



Universidad de Granada

Máster profesional en Ingeniería Informática

Planificación y Gestión de Proyectos Informáticos

Práctica 8: Selección de herramientas 1: Fichas de las
distintas herramientas estudiadas.

Autores:

Jose Angel Diaz García
Manuel Jesús García Manday

Índice

	2
8 Selección de Herramientas 1:	3
8.1 Herramientas de control de versiones.	3
8.2 Herramientas de compilación.	5
8.3 Herramientas de automatización de pruebas.	6
8.4 Herramientas de integración continua.	8
8.5 Herramientas de seguimiento de errores/defectos.	9

8 Selección de Herramientas 1

En este capítulo de la especificación del proyecto a realizar en el museo Caja Granada haremos un estudio de las herramientas que podrían usarse para distintas áreas de desarrollo. Posteriormente, se hará un estudio detallado de los factores que lleva a la elección de una u otra herramienta.

8.1 Herramientas de control de versiones.

Herramienta	Git
URL	https://git-scm.com
Coste	En algunas herramientas como Bitbucket gratuito para repositorios privados incluso.
Pros	<p>1-Es conocido por todos los miembros del equipo.</p> <p>2-Ofrece una gran comunidad sobre ella que puede servir de referencia fácilmente y accesible.</p> <p>3-Conexión no es necesaria salvo para sincronizar.</p>
Contras	<p>1-Si usamos archivos de gran tamaño y muy numerosos puede ofrecer problemas.</p> <p>2-No guarda directorios vacíos, cosa que puede ser interesante en algunos casos.</p> <p>3-No permite asociar permisos en función de usuarios o rutas.</p>

Herramienta	Subversion
URL	https://subversion.apache.org
Coste	Gratis incluso para repositorios gratuitos en algunos sistemas.
Pros	<p>1- Permite archivos de gran tamaño sin problema ya que no tiene redundancia.</p> <p>2- Permite directorios vacíos.</p> <p>3- Permite configurar accesos en función de la ruta.</p>
Contras	<p>1- Es necesaria una conexión a internet para cada acceso.</p> <p>2- Es menos conocido que git entre los miembros del equipo.</p> <p>3-Si el repositorio central se pierde, todo se pierde pues solo hay un repositorio para el proyecto.</p>

Herramienta	Mercurial
URL	https://www.mercurial-scm.org
Coste	Gratuito
Pros	<p>-Es distribuido por lo que no usa el modelo "cliente-servidor" lo que para nosotros es un beneficio.</p> <p>-Es más fácil de aprender que Git.</p> <p>-Ofrece una muy buena documentación.</p>
Contras	<p>-No podemos hacer merge de dos padres.</p> <p>-Menos potencia.</p> <p>-Basado en extensiones.</p>

8.2 Herramientas de compilación.

Herramienta	Ant
URL	http://ant.apache.org/
Coste	Gratuito
Pros	<ul style="list-style-type: none">-Ofrece un gran control del proceso de compilado.-Permite añadir cantidad de plugins.-Muy semejante a make por lo que la curva de aprendizaje puede ser menor.
Contras	<ul style="list-style-type: none">-Está basada en ficheros xml lo que hace que rápidamente pueden ser inmanejables incluso en proyectos pequeños.-No tiene sistema de control de dependencias.-Basado en procedimientos.

Herramienta	Maven
URL	https://maven.apache.org/
Coste	Gratuito
Pros	<ul style="list-style-type: none">-Tiene tareas predefinidas muy útiles.-Puede bajar dependencias.-Permite proyecto multi-modulo.
Contras	<ul style="list-style-type: none">-Las dependencias entre versiones de la misma librería pueden dar lugar a problemas.-Escribir scripts de compilado personalizados es complicado.-El lenguaje es XML lo que tiene de nuevo como en Ant, grandes problemas.

Herramienta	Gradle
URL	http://www.gradle.org/
Coste	Gratuito
Pros	<ul style="list-style-type: none"> -El DSL que usa es Groovy, lo que le aporta grandes ventajas frente al XML. -Los scripts de compilado son simples y cortos. -Es extensible via plugins y ofrece gestión de dependencias.
Contras	<ul style="list-style-type: none"> -Puede haber problemas con la herencia ya que Gradle no tiene el concepto de directorio padre.

8.3 Herramientas de automatización de pruebas.

Herramienta	Telerik Test Studio
URL	http://www.telerik.com/teststudio
Coste	Desde 79€ al mes o licencia completa por 2499\$
Pros	<ul style="list-style-type: none"> -Casa con muchos lenguajes distintos, entre ellos los usados nuestro proyecto. -Colaboración entre QAs y desarrolladores muy fluida. -Ofrece integración continua perfectamente integrada lo que facilita el no tener que usar otra herramienta adicional.
Contras	<ul style="list-style-type: none"> -Si el nivel de profundidad de detalle a la hora de modificar nuestro proceso de test es muy elevado conlleva un esfuerzo mayor de lo habitual el configurar todo correctamente.

Herramienta	Selenium
URL	http://www.seleniumhq.org
Coste	-Gratuito pero con costes adicionales por funcionalidades que a muy seguro necesitaremos.
Pros	<ul style="list-style-type: none"> -Ejecuta test en paralelo en múltiples OS. -Integración entre el equipo de calidad y desarrollo muy fácil. -Tiene una gran cantidad de frameworks de test.
Contras	<ul style="list-style-type: none"> -Es de código abierto, por lo que en caso de problemas debes atender a la comunidad. -No tiene capacidad de obtener reportes. -Debes conocer al menos uno de los lenguajes soportados muy bien para poder automatizar el proceso de test.

Herramienta	Appium
URL	http://appium.io
Coste	Gratuito.
Pros	<ul style="list-style-type: none"> -Soporta iOS y Android. -Soporta una gran cantidad de los lenguajes de programación más comunes. -No requiere APK para ser usada.
Contras	<ul style="list-style-type: none"> -No soporta comparación de imágenes. -Documentación muy básica. -Aún en desarrollo por lo que puede estar sujeta a cambios importantes.

8.4 Herramientas de integración continua.

Herramienta	Bamboo
URL	http://www.atlassian.com/software/bamboo
Coste	Desde 10\$ para equipos pequeños.
Pros	<ul style="list-style-type: none">-Muy buena integración con Git.-Ofrece mucho detalle sobre los errores que puedan suceder.-Se pueden modificar los comentarios que recibimos acerca de cada tarea y recibirlos por mail, RSS además de otras vías.
Contras	<ul style="list-style-type: none">-No es gratis.-El tiempo de arreglar ciertos problemas puede ser muy elevado.

Herramienta	Jenkins
URL	https://jenkins.io/
Coste	Gratuito.
Pros	<ul style="list-style-type: none">-Es gratis.-Gran cantidad de plugins.-Una buena documentación tanto oficial como de la comunidad.
Contras	<ul style="list-style-type: none">-Es más complicado de comenzar a usar que las demás herramientas estudiadas.-Los plugins le restan estabilidad ya que pueden variar y hacer que haya fallos de compatibilidad.

Herramienta	Travis
--------------------	--------

URL	https://travis-ci.org/
Coste	Gratis.
Pros	<ul style="list-style-type: none"> -Es gratis. -Es una herramienta de integración continua y testing al mismo tiempo. -Notifica por mail los problemas.
Contras	<ul style="list-style-type: none"> -No soporta Android. -Para proyectos muy grandes puede dar problemas.

8.5 Herramientas de seguimiento de errores/defectos.

Herramienta	Mantis
URL	http://www.mantisbt.org/
Coste	Gratis
Pros	<ul style="list-style-type: none"> -Puede usarse en dispositivos móviles tanto iOS como Android lo que lo hace interesante para nuestro proyecto. -Simple e intuitivo. -Múltiples usuarios pueden colaborar en el proceso de bug tracking.
Contras	<ul style="list-style-type: none"> -Para proyectos muy grandes puede ver su potencia limitada. -La opción de exportar resultados para reportes es algo simple. -La documentación no está disponible en la instalación local.

Herramienta	Trac
--------------------	------

URL	http://trac.edgewall.org/
Coste	Gratis
Pros	<ul style="list-style-type: none"> -Se puede usar en sistemas operativos fuera de los tres más extendidos, tales como FreeBSD o NetBSD. -Puede usarse en dispositivos móviles tanto iOS como Android lo que lo hace interesante para nuestro proyecto. -Tiene una gran cantidad de plugins para hacerlo más versátil. -Integra sistemas de control de versiones como git.
Contras	<ul style="list-style-type: none"> -No puede trabajar con dos proyectos distintos en conjunción sin plugins adicionales. -Centrada en issue tracking.

Herramienta	Bugzilla
URL	bugzilla.org
Coste	Gratis
Pros	<ul style="list-style-type: none"> -Puede usarse en dispositivos móviles tanto iOS como Android lo que lo hace interesante para nuestro proyecto. -Centrada en bugtracking. -Tiene una gran comunidad y documentación oficial para solventar posibles errores.
Contras	<ul style="list-style-type: none"> -Es complicado de configurar y usar ofreciendo una curva de aprendizaje elevada. -En proyectos muy grandes es complicado de navegar y atender a los distintos bugs.

