

# README

October 31, 2016

## 1 Introduccion

El presente proyecto contiene los ejercicios correspondientes a la primera práctica de la asignatura “Computación Evolutiva” del máster “Máster Universitario en I.A. Avanzada: fundamentos, métodos y aplicaciones”

El lenguaje de programación en el que se realizan estos ejercicios es **Clojure**, un dialecto de Lisp que puede ejecutarse sobre la máquina virtual java o .Net. Existen también intérpretes de Clojure para javascript.

## 2 Estructura de directorios de este proyecto

- **project.clj**: archivo típico de configuración de un proyecto Clojure con lein como gestor de dependencias
- **README.pdf**: archivo con descripción del proyecto e instrucciones para su ejecución
- **resources**: carpeta con recursos estáticos
  - **backend.php**: implementación del Web Service que nos permite visualizar en la nube la ejecución de un experimento
  - **dashboard.html**: implementación del frontal que nos permite visualizar en la nube la ejecución de un experimento
  - **README.md**: código fuente en markdown del archivo README.pdf
  - **rosettacode.scm**: archivo con los datos de prueba del problema de la mochila publicado en [https://www.rosettacode.org/wiki/Knapsack\\_problem/0-1](https://www.rosettacode.org/wiki/Knapsack_problem/0-1)
- **src/ce**: carpeta con el código fuente de la implementaciones del algoritmo de aprendizaje
  - **p1.clj**: implementación del algoritmo de aprendizaje pedido en la primera práctica
  - **utils.clj**: implementación de las funcionalidades para reporte en la nube de la evolución del aprendizaje y resultado de la ejecución del experimento
- **test/ce**: carpeta con el código fuente de las distintas ejecuciones de los distintos experimentos pedidos en las prácticas
  - **p1\_test.clj**: código fuente de los distintos experimentos pedidos en la práctica 1 - se acompañan sus configuraciones

### 3 Instrucciones para ejecutar el código de los ejemplos

Hay bastantes formas de ejecutar código Clojure.

Seguramente, lo más sencillo es utilizar un IDE que además de ejecutar el código nos permite explorar el código de una forma muy hergonómica.

Dependiendo del entorno de trabajo de la persona que explora este proyecto y de sus preferencias, se pueden elegir entre las siguientes opciones:

1. **Eclipse con CounterClockWise** <http://doc.ccw-ide.org> Escasamente desarrollado
2. **Emacs**, en mi opinión el mejor plugin es emacs live <http://overtone.github.io/emacs-live>  
No es muy recomendable para alguien que no tenga manejo con emacs porque este IDE realiza cierto abuso de combinaciones de teclas que pueden resultar difíciles al principio
3. **IntelliJIDEA** con el plugin cursive <https://cursive-ide.com/userguide> Muy recomendable, muy cómodo y robusto
4. **LightTable** <http://lighttable.com/> **Creo que para iniciarse en Clojure, este es el IDE más sencillo y directo.** No conlleva complicaciones de ningún tipo: sólo descargar, abrir proyecto y ejecutar. **En el siguiente punto detallo mejor como usar LightTable para ejecutar el código contenido en este proyecto**
5. ... algunos otros ...

### 4 Ejecutar el código de los ejemplos con LightTable

1. Descargue LightTable desde <http://lighttable.com/>
2. Descomprima el archivo descargado en un lugar de su preferencia y arranque el IDE - en entorno Windows esto se hace simplemente clickando dos veces en LightTable.exe. Lo mismo ocurre en entornos linux.
3. Una vez arrancado el IDE, haga:
  - File > Open Folder
  - Y seleccione la carpeta en donde esté este proyecto
4. En el navegador de la izquierda, clicar en la estructura de directorios hasta que se abra el archivo **src/ce/p1.clj** en el editor LightTable
5. Para ejecutar código Clojure de modo interactivo, primero necesitamos instanciar un REPL. Para ello, simplemente, en el archivo que tiene abierto, posicione el cursor justos después de:

```
(ns ce.p1
  (:require [ce.utils :refer :all]))
```

y pulse **Ctrl+Enter**. Después de esperar unos segundos, una nueva instancia del REPL clojure está abierta y a nuestra disposición.

6. Podemos seguir ejecutando bloques de código posicionando el cursor justo detrás del bloque y pulsando **Ctrl+Enter**.

7. O mejor aún, podemos evaluar el fichero de una vez en el REPL seleccionándolo por completo (**Ctrl+A**) y enviándolo al REPL (**Ctrl+Enter**) - LE RECOMIENDO ESTA OPCIÓN
8. Ante cualquier duda o problema no dude en contactarme e intentaré hacer todo lo que pueda para resolver cualquier incidencia con la configuración de un entorno de ejecución clojure.