# **Manual de Control de Errores Post-Entrega**

## **1. Introducción**

Este manual establece las directrices para gestionar los errores identificados después de la entrega del proyecto al cliente. El objetivo es proporcionar un enfoque organizado y eficiente para la resolución de errores en la plataforma de **Generative Art Generation Platform (GAAP)**, especialmente enfocado en problemas relacionados con el sistema de generación de arte, la base de datos, y la interacción con los usuarios.

## **2. Identificación de Errores**

### **2.1 Proceso de Reporte de Errores**

Los errores deben ser reportados por el cliente o el equipo de soporte siguiendo estos pasos:

1. **Descripción del Error**: El cliente debe proporcionar una breve descripción sobre el problema experimentado.
2. **Componente Afectado**: Identificar si el error está relacionado con la base de datos, la interfaz de usuario (UI), el sistema de algoritmos de IA, o la plataforma de generación de arte.
3. **Pasos para Reproducir el Error**: El cliente debe indicar si puede replicar el error y los pasos que siguió.
4. **Impacto del Error**: El cliente debe detallar el impacto del error (por ejemplo, pérdida de arte generado, falta de funcionalidad de la plataforma, error en la visualización).

### **2.2 Prioridad del Error**

* **Alta**: Errores críticos que afectan la funcionalidad principal de la plataforma, como fallas en la generación de arte o pérdidas de datos.
* **Media**: Errores que afectan funciones secundarias pero no interrumpen completamente el uso de la plataforma, como problemas de visualización de ciertos parámetros o interacciones.
* **Baja**: Errores menores, como problemas visuales en la interfaz de usuario o pequeños detalles en la funcionalidad de la plataforma.

## **3. Clasificación de Errores**

### **3.1 Tipos de Errores Comunes**

1. **Errores Funcionales**
   * **Descripción**: El sistema no cumple con las funciones esperadas, como la generación incorrecta de arte o fallas en los comandos de voz.
   * **Ejemplos**: La generación de arte no se activa, la interfaz de usuario no responde a los comandos, o los datos no se procesan adecuadamente.
2. **Errores de Base de Datos**
   * **Descripción**: Errores relacionados con la persistencia de los datos, como fallas en la base de datos que causan pérdida o corrupción de los datos generados.
   * **Ejemplos**: Pérdida de registros de arte generados, errores al guardar las preferencias del usuario, problemas con las consultas a la base de datos.
3. **Errores de Rendimiento**
   * **Descripción**: El sistema se vuelve lento o no responde correctamente bajo ciertas condiciones de carga.
   * **Ejemplos**: Lenteza al generar arte, caída de la plataforma cuando muchos usuarios acceden simultáneamente.
4. **Errores de Interfaz de Usuario (UI/UX)**
   * **Descripción**: Problemas visuales o de interacción que afectan la experiencia del usuario.
   * **Ejemplos**: Elementos de la interfaz que no se alinean correctamente, botones no funcionales, problemas con la visualización de arte generado.
5. **Errores de Seguridad**
   * **Descripción**: Vulnerabilidades en la plataforma que pueden exponer a los usuarios o al sistema a riesgos.
   * **Ejemplos**: Fallos en la autenticación, acceso no autorizado a datos de usuario.

## **4. Proceso de Resolución de Errores**

### **4.1 Diagnóstico del Error**

Una vez que un error ha sido reportado, se sigue un proceso de diagnóstico para comprender el problema:

1. **Reproducir el Error**: Intentar replicar el error en un entorno controlado.
2. **Análisis del Log**: Revisar los registros de la plataforma para identificar el origen del error.
3. **Revisión de Código/Algoritmos**: Si es un error funcional relacionado con los algoritmos de IA o generación de arte, revisar el código para identificar fallos lógicos.
4. **Pruebas en Entorno de Desarrollo**: Probar el error en un entorno de desarrollo para entender mejor su comportamiento.

### **4.2 Plan de Acción**

Una vez identificado el origen del error, se implementan las siguientes acciones:

1. **Corrección del Error**:
   * Modificar el código de la plataforma, los algoritmos de IA, o la base de datos según sea necesario.
   * Si el error está relacionado con la UI/UX, se realizarán ajustes visuales.
2. **Pruebas**:
   * **Pruebas Unitarias**: Realizar pruebas unitarias en el código modificado.
   * **Pruebas de Integración**: Asegurar que la solución no cause otros problemas en la integración del sistema.
3. **Despliegue**: Implementar la solución en el entorno de producción, asegurando que se minimicen los tiempos de inactividad.

## **5. Asignación de Recursos (Horas y Equipos)**

### **5.1 Estimación de Tiempo para la Resolución**

El tiempo estimado para resolver un error depende de la complejidad del mismo. Aquí se asignan horas basadas en el tipo de error:

| **Tipo de Error** | **Horas Estimadas** | **Descripción** |
| --- | --- | --- |
| **Errores Funcionales** | 20-30 horas | Problemas en la generación de arte, algoritmos de IA o comandos de voz. Resolución de errores en el backend o en la interacción del usuario. |
| **Errores de Base de Datos** | 15-25 horas | Problemas en la persistencia de datos, integridad de la base de datos o recuperación de registros de usuario o arte generado. |
| **Errores de Rendimiento** | 25-35 horas | Optimización de la plataforma para mejorar tiempos de respuesta o gestión de múltiples usuarios. |
| **Errores de UI/UX** | 15-20 horas | Ajustes en la interfaz de usuario, como cambios en el diseño o mejora de la experiencia del usuario. |
| **Errores de Seguridad** | 20-30 horas | Implementación de medidas de seguridad como cifrado de datos, autenticación más robusta, o parches de vulnerabilidad. |

### 

### **5.2 Gestión de Recursos Humanos**

Dependiendo del tipo de error, los siguientes perfiles deben ser asignados:

* **Desarrolladores de Software**: Para resolver errores funcionales y de rendimiento.
* **DBA (Administrador de Base de Datos)**: Para resolver errores relacionados con la base de datos.
* **Especialistas en IA/Algoritmos**: Para corregir problemas relacionados con la generación de arte o inteligencia artificial.
* **Diseñadores UI/UX**: Para solucionar problemas de interfaz o experiencia de usuario.
* **Expertos en Seguridad**: Para abordar vulnerabilidades y mejorar la seguridad de la plataforma.

## **6. Seguimiento y Monitoreo**

### **6.1 Seguimiento del Progreso**

El progreso de la resolución de errores se debe supervisar de manera continua:

1. **Revisión Diaria**: Revisión diaria del avance en la resolución de los problemas reportados.
2. **Actualizaciones Regulares**: El equipo debe proporcionar actualizaciones regulares sobre el estado de cada error.
3. **Informe Final**: Cuando el error se solucione, se debe generar un informe detallado sobre las acciones tomadas.

### **6.2 Control de Calidad**

Después de implementar una solución, es crucial realizar las siguientes pruebas:

* **Pruebas de Regresión**: Asegurarse de que las soluciones no generen nuevos problemas.
* **Pruebas de Usuario**: Verificar que el cliente esté satisfecho con la solución.

## **7. Comunicación con el Cliente**

### **7.1 Reporte de Progreso**

La comunicación con el cliente es fundamental:

1. **Notificación Inicial**: Informar al cliente sobre el error y el proceso de resolución.
2. **Actualizaciones**: Proveer actualizaciones periódicas sobre el progreso de la solución.
3. **Informe Final**: Al resolver el error, se enviará un informe detallado con las acciones realizadas y los resultados obtenidos.

## **8. Cierre de Caso y Retroalimentación**

Una vez resuelto el error:

1. **Confirmación de Satisfacción**: El cliente debe confirmar que el error ha sido resuelto satisfactoriamente.
2. **Análisis Post-Mortem**: Se debe realizar una reunión de retroalimentación para discutir qué funcionó bien y qué podría mejorarse en el proceso de resolución de errores.