



Prácticas 7 y 8

Análisis de herramientas

Planificación y Gestión de Proyectos Informáticos

23 de diciembre de 2018

Felipe Peiró Garrido - felipepg@correo.ugr.es

Juan Carlos Serrano Pérez - jcsp0003@correo.ugr.es

Pedro Manuel Gómez-Portillo López - gomezportillo@correo.ugr.es

Índice

1. Análisis de herramientas	2
1.1. Herramientas de control de versiones	2
1.1.1. Git	2
1.1.2. Mercurial	2
1.1.3. Subversion	3
1.2. Herramientas de compilación	3
1.2.1. Maven	3
1.2.2. Gradle	4
1.2.3. Spring	4
1.3. Herramientas de automatización de pruebas	5
1.3.1. JUnit	5
1.3.2. Cucumber	5
1.3.3. Selenium	6
1.4. Herramientas de integración continua	6
1.4.1. Travis-CI	6
1.4.2. Jenkins-CI	7
1.4.3. Bamboo	7
1.5. Herramientas de seguimiento	8
1.5.1. Trello	8
1.5.2. Microsoft Planner	8
1.5.3. GitLab	9
2. Tablas comparativas	10
2.1. Herramientas de control de versiones	10
2.2. Herramientas de compilación	10
2.3. Herramientas de automatización de pruebas	11
2.4. Herramientas de integración continua	11
2.5. Herramientas de seguimiento	12

1. Análisis de herramientas

1.1. Herramientas de control de versiones

1.1.1. Git

- **URL.** <https://git-scm.com/>
- **Coste.** Open software
- **Características destacadas.**
 - La hemos estudiado durante la carrera y la conocemos
 - Permite trabajar sin estar conectado a internet
 - Está integrada con GitHub y permite trabajar con issues fácilmente
- **Limitaciones identificadas.**
 - Los repositorios privados en GitHub son de pago
 - Puede ser complejo para alguien que no lo haya usado nunca
 - Los desarrolladores no ven los cambios hasta que no se suben al servidor

1.1.2. Mercurial

- **URL.** <https://www.mercurial-scm.org/>
- **Coste.** Open source
- **Características destacadas.**
 - Repositorios privados gratuitos en BitBucket
 - Permite trabajar sin estar conectado a internet
 - Está integrada en BitBucket y permite funciones como sincronizar Slack o Trello
- **Limitaciones identificadas.**
 - Solo la hemos utilizado un par de veces
 - Puede ser complejo para alguien que no lo haya usado nunca
 - Los desarrolladores no ven los cambios hasta que no se suben al servidor

1.1.3. Subversion

- **URL.** <https://subversion.apache.org/>
- **Coste.** Open source
- **Características destacadas.**
 - Mucha documentación
 - Gestión centralizada
- **Limitaciones identificadas.**
 - No lo hemos utilizado nunca
 - Hay varias limitaciones respecto a los merges
 - Casi nadie lo usa

1.2. Herramientas de compilación

1.2.1. Maven

- **URL.** <https://maven.apache.org/>
- **Coste.** Open source
- **Características destacadas.**
 - Lo hemos utilizado durante la carrera
 - Permite descargar librerías automáticamente
 - Es muy fácil de configurar
- **Limitaciones identificadas.**
 - Puede dar problemas si se trabaja con dos versiones de la misma librería
 - Solo funciona para Java
 - Puede ser algo lento

1.2.2. Gradle

- **URL.** <https://gradle.org/>
- **Coste.** Open source
- **Características destacadas.**
 - Soporta varios lenguajes
 - Permite descargar librerías automáticamente
 - Permite distinguir entre testeo y despliegue
- **Limitaciones identificadas.**
 - Solo lo hemos utilizado un par de veces
 - Su configuración puede ser más compleja que sus alternativas

1.2.3. Spring

- **URL.** <https://spring.io/>
- **Coste.** Open source
- **Características destacadas.**
 - Mucha documentación
 - Soporta varios lenguajes
 - Permite trabajar con librerías web fácilmente
- **Limitaciones identificadas.**
 - Solo lo hemos utilizado un par de veces
 - Su configuración puede ser compleja
 - No es tan usado como sus alternativas

1.3. Herramientas de automatización de pruebas

1.3.1. JUnit

- **URL.** <https://junit.org/junit5/>
- **Coste.** Open source
- **Características destacadas.**
 - Tiene librerías en la mayoría de lenguajes
 - Lo hemos utilizado varias veces durante la carrera
 - Permite realizar operaciones antes y después de los test de forma nativa
- **Limitaciones identificadas.**
 - Solo soporta test unitarios
 - Puede quedarse corto para proyectos grandes

1.3.2. Cucumber

- **URL.** <https://cucumber.io/>
- **Coste.** Open source
- **Características destacadas.**
 - Permite seguir una metodología BDD¹
 - Gran flexibilidad para testear los requisitos
 - Se adapta al lenguaje natural
- **Limitaciones identificadas.**
 - No lo hemos utilizado en la carrera
 - Puede ser complicado adaptarse a esta nueva metodología
 - No tiene una configuración tan directa

¹Behaviour-Driven Development

1.3.3. Selenium

- **URL.** <https://www.seleniumhq.org/>
- **Coste.** 0€
- **Características destacadas.**
 - Centrado en el web-testing
 - Permite automatizar el testing una web a través del navegador
 - Mucho más rápido que hacerlo a mano
 - Tiene librerías en los lenguajes más usados
- **Limitaciones identificadas.**
 - La web del proyecto es solo una parte
 - Tiene que cargar la interfaz entera para acceder a cada página

1.4. Herramientas de integración continua

1.4.1. Travis-CI

- **URL.** <https://travis-ci.org/>
- **Coste.** 0€
- **Características destacadas.**
 - Permite el testeo de repositorios tras cada *push*
 - Permite la integración con otros servicios como Heroku
 - Su interfaz web es muy usable
- **Limitaciones identificadas.**
 - Es muy lento a la hora de testear
 - Tiene una cola de espera, y a veces puede tardar minutos en empezar

1.4.2. Jenkins-CI

- **URL.** <https://jenkins.io/>
- **Coste.** Open source
- **Características destacadas.**
 - Muy buena documentación
 - Fácil instalación y configuración
 - Su funcionalidad es fácilmente extendible
- **Limitaciones identificadas.**
 - No lo hemos usado antes
 - No tiene versión web nativa

1.4.3. Bamboo

- **URL.** <https://es.atlassian.com/software/bamboo>
- **Coste.** Tiene varias tarifas
- **Características destacadas.**
 - Está integrado con el resto de servicios de Atlassian
 - Soporta varios lenguajes
- **Limitaciones identificadas.**
 - No es gratuito
 - No tiene versión web
 - Es más difícil de configurar que sus alternativas

1.5. Herramientas de seguimiento

1.5.1. Trello

- **URL.** <https://trello.com/>
- **Coste.** Tableros limitados por 0€, más opciones de pago
- **Características destacadas.**
 - Permite dividir cada tablero en columnas y añadirles tarjetas
 - Interfaz web clara y simple
 - Se pueden asignar personas y plazos a las tareas
- **Limitaciones identificadas.**
 - Funcionalidad muy limitada
 - Para agregar otros servicios es necesario pagar
 - Su uso sólo es sostenible para proyectos pequeños

1.5.2. Microsoft Planner

- **URL.** <https://tasks.office.com/>
- **Coste.** De 4 a 10/mes
- **Características destacadas.**
 - Mucho más completo que sus alternativas
 - Permite integrar repositorios y referenciarlos
 - Permite integrar el resto de herramientas de Microsoft
- **Limitaciones identificadas.**
 - La interfaz es un poco lenta
 - No tiene integración con otros servicios
 - Es código cerrado

1.5.3. GitLab

- **URL.** <https://gitlab.com/>
- **Coste.** De 0 a 100€/mes
- **Características destacadas.**
 - Soporta git
 - Mucha documentación
 - Es de código abierto
- **Limitaciones identificadas.**
 - No permite integrar otros servicios
 - Es el más caro
 - Quizá no sea tan completo como el de Microsoft

2. Tablas comparativas

2.1. Herramientas de control de versiones

Criterio	Peso	Git	Mercurial	Subversion
Conocida	30%	1	0.5	0
Soporta offline	40%	1	1	0
Integrada en un servicio web	30%	1	1	0.5
Total	100%	1	0.85	0.15

2.2. Herramientas de compilación

Criterio	Peso	Maven	Gradle	Spring
Conocida	20%	1	0.5	0.5
Fácil configuración	50%	0.75	0.5	0.5
Rapidez	30%	0.5	0.75	0.75
Total	100%	0.7	0.5	0.35

2.3. Herramientas de automatización de pruebas

Criterio	Peso	JUnit	Cucumber	Selenium
Soporta varios tipos de testing	30%	0	0.5	0
Se adapta al lenguaje natural	40%	0	1	0
Soporta navegadores web	30%	0	0	1
Total	100%	0	0.55	0.3

2.4. Herramientas de integración continua

Criterio	Peso	Travis-CI	Jenkins-CI	Bamboo
Rapidez	30%	0.25	0.5	0.75
Facilidad de configuración	40%	1	0.5	0.25
Integración nativa con repo.	30%	0.25	0.25	1
Total	100%	0.7	0.55	6.5

2.5. Herramientas de seguimiento

Criterio	Peso	Trello	Planner	GitLab
Claridad	40%	0.75	1	0.75
Asignar recursos y tareas	30%	0.5	1	1
Integración con herramientas externas	30%	0.5	0.75	0.75
Total	100%	0.6	7.2	0.825