



PLATAFORMA DE DESENVOLVIMENTO PINHÃO PARANÁ

GUIA DE UTILIZAÇÃO DO MAVEN NA CELEPAR

(Visão Específica Plataforma Java)

Março – 2010

Sumário de Informações do Documento

Tipo do Documento: Guia

Título do Documento: Guia de Utilização do Maven na Celepar

Estado do Documento: Elaborado

Responsáveis: Giuliano Diego Barbarra, Elvis Pelizari - Revisado por: Cleverson Budel

Palavras-Chaves: Maven, Java,

Resumo: Este documento contém um roteiro para utilização da tecnologia Maven na Celepar

Número de páginas: 22

Software utilizados: Open Office 2.0.3

Versão	Data	Mudanças
1.0	19/03/10	Elaboração da guia.
1.1	20/04/11	Pequenos ajustes para adequação a Plataforma Pinhão versão 2.
1.2	24/10/11	Elaboração do tópico guia rápido.

SUMÁRIO

1 MAVEN.....	4
1.1 OBJETIVO.....	4
1.2 VISÃO GERAL.....	4
1.3 ARQUITETURA MAVEN NA CELEPAR.....	5
1.3.1 <i>Estação de Trabalho</i>	5
1.3.2 <i>Archiva</i>	6
1.3.3 <i>CVS / SVN</i>	6
1.3.4 <i>Estaleiro</i>	6
1.4 UTILIZANDO O MAVEN.....	7
1.4.1 <i>M2Eclipse</i>	7
1.4.2 <i>Instalação e Configuração</i>	7
1.4.3 <i>Estrutura do Projeto – Plataforma Pinhão versão 1</i>	8
1.4.4 <i>Estrutura do Projeto – Plataforma Pinhão versão 2</i>	9
1.4.5 <i>Arquetype</i>	10
1.4.6 <i>Dependências</i>	13
1.4.1 <i>Repositórios</i>	14
1.4.2 <i>Ciclo de Vida</i>	15
1.4.3 <i>Deploy Local JBoss</i>	15
1.5 GUIA RÁPIDO PARA CONFIGURAR UM PROJETO.....	16
2 REFERÊNCIAS.....	21

1 MAVEN

1.1 Objetivo

O foco deste documento não é explicar o funcionamento da ferramenta Maven, mas sim, esclarecer como será o seu uso na Celepar, demonstrando a forma indicada para utilização integrada ao ambiente Eclipse, as diferenças de manipulação de um projeto Java com esta ferramenta, além de descrever a arquitetura existente na Celepar para comporta-la.

1.2 Visão Geral

Maven é um projeto da *Apache Software Foundation* para uma ferramenta de gerenciamento e automação de projetos em Java. Ela é similar à ferramenta *Ant*, mas possui um modelo de configuração simplificado, baseado no formato XML.

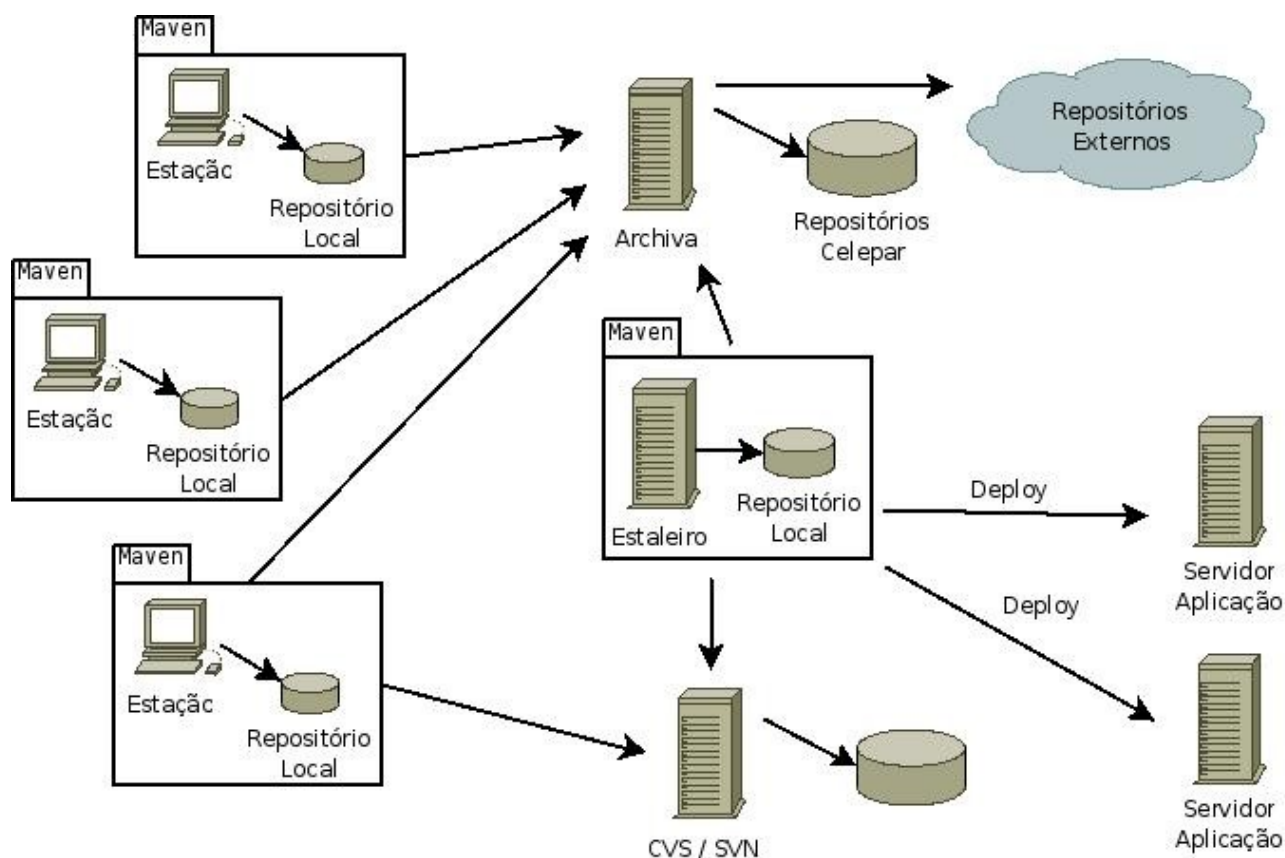
Utiliza uma construção conhecida como *Project Object Model* (POM), o qual descreve todo o processo de construção de um projeto de software, suas dependências, módulos, componentes e a sua sequência de construção. O Maven contém tarefas pré-definidas que realizam funções bem conhecidas como compilação e empacotamento de código.

Uma característica chave do Maven é sua construção para trabalhar em rede. O núcleo da ferramenta pode “baixar” *plugins* de um repositório. (o mesmo repositório utilizado pelos outros projetos Java da *Apache* e outras organizações). O Maven disponibiliza suporte nativo para a recuperação de arquivos deste repositório, e para a inclusão dos artefatos resultantes no final do processo. Um cache de artefatos atua como ponto de sincronização para os projetos locais.

Obs: Informações detalhadas sobre o Maven podem ser encontradas na secção de referências.

1.3 Arquitetura Maven na Celepar

Esta arquitetura muda a forma de como é feito o controle e uso de bibliotecas (.jar), validações de código fonte, compilação, testes, empacotamento, instalação e “limpeza” do ambiente. Abaixo é ilustrada a arquitetura do ambiente Maven utilizado na Celepar:



1.3.1 Estação de Trabalho

A mudança mais significativa está no fato de que nenhuma biblioteca permanecerá armazenada na pasta `.../WEB-INF/lib` dos projetos. Estas bibliotecas estarão localizadas em um repositório local `~/m2/repository`. O projeto fará apenas referência para estas bibliotecas. Caso alguma delas não seja encontrada no repositório local, o Maven se encarregará de buscá-la no repositório corporativo da Celepar (<http://maven.celepar.parana>).

1.3.2 Archiva

O repositório corporativo de bibliotecas é gerido pela aplicação *Archiva*, entre suas responsabilidades estão: controlar o armazenamento das bibliotecas, buscar bibliotecas em repositórios remotos, possibilitar inclusão manual de novas bibliotecas, possibilitar inclusões automáticas (realizadas pelo Maven), possibilitar pesquisa das bibliotecas existentes, controlar acesso as bibliotecas, etc.

1.3.3 CVS / SVN

Para os projetos que utilizarem o Maven, as bibliotecas não serão mais armazenadas no sistema de controle de versão CVS / SVN, sendo responsabilidade deste, apenas o controle do código fonte. Isso se deve ao fato de não mais necessitarmos guardar bibliotecas - jar - dentro do projeto).

Esta alteração reduz drasticamente o consumo de espaço em disco utilizado pelo sistema de controle de versões, visto que geralmente são as bibliotecas que consomem a maior quantidade de espaço dentro de um projeto.

Atenção: quando for realizar a sincronização do projeto, não realizar o *commit* da pasta *target*.

1.3.4 Estaleiro

O sistema Estaleiro (para gestão de *deploys*) também foi alterado visando a utilização da ferramenta Maven. Cabe ao Estaleiro realizar a busca do código fonte no CVS / SVN e das bibliotecas do projeto no sistema *Archiva*. Sendo assim, o próprio Estaleiro fará uso do Maven para realizar todo processo de automação, até o efetivo empacotamento e *deploy* (publicação) no servidor de aplicações indicado.

1.4 Utilizando o Maven

O Maven pode ser utilizado independente de uma IDE de desenvolvimento. Para isto, no ambiente Linux-Debian, basta instalar a ferramenta através do comando:

```
sudo apt-get install maven2
```

Com o aplicativo *mvn* em linha de comando pode-se realizar as operações disponíveis na ferramenta, bastando que o projeto tenha o arquivo *pom.xml* na sua pasta raiz devidamente configurado.

1.4.1 M2Eclipse

Outra forma de se utilizar o Maven é através do *plugin* M2Eclipse, disponível na IDE Eclipse. Ele permite utilizar o Maven de forma integrada ao ambiente de desenvolvimento provido pelo Eclipse. Sendo que o uso do *plugin* é independente da instalação do pacote *maven2* descrito anteriormente.

Obs: Esta é a forma indicada para uso do Maven na Celepar.

1.4.2 Instalação e Configuração

Para a utilização do Maven na Celepar é pré-requisito a instalação da versão atualizada do Eclipse, o qual já possui uma versão estável do *plugin* M2Eclipse. Para isto, remova a instalação atual do seu ambiente de desenvolvimento, atualize a lista de pacotes e instale a nova versão, conforme descrito abaixo:

```
sudo apt-get remove --purge celepar-dev eclipse-celepar  
sudo apt-get update  
sudo apt-get install celepar-dev
```

Agora é necessário incluir um catálogo remoto de *Arquetypes* no *plugin* do Maven, que será descrito e utilizado mais adiante. Para isto, acione o menu do Eclipse: *Window>Preferences>Maven>Archetypes* selecione a opção *Add Remote Catalog* para adicionar um novo catálogo de *archetypes*. Então preencha os campos da seguinte forma:

Catalog File:

<http://maven.celepar.parana/archiva/repository/celepar/br/gov/pr/component/pinhao/archetypes/archetype-catalog/release/archetype-catalog-release.pom>

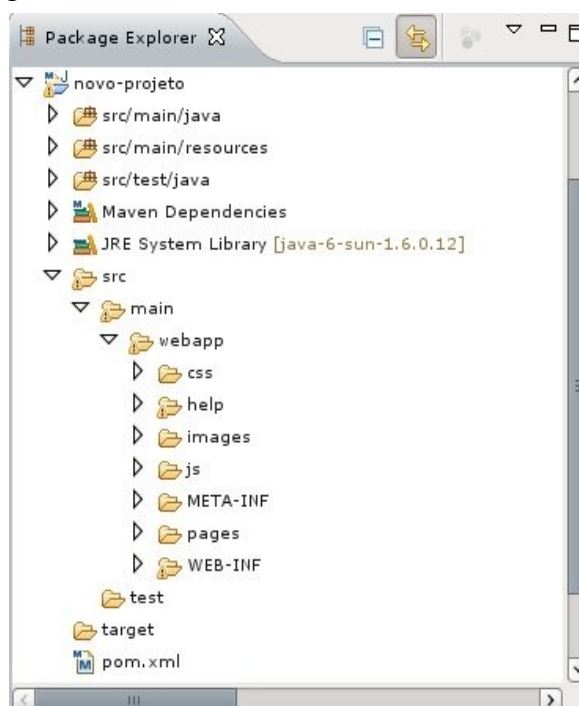
Description: *Catálogo Remoto Celepar*

1.4.3 Estrutura do Projeto – Plataforma Pinhão versão 1

Com a utilização do Maven, a primeira mudança que ocorre é a reestruturação dos diretórios do projeto, agora baseado na estrutura sugerida pelo Maven. Conforme pode ser visto na figura a seguir:

src/main/java	Application/Library sources
src/main/resources	Application/Library resources
src/main/filters	Resource filter files
src/main/assembly	Assembly descriptors
src/main/config	Configuration files
src/main/webapp	Web application sources
src/test/java	Test sources
src/test/resources	Test resources
src/test/filters	Test resource filter files
src/site	Site
LICENSE.txt	Project's license
README.txt	Project's readme

Esta nova estrutura pode ser melhor interpretada através da visão de um projeto no Eclipse, conforme a imagem abaixo:



As pastas *src/main/java* e *src/test/java* correspondem respectivamente ao código fonte da aplicação e ao código fonte dos testes da aplicação.

Temos também a representação de duas fontes de bibliotecas, a *JRE System Library* (biblioteca nativa do JRE) e a *Maven Dependencies* (dependências de bibliotecas definidas no arquivo *pom.xml*).

Na pasta *src/main/resources* estão os arquivos de configuração do sistema, recursos extras que não fazem parte do código compilado ou interpretado do sistema (arquivos de mensagens, mapeamento, configuração de aplicativos, etc).

Já a pasta *src/main/webapp* possui a mesma estrutura do projeto *web* que já vinha sendo utilizado pela Celepar.

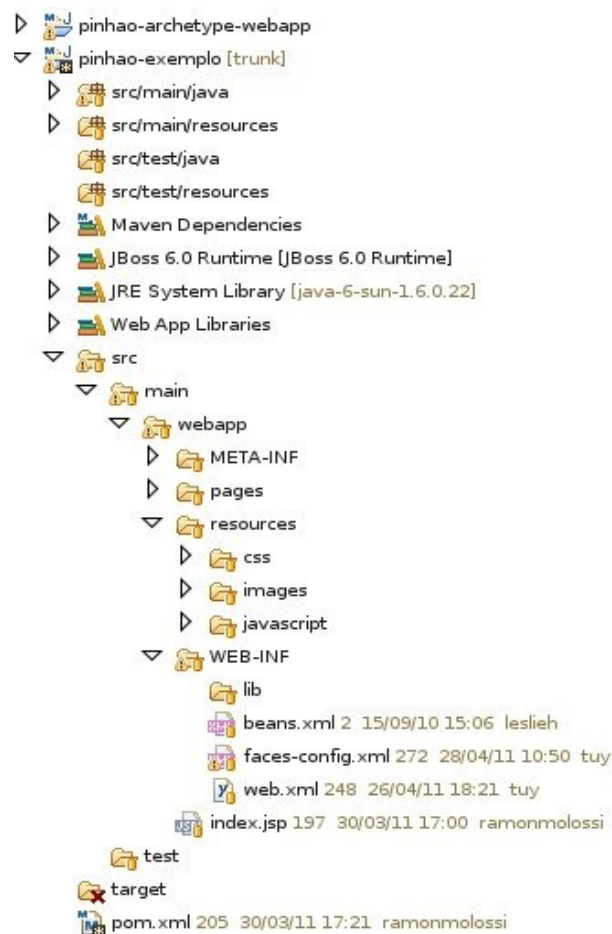
A pasta *target* é utilizada como *output* da projeto, ou seja, todo o conteúdo gerado pelo Maven, será armazenado, alterado ou excluído dentro desta pasta.

Finalmente temos o arquivo *pom.xml* onde toda a configuração do Maven é definida. Informações sobre como alterar estas configurações podem ser obtidos na secção de referências.

Atenção: alterações indevidas no arquivo *pom.xml* podem ocasionar problemas durante o *deploy* via Estaleiro.

1.4.4 Estrutura do Projeto – Plataforma Pinhão versão 2

Na Plataforma Pinhão versão 2 a maneira de se trabalhar com o Maven é a mesma, porém a estrutura de diretórios apresenta pequenas mudanças para agregar algumas facilidades oferecidas pelas novas tecnologias utilizadas. Verificar no documento 'Organização e Nomenclatura de Arquivos' maiores detalhes sobre a estrutura da Plataforma Pinhão 2.

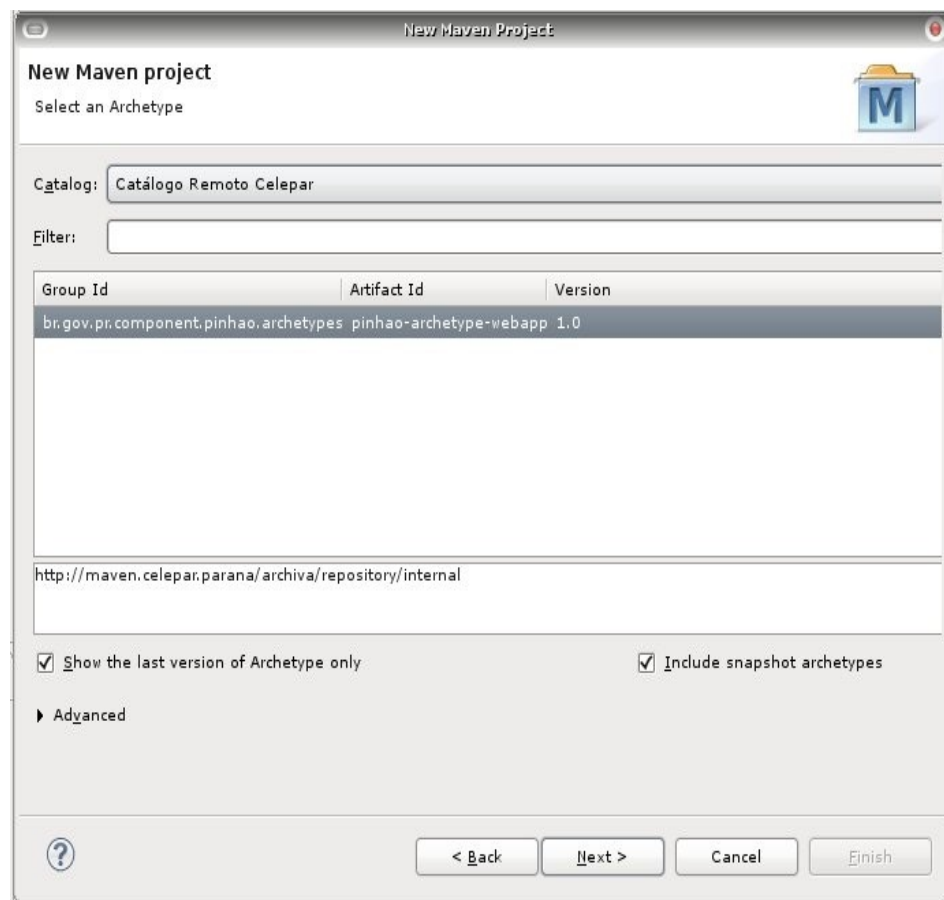


1.4.5 Arquetipo

Com a utilização do Maven, a criação de um novo projeto não depende mais da cópia de um projeto mínimo, no momento da criação de uma nova área no CVS.

Para se criar um novo projeto é utilizado um arquetipo do Maven, que consiste de uma estrutura (previamente definida) capaz de criar um projeto modelo, com uma estrutura de diretórios, arquivos de configuração e artefatos necessários a finalidade específica do projeto pretendido.

Para criar um novo projeto através de um *arquetipo*, abra o *wizard* de criação de novos projetos do Eclipse e selecione *Maven>Maven Project*, apenas avance para a tela seguinte, então será exibido a tela de seleção de *archetypes*, conforme ilustrado a seguir:



Nesta tela selecione o catálogo remoto que foi adicionado anteriormente, então escolha o *arquetype* e a versão desejada do modelo, podendo escolher entre as versões da Plataforma Pinhão versão 1 e também as versões da Plataforma Pinhão versão 2. Nesta tela também existem as opções para mostrar as versões antigas e as *snapshots* (versões recentes que ainda não foram liberadas oficialmente) porém é recomendado mostrar somente as últimas versões.

Na tela seguinte as propriedades básicas de um projeto Maven devem ser preenchidas:

New Maven Project

Enter a group ID for the artifact.

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value
------	-------

Advanced

< Back Next > Cancel Finish

Estas propriedades podem ser descritas resumidamente da seguinte forma:

- *Group Id*: descreve o domínio do projeto
- *Artifact Id*: descreve o nome do projeto
- *Version*: descreve a versão do projeto
- *Package*: descreve o pacote base do projeto

Estas propriedades são a identificação única de um projeto, que mais tarde pode servir como uma biblioteca, podendo assim, ser referenciado em outros projetos. Mais detalhes sobre esta característica podem ser obtidos no material de referência.

Após esta tarefa teremos um projeto pré-configurado, pronto para ser utilizado.

1.4.6 Dependências

O Maven permite uma inclusão fácil e controla de dependência de bibliotecas (*jar*). Estas dependências são definidas no arquivo *pom.xml*, dentro da tag `<dependencies>`. A forma mais simples de adicionar dependência em um projeto é realizar uma pesquisa no repositório corporativo de bibliotecas (gerenciado pelo Archiva - <http://maven.celepar.parana.br>), selecionar a versão desejada e copiar o código XML de identificação da biblioteca, junto a tag `<dependency>`, conforme tela a baixo:



The screenshot shows the Archiva web interface. On the left is the Archiva logo and a sidebar with 'Find' (Search, Find Artifact, Browse) and 'Manage' (Upload Artifact, Delete Artifact) buttons. The main content area shows the search results for 'sentinela_client'. It includes a search bar at the top, a breadcrumb trail: `[top] / data / repositories / internal / sentinela / sentinela_client / 1.6.0`, and a table of metadata:

Repository	internal
Group ID	data.repositories.internal.sentinela
Artifact ID	sentinela_client
Version	1.6.0
Packaging	

Below the metadata is a 'POM Snippet' section containing the following XML code:

```
<dependency>
  <groupId>data.repositories.internal.sentinela</groupId>
  <artifactId>sentinela_client</artifactId>
  <version>1.6.0</version>
  <type></type>
</dependency>
```

Deve-se inserir este trecho de XML dentro da tag `<dependencies>` do arquivo *pom.xml* e salvar o arquivo. Após esta ação o Maven se encarregará de “baixar” a(s) nova(s) dependência(s) para o repositório local.

Nos casos onde não forem encontradas as dependências pretendidas no repositório corporativo, deve-se procurar a biblioteca nos repositórios externos, conectados ao nosso gerenciador Archiva, são eles:

- Central Repository - <http://repo1.maven.org/maven2>
- Repositório do Jboss - <http://repository.jboss.com/maven2>
- JBoss Repository Org - <http://repository.jboss.org/maven2>
- Java.net Repository for Maven 2 - <http://download.java.net/maven/2/>
- JBoss Repository Org (Novo) - <http://repository.jboss.org/nexus/content/groups/public-jboss/>

Encontrado as biblioteca em um destes repositório, basta adicionar a dependência no arquivo *pom.xml* do projeto, conforme descrito anteriormente, que o *Archiva* se encarregará de “baixar” a(s) biblioteca(s) para o repositório corporativo, e o Maven para o repositório local.

Caso haja necessidade de utilizar bibliotecas existentes em outros repositórios externos (que não os informados anteriormente), deve-se realizar um pedido de inclusão a equipe da GIC. Esta equipe fará uma breve análise e não havendo problemas procederá a inclusão.

Atenção: qualquer tentativa de utilizar bibliotecas que não existam no repositório corporativo, resultarão em erros durante o *deploy* da aplicação para os servidores corporativos.

1.4.1 Repositórios

No arquivo *pom.xml* do projeto modelo (discutido anteriormente) estão definidos três repositórios de bibliotecas (localizados em nosso servidor *Archiva*) utilizados pelo Maven. Sendo que o repositório chamado *celepar* é o oficial das bibliotecas da empresa, o repositório chamado *internal* possui as bibliotecas de terceiros utilizadas nos projetos e o repositório denominado *snapshots* serve apenas para armazenar versões intermediárias, que não estão estáveis, portanto, não devem ser utilizadas em ambientes de produção. Segue definição do arquivo *pom.xml*:

```

<repositories>
  <repository>
    <id>celepar</id>
    <url>http://maven.celepar.parana/archiva/repository/celepar</url>
  </repository>
  <repository>
    <id>internal</id>
    <url>http://maven.celepar.parana/archiva/repository/internal</url>
  </repository>
  <repository>
    <id>snapshots</id>
    <url>http://maven.celepar.parana/archiva/repository/snapshots</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>

```

Conforme ilustrado na arquitetura (discutida anteriormente), o Maven não fará acesso direto a repositórios externos, ele sempre solicitará bibliotecas para o nosso *Archiva*, que por sua vez está autorizado e configurado para acessar repositórios externos.

1.4.2 Ciclo de Vida

Para executar alguma *goal* (meta) do ciclo de vida do Maven, basta, no Eclipse, clicar com o botão direito sobre o projeto e selecionar *Run as > Maven Package*, por exemplo, para empacotar o projeto. Recordando que o *output* do projeto fica na pasta *target*, logo os produtos resultantes da meta *package* serão automaticamente inseridos nesta pasta.

Para limpar a pasta *target* utilize a *goal clean* do ciclo de vida do Maven.

1.4.3 Deploy Local JBoss

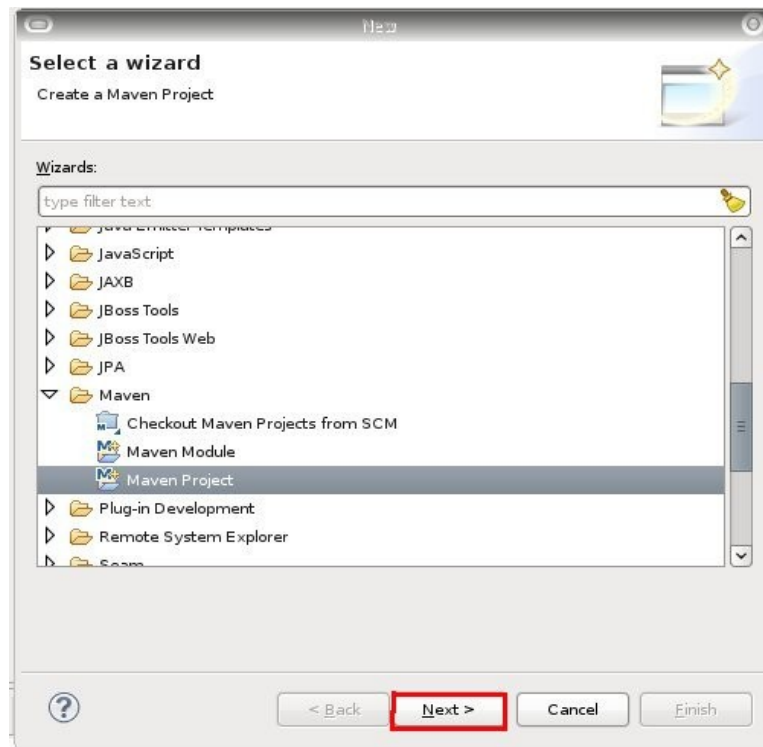
Para realizar o *deploy* local da aplicação podemos utilizar a *Eclipse View - Servers*. Adicionar um novo servidor *JBoss Community>JBoss as <version>*. Com um duplo clique no servidor adicionado, editar a aba *deployment*, alterando a forma de *deploy* para *Use the JBoss*

deploy folder. Depois basta adicionar a aplicação no servidor e inicia-lo.

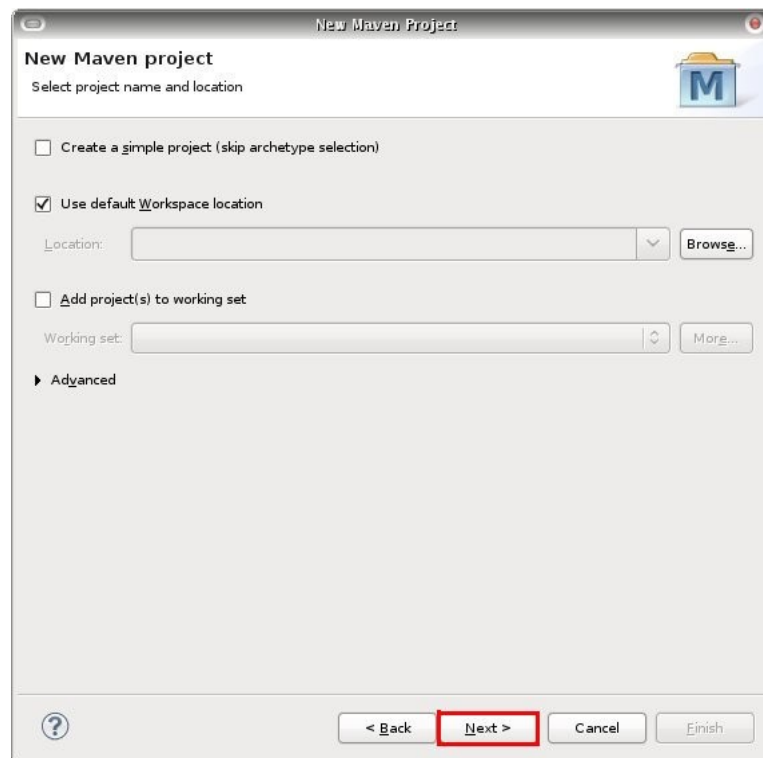
1.5 Guia rápido para configurar um projeto

Abaixo temos um resumo dos passos para se criar um projeto que fará uso do *Maven*:

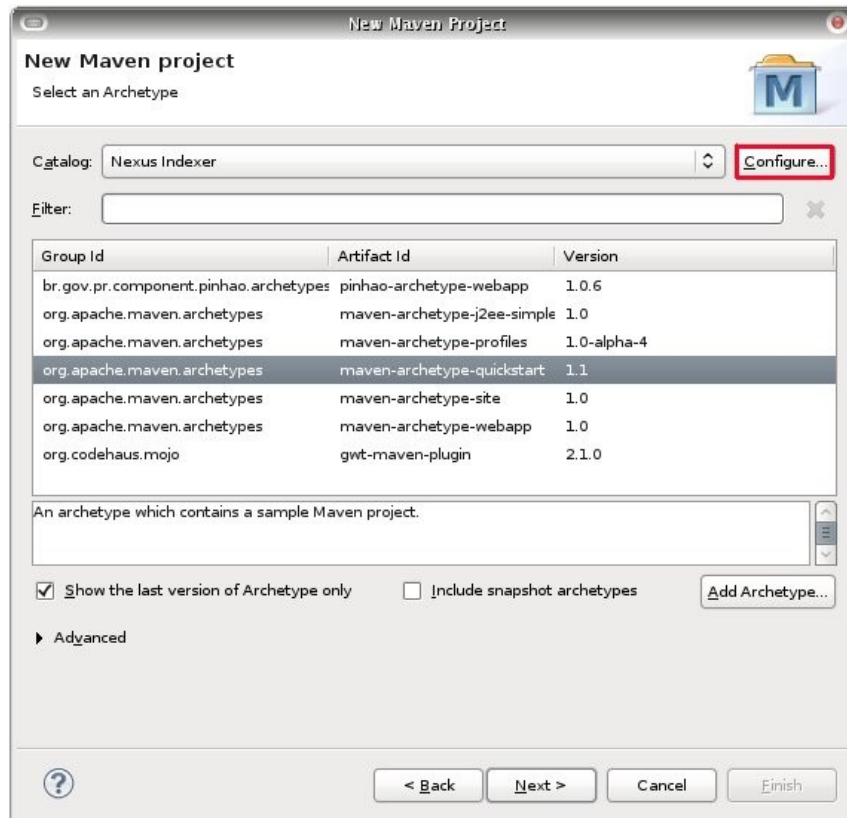
- a) Instalar o Eclipse: Verificar se sua instalação do Eclipse esta na versão 3.5.0-2 ou superior (comando: *dpkg --get-configuration-files eclipse-celepar*). Caso a versão seja inferior deve remover o Eclipse atual (*apt-get remove --purge eclipse-celepar*), atualizar os repositórios (*apt-get update*) e instala-lo novamente (*apt-get install eclipse-celepar*).
- b) Instalar o Maven: *apt-get install maven2*.
- c) Inserir o arquivo *settings.xml* (encontrado no link: <http://www.documentador.pr.gov.br/documentador/pub.do?action=d&uuid=@gtf-escriba@1cf94787-8746-4b6a-9047-1ae69721c6d0>) na pasta */home/<usuario>/.m2/*.
- d) Criar um novo projeto Maven: No Eclipse, use o botão *New >> Project*. Na lista de tipos de projetos vá até a opção *Maven >> Maven Project* e pressione o botão “Next”:



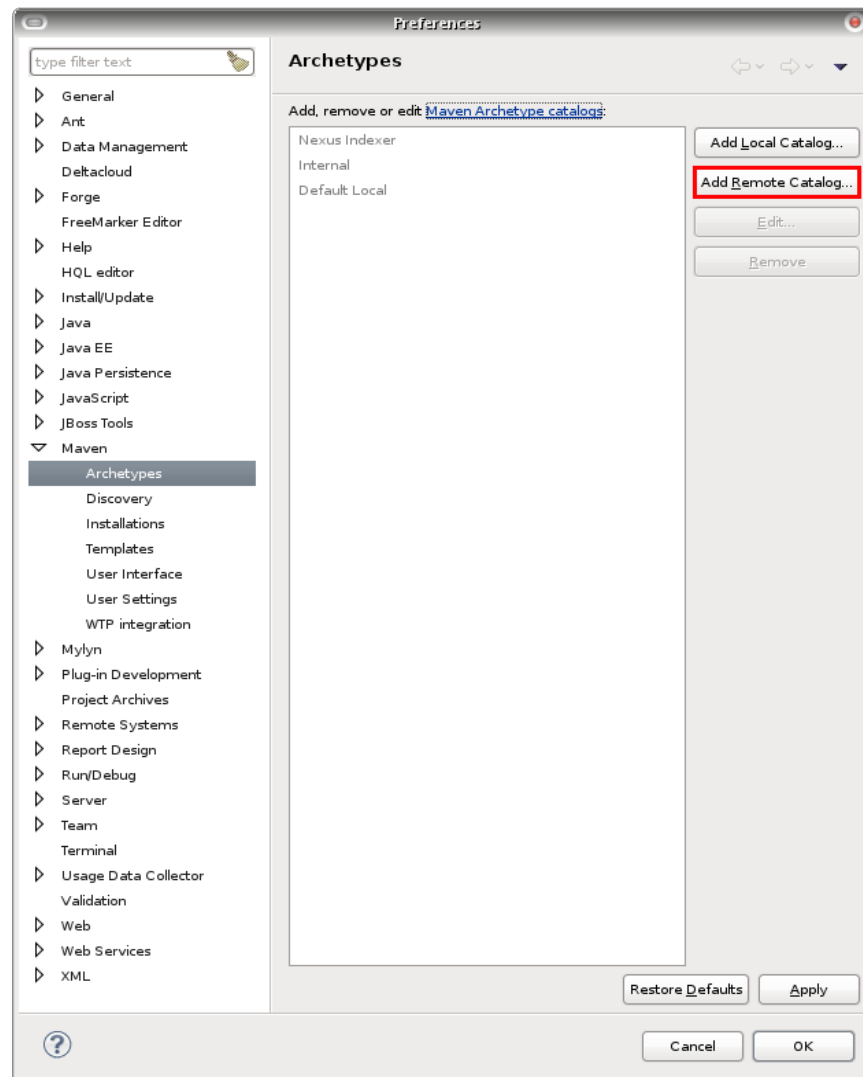
- Na tela seguinte apenas acionar o botão “Next”:



- Em seguida será mostrado os catálogos disponíveis, acionar o botão “Configure”:



- Será mostrada uma tela com todos os catálogos configurados. Clicar no botão “Add Remote Catalog” para adicionar o catálogo da Celepar:

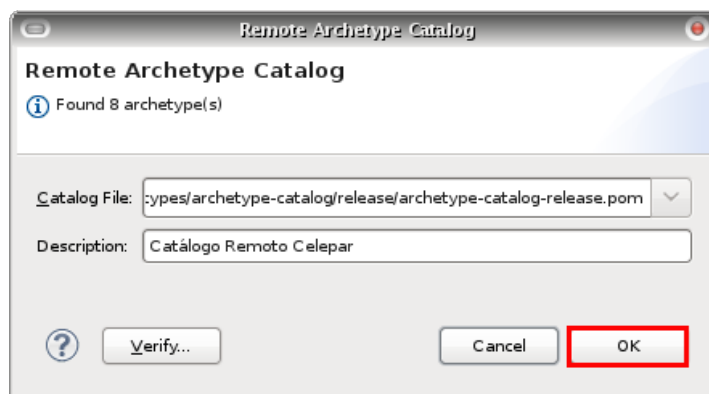


- A próxima tela possui dois campos:

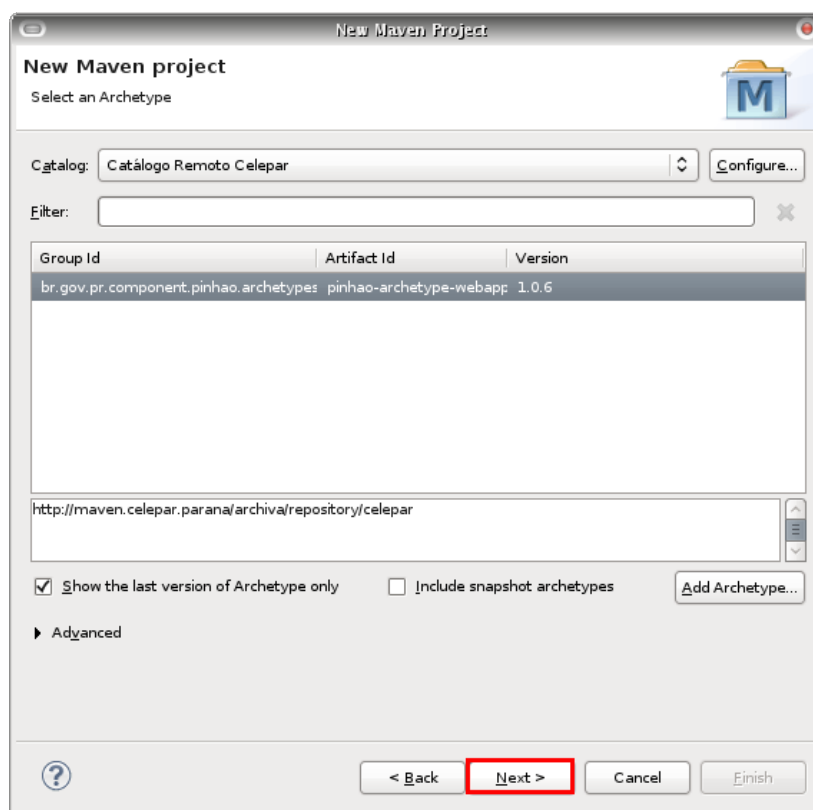
O primeiro “*Catalog File*”, representa o caminho do catálogo remoto (encontrado no <http://maven.celepar.parana>):

“<http://maven.celepar.parana/archiva/repository/celepar/br/gov/pr/component/pinhao/archetypes/archetype-catalog/release/archetype-catalog-release.pom>”

O segundo “*Description*”, representa o nome do catálogo: “Catálogo Remoto Celepar”.



- Confirmar as alterações e na tela de catálogos, escolher o “Catálogo Remoto Celepar”. Selecionar o *archetype* desejado.



- Insira as informações de *Group Id* e *Artefact Id* da aplicação e acione o botão “*Finish*”. Seu projeto *Maven* será criada conforme os padrões da Celepar.

New Maven Project
Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

Name	Value
------	-------

► Advanced

Obs: Maiores detalhes podem ser obtidos nas seções anteriores deste documento ou no conteúdo das referências.

2 REFERÊNCIAS

<http://maven.apache.org>

http://www.eteg.com.br/treinamento/material/Eteg_Maven_0108.pdf