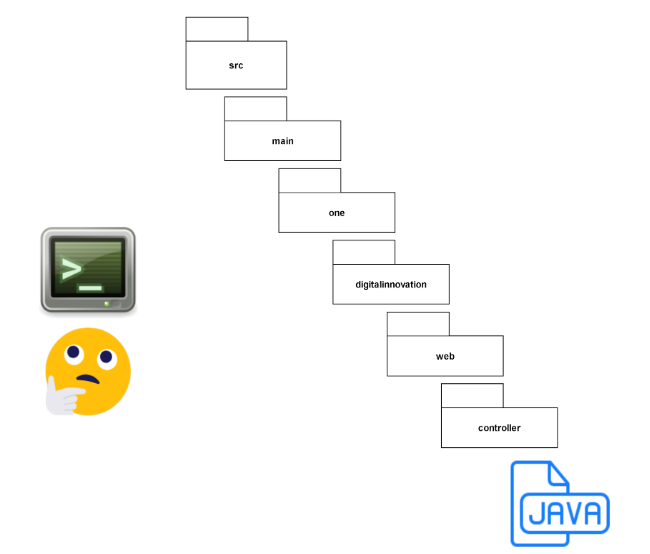
Gerenciamento de Dependências e Build em Java com Maven

* O que é o Apache Maven e sua utilidade?
* Ferramenta para gerenciar build e dependências de um projeto
* Primeira versão em julho de 2004, mantido pela Apache Software Foundation
* O que ela resolve?

A ideia de ter trazido um problema pra explicar o Maven é tentar contextualizar ao máximo um cenário comum para todos os devs e isso ficar de fácil compreensão. Dado que é bastante comum um quick start rápido, muitas pessoas desenvolvedoras terão um nível de abstração alto e isso impedirá que elas percebam certos nuances do desenvolvimento Java. Um deles é o contexto em questão onde a proposta é entender o quão trabalhoso seria construir um projeto de ponta a ponta considerando também execução de testes, empacotamento para deploy e etc.



* O que é Maven ?
* Endereça como o software foi construído e suas dependências através do POM(Project Object Model)
* Facilita a compreensão do desenvolvedor
* Fornecer informações de qualidade

**Comandos que auxiliam o dia a dia**

1. Compilar: compile

2. Testar: test

3. Empacotar: package

4. Limpar diretório de trabalho: clean

**Criando diferentes tipos de projeto**

Como?

Maven archetype

Pesquise na internet por “maven archetype list” para encontrar a opção que faz mais sentido pro seu cenário

Procure pelas instruções de execução do comando mvn archetype

Diagrama

Descrição gerada automaticamente

**POM, Dependências e Repositórios**

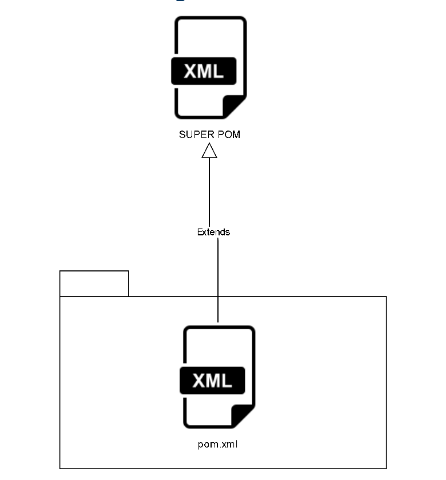
**Entendendo o POM**

* POM – Project Object Model
* Unidade fundamental de trabalho
* Formato XML
* Detalha o projeto
* Detalha como construir o projeto
* Maven sempre procura pelo pom.xml para realizar sua execução

**Detalhes**:

* Nome do projeto
* Dependências
* Módulos
* Configurações de build
* Detalhes do projeto (nome, descrição, licença, url)
* Configurações de ambiente (repositórios, tracking, profiles)
* Exemplo

Super Pom – modelo base no qual tem todas as configurações padrões que o maven utiliza.



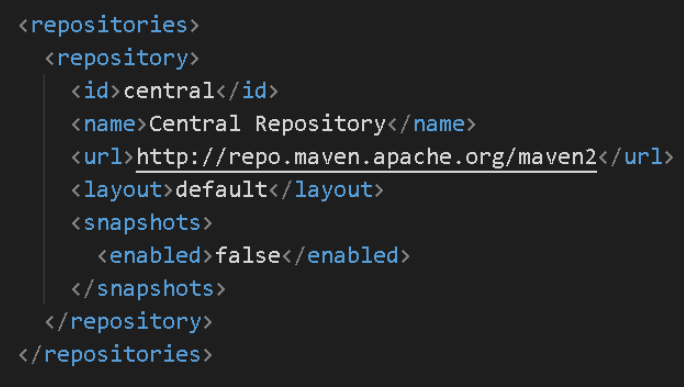
**Repositórios**

**O que são repositórios?**

São locais onde podemos encontrar plugins e bibliotecas que o Maven provê Dois tipos: Local e Remoto

**Repositório remoto**

É o local central utilizado pelo Maven para buscar os artefatos. Configurado automaticamente pelo Super POM para utilizar o Maven Central



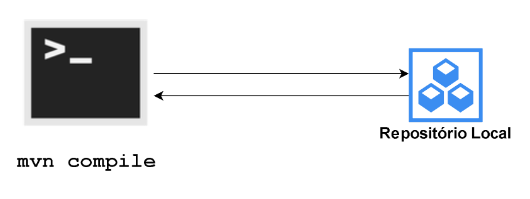
 Via pom.xml do projeto

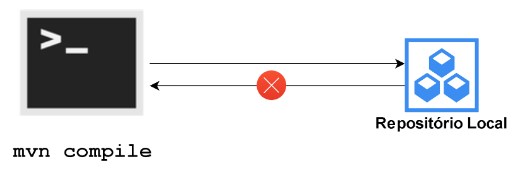
**Configuração**

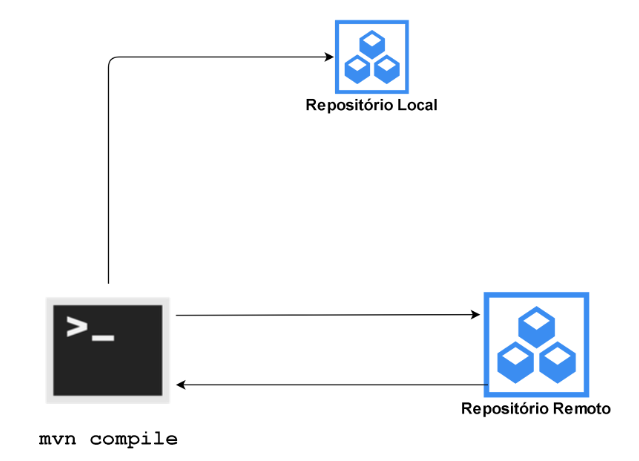
* Via settings.xml
* Localização: pasta\_apache\_maven/conf/settings.xml

**Repositório local**

* É o repositório na máquina utilizado pelo Maven para buscar os artefatos.
* Estratégia de caching
* Localizações
* Windows: %USERPROFILE%\.m2\repository
* Linux: $HOME/.m2/repository







**Como adicionar dependências**



**Propriedades**

* groupId: É como se fosse o id da organização. Segue as regras de nomes de pacote Java
* artifactId: Nome do projeto em si
* Version: Número da versão que será utilizada



**Tipos de dependências**

Direta: dependências declaradas no pom.xml

Transitiva: dependências obrigatórias das dependências declaradas no pom.xml

**Classpath**

* Runtime
* Test
* Compile

**Escopo compile**

* Escopo default
* Disponível em todos os classpaths
* É transitivo

**Escopo provided**

* Indica que a dependência será fornecida em tempo de execução por uma implementação na JDK ou via container
* Exemplos: Servlet API, Java EE APIs
* A dependência com esse escopo é adicionado no classpath usado para compilação(compile) e teste(test) mas não em runtime;
* Não é transitiva



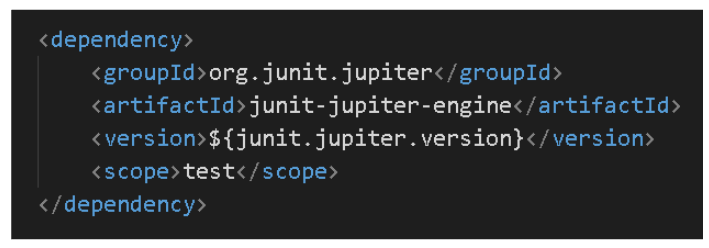
**Escopo runtime**

* Indica que a dependência é necessária para execução e não para compilação
* Maven inclui no classpath dos escopos de runtime e test



**Escopo test**

* Disponível somente para compilação e execução de testes
* Não é transitivo

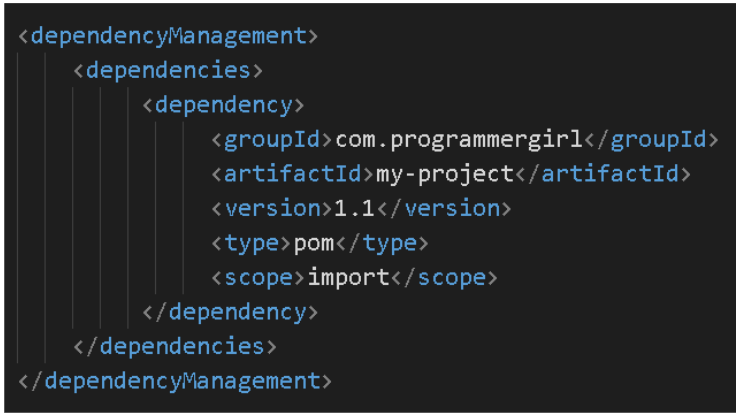


**Escopo system**

* Similar ao escopo provided exceto por ser necessário prover o JAR explicitamente
* A dependência com esse escopo é adicionado no classpath usado para compilação(compile) e teste(test) mas não em runtime;
* Não é transitiva

**Escopo import**

* Este escopo é disponível apenas com uma dependência do tipo pom e com tag <dependencyManagement>
* Indica um processo de reutilizar dependências de um projeto



**Dica sobre escopos, dependências opcionais e exclusões**

**Ver o classpath**

* mvn dependency:build-classpath –DincludeScope=compile
* mvn dependency:build-classpath –DincludeScope=test
* mvn dependency:build-classpath –DincludeScope=runtime

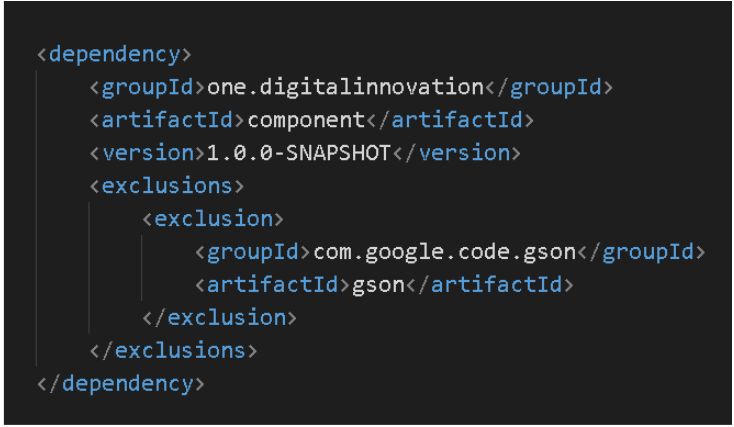
**Dependências Opcionais**

Utilizado quando uma dependência não é necessária para os projetos que irão reutilizar seu componente



**Exclusions**

Utilizado quando o componente que você usa compartilha uma biblioteca que você já tem ou não quer ter disponível

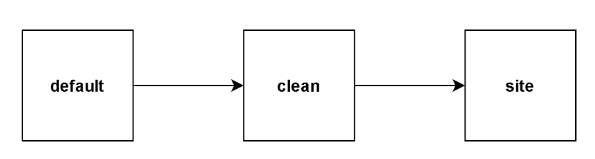


**Maven Build Lifecycle**

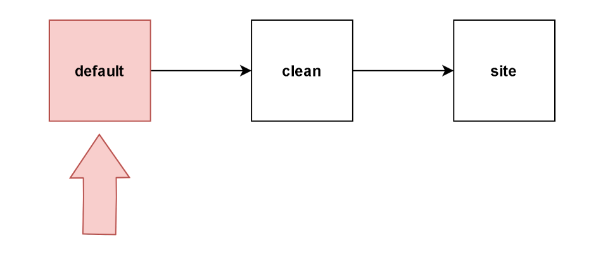
**O que é?**

* Conceito de ciclo de vida de construção
* Conceito e os comandos da ferramenta
* Composto por 3 ciclos de vida
* Cada ciclo possui fases (Maven Phases)
* Cada fase possui objetivos (Maven Goals)

**Como é?**



**Default Lifecycle**

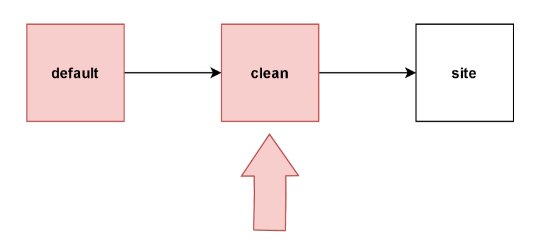
****

* Principal ciclo
* Responsável pelo deploy local
* Composto por 23 fases

**Principais fases**

* validate
* compile
* test-compile
* test
* integration-test
* package
* install
* deploy

**Clean Lifecycle**

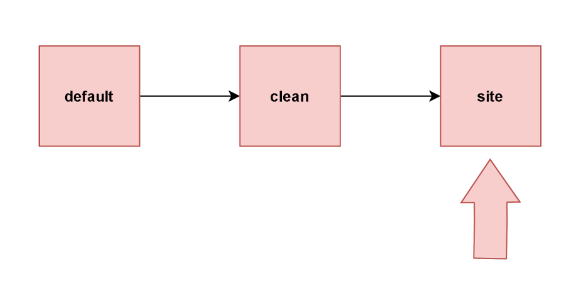


* Ciclo intermediário
* Responsável pela limpeza do projeto
* Composto por 3 fases

**Fases**

* pre-clean
* clean
* post-clean

**Site Lifecycle**



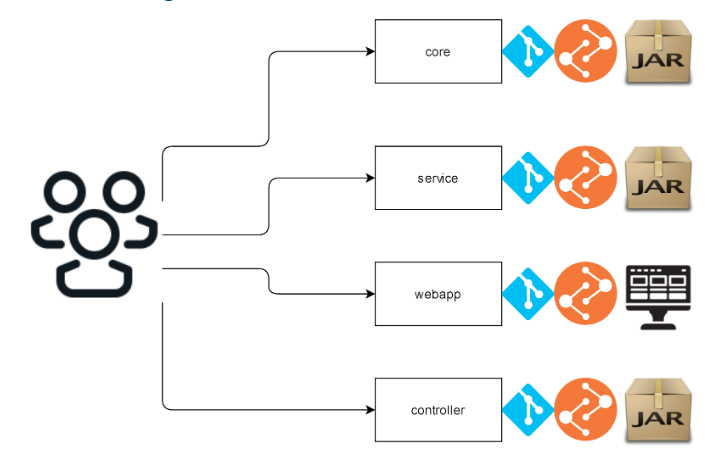
* Ciclo final
* Responsável pela criação do site de documentação do projeto
* Composto por 4 fases

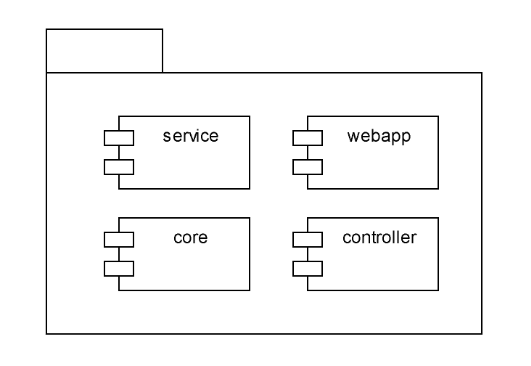
**Fases**

* pre-site
* site
* post-site
* site-deploy

**Multi-módulos**

**Projetos multi-módulos**



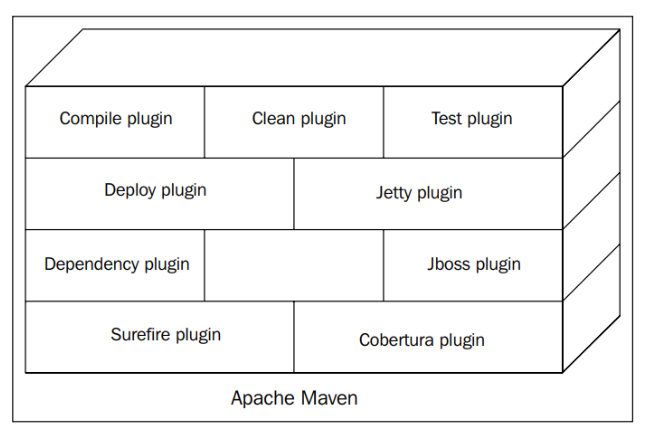


**Plugins**

**O que são os plugins?**

* A maioria das funcionalidades são providas por plugins
* Estilo arquitetural para extensibilidade (criar seu próprio plugin)
* Escrito prioritariamente em Java e disponibilizados comumente como JARs

**Plugins Maven**



**Plugins mais utilizados**

* eclipse
* jacoco
* ear
* war
* compile
* clean
* checkstyle
* javadoc

**Uso**

mvn [plugin-name]:[goal-name]

**Configuração**



**Para aprofundar**

* https://maven.apache.org/guides/
* Apache Maven 3 Cookbook, Srirangan, 2011, Packt Publishing