



## **Ciclo de vida de un defecto**





# Ciclo de vida de un defecto

ELDAR ACADEMY

Versión: 1.0

Fecha: 05 de Diciembre de 2024

<b>Objetivo</b>	<b>3</b>
<b>Introducción</b>	<b>3</b>
<b>Etapas del ciclo de vida</b>	<b>4</b>
<b>Ejemplo práctico de bug con herramientas</b>	<b>6</b>
<b>Conclusión</b>	<b>9</b>
<b>Bibliografía</b>	<b>10</b>



## **Objetivo**

El objetivo de esta actividad o reto es comprender y documentar el ciclo de vida de un bug dentro de un flujo de trabajo típico en QA. Este desafío está pensado para que el colaborador pueda fortalecer las habilidades de documentación y el análisis crítico.

## **Introducción**

Para comenzar con esta actividad vamos a desarrollar de manera conceptual que es un bug.

- En el desarrollo de software, un bug es una inconsistencia o defecto detectado durante las etapas de prueba, aunque también pueden surgir en cualquier fase del proceso de creación de programas. Los bugs pueden ser tan simples como un error tipográfico o tan complejos como un problema de lógica que sólo se manifiesta bajo ciertas condiciones.

## **¿Qué es un ciclo de vida de un bug?**

Es el proceso mediante el cual se identifican, informan, gestionan y finalmente se resuelve, a esto se lo conoce como el ciclo de vida. A lo largo de este ciclo, un bug atraviesa diversas etapas, cada una con sus propias actividades y responsabilidades las cuales son fundamentales para garantizar una alta calidad del software y un flujo de trabajo de desarrollo optimizado.



## Etapas del ciclo de vida

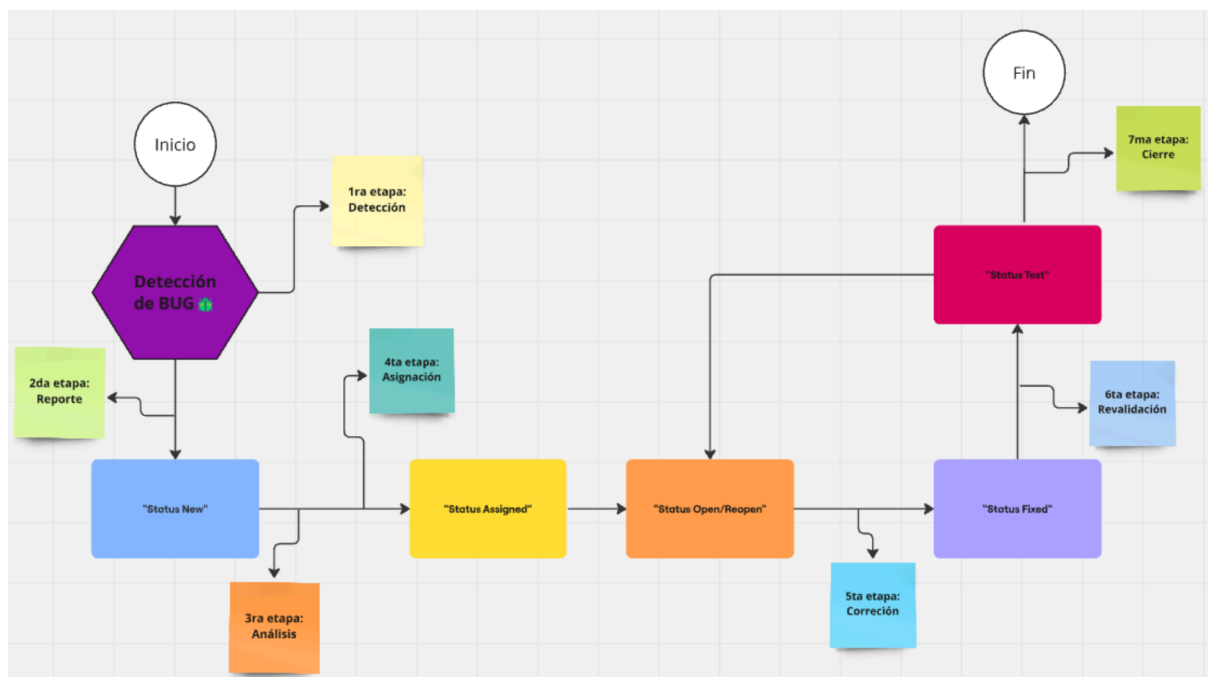
Como mencionamos anteriormente, el ciclo de vida de un bug incluye **varias etapas** clave que deben desarrollarse para garantizar su resolución. A lo largo de estas etapas, el bug atraviesa diferentes **status** o **estados** que reflejan su progreso, desde su detección inicial hasta su cierre final.

- **Detección:** En esta etapa inicial, un bug puede ser identificado por un usuario, tester o mediante herramientas de prueba automatizadas. Los bugs suelen ser encontrados durante diferentes actividades de prueba, como las pruebas funcionales, pruebas de regresión o incluso durante el entorno de producción.
- **Reporte:** Una vez descubierto, el bug se coloca en un “**Status New**” donde se documenta y reporta al equipo de desarrollo. Este informe suele incluir una descripción detallada del problema, pasos para reproducirlo, el entorno en el que surgió y cualquier otro dato relevante que facilite su corrección.
- **Análisis:** Una vez reportado el bug es analizado, y este comienza un proceso de triaje, es decir se clasifica o evalúa según su severidad y prioridad. Recordar que la severidad indica el impacto en el sistema y la prioridad determina el orden en que se aborda el bug. Esta etapa es importante ya que se establece que bug se soluciona primero según importancia y urgencia.
- **Asignación:** Seguido del análisis, el bug pasa a un “**Status Assigned**”. El líder de desarrollo deberá asignar a un desarrollador o a un equipo para su corrección y el bug pasa a un “**Status Open**”. El responsable analizará el informe e intentará reproducir el problema y comienza a investigar su causa raíz.
- **Corrección:** en esta etapa, el desarrollador trabaja en la solución del bug reportado, lo que puede implicar modificar el código fuente, ajustes de configuración u otros componentes del software. Una vez implementada la solución, el estado del bug pasa a

ser **"Status Fixed"**, se ejecutan pruebas para verificar que el problema ha sido resuelto y que no se han introducido nuevos errores.

- **Revalidación/Verificación:** Una vez corregido, el software actualizado es sometido a pruebas, y el estado del bug pasa a tener un **"Status Test"** donde los testers se encargaran de evaluar la funcionalidad afectada para garantizar que el error haya sido solucionado. Si el bug sigue siendo reproducible, se reabre y se vuelve al desarrollador para una mayor investigación y corrección, en este caso pasa a un estado de **"Status Open/Reopen"**.
- **Cierre:** En este último paso, siendo que el bug haya sido verificado correctamente se cierra el bug y pasa al **"Status Closed"**, considerándose oficialmente corregido. Los detalles de la resolución se registran y el bug se marca como "Cerrado" en el sistema de seguimiento concluyendo así su ciclo de vida.

## Diagrama de ciclo de vida del bug con sus estados





## **Ejemplo práctico de bug con herramientas**

- Bug detectado: **Al cargar una imagen en el perfil, esta no realiza el cambio.**

Para abordar este problema, se realizó un análisis de causa raíz con los siguientes pasos:

### **Identificación del problema:**

- Al intentar cargar una imagen en el perfil, la app no la cambia y no lanza ningún tipo de error.
- El problema ocurre siempre que se suben imágenes mayores a 5MB.

### **Recolección de datos:**

- Se recopilan capturas de pantalla y videos que muestran ese comportamiento.
- Se documentan y replican los pasos exactos para reproducir el error.
- Revisión de los registros de la app (mensajes de error, información del rendimiento, etc).

### **Análisis del problema:**

- Los registros muestran un error de "Out of Memory" al procesar imágenes grandes.
- El problema ocurre cuando las imágenes son grandes y en dispositivos con poca memoria disponible.

### **Identificación de la Causa Raíz:**

- La app no implementa validaciones para limitar el tamaño de las imágenes cargadas.

### **Acciones correctivas:**

- Implementación de validaciones para limitar el tamaño de las imágenes que los usuarios pueden subir.
- Optimizar el manejo de la memoria al procesar imágenes grandes.

### **Verificación:**

- Probar la solución con imágenes de diferentes tamaños para garantizar que la app no se bloquee.

### **Seguimiento y prevención:**

- Actualizar los casos de prueba para incluir la validación del tamaño de las imágenes.

Seguido de este análisis será informado a través del sistema de Gestión de **JIRA**



Proyectos /  PrimeraPractica /  SCRUM-4 /  SCRUM-5

## Error al cargar imagen en perfil

+ Añadir

🔗 Aplicaciones

### Descripción

📄 **Version:** 1.0

🔍 **Descripción:** Al intentar cargar una imagen de perfil, esta no realiza el cambio y no lanza ningún tipo de error.

🌿 **Severidad:** Baja

🔧 **Prioridad:** Baja

### 👤 Pasos para reproducir:

1. Iniciar sesión en la app.
2. Ir a la sección del perfil del usuario.
3. Seleccionar la opción "Editar imagen de perfil".
4. Elegir una imagen mayor a 5MB.
5. Seleccionar "Guardar".

### ⚠️ Resultados actual:

- La imagen no se carga correctamente
- El sistema no lanza ningún tipo de error.

### ✅ Resultado esperado:

- La imagen seleccionada debería cargarse y mostrarse en el perfil del usuario sin errores.

### 🔴 Info adicional:

- El formato de la imagen debe ser .jpg o .png.
- El tamaño de la imagen debe ser menor o igual a 5MB para guardarse.

En principio el QA **detecta** el bug, el cual corresponde que "al intentar cargar una imagen de perfil, esta no realiza el cambio y no lanza ningún tipo de error". Seguido de un **reporte** el cual se abre una incidencia en el sistema de gestión de JIRA, en este punto el bug se pone en un estado de **"new"** donde se registran los pasos a reproducir y otra información detallada para que el líder de los desarrolladores pueda verificar y **analizar** el bug y le dará cierta prioridad para resolverlo.

Luego de esto se realiza una **asignación** por parte del líder de desarrollo y este se lo asigna a un desarrollador, en este tiempo el bug pasa de un estado **"asignado"** a un estado de **"open"**.



Una vez asignado el bug pasa a **corregirse** por parte del desarrollador, cuando lo soluciona el estado del bug pasa a ser **"Fixed"**. Una vez corregido, el bug pasa a un estado de **"Test"** y el área de QA pasa a realizar una **verificación** del bug donde evaluarán la funcionalidad de este. Por último, si la verificación es correcta el bug pasa a un estado **"Closed"**, donde se da un **cierre** al igual que la incidencia en JIRA.

## **Conclusión**

Se puede observar que el ciclo de vida de un defecto es un proceso fundamental en el control de calidad del software, ya que permite gestionar de manera eficiente los errores detectados durante el desarrollo. Asegurando que los bugs sean identificados, documentados, analizados, corregidos y verificados antes de cerrar la incidencia.

También, se puede observar como hay una comunicación efectiva y un flujo de trabajo organizado entre los equipos de QA y desarrollo. El uso de las herramientas de gestión como JIRA facilita el seguimiento y la trazabilidad de cada bug permitiendo un control sobre la resolución.

En conclusión entender y aplicar correctamente este ciclo no solo ayuda a reducir los errores del software, sino que también mejora la calidad del producto y satisfacción del cliente.





## Bibliografía

- <https://es.abstracta.us/blog/ciclo-vida-bug-software/>
- <https://www.linkedin.com/pulse/el-ciclo-de-vida-un-bug-y-la-importancia-control-calidad-marco-amaya/>
- <https://msmk.university/bug/>
- <https://leanmind.es/es/blog/ciclo-de-vida-de-un-error-bug-en-el-software/>

### Videos:

- <https://www.youtube.com/watch?v=CWXORakpZRw&t=403s>
- <https://www.youtube.com/watch?v=LPEKOtVwiFE>