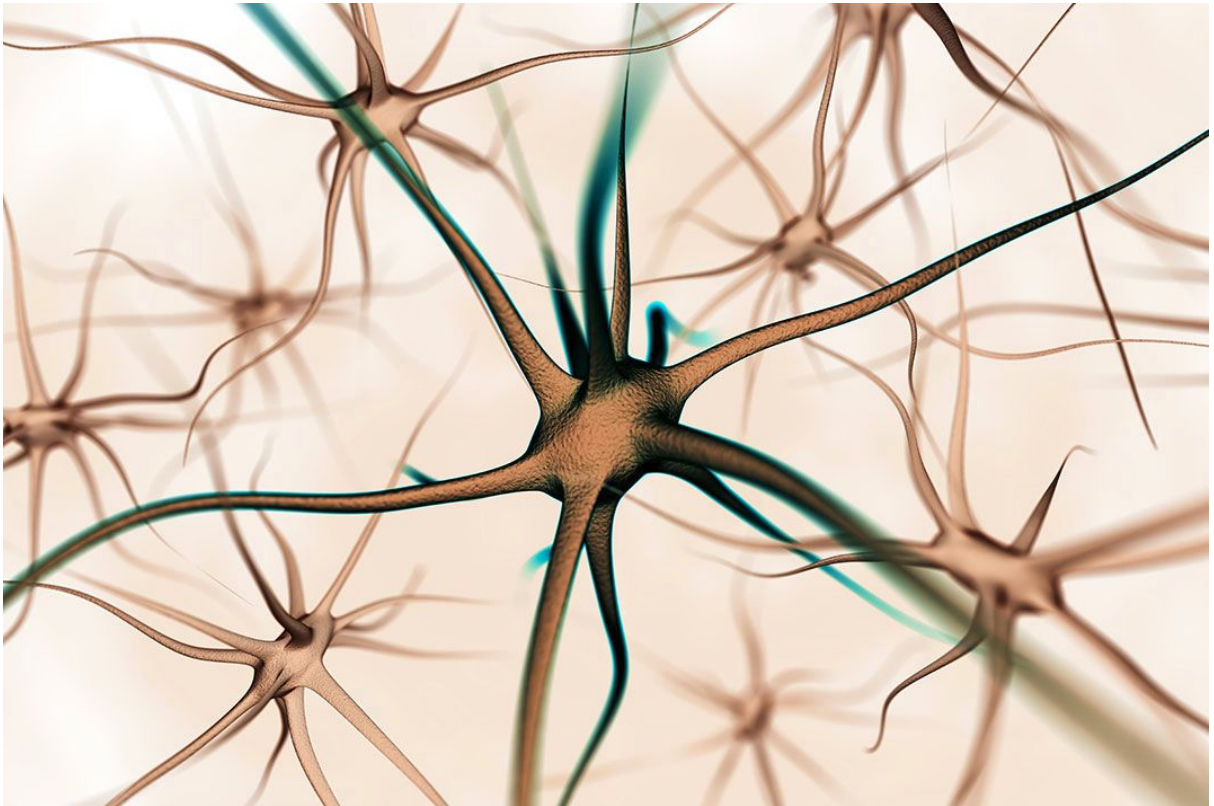


Herramientas

Práctica 9



Integrantes:

- **Alejandro Núñez Pérez**
- **María Matilde Cabrera González**

ÍNDICE:

Herramientas existentes	3
Herramientas de control de versiones	3
Herramientas de compilación	4
Herramientas de automatización de pruebas	5
Herramientas de integración continua	6
Herramientas de seguimiento de errores/defectos	7
Tablas comparativas para cada categoría de herramientas	8
Herramientas de control de versiones	8
Herramientas de compilación	8
Herramientas de automatización de pruebas	9
Herramientas de integración continua	9
Herramientas de seguimiento de errores/defectos	9
Herramientas que usaremos en nuestro proyecto	10

Herramientas existentes

Herramientas de control de versiones

Nombre de la herramienta	Git
URL	https://git-scm.com/
Coste	Gratuita
Características destacadas	control de versiones distribuido, es rápido, eficiente, admite ramificación y unificación de un proyecto.
Limitaciones identificadas	Dificultad para comprimir y almacenar archivos binarios.

Nombre de la herramienta	Subversion
URL	http://subversion.apache.org/
Coste	Gratuita
Características destacadas	bloqueo de archivos, archivos binarios manejados eficientemente
Limitaciones identificadas	Retroceso defectuoso, difícil arreglo de conflictos en sus ramas.

Nombre de la herramienta	Bazaar
URL	https://bazaar.canonical.com/en/
Coste	Gratuita
Características destacadas	Sencillez, seguimiento de carpetas vacías, renombre de ficheros, se integra con git y subversion.
Limitaciones identificadas	Desarrollo lento, requisitos complicados

Herramientas de compilación

Nombre de la herramienta	Maven
URL	http://maven.apache.org/
Coste	Gratuito
Características destacadas	Podemos compilar, empaquetar, generar documentación, pasar los test, preparar las builds, gestionar dependencias.
Limitaciones identificadas	La definición de los proyectos se hace en un documento XML, lo cual puede dificultar la escritura de estos, o que la lectura sea demasiado verbosa.

Nombre de la herramienta	CMake
URL	https://cmake.org/
Coste	Gratuito
Características destacadas	Flexible, multiplataforma,
Limitaciones identificadas	Generador de generadores: Se adapta muy bien a otros generadores, pero no funciona para la compilación por sí solo.

Nombre de la herramienta	Gradle
URL	https://gradle.org/
Coste	Gratuito
Características destacadas	Cualquier lenguaje de programación. Integración con Android Studio, multi repositorio, alto rendimiento.
Limitaciones identificadas	Gradle se presta muy lento para ciertas operaciones, la sintaxis está basada en Groovy (o Kotlin), un lenguaje de programación, lo cual complica su aprendizaje.

Herramientas de automatización de pruebas

Nombre de la herramienta	Postman
URL	https://www.getpostman.com/
Coste	versión gratuita, versión 8\$-21\$ mes/usuario
Características destacadas	Interfaz gráfica simple
Limitaciones identificadas	No contiene monitoreo de casos de prueba, no admite pruebas de servicio de forma nativa, depende de herramientas externas.

Nombre de la herramienta	JMeter
URL	https://jmeter.apache.org/
Coste	Gratis
Características destacadas	Prueba de carga para conexiones(POP3,HTTP...), automatiza usuarios virtuales, resultados en diagramas, tablas, etc. Independiente de la plataforma
Limitaciones identificadas	Demasiados ajustes de configuración. Interfaz difusa, ayuda y manuales poco entendibles.

Nombre de la herramienta	Cucumber
URL	https://cucumber.io/
Coste	Versión gratuita, versión comercial 45\$ usuario/mes
Características destacadas	Retroalimentación rápida y pruebas de regresión, inyección de características, simulacros de soporte para módulos de código.
Limitaciones identificadas	Es relativamente moderno y no tiene la misma penetración en el mercado que otras herramientas similares.

Herramientas de integración continua

Nombre de la herramienta	Jenkins
URL	https://jenkins.io/
Coste	Gratis
Características destacadas	Multiplataforma, arquitectura extensible y basada en plugins, control total del sistema, personalización.
Limitaciones identificadas	Servidor dedicado, lo que requiere un gasto adicional, configuración/personalización complicada.

Nombre de la herramienta	CircleCI
URL	https://circleci.com/
Coste	Gratis para proyectos open source, para Linux 50\$/mes, para MacOS 39\$/mes
Características destacadas	Fácil y rápida, despliegue de aplicaciones web y móviles. No necesita servidor dedicado.
Limitaciones identificadas	Solo admite plataformas Linux o MacOS, o una plataforma de cloud. Solo admite 2 Ubuntu 12 y 14 de forma gratuita. Lenguajes de programación: Go (Golang), Haskell, Java, PHP, Python, Ruby / Rails, Scala

Nombre de la herramienta	Travis CI
URL	https://travis-ci.org/
Coste	Gratis para proyectos open source, plan 69\$/mes
Características destacadas	Integración continua distribuido, No necesita servidor dedicado. Comienzo rápido
Limitaciones identificadas	El proyecto tiene que estar alojado en GitHub, para su personalización necesita software de terceros.

Herramientas de seguimiento de errores/defectos

Nombre de la herramienta	JIRA
URL	https://www.atlassian.com/es/software/jira
Coste	10,00 US\$/mes
Características destacadas	Gestión de flujo de trabajo, tareas, trabajos pendientes, problemas, además de gestión de tickets y versiones.
Limitaciones identificadas	Formación sólo en seminarios web, si usamos Linux podremos usar su plataforma sólo en la nube.

Nombre de la herramienta	Asana
URL	https://asana.com/es
Coste	9,99 US\$/mes, versión gratuita
Características destacadas	Gestión de flujo de trabajo, tareas, trabajos pendientes, problemas. Formación en persona y web.
Limitaciones identificadas	No tiene gestión de tickets y versiones, si usamos Linux podremos usar su plataforma sólo en la nube.

Nombre de la herramienta	Trello
URL	https://trello.com/es
Coste	9,99 US\$/mes, versión gratuita
Características destacadas	Diagramas de Gantt, gestión de contactos, gestión de documentos, gestión de notas, de flujo de trabajo, de tareas, modelos personalizables, planificación de medios, supervisión KPI.
Limitaciones identificadas	Sólo desde la nube. No tiene seguimiento de problemas, formación solo en línea.

Tablas comparativas para cada categoría de herramientas

Herramientas de control de versiones

Criterio	Peso	Git	Subversion	Bazaar
Ramificación y Unificación de Código	50%	8	5	6
Popularidad	30%	10	6	3
Proveedores de Servicio	20%	10	4	2
Total	100%	9	5.3	4.3

Herramientas de compilación

Criterio	Peso	Maven	Cmake	Gradle
Compatibilidad con diferentes entornos	40%	5	9	7
Gestión de Dependencias	30%	10	6	10
Facilidad de Configuración	30%	8	6	5
Total	100%	7.4	7.2	7.3

Herramientas de automatización de pruebas

Criterio	Peso	Postman	JMeter	Cucumber
Precio	20%	6	10	4
Interfaz Gráfica	50%	8	2	8
Dependencia de otros servicios	30%	2	8	6
Total	100%	5.8	5.4	6.6

Herramientas de integración continua

Criterio	Peso	Jenkins	CircleCI	Travis CI
Servidor dedicado	40%	2	8	8
Dependencia de plataforma	40%	8	6	2
Precio	20%	9	7	8
Total	100%	5.8	7	5.6

Herramientas de seguimiento de errores/defectos

Criterio	Peso	JIRA	Asana	Trello
Precio	40%	8	9	9
Alcance de gestión	30%	9	6	10
Formación	30%	6	10	6
Total	100%	6.2	8.4	8.4

Herramientas que usaremos en nuestro proyecto

Hemos elegido como herramienta de control de versiones Git porque cuenta con una gran comunidad donde apoyarse, tiene ramificación, cosa indispensable para nuestro proyecto ya que cada trabajador puede avanzar en una funcionalidad diferente. Además tiene varios proveedores de servicios pudiendo migrar nuestro proyecto con rapidez.

Como herramienta de ayuda a la compilación, hemos elegido Cmake porque nos ofrece más versatilidad a la hora de gestionar un proyecto software, y como se puede ver en la tabla comparativa, no hay mucha diferenciación con el resto de herramientas respecto a nuestras necesidades.

Para la herramienta de automatización de pruebas, elegimos Cucumber que, aunque es relativamente moderno, presenta un flujo de trabajo muy sencillo, que nos permite reunir todos los requisitos con un lenguaje natural y automatizable.

Para la integración continua hemos decidido usar CircleCI, es la que mejor se ajusta a nuestro proyecto, lo podemos usar en Git, no tiene servidor dedicado lo que supone un gran ahorro. Además la plataforma no va a ser un problema ya que hemos decidido usar Linux.

Como herramienta de seguimiento de errores vamos a usar Jira ya que, aunque es un poco más cara, tiene el alcance de gestión que necesitamos en nuestro proyecto y además tiene un plugin para Cucumber que nos va a facilitar bastante el trabajo.