**Desarrollo Back-End y Front-End**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tipo** | **Riesgo** | **Probabilidad** | | | | | **Impacto** | | | | | **Medidas de Mitigación** |
| 1 | Errores en validaciones del lado cliente | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Implementar validación tanto en front como back-end |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 2 | Código duplicado o poco reutilizable | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Uso de funciones/módulos reutilizables y revisión por pares |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 3 | Bajo rendimiento del front-end | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Lazy loading, optimización de assets y uso de frameworks eficientes |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 4 | Bugs persistentes en funciones críticas | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Pruebas unitarias y pruebas funcionales desde el inicio |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 5 | Incompatibilidades entre navegadores | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Pruebas multiplataforma y uso de librerías compatible |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 6 | Fugas de memoria por mal manejo de estados | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | |  | | --- | | Revisión técnica con herramientas como Chrome DevTools |  |  | | --- | |  | |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 7 | Accesos concurrentes mal manejados | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Gestión adecuada de concurrencia con locks y semáforos |
| Cuantitativo |  |  |  |  |  |  |  |  |  |  |
| 8 | APIs sin manejo de errores adecuado | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Diseño robusto de manejo de excepciones y logs |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 9 | Falta de documentación del código | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Buenas prácticas de comentarios y herramientas como Swagger |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 10 | No manejo de errores asíncronos | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Uso correcto de async/await, try/catch |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 11 | Integraciones fallidas con servicios externos | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Simulaciones con mocks y fallback automático |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 12 | Pérdida de datos por mal diseño de base de datos | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Pruebas de integridad y backup regular |
| Cuantitativo |  |  |  |  |  |  |  |  |  |  |
| 13 | Desacoplamiento deficiente entre componentes | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Aplicar principios de inyección de dependencias |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 14 | Librerías de terceros sin mantenimiento | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Verificación periódica de dependencias |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 15 | Inestabilidad en ambientes de desarrollo | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Contenerización y automatización con Docker |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 16 | Sobrecarga del servidor por consultas pesadas | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Indexación, paginación y optimización de queries |
| Cuantitativo |  |  |  |  |  |  |  |  |  |  |
| 17 | Mal uso del control de versiones | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Establecer una estrategia Git (flow) clara |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 18 | Falta de pruebas unitarias y de integración | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Definir cobertura mínima desde el inicio |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 19 | Fallos en carga de archivos multimedia | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Validaciones, límites y compresión previa |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 20 | Pérdida de sesiones o tokens de usuarios | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Autenticación con JWT y políticas de expiración segura |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 21 | No controlar el crecimiento del código | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Refactorización continua y revisión técnica |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 22 | No contemplar internacionalización/  localización | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Implementar i18n desde etapas tempranas |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 23 | Exposición de claves API en el código | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Uso de variables de entorno y herramientas como dotenv |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 24 | Testing inadecuado de interacciones entre módulos | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Pruebas end-to-end y de integración continua |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |
| 25 | Cambios frecuentes en lógica sin pruebas de regresión | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | Implementar pruebas automatizadas en cada commit |
| Cualitativo |  |  |  |  |  |  |  |  |  |  |