INFORME FINAL DEL SISTEMA

BISOFT-32 PROCESO DE INGENIERÍA DEL SOFTWARE I

19 DE MARZO DE 2025

Sede San Pedro

1. Informe de Pruebas de Control del Sistema de Información

1.1. Datos Generales del Informe

- **Nombre del Sistema**: P-Logistics Rutas
- **Versión del Sistema**: V1.0
- **Fecha del Informe**: 19 marzo de 2025
- **Responsable de la Prueba**: Javier Diaz Mora
- **Equipo de Pruebas**: Sebastian Vargas Delgado, Mauricio Taylor Fonseca
- **Entorno de Pruebas**: Desarrollo

1.2. Introducción

1.2.1 Descripción del Sistema

La aplicación será una aplicación web que permite gestionar rutas de forma dinámica. Se podrán crear nuevas rutas, asignar las rutas y modificarlas según sea necesario. La interfaz es amigable y accesible, garantizando que tanto usuarios técnicos como usuario normales puedan operar sin dificultad.

La aplicación será un sistema de gestión de rutas eficiente que permita a los administradores crear, modificar y asignar rutas paquetes. Utilizado las siguientes tecnologías: Html, css, javascript, C# y Azure SQL Server

Se espera que la aplicación permita la creación, asignación y modificación de rutas de manera eficiente y que cuente con una interfaz intuitiva, para una mejor experiencia de usuario y que facilite la interacción entre la aplicación y los usuarios normales. Se espera poder escalar la aplicación para que permita futuras actualizaciones y la incorporación de nuevas funcionalidades según las necesidades del usuario.

- **Objetivo del Sistema**: Desarrollar un sistema de gestión de rutas eficiente que permita a los administradores crear, modificar y asignar rutas paquetes.
- **Tecnología**: C#, JavaScript, HTML, CSS, Azure SQL Server
- **Usuarios Objetivo**: Los administradores y choferes de envíos de paquetes.

1.2.2 Objetivo de las Pruebas de Control

Se debe especificar el propósito de realizar las pruebas de control. Aquí se explican las razones por las que es necesario validar los controles en el sistema y qué aspectos se espera cubrir con estas pruebas.

- **Propósito**: Verificar la integridad, seguridad, accesibilidad, y confiabilidad del sistema.
- **Importancia**: Asegurar que el sistema cumpla con las normativas, estándares internos o externos y que no tenga vulnerabilidades que pongan en riesgo la confidencialidad, integridad y disponibilidad de la información.

1.2.3 Alcance de las Pruebas

El alcance de las pruebas define qué áreas del sistema fueron probadas y cuáles quedaron fuera de este informe.

- **Componentes Evaluados**: seguridad, interfaces, APIs
- **Áreas Excluidas**:
    - Aplicación móvil: No es posible acceder a la aplicación móvil
    - Base de datos: se tendría que crear desde cero la base de datos(tablas, procesos almacenados, datos), no hay posibilidad de restaurarla sin demorar mucho tiempo.

1.3. Metodología de Pruebas

1.3.1 Enfoque General de las Pruebas

Aquí debes detallar el enfoque y tipo de pruebas que se utilizaron. Esto puede incluir el tipo de pruebas funcionales, de seguridad, de rendimiento, etc.

- **Pruebas Funcionales**: Evaluar que el guardado de datos sea correcto y que las pestañas funcionen correctamente.
- **Pruebas de Seguridad**: Que los datos estén cifrados, que la sesión se cierre automáticamente después de un tiempo de inactividad.
- **Pruebas de Integridad de Datos**: Verificar que los datos se guarden correctamente en base de datos
- **Pruebas de Rendimiento**: Evaluar los tiempos de respuesta de las páginas web.
- **Pruebas de Usabilidad**: Evaluación de la facilidad de uso y accesibilidad del sistema para los usuarios.

1.3.2 Técnicas de Pruebas

Describe las técnicas y herramientas utilizadas para realizar las pruebas.

- **Técnicas Manuales**:Pruebas manuales con postman para validar, como pruebas de accesos para verificar funcionamiento de usuarios, revisión de logs y salidas de consolas, pruebas de interfaz y navegación entre ellas.
- **Técnicas Automáticas**:No se realizaron pruebas automáticas.
- **Pruebas de Caja Negra/Caja Blanca**:Las pruebas fueron de caja blanca.

1.3.3 Criterios de Aceptación

Define qué condiciones deben cumplirse para que las pruebas sean consideradas exitosas.

- **Criterios para Aprobación de la Prueba**: El cumplimiento de los requisitos primordiales, la validación de que los datos sean correctos y que presente los errores en lugar de caerse el programa.

1.4. Descripción de los Controles Probados

En esta sección se describen detalladamente los controles específicos que fueron evaluados. Esto incluye controles de acceso, seguridad, integridad de datos, etc.

1.4.1 Control de Acceso

- **Descripción**: Control de roles y permisos, autenticación y autorización de usuarios.
- **Pruebas Realizadas**: Pruebas sobre diferentes roles de usuarios y el manejo de datos como rutas, camiones y choferes los cuales son administrables solo por el rol admin para el resto de roles se debe solo leer la información y asignar más no editar crear o inactivar verificando que los accesos están restringidos adecuadamente.
- **Resultado**: Aprobado
- **Observaciones**:Se debe dar cierta gestión para los usuarios ya que los encargados de logística deben poder asignar rutas y camiones a un chofer más no editar la información, así como choferes deben solo visualizar la información del camion y la ruta asignada sin tener que cambiar nada

1.4.2 Control de Integridad de Datos

- **Descripción**: Verificar que los datos a ingresar sean los correctos y que no inyeccion de codigo
- **Pruebas Realizadas**: Pruebas de la integridad de los datos y que los campos solicitados fueran los correctos al ingresar.
- **Resultado**: Aprobado
- **Observaciones**: Se debe de dar un nuevo mantenimiento para comprobar que dichas validaciones sigan funcionando correctamente.

1.4.3 Control de Seguridad

- **Descripción**: Evaluación de las protecciones de seguridad implementadas.
- **Pruebas Realizadas**: los datos más vulnerables enviados desde la página web a la base de datos se encriptan.
- **Resultado**: Aprobado.
- **Observaciones**: solo se encriptan las contraseñas.

1.4.4 Control de Backup y Recuperación

- **Descripción**: Evaluación de los procedimientos de copia de seguridad y recuperación ante desastres.
- **Pruebas Realizadas**: Ninguna ya que no había métodos de recuperación de datos.
- **Resultado**: No Aprobado
- **Observaciones**: N/A.

1.4.5 Control de Monitoreo

- **Descripción**: Revisión de los sistemas de monitoreo de eventos y alertas.
- **Pruebas Realizadas**: El sistema no llegó a un punto de producción por lo que no se generaron modelos de monitoreo de uso de eventos y alertas
- **Resultado**: No Aprobado.
- **Observaciones**: N/A

1.5. Resultados de las Pruebas

1.5.1 Resumen de Resultados Generales

Proporciona un resumen global de cómo han resultado las pruebas. Indica si se lograron los objetivos y si el sistema está en condiciones adecuadas.

- **Porcentaje de Controles Aprobados**: 60%
- **Deficiencias Críticas**: La falta de mantenimiento que se le ha brindado al sistema ha generado que muchas de sus librerías se mantengan desactualizadas.
- **Áreas de Mejora**: implementar sistema de recuperación de datos y backups.

1.5.2 Incidencias y Deficiencias Encontradas

Detalles específicos de las incidencias y deficiencias encontradas durante las pruebas.

Incidencia	Descripción	Impacto	Estado	Recomendaciones
Gestión eficiente de credenciales	Se encripta sólo las contraseñas lo cual deja el resto de datos expuestos	Alto	Pendiente	Evaluar datos sensibles para encriptar y que datos pueden quedar públicos
Vulnerabilidad de API	Necesita mantenimiento y actualización de versiones de API compatibles o necesarias	Alto	Pendiente	Brindar mantenimiento necesario a las APIs
Control de Versiones	Mantenimiento, control y actualización de versiones	Medio	Pendiente	Actualizar los controles y revisar los conflictos generados para resolver.
Creación de backups	Crear backups de los datos cada cierto tiempo.	Alto	Pendiente	Crear un método en la base de datos que guarde backups cada 30 días.
Creación de respaldo	Crear un respaldo del sistema en caso de fallo del actual.	Alto	Pendiente	Crear un respaldo del sistema para evitar retrasos al negocio.

1.5.3 Recomendaciones Generales

Proporciona recomendaciones sobre cómo mejorar los controles, procedimientos y configuraciones del sistema.

- **Recomendación 1**: Brindar mantenimiento constante y las actualizaciones del sistema necesarios.
- **Recomendación 2**: Implementar un proceso de backup y establecer políticas claras de respaldo y custodia de datos.

- **Recomendación 3**: Evaluar los datos sensibles para encriptar y que datos pueden quedar públicos.

- **Recomendación 4**: Desarrollar un plan para actualizar los controles y revisar el las versiones de los paquetes utilizados.

2. Plan Detallado para Realizar un Presupuesto para un Sistema de Información

2.1. Definición de los Requisitos del Sistema

Actividades necesarias:

- **Reuniones con stakeholders**: Participantes: Gerentes, usuarios finales (administradores y choferes), equipo técnico. Objetivo: Definir expectativas y necesidades del sistema. Frecuencia: 3-5 reuniones de planificación.
- **Análisis de requisitos funcionales**: Documentar todas las funciones que el sistema debe realizar. Login, ingreso de datos, incidentes, gestión de datos y encriptación de datos
- **Análisis de requisitos no funcionales**:
    - Escalabilidad: Posibilidad de integrar futuras funcionalidades.
    - Usabilidad: Diseño accesible para usuarios no técnicos.
    - Seguridad: Asegurar la integridad de los datos mediante técnicas de encriptación.
    - Rendimiento: Optimizar para que las respuestas sean rápidas y los tiempos de carga sean mínimos al acceder y manipular registros.
    - Interoperabilidad: Habilitar la capacidad de la aplicación para integrarse con APIs externas.

Presupuesto asociado:

	Costo Estimado	Observaciones
Reuniones y consultoría	2000$ - 5000$	Incluye horas de analistas y documentación
Documentación de Requisitos	1500$ - 3000$	Elaboración de especificaciones detalladas

2.2. Identificación de Componentes del Sistema

Componentes esenciales a identificar y presupuestar:

2.2.1 Desarrollo del Software

El desarrollo del software incluye el diseño, la programación, las pruebas y la implementación del sistema. Esto requiere un equipo adecuado de desarrollo y herramientas especializadas.

Actividades y costos:

- **Diseño del sistema:**
    - Arquitectura de software, diseño de base de datos (Azure SQL Server), interfaces de usuario (HTML, CSS, JavaScript).
    - Costo estimado: 5,000$ − 10,000$$.

- **Desarrollo**:
    - Equipo de desarrollo (4 programadores en C#, JavaScript, APIs).
    - Costo estimado: 54,000$ − 81,000$ (4 meses de trabajo).

- **Pruebas de desarrollo:**
    - Pruebas funcionales, de seguridad y rendimiento.
    - Costo estimado: 8,000−15,000.

- **Revisión de código y documentación:**
    - Costo estimado: 3,000−6,000.

2.2.2 Licencias de Software y Herramientas

Licencias	**Costo Estimado**	**Observaciones**
Licencia Azure SQL Server	2,000$−5,000$ anual	Depende del tamaño de la base de datos.
Herramientas de desarrollo (Visual Studio, Postman)	1,000$−3,000$	Licencias para el equipo.
Herramientas de testing (JIRA, Selenium)	2,000$−5,000$	Para gestión de pruebas.

2.2.3 Infraestructura Tecnológica

Concepto	**Costo Estimado**	**Observaciones**
Servidores en la nube (Azure)	10,000$ − 20,000$ anual	Configuración escalable según demanda.

Redes y Firewall	3,000$ − 8,000$	Seguridad básica y configuración de red.

2.2.4 Seguridad del Sistema

Concepto	**Costo Estimado**	**Observaciones**
Encriptación avanzada (SSL, cifrado de datos)	2,000$−5,000$	Implementación de seguridad adicional.
Auditoría de seguridad externa	5,000$−10,000$	Evaluación de vulnerabilidades.

2.3. Estimación de Costos Detallados

Ejemplo de presupuesto detallado:

Categoría	**Descripción**	**Costo Estimado**	**Observaciones**
Desarrollo de Software	Diseño, programación, pruebas	$100,000	Equipo de 4 desarrolladores (4 meses).
Licencias de Software	Azure SQL, herramientas de desarrollo	$10,000	Incluye IDEs y testing.
Infraestructura	Servidores en Azure, redes	$30,000	Configuración escalable.

Categoría	Descripción	Costo Estimado	Observaciones
Seguridad	Encriptación, auditorías	$12,000	Protección de datos sensibles.
Capacitación	Entrenamiento para usuarios	$8,000	Sesiones para administradores y choferes.
Mantenimiento Anual	Soporte y actualizaciones	$25,000	15-20% del costo total.
Contingencia (10%)	Imprevistos	$23,500	Ajustes por cambios no planificados.
Total Estimado	**Suma de todos los componentes**	**$258,500**	**Depende de complejidad y alcance.**

2.4. Cronograma de Implementación y Desembolso

Fase de Planificación (1 mes):

- Reuniones de definición de requisitos.
- Documentación de especificaciones.
- Estimación de costos.

Fase de Desarrollo (2-3 meses):

- Diseño del sistema.
- Programación.
- Pruebas de integración.
- Revisión del código.

Fase de Implementación (1 semana):

- Instalación y configuración de servidores.

- Despliegue del sistema en producción.
- Capacitación a usuarios finales.

Fase de Mantenimiento (Continuo):

- Soporte técnico.
- Actualizaciones periódicas.

Cronograma de pagos:

- Fase de planificación: 5% del total.
- Fase de desarrollo: 60% del total.
- Fase de implementación: 10% del total.
- Fase de mantenimiento anual: 25% del total.

2.5. Revisión y Aprobación del Presupuesto

- Revisión interna: Presentación al equipo de TI y finanzas para ajustes.
- Aprobación final: Validación por parte de la gerencia antes de ejecución.

Recomendaciones clave:

1. Priorizar la implementación de backups automáticos (estimado: $5,000 adicionales).
2. Incluir un fondo de contingencia para posibles retrasos.

3. Plan Detallado para Medir los Riesgos de un Sistema de Información

3.1. Definición del Alcance y los Objetivos del Plan de Gestión de Riesgos

1.1 Definición del Alcance

El alcance del plan de riesgos abarca todos los componentes críticos del sistema. como por ejemplo:

Actividades específicas:

- **Identificar componentes del sistema**:
    - El sistema P-Logistics Rutas incluye módulos de crear usuarios administradores, choferes, rutas e incidentes, además incluye bases de datos SQL e incluirá servidores.
- **Establecer límites**:
    - El sistema cuenta con un módulo web y un módulo móvil, ambos módulos no están en producción como tal, pero para la gestión de riesgos solo contamos con el módulo web.

3.1.2 Establecimiento de Objetivos

El objetivo del plan es identificar, evaluar, mitigar y monitorear riesgos que puedan afectar al sistema. Los objetivos deben alinearse con las metas de negocio y asegurar que se protejan los activos más críticos.

Actividades específicas:

- **Proteger datos sensibles:** Garantizar la confidencialidad de contraseñas y datos de rutas.
- **Asegurar disponibilidad:** Minimizar los tiempos de inactividad del sistema.

- **Estándares normativos:** Verificar que se sigan las reglas con los estándares de seguridad básica.

3.2. Identificación de Riesgos

3.2.1 Fuentes de Riesgos

En esta etapa, el objetivo es identificar todos los posibles riesgos que pueden afectar al sistema. Existen diversas fuentes de riesgos, que se deben evaluar de manera exhaustiva.

Actividades específicas:

- **Evaluación de procesos críticos**: Analizar los procesos de negocio asociados al sistema.
    - El sistema está conectado a un app móvil, por lo tanto la interrupción del app web con el app móvil y viceversa, podría generar un riesgo de crítico para su operación.
- **Análisis de amenazas externas**:
    - Phishing, DDOS, Inyección SQL, Intercepción de datos y acceso no autorizado son los riesgos más críticos que podrían afectar la seguridad del sistema.
- **Análisis de amenazas internas**:
    - Errores humanos, malas prácticas de desarrollo, malas prácticas de base de datos, entrega de datos confidenciales por parte de un empleado y robo o pérdida de dispositivos de choferes.
- **Condiciones externas**:
    - Desactualización de la plataforma en la nube de base de datos y desactualización de los lenguajes o herramientas utilizadas.

Tipos comunes de riesgos:

1. **Riesgos Tecnológicos**:
    - **Fallas en Azure SQL Server:** que se puedan llegar a corromper los datos por falta de backups.
    - **Desactualización de APIs**: Que se lleguen a volver vulnerables las APIs por falta de mantenimiento.
2. **Riesgos de Seguridad**:
    - **Acceso no autorizado**: Roles mal configurados.

- **Ataques cibernéticos**: Phishing, malware, inyecciones SQL.
- **Pérdida de datos**: Por corrupción de archivos o errores humanos.

3. **Riesgos Operacionales**:
   - **Errores humanos**: Como cargar datos incorrectos en el sistema.
   - **Fallas en procesos de negocio**: Pérdida de datos por falta de backups.
4. **Riesgos Regulatorios**:
   - **Cumplimiento de normativas**: Leyes de privacidad (como GDPR), normativas de seguridad de datos.
   - **Requisitos industriales**: Cumplimiento con estándares como ISO 27001 para la seguridad de la información.

3.2.2 Herramientas para Identificar Riesgos:

Utilizar diferentes **herramientas de gestión de riesgos** como:

- **Brainstorming:** Con el equipo de desarrollo y usuarios del sistema(choferes y administradores).
- **Analisis FODA:**
  - **Debilidad:** Falta de monitoreo en producción.
  - **Amenaza:** Ataque a APIs desactualizadas.
  - **Oportunidad:** Escalabilidad gracias a Azure SQL server.
  - **Fortalezas:** Interfaz amigable y accesible.

3.3. Evaluación de Riesgos

3.3.1 Evaluación de Probabilidad e Impacto

Matriz de Riesgo

Riesgo	Probabilidad (Alta/Media/Baja)	Impacto (Alto/Medio/Bajo)	Nivel de Riesgo (Crítico/Moderado/Bajo)	Medidas de Mitigación
Desconocimiento del patrón SOLID	Media	Alto	Crítico	Capacitación en principios SOLID y buenas prácticas de desarrollo.
Aplicación inadecuada de SQL Azure	Alta	Alto	Crítico	Optimización de consultas, tuning de base de datos y revisión de configuraciones de seguridad.
Gestión deficiente de credenciales	Medio	Medio	Medio	Implementación de MFA, rotación de credenciales, uso de un gestor de contraseñas seguro.
Vulnerabilidad en las API	Alta	Alto	Crítico	Pruebas de seguridad, uso de autenticación segura y validación de entrada.
Vulnerabilidad web	Alta	Alto	Crítico	Implementación de OWASP Top 10, pruebas de seguridad y validaciones en el backend.

Fallas en el servidor	Media	Alto	Crítico	Uso de balanceo de carga, monitoreo de servidores y redundancia de hardware.
Actualización de versiones	Media	Medio	Moderado	Plan de actualización continua, pruebas en entorno de staging antes del despliegue.
Falta de encriptación de datos sensibles	Alta	Alto	Crítico	Implementación de encriptación AES-256 en datos en reposo y TLS en transmisión.
Falta de planeamiento de escalabilidad	Media	Alto	Crítico	Diseño de arquitectura escalable desde el inicio, uso de cloud computing.
Falta de historial de Logs en BD	Media	Medio	Moderado	Implementación de auditoría de logs, uso de SIEM para monitoreo de eventos.
Pérdida de datos	Media	Alto	Crítico	Implementación de backups automáticos, redundancia y replicación de datos.

Capacidad de escalabilidad	Media	Medio	Moderado	Uso de infraestructura elástica, monitoreo de uso de recursos.
Falta de presupuesto	Alta	Medio	Moderado	Planificación financiera, optimización de costos y priorización de recursos críticos.
Falta de concurrencia	Media	Medio	Moderado	Implementación de patrones de concurrencia
Falta de hardware	Media	Medio	Moderado	Planificación de recursos, monitoreo de rendimiento y escalamiento según demanda.
Fallas mecánicas	Baja	Alto	Moderado	Mantenimiento preventivo y redundancia de hardware crítico.
Riesgo de rutas obsoletas	Bajo	Medio	Moderado	Refactorización periódica del código, documentación y pruebas continuas.

3.3.2 Herramientas de Evaluación de Riesgos:

- **Matriz de Riesgos**: Para categorizar y priorizar los riesgos.
- **Análisis de impacto en el negocio (BIA)**: Por la falta de backups y actualizaciones el sistema puede llegar a representar un problema financiero en pérdidas monetarias.

3.4. Planificación de Respuestas a los Riesgos

3.4.1 Estrategias de Respuesta

Una vez que se han evaluado los riesgos, es necesario definir cómo **responder** a cada uno de ellos de manera eficaz.

Respuestas típicas a los riesgos:

1. **Evitar**: Actualizar APIs y librerías.
2. **Mitigar**: Implementar backups automáticos.
3. **Transferir**: Contratar auditoría externa de seguridad.
4. **Aceptar**: Monitorear roles de usuario con revisiones trimestrales.

3.4.2 Planes de Contingencia

- **Recuperación de datos:** Restaurar desde backups (si se implementan).
- **Incidentes de seguridad:** Protocolo de revocación de accesos y notificación a usuarios.

3.5. Monitoreo y Control de Riesgos

3.5.1 Herramientas de Monitoreo:

- **Sistema de monitoreo de infraestructura**: Actualmente, el desarrollo de la infraestructura no cuenta con un sistema de monitoreo integral que permita supervisar en tiempo real el estado del servidor y de la red. Para ello, se recomienda implementar una solución básica que permita, alertas tempranas configurar notificaciones básicas para detectar caídas o

anomalías significativas, métricas en tiempo real para monitorear parámetros fundamentales como el estado de la CPU, memoria, tráfico de red y disponibilidad de servicios esenciales.

- **Auditorías de seguridad**: Es importante realizar una evaluación exhaustiva de estas métricas con el fin de encontrar o modelar la herramienta especializada de monitoreo que más se adapte y permita detectar puntos de mejora.

3.5.2 Revisiones Periódicas:

Programar **revisiones periódicas:** La implementación de revisiones periódicas del plan de riesgos es fundamental para la evolución continua de la aplicación web y móvil. Este proceso no solo permite la actualización y la optimización de las medidas de protección, sino que también garantiza una respuesta ágil ante nuevos desafíos. La propuesta incluye la planificación de revisiones programadas, un enfoque integral en áreas críticas y el uso de herramientas específicas que faciliten la recopilación y análisis de datos. Esto asegurará que, a medida que la infraestructura y el entorno evolucionan, las estrategias de mitigación se ajusten de manera dinámica y efectiva.

3.6. Documentación y Comunicación de Riesgos

3.6.1 Registro de Riesgos:

ID	Riesgo Identificado	Probabilidad	Impacto	Nivel de Riesgo	Respuesta Planeada	Responsable	Estado Actual
R001	Fallas en la conexión con	Alta	Alta	Crítico	Implementar reintentos automáticos	Equipo de Backend	En Curso

	la base de datos.				y logs de error.		
R002	Pérdida de datos	Media	Alta	Medio	Implementar backups automáticos y control de versiones.	Equipo de Backend	En Curso
R003	Escalabilidad insuficiente ante aumento de usuarios	Media	Media	Medio	Diseñar arquitectura modular y utilizar servicios escalables de Azure.	Arquitecto de Software	Planificado
R004	Baja adopción del sistema por parte de choferes	Baja	Media	Medio	Capacitación y diseño de interfaz móvil sencilla.	Equipo de UX	En análisis
R005	Incompatibilidad entre versiones	Media	Media	Bajo	Garantizar soporte para	QA	En análisis

					navegadores y móviles		
R006	Vulnerabilidad en las API	Alta	Alto	Crítico	Implementar pruebas de seguridad, uso de autenticación y validación de entrada.	Equipo de Backend	En análisis
R007	Fallas en el servidor	Media	Alta	Crítico	Uso de un balanceador de cargas y monitoreo de servidores.	Infraestructura / DevOps	En proceso
R008	Actualización de versiones	Media	Media	Moderado	Plan de actualización continua y pruebas en entorno de desarrollo antes del despliegue.	Equipo de Backend / Frontend	Planificado

3.6.2 Reportes Regulares:

- Se conjunta un informe que contiene un análisis de datos del monitoreo y posibles fallos que tenga el desarrollo con sus soluciones.

3.7. Revisión y Mejora Continua

El plan de riesgos debe ser **dinámico** y adaptarse a los cambios en el sistema y en el entorno operativo. Es esencial una **mejora continua** basada en las lecciones aprendidas.

Actividades:

- **Implementar las lecciones aprendidas:** Incorporar los hallazgos de las pruebas (creación de backups, actualizar APIs y encriptar más datos).
- **Actualizar plan:** Basado en los nuevos riesgos identificados durante la revisión.

3.8 Conclusión

3.8.1 Resumen de las Pruebas Realizadas

- **Pruebas realizadas:** 60% de los controles aprobados. Áreas críticas en seguridad y backups
- **Condición del Sistema**: Funcional pero requiere mejoras urgentes.

3.8.2 Acciones a Tomar

- **Acciones Inmediatas**: Implementar backups y actualizar APIs.
- **Acciones a largo plazo**: Ejecutar plan de mantenimiento y monitoreo, los cuales van a servir de guía para las acciones a tomar.

3.9. Anexos

- **Anexo 1**: Documento de prueba realizada (Adjuntada)