

# **POLÍTICA DE LA GESTIÓN DE LAS ISSUES**

**Fecha de Emisión:** 17/10/2024

**Descripción:** Normativa para la creación, clasificación, asignación y gestión de issues en el proyecto, con el fin de garantizar una adecuada organización y seguimiento del trabajo del equipo.

## **Introducción y Propósito**

Este documento tiene como objetivo definir una política formal para el manejo de *issues* en GitHub, permitiendo a los integrantes del proyecto colaborar de manera eficiente en la gestión de tareas, errores y mejoras. La correcta gestión de *issues* asegura que el trabajo se documente adecuadamente y que los problemas y funcionalidades pendientes se puedan resolver de manera priorizada y estructurada.

## **Tipos de Issues**

Para clasificar adecuadamente cada *issue*, utilizaremos **etiquetas estándar** que ayudarán a identificar su propósito y prioridad:

### **1. Funcionalidades Nuevas (Feature)**

- **Descripción:** Solicitudes para implementar una nueva funcionalidad o mejorar una existente.
- **Etiqueta:** **feature**
- **Requisito:** El *issue* debe describir claramente qué funcionalidad se necesita y por qué es importante.

### **2. Corrección de Errores (Bug)**

- **Descripción:** Reportes de errores en el sistema que están causando fallos, comportamientos inesperados o problemas de rendimiento.
- **Etiqueta:** **bug**
- **Requisito:** El *issue* debe incluir una descripción clara del error, los pasos para reproducirlo, el entorno en el que ocurre (por ejemplo, versión de software, sistema operativo) y una posible captura de pantalla o mensaje de error.

### **3. Mejoras (Enhancement)**

- **Descripción:** Sugerencias para mejorar una funcionalidad ya existente, sin que sea una nueva característica.
- **Etiqueta:** **enhancement**
- **Requisito:** Describir qué mejora se propone y cómo impacta en la experiencia del usuario o el rendimiento del sistema.

## 4. Documentación (Docs)

- **Descripción:** Solicitudes o reportes relacionados con la creación o actualización de la documentación del proyecto.
- **Etiqueta:** docs
- **Requisito:** El *issue* debe especificar qué parte de la documentación requiere cambios o mejoras.

## 5. Tareas Generales (Chore)

- **Descripción:** Trabajo que no afecta directamente la funcionalidad del producto, como actualizaciones de dependencias, configuraciones del entorno o mantenimiento general.
- **Etiqueta:** chore
- **Requisito:** Deben especificar claramente cuál es la tarea y su impacto en el proyecto.

## 6. Consultas o Investigaciones (Research)

- **Descripción:** Solicitudes de investigación o consulta para analizar posibles soluciones antes de implementar una funcionalidad o corregir un error.
- **Etiqueta:** research
- **Requisito:** El *issue* debe describir qué se debe investigar y cuál es el objetivo de la investigación.

# Creación de Issues

### 1. Título del Issue:

- Debe ser **claro y conciso**, describiendo el propósito del *issue*. Evitar títulos genéricos como "Problema" o "Error".
- Ejemplo de título correcto: Error al guardar datos en la base de datos, Agregar autenticación por Google.

### 2. Descripción del Issue:

- La descripción debe contener la siguiente estructura:
  - **Resumen del problema** o solicitud.
  - **Pasos para reproducir el error** (si es un *bug*).
  - **Comportamiento esperado** frente al comportamiento real (si aplica).
  - **Contexto** adicional, como capturas de pantalla, logs, o referencias a otros *issues*.
  - **Entorno** (en caso de un error), especificando versiones del software, sistema operativo, etc.

### 3. Asignación de Etiquetas:

- Cada *issue* debe tener una o más etiquetas que representen su tipo (por ejemplo, bug, feature, docs).
- Las etiquetas ayudarán a priorizar el trabajo y definir qué acciones son necesarias.

#### 4. Asignación de Prioridad:

- Los *issues* deben clasificarse en términos de prioridad:
  - **Alta** (*priority: high*): Problemas críticos o características esenciales que deben resolverse pronto.
  - **Media** (*priority: medium*): Funcionalidades importantes o errores que no bloquean el proyecto pero que deben solucionarse.
  - **Baja** (*priority: low*): Tareas que pueden resolverse eventualmente, sin urgencia inmediata.

#### 5. Asignación de Responsables:

- Todo *issue* debe ser asignado a un responsable específico que será el encargado de resolverlo. Si no se asigna inicialmente, se debe discutir en la reunión de planificación quién será responsable de su resolución.
- Un *issue* no debe tener múltiples responsables principales para evitar confusión.

#### 6. Referencias a Commits y Pull Requests:

- Siempre que un commit o un pull request resuelva un *issue*, debe referenciarse adecuadamente en el cuerpo del commit o pull request.
- Utiliza el formato adecuado para cerrar automáticamente el *issue* cuando se fusione el pull request:
  - **Closes** #<número-del-issue>
  - **Fixes** #<número-del-issue>

## Flujo de Trabajo de los Issues

1. **Creación:** Los *issues* pueden ser creados por cualquier miembro del equipo o cualquier persona autorizada para ello.
2. **Clasificación y Etiquetado:** Los líderes del proyecto o encargados del equipo revisarán los nuevos *issues*, asignando las etiquetas apropiadas y clasificándolos por prioridad.
3. **Asignación de Responsables:** Se asignará un desarrollador o un equipo responsable para cada *issue*.
4. **Revisión de Progreso:**
  - Los *issues* deben revisarse regularmente en las reuniones de planificación (por ejemplo, reuniones semanales o de sprint) para ajustar su prioridad y asignación si es necesario.
  - Los *issues* que no estén progresando deberán reevaluarse para identificar bloqueos o reasignar recursos.
5. **Resolución:**
  - Una vez que un *issue* se haya resuelto, debe mencionarse en el pull request correspondiente para cerrar el *issue* automáticamente.
  - El revisor del pull request debe asegurarse de que el *issue* esté completamente resuelto antes de proceder a la fusión.
6. **Cierre de Issues:**
  - Un *issue* debe cerrarse cuando se haya resuelto el problema, y el cambio correspondiente ha sido integrado en la rama **develop** o **main**.

- Si un *issue* no tiene sentido o se considera obsoleto, debe cerrarse con una explicación clara.

## Reglas Adicionales

### 1. Evitar la Duplicación de Issues:

- Antes de crear un nuevo *issue*, verifica si ya existe un *issue* abierto que trate el mismo problema o solicitud. Si es así, comenta en el *issue* existente en lugar de crear uno nuevo.

### 2. Comunicación en los Issues:

- Usa los *issues* como punto central de discusión para tareas específicas. Los comentarios deben ser claros, útiles y profesionales.
- Si un *issue* está bloqueado por otro, indícalo claramente en los comentarios del *issue* afectado.

### 3. Reapertura de Issues:

- Si un *issue* se cierra prematuramente o el problema persiste, debe reabrirse y actualizarse con nueva información.

## Firma de los integrantes:

- Espinosa Naranjo, Pablo:
- Garate Fuentes, Yesica:
- Harana Mancilla, Rafael:
- Pizarro López, Eduardo:
- Portillo Sánchez, Alonso:
- Sevillano Barea, Alejandro: