

## 1. Introducción a Releases en Jira

**Releases** o versiones son puntos clave en la evolución de un producto. En Jira, gestionar releases implica definir versiones, asignar incidencias a cada release y usar **JQL** para filtrar y obtener reportes precisos. Esta práctica es fundamental para el seguimiento de la calidad y la evolución del software a lo largo del ciclo de desarrollo.

## 2. Requisitos y Configuración del Proyecto

### 2.1. Requisitos Básicos

- **Cuenta en Jira:** Acceso a Jira Cloud o a una instancia local (Jira Server/Data Center).
- **Permisos Adecuados:** Permisos para crear y modificar proyectos, versiones y filtros.
- **Proyecto Configurado:** Un proyecto con las incidencias y, opcionalmente, historias de usuario, bugs, tareas, etc.

### 2.2. Configuración del Proyecto para Releases

1. **Acceso al Proyecto:** Ingresa a la instancia de Jira y abre el proyecto en el que deseas gestionar releases.
2. **Definir Versiones:**
  - Ve a **"Project Settings"** (Configuración del Proyecto) y selecciona **"Releases"** o **"Versions"**.
  - Crea una nueva versión (por ejemplo, "Version 1.0", "Version 2.0") asignándole una fecha de lanzamiento y una descripción.
3. **Asignación de Incidencias a Releases:**
  - En cada incidencia, utiliza el campo **"Fix Version/s"** para asignarla a la versión en la que se corregirá o implementará la funcionalidad.
  - Si se trata de incidencias que afectan a versiones previas, también puedes usar el campo **"Affects Version/s"**.

## 3. Gestión y Creación de Releases

### 3.1. Crear y Administrar Releases

- **Crear una Nueva Versión:**  
Desde la sección de Releases del proyecto, haz clic en **"Crear versión"**. Ingresa el nombre (por ejemplo, "Version 1.0"), la fecha estimada de lanzamiento y cualquier detalle relevante.
- **Editar o Eliminar Releases:**  
Puedes modificar o eliminar versiones en caso de cambios en la planificación. Es importante actualizar la información de cada release para mantener una trazabilidad clara.

### 3.2. Flujo de Trabajo con Releases

- **Planificación:** Define qué incidencias se incluirán en la versión.
- **Asignación:** Asigna incidencias mediante el campo **Fix Version/s**.
- **Monitoreo:** Utiliza filtros y dashboards para seguir el progreso de la versión y evaluar la calidad.
- **Lanzamiento:** Una vez finalizada la versión, márcala como **"Released"** en Jira.

## 4. Uso de JQL para Releases

### 4.1. Filtrado por Versión con JQL

JQL te permite crear consultas para filtrar incidencias según la versión en la que fueron asignadas. Algunos ejemplos básicos son:

- **Incidencias asignadas a una versión específica:**

`fixVersion = "Version 1.0"`

- **Incidencias que afectan a una versión determinada:**

`affectedVersion = "Version 1.0"`

- **Incidencias asignadas a versiones no liberadas (Unreleased):**

`fixVersion in unreleasedVersions()`

- **Incidencias de versiones ya liberadas:**

`fixVersion in releasedVersions()`

### 4.2. Consultas Avanzadas

#### Filtro 1: Incidencias Pendientes en un Release Específico

`fixVersion = "Version 2.0" AND status not in (Cerrado, Resuelto)`

Este filtro muestra todas las incidencias que pertenecen a "Version 2.0" y que aún no se han cerrado ni resuelto.

#### Filtro 2: Incidencias con Prioridad Alta en Releases No Liberadas

`fixVersion in unreleasedVersions() AND priority = Alta`

Ideal para identificar incidencias críticas que requieren atención antes del lanzamiento.

#### Filtro 3: Incidencias Combinadas entre Releases y Otros Criterios

`fixVersion = "Version 1.0" AND (issuetype = Bug OR issuetype = Historia) AND updated >= -14d`

Permite ver incidencias recientes (últimas 2 semanas) de tipo Bug o Historia asignadas a "Version 1.0".

## 5. Aplicación Práctica: Dashboards y Reportes de Releases

### 5.1. Guardar Filtros y Configurar Gadgets

### 1. **Guardar Filtros:**

Después de crear consultas en JQL, guarda los filtros (por ejemplo, “Incidencias Version 1.0 Abiertas”) para su uso futuro.

### 2. **Configurar Dashboards:**

- Accede a la sección de "**Dashboards**" y crea un nuevo panel o edita uno existente.
- Añade gadgets como "**Filter Results**", "**Pie Chart**" o "**Created vs. Resolved Chart**" basados en los filtros guardados.
- Esto te permitirá visualizar el progreso de cada release, identificar cuellos de botella y monitorear la calidad del lanzamiento.

## 5.2. Integración con el Flujo de Trabajo

- **Revisión Periódica:** Revisa los dashboards durante las reuniones de seguimiento (por ejemplo, reuniones de lanzamiento o scrum) para evaluar el estado de las incidencias en cada release.
- **Ajustes en la Planificación:** Utiliza la información obtenida para priorizar incidencias críticas y ajustar la planificación del release en función de los resultados.