

UM COMPARATIVO ENTRE OS GERENCIADORES DE DEPENDÊNCIAS EM JAVA

RENAN SILVA FREITAS^[1]; WALTENO MARTINS PARREIRA JUNIOR^[2]

[1, 2] INSTITUTO FEDERAL DO TRIANGULO MINEIRO - *Campus* Uberlândia Centro

Palavras-chave: *Desenvolvedor; Programação; Bibliotecas; Repositório*

Licença:



Resumo

Neste artigo serão apresentadas três ferramentas de gerenciamento de dependências utilizadas em projetos Java, que são, o Maven, o Ivy e o Gradle. Essas ferramentas são de grande importância para simplificar o controle da execução de tarefas repetitivas do desenvolvedor. A partir dessas ferramentas é possível identificar que para cada projeto existe uma melhor ferramenta, porém em termos de maior utilização o Maven é mais popular em função do alto nível de conhecimento da ferramenta pelos desenvolvedores. O uso de gerenciadores de dependência pelos desenvolvedores e suas vantagens e desvantagens. O objetivo deste trabalho é apresentar um estudo de caso comparativo entre três gerenciadores de dependência: o Apache Maven, Apache Ivy e o Gradle, e através desse estudo mostrar qual dos gerenciadores é o mais utilizado no mercado de trabalho atual. A metodologia aplicada foi o estudo dos gerenciadores de dependências mais utilizados em Java, em que foi selecionado os três mais utilizados, buscando informações detalhadas através da WWW (World Wide Web), permitindo assim a comparação entre Apache Maven, Apache Ivy e Gradle. A automação permitida pelo Maven, quando comparamos com a configuração manual dos projetos, traz um ganho em manutenção e velocidade de desenvolvimento, diminuindo bastante a complexidade do gerenciamento das dependências. Além disso, o Maven permite o compartilhamento de um projeto como dependência de outros projetos, fazendo do Maven uma ferramenta muito vantajosa e uma das mais utilizadas por desenvolvedores em todo o mundo. No Apache Ivy o conjunto de instruções permitidas é bem maior que as instruções oferecidas pelo Maven, porém essas instruções fora do padrão tornam-se extremamente mais complexas de implementar, principalmente quando necessita-se de lógicas extensas ou tarefas fora do padrão do Ivy, por exemplo, o uso de lógica de programação para configurar os pacotes de um projeto. Por fim, o Gradle proporciona fortemente a flexibilidade para configurar as dependências de um projeto, principalmente nas situações em que existem projetos pai e filho. De acordo com os gerenciadores de dependências analisados, podemos afirmar que todas as soluções são importantes para os desenvolvedores, não existindo a melhor ferramenta a ser utilizada em todas as situações e projetos e sim a mais adequada para cada tipo de projeto e desenvolvedores que se está trabalhando. Ao analisar cenários reais do mercado de trabalho atual, é possível observar que o Maven é a ferramenta mais adotada pelos desenvolvedores, sendo a mais popular e a mais conhecida em termos técnicos pela maioria deles. Esta análise comparativa entre os gerenciadores de dependências Java pode contribuir para os pesquisadores que tem o interesse de efetuar estudos mais detalhados e também para os desenvolvedores que tem o interesse de aprender mais detalhes sobre as ferramentas citadas.

Introdução

Nos sistemas corporativos dos dias atuais, com benefícios como baixo acoplamento e divisão de responsabilidades, a criação de sistemas projetados a partir desses conceitos se torna algo muito produtivo e atrativo. O conceito citado é a orientação à objetos, que permite o reuso de componentes caso tenha a necessidade de se desenvolver um novo projeto com características semelhantes a outro já existente.

Na linguagem de programação Java, o reuso de código pode ser feito através da utilização de pacotes JAR, que são arquivos compactados com classes já desenvolvidas e podem ser adicionadas a um projeto. Para projetos de grande porte, configurar manualmente todas as bibliotecas que deveriam ser utilizadas é muito

trabalhoso, portanto, para resolver esses problemas, ferramentas de gerenciamento de dependências foram criadas para gerenciar bibliotecas externas ao projeto (DEVMEDIA, 2017).

O objetivo deste trabalho é apresentar um estudo de caso comparativo entre três gerenciadores de dependência: o Apache Maven, Apache Ivy e o Gradle, e através desse estudo mostrar qual dos gerenciadores é o mais utilizado no mercado de trabalho atual.

Os gerenciadores de dependência servem para que seja possível o desenvolvedor definir as bibliotecas e as versões que devem ser incluídas nos projetos. Além disso, uma das grandes vantagens desse tipo de ferramenta é realizar o download das bibliotecas automaticamente, através do uso de sites que fornecem o repositório de dependências, uma espécie de arquivo público com diversos pacotes.

O Apache Maven é uma ferramenta de gerenciamento de dependências criada com o princípio de automatizar o processo de compilação, similar ao Apache Ant, utilizada principalmente em projetos Java (APACHE, 2017b).

O Maven utiliza um arquivo XML (POM) para descrever o projeto de software que está sendo construído, suas dependências sobre módulos e componentes externos, a ordem de compilação, diretórios e plug-ins necessários. Ele tem objetivos pré-definidos para realizar tarefas bem definidas como compilação de código e empacotamento (APACHE, 2017a).

O Maven busca bibliotecas Java e plug-ins dinamicamente de um ou mais repositórios e os armazena em uma área de cache local. Este cache local de artefatos baixados pode ser atualizado com artefatos criados por projetos locais, além de que, os repositórios públicos também podem ser atualizados (APACHE, 2017b).

O Apache Ivy é um gerenciador de pacotes transitivo, ele é um subprojeto do projeto Apache Ant, com o qual a Ivy trabalha para resolver as dependências de projeto. Um arquivo XML externo define as dependências do projeto e lista os recursos necessários para construí-lo. O Ivy realiza o download dos recursos de um repositório de artefatos, seja ele um repositório privado ou um disponível publicamente na Internet (APACHE, 2014).

Até certo ponto, o Apache Ivy compete com o Apache Maven, que também gerencia dependências, no entanto, o Maven é uma ferramenta de construção completa, enquanto a Ivy se concentra exclusivamente no gerenciamento de dependências transitivas.

O gerenciamento de dependências transitivas é uma característica que permite que você obtenha dependências de suas dependências, de forma transitiva. Para resolver esse problema em geral, o Ivy precisa encontrar metadados sobre seus módulos, geralmente em um arquivo Ivy. Para encontrar os metadados e os artefatos de suas dependências (geralmente JARs), o Ivy pode ser configurada para usar muitos repositórios diferentes (APACHE, 2014).

As dependências do Ivy são gerenciadas em um arquivo ivy.xml. Sua configuração é bastante semelhante ao pom.xml do Maven.

O Gradle é a ferramenta para gerenciamento de dependências que eliminou o uso de XMLs e aplica, dentro do processo de compilação e empacotamento, o uso de uma linguagem de programação, o Gradle disponibiliza uma ferramenta completa de gerenciamento e automatização de builds (GRADLE, 2017).

Seu princípio fundamental é o uso da linguagem Groovy nos scripts de build, permitindo que qualquer tarefa seja programável utilizando recursos de linguagens de programação como a criação de funções, loops e ifs (GRADLE, 2017).

O Gradle disponibiliza uma sintaxe simples para declarar as dependências, semelhante ao modo de declaração do Maven e, além disso, possibilita a invocação de scripts Ant.

Pode-se afirmar que o Gradle é uma ferramenta extremamente poderosa, principalmente quando precisamos criar scripts de build um pouco mais complexos.

Projetos importantes, como o SDK do Android e o Hibernate, já utilizam o Gradle em seus desenvolvimentos, mostrando a confiabilidade e qualidade desta solução.

Materiais e métodos

Sabendo que o Maven tem a capacidade de baixar automaticamente as dependências de repositórios online, pode-se afirmar que ele tem como grande vantagem a praticidade na configuração de suas bibliotecas e, além disso, permite a criação de templates que auxilia na criação de um esqueleto do projeto, através dos archetypes.

Por outro lado, como pontos negativos, é possível citar a dificuldade de eliminação de conflitos de dependências, ou seja, quando temos duas dependências iguais no mesmo projeto, principalmente quando a árvore de dependências é muito extensa. Outra desvantagem é o fato do Maven basear-se em XMLs para a configuração do processo de build, tornando-a limitada a instruções pré-definidas, sem espaço para customização.

A ferramenta do Ivy atende a proposta esperada de eliminar a dificuldade de customizar o gerenciamento de dependências e processos de build, o que não ocorre no Maven, possibilitando a criação de tarefas complexas de construção de projetos de maneira simples.

Da mesma maneira que existe um lado positivo para o Ivy, também existe o lado negativo, principalmente pela necessidade de utilizar duas ferramentas distintas para atingir sua proposta, pois o Ivy sozinho realiza o gerenciamento de dependências e não a construção de projetos, que é um papel do Apache Ant.

Apesar do dinamismo e flexibilidade no processo de build do Apache Ant, a ferramenta Ivy nos deixa presos a instruções de build pré-definidas em sua especificação, o que deixa claro uma limitação à automatização dos builds.

No Gradle, assim como no Ivy ou no Maven, é possível ter o controle do processo de build e das dependências que o projeto necessita. Por outro lado, ele permite a criação dos chamados multi-projetos, permitindo uma melhor organização de código, quebrando um sistema grande em projetos menores, cada um com suas dependências.

Essa abordagem proposta pelo Gradle permite maior desacoplamento entre as partes do projeto e, caso necessário a reutilização de uma das partes do mesmo, torna-se muito simples o uso apenas do que realmente é necessário em relação às dependências.

Resultados e discussões

A automação permitida pelo Maven, quando comparamos com a configuração manual dos projetos, traz um ganho em manutenção e velocidade de desenvolvimento, diminuindo bastante a complexidade do gerenciamento das dependências. Além disso, o Maven permite o compartilhamento de um projeto como dependência de outros projetos, fazendo do Maven uma ferramenta muito vantajosa e uma das mais utilizadas por desenvolvedores em todo o mundo.

No Apache Ivy o conjunto de instruções permitidas é bem maior que as instruções oferecidas pelo Maven, porém essas instruções fora do padrão tornam-se extremamente mais complexas de implementar, principalmente quando necessita-se de lógicas extensas ou tarefas fora do padrão do Ivy, por exemplo, o uso de lógica de programação para configurar os pacotes de um projeto.

Por fim, o Gradle proporciona fortemente a flexibilidade para configurar as dependências de um projeto, principalmente nas situações em que existem projetos pai e filho.

Considerações finais

De acordo com os gerenciadores de dependências analisados, podemos afirmar que todas as soluções são importantes para os desenvolvedores, não existindo a melhor ferramenta a ser utilizada em todas as situações e projetos e sim a mais adequada para cada tipo de projeto e desenvolvedores que se está trabalhando.

Ao analisar cenários reais do mercado de trabalho atual, é possível observar que o Maven é a ferramenta mais adotada pelos desenvolvedores, sendo a mais popular e a mais conhecida em termos técnicos pela maioria deles.

Referências:

APACHE Maven Project. **Release Notes – Maven 3.3.9.** 2017a. Disponível em: <<https://maven.apache.org/docs/3.3.9/release-notes.html>>. Acesso em 26 set. 2017.

APACHE Maven Project. **Introduction to Archetypes.** 2017b. Disponível em: <<http://maven.apache.org/guides/introduction/introduction-to-archetypes.html>>. Acesso em 26 set. 2017.

APACHE Ant Project. **Documentation.** 2014. Disponível em: <<http://ant.apache.org/ivy/history/latest-milestone/index.html>>. Acesso em 27 set. 2017.

PARMIGIANI JÚNIOR, Ednei. **Gerenciador de dependências Apache Ivy.** 2017. Disponível em: <<http://www.edneiparmigiani.com.br/gerenciador-de-dependencias-apache-ivy/>>. Acesso em 02 out. 2017.

DEV MEDIA. **Gerenciamento de dependências no Java.** 2017. Disponível em: <<https://maven.apache.org/docs/3.3.9/release-notes.html>>. Acesso em 05 out. 2017.

GRADLE. **Gradle Build Tool 4.2.1.** 2017. Disponível em: <<https://docs.gradle.org/4.2.1/userguide/introduction.html>>. Acesso em 10 out. 2017.