

# MAVEN

Marcelo Eduardo Guillen Castillo





```
va.io.*;  
va.util.Date;
```

```
ass SaveDate {
```

```
static void main(String args[])  
{  
    OutputStream fos = new FileOutputStream("date.txt");  
    ObjectOutputStream oos = new ObjectOutputStream(fos);  
    Date date = new Date();  
    oos.writeObject(date);  
    oos.flush();  
    oos.close();  
    fos.close();  
}
```

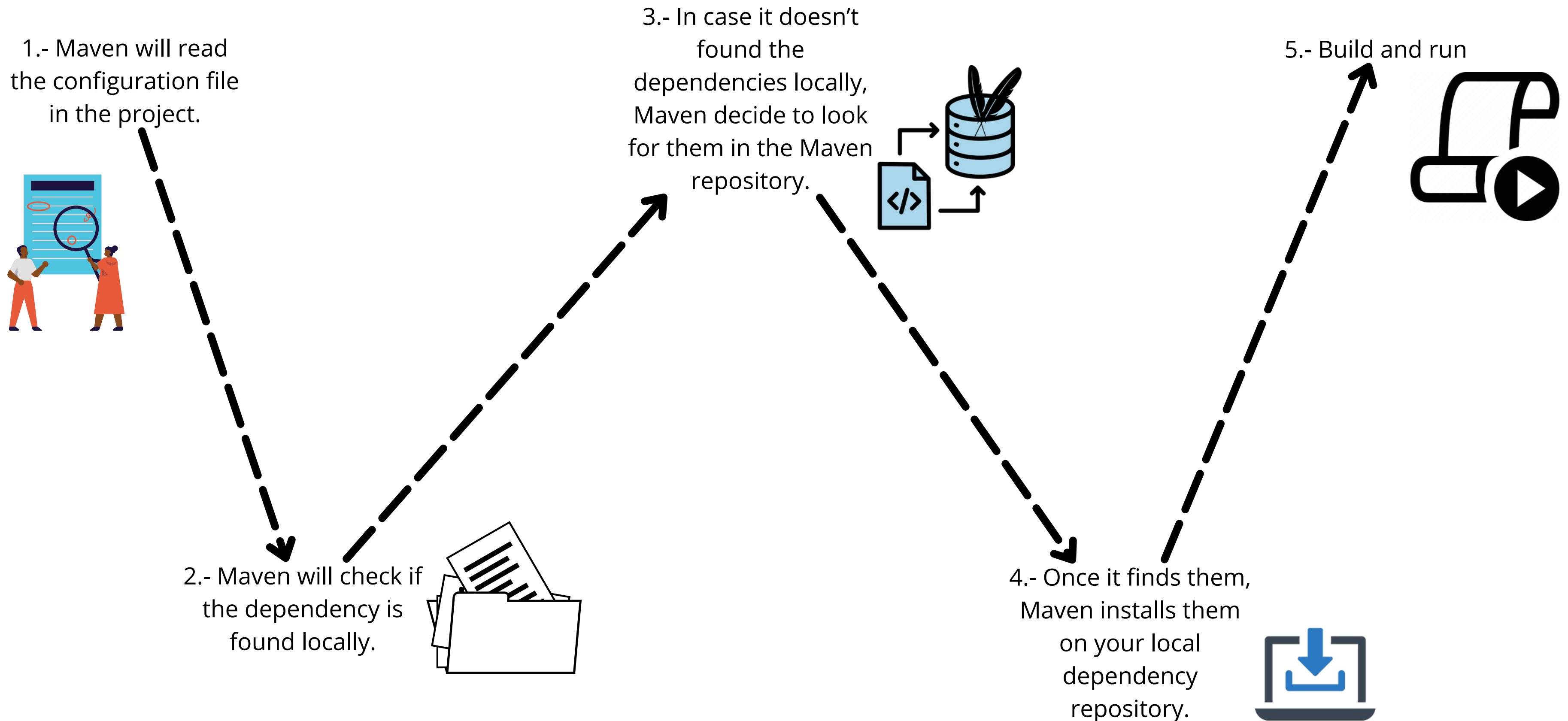
## ¿WHAT?

JAVA PROJECT MANAGER



It consists of a project manager for Java, which is capable of managing the dependencies or libraries that the user uses in their projects, installing them from the central Maven repository into the local project so that the user does not have to install them manually , in addition to providing testing codes and even code for Front-End.

# DEPENDENCY MANAGER



# PROYECT CONFIGURATION

In Maven, the project and its main configurations are managed in a file called POM (Project Object Modeling) in XML format. The file itself has a base structure divided into three sections.

```
pom.xml (BookStore)

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.ap
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example.maven</groupId>
  <artifactId>BookStore</artifactId>
  <packaging>pom</packaging>
  <version>1.0-SNAPSHOT</version>

  <modules>
    <module>
    </module>
  </modules>

  <profiles>
    <profile>
    </profile>
  </profiles>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
</project>
```

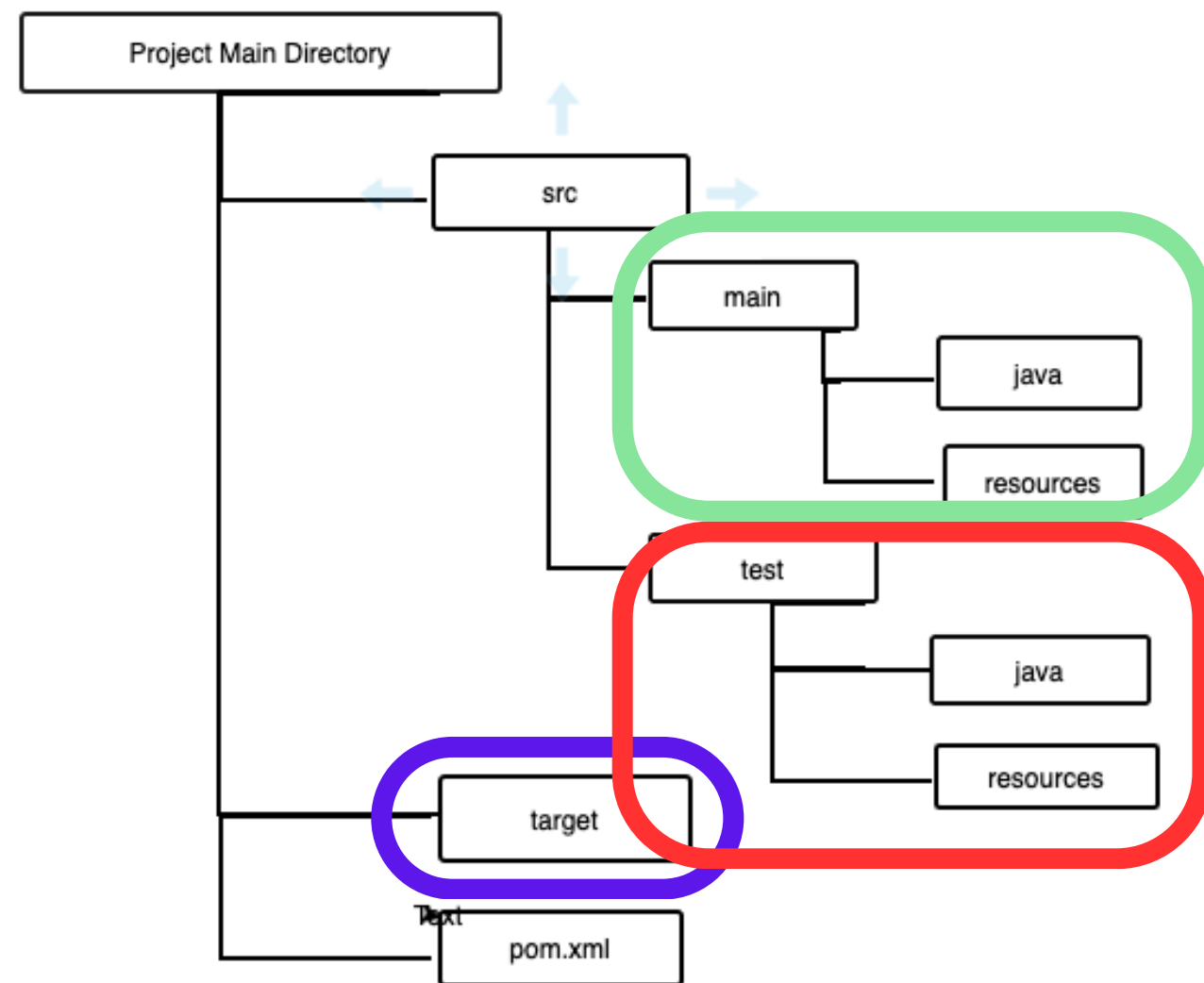
Header where metadata of the project is placed, like its name, version or artifactId

Body where all dependencies are described

Aditional dependencies

# PROYECT STRUCTURE

In Maven, the project and its main configurations are managed in a file called POM (Project Object Modeling) in XML format. The file itself has a base structure divided into three sections.



Contains the main code of your project, all the classes that you create, will be there.

Contains the testing codes for your main project.

Will contain your JAR build files from the main code.

## Additional References

<https://www.baeldung.com/maven-directory-structure>