NOTAS CLASE

frameworks como spring nacen justamente porque no se podían realizar test. Entonces cualquier cambio que yo agrego al proyecto, algún test debería romper (si es que estos son robustos).

tests: existen precondiciones, acciones, postcondiciones (en ese orden).

Junit

SOFTWARE PROJECT MANAGEMENT

Problemas

- los proyectos grandes por lo general contienen gran cantidad de sub-proyectos | módulos | librerías (dependencias)

- se vuelve incomprensible y desprolijo si no se sigue un principio común

- es muy complejo y tedioso construir todos los proyectos manualmente.

Solución Preferida

- Utilizar una herramienta de administración e proyectos (Maven)

- Maven proporciona ayuda en varios aspectos

-- Procesos de build

-- Estructura de los proyectos

-- Administración e dependencias: Solo hay que decirle lo que queremos y se importa solo.

-- Acceso a información y documentación.

APACHE MAVEN 2

Build process

- Proyect Object Model (POM) - archivo XML: Defino toda la metadata de mi proyecto y madle usa este POM para poder construir y desplegar mi proyecto.

- contiene información del proyecto y detalles de configuración

-- Utilizado para construir y desplegar el proyecto

-- Administrar dependencias el proyecto

-- ES posible ejecutar comandos (goals)

-- posee una arquitectura extensible e plugins

-- Administra servicios de metadatos.

-> Maven nos permite realizar herencia de proyectos. ES SOLO PARA JAVA

Ciclo de vida construcción y fases

- el ciclo de vida de construcción (build lifecycle) es el proceso de construcción e implementación de artefactos

Estructura e directorios Estándar

- Ventajas

-- Rápida adaptación para los desarrolladores familiarizados con Mave2

-- no se pierde tiempo reinventando una estructura de directorios (son las que están y chau.)

Administración de dependencias

- Dependencias: librerías de software de terceras partes o propias (JAR o WAR)

- La administración e dependencias es tediosa en las grandes proyectos

-- Dependencia de dependencias (arbol o grafo de dependencias)

-- Errores en tiempo de ejecución por falta de librerías

# TODO ESTO SE ENCUENTRA EN MAVEN CENTRER - que es open source. Define una lista de si que necesito una dependencia vaya a buscar a un servidor, si no lo encuentra va a la siguiente y así.

# un proxy, si quiero acceder a una pagina del exterior, primero le pregunto al proxy y el lo obtiene por mi. Si no lo puede obtener entonces ahí si voy a buscarlo yo. Me ahorro ancho de banda y demás. Maven nos da esto.

- El enfoque manual

-- Descargar las librerías y replicarlas en los proyectos

--- Ineficiente cuando l tamaño de librerías aumenta

--- Difícil de llevar versiones de librerías

--- El checkout de proyecto es lento por necesitar descarga las librerías.

-> hash toma algo de contenido variable y toma algo de contenido fijo, si yo cambio algo en la entrada la salida es completamente distinta. Puedo agarrar un archivo entero y aplicarle una function de HASH y me va a generar un código o un hash de 256 caracteres. Son determinísticas. Siempre es de tamaño fijo y cortito. Me sirve para verificar que el contenido es auténtico, puedo publicar mi librería y el hash de la librería.

-> Maven define repositorios, define una estructura jerárquica

Repositorios

- Repositorios remotos

-- proveen los artefactos para las descargas

--- http://repol.mave.apache.org

--- Repositorio central de maven

- Repositorio Local

- Copia local de artefactos descargados

- USER\_HOME -- acá iba algo más

dependencias transitivas

- maven lee los POM de las dependencias las agrega automáticamente como librerías requeridas

- No hay limite en el número de niveles de ... Nuestro proyecto depende de COMMON\_LOGGINS. Cuando este se publica se publica el hash, el pom, la librería, fuentes y java docs. Cada librería que publica también se publica con sus POM.

Crear un proyecto

$ mvn archetype:generate \

-DgrpupId=ar.edu.utn.frre.cs \

-DartifactId=app1

me genera un proyecto vacío con este arquetipo.

| NOTA: las dependencias tienen la mismas coordenadas de mi proyecto. También posee scope por defecto es la compilación, pero le puedo decir a maven que no las necesito ahí, sino que las necesito en otra instancia, por ejemplo en las pruebas, esto me sirve para eliminar instancias que no necesito en esa parte. |

| NOTA: sdk es una herramienta que me permite cambiar |

++ no se que es esto ++

test -> ??

runtime -> lo coloca únicamente a la hora de armar el empaquetado final.

provide -> lo pone en la compilación pero no en el empaquetado final.

| NOTA: Todo debe tener una estructura y eso me proporciona maven. |

| NOTA: diferencia entre maven y gradle. Antes podía hacer lo que quería, desarrollador 1 hacia de una forma y desarrollador 2 lo hacía de otra forma. Maven organiza todo y coloca un proceso, pero esto no saca libertad de las cosas que puedo hacer y lo que no. Gradle toma todo lo bueno de maven y además me da la posibilidad de agregar mis propias tareas y demás, Gradle se coloca encima e maven, utiliza todos los repos de maven, mismas dependencias de maven, etc. GRADLE ES PARA TODOS LOS LENGUAJES. |

| INFIERNO DE LAS DDL: cuando existía window 95, el sistema operativo tiene librerías que se llaman DDL, en windows la forma de reutilizar código es por medio de las DDL. Las DDL se registraban en el sistema operativo en el registry, cuando un proceso X quiere ejecutar algo va a las DDL a buscar la ruta y lo levanta. En su momento eso se armo hermoso despelote porque había diferentes versiones y cuando una aplicación quería levantar algo se rompía todo. REGISTRY EDITOR |

GRADLE

- la diferencia es que tiene es la sintaxis. Gradle is an opinionated framework on top of an opinionated toolkit.

ANT - te permitía automatizar tareas

- PROYECTO APACHE - open source proyect desarrolla una herramienta llamada MEAVEN (gestor de dependencia, más utilizado a la actualidad)

- Maven viene a solucionar este problema, tiene un default de cada proyecto si o si. Queremos ejecutar tareas, okay tenemos una parte de test. Tenemos parte de empaquetado, deploy, etc. Tambien tiene una tarea que se llama site, agarra toda la documentación como parte de la documentación del proyecto y se la publica en algún sitio.