

PLANIFICACIÓN Y GESTIÓN DE PROYECTOS

PRÁCTICA 9. GESTIÓN DE CAMBIOS (HERRAMIENTAS)

Lidia Sánchez Mérida.

Fernando Roldán Zafrá.

Cristina de la Caridad Vaillant Valdéz.

Herramientas de control de versiones

Git.	Mercurial.	Helix Core.
URL Git.	URL Mercurial SCM.	URL Helix Core.
<ul style="list-style-type: none"> Suscripción individual: gratuita. Suscripción <i>Pro</i>: 7\$/mes. Suscripción <i>Equipos</i>: 9\$/mes por miembro. Aprendizaje: seminario impartido por los gestores del proyecto durante tres días. Coste: $40\text{€}/\text{día}/\text{persona} * 3 \text{ días} * 11 \text{ trabajadores} = \mathbf{1.320\text{€}}.$ 	<ul style="list-style-type: none"> Suscripción: gratuita. Aprendizaje: seminario impartido por los gestores del proyecto durante tres días. Coste: $40\text{€}/\text{día}/\text{persona} * 3 \text{ días} * 11 \text{ trabajadores} = \mathbf{1.320\text{€}}.$ 	<ul style="list-style-type: none"> Suscripción de hasta cinco usuarios y 1GB de almacenamiento: gratuita. Suscripción <i>standard</i> para más de seis usuarios y 1GB por usuario: 19\$/usuario/año. Suscripción <i>premium</i> con 5GB por usuario y para más de seis usuarios: 29\$/usuario/año. Aprendizaje: seminario online de tres días de duración con cuatro horas diarias: 3000\$ = 2691,60€.
<ul style="list-style-type: none"> Desarrollo distribuido: cada desarrollador dispone de una copia completa del repositorio general. <i>Pull requests</i>: proporciona un 	<ul style="list-style-type: none"> Comandos funcionales y sencillos. Cada uno realiza un acción unívoca y diferente. Implementación de nuevos comandos sin necesidad de 	<ul style="list-style-type: none"> Permite almacenar en un único repositorio ficheros de diversa naturaleza tales como ejecutables, documentos, ficheros multimedia,

<p>control sobre la integración de código nuevo a la aplicación, de modo que el jefe del equipo sea capaz de analizar la calidad del nuevo desarrollo y decidir si se puede incorporar a producción o no.</p> <ul style="list-style-type: none"> • Diversidad para aplicar git no solo a proyectos de desarrollo de código, si no también para <i>marketing</i>, recursos humanos, diseño, entre otros. 	<p>modificar el código fuente de la herramienta.</p> <ul style="list-style-type: none"> • Gran facilidad para aprender a manejar esta herramienta desde cero. 	<p>además de código.</p> <ul style="list-style-type: none"> • Rápido y eficiente escalado de usuarios. • Configuración de políticas de mezcla de ramas particulares para cada una de las partes de una aplicación, como el <i>back-end</i> y el <i>front-end</i>.
<ul style="list-style-type: none"> • Se necesitan abrir varias ramas para permitir el desarrollo en paralelo. • Algunas instrucciones pueden provocar confusión o no ser intuitivas en relación a la acción que desarrollan. • La interfaz asociada está bastante incompleta porque está más enfocada a utilizarla mediante comandos. 	<ul style="list-style-type: none"> • Las extensiones propias solo se pueden implementar en Python. • No dispone de filtros de paginación para salidas de gran tamaño como registro de modificaciones, <i>logs</i>, entre otros. • No permite realizar modificaciones por etapas, es decir, todos los cambios se realizan de una sola vez. 	<ul style="list-style-type: none"> • Es difícil de aprender a utilizar si anteriormente has utilizado otras herramientas tales como <i>Git</i> o <i>BitBucket</i>. • Ciertos parámetros de configuración son difíciles de establecer, como clonar repositorios, acceso por SSH, entre otros. • No permite crear subramas específicas e independientes de la rama principal.

Criterio	Peso	Git.	Mercurial.	Helix Core.
Creación de ramas y flexibilidad de los <i>commits</i> .	35%	9	6	7

Mantener la historia de ficheros y directorios.	10%	7	9	8
Facilidad para clonar un repositorio.	15%	10	10	7
Facilidad para establecer diversos permisos de un repositorio para cada miembro.	15%	10	5	6
Facilidad de uso y aprendizaje.	25%	7	9	4
Total	100%	8.6	7.5	6.2

Tal y como se puede comprobar tanto en las fichas individuales de cada herramienta como en el anterior estudio cuantitativo, la herramienta que mejor se adapta a nuestro proyecto y equipo es **git**. Si bien dispone de las suscripciones más baratas de las tres herramientas, también destaca la ventaja de que los gestores del proyecto la conocen perfectamente y tienen bastante experiencia en su uso, por lo que pueden ser ellos mismos los que impartan un seminario a la plantilla, en caso de que estos no hayan trabajado con ella.

Del mismo modo, en la mayoría de las funcionalidades que ofrece dispone de diversas configuraciones, como por ejemplo los diferentes permisos que se pueden establecer para un usuario dado un repositorio, así como su amplia documentación y comunidad que se encuentra detrás de git.

Herramientas de compilación

PyCharm.	Jenkins.	CMake.
URL PyCharm.	URL Jenkins.	URL CMake.
<ul style="list-style-type: none"> Suscripción por usuario: 199€. Aprendizaje: libro más curso <i>online</i> de 7 horas por 79\$ o 70.92€/persona. 	<ul style="list-style-type: none"> Gratuito sin incluir despliegue en la nube. Aprendizaje: curso <i>online</i> de seis horas por 10.99€/persona. 	<ul style="list-style-type: none"> Gratuito. Aprendizaje: libro de introducción a la herramienta por 49.30\$ o 44.27€.

<ul style="list-style-type: none"> • Incluye un editor inteligente de código, un depurador, testeador, <i>profiler</i>, herramientas de control de versiones y de bases de datos, acceso a <i>Docker</i> y desarrollo remoto en <i>Vagrant</i>. • Soporte para desarrollo web incluyendo <i>frameworks</i> como <i>Django</i>. • Incluye una API en la que los desarrolladores pueden implementar nuevas funcionalidades. 	<ul style="list-style-type: none"> • Ofrece soporte para herramientas de integración, control de versiones, despliegue, entre otros. • Es <i>open source</i>. • Fácil configuración mediante la interfaz web. • Gran cantidad de extensiones. 	<ul style="list-style-type: none"> • Herramienta de construcción, ejecución y testeo multiplataforma y <i>open source</i>. • Capaz de generar ficheros de construcción, como los <i>makefile</i>, que son independientes de los ficheros de configuración y del entorno de ejecución. • Se adapta a diferentes editores de texto. • Gran soporte por parte de la comunidad.
<ul style="list-style-type: none"> • Hace uso de una cantidad masiva de memoria de almacenamiento, que incrementa conforme el proyecto es más complejo. • Consume bastantes recursos computacionales, especialmente en problemas de análisis de datos. • Pese a incluir desarrollo web no incluye soporte para lenguajes populares en este ámbito, como Java. 	<ul style="list-style-type: none"> • Falta de funcionalidades en la interfaz de la herramienta. • No provee el servicio de realizar y/o restaurar copias de seguridad o versiones anteriores del código. • Escaso soporte de la comunidad y de mantenimiento que provoca que debas tú mismo solucionar los errores o <i>bugs</i> que tenga la herramienta. 	<ul style="list-style-type: none"> • Es una herramienta de programación a bajo nivel y por lo tanto son necesarios conocimientos avanzados de informática. • La sintaxis es algo complicada de aprender. • Compilación más lenta que el resto de herramientas.

Criterio	Peso	PyCharm	Jenkins	CMake
----------	------	---------	---------	-------

Soporte para varios lenguajes, entre ellos <i>Python</i> y lenguajes web.	20%	3	10	7
Facilidad para generar ejecutables independientes de la plataforma.	20%	10	10	10
Facilidad de integración con herramientas de testeo y de integración continua.	20%	3	10	8
Interfaz completa y amigable.	10%	9	6	2
Facilidad de uso y aprendizaje.	30%	8	8	4
Total	100%	6.5	9	6.4

En base al estudio cuantitativo anterior podemos determinar que la herramienta de compilación que ofrece las mejores prestaciones es **Jenkins**. Destaca por la inclusión de un editor de textos inteligente así como diversas funcionalidades de integración continua, testeo y despliegue en la nube. Es de código abierto, lo que supone una ventaja y una desventaja a partes iguales, ya que si bien es gratuita, por lo general, las herramientas con este tipo de licencias no disponen de un buen soporte para el arreglo de errores. Sin embargo, al mismo tiempo ofrece la posibilidad de modificar ciertos aspectos y adaptarlos a tu proyecto así como el desarrollo de extensiones útiles para el producto se desea desarrollar.

Herramientas de automatización de pruebas

TestArchitect	AutoIt	Sikulix
URL TestArchitect	URL AutoIt	URL Sikulix

<ul style="list-style-type: none"> • Herramienta gratuita. • Coste de aprendizaje: curso <i>online</i> de formación de 13 horas : 199.99€/persona. 	<ul style="list-style-type: none"> • Herramienta gratuita. • Coste de aprendizaje: curso <i>online</i> de aprendizaje de 1,5 horas: 24,99€/persona. 	<ul style="list-style-type: none"> • Herramienta gratuita. • Coste de aprendizaje: curso <i>online</i> de aprendizaje de 3 horas: 99,99€/persona.
<ul style="list-style-type: none"> • <i>TestArchitect</i> es compatible con herramientas de integración continua haciendo que los tests desarrollados mediante esta herramienta se ejecuten al realizar las pruebas de integración. • Permite ejecutar tests de forma paralela, reduciendo el tiempo y aumentando la eficiencia computacional de ejecutar los tests. • No necesita escribir código para realizar los tests ya que se realizan sobre grabaciones de acciones. 	<ul style="list-style-type: none"> • Los tests se realizan mediante simulaciones de movimientos del ratón y pulsación de teclas del teclado. • Proporciona un editor propio con corrección de sintaxis en el código mejorada. • Los tests se ejecutan mediante archivos ejecutables. 	<ul style="list-style-type: none"> • Está basado en imágenes, por lo que no se necesita conocer las características de los componentes de la aplicación para testearla. • Interacciona con la interfaz por medio del movimiento del ratón y de pulsaciones del teclado con elementos identificados de la interfaz. • Desarrollado en Python, por lo que además de ser compatible con este lenguaje lo es también con Java y Ruby.
<ul style="list-style-type: none"> • No se pueden paralelizar los tests en la misma máquina, sin embargo se pueden paralelizar en máquinas diferentes lo que conlleva un gasto mayor de recursos. • No se puede realizar un control de 	<ul style="list-style-type: none"> • Esta herramienta esta enfocada a su uso en sistemas Windows con lo que no permite usarlo en otros sistemas operativos. • No permite la ejecución de test en paralelo. • Compatible solo con su sintaxis propia, la 	<ul style="list-style-type: none"> • No se puede paralelizar la ejecución de los tests. • Los tests son dependientes de las imágenes de la interfaz, por lo que si estas cambian también hay que cambiar los tests. • Necesita que la

<p>versiones con <i>git</i> o con ninguna otra herramienta externa al programa, ya que este proporciona su propio control de versiones. Esto puede provocar problemas de consistencia en caso de que en el proyecto se utilice <i>git</i>, puesto que existirían dos controles de versiones diferentes.</p> <ul style="list-style-type: none"> El número de lenguajes permitidos para realizar acciones es muy limitado, siendo estos Java, Python y C#. 	cual es un estilo parecido a BASIC.	aplicación esté visible en la pantalla para testearla.
---	-------------------------------------	--

Criterio	Peso	TestArchitect	Autolt	Sikulix
Soporte con varios lenguajes, entre ellos Python.	30%	7	2	5
Uso entre la comunidad de desarrolladores .	20%	7	4	3
Precio.	10%	10	10	10
Interfaz completa y amigable.	10%	8	7	4
Facilidad de uso y aprendizaje.	30%	4	8	6

Total	100%	6.5	5.5	5.3
--------------	-------------	------------	------------	------------

Como se ha podido ver en base al estudio anterior, la herramienta de automatización de pruebas que se va a utilizar para este proyecto va a ser **TestArchitect**. Uno de los factores principales para elegir esta herramienta es la facilidad de uso y configuración, junto con que es la más extendida llevando esto a que existan una gran cantidad de tutoriales, problemas resueltos, soporte, etc.

Herramientas de integración continua

Jenkins	Travis	Circle CI
URL Jenkins	URL Travis	URL CircleCi
<ul style="list-style-type: none"> • Gratuita. • Coste del aprendizaje: ya ha sido explicado anteriormente. 	<ul style="list-style-type: none"> • Gratis para proyectos open source. • Versión Bootstrap: 69\$/mes. • Versión Startup: 129\$/mes. • Versión Small Business: 249\$/mes. • Versión Premium: 489\$/mes. • Coste de aprendizaje: seminario impartido por los gestores del proyecto durante dos días: 40€/día/persona * 2 días * 11 trabajadores = 880€. 	<ul style="list-style-type: none"> • Versión básica: gratuita. • Versión performance: 30\$/mes. • Versión Custom: 3000\$/mes. • Coste de aprendizaje: seminario impartido por los gestores del proyecto durante dos días: 40€/día/persona * 2 días * 11 trabajadores = 880€
<ul style="list-style-type: none"> • Gran cantidad de <i>plugins</i>. • Trabaja de forma conjunta con las herramientas de control de versiones más comunes. • Si un plugin no existe, puede ser programado de forma fácil. 	<ul style="list-style-type: none"> • Puede ejecutar test de forma paralela. • Al contrario que otras herramientas de integración continua, Travis implementa la "build matrix", una herramienta que proporciona la posibilidad de 	<ul style="list-style-type: none"> • Permite la ejecución de tests en paralelo. • Integración con IOS. • Permite configurar la RAM y CPU para cada test.

	<p>realizar test con diferentes versiones de lenguajes y paquetes.</p> <ul style="list-style-type: none"> • Compatibilidad con un gran conjunto de lenguajes de programación. 	
<ul style="list-style-type: none"> • Jenkins no provee suficiente información sobre el ciclo del despliegue, que aunque hay plugins que aportan más información, esta sigue sin ser suficiente. • Requiere mantener un servidor dedicado y un empleado que lo gestione. • En el caso de proyectos con varios integrantes, es muy difícil mantener el control sobre quién ha realizado qué cambios ya que no ofrece visión sobre los cambios individuales de cada miembro del grupo. 	<ul style="list-style-type: none"> • No permite un gran nivel de personalización, al contrario que otras alternativas. • No incluye servicio gratuito para proyectos privados, al contrario que otras herramientas del mismo tipo. • En ocasiones, la ejecución de los tests puede ser muy lenta. 	<ul style="list-style-type: none"> • El número de lenguajes de programación con los que es compatible no es demasiado amplio. • Al tratarse de un servicio en la nube, es susceptible a caídas del servicio o a que debido a actualizaciones, deje de dar soporte a alguna de las imágenes que se utilizan en el proyecto. • <i>CircleCi</i> permite la ejecución de test en un entorno Linux, sin embargo, no permite realizar test en entornos basados en sistemas operativos <i>Windows</i>.

Criterio	Peso	Jenkins	Travis	CircleCi
Facilidad de uso y aprendizaje	30%	6	8	10
Uso entre la comunidad de desarrolladores	20%	7	9	10
Precio y costes	10%	10	3	5

Compatibilidad con lenguajes de programación	10%	8	9	6
Facilidad de configuración	15%	8	7	8
Velocidad de ejecución de los tests	15%	7	6	9
Total	100%	7.25	7.35	8.65

En este caso y al contrario de lo que se eligió como herramienta de compilación elegiremos **CircleCi** como herramienta de integración continua. Los principales factores que han llevado a esta decisión, tal y como se ve en el estudio anterior, son su facilidad de uso, la gran cantidad de desarrolladores que la utilizan y la velocidad de ejecución de los tests.

Herramientas de seguimiento de errores/defectos

Bugzilla	Redmine	Trac
URL Bugzilla	URL Redmine	URL Trac
<ul style="list-style-type: none"> • Gratuita. • Coste de aprendizaje: curso <i>online</i> de formación en la plataforma Udemy de 23 horas por 12.34€/persona*11 trabajadores=135,74€. 	<ul style="list-style-type: none"> • Gratuita. • Coste de aprendizaje: puede ser mediante un curso inicial básico gratuito en la plataforma easyredmine por 30 días y luego para profundizar más posibilita la opción de un curso <i>online</i> de formación por un mes 58€/persona*11 trabajadores=638€. 	<ul style="list-style-type: none"> • Gratuita. • Coste de aprendizaje: seminario impartido por los gestores, que previamente estudiarán la herramienta mediante la documentación de la página oficial de la herramienta. Este curso durará dos días y por ello su coste será 40€/día/persona * 2 días * 11 trabajadores = 880€.
<ul style="list-style-type: none"> • Está desarrollado por la Fundación Mozilla, permite a 	<ul style="list-style-type: none"> • Nos proporciona unas funciones integradas para 	<ul style="list-style-type: none"> • Se trata de un proyecto de código abierto a base de

<p>sus usuarios registrar y rastrear errores en sus aplicaciones de manera efectiva. Está desarrollado en Perl.</p> <ul style="list-style-type: none"> • Es una herramienta robusta y con muchas funciones, como capacidad de búsqueda avanzada, listas de errores en varios formatos, informes programados, detección automática de errores duplicados. • También tiene la virtud de archivar, modificar, enviar los errores por correo electrónico, seguimiento de tiempo, sistema de solicitudes, visor de parches, y archivos adjuntos y comentarios. 	<p>gestionar proyectos bien desarrolladas y muy útiles.</p> <ul style="list-style-type: none"> • Redmine admite múltiples proyectos, seguimiento de problemas, es compatible con opciones de control de versiones, tiene un control de acceso flexible basado en roles, calendario y diagramas de Gantt que te ayudan a la hora de representar visualmente los proyectos. • Tiene feeds, notificaciones por correo electrónico, seguimiento de tiempos, wiki, foros, etc... 	<p>Python.</p> <ul style="list-style-type: none"> • Es compatible con múltiples plataformas como Mac OS X, Windows, Linux y UNIX. • Permite enlazar información entre una base de datos de errores de software, un sistema de control de versiones y el contenido de un wiki. • Sirve como interfaz web de un sistema de control de versiones como Subversion, Git, Mercurial, Bazaar o Darcs. • Utiliza un sistema de plantillas web propio llamado Genshi.
<ul style="list-style-type: none"> • El programa es gratuito y <i>opensource</i> pero el soporte técnico y la ayuda a empresas constituyen un servicio de pago. • Carece de características de personalización. • Funciona lentamente. • La navegación de errores es algo compleja. • La IU es poco atractiva, 	<ul style="list-style-type: none"> • La navegación es algo complicada para los nuevos usuarios. • Es imposible transferir la tarea a otro miembro del equipo para que este la finalice. • No hay cadenas de tareas lógicas. No puede especificar que una tarea debe comenzar después de que se haya realizado la anterior. 	<ul style="list-style-type: none"> • No se pueden manejar múltiples proyectos. • Funcionalidad limitada si no utilizas todos sus complementos. • Instalación compleja.

desactualizada e ineficiente.		
-------------------------------	--	--

Criterio	Peso	Bugzilla	Redmine	Trac
Soporte para varios lenguajes, entre ellos <i>Python</i> y lenguajes web.	20%	6	7	10
Facilidad para generar ejecutables independientes de la plataforma.	20%	5	6	7
Facilidad de integración con herramientas de testeo y de integración continua.	20%	10	10	10
Interfaz completa y amigable.	10%	2	2	5
Facilidad de uso y aprendizaje.	30%	3	3	3
Total	100%	5.3	5.7	6.8

Se puede comprobar tanto en la tabla comparativa de cada herramienta como en el anterior estudio cuantitativo que la herramienta que mejor se adapta a nuestro proyecto y equipo es **Trac**. Además de ser un proyecto de código abierto a base de Python, no es solo un software de seguimiento de errores sino también como una plataforma de proyectos de seguimiento de problemas y desarrollo de código general. Además soporta de múltiples plataformas: Linux, Unix, Mac OS X, Windows, etc. Aporta opciones de gestión de usuarios y tiene un gran número de plugins.

Conclusiones

Nuestro proyecto empleará las siguientes herramientas de software que fueron

seleccionadas anteriormente con el objetivo de facilitar los distintos aspectos de la gestión de cambios durante el desarrollo del proyecto.

- Herramienta para el control de versiones: ***Git***.
- Herramienta de compilación: ***Jenkins***.
- Herramienta de automatización de pruebas: ***TestArchitect***.
- Herramientas de integración continua: ***CircleCi***.
- Herramienta para el seguimiento de errores/defectos: ***Trac***.