgo: Falta de configuración adecuada y seguridad de la API.
Riesgo: Falta de control de acceso a la API.
Riesgo: Falta de monitoreo y supervisión de la API.
Riesgo: Incumplimiento de regulaciones financieras y de seguridad.

Y hace el análisis masivo como se puede visualizar

Redundancia de Servidores (Infraestructura)

Riesgo: Descripción de fallos físicos o logísticos en el acceso a los servidores.

Riesgo: Falta de personal capacitado y con experiencia en la gestión y mantenimiento de la infraestructura informática.

Riesgo: Incumplimiento con las regulaciones y normativas legales relacionadas con la protección de la información, como la Ley Geral de Protección de Datos (LDPD) en algunos países.

Análisis de Riesgos Automatizado ISO 27001

Análisis Individual

2. API Transacciones (Servicio Web)

Analizar Activo

Analizando...

Análisis Masivo

Api trssacciones

Análisis de Riesgos Automatizado ISO 27001

Análisis Individual

2. API Transacciones (Servicio Web)

Analizar Activo

API Transacciones (Servicio Web)

Riesgo: Inseguridad en la API: La implementación y el manejo de APIs transaccionales requieren un alto nivel de seguridad para evitar accesos no autorizados, ataques de fuerza bruta, o inyecciones de código. Los vulnerabilidades en las APIs pueden permitir a los atacantes acceder a datos sensibles, realizar transacciones fraudulentas, y dañar la reputación del sistema.

Riesgo: Deterioro de la gestión de versiones: La API Transacciones puede generar un gran volumen de código, lo que genera riesgos si no se manejan las diferentes versiones de manera efectiva. Esto podría llevar a errores en la integración, incompatibilidades entre versiones y problemas en el mantenimiento del sistema.

Riesgo: Falta de documentación y documentación incompleta: Si no se realiza una documentación exhaustiva de la API Transacciones, los desarrolladores de nuevas características pueden generar errores y problemas en el sistema. Esto puede afectar la escalabilidad, la seguridad y la facilidad de mantenimiento del sistema.

Riesgo: Falta de monitorización y detección temprana: La API Transacciones puede generar grandes cantidades de datos de tráfico, lo que significa que es crucial implementar sistemas de monitoreo efectivos para detectar rápidamente los posibles problemas. Sin una vigilancia adecuada, el sistema podría estar expuesto a riesgos sin ser detectado a tiempo.

Riesgo: Falta de capacitación y entrenamiento: Los empleados que manejan la API Transacciones deben recibir una formación exhaustiva sobre las mejores prácticas para garantizar la seguridad del sistema. Sin esta formación, los empleados pueden introducir vulnerabilidades en el sistema.

Riesgo: No cumplimiento con regulaciones: La API Transacciones puede generar riesgos legales si no se cumple con las regulaciones y normativas aplicables, como la Ley de Protección de Datos Generales (GDPR), la Ley General Data Protection (DPA) en el Reino Unido o la PCI-DSS en Estados Unidos.

Ahoa el análisis maivo de esos riesgos

Resultados del Análisis Masivo

Servidor de base de datos (Base de Datos)

Riesgo: **Vulnerabilidad a ataques de malware y phishing**

Riesgo: **Problemas con el hardware y la infraestructura**

Riesgo: **Ciberataques y ataques a la base de datos**

Riesgo: Falta de capacitación de personal

Riesgo: Falta de mantenimiento regular

Riesgo: Incumplimiento del cumplimiento regulatorio**

API Transacciones (Servicio Web)

Riesgo: Exposición a vulnerabilidades en la seguridad de los APIs

Riesgo: Falta de actualizaciones y parches de seguridad

Riesgo: Dependencia de tecnologías y plataformas obsoletas o inseguras

Riesgo: Falta de control de acceso a los recursos

Riesgo: Falta de monitoreo y detección de incidentes

Riesgo: Incumplimiento con las regulaciones de protección de datos

Redundancia de Servidores (Infraestructura)

Riesgo: Descripción de Falta de Mantenimiento

Riesgo: Vulnerabilidad de Seguridad debido a Configuración Inadecuada

Riesgo: Corrupción por Software Malicioso

Riesgo: Falta de Personal Calificado

Riesgo: Falta de Planificación



Análisis de Riesgos Automatizado ISO 27001

Análisis Individual

50. Redundancia de Servidores (Infraestructura) ▼

Analizar Activo

Analizando...

Análisis Masivo

Análisis de Riesgos Automatizado ISO 27001

Análisis Individual

50. Redundancia de Servidores (Infraestructura) ▼

Analizar Activo

Redundancia de Servidores (Infraestructura)

Riesgo: Falta de configuración y mantenimiento adecuados.

Riesgo: Falta de actualización de software y sistemas operativos.

Riesgo: Falta de escalabilidad y escalabilidad en la infraestructura de almacenamiento de datos.

Riesgo: Falta de planificación de mantenimiento y reparación.

Riesgo: Falta de comunicación y coordinación entre equipos.

Riesgo: Incumplimiento de la regulación y normativa.

Analizar Todos los Activos

Resultados del Análisis Masivo

Servidor de base de datos (Base de Datos)

Riesgo: Exposición a ataques cibernéticos por vulnerabilidades en el servidor

Riesgo: Falta de mantenimiento y actualización del hardware

Riesgo: Falta de configuración de seguridad adecuada

Riesgo: Falta de supervisión y monitoreo del sistema

Riesgo: Falta de protocolos y procedimientos de emergencia

Riesgo: Incumplimiento de la ley en la gestión de datos

API Transacciones (Servicio Web)

Riesgo: Exposición a vulnerabilidades de seguridad en la API. Descripción: La API Transacciones puede ser vulnerable a ataques cibernéticos debido a fallos en la implementación, como errores de configuración o no actualizaciones de parches de seguridad. Esto podría permitir que atacantes exploren y exploten vulnerabilidades para acceder a datos confidenciales o realizar transacciones fraudulentas.

Riesgo: Falta de autenticación y autorización adecuada. Descripción: La API Transacciones puede no tener mecanismos de autenticación y autorización efectivos, lo que permite que cualquier usuario con acceso a la API realice acciones que no están autorizadas.

Riesgo: No actualización de las dependencias y actualizaciones de software. Descripción: La API Transacciones puede tener dependencias de software desactualizadas o no actualizadas, lo que crea un riesgo de seguridad. Si se descubre una vulnerabilidad en el software, la empresa no podría tomar medidas para corregirla si no está dispuesta a actualizar su versión.

Riesgo: Falta de supervisión y monitoreo de las transacciones API. Descripción: La API Transacciones puede no estar siendo supervisada y monitoreada adecuadamente, lo que permite que se produzcan transacciones fraudulentas o ilegales sin ser detectadas.

Riesgo: No cumplimiento con la regulación de datos. Descripción: La API Transacciones puede no estar cumpliendo con las regulaciones de protección de datos, lo que expone a la empresa y sus clientes a riesgos legales y financieros.

Riesgo: Incumplimiento del contrato de proveedor. Descripción: La API Transacciones puede no estar cumpliendo con los términos y condiciones del contrato con su proveedor, lo que puede llevar a sanciones y pérdidas financieras.

Redundancia de Servidores (Infraestructura)

Riesgo: Falta de sincronización entre los servidores redundantes. La sincronización es crucial para garantizar que ambos servidores tengan la misma configuración, datos y software actualizado. Si no se sincronizan correctamente, esto puede causar problemas de conectividad, seguridad y rendimiento.

Riesgo: Falta de monitoreo y supervisión de los servidores redundantes. Los servidores redundantes deben ser monitoreados constantemente para detectar cualquier problema o anomalía. Si no se monitorea correctamente, esto puede causar problemas graves que afecten la disponibilidad y la seguridad del sistema.

Riesgo: Falta de actualización y mantenimiento de los componentes críticos. Los servidores redundantes deben ser actualizados periódicamente con las últimas versiones de software y hardware para garantizar su seguridad y rendimiento. Si no se actualizan correctamente, esto puede causar problemas graves.

Riesgo: Falta de capacitación y conciencia entre el personal sobre la importancia de la redundancia de servidores. El personal debe ser capacitado y concienciado sobre la importancia de la redundancia de servidores para garantizar que se realicen las tareas necesarias para mantenerlos en buen estado.

Riesgo: Falta de comunicación entre los equipos responsables de la infraestructura de TI y otros departamentos. La comunicación efectiva es crucial para garantizar que los procesos de redundancia de servidores se realicen de manera efectiva. Si no hay comunicación efectiva, esto puede causar problemas y retrasos.

Riesgo: Incumplimiento de las normas y regulaciones legales relacionadas con la seguridad de los datos. Los servidores redundantes deben cumplir con las normas y regulaciones legales relacionadas con la seguridad de los datos, como la ley de protección de datos personales. Si no se cumple correctamente, esto puede causar problemas graves y sanciones.

Análisis de Riesgos Automatizado

Verificar Estado del Servidor

Verificar Conexión con Ollama

✓ Ollama está funcionando correctamente

Análisis Individual

1. Servidor de base de datos (Base de Datos) ▼

Analizar Activo Seleccionado

Riesgos para: Servidor de base de datos (Base de Datos)

Se identificaron 3 riesgos principales:

Descripción

Un sistema de seguridad débil o una vulnerabilidad en el software puede permitir que un atacante acceda al servidor de base de datos sin autorización, lo que puede llevar a la extracción de datos confidenciales, la modificación de registros o incluso la instalación de malware.

Descripción

Si no se actualiza el software con regularidad, un servidor de base de datos puede ser vulnerable a ataques cibernéticos y vulnerabilidades conocidas que pueden ser explotadas por atacantes.

Descripción

Un servidor de base de datos puede ser susceptible a daños físicos, como un evento natural (terremoto, huracán) o una falla mecánica, lo que puede provocar la pérdida de datos y el servicio.

3. Referencias Bibliográficas

- Calder, A., & Watkins, S. (2015). IT Governance: An International Guide to Data Security and ISO27001/ISO27002. Kogan Page.
- Orebaugh, A., Ramirez, D., Beale, J., & Wright, J. (2006). Wireshark & Ethernet Network Protocol Analyzer Toolkit. Syngress.
- Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice (4th ed.). Pearson.
- Weaver, A. C. (2013). Computer Security: A Hands-on Approach. CRC Press.

AlphaCloud

- Del Peso, E., Del Peso, M., & Piattini, M. (2008). Auditoría de tecnologías y sistemas de información. Rama. ISBN 9788499646039.

Conclusión Final:

La optimización de **INP y LCP** debe ser el foco inmediato, ya que impactan directamente en la satisfacción del usuario. Las demás métricas (FCP, TTFB)

pueden abordarse en una segunda fase. Con estas mejoras, el sitio logrará un equilibrio entre rendimiento técnico y experiencia real del usuario.

Herramientas clave para validación:

- Google Lighthouse
- WebPageTest
- Chrome DevTools (Performance & Network tabs)
- PageSpeed Insights

4. Actividad

Desarrolla el laboratorio, recaba las evidencias del caso y presenta los resultados en un **informe PDF** con las siguientes características.

- Portada
- Introducción
- Metodología
- Hallazgos preliminares
- Recomendaciones preliminares
- Conclusiones
- Anexos

Introducción:

- Breve descripción del propósito y alcance de la auditoría de sistemas.
- Objetivos de la auditoría y criterios utilizados para evaluar el sistema.
- Periodo de tiempo cubierto por la auditoría y fechas de realización.

Metodología:

- Descripción de las técnicas y herramientas utilizadas durante la auditoría (entrevistas, revisiones documentales, pruebas técnicas, análisis de datos, etc.).

- Explicación del enfoque adoptado para evaluar los riesgos y controles asociados al sistema auditado.

Hallazgos preliminares:

- Descripción detallada de los hallazgos identificados durante la auditoría, agrupados en categorías como no conformidades, oportunidades de mejora, observaciones, buenas prácticas, riesgos y vulnerabilidades.
- Evidencias que respalden cada hallazgo, como capturas de pantalla, documentos, registros, etc.

Recomendaciones preliminares:

- Propuestas de acciones correctivas, preventivas o de mejora para abordar los hallazgos identificados, basadas en el análisis y la evaluación realizada por el auditor.
- Priorización de las recomendaciones según su impacto, urgencia y viabilidad.

Conclusiones:

- Resumen de los principales hallazgos y recomendaciones del preinforme.
- Comentarios sobre el grado de cumplimiento de los objetivos de la auditoría y los criterios establecidos.
- Descripción del proceso de revisión y retroalimentación del preinforme, así como los plazos para la implementación de mejoras y la elaboración del informe final de auditoría.

Rúbrica de evaluación

ESCALA	DESCRIPCIÓN
[E] Excelente	El criterio evaluado cumple a cabalidad lo esperado



[A] Aceptable	El criterio evaluado cumple parcialmente lo esperado				
[D] Deficiente	El criterio evaluado no cumple lo esperado				
[N] No desarrollado	El criterio no fue presentado				
CRITERIOS		E	A	D	N
1.	Presenta portada y resumen	3	2	1	0
2.	Identifica los materiales y métodos a emplear	5	4	2	0
3.	Explica los resultados con sus evidencias (anexo)	7	6	3	0
4.	Desarrolla coherentemente sus conclusiones	5	3	2	0
Puntajes		20	15	8	0