

clojure

Gotta learn 'em all



# Por onde eu começo???



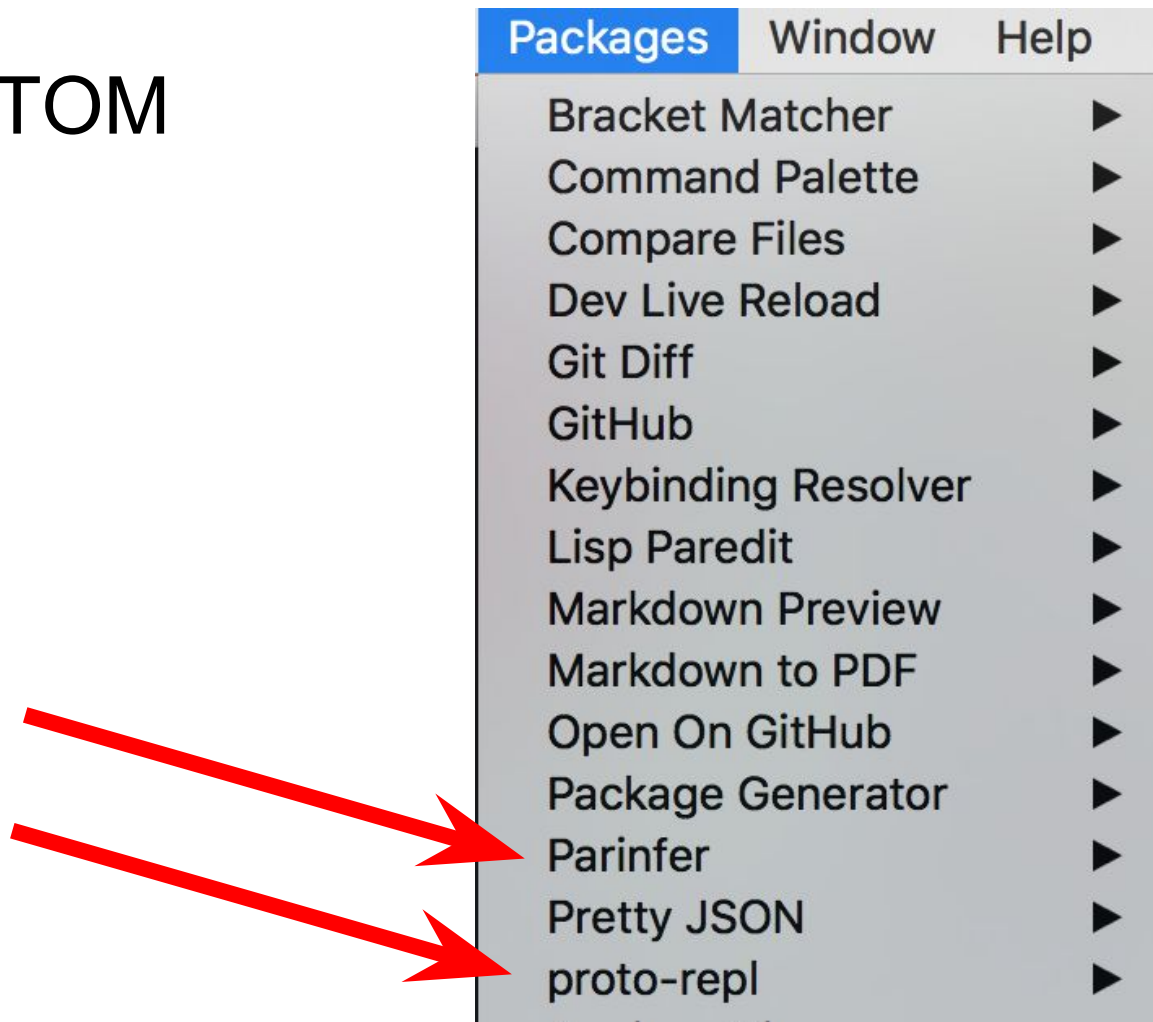
- Java, preferencialmente 8u152+
- Leinengen:  
<https://github.com/technomancy/leiningen>
- <https://leiningen.org/>
- `lein new app rest-da-jujuba`

# Instalando o Lein

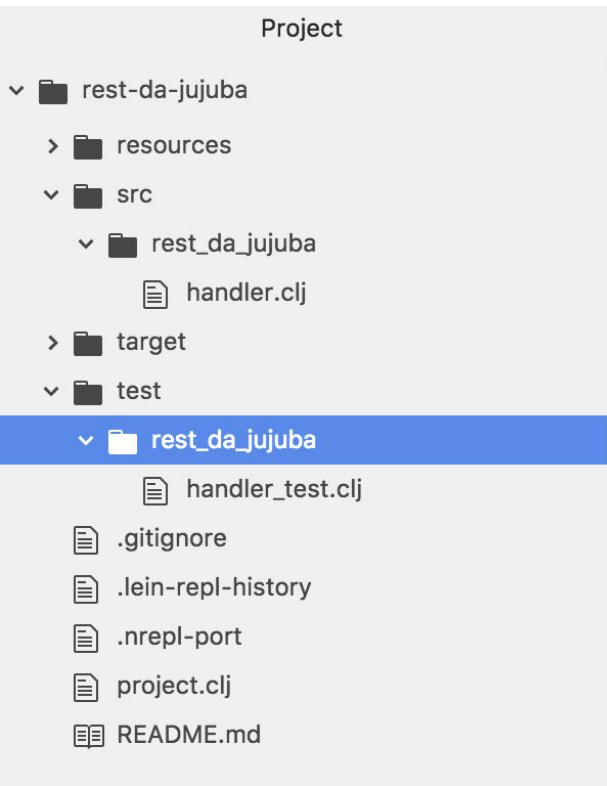
1. Download o [lein script](#) (ou no Windows [lein.bat](#))
2. Coloque em seu \$PATH de forma que seu sheel encontre (eg. ~/bin)
3. Defina para ser executado (chmod a+x ~/bin/lein)
4. Rode lein e ele fará download de um package auto-intalador.



# No ATOM



# lein new compojure rest-da-jujuba



*project.clj*

```
1 (defproject rest-da-jujuba "0.1.0-SNAPSHOT"
2   :description "FIXME: write description"
3   :url "http://example.com/FIXME"
4   :min-lein-version "2.0.0"
5   :dependencies [[org.clojure/clojure "1.8.0"]
6                 [compojure "1.5.1"]
7                 [ring/ring-defaults "0.2.1"]]
8   :plugins [[lein-ring "0.9.7"]]
9   :ring {:handler rest-da-jujuba.handler/app}
10  :profiles
11  {:dev {:dependencies [[javax.servlet/servlet-api "2.5"]
12                    [ring/ring-mock "0.3.0"]]]}}
13
```

**Lein repl** + cmd+shift+p

remote

Proto Repl: **Remote** Nrepl Connection

⌘⌘Y

^, Y

^⌘, Y

Parinfer: Indent

LF

UTF-8

Clojure

(λ) strict



# cmd+option+b

```
(deftest test-app
  (testing "main route"
    (let [response (app (mock/request :get "/"))]
      (is (= (:status response) 200))
      (is (= (:body response) "Hello World")))) [ true
```

```
(testing "not-found route"
  (let [response (app (mock/request :get "/invalid"))]
    (is (= (:status response) 404)))) [ true ]
```

```
      (let [response (app (mock/request :get "/invalid"))]
        (is (= (:status response) 404))))
```

① [rest-da-jujuba.handler-test=>](#)

✓ #'rest-da-jujuba.handler-test/test-app

```
> (testing "main route"
  (let [response (app (mock/request :get "/"))]
    (is (= (:status response) 200))
    (is (= (:body response) "Hello World"))))
```

① [rest-da-jujuba.handler-test=>](#)

✓ true

```
> (testing "not-found route"
  (let [response (app (mock/request :get "/invalid"))]
    (is (= (:status response) 404))))
```

① [rest-da-jujuba.handler-test=>](#)

✓ true

# Partiu Clojure

```
[ring.mock.request :as mock]
[rest-da-jujuba.handler :refer :all]))
```

```
(deftest test-app
```

```
> (testing "main route" ☞
```

```
> (testing "not-found route" ☞
```

```
(testing "name route"
```

```
(let [response (app (mock/request :get "/hello/julia"
```

```
(is (= (:status response) 200))
```

```
(is (= (:body response) "Hello Julia")))) [ false
```

“

```
FAIL in () (form-init7932219376232172142.clj:18)
name route
```

```
expected: 404
actual: 200
```

```
diff: - 404
      + 200
```

```
FAIL in () (form-init7932219376232172142.clj:19)
name route
```

```
expected: "Not Found"
actual: "Hello Julia"
```

```
diff: - "Not Found"
      + "Hello Julia"
```

ⓘ `rest-da-jujuba.handler-test=>`

# (GET "/hello/:nome" [nome] (str "Hello "))

```
> (testing "name route"
  (let [response (app (mock/request :get
    "/hello/julia"))]
    (is (= (:status response) 200))
    (is (= (:body response) "Hello Julia"))))
```

---

“

```
FAIL in () (form-init7932219376232172142.clj:18)
name route
```

```
expected: "Hello "
actual: "Hello Julia"
```

```
diff:  "Hello"
      + "Julia"
```

---

① `rest-da-jujuba.handler-test=>`

---

✓ false

---

>