

## Gerando nosso fatjar

### Transcrição

Com a aplicação finalizada, precisamos enviá-la para o cliente. Como iremos fazer isso? Vamos empacotar toda a aplicação em um único JAR e enviá-la.

Semelhante ao que fizemos com a aplicação `enviadorEmail`, iremos ao diretório da aplicação `Lista VIPs` e usaremos o comando `mvn package`. Assim o *Maven* fará o empacotamento da aplicação e a deixará disponível na pasta `target` do projeto.

Após o empacotamento, devemos testar a aplicação a fim de verificar que está tudo funcionando perfeitamente. Usando para isto o comando `java -jar` passando para o comando o arquivo gerado pelo *Maven*. Por exemplo:

```
java -jar target/listavip-1.0-SNAPSHOT.jar
```

Mas por algum motivo, recebemos um erro:

```
no main manifest attribute, in target/listavip-1.0-SNAPSHOT.jar
```

O que acontece é que não instruímos ao *Maven* como o empacotamento da aplicação deve ser feito. Também não iremos fazer isso manualmente. Iremos utilizar um plugin do *Spring Boot* para isto.

A [documentação](http://docs.spring.io/spring-boot/docs/current/reference/html/build-tool-plugins-maven-plugin.html#build-tool-plugins-include-maven-plugin) (<http://docs.spring.io/spring-boot/docs/current/reference/html/build-tool-plugins-maven-plugin.html#build-tool-plugins-include-maven-plugin>) do *Spring Boot* além de explicar como gerar o pacote com nossa aplicação, disponibiliza o plugin e a regra de como o *build* será realizado. O *Maven* executará a seguinte regra ao empacotar a aplicação - Insira esta regra no `pom.xml`.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <version>1.4.2.RELEASE</version>
      <executions>
        <execution>
          <goals>
            <goal>repackage</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

Para garantir que o novo pacote seja gerado sem correr o risco de ainda utilizar coisas do pacote gerado anteriormente, utilizaremos o comando `mvn clean`. Assim o *Maven* fará uma limpeza na pasta `target`. Após isso basta executar o comando `mvn package` novamente.

Agora, ao executar o comando:

```
java -jar target/listavip-1.0-SNAPSHOT.jar
```

A aplicação deve iniciar sem problemas.