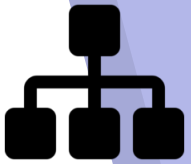
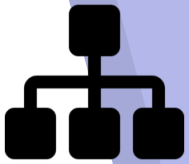
BUENAS PRÁCTICAS



- *Una jerarquía de módulos debe ser consistente*
- *Se deben evitar ciclos de dependencias*
- *En el POM maestro se deben definir:*
  - *las versiones de dependencias a través de propiedades*
  - *La declararación de plugins y dependencias:*
    - `<pluginManagement />`
    - `<dependencyManagement />`
  - *La declaración de repositorio remoto de despliegue:*
    - `<distributionManagement />`

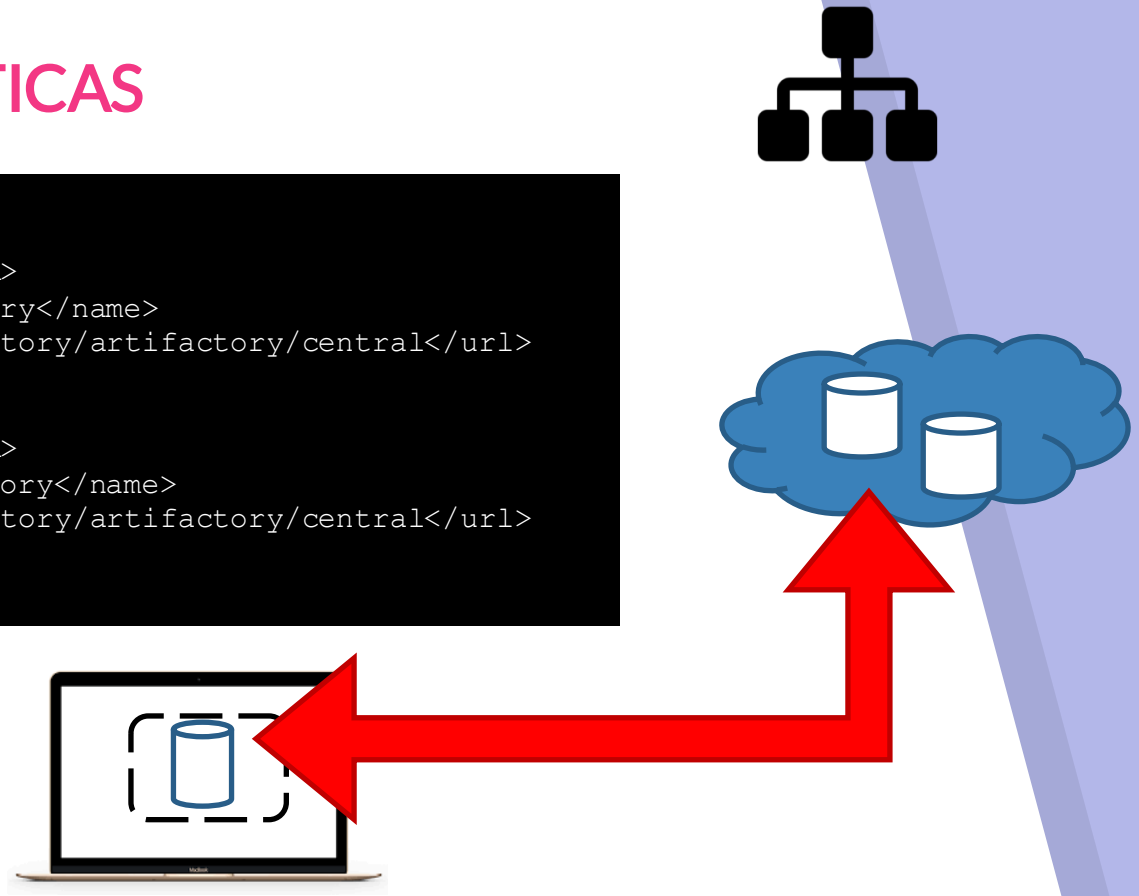
BUENAS PRÁCTICAS



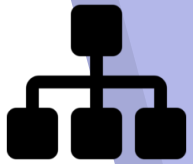
```
<build>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <version>2.6</version>
      </plugin>
    </plugins>
  </pluginManagement>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <configuration>
        <archive>
          <manifestFile>
            ${project.build.outputDirectory}/META-INF/MANIFEST.MF
          </manifestFile>
        </archive>
      </configuration>
    </plugin>
```

BUENAS PRÁCTICAS

```
<distributionManagement>
  <repository>
    <id>my_artifactory</id>
    <name>Release Repository</name>
    <url>http://my_artifactory/artifactory/central</url>
  </repository>
  <snapshotRepository>
    <id>my_artifactory</id>
    <name>Snapshot Repository</name>
    <url>http://my_artifactory/artifactory/central</url>
  </snapshotRepository>
</distributionManagement>
```



VENTAJAS VS DESVENTAJAS



- ***VENTAJAS***
  - *Los proyectos jerárquicos ofrecen una visión directa de la estructura de un proyecto complejo*
  - ***Well-defined***

- ***DESVENTAJAS***
  - *Aumenta la complejidad en los entornos de desarrollo*
  - *Se pueden producir problemas durante la publicación automática en servirdores de aplicaciones*