

Programação Funcional

Unidade 11 – Interação com Banco de Dados

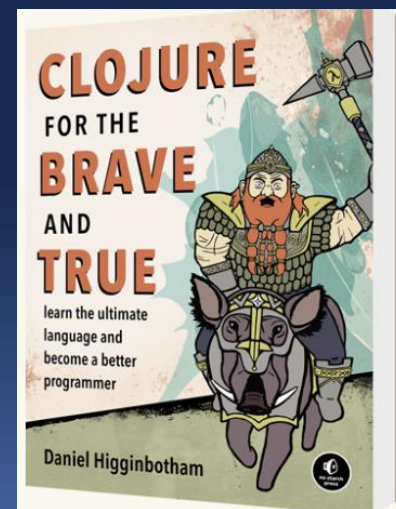
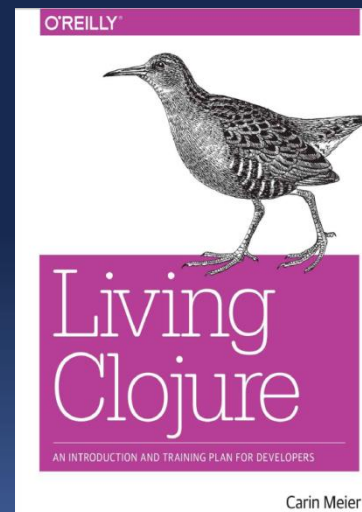
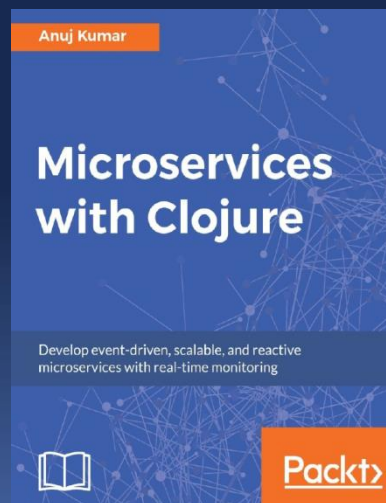
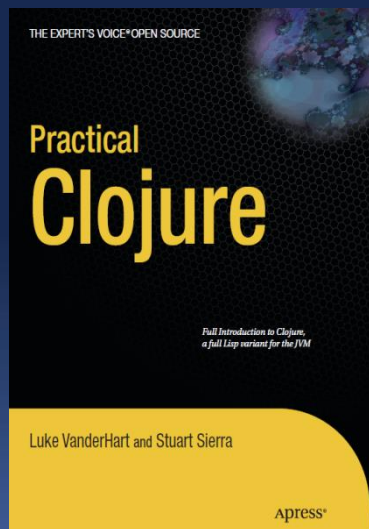
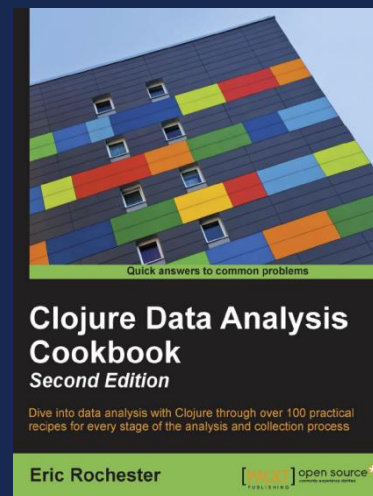
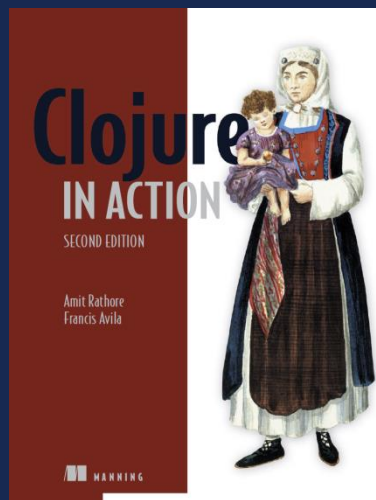
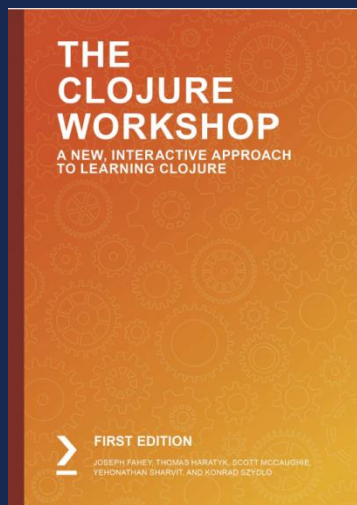


Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecido.freitas@prof.uscs.edu.br
aparecidovfreitas@gmail.com



Revisão Técnica: Mauricio Szabo
mauricio.szabo@gmail.com

Bibliografia



Introdução

- Cada vez que reiniciamos nosso **REPL**, perdemos os dados manipulados previamente;
- Veremos nessa unidade a interação de Clojure com sistemas de bancos de dados relacionais;
- Utilizaremos, por simplicidade, o **RDBMS Apache Derby**, implementado inteiramente em **Java**, proporcionando dessa forma mais tempo para focar em aspectos de modelagem de dados, persistência e recuperação.



Conectando ao Banco de Dados

- Em termos de interação com o banco de dados Apache **Derby** a partir da linguagem Clojure, utilizaremos a **Clojure.java.jdbc**, um biblioteca de baixo nível para interagir com o Banco de Dados, por meio da **JDBC**.



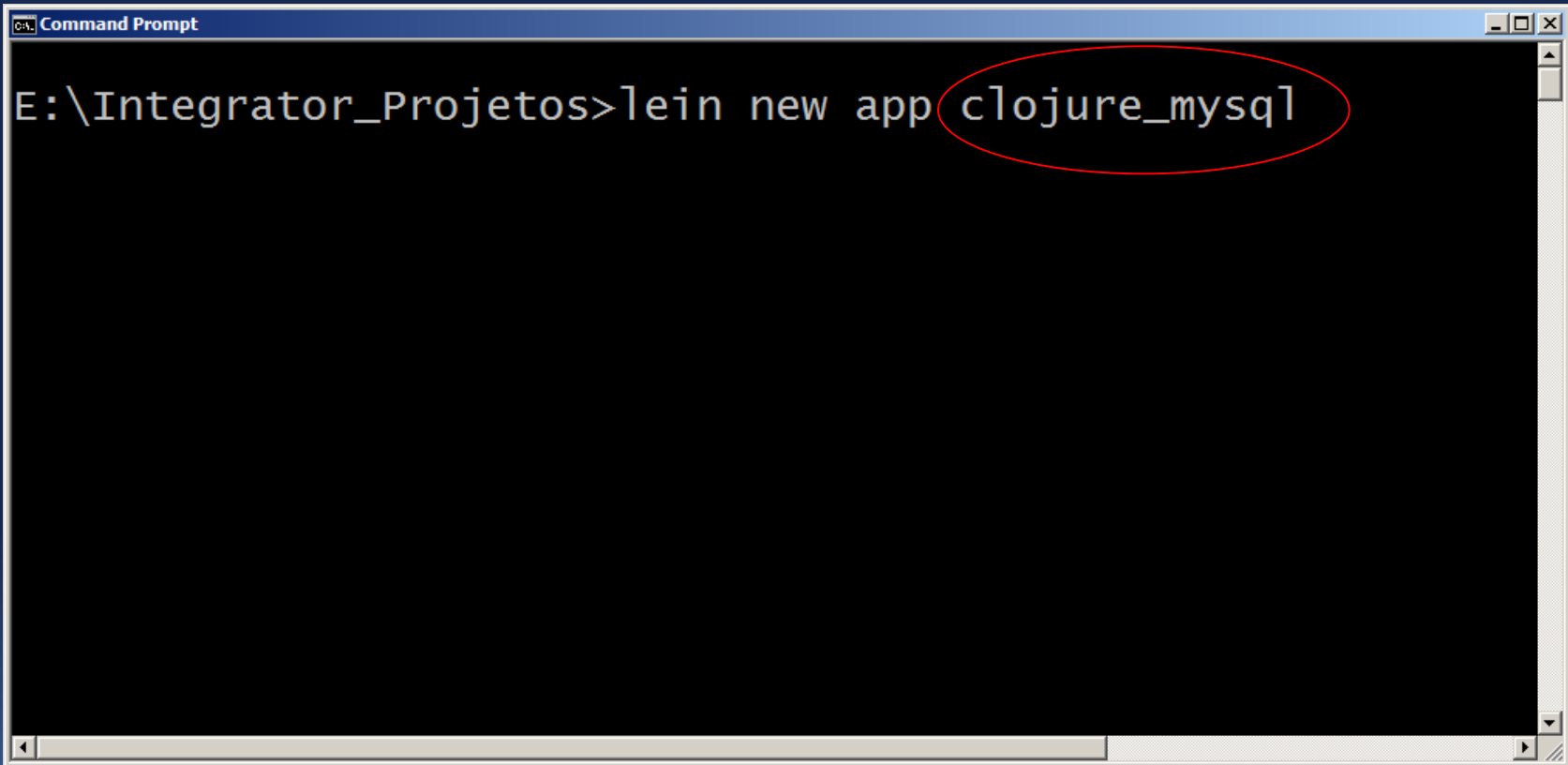


Acessando MySQL com projeto Leiningen



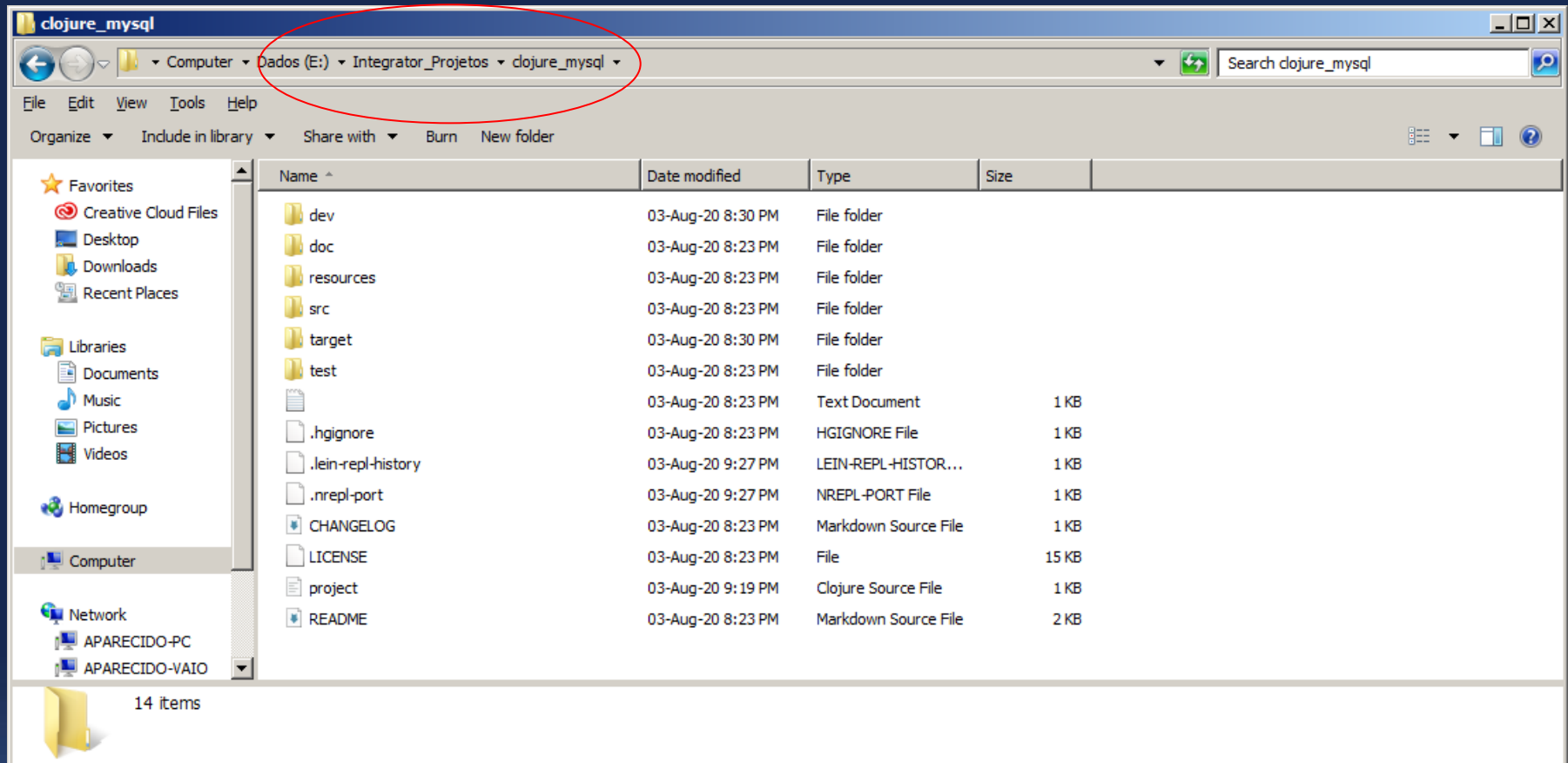
Recursos

Criando o projeto com Lein



```
Command Prompt
E:\Integrator_Projetos>lein new app clojure_mysql
```


Projeto criado



Configurando o projeto

- Incluindo no arquivo **project.clj** as dependências do **jdbc** e do Driver jdbc **MySQL**

```
project.clj
(defproject clojure_mysql "0.1.0-SNAPSHOT"
  :description "FIXME: write description"
  :url "http://example.com/FIXME"
  :license { :name "EPL-2.0 OR GPL-2.0-or-later WITH Classpath-exception-2.0"
            :url "https://www.eclipse.org/legal/epl-2.0/" }
  :repl-options { :init-ns user }

  :dependencies [ [org.clojure/clojure "1.10.1"]
                  [org.clojure/java.jdbc "0.7.9"]
                  [mysql/mysql-connector-java "8.0.21"] ]
  :jvm-opts [ "-Dclojure.server.myrepl={:port,5555,:accept,clojure.core.server/repl}" ]

  :main ^:skip-aot clojure_mysql.core
  :target-path "target/%s"
  :profiles { :uberjar { :aot :all
                        :jvm-opts [ "-Dclojure.compiler.direct-linking=true" ]
                        :dev { :dependencies []
                              :source-paths [ "dev" ] }}} )
```


Criando o Banco de Dados

- Banco de Dados: scpe
- Porta: 3307
- Host: localhost
- Tabela: cursos

```
use scpe;
```

```
describe curso;
```

Result Grid						
		Filter Rows:				
		Export:	Wrap Cell Content:			
	Field	Type	Null	Key	Default	Extra
▶	id_curso	int(11)	NO	PRI	NULL	auto_increment
	datetime_curso	varchar(255)	YES		NULL	
	nome_curso	varchar(255)	YES		NULL	
	timestamp_curso	varchar(255)	YES		NULL	

```
select * from curso;
```

Result Grid				
		Filter Rows:		
		Edit:	Export/Import:	
	id_curso	datetime_curso	nome_curso	timestamp_curso
▶	1	2020-07-20 21:51:15	Sistemas de Informação	2020-07-20 21:51:15
	2	2020-07-20 21:51:18	Gestão de Tecnologia da Informação	2020-07-20 21:51:18
	3	2020-07-20 21:51:21	Tecnologia ADS	2020-07-20 21:51:21

Código Clojure para acessar a tabela

core.clj

```
(ns clojure_mysql.core
  (:gen-class))

(require '[clojure.java.jdbc :as jdbc])

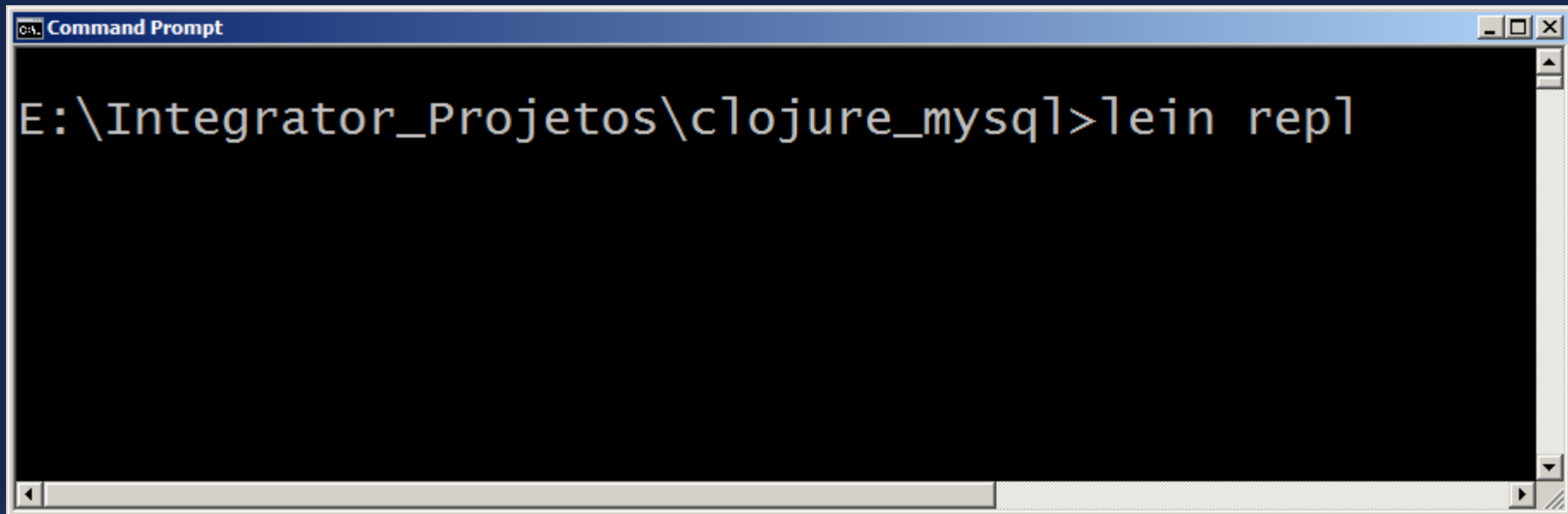
(let [db-host "localhost"
      db-port 3307
      db-name "scpe"]

  (def db {:classname "com.mysql.cj.jdbc.Driver"
           :subprotocol "mysql"
           :subname (str "://" db-host ":" db-port "/" db-name "?useTimezone=true&serverTimezone=UTC")
           :user "root"
           :password "maua"}))

(def resp (jdbc/query db ["SELECT * FROM curso WHERE id_curso = 1"]))

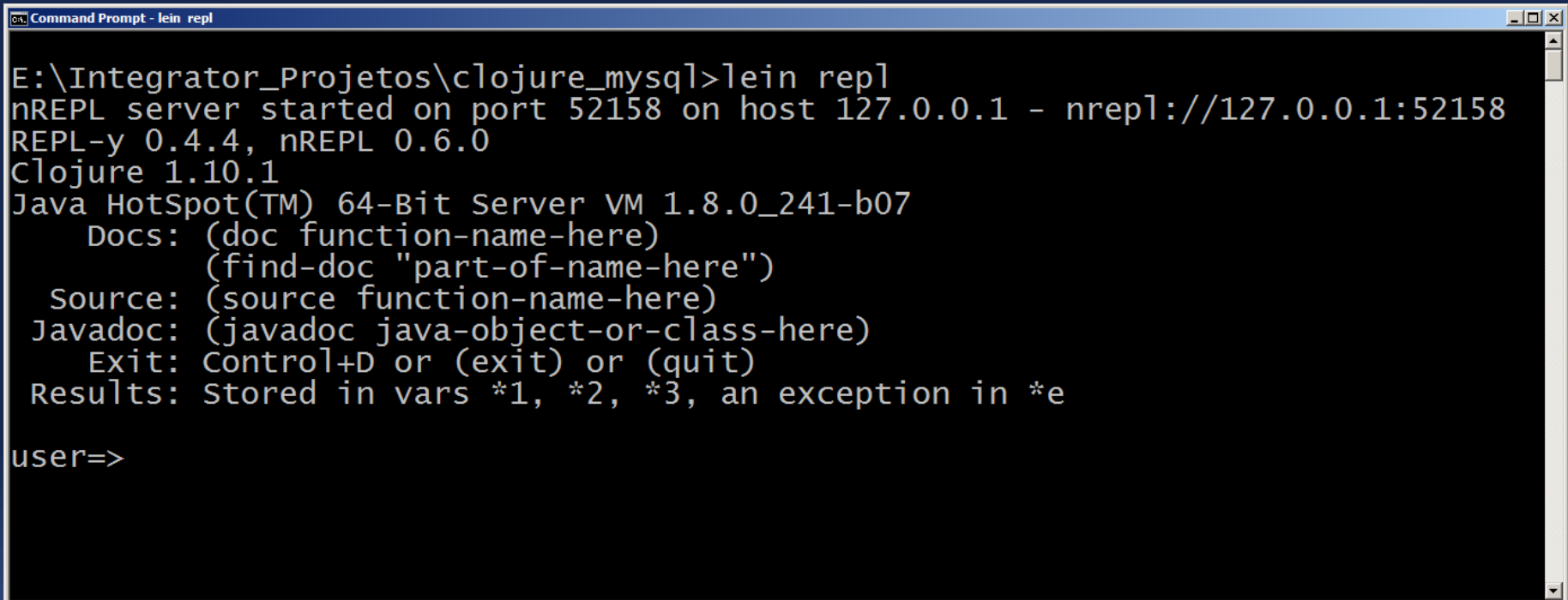
(defn -main
  "Código clojure para acessar MySQL via jdbc"
  [& args]
  (println resp))
```

Executando sob REPL - Chlorine



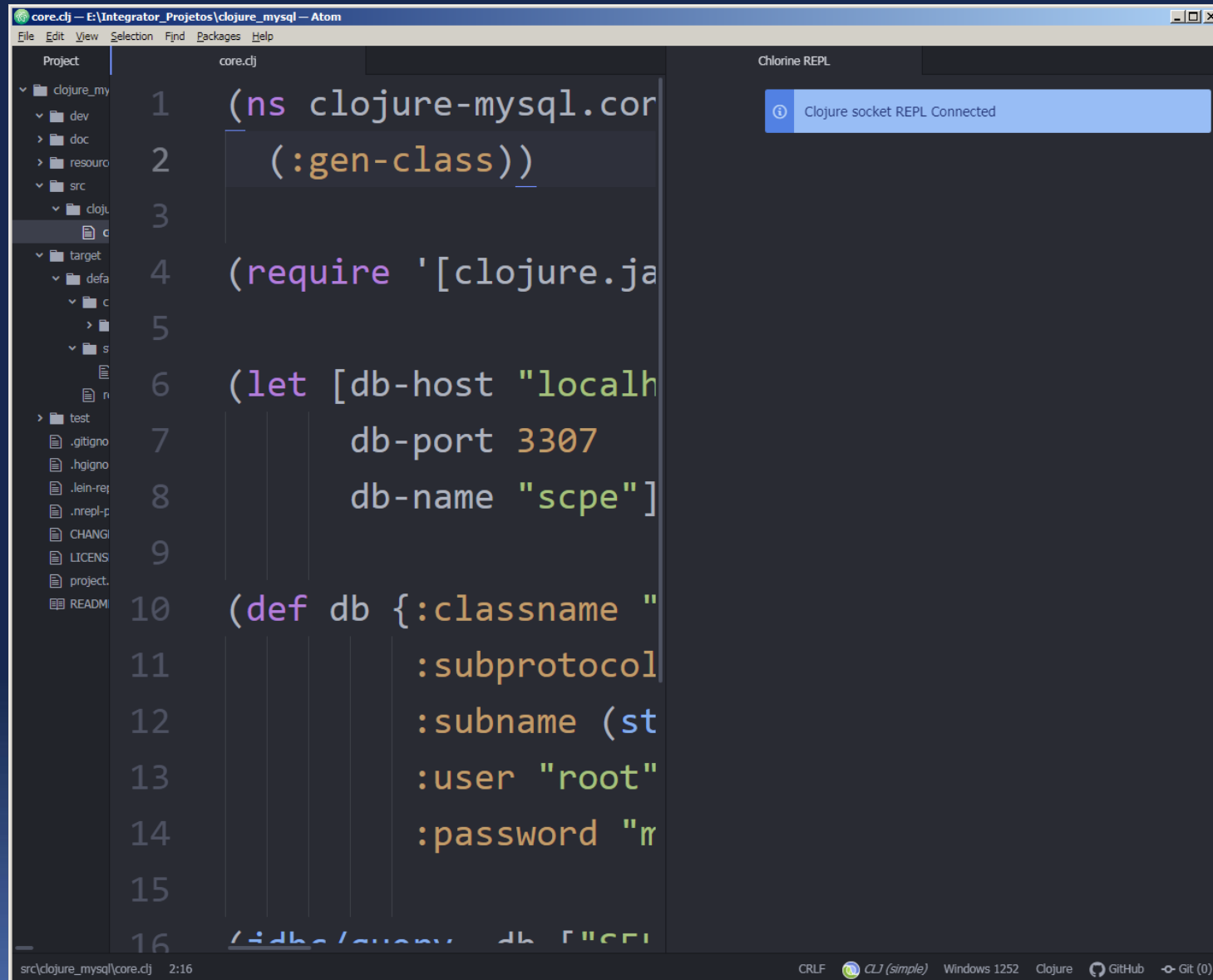
```
Command Prompt
E:\Integrator_Projetos\clojure_mysql>lein repl
```

Servidor REPL ativo



```
Command Prompt - lein repl
E:\Integrator_Projetos\clojure_mysql>lein repl
nREPL server started on port 52158 on host 127.0.0.1 - nrepl://127.0.0.1:52158
REPL-y 0.4.4, nREPL 0.6.0
Clojure 1.10.1
Java HotSpot(TM) 64-Bit Server VM 1.8.0_241-b07
  Docs: (doc function-name-here)
        (find-doc "part-of-name-here")
  Source: (source function-name-here)
  Javadoc: (javadoc java-object-or-class-here)
  Exit: Control+D or (exit) or (quit)
  Results: Stored in vars *1, *2, *3, an exception in *e
user=>
```

Conectando Atom Chlorine



The screenshot shows the Atom editor interface with a project named 'core.clj' open. The editor displays a Clojure script for connecting to a MySQL database using the Chlorine library. The script includes comments in Portuguese and Clojure code for setting up the database connection. The Chlorine REPL on the right shows a successful connection message: 'Clojure socket REPL Connected'.

```
1  (ns clojure-mysql.core
2    (:gen-class))
3
4  (require '[clojure.java.jdbc :as jdbc])
5
6  (let [db-host "localhost"
7        db-port 3307
8        db-name "scpe"]
9
10   (def db {:classname "com.mysql.jdbc.Driver"
11             :subprotocol "mysql"
12             :subname (str db-host db-port)
13             :user "root"
14             :password "root"})
15
16   (jdbc/db-query db ["SELECT * FROM users"])
```

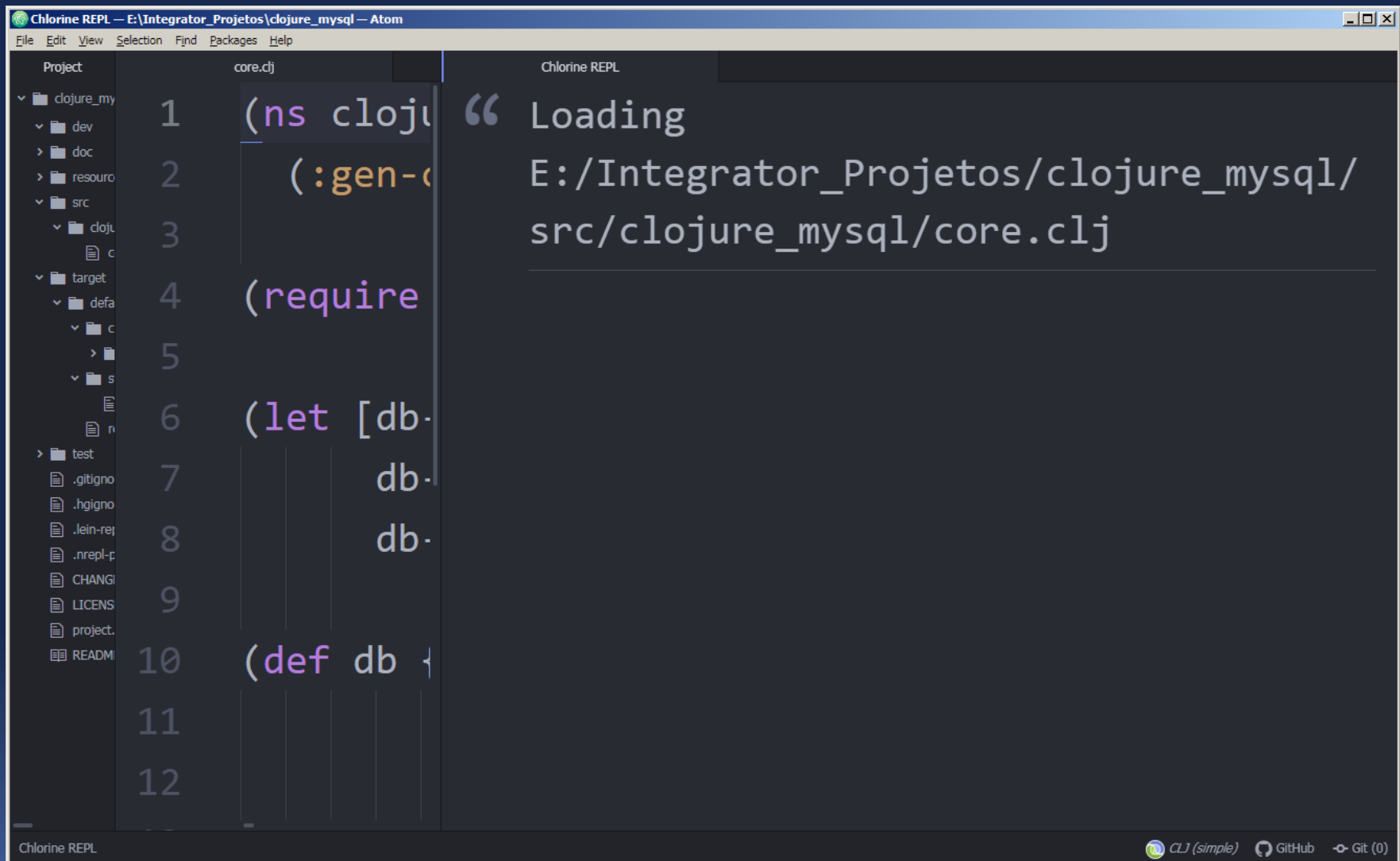
Chlorine REPL

Clojure socket REPL Connected

src\clojure_mysql\core.clj 2:16

CRLF CLJ (simple) Windows 1252 Clojure GitHub Git (0)

Carregando código no REPL (Load-file)



Chlorine REPL — E:\Integrator_Projetos\clojure_mysql — Atom

File Edit View Selection Find Packages Help

Project: core.clj

Chlorine REPL

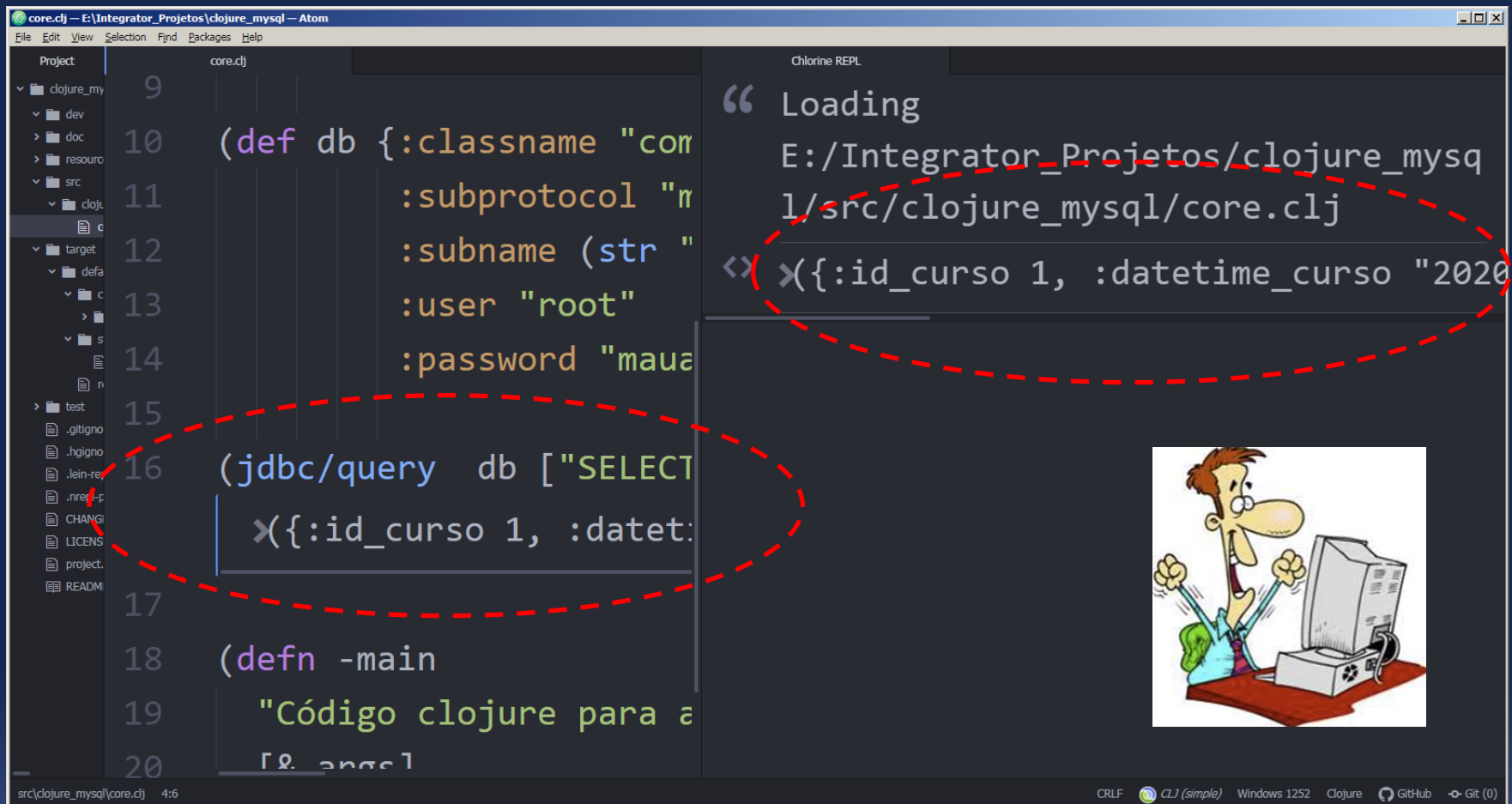
```
1 (ns clojure.mysql)
2 (:gen-class)
3
4 (require [clojure.mysql.core :as db])
5
6 (let [db-conn (db/connect)]
7   db-conn
8   db-conn)
9
10 (def db-conn (db/connect))
11
12
```

“ Loading
E:/Integrator_Projetos/clojure_mysql/
src/clojure_mysql/core.clj

Chlorine REPL

CLJ (simple) GitHub Git (0)

Executando sob REPL (Chlorine)




The screenshot shows the Atom editor interface with a project named 'core.clj' open. The left sidebar displays a file tree with folders like 'dev', 'doc', 'resources', 'src', 'target', 'test', and files like '.gitignore', '.hgignore', '.lein-repl', '.nrepl-rpc', 'CHANGELOG.md', 'LICENSE', 'project.clj', and 'README.md'. The main editor area shows the following Clojure code:

```
9  
10 (def db {:classname "com.mysql.jdbc.Driver"  
11          :subprotocol "mysql"  
12          :subname (str "localhost:3306/"  
13                   "clojure_mysql")  
14          :user "root"  
15          :password "mauagostinho@1234567890!"})  
16 (jdbc/query db ["SELECT  
17                  >{:id_curso 1, :datetime_curso "2020-01-01 00:00:00"}])  
18 (defn -main  
19   "Código clojure para a  
20   >[> args]"
```

On the right, the Chlorine REPL window shows the following output:

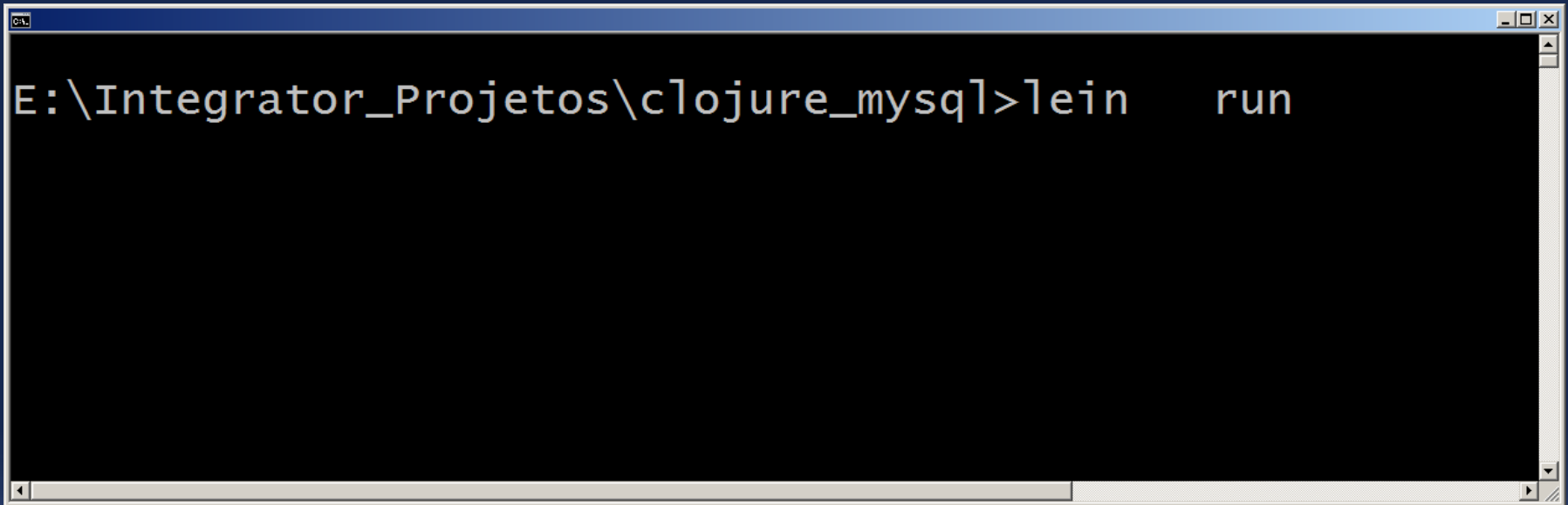
```
“ Loading  
E:/Integrator_Projetos/clojure_mysql/src/clojure_mysql/core.clj  
<> >{:id_curso 1, :datetime_curso "2020-01-01 00:00:00"}
```

Two red dashed circles highlight the code in the editor and the REPL output. The first circle highlights the JDBC query call in the code. The second circle highlights the REPL output, which shows the loading of the file and the execution of the query.



Executando – Linha de Comando REPL

No diretório do projeto > `lein run`



```
CL
```

```
E:\Integrator_Projetos\closure_mysql>lein run
```

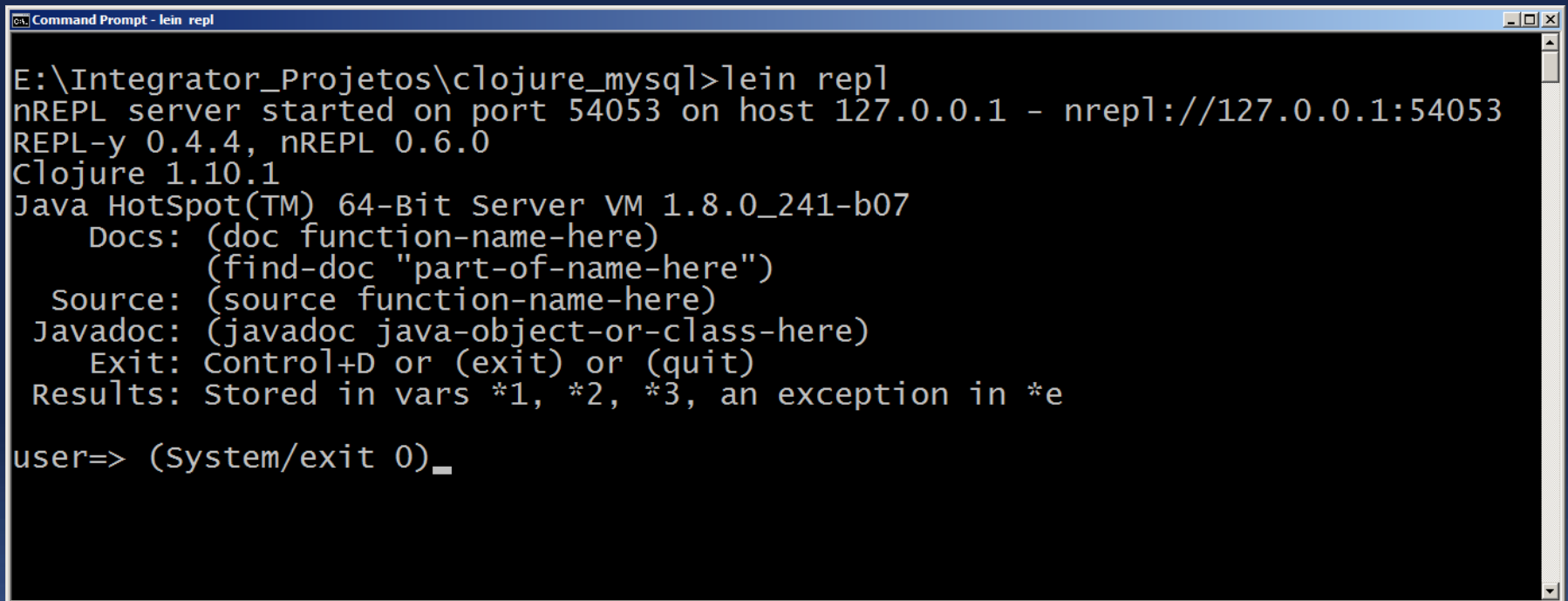
Executando – Linha de Comando REPL

- Caso, ocorra o erro "Address already in use: JVM_Bind, será necessário efetuar shutdown no repl server ativo.

```
E:\Integrator_Projetos\clojure_mysql>lein run
Exception in thread "main" java.net.BindException: Address already in use: JVM_B
ind
    at java.net.DualStackPlainSocketImpl.bind0(Native Method)
    at java.net.DualStackPlainSocketImpl.socketBind(DualStackPlainSocketImpl
.java:106)
    at java.net.AbstractPlainSocketImpl.bind(AbstractPlainSocketImpl.java:38
7)
    at java.net.PlainSocketImpl.bind(PlainSocketImpl.java:190)
    at java.net.ServerSocket.bind(ServerSocket.java:375)
    at java.net.ServerSocket.<init>(ServerSocket.java:237)
    at clojure.core.server$start_server.invokeStatic(server.clj:101)
    at clojure.core.server$start_servers.invokeStatic(server.clj:160)
    at clojure.core.server$start_servers.invoke(server.clj:157)
    at clojure.lang.Var.invoke(Var.java:384)
    at clojure.lang.RT.doInit(RT.java:493)
    at clojure.lang.RT.init(RT.java:467)
    at clojure.main.main(main.java:38)
```

Executando – Linha de Comando REPL

- Para efetuar shutdown no REPL server => `(System/exit 0)`



```
Command Prompt - lein repl

E:\Integrator_Projetos\clojure_mysql>lein repl
nREPL server started on port 54053 on host 127.0.0.1 - nrepl://127.0.0.1:54053
REPL-y 0.4.4, nREPL 0.6.0
Clojure 1.10.1
Java HotSpot(TM) 64-Bit Server VM 1.8.0_241-b07
  Docs: (doc function-name-here)
        (find-doc "part-of-name-here")
  Source: (source function-name-here)
  Javadoc: (javadoc java-object-or-class-here)
  Exit: Control+D or (exit) or (quit)
  Results: Stored in vars *1, *2, *3, an exception in *e

user=> (System/exit 0)
```

Executando – Linha de Comando REPL

```
Command Prompt
E:\Integrator_Projetos\clojure_mysql>lein run
({:id_curso 1, :datetime_curso 2020-07-20 21:51:15, :nome_curso Sistemas de Info
rma | ° | ú o, :timestamp_curso 2020-07-20 21:51:15})
E:\Integrator_Projetos\clojure_mysql>_
```

