

CENTRO UNIVERSITÁRIO DE ANÁPOLIS
ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: Gerência de Configuração de Software
PROFESSORA: Ma. Renata Dutra Braga
ALUNOS: Cairo Mateus, Ketlen Fernandes e Rodrigo Martins

ATIVIDADE COMPLEMENTAR

✓ O que é o Maven?

Maven é uma ferramenta desenvolvida pela Apache, responsável por gerenciar dependências e automatizar seus builds

✓ Benefícios

O Maven fornece maior facilidade para se trabalhar com vários módulos de um mesmo sistema e sua extensibilidade para novas funcionalidades como uso de plugins.

Com o Maven é possível isolar as bibliotecas usadas no projeto em um repositório compartilhado pela equipe. Eliminando a possibilidade de duplicidade de dependências entre módulos do projeto.

A ferramenta incentiva a criação de testes unitários, quando ao realizar o build em um servidor, são executados todos os testes unitários. Caso algum teste falhe, o build é abortado.

✓ Modelo de Objetos de Projetos (POM)

O Modelo de Objetos de Projeto define as informações necessárias para que o Maven possa executar um conjunto de metas e que a construção do software possa ser realizada. Todas as definições são feitas em um arquivo chamado pom.xml (isso na versão 2.0 do Maven, porque antes era pelo arquivo project.xml).

O projeto contém arquivos de configuração, uma lista de desenvolvedores e quais papéis atuam, um sistema de controle de ocorrências, informações sobre a empresa e licenças, a URL onde o projeto reside, as dependências e todo o código que dá vida ao projeto.

✓ Repositório

Seus repositórios podem ser tanto Locais quanto Remotos. Quando for local, um diretório será criado em seu computador com o nome de .m2. Suas vantagens são a rápida localização e acesso sempre que o Maven precisar.

Já quando for remoto, o repositório pode estar em um dos vários repositórios Maven espalhados pela internet ou até mesmo na intranet local. O principal repositório remoto atualmente é o site “<http://repo1.maven.org/maven2>”, onde vários projetos de código livre publicam suas bibliotecas nestes repositórios remotos, estando todos disponíveis para download pelo Maven.

✓ O ciclo de vida do processo build

Compilar (Compile): Compila o código fonte.

Teste (Test): Executa teste unitário no código compilado.

Pacote (Package): Empacota o código compilado de acordo com o empacotamento escolhido.

Teste de Integração (Integration-Test): Processa e faz deploy do pacote em um ambiente onde os testes de integração podem ser rodados.

Instalação: Instala o pacote no repositório local, pode ser usado como dependência de outros projetos locais.

Implantação (Deploy): Feito em ambiente de integração ou release

✓ Principais Conceitos Básicos

Project Object Model (POM): Cada projeto Maven fornece um arquivo pom.xml que captura dependências, propriedades de estrutura de projeto, tarefas de fase de construção e comportamento.

Fases de construção e o ciclo de vida de construção: Consiste em várias fases. São fornecidos comandos que o Maven executa cada fase na sequência até a da fase definitiva. Após o pm.xml ser definido, as ferramentas Maven priorizam as fases de construção específicas e reagem às fases de validação, geração de códigos, conjunto de recursos e compilação.

Objetivo: Um objetivo representa uma tarefa específica que é melhor do que uma fase de construção e que contribui para a criação e o gerenciamento de um projeto.

Pacote: A configuração do empacotamento designa um conjunto de objetivos padrão. Exemplos de valores de empacotamentos válidos incluem jar, war, ear e pom.

Plug-in Maven: Um plug-in descreve o conjunto de objetivos que estão ligados a um esquema ou processo de empacotamento específico.

Mojo: Uma tarefa específica que é implementada dentro de um plug-in. Por exemplo, uma implementação de classe Java para implementar em seu ambiente de tempo de execução preferencial.

Archetype: Archetypes são utilizados como modelos de projeto para configurar novos projetos. Esses modelos tornam mais fácil ativar padrões dentro de sua organização, definindo objetivos de empacotamento, configurações de plug-in e dependências predefinidas para as bibliotecas padrão.

Repositórios Maven: Repositórios são utilizados para armazenar artefatos de construção e dependências de tipos variados. Por exemplo, archetypes, plug-ins, e arquivos JAR, entre outros. Repositórios locais são preenchidos vagarosamente, conforme necessário, a partir de repositórios remotos para propósitos de construção.