

# Pasos para crear un proyecto en Clojure usando Leiningen (en Windows)

*Prof. Dr. Diego P. Corsi*

Más detalles (sobre todo, cómo hacerlo en otros sistemas operativos):

Instalación:

<https://github.com/technomancy/leiningen/blob/stable/README.md>

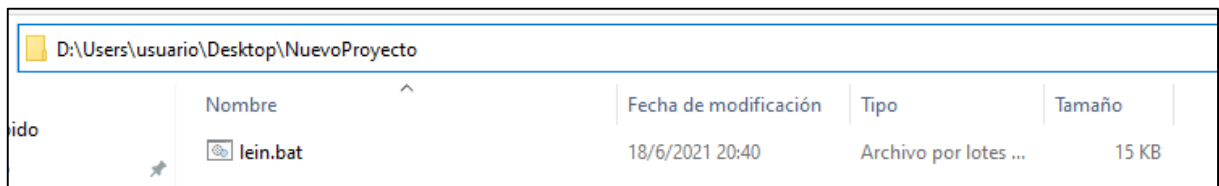
Creación de proyectos:

<https://github.com/technomancy/leiningen/blob/stable/doc/TUTORIAL.md>

1. Descargar **lein.bat** desde <https://leiningen.org>. El enlace directo es el siguiente:

<https://raw.githubusercontent.com/technomancy/leiningen/stable/bin/lein.bat>

2. Crear una carpeta (por ejemplo, **NuevoProyecto**) y guardar en ella el archivo anterior.

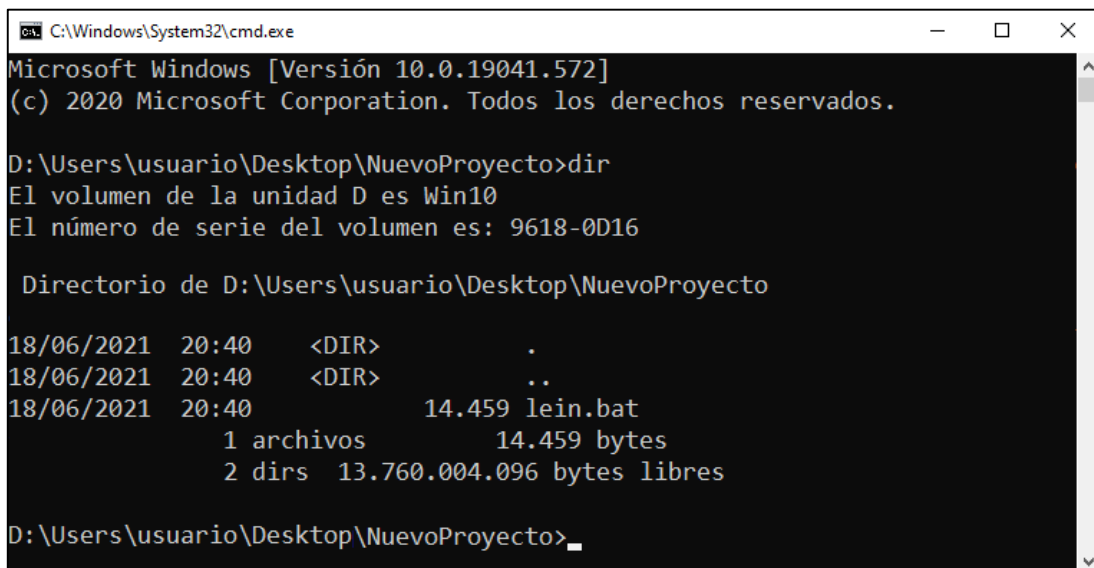


	Nombre	Fecha de modificación	Tipo	Tamaño
ido	lein.bat	18/6/2021 20:40	Archivo por lotes ...	15 KB

3. Abrir una consola (*Símbolo del Sistema*) en la carpeta anterior. Una manera de hacerlo es escribiendo

**cmd** en la barra de direcciones (para el ejemplo anterior, hacerlo en reemplazo de

**D:\Users\usuario\Desktop\NuevoProyecto**).



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19041.572]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

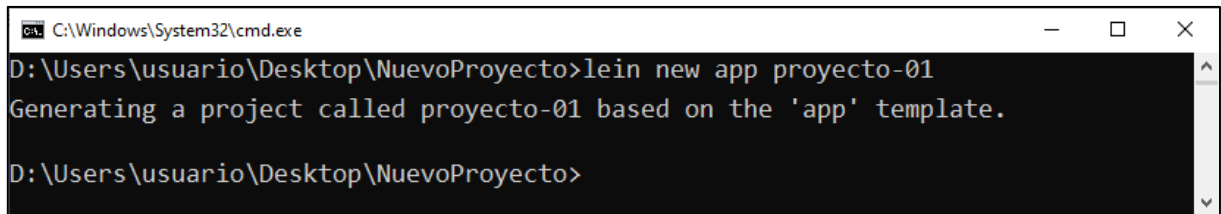
D:\Users\usuario\Desktop\NuevoProyecto>dir
El volumen de la unidad D es Win10
El número de serie del volumen es: 9618-0D16

Directorio de D:\Users\usuario\Desktop\NuevoProyecto

18/06/2021  20:40    <DIR>          .
18/06/2021  20:40    <DIR>          ..
18/06/2021  20:40                14.459 lein.bat
               1 archivos             14.459 bytes
               2 dirs 13.760.004.096 bytes libres

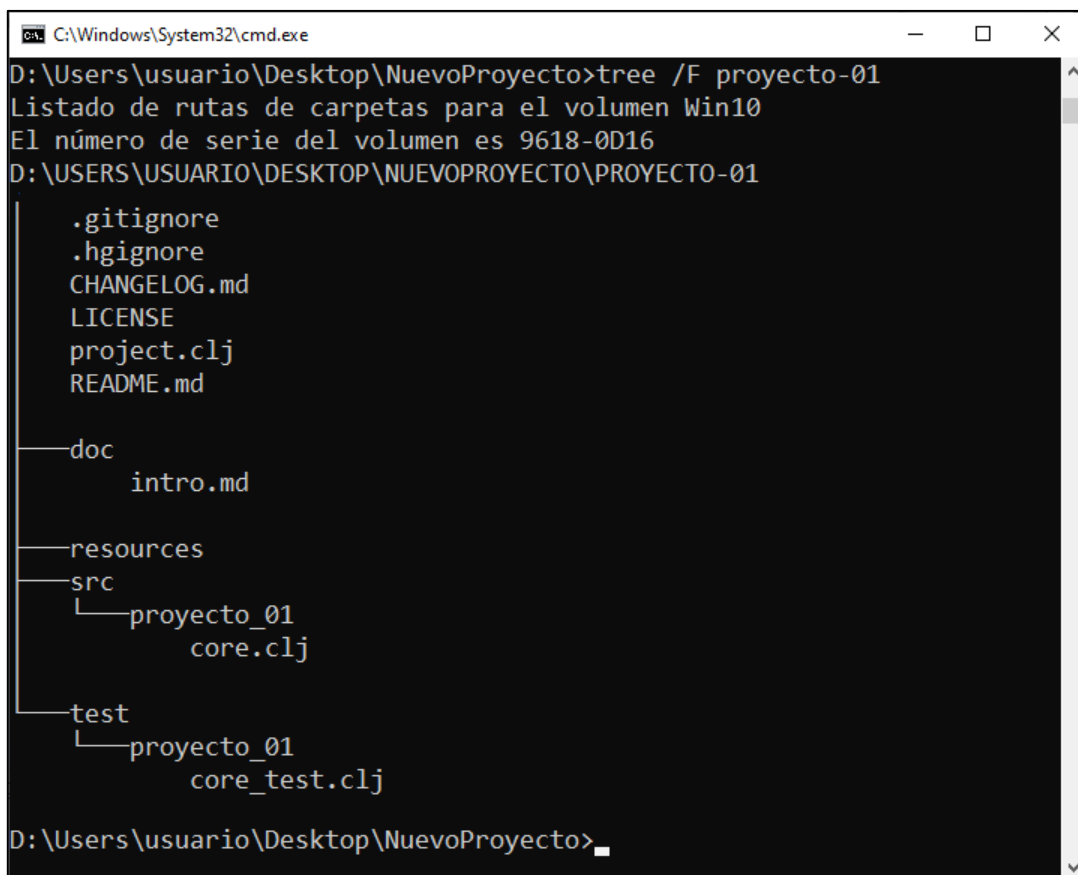
D:\Users\usuario\Desktop\NuevoProyecto>_
```

4. Ejecutar **lein self-install**.
5. Ejecutar **lein new app** seguido del nombre del nuevo proyecto, por ejemplo, **proyecto-01**:



```
C:\Windows\System32\cmd.exe
D:\Users\usuario\Desktop\NuevoProyecto>lein new app proyecto-01
Generating a project called proyecto-01 based on the 'app' template.
D:\Users\usuario\Desktop\NuevoProyecto>
```

6. Ejecutar **tree /F** seguido del nombre del nuevo proyecto, por ejemplo, **proyecto-01**:



```
C:\Windows\System32\cmd.exe
D:\Users\usuario\Desktop\NuevoProyecto>tree /F proyecto-01
Listado de rutas de carpetas para el volumen Win10
El número de serie del volumen es 9618-0D16
D:\USERS\USUARIO\DESKTOP\NUEVOPROYECTO\PROYECTO-01
.
├── .gitignore
├── .hgignore
├── CHANGELOG.md
├── LICENSE
├── project.clj
├── README.md
├── doc
│   └── intro.md
├── resources
├── src
│   └── proyecto_01
│       └── core.clj
└── test
    └── proyecto_01
        └── core_test.clj
D:\Users\usuario\Desktop\NuevoProyecto>
```

7. Editar **project.clj**, agregando, por ejemplo, la descripción del proyecto en el campo **:description** o las dependencias necesarias en el campo correspondiente.
8. Editar **core.clj**, colocando allí las declaraciones y definiciones de las funciones que forman parte del proyecto, la primera de las cuales deberá ser evaluada en **-main**.

Por ejemplo:

core.clj

```
(ns proyecto-01.core
  (:gen-class))

(declare ejemplo)
(declare es-el-doble?)
(declare spy)

(defn -main
  "Ejemplo de Proyecto en Clojure"
  [& args]
  (ejemplo))

(defn ejemplo []
  (let [mensaje-1 (do (print "Ingrese un numero: ") (flush)),
        n1 (read),
        mensaje-2 (do (print "Ingrese el doble del numero anterior: ") (flush)),
        n2 (read),
        salida (println (str "Ud." (if (es-el-doble? n1 n2) " " " no ") "sabe calcular!"))]
    (do (print "Presione Enter...") (flush) (read-line) (read-line) 'Chau!)))

(defn es-el-doble? [a b]
  (= (* 2 a) b))

(defn spy
  ([x] (do (prn x) x))
  ([msg x] (do (print msg) (print ": ") (prn x) x)))
```

9. Editar **core\_test.clj**, colocando allí las pruebas de las funciones (este paso es fundamental, sobre todo, cuando se lleva a cabo *TDD* o *Test-Driven Development*).

Por ejemplo:

core\_test.clj

```
(ns proyecto-01.core-test
  (:require [clojure.test :refer :all]
            [proyecto-01.core :refer :all]))

(deftest es-el-doble?-test
  (testing "Prueba de la funcion: es-el-doble?"
    (is (= true (es-el-doble? 4 8)))
    (is (= false (es-el-doble? 4 7)))
  )
)
```

10. Posicionarse en el directorio del proyecto (ejecutando **cd** seguido, por ejemplo, de **proyecto-01**) y copiar allí el archivo **lein.bat** (ejecutando **copy ..\lein.bat**).

A partir de aquí pueden ejecutarse diferentes tareas, siendo las más comunes:

- **lein help**: mostrar las distintas tareas posibles;
- **lein test**: correr las pruebas de las funciones;
- **lein run**: evaluar la función **-main** del proyecto;
- **lein uberjar**: generar (en la carpeta **target**) un archivo ejecutable *jar* con todas las dependencias incluidas;
- **lein repl**: iniciar un REPL (*read-eval-print-loop*) para trabajar interactivamente en el proyecto.

```
C:\Windows\System32\cmd.exe
D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>lein test

lein test proyecto-01.core-test

Ran 1 tests containing 2 assertions.
0 failures, 0 errors.

D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>lein run
Ingrese un numero: 4
Ingrese el doble del numero anterior: 8
Ud. sabe calcular!
Presione Enter...

D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>lein run
Ingrese un numero: 4
Ingrese el doble del numero anterior: 7
Ud. no sabe calcular!
Presione Enter...

D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>lein uberjar
Compiling proyecto-01.core
Created D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01\target\uberjar\proyecto-01-0.1.0-SNAPSHOT.jar
Created D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01\target\uberjar\proyecto-01-0.1.0-SNAPSHOT-standalone.jar

D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>java -jar target\uberjar\proyecto-01-0.1.0-SNAPSHOT-standalone.jar
Ingrese un numero: 4
Ingrese el doble del numero anterior: 8
Ud. sabe calcular!
Presione Enter...

D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>
```

```
C:\Windows\System32\cmd.exe - lein repl
D:\Users\usuario\Desktop\NuevoProyecto\proyecto-01>lein repl
nREPL server started on port 59209 on host 127.0.0.1 - nrepl://127.0.0.1:59209
REPL-y 0.4.4, nREPL 0.8.3
Clojure 1.10.1
OpenJDK 64-Bit Server VM 13.0.2+8
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

proyecto-01.core=>
```