

Como gerar um .jar executável?

Um **JAR** nada mais é do que um arquivo compactado, como se fosse um arquivo “.zip”, que contém o código intermediário de um projeto Java que pode ser executado pela **JVM** (Java Virtual Machine).

Esse tutorial não visa explicar nos mínimos detalhes o que é e como funciona, mas sim, demonstrar uma forma (existem algumas) de gerar o .jar de sua aplicação, para que consiga visualizar na **segunda sprint**, uma versão “**intermediária**” do artefato que será entregue ao final desse ciclo(semestre/ano).

Vamos lá, faça o seguinte:

1. Abra seu projeto Java no **NetBeans**;
2. Localize o arquivo **pom.xml**;
3. Insira o seguinte conteúdo após o fechamento da tag **properties**:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-assembly-plugin</artifactId>
      <version>2.4.1</version>
      <configuration>
        <descriptorRefs>
          <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
        <archive>
          <manifest>
            <mainClass>school.sptech.exemplo.imagem.jframe.jar.executavel.Demo</mainClass>
          </manifest>
        </archive>
      </configuration>
      <executions>
        <execution>
          <id>make-assembly</id>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

4. Feito isso, é só substituir a classe em destaque (negrito) para a sua classe que contém o **método main (tela de início ou classe inicial do projeto)**.

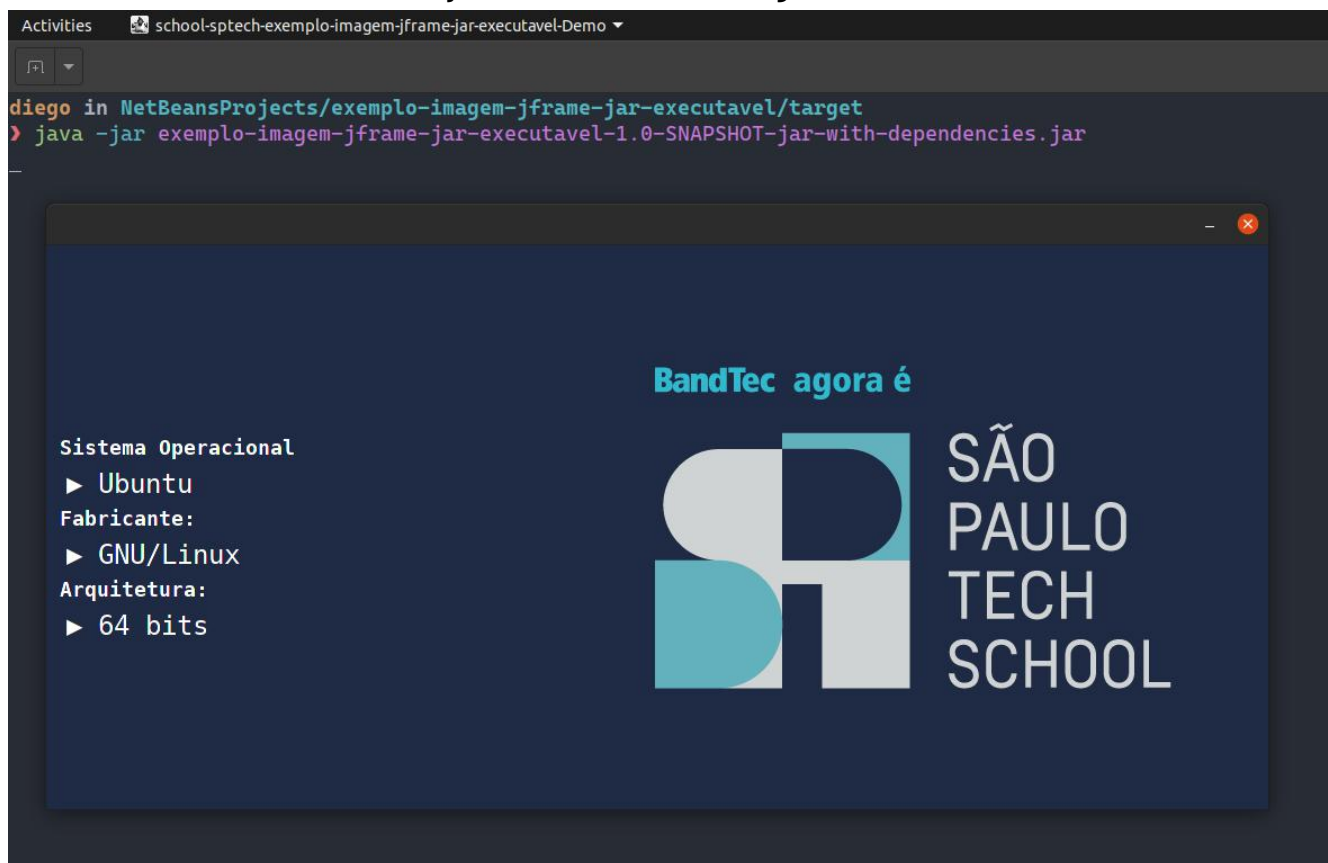
5. Clique com o botão direito no projeto (lado direito) e selecione a opção **"Build with dependencies"**. Logo após, uma pasta chamada **"target"** aparecerá e dentro dela haverá o **.jar** executavel com as dependências:

```
diego in ~/NetBeansProjects/exemplo-imagem-jframe-jar-executavel
> tree
.
├── pom.xml
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── school
│   │   │   │   ├── sptech
│   │   │   │   │   ├── exemplo
│   │   │   │   │   │   ├── imagem
│   │   │   │   │   │   │   ├── jframe
│   │   │   │   │   │   │   │   ├── jar
│   │   │   │   │   │   │   │   └── executavel
│   │   │   │   │   │   │   │       ├── Demo.form
│   │   │   │   │   │   │   │       └── Demo.java
│   │   │   └── resources
│   │   │       ├── assets
│   │   │       │   └── sptech_logo.png
│   │   └── test
│   │       └── java
│   └── target
│       ├── archive-tmp
│       ├── classes
│       │   ├── assets
│       │   │   └── sptech_logo.png
│       │   ├── school
│       │   │   ├── sptech
│       │   │   │   ├── exemplo
│       │   │   │   │   ├── imagem
│       │   │   │   │   │   ├── jframe
│       │   │   │   │   │   │   ├── jar
│       │   │   │   │   │   │   │   └── executavel
│       │   │   │   │   │   │   │       ├── Demo$1.class
│       │   │   │   │   │   │   │       └── Demo.class
│       │   ├── exemplo-imagem-jframe-jar-executavel-1.0-SNAPSHOT.jar
│       │   ├── exemplo-imagem-jframe-jar-executavel-1.0-SNAPSHOT-jar-with-dependencies.jar
│       │   ├── generated-sources
│       │   │   └── annotations
│       │   ├── maven-archiver
│       │   │   └── pom.properties
│       │   ├── maven-status
│       │   │   ├── maven-compiler-plugin
│       │   │   │   └── compile
│       │   │   │       ├── default-compile
│       │   │   │       │   ├── createdFiles.lst
│       │   │   │       │   └── inputFiles.lst
│       └── test-classes
└── 33 directories, 12 files
```

Obs. Essa pasta target é equivalente a node_modules, ou seja, ele sempre será gerada automaticamente quando você executar comandos de construção do Maven (Clean Build, Build with dependencies e etc). Assim como no Node, devemos evitar subir essa pasta para o git, para isso, basta inseri-la no arquivo .gitignore de seu projeto.

6. Para executar, basta ir até o diretório onde está o .jar, nesse caso, dentro de /target.

Dentro do diretório onde está o jar, basta executar o seguinte comando:



```
Activities school-sptech-exemplo-imagem-jframe-jar-executavel-Demo
diego in NetBeansProjects/exemplo-imagem-jframe-jar-executavel/target
> java -jar exemplo-imagem-jframe-jar-executavel-1.0-SNAPSHOT-jar-with-dependencies.jar
```

BandTec agora é

Sistema Operacional
▶ Ubuntu
Fabricante:
▶ GNU/Linux
Arquitetura:
▶ 64 bits

SÃO PAULO TECH SCHOOL

Pronto! Seu software construído em Java pode ser executado em qualquer sistema que possua uma JRE (Java Runtime Environment) devidamente instalada.

Caso possua alguma dúvida entre em contato por email, o projeto utilizado para explicar como inserir arquivos de mídia está parametrizado caso queira testar antes de colocar em prática no seu projeto de PI.

Forte abraço e bons estudos!