

# *Políticas de testing*

## **1. Objetivo de la Política de Testing**

Esta política tiene como objetivo establecer los lineamientos para el testing del módulo de gestión de reportes, usuarios y publicaciones. Al asegurar la precisión y confiabilidad en la gestión de reportes y la información de usuarios y publicaciones, buscamos:

- **Mejorar la moderación de contenido:** Garantizar que los reportes sean procesados de manera adecuada para proteger la integridad de la plataforma.
- **Mantener la calidad y transparencia de datos:** Asegurar que los datos almacenados en cada entidad son consistentes, válidos y responden correctamente a las consultas y reportes generados.
- **Optimizar el flujo de moderación:** Validar que el sistema responda eficazmente en el seguimiento del estado de los reportes y en la actualización de las métricas de usuarios y publicaciones reportadas.

## **2. Alcance**

Esta política se aplica a las pruebas relacionadas con el módulo de reportes, los usuarios y las publicaciones en la plataforma. Incluye los siguientes niveles de testing:

- **Testing de la Entidad Reporte (Report):**
  - **Pruebas de Integridad de Datos:** Verificación de que todos los campos obligatorios, como reportId, reporterUserId, reportedUserId, reportDate y reportReason, están correctamente validados.
  - **Pruebas de Estado del Reporte:** Validar los posibles estados (reportStatus) de los reportes, como "Pendiente", "Revisado" y "Resuelto", para asegurar que el flujo de estados se siga correctamente.
  - **Pruebas de Relaciones de Identificadores:** Asegurar que postId, reporterUserId, y reportedUserId están correctamente asociados a registros válidos en las entidades correspondientes.

- **Testing de la Entidad Usuario (User):**
  - **Pruebas de Integridad y Seguridad de Datos:** Validación de los campos `userId`, `username`, `email`, y `accountStatus`, asegurando la confidencialidad de datos personales y su correcta encriptación.
  - **Pruebas de Estado de Cuenta:** Validación de los estados `accountStatus`, como "Activa", "Suspendida" o "Baneada", garantizando el flujo de actualización adecuado en cada estado.
  - **Métricas de Reportes:** Pruebas para verificar que `reportsReceived` y `lastReportDate` se actualicen correctamente en cada reporte que involucra al usuario.
- **Testing de la Entidad Publicación (Post):**
  - **Pruebas de Contenido:** Verificación de que el `content` de la publicación cumple con las reglas de formato y tamaño permitidas.
  - **Pruebas de Visibilidad:** Validación de los niveles de `visibility` para asegurar que solo los usuarios autorizados pueden ver cada publicación.
  - **Métricas de Reportes en Publicaciones:** Asegurar que los campos `reportsCount` y `lastReportDate` reflejan correctamente el número y la fecha de los reportes recibidos.

### 3. Metas de Calidad

Para garantizar la confiabilidad y exactitud del módulo de gestión de reportes, se establecen los siguientes objetivos de calidad:

- **Integridad de los Datos:** Asegurar que todas las entidades se crean y actualizan correctamente según los parámetros especificados y que no existen datos inconsistentes en el sistema.
- **Correcta Gestión de Estados:** Validar que los estados, tanto de reportes (`reportStatus`) como de usuarios (`accountStatus`), reflejen el estado actual de cada entidad y que los cambios de estado se realicen de forma controlada.

- **Precisión en Métricas de Reportes:** Verificar que las métricas como reportsReceived en usuarios y reportsCount en publicaciones son exactas y se actualizan en tiempo real o de acuerdo con el flujo de procesos.

#### 4. Revisión y Actualización

Esta política de testing será revisada y actualizada de acuerdo con los siguientes parámetros:

- **Periodicidad:** La revisión se realizará cada 6 meses o tras la introducción de cambios en las especificaciones de la entidad.
- **Responsables:** El equipo de QA junto con el líder de proyecto y los administradores de la plataforma revisarán los cambios necesarios para reflejar mejoras continuas en el módulo de reportes.
- **Capacitación del Equipo:** Cada actualización en esta política será comunicada mediante capacitaciones para garantizar su correcta implementación y seguimiento.

#### 5. Responsabilidades del Equipo

Para asegurar un testing efectivo y coherente, se define la responsabilidad de cada rol en el equipo:

- **QA:** Responsable de definir y ejecutar los casos de prueba para cada entidad, identificando defectos y garantizando la integridad de los datos.
- **Desarrolladores:** Ejecutan pruebas unitarias para asegurar que los métodos y validaciones de cada entidad funcionan como se espera. Corrigen los defectos detectados en su código y colaboran con el equipo de QA en las pruebas de integración.
- **Analistas de Datos:** Ayudan a establecer criterios de aceptación para métricas y valores relacionados con los reportes en usuarios y publicaciones.
- **Moderadores:** Revisan los reportes generados para confirmar que el flujo de gestión de reportes es adecuado y que los cambios en estados se aplican correctamente.

## 6. Herramientas y Entorno de Testing

Las herramientas y el entorno deben reflejar los datos reales de la plataforma, respetando la confidencialidad y asegurando la precisión en las pruebas.

- **Herramientas de Gestión de Pruebas:** Uso de TestRail o Jira para documentar y dar seguimiento a los resultados de pruebas de cada entidad.
- **Herramientas de Validación de Datos:** Herramientas de base de datos (como SQL) para verificar la correcta actualización y consistencia de los datos en cada entidad.
- **Entornos de Testing:** En el entorno de pruebas se cargarán datos ficticios representativos que imiten casos de uso reales, asegurando que los cambios en estados y métricas son reflejados correctamente y de manera segura.

## 7. Planificación y Cronograma

- **Frecuencia de Pruebas:** Testing continuo en cada commit importante, además de pruebas de regresión para las funciones críticas como el cambio de estados de cuenta o visibilidad de publicaciones.
- **Plazos de Ejecución:** Cada ciclo de testing se organizará en fases cortas, asegurando la disponibilidad de cada entidad (Report, User, Post) para ser testeada por completo en el tiempo estipulado.
- **Pruebas de Regresión:** Los cambios en estados y métricas serán evaluados en cada actualización de los métodos correspondientes para asegurar que se preserva la integridad de las funcionalidades.

## 8. Gestión de Riesgos y Mitigación

- **Riesgos Técnicos:** Riesgos de integridad de datos en los casos en que un reporte pueda fallar, lo que puede comprometer la precisión de métricas o estados en usuarios y publicaciones.
- **Mitigación de Riesgos:** Implementación de validaciones adicionales para asegurar que los datos críticos (como reportsCount y reportsReceived) sean consistentes en caso de fallos en la aplicación.

## **9. Documentación y Revisión de las Políticas**

- **Registro de Pruebas:** Cada prueba será documentada y registrada para auditar el cumplimiento de la política y la integridad de cada entidad.
- **Actualización de la Política:** La política será revisada anualmente o después de cualquier cambio significativo en la estructura de datos o en los métodos de gestión de reportes.
- **Capacitación del Equipo:** Se ofrecerán capacitaciones cada vez que se actualice esta política, asegurando que los equipos de QA y desarrollo aplican las normas adecuadamente.