Plan de Contingencia - Proyecto Webinnova

- **Fallos en el sistema o servidores**

Riesgo: Este riesgo se refiere a la posibilidad de que los servidores que alojan la aplicación Webinnova experimenten problemas que provoquen una caída del servicio. Esto puede ser causado por sobrecargas, fallos en el hardware, interrupciones en el servicio de internet, o problemas de software en el servidor.

Impacto: Si el sistema se cae, los docentes, estudiantes y administradores no podrán acceder a las funcionalidades de la plataforma, como la gestión de salones, el acceso a listas de alumnos, y la comunicación entre los diferentes actores. Esto interrumpiría el flujo de trabajo y la operativa diaria del proyecto.

Estrategia de mitigación: Sistemas de monitoreo: Implementar herramientas de monitoreo en tiempo real como Nagios o Zabbix, que alerten de posibles caídas o problemas de rendimiento del servidor antes de que seconviertan en un fallo crítico. Estas herramientas envían notificaciones automáticas cuando detectan un fallo inminente.

Servidor de respaldo (backup): Tener configurado un servidor secundario o una arquitectura en la nube (AWS, Azure, drive, etc.) que pueda asumir la carga si el servidor principal falla. Esto minimiza el tiempo de inactividad.

Infraestructura escalable: Asegurar que el sistema pueda escalar de manera automática ante aumentos inesperados en el tráfico, usando tecnologías como balanceadores de carga y servidores en la nube que se escalan automáticamente según la demanda.

Plan de respuesta: Notificación temprana a los usuarios: En cuanto se detecta una caída del sistema, enviar notificaciones por correo electrónico o a través de las

redes sociales para informar a los usuarios del problema y dar una estimación del tiempo de resolución.

Activar servidor de respaldo: Si el servidor principal no puede ser restaurado en un tiempo razonable (por ejemplo, en menos de 30 minutos), activar el servidor de respaldo para garantizar que el servicio se mantenga operativo mientras se soluciona el problema original.

Equipo de Desarrollo en alerta: Mantener un equipo de soporte técnico disponible las 24 horas para responder a fallos de servidores críticos. Este equipo debe estar entrenado en la recuperación de servidores y la resolución de problemas relacionados con el hardware o el software del sistema.

- **Errores en la programación o funcionalidad**

Riesgo: Este riesgo se refiere a la posibilidad de que el código que sustenta Webinnova tenga errores (bugs) que afecten su funcionalidad. Esto puede incluir problemas en el registro de nuevos usuarios, fallos en la asignación de salones, errores al mostrar datos, o cuelgues del sistema en situacionescríticas.

Impacto: Los errores en la programación pueden causar que ciertas funcionalidades no estén disponibles o que se generen comportamientos inesperados en la plataforma. Esto podría frustrar a los usuarios,reducir su confianza en el sistema, y afectar negativamente el rendimiento de la aplicación, especialmente si son funciones clave como la gestión de alumnos y profesores.

Estrategia de mitigación: Pruebas exhaustivas: Implementar un riguroso proceso de pruebas que incluya pruebasunitarias, pruebas de integración, y pruebas de

aceptación del usuario. Estas pruebas deben hacerse antes decada despliegue, usando datos reales y simulaciones de uso intensivo.

Sistema de seguimiento de errores: Utilizar herramientas como Jira, Bugzilla o GitHub Issuespara gestionar, rastrear y priorizar errores reportados. Esto facilita la asignación de tareas a los desarrolladores y el seguimiento del estado de los errores desde su detección hasta su resolución.

Revisión por pares (Code Review): Antes de implementar cambios importantes en el sistema,establecer un proceso formal de revisión de código por otros desarrolladores para detectar errorespotenciales y mejorar la calidad del código.

Plan de respuesta: Equipo de desarrollo en alerta: Mantener un equipo de desarrolladores listo para actuarrápidamente en caso de que se descubran errores críticos. Este equipo debe ser capaz de analizar los logs del sistema para identificar la causa del problema y desplegar parches correctivos.
Interfaz de reporte de errores: Incluir en la plataforma una funcionalidad que permita a losusuarios reportar errores directamente desde la interfaz. Estos reportes deben ir acompañados de capturas de pantalla o registros para facilitar la reproducción del error.

Parches de emergencia: En caso de un error crítico que afecte la funcionalidad básica delsistema, debe existir un protocolo para aplicar parches de emergencia lo antes posible, con la posibilidad de hacer un despliegue rápido en producción (hotfix).

- **Pérdida o corrupción de datos**

Riesgo: La posibilidad de que los datos almacenados en la base de datos de Webinnova se corrompan, se pierdan o se alteren de forma no intencionada, ya sea por fallos en la base de datos, ataques demalware, o errores humanos durante la manipulación de los datos.

Impacto: La pérdida de datos podría implicar la desaparición de registros de alumnos,información de docentes, o configuraciones de los salones, lo que afectaría

gravemente la operativa del sistema y la confiabilidad de la plataforma. Además, la recuperación de los datos podría llevar tiempo,afectando la experiencia de los usuarios.

Estrategia de mitigación: Copias de seguridad automáticas: Configurar un sistema de copias de seguridad automáticasde la base de datos que se realice al menos una vez al día. Las copias deben almacenarse en un entornoseguro y estar accesibles en caso de emergencia. Además, se deben realizar pruebas periódicas derestauración para garantizar que las copias son funcionales.
Control de versiones: Implementar un sistema de control de versiones en la base de datos que permita revertir a estados anteriores en caso de que los datos actuales estén corruptos. Esto se puede hacer utilizando técnicas como snapshots de la base de datos o sistemas de logueo detransacciones.

Encriptación de datos: Proteger los datos sensibles de los usuarios mediante encriptación en la base de datos, lo que asegurará que incluso si se produce una pérdida de datos, estos no puedan ser utilizados de manera indebida por atacantes.

Plan de respuesta: Restauración de la copia de seguridad: En caso de pérdida de datos, el equipo de desarrollo debe estar preparado para restaurar los datos desde la última copia de seguridad disponible. Este proceso debe ser rápido y eficiente para minimizar el tiempo de inactividad.
Verificación de la integridad: Después de la restauración, el equipo debe verificar que todos los datos hayan sido restaurados correctamente y no se haya perdido información crítica.

Investigación de la causa: Si se detecta una corrupción de datos, es necesario realizar una investigación exhaustiva para identificar la causa del fallo, sea un error en el sistema, un ataque o un error humano.

- **Problemas de seguridad**

Riesgo: La posibilidad de que vulnerabilidades en el sistema permitan ataques cibernéticos, como el robo de información confidencial de los usuarios (alumnos, docentes, administradores) o la modificación maliciosa de los datos.

Impacto: Un ataque exitoso podría comprometer información sensible, como datos personales o de rendimiento de los alumnos, y afectar la integridad del sistema. Además, un incidente de seguridad puede dañar la reputación del proyecto, generando desconfianza en los usuarios.

Estrategia de mitigación: Autenticación multifactor (MFA): Implementar autenticación multifactor para el acceso a cuentas de administradores y usuarios de alto nivel. Esto reduce significativamente el riesgo de acceso no autorizado incluso si las contraseñas son comprometidas.

Actualizaciones de seguridad regulares: Mantener todos los componentes del sistema (frameworks, librerías, servidores) actualizados con los últimos parches de seguridad para minimizar el riesgo de explotación de vulnerabilidades conocidas.

Pruebas de penetración: Realizar auditorías de seguridad y pruebas de penetración periódicas para identificar posibles vulnerabilidades antes de que sean explotadas por atacantes. Estas pruebas pueden ser realizadas por equipos especializados o servicios externos.

Plan de respuesta: Bloqueo de cuentas comprometidas: En caso de detectar actividad sospechosa o un ataque en curso, bloquear de inmediato las cuentas afectadas para prevenir un mayor acceso no autorizado.

Notificación a los usuarios: Informar de manera rápida y transparente a los usuarios afectados sobre el incidente, detallando qué datos pudieron haber sido comprometidos y qué medidas se están tomando para solucionar el problema.

Colaboración con expertos en ciberseguridad: Si ocurre un incidente, es importante contar con un equipo de ciberseguridad para investigar el ataque, determinar cómo ocurrió, y aplicar las medidas correctivas necesarias para cerrar las vulnerabilidades.

- **Retrasos en el desarrollo o despliegue**

Riesgo: Posibilidad de que se produzcan retrasos en la entrega de funcionalidades o en la implementación de nuevas actualizaciones debido a problemas técnicos, falta de recursos, o complicaciones no previstas en el desarrollo.

Impacto: Un retraso en el desarrollo podría afectar negativamente la capacidad de cumplir con los plazos prometidos a los usuarios o al equipo de gestión del proyecto, lo que impacta la confianza de los stakeholders y puede llevar a la acumulación de trabajo pendiente.

Estrategia de mitigación: Cronograma detallado: Elaborar un cronograma detallado de desarrollo y despliegue que incluya tiempos de buffer para gestionar los imprevistos. Este cronograma debe ser realista y permitir ajustes sin comprometer los plazos finales.

Metodología ágil (Scrum): Implementar una metodología ágil para el desarrollo, dividiendo el trabajo en sprints cortos con entregas frecuentes de pequeños avances.

Esto permite adaptarse rápidamente a los cambios de prioridades y mitigar el riesgo de retrasos acumulados.

Asignación adecuada de recursos: Garantizar que cada tarea tiene los recursos humanos y técnicos necesarios para ser completada en el plazo previsto. Esto incluye evitar la sobrecarga de trabajo en los desarrolladores, lo que puede llevar a errores y retrasos.

Plan de respuesta: Priorización de funcionalidades: En caso de que un retraso sea inevitable, priorizar el desarrollo de las funcionalidades más críticas y posponer las de menor importancia. Esto asegura que lo más necesario esté disponible a tiempo.

Comunicación clara con los stakeholders: Informar de manera proactiva a los stakeholders sobre cualquier retraso y proporcionar un nuevo cronograma ajustado. La comunicación clara es clave para evitar malentendidos y mantener la confianza en el proyecto.

Redistribución de recursos: Si un área del proyecto se está retrasando significativamente, reasignar recursos desde otras áreas para equilibrar la carga de trabajo y garantizar que las tareas más críticas se completen a tiempo.