

03

Empacotando o jogo

Capítulo 8 - Empacotando o jogo

Para terminarmos o curso de Closure, vamos empacotar nosso jogo de forca. Criamos uma função que inicia o jogo:

```
(defn comeca-o-jogo [] (jogo total-de-vidas palavra-secreta #{}))
```

Ela não recebe nada e retorna o total de vidas, que definimos como 6, e a palavra secreta que também precisaremos definir:

```
(def palavra-secreta "MELANCIA")
```

É necessário também implementar a função principal (`main`) que invoca o `comeca-o-jogo`:

```
(defn -main [& args]
  (comeca-o-jogo))
```

No terminal não usaremos o `Lein REPL`, pois agora queremos rodar o jogo. Então, dentro do diretório do programa fazemos:

```
lein run
```

E pronto! Nosso jogo está rodando perfeitamente.

Como havíamos comentado anteriormente, o código em Closure roda na JVM. Então conseguimos empacotar o programa criando um arquivo `.jar`:

```
lein uberjar
Compiling forca.core
...
```

Com isso criamos um código bitcode que roda na JVM.

Chegamos ao final do nosso curso de *Introdução à Programação Funcional*. Esperamos ter passado o conhecimento sobre como pensar diferente usando a linguagem funcional, em contraposição à imperativa. Desde o começo esse foi nosso objetivo, mais do que ensinar a programar em Closure. Comece a pensar em como aplicar esse conhecimento na sua linguagem, pois Java, C#, Scala, Ruby, por exemplo, dão suporte para a programação funcional. Até o próximo curso!

