GITHUB, GRADLE Y MAVEN

POR:

HUGO QUINTERO BETANCUR

HUQUIBE

ARQUITECTURA DE SOFTWARE

UNIVERSIDAD DE ANTIOQUIA

FACULTA DE INGENIERIA

2020

CONTENIDO

PRESENTACIÓN

Hugo Quintero Betancur.

Estudiante de Ingeniería de Sistemas de la Universidad de Antioquia, actualmente laboró como Ejecutivo de Ventas de una Empresa de Software que tiene presencia en todo el país, a veces ejerzo además funciones en consultoría y Service Desk.

Apasionado de la Tecnología, la mayoría de las habilidades que he adquirido han sido de manera autodidacta mediante la lectura de libros, la investigación en Internet y la puesta en práctica de lo aprendido; de esta manera incursioné en lenguajes de programación como Basic, Visual Basic 5 y 6, FoxPro y Visual FoxPro, Visual Basic .Net, C#, Java. Mi trabajo además me ha requerido aprender T-SQL, PHP y HTML; en anteriores empleos he realizado labores de Administración de Sistemas, gestión de recursos informáticos y administración de bases de datos.

A nivel personal soy alguien dedicado a la familia, casado por mas de 22 años, en nuestro hogar hay una hija que también es estudiante de la Universidad de Antioquia, en una carrera de la Facultad de Ciencias Humanas.

OBJETIVO GENERAL

Dar una visión sobre las herramientas GitHub, Gradle y Maven, y comprender su importancia en la automatización y simplificación de los procesos relacionados con la gestión de código, el manejo de las versiones y la automatización de la compilación con miras a optimizar y hacer más ágil el proceso de producción de aplicaciones.

OBJETIVOS ESPECIFICOS

- Analizar la utilidad de GitHub, comprender su utilización y funcionamiento.
- Analizar la utilidad de Gradle, comprender su utilización y funcionamiento.
- Analizar la utilidad de Maven, comprender su utilización y funcionamiento.

HERRAMIENTAS DE SOFTWARE

Se utiliza la aplicación Microsoft Word 2013 como editor de Texto, Google Chrome Versión 85.0.4183.102 como navegador Web para las consultas y GitHub Desktop Versión 2.5.5 para Windows como herramienta de sincronización con GitHub.

GITHUB

Según Wikipedia “GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git.”, esta plataforma es propiedad de GitHub Inc, la cual es una filial de Microsoft Corporation. La adquisición de GitHub Inc. Por parte de Microsoft en 2018 hizo que muchos proyectos se retiraran pues existía la posibilidad de que esta compañía, que se dedica al software, accediera al código fuente de los mismos; El código de los proyectos alojados allí se almacena típicamente de forma pública.

Básicamente es una plataforma que permite el trabajo colaborativo entre programadores, facilitando para el equipo de desarrollo tener concentrado el repositorio en un solo lugar. Github fue desarrollado en ***Ruby on Rails***

GitHub está basado en Git, que es un sistema de control de versiones desarrollado originalmente por Linus Torvalds, quien es ampliamente conocido por haber creado Linux. Dado que el código en un proyecto de desarrollo está siendo continuamente modificado, Git permite llevar el control de cada modificación guardando un repositorio y un registro de los cambios lo que permite recuperar código modificado o borrado regresando a versiones anteriores del mismo sin que haya sobre escritura o pérdida.

GitHub toma esa posibilidad que brinda Git y le incorpora el concepto de red social facilitando la interacción y trabajo colaborativo entre programadores convirtiéndola en la mas grande red social para desarrolladores existente, esto permite además que los usuarios interactúen y socialicen con personas afines dentro del ámbito de la programación de aplicaciones.

Está compuesto por repositorios llamados comúnmente “repos”, que son directorios donde se almacenan los archivos del proyecto que pueden ser código, multimedia o en general todo lo relacionado con el proyecto.

Cuando se desea trabajar con una copia del repositorio de manera independiente se crea un ***Branch***, que es una copia independiente que no afecta al repositorio principal o a otras ramas del mismo; cuando se completa el trabajo se puede realizar una ***pull request*** que es una solicitud de aceptación de los cambios realizados en la rama para que sean incorporados al repositorio principal mediante un ***merge.***

Otro proceso que se puede realizar es la bifurcación de un repositorio que consiste en sacar una copia del repositorio actual para que sirva de base a un nuevo proyecto independiente en una acción denominada **fork**.

Entre las características que ofrece GitHub están:

- Wiki para cada proyecto
- Página web para cada proyecto
- Gráfico que permite ver la forma en que los colaboradores trabajan en el proyecto asi como las bifurcaciones del mismo.
- Funcionalidades de red social, como seguidores.
- Herramienta de trabajo colaborativo
- Gestión de proyectos

GRADLE

Gradle es una herramienta de automatización para la construcción de código gestionando las dependencias y generando el código objeto de nuestra aplicación bien sea de java o de otros lenguajes. Los script se crean utilizando un Lenguaje Específico de Dominio o ***DSL*** (***Domain Specific Language***) como ***Groovy*** o ***Kotlin***.

A través de *plugins*, esta herramienta se incorpora en los Entornos Integrados de Desarrollo **IDEs** (***Integrated Development Environment)*** tales como Eclipe o Visual Studio facilitando la automatización de los procesos de compilación y la gestión de dependencias.

Al ejecutar el generador *Gradle Build* este se encarga de compilar, copiar los archivos de recursos, compilar las pruebas unitarias si están definidas y finalmente hace la generación de los artefactos o entregables definidos para el proyecto.

Entre las características a destacar de Gradle están:

- **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 como un repositorio de *Artifacts*.

- **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 GoogleTest:** 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 GoogleTest.
- **Publicar en repositorios Ivy y Maven**: Permite publicar Artifacts en repositorios Ivy con diseños de directorios completamente personalizables. De igual modo, sucede con Maven en Bintray o Maven Central.
- **TestKit para pruebas funcionales**: Permite la ejecución prágramática 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

Maven es una herramienta *Open-Source* creada en 2001 con el objetivo de facilitar y simplificar los procesos de compilación y generación de ejecutables a partir del código fuente.

Además de permitir hacer este proceso de manera más fácil, Maven es capaz de gestionar un proyecto de software, desde la verificación de la corrección del código pasando por la ejecución de pruebas, generación de informes y documentación hasta el despliegue de la aplicación.

El ciclo por defecto de Maven incluye las siguientes etapas:

- **Validación** (*validate*): Validar que el proyecto es correcto.
- **Compilación** (*compile*).
- **Test** (*test*): Probar el código fuente usando un *framework* de pruebas unitarias.
- **Empaquetar** (*package*): Empaquetar el código compilado y transformarlo en algún formato tipo .jar o .war.
- **Pruebas de integración** (*integration-test*): Procesar y desplegar el código en algún entorno donde se puedan ejecutar las pruebas de integración.
- **Verificar** que el código empaquetado es válido y cumple los criterios de calidad (*verify*).
- **Instalar** el código empaquetado en el repositorio local de Maven, para usarlo como dependencia de otros proyectos (*install*).
- **Desplegar** el código a un entorno (*deploy*).

Para llevar a cabo estas tareas se ejecuta la instrucción mvm y el nombre de la fase correspondiente al nombre puesto entre paréntesis en el listado anterior; Maven se asegura de ejecutarlas en cadena de tal manera que si ejecutamos la instrucción mvn *package* ejecutará primero las opciones de *validate*, *compile*, *test* y finalmente *package*.

Maven provee una semántica común al proceso de generación y desarrollo de software.

Puede integrarse a IDEs como Eclipse o Visual Studio.

CONCLUSIONES

La creación de aplicaciones es una tarea que involucra múltiples procesos que van mas allá de digitar interminables líneas de código en un editor de texto; para garantizar el éxito es necesaria la utilización de herramientas que permitan actividades tan variadas como la colaboración entre desarrolladores, el control de versiones, la verificación del código, la compilación, comprobación de dependencias, entre otros, que den como resultados los artefactos definidos para el proyecto.

Herramientas como GitHub para el trabajo colaborativo y el control de versiones o de Maven o Gradle para la automatización del proceso de verificación, control de dependencias y generación de paquetes entre otros optimizan la labor del desarrollador a la vez que la hacen más productiva y organizada simplificando el ciclo de desarrollo.

Webgrafía

@gustavohostinger, G. B. (13 de Mayo de 2019). *Hostinger*. Obtenido de https://www.hostinger.co/tutoriales/que-es-github/

Caules, C. Á. (1 de Abril de 2015). *Arquitectura Java*. Obtenido de https://www.arquitecturajava.com/que-es-gradle/

Garzas, J. (6 de Junio de 2014). *Jgarzas*. Obtenido de https://www.javiergarzas.com/2014/06/maven-en-10-min.html

González, A. G. (15 de Junio de 2015). *Panamahitek*. Obtenido de http://panamahitek.com/que-es-maven-y-para-que-se-utiliza/

Miró, A. (17 de Agosto de 2017). *Deusto Formación*. Obtenido de https://www.deustoformacion.com/blog/programacion-diseno-web/que-es-para-que-sirve-github

Muradas, Y. (25 de Febrero de 2020). *Open Webinars*. Obtenido de https://openwebinars.net/blog/que-es-gradle/

Wikipedia. (21 de Agosto de 2020). Obtenido de https://es.wikipedia.org/wiki/GitHub

Zamudio, E. (12 de Septiembre de 2020). *SG*. Obtenido de https://sg.com.mx/revista/33/construccion-proyectos-gradle