

**PLANIFICACIÓN Y GESTIÓN DE PROYECTOS INFORMÁTICOS (2018-2019)**  
MÁSTER PROFESIONAL DE INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Análisis de herramientas

---

Alejandro Campoy Nieves  
Luis Gallego Quero

23 de diciembre de 2018

## Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Control de versiones</b>	<b>3</b>
2.1	Git . . . . .	3
2.2	Bazaar . . . . .	3
2.3	Mercurial . . . . .	4
<b>3</b>	<b>Compilación</b>	<b>4</b>
3.1	Eclipse . . . . .	4
3.2	NetBeans . . . . .	5
3.3	Android Studio . . . . .	5
<b>4</b>	<b>Automatización de pruebas</b>	<b>5</b>
4.1	Selenium . . . . .	5
4.2	Apache JMeter . . . . .	6
4.3	JUnit . . . . .	6
<b>5</b>	<b>Integración continua</b>	<b>7</b>
5.1	Travis IC . . . . .	7
5.2	Bamboo . . . . .	7
5.3	Jira . . . . .	8
<b>6</b>	<b>Seguimiento de errores/defectos</b>	<b>8</b>
6.1	Sentry . . . . .	8
6.2	GitLab . . . . .	9
6.3	Mantis Bug Tracker . . . . .	9
<b>7</b>	<b>Comparativa de herramientas</b>	<b>10</b>
7.1	Control de versiones . . . . .	10
7.2	Compilación . . . . .	10
7.3	Automatización de pruebas . . . . .	10
7.4	Integración continua . . . . .	11
7.5	Seguimiento de errores/defectos . . . . .	11

## Índice de tablas

7.1	Tabla comparativa de herramientas de control de versiones. . . . .	10
7.2	Tabla comparativa de herramientas de compilación. . . . .	10
7.3	Tabla comparativa de herramientas de automatización de pruebas. . . . .	10
7.4	Tabla comparativa de herramientas de integración continua. . . . .	11
7.5	Tabla comparativa de herramientas de seguimiento de errores/defectos. . . . .	11

# 1. Introducción

En este documento, se realizará un análisis de las herramientas existentes para facilitar distintos aspectos de la gestión de cambios de un proyecto de desarrollo de software.

## 2. Control de versiones

### 2.1. Git

- **Nombre de la herramienta:** Git
- **URL:** <https://git-scm.com/>
- **Coste:** Gratuito.
- **Características destacadas:**
  1. Escrito en una combinación de Perl, C y varios scripts de shell.
  2. Descentralización, rápido, flexible y robusto.
  3. Sencillo de aprender a utilizar.
- **Limitaciones identificadas:**
  1. Gran número de archivos.
  2. Archivos de gran tamaño.
  3. Paquetes enormes.

### 2.2. Bazaar

- **Nombre de la herramienta:** Bazaar
- **URL:** <http://bazaar.canonical.com/en/>
- **Coste:** Gratuito.
- **Características destacadas:**
  1. Software Libre y Opensource.
  2. Desarrollado en Python.
  3. Versiones empaquetadas para la mayoría de distribuciones GNU/Linux, Mac OS X y MS Windows.
- **Limitaciones identificadas:** Tiene algunos bugs como
  1. No se puede mezclar una rama vacía.
  2. Bazaar debería tener una caché para versiones descargables.
  3. Problema con codificación ASCII.

### 2.3. Mercurial

- **Nombre de la herramienta:** Mercurial
- **URL:** <https://www.mercurial-scm.org/>
- **Coste:** Gratuito
- **Características destacadas:**
  1. Curva de aprendizaje más rápida que git.
  2. Gestión distribuida.
  3. Al hacer un cambio, se almacenan dichos cambios (no el fichero al completo).
- **Limitaciones identificadas:**
  1. Gran dependencia de Atlassian.
  2. Comunidad menos amplia que git.
  3. Al añadir extensiones, combinar características y funcionalidad es más difícil.

## 3. Compilación

### 3.1. Eclipse

- **Nombre de la herramienta:** Eclipse
- **URL:** <https://www.eclipse.org/>
- **Coste:** Gratuito.
- **Características destacadas:**
  1. Permite gestionar proyectos.
  2. Incluye depurador de código.
  3. Extensa colección de plugins.
- **Limitaciones identificadas:**
  1. Algunos plugins son difíciles de configurar, incluso se encuentran anticuados.
  2. Demasiadas versiones diferentes que pueden resultar confusas.
  3. Cantidad de recursos consumidos excesivos.

### 3.2. NetBeans

- **Nombre de la herramienta:** NetBeans
- **URL:** <https://netbeans.org/>
- **Coste:** Gratuito
- **Características destacadas:**
  1. Lenguaje multiplataforma.
  2. Permite desarrollar aplicaciones web dinámicas.
  3. Manejo automático de la memoria.
- **Limitaciones identificadas:**
  1. Lentitud a la hora de ejecutar aplicaciones.
  2. Requiere un intérprete.
  3. Algunas herramientas tienen un costo adicional.

### 3.3. Android Studio

- **Nombre de la herramienta:** Android Studio
- **URL:** <https://developer.android.com/studio/>
- **Coste:** Gratuito.
- **Características destacadas:**
  1. Se puede programar para diferentes versiones de Android.
  2. Compilación rápida.
  3. Ejecución del app en tiempo real gracias al emulador.
- **Limitaciones identificadas:**
  1. Entorno algo inestable.
  2. Los requisitos son un poco elevados para el equipo que lo utiliza.
  3. Gasta batería en consecuencia.

## 4. Automatización de pruebas

### 4.1. Selenium

- **Nombre de la herramienta:** Selenium
- **URL:** <https://www.seleniumhq.org/>

- **Coste:** Gratuito.
- **Características destacadas:**
  1. Viene con una serie de herramientas ya integradas.
  2. Ofrece distintas soluciones para atender distintos requisitos.
  3. Uso extensible por medio de plugins.
- **Limitaciones identificadas:**
  1. Solo se puede utilizar para probar aplicaciones de escritorio o otro tipo de software, solo web.
  2. No hay soporte garantizado disponible para Selenium, hay que apoyarse en la comunidad.
  3. No es posible realizar pruebas en imágenes.

#### 4.2. Apache JMeter

- **Nombre de la herramienta:** Apache JMeter
- **URL:** <https://jmeter.apache.org/>
- **Coste:** Gratuito.
- **Características destacadas:**
  1. Capacidad de carga y prueba de rendimiento de muchos tipos diferentes de aplicaciones/servidores/protocolos.
  2. Se pueden hacer usos de comandos nativos o scripts de shell.
  3. JMeter puede generar informes robustos y efectivos y puede ser visualizado con una gran cantidad de herramientas.
- **Limitaciones identificadas:**
  1. JMeter aún no tiene complemento de autocorrelación.
  2. Todas las instancias de JMeter deben cerrarse mientras se ejecuta una prueba de carga.
  3. En ocasiones, si al copiar un elemento utilizamos versiones de JMeter distintas, se producirán errores.

#### 4.3. JUnit

- **Nombre de la herramienta:** JUnit
- **URL:** <https://junit.org/junit5/>
- **Coste:** Gratuito.

- **Características destacadas:**

1. Integración con Android Studio.
2. Gran cantidad de documentación.
3. Soportado en gran cantidad de IDEs.

- **Limitaciones identificadas:**

1. Exclusivo para Java.
2. Exclusivo para test unitarios.
3. Encontramos limitaciones en su funcionalidad.

## 5. Integración continua

### 5.1. Travis IC

- **Nombre de la herramienta:** Travis IC

- **URL:** <https://travis-ci.org/>

- **Coste:** Gratuito para repositorios git públicos. Para empresas, tenemos precios desde 69\$ el más básico hasta 489\$ por mes.

- **Características destacadas:**

1. Facilidad de integración con repositorios git.
2. Fácil de utilizar.
3. Se puede vincular a correo electrónico para ver los resultados.

- **Limitaciones identificadas:**

1. Precio elevado.
2. La velocidad no es muy grande.
3. Versión gratuita exclusiva a repositorios públicos.

### 5.2. Bamboo

- **Nombre de la herramienta:** Bamboo

- **URL:** <https://es.atlassian.com/software/bamboo>

- **Coste:** Tiene prueba gratuita de 30 días con número de usuarios ilimitados. Después son 10\$ al año hasta 10 trabajos o 1100\$ con trabajos ilimitados.

- **Características destacadas:**

1. Disponible para cualquier lenguaje.

2. Puede ejecutar múltiples construcciones en paralelo.
3. Proporciona una API REST, esta proporciona información sobre el servidor, el estado actual de sus compilaciones, etc.

■ **Limitaciones identificadas:**

1. Márgenes de precios demasiado extremos, no podemos ajustar precio a necesidades concretas.
2. Si hacemos uso de Perforce, no se podrán ejecutar compilaciones en varios agentes remotos.
3. Complejo de usar.

### 5.3. Jira

■ **Nombre de la herramienta:** Jira

■ **URL:** <https://es.atlassian.com/software/jira>

■ **Coste:** Prueba gratuita durante 10 días. Después son 10\$ al mes hasta 10 usuarios o 7\$ al mes por cada usuario cuando son entre 11 y 100.

■ **Características destacadas:**

1. La asignación es granular.
2. Encontramos dos niveles de issues (tareas): issues y sub-issues.
3. Posee una gran cantidad de herramientas en torno a él.

■ **Limitaciones identificadas:**

1. Difícil de instalar.
2. Difícil de usar.
3. Excesivas características.

## 6. Seguimiento de errores/defectos

### 6.1. Sentry

■ **Nombre de la herramienta:** Sentry

■ **URL:** <https://sentry.io/welcome/>

■ **Coste:** Gratuito.

■ **Características destacadas:**

1. Impacto de despliegues en tiempo real.
2. Nos proporciona ayuda en ciertos momentos.



3. Fácil integración externa.

- **Limitaciones identificadas:**

1. Documentación incompleta.
2. Problemas de seguridad.
3. Requiere una dependencia binario para poder operar.

## 6.2. GitLab

- **Nombre de la herramienta:** GitLab

- **URL:** <https://about.gitlab.com/>

- **Coste:** Gratuito.

- **Características destacadas:**

1. Buena curva de aprendizaje.
2. Amplia comunidad.
3. Robusto y flexible.

- **Limitaciones identificadas:**

1. Almacenamiento del repositorio limitado a 10GB.
2. Interfaz relativamente lenta.
3. Frecuentes problemas técnicos con los repositorios.

## 6.3. Mantis Bug Tracker

- **Nombre de la herramienta:** Mantis Bug Tracker.

- **URL:** <https://www.mantisbt.org/>

- **Coste:** Gratuito.

- **Características destacadas:**

1. Se permite configurar la transición de estados.
2. Se puede especificar un número indeterminado de estados.
3. Permite introducir diferentes perfiles.

- **Limitaciones identificadas:**

1. Falta de documentación.
2. Interfaz anticuada.
3. Menos funcionalidad que otras herramientas.

## 7. Comparativa de herramientas

En esta sección se va a realizar una comparativa de herramientas, las cuales serán valoradas del 0 al 1, de tal forma que al multiplicar por los pesos siempre salgan valores entre 0 y 1. Destacar un criterio que será común en todas las tablas comparativas; el precio. Inicialmente, el proyecto se buscó que fuese económico y, por tanto, este criterio es especialmente importante.

### 7.1. Control de versiones

Criterio	Peso	Git	Bazaar	Mercurial
Precio	0.4	1	1	1
Robustez	0.3	0.6	0.45	0.6
Estabilidad	0.3	0.7	0.5	0.7
<b>Total</b>	<b>1</b>	<b>0.79</b>	<b>0.685</b>	<b>0.79</b>

Tabla 7.1: Tabla comparativa de herramientas de control de versiones.

### 7.2. Compilación

Criterio	Peso	Eclipse	NetBeans	Android Studio
Precio	0.4	1	0.7	1
Rapidez	0.3	0.25	0.25	0.8
Uso de recursos	0.3	0.6	0.5	0.2
<b>Total</b>	<b>1</b>	<b>0.655</b>	<b>0.505</b>	<b>0.7</b>

Tabla 7.2: Tabla comparativa de herramientas de compilación.

### 7.3. Automatización de pruebas

Criterio	Peso	Selenium	Apache JMeter	JUnit
Precio	0.4	1	1	1
Integración	0.35	0.4	0.7	0.8
Velocidad	0.25	0.6	0.3	0.7
<b>Total</b>	<b>1</b>	<b>0.69</b>	<b>0.72</b>	<b>0.855</b>

Tabla 7.3: Tabla comparativa de herramientas de automatización de pruebas.

#### 7.4. Integración continua

Criterio	Peso	Travis IC	Bamboo	Jira
Precio	0.4	0.5	0.6	0.5
Velocidad	0.5	0.3	0.75	0.45
Usabilidad	0.1	0.9	0.2	0.1
<b>Total</b>	<b>1</b>	<b>0.44</b>	<b>0.635</b>	<b>0.435</b>

Tabla 7.4: Tabla comparativa de herramientas de integración continua.

#### 7.5. Seguimiento de errores/defectos

Criterio	Peso	Sentry	GitLab	Mantis Bug Tracker
Precio	0.4	1	1	1
Comunidad y soporte	0.2	0.85	0.8	0.4
Funcionalidad	0.4	0.15	0.5	0.3
<b>Total</b>	<b>1</b>	<b>0.63</b>	<b>0.76</b>	<b>0.6</b>

Tabla 7.5: Tabla comparativa de herramientas de seguimiento de errores/defectos.