

A - sobre as 3 ferramentas

A maioria dos IDEs para desenvolvimento em Java (e outras linguagens) usa um (ou vários) dos seguintes utilitários:

- ant
- maven
- gradle

notas sobre ant

O ant é um utilitário, usado como comando ou como biblioteca, para gerir processos e projetos.

Mais conhecido no âmbito das aplicações em java, pode ser usado para controlar a execução de tarefas com dependências.

Num projeto em NetBeans com ant, na pasta base do projeto, para além dos comandos que já usámos, ativando diretamente os comandos java ou javac, poderia também executar:

ant clean

ant compile

ant run

e é executada a classe principal do projeto, de acordo com o ficheiro de configuração (build.xml)

Mais informação:

- <https://ant.apache.org/>
- <https://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html>

notas sobre maven

Maven

- ferramenta para gestão de projetos (mais funcional que ant ou make).
- configuração: ficheiro **pom.xml** (nome, dependências, plugins)
- comando: mvn
- uso frequente:
mvn clean, mvn compile, mvn exec:java, mvn test, mvn package, mvn install

A opção test aplica os testes unitários ao projeto, para automatizar a validação de uma nova versão do código fonte.

A opção package arruma os recursos necessários, sem código fonte, num ficheiro jar com tudo o necessário para a aplicação executar.

Como passar argumentos em tempo de execução, ao método main(), via pom.xml:

```
<configuration>
  <mainClass>so2.App</mainClass>
  <arguments>
    <argument>Um Argumento para o main()</argument>
    ...
  </arguments>
  ...
</configuration>
```

O maven faz automaticamente download das bibliotecas necessárias à compilação ou à execução de uma aplicação. Na primeira execução vai observar o download de vários pacotes (para a pasta ~/.m2).

Ajuda também na preparação de documentação ou site de um projeto.

Mais informação:

<http://maven.apache.org/>

<https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

Como instalar em Debian/Ubuntu:

```
$ sudo apt install maven
```

Como saber os **goals** para um determinado plugin:

```
$ mvn plugin:help
```

Exemplo para listar goals de SpringBoot:

```
$ mvn spring-boot:help
```

Exemplo de execução na linha de comandos de qualquer classe executável do projeto:

```
$ mvn exec:java -Dexec.mainClass=sd.ClassName
```

```
$ mvn exec:java -Dexec.mainClass=sd.ClassName -Dexec.args="hostname port"
```

Testes JUnit:

<http://www.oracle.com/technetwork/articles/java/integrationtesting-487452.html>

https://netbeans.org/kb/docs/java/junit-intro.html#Exercise_30

notas sobre gradle

O **gradle** é uma ferramenta de desenvolvimento, open-source, muito apreciada pela sua flexibilidade, e funcionalidades avançadas na gestão de dependências e processos de engenharia de software (**continuous integration**), usável em IDEs e via linha de comandos.

Como instalar em Ubuntu?

```
$ sudo apt install gradle
```

Caso precise de efetuar upgrade para uma versão superior à que existe disponível nos pacotes do sistema, pode obter uma versão mais recente, em formato binário, de:

<https://gradle.org/install/#manually>

a) aplique os passos 1 e 2 (download e arrumação no disco)

b) ative o binário do gradle:

- passo prévio, no caso de já ter um gradle (4.x) no seu sistema:
 - `sudo apt remove gradle`
- uma de duas:
 - altere a sua variável PATH para incluir a pasta bin da aplicação;
 - ou coloque um link para o executável numa das pastas de executáveis do sistema:
 - `sudo ln -s /opt/gradle/gradle-7.2/bin/gradle /usr/bin/gradle`
 - # (pasta para onde descompactou o zip, depois o link que apontará para o principal executável)
 - `gradle -version`
 - Welcome to Gradle 7.2!

Como criar uma aplicação em java, com linha de comandos:

<https://guides.gradle.org/building-java-applications/>

<https://docs.gradle.org/current/userguide/userguide.html>

O gradle **wrapper** (gradlew) permite a utilização do gradle sem o ter instalado realmente no sistema. É um script que, juntamente com software "local" adicional, realiza as operações do gradle (build...). Não é necessário, se tiver o gradle efetivamente instalado, poderá acelerar a troca da versão de gradle. Como o gerar:

```
$ gradle wrapper
```

Listar as tasks disponíveis:

```
$ gradle tasks
```

Para quem está familiarizado com maven, pode ver uma analogia com gradle aqui:

<https://gradle.org/maven-vs-gradle/>

B - gradle quick start

Seguindo [um exemplo](#) da documentação de gradle:

1. verifique a versão de gradle no seu PC
 - \$ gradle -version
 - Gradle 7.0
2. criar uma pasta para um projeto de demonstração:
 - mkdir gradle.demo
 - cd gradle.demo
3. Criar automaticamente a estrutura inicial do projeto para uma aplicação em Java, com a task init:
 - gradle init
 - type of project: 2. application
 - implementation language: Java
 - Split functionality across multiple subprojects?: no
 - DSL= 'Domain Specific Language': 1 - [Groovy](#)
 - Select test framework: 1 - JUnit 4
 - Project name: gradle.demo
4. O que podemos acionar com o comando gradle?
 - gradle tasks
5. Executar o projeto:
 - \$ gradle run
 - ou:
 - \$./gradlew run # o mesmo, mas com o wrapper
6. Rever os ficheiros (após gradle clean):

```

◦
◦ |— app
◦ | |— build.gradle
◦ | |— src
◦ | |   |— main
◦ | |   |   |— java
◦ | |   |   |   |— gradle
◦ | |   |   |   |   |— demo
◦ | |   |   |   |   |— App.java
◦ | |   |   |— resources
◦ | |   |— test
◦ | |   |   |— java
◦ | |   |   |   |— gradle
◦ | |   |   |   |   |— demo
◦ | |   |   |   |   |— AppTest.java
◦ | |   |   |— resources
◦ |— gradle
◦ |   |— wrapper
◦ |   |   |— gradle-wrapper.jar
◦ |   |   |— gradle-wrapper.properties
◦ |— gradlew
◦ |— gradlew.bat
◦ |— settings.gradle
◦

```

7. **Nota:** se o seu sistema tiver uma versão de gradle compatível, poderia apagar a pasta gradle e o wrapper, ficando só com o conteúdo da pasta app.
8. Aplicar os testes unitários:
 - gradle test
9. Ver o resultado dos testes unitários
 - abra a página de relatório com o seu browser. Exemplo:

- \$ firefox app/build/reports/tests/test/index.html

Opcional: gradle [Build Scan](#)

a task "build --scan"

pode organizar informação sobre o projeto, como as dependências, memória usada, etc...

Note que esta informação fica online em cloud (cautela na exposição de projetos privados).

Veja um exemplo em:

<https://gradle.com/s/5u4w3gxeurtd2>

C- transformar gradle project em maven project

Estas instruções têm alguns nomes e dependências específicas de um projeto inicial fornecido... Poderá ter de rever dependências ou nomes!

1- Editar o Gradle build script (build.gradle) e adicionar:

```
apply plugin: 'maven-publish'

publishing {
    publications {
        mavenJava(MavenPublication) {
            artifactId = 'tw'
            groupId = 'spring-boot-security-jdbc-auth_tw'
            version = 1.0
            from components.java
            pom {
                name = 'spring-boot-security-jdbc-auth_tw'
                description = 'spring-boot-security-jdbc-auth_tw description'
                properties = [
                    "maven.compiler.source": "1.8",
                    "maven.compiler.target": "1.8" ]
            }
        }
    }
}
```

2- Gerar o pom.xml:

\$ gradle build

\$ gradle generatePom

3- E trazer o ficheiro gerado para a pasta base:

\$ cp build/publications/mavenJava/pom-default.xml pom.xml

4- Executar a aplicação, desta vez com maven:

\$ mvn compile

\$ mvn spring-boot:run

Caso não tenha a opção spring-boot, adicione o plugin ao pom.xml:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Se algo correu mal (classes não encontradas?), veja se é preciso alterar o scope das dependências, no pom.xml:

```
<scope>compile</scope> /* troque para este scope as 4 primeiras dependências */
```

5- Se tudo estiver a funcionar, e para não haver confusão entre gradle e maven quando o projeto é aberto no IDE, pode opcionalmente apagar a parte do gradle:

```
$ gradle clean
```

```
$ rm -r .gradle build.gradle
```

 [Contactar suporte do site](#) 

Nome de utilizador: [Rodrigo Alves](#) ([Sair](#))

[Resumo da retenção de dados](#)

[Obter a Aplicação móvel](#)

Fornecido por [Moodle](#)