

INVESTIGACIÓN GITHUB, GRADLE Y MAVEN

POR

ASTRID CAROLINA DÍAZ GÓMEZ

LUIS CARLOS MARIN CAMPOS

ANDRES DARIO HIGUITA PEREZ

LUIS MATEO HINCAPIÉ MARTÍNEZ

ARQUITECTURA DE SOFTWARE

INSTRUCTOR

ROBINSON CORONADO GARCIA

INGENIERÍA DE SISTEMAS



**UNIVERSIDAD
DE ANTIOQUIA**

1 8 0 3

UNIVERSIDAD DE ANTIOQUIA

2020-1

Introducción

Al iniciar en el mundo de la arquitectura de software, tenemos que tener en cuenta varias herramientas que nos puedan generar una facilidad a la hora de realizar diversos trabajos colaborativos con los cuales se pueda llevar a cabo la realización de código, es así como en el siguiente documento nos encargaremos de dar un vistazo y algo de profundización sobre algunas herramientas que nos podrán facilitar la realización de código.

Objetivo General

Nuestro objetivo general es conocer las tecnologías necesarias para construir cualquier proyecto de desarrollo.

Objetivos Específicos

Como objetivos específicos nos dispondremos a investigar sobre las diferentes herramientas que encontramos en el medio, adentrándonos en sus características y funcionamiento.

Nos centraremos en 3 herramientas:

- GITHUB
- GRADLE
- MAVEN

Procedimiento

GITHUB

¿Qué es GitHub?

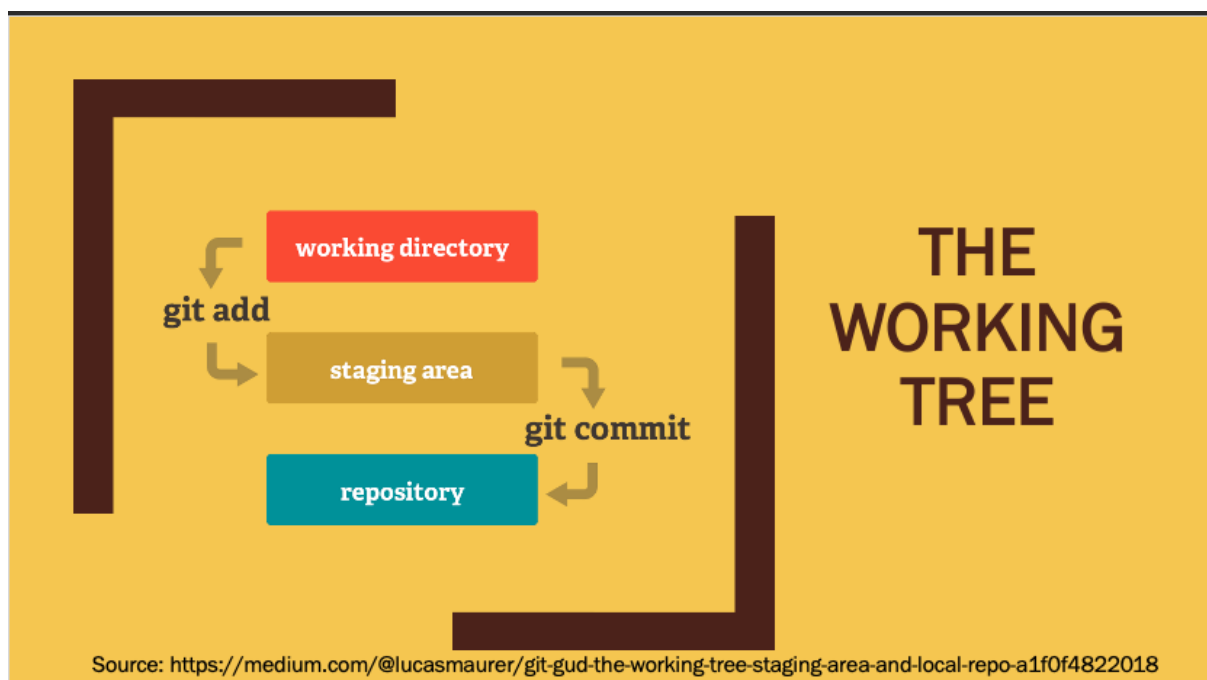
GitHub es una plataforma de alojamiento de código para el control de versiones y la colaboración (Git). Le permite a usted y a otros trabajar juntos en proyectos desde cualquier lugar.

¿Qué es Git?

Git es un sistema de control de versiones distribuido, gratuito y de código abierto. Diseñado para manejar todo, desde proyectos pequeños hasta muy grandes, con rapidez y eficiencia.

Comandos básicos

- **git init:** crea un repositorio de Git vacío o reinicia uno existente
- **Git Add:** agrega el contenido del archivo al índice
- **Estado de Git:** muestra el estado del árbol de trabajo
- **Git Commit:** registra los cambios en el repositorio
- **Git push:** actualice las referencias remotas junto con los objetos asociados
- **Git Pull:** obtenga e integre con otro repositorio o una sucursal local
- **Git Merge:** une dos o más historias de desarrollo juntas



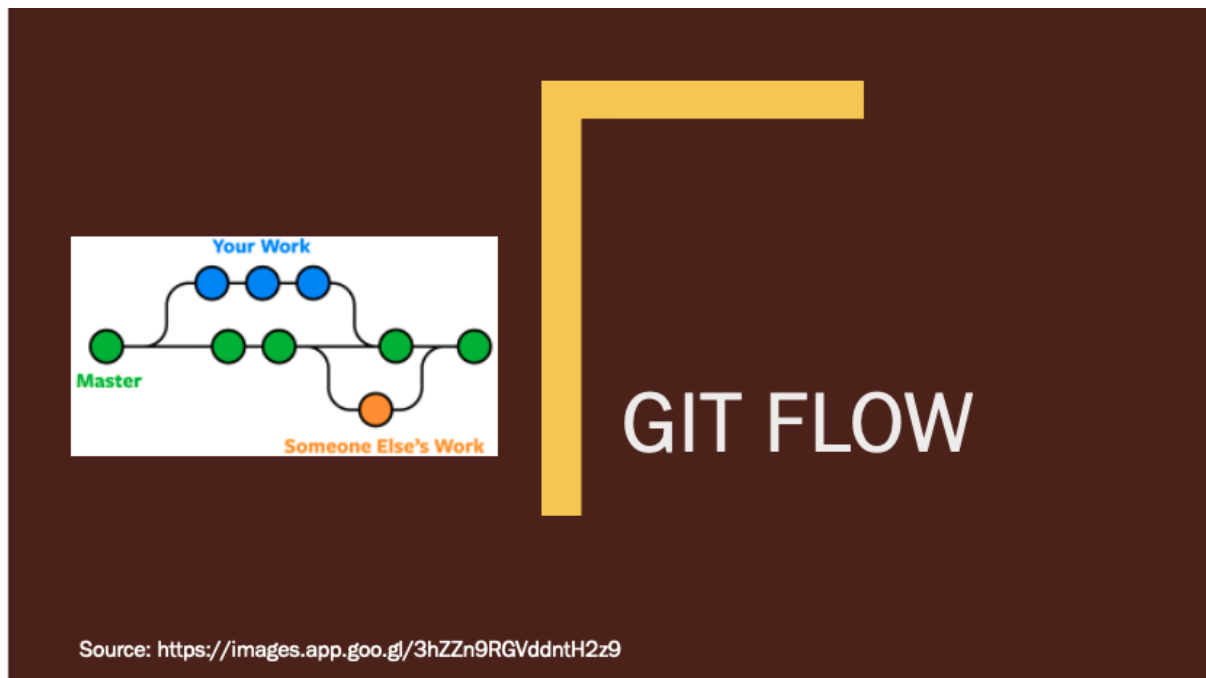
¿Cómo puede ver lo que hay en su árbol de trabajo?

Ejecute el comando `git status`.

¿Cómo agrega archivos a su área de ensayo? Ejecutar el comando `git add # filename #` agregará un archivo específico al Área de ensayo desde su Árbol de trabajo. Si desea agregar todo desde el Árbol de trabajo, ejecute el comando `git add`.

¿Cómo puedo cargar el compromiso en el repositorio remoto?

Ejecute el comando `git push origin master`



Que es Git flow

Es un modelo de ramificación para Git, creado por Vincent Driessen. Ha atraído mucha atención porque se adapta muy bien a la colaboración y al escalado del equipo de desarrollo.

Beneficios clave:

- Desarrollo paralelo
- Colaboración
- Liberar el área de preparación
- Soporte para arreglos de emergencia

GRADLE



Esta es otra de las herramientas que nos permite la automatización de compilación de código abierto, la cual se centra en la flexibilidad y rendimiento, debiéndole aportes a herramientas como ant y maven, pero esta intenta ir mas allá. Los scripts de compilación de Gradle se escriben utilizando Groovy o Kotlin DSL (Lenguaje específico de dominio) para trabajar con un lenguaje sencillo y claro a la hora de

construir el build, además dispone de una gran flexibilidad que permite trabajar con ella utilizando otros lenguajes y no solo Java y también cuenta con un sistema de gestión de dependencias muy estable.

Esta herramienta es muy personalizable, rápida y potente; es la herramienta de compilación oficial para Android, con soporte para muchos idiomas y diferentes tecnologías populares.

Algunas de las características de Gradle son:

- Depuración colaborativa: Permite compartir los resultados de la compilación para resolver en equipo de forma eficiente posibles problemas que aparezcan.
- Construcción incremental: Valida en el proceso de compilación si la entrada, salida o implementación de una tarea ha cambiado, en caso de no existir algún cambio la considera actualizada y no se ejecuta.
- Diseño de repositorio personalizado: Podremos tratar prácticamente cualquier estructura de directorios del sistema de archivos.
- Dependencias transitivas: Es uno de los principales beneficios que ofrece al utilizar la gestión de dependencias ya que se encarga de descargar y administrar las dependencias transitivas.
- Soporte a Groovy y Scala incorporado: Compatibilidad con los proyectos de Groovy, permitiendo trabajar con código Groovy o código Scala e inclusive desarrollar código mixto Java y Groovy o Java y Scala.
- Compilación incremental para Java: En caso de que el código fuente o la ruta de clase cambien, Gradle cuenta con la capacidad para detectar todas las clases que se vean afectadas por dicho cambio y procederá a recompilarlas.
- Embalaje y distribución de JAR, WAR y EAR: Cuenta con herramientas para empaquetar el código basado en JVM (Java Virtual Machine) en archivos de archivo comunes.
- Integración con Android Studio: Android Studio no cuenta con un generador interno, sino que delega todas las tareas de compilación en Gradle, garantizando la corrección en todas las construcciones, ya sea que se ejecuten desde Android Studio, la línea de comandos o un servidor de construcción de integración continua.
- Soporte de MS Visual C ++ y Google Test: Gradle acepta la construcción con el compilador de Visual C de Microsoft en Windows. (VS 2010, VS 2013 y VS 2015 compatibles), así como también realizar pruebas de aplicaciones C con Google Test.

- Publicar en repositorios Ivy y Maven: Permite publicar Artifacts en repositorios Ivy con diseños de directorios completamente personalizables
- Test Kit para pruebas funcionales: Permite la ejecución prágmatca de builds inspeccionando los resultados de compilación, ésta es una prueba de compatibilidad entre versiones.
- Distribuciones personalizadas: En Gradle cada distribución cuenta con un directorio init.d en el que se pueden colocar scripts personalizados que pre configuran su entorno de compilación.
- Lee el formato POM: Es compatible con el formato de metadatos POM, por lo que es posible recuperar dependencias de cualquier repositorio compatible con Maven.
- Compara builds: Resalta de forma rápida las diferencias entre compilaciones, lo que hace que el análisis de la causa raíz sea mucho más rápido y eficaz.
- Compilador daemon: Gradle crea un proceso de daemon que se reutiliza dentro de una compilación de múltiples proyectos, cuando necesita bifurcar el proceso de compilación, mejorando la velocidad de compilación.
- Personalizar y extender escaneos: Ofrece la opción de agregar sus propios datos para construir escaneos como etiquetas, valores y enlaces, integrando escaneos de compilación en la cadena de herramientas.
- Caché de dependencia de terceros: Las dependencias de repositorios remotos se descargan y almacenan en caché localmente, las compilaciones posteriores utilizan los Artifacts almacenados en caché para evitar el tráfico de red innecesario.

MAVEN



Apache Maven es una herramienta de gestión y comprensión de proyectos de software. Basado en el concepto de un modelo de objetos de proyecto (POM), Maven puede administrar la construcción, informes y documentación de un proyecto a partir de una pieza central de información.

El concepto de Maven es aglutinar conocimiento y reutilización de código, estandarización y una buena definición de configuraciones de proyectos.

Basado en el concepto de un modelo de objeto de proyecto: las compilaciones, la gestión de dependencias, la creación de documentación, la publicación del sitio y la publicación de la distribución se controlan desde el archivo declarativo pom.xml. Maven puede ampliarse mediante complementos para utilizar una serie de otras herramientas de desarrollo para la elaboración de informes o el proceso de compilación.

Apache Maven es una herramienta que estandariza la configuración de un proyecto en todo su ciclo de vida, como por ejemplo en todas las fases de compilación y empaquetado y la instalación de mecanismos de distribución de librerías, para que puedan ser utilizadas por otros desarrolladores y equipos de desarrollo.

También contempla temas relacionados con la integración continua, para poder realizar la ejecución de test unitarios y pruebas automatizadas, test de integración, etc.

Principales características de Maven

Básicamente, Apache Maven no deja de ser la base de los compiladores actuales, de IDEs como Eclipse, NetBeans o IntelliJ, a los que ofrece soporte gracias a algunas de sus características, como, por ejemplo:

- Un sistema de gestión de dependencias.
- Un mecanismo distribuido de distribución de librerías. El comportamiento distribuido es siempre desde el repositorio local de Maven hacia los repositorios que están publicados en Internet o en la red corporativa.
- Mecanismos para ser extensible, por la creación de plugins personalizables.
- Es multi-plataforma, puede funcionar tanto en entornos Linux como Windows al ser una aplicación Java.
- Es software libre, con lo cual es el código está disponible, se podría modificar y customizar en caso de que fuera necesario.
- Fomenta la reutilización de código y de librerías. El hecho de que Apache Maven ofrezca repositorios oficiales y públicos de software libre, con librerías desplegadas, que toda la comunidad de desarrolladores de software utiliza, hace que este concepto también pueda trasladarse al mundo empresarial, a través de repositorios remotos corporativos, compartidos por distintos equipos de proyectos o el propio equipo de desarrollo.
- Es compatible con múltiples IDEs.

Definición de Maven

Usando una traducción más ajustada, se podría decir que Maven es:

Una herramienta de gestión de proyectos que fomenta, a través de la definición de un Project Object Model (el fichero POM que se define en cada uno de los proyectos o módulos), un conjunto de estándares que definen el ciclo de vida del proyecto.

Un sistema de gestión de dependencias muy potente.

Una lógica de ejecución de plugins que permite, en determinadas las fases del ciclo de vida, poder ejecutar un plugin concreto que ofrezca una solución a un problema. Por ejemplo, antes de la ejecución de los test, en muchos casos interesa levantar un servidor de aplicaciones, ejecutar los tests y una vez hayan finalizado, bajar ese servidor para que no ocupe ningún puerto y para mejorar el rendimiento de la máquina.

Conclusiones

Al día de hoy podemos ver claramente que hay diferentes herramientas en el medio con las cuales podemos hacer más llevadero la realización de proyectos y código abierto.

Tanto Gradle como Maven son las herramientas de automatización que se utilizan para crear nuevos proyectos en varios lenguajes de programación que admiten múltiples entornos IDE. Tanto la herramienta tiene sus propias ventajas como desventajas y seleccionar la herramienta entre dos depende completamente de los requisitos del proyecto.

Gradle parece tener una curva de aprendizaje más pronunciada, especialmente si no está familiarizado con Groovy o Kotlin.

La "flexibilidad" es la razón principal por la que a los desarrolladores les gusta Gradle, mientras que la "gestión de dependencias" es la principal causa por la que los desarrolladores eligen Maven.

Gradle es más poderoso. Sin embargo, hay ocasiones en las que realmente no necesitas la mayoría de las características y funcionalidades que ofrece. Maven puede ser mejor para proyectos pequeños, mientras que Gradle es mejor para proyectos más grandes.

Webgrafía

Git. [Online]. <https://git-scm.com/>.

Introducing GitFlow. [Online]. <https://datasift.github.io/gitflow/IntroducingGitFlow.html>.

C. Á. Caules, “¿Qué es Gradle?,” Arquitectura Java, 03-May-2019. [Online]. <https://www.arquitecturajava.com/que-es-gradle/>.

“Community Home,” Gradle Docs. [Online]. <https://docs.gradle.org/current/userguide/userguide.html>.

“Construcción de Proyectos con Gradle,” SG Buzz. [Online]. <https://sg.com.mx/revista/33/construccion-proyectos-gradle>.

“GitHub,” Wikipedia, 09-Sep-2020. [Online]. <https://en.wikipedia.org/wiki/GitHub>.

“GitHub.com Documentación de Ayuda,” GitHub.com Documentación de Ayuda - GitHub Docs. [Online]. <https://docs.github.com/es/github>.

“Hello World,” Hello World · GitHub Guides. [Online]. <https://guides.github.com/activities/hello-world/>.

Y. Muradas, “Qué es Gradle y sus características,” OpenWebinars.net, 06-Jul-2020. [Online]. <https://openwebinars.net/blog/que-es-gradle/>.

B. Porter, J. van Zyl, and O. Lamy, “Welcome to Apache Maven,” Maven. [Online]. <https://maven.apache.org/>.

E. Pérez, “Github, el inesperado último reducto de libertad de expresión ante el 'Gran Firewall' pone en jaque los intereses de China,” Xataka, 09-Sep-2020. [Online]. <https://www.xataka.com/servicios/github-inesperado-ultimo-reducto-libertad-expresion-gran-firewall-que-pone-jaque-intereses-china>.

C. Yagüe, “Qué es Apache Maven,” OpenWebinars.net, 14-Apr-2020. [Online]. <https://openwebinars.net/blog/que-es-apache-maven/>.

“git Documentation,” Git. [Online]. <https://git-scm.com/docs/git>.

“¿Qué Es GitHub Y Para Qué Se Utiliza?,” Tutoriales Hostinger, 13-May-2019. [Online]. <https://www.hostinger.es/tutoriales/que-es-github/>.