Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Planeación de sistemas de software

Gpo 105

Plan de Riesgos

Diego Dávila Hernández A01285584

Fernando Morán Fougerat A01284623

Imanol Armando González Solís A00835759

Ramiro Alejandro Garza Villarreal A01178167

Rogelio Garza Rendón A01571384

Campus Monterrey

3 may 2025

ÍNDICE

1. Introducción

La gestión de riesgos es esencial para el éxito de cualquier proyecto de desarrollo de software. Un plan de riesgos bien estructurado nos permite anticipar y mitigar posibles problemas. A continuación, presentamos un plan de riesgos adaptado a un proyecto de desarrollo de software.

2. Identificación de Riesgos

Al momento de analizar los riesgos, usaremos las historias de usuario y exploramos posibles riesgos en cada etapa de desarrollo. Tomaremos en cuenta factores técnicos, como la estabilidad, o la compatibilidad de tecnologías, así como los sistemas operativos, como la disponibilidad de servicios externos y la integridad de los datos. Estaremos tomando en cuenta el impacto en la experiencia de usuario, asegurando que la plataforma sea accesible y funcional. A través de lluvia de ideas, revisión continua y búsqueda exhaustiva, identificaremos posibles riesgos, y definiremos estrategias para mitigarlos antes de que tengan un impacto en el proyecto.

1. **RI-001** El LLM que utilizamos se equivoca al modificar un problema
2. **RI-002** Alguna dependencia no es compatible con las demás tecnologías que utilizamos
3. **RI-003** Inventario de la tienda inexacto
4. **RI-004** Los usuarios no se sienten cómodos con la dificultad de los problemas
5. **RI-005** Sobrecarga de la plataforma por falta de recursos para mejorar el servidor
6. **RI-006** Algún servicio externo se cae durante la presentación al socio
7. **RI-007** Error de precisión en alguna de las tags de un problema
8. **RI-008** Cantidad de dinero digital inexacto
9. **RI-009** Se presenta algún bug en la tienda que impida a los usuarios redimir sus recompensas
10. **RI-010** Corrupción de la base de datos debido a un error en el sistema de respaldo, afectando a todos los usuarios
11. **RI-011** Los problemas de programación no se refrescan correctamente, mostrando desafíos obsoletos o eliminados

12. **RI-012** Se elimina accidentalmente información en la base de datos
13. **RI-013** Cierre forzado de la plataforma por problemas legales o de derechos de autor
14. **RI-014** Desincronización con Judge0, causando que el código enviado no se evalúe correctamente o tome demasiado tiempo
15. **RI-015** Ataque cibernético que tumbe el proyecto por completo
16. **RI-016** Vulnerabilidades en dependencias de código.

3. Evaluación de Riesgos

Análisis de riesgos

Clave	Probabilidad	Impacto	Prioridad
RI-001	Muy Probable (5)	Insignificante (1)	5
RI-002	Muy Probable (5)	Moderada (3)	15
RI-003	Probable (4)	Menor (2)	8
RI-004	Probable (4)	Moderada (3)	12
RI-005	Probable (4)	Importante (4)	16
RI-006	Probable (4)	Catastrófica (5)	20
RI-007	Posible (3)	Insignificante (1)	3
RI-008	Posible (3)	Moderada (3)	9
RI-009	Posible (3)	Importante (4)	12
RI-010	Posible (3)	Catastrófica (5)	15
RI-011	No es probable (2)	Moderada (3)	6
RI-012	No es probable (2)	Importante (4)	8
RI-013	No es probable (2)	Catastrófica (5)	10
RI-014	Muy improbable (1)	Importante (4)	4
RI-015	Muy improbable (1)	Catastrófica (5)	5
RI-016	Posible (2)	Menor (3)	6

Riesgo Bajo (Verde)

- RI-001
- RI-007
- RI-011
- RI-014
- RI-015
- RI-016

Riesgo Medio (Amarillo)

- RI-003
- RI-004
- RI-008
- RI-009
- RI-012
- RI-013

Riesgo Alto (Rojo)

- RI-002
- RI-005
- RI-006
- RI-010

	1 Insignificante	2 Menor	3 Moderada	4 Importante	5 Catastrófica
5 **Muy Probable**	5 El LLM que utilizamos se equivoca al modificar un problema	10	15 Alguna dependencia no es compatible con las demás tecnologías que utilizamos	20	25
4 **Probable**	4	8 Inventario de la tienda inexacto	12 Los usuarios no se sienten cómodos con la dificultad de los problemas	16 Sobrecarga de la plataforma por falta de recursos para mejorar el servidor	20 Algún servicio externo se cae durante la presentación al socio
3 **Posible**	3 Error de presición en alguna de las tags de un problema	6 Vulnerabilidades en dependencias de código	9 Cantidad de dinero digital inexacto	12 Se presenta algún bug en la tienda que hace que impide a los usuarios redimir sus recompenzas	15 Corrupción de la base de datos debido a un error en el sistema de respaldo, afectando a todos los usuarios.
2 **No es probable**	2	4	6 Los problemas de programación no se refrescan correctamente, mostrando desafíos obsoletos o eliminados	8 Se elimina accidentalmente información en la base de datos	10 Cierre forzado de la plataforma por problemas legales o de derechos de autor.
1 **Muy improbable**	1	2	3	4 Desincronización con Judge0, causando que el código enviado no se evalúe correctamente o tome demasiado tiempo.	5 Ataque cibernético que tumbe el proyecto por completo

4. Planificación de Estrategias de Mitigación

A continuación se presentan planes de acción para cada riesgo identificado:

1. **El LLM que utilizamos se equivoca al modificar un problema**
    - Limitar el alcance de la LLM a tareas muy específicas para que solamente se modifique la redacción de los problemas.
    - Hacer una revisión manual de algunos problemas para verificar que la mayoría de los problemas estén redactados correctamente.

2. **Alguna dependencia no es compatible con las demás tecnologías que utilizamos**
    - Realizar pruebas de integración antes de la implementación de cada tecnología.

- Tener un plan de contingencia para reemplazar dependencias que puedan causar problemas.

3. **Inventario de la tienda inexacto**
    - Realizar revisiones rutinarias del inventario para verificar discrepancias.
    - Utilizar un sistema de queues para manejar actualizaciones concurrentes.

4. **Los usuarios no se sienten cómodos con la dificultad de los problemas**
    - Permitir a los usuarios filtrar los problemas por dificultad.
    - Recibir feedback constante de los usuarios para ajustar la dificultad de los problemas.

5. **Sobrecarga de la plataforma por falta de recursos para mejorar el servidor**
    - Optimizar el código y las consultas a la base de datos para reducir la carga
    - Realizar pruebas de estrés periódicamente para identificar problemas

6. **Algún servicio externo se cae durante la presentación al socio**
    - Tener un respaldo de recursos audiovisuales disponibles, ya sean capturas de pantalla o vídeos.

7. **Error de precisión en alguna de las tags de un problema**
    - Permitir a los usuarios reportar tags incorrectas para cambiarlas rápidamente
    - Realizar revisiones manuales periódicas de las tangas para verificar que sean correctas

8. **Cantidad de dinero digital inexacto**
    - Utilizar un sistema de respaldo para restaurar saldos en caso de errores

9. **Se presenta algún bug en la tienda que hace que impida a los usuarios redimir sus recompensas**
    - Que el equipo de soporte esté preparado para resolver problemas rápidamente
    - Ofrecer recompensas a los usuarios que se les presente un bug

10. **Corrupción de la base de datos debido a un error en el sistema de respaldo, afectando a todos los usuarios**
    - Tener un respaldo de la base de datos para poder restaurarlos en caso de un incidente.
    - Monitorear periódicamente la base de datos para detectar errores.

11. **Los problemas de programación no se refrescan correctamente, mostrando desafíos obsoletos o eliminados**
    - Implementar un sistema de caché con invalidación automática para problemas actualizados.
    - Permitir a los usuarios reportar problemas obsoletos

12. **Se elimina accidentalmente información en la base de datos**
    - Contar con un respaldo de la base de datos
    - Utilizar un sistema de papelera de reciclaje

13. **Cierre forzado de la plataforma por problemas legales o de derechos de autor**
    - Contar con un equipo legal para garantizar el cumplimiento de todas las normas.
    - Asegurar que todos los contenidos sean originales o que cuenten con las licencias adecuadas.
14. **Desincronización con Judge0, causando que el código enviado no se evalúe correctamente o tome demasiado tiempo**
    - Tener un servicio alternativo de evaluación de código como respaldo.
    - Implementar un sistema de reintentos automáticos para las evaluaciones fallidas.

15. **Ataque cibernético que tumbe el proyecto por completo**
    - Implementar medidas de seguridad robustas, como encriptación y autenticación de dos factores.
    - Tener un plan de respuesta a incidentes.

16. **Vulnerabilidad en dependencias de código.**
    - Uso de dependencias confiables y actualización constante.
    - Utilizar lock files para utilizar versiones específicas y probadas de las dependencias.

5. Establecimiento de Medidas de Control

A continuación se define cómo se monitorearán y controlarán los riesgos durante la ejecución del proyecto:

- **Seguimiento continuo:** Realizaremos monitoreo periódico de los sistemas y procesos clave para detectar señales tempranas de posibles problemas durante 2 meses.
- **Registro y revisión de logs y auditorías**: Se implementarán auditorías de seguridad y análisis de logs para identificar anomalías en el sistema.
- **Pruebas regulares:** Realizaremos pruebas de estrés, integración y unidad para evaluar la estabilidad del software y prevenir errores.
- **Mantenimiento de dependencias:** Se revisarán y actualizarán regularmente las dependencias del sistema para minimizar vulnerabilidades.
- **Evaluación de rendimiento para escalabilidad:** Se realizarán análisis de carga y rendimiento en la plataforma para evitar problemas de escalabilidad.
- **Estrategia de mitigación de incidentes**: Se definirá planes de acciones para responder ante cualquier incidente.

6. Comunicación y Colaboración

Durante el proceso del proyecto, debemos tener una comunicación abierta con el equipo y las partes interesada, para eso, implementaremos las siguientes estrategias para poder abordar cualquier riesgo de manera efectiva:

- **Comunicación con socio formador:** Estaremos en contacto con el socio durante todo el desarrollo del proyecto, notificando el estado del proyecto, los posibles riesgos y discutiremos soluciones según sus necesidades.
- **Documentación**: Estaremos documentando todos los detalles sobre los riesgos, planes de mitigación, y procedimientos de respuesta.
- **Feedback:** Se fomentará la retroalimentación del equipo para mejorar la identificación y gestión de riesgos.

7. Conclusión

En este documento identificamos, evaluamos y creamos planes de mitigación para los riesgos que pueden afectar el desarrollo de nuestro proyecto para Code Courses. Entre los riesgos más críticos destacan la caída de servicios externos durante la presentación (RI-006), la corrupción de la base de datos (RI-010) y la sobrecarga del servidor (RI-005). Para evitar y poder abordar estos riesgos, deberemos seguir los planes de acción que hemos desarrollado.

El proyecto presenta un nivel de riesgo moderado,sin embargo una buena ejecución de nuestras tareas y seguimiento a nuestros estándares y planes de mitigación propuestas reducirán de manera efectiva su impacto. Un desarrollo proactivo en la prevención y control de riesgos garantizará la estabilidad y seguridad de nuestro sistema, minimizando interrupciones y ofreciendo una plataforma confiable para los usuarios de Tech Mahindra.