

MANUAL DE USUARIO

PDC – Problema del Caballo

Jose Antonio Ortega González / Andrey Sibaja Garro / Christian Alpizar Monge

El siguiente programa busca solucionar recorrido que hace la pieza del caballo en el tablero de ajedrez para que pueda pasar por todas las casillas y únicamente se ubique una vez en cada una. Realizando un total de $n^2 - 1$ movimientos, siendo el tablero de tamaño $n \times n$. Cabe destacar que el caballo puede encontrar soluciones con tableros de $n \geq 5$.

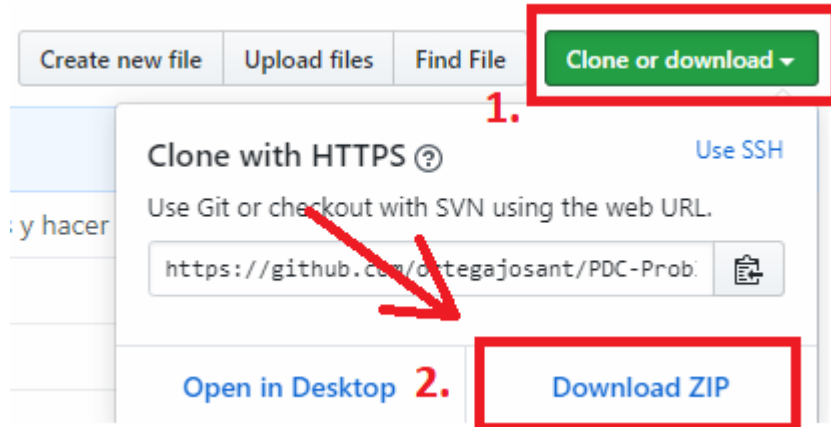
- 1) Inicialmente para poder ejecutar el programa se debe instalar el entorno Dr. Racket, el mismo se puede obtener en el siguiente enlace:

<https://download.racket-lang.org/>

- 2) Una vez instalado Dr. Racket, es necesario descargar los contenidos del programa, en el siguiente repositorio puede obtenerlos:

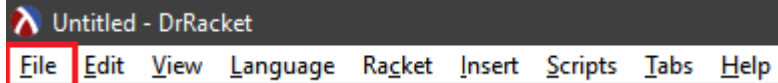
<https://github.com/ortegajosant/PDC-Problema-del-Caballo>

- a. Se puede clonar o simplemente descargar un archivo .zip que contiene todo lo necesario para la ejecución haciendo click en la sección de **Clone or download** -> **Download ZIP**.

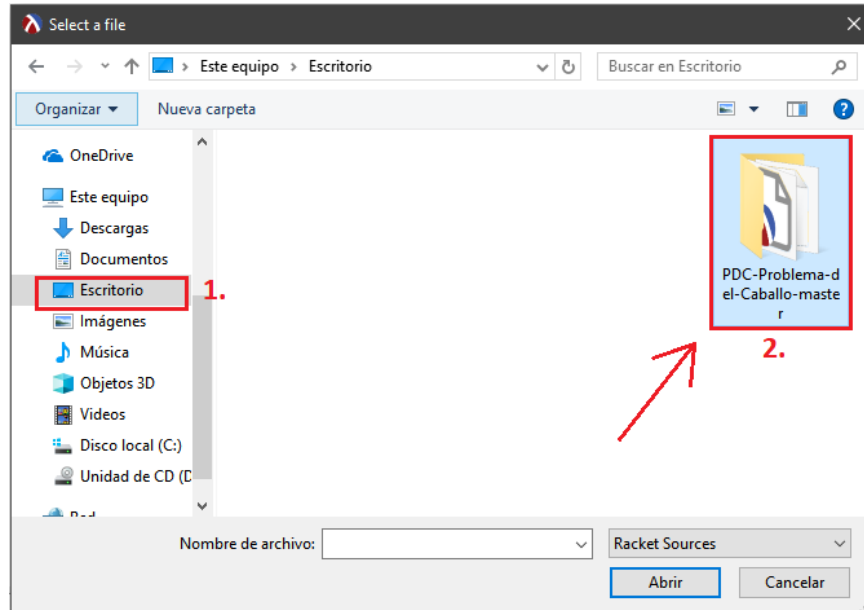


- b. Ahora se debe descomprimir el archivo en la ubicación de preferencia personal, en este caso se descomprimirá en el escritorio.

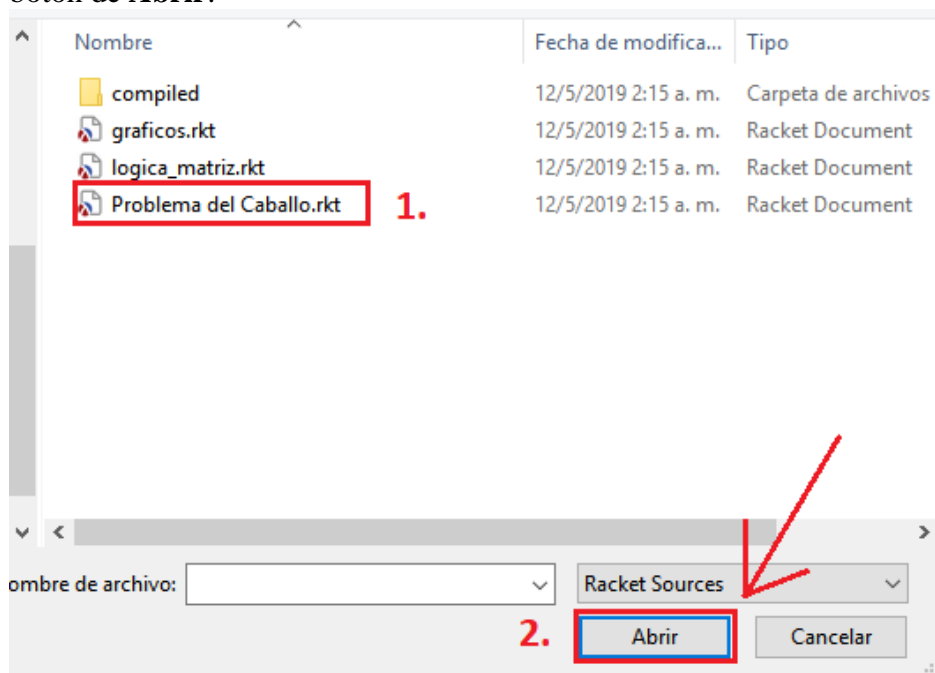
- 3) Debemos abrir el programa Dr. Racket y clickear la pestaña **File** -> **Open** o bien, presionar **Ctrl + O**.




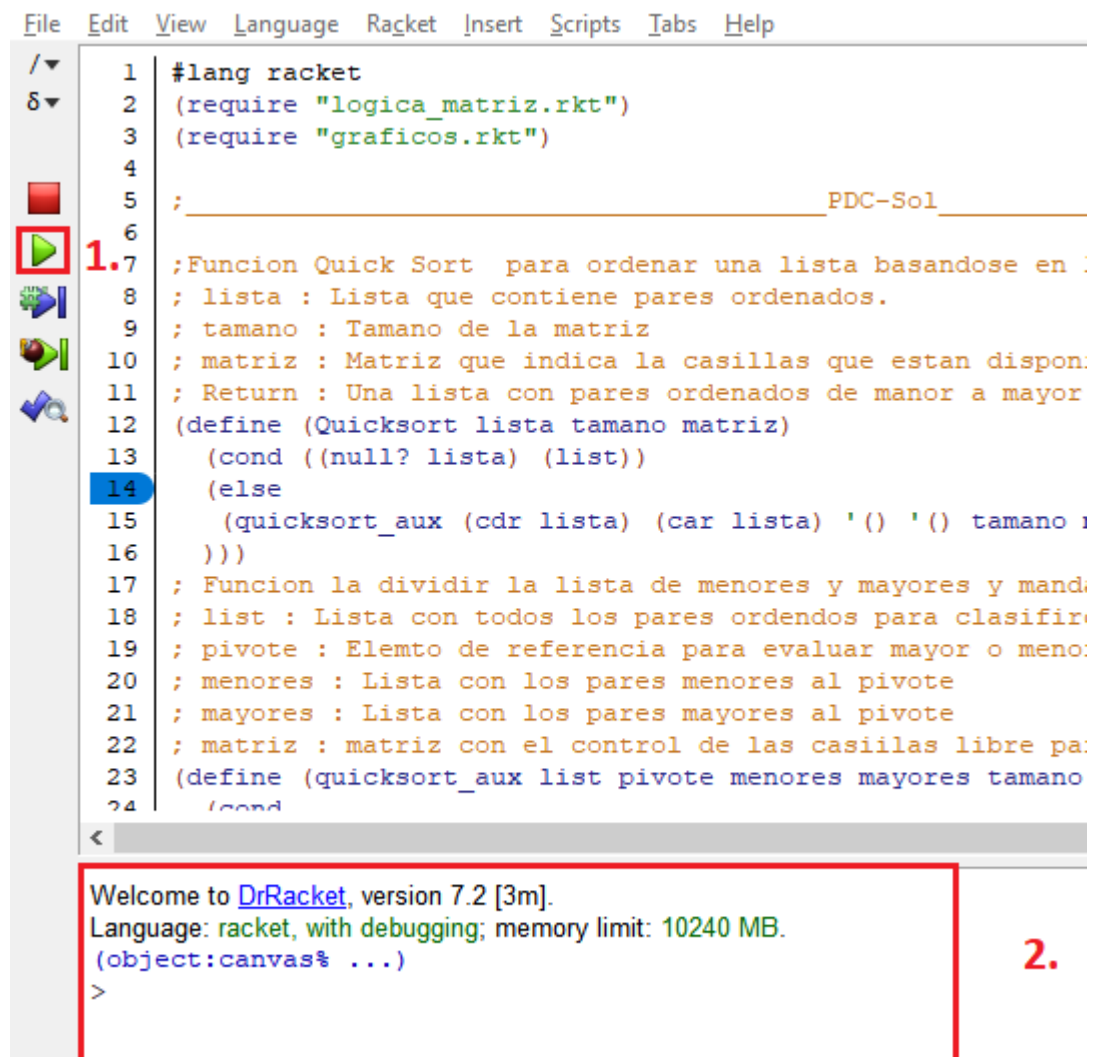
- a) Ahora se abrirá una ventana para seleccionar el archivo que se desea abrir, en este caso nos dirigimos a **Escritorio** y abrimos la carpeta **PDC-Problema-del-Caballo-master**:



- b) Accedemos a la carpeta de **PDC-Problema-del-Caballo-master** haciendo doble click y nos mostrará la siguiente ventana, donde seleccionamos el archivo **Problema del Caballo.rkt** y lo abrimos haciendo doble click o presionando el botón de **Abrir**:



- 4) Ahora basta con hacer click en el ícono **Run**  de Dr. Racket para iniciar el programa. De esta manera se mostrará la siguiente ventana, donde se escribirán los comandos en la terminal de Dr. Racket:



The screenshot shows the Dr. Racket IDE interface. The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The editor window contains a Racket script with the following code:

```

1 #lang racket
2 (require "logica_matriz.rkt")
3 (require "graficos.rkt")
4
5 ; _____ PDC-Sol _____
6
7 ;Funcion Quick Sort para ordenar una lista basandose en :
8 ; lista : Lista que contiene pares ordenados.
9 ; tamano : Tamano de la matriz
10 ; matriz : Matriz que indica la casillas que estan dispon
11 ; Return : Una lista con pares ordenados de manor a mayor
12 (define (Quicksort lista tamano matriz)
13   (cond ((null? lista) (list))
14         (else
15          (quicksort_aux (cdr lista) (car lista) '() '() tamano :
16                        )))
17 ; Funcion la dividir la lista de menores y mayores y manda
18 ; list : Lista con todos los pares ordenados para clasifi
19 ; pivote : Elemto de referencia para evaluar mayor o meno
20 ; menores : Lista con los pares menores al pivote
21 ; mayores : Lista con los pares mayores al pivote
22 ; matriz : matriz con el control de las casiilas libre pa
23 (define (quicksort_aux list pivote menores mayores tamano
24   (cond

```

The terminal window at the bottom shows the following output:

```

Welcome to DrRacket, version 7.2 [3m].
Language: racket, with debugging; memory limit: 10240 MB.
(object:canvas% ...)
>

```

A red box highlights the terminal output, and a red number '2.' is placed to the right of the terminal window.

- a) En la terminal de Dr. Racket podemos escribir las siguientes líneas para obtener diferentes resultados:

- a. (PDC-Sol n '(x y)):

Esta función devuelve una solución (lista con pares ordenados) para el recorrido del caballo en un tablero de nxn indicado por el parámetro **n**, iniciando desde una posición en el tablero (**x y**) donde **x** es la fila y **y** la columna. Ejemplo:

```

> (PDC-Sol 5 '(0 0))
'((0 0) (1 2) (0 4) (2 3) (4 4)
  (3 2) (4 0) (2 1) (4 2) (3 4)
  (1 3) (0 1) (2 0) (4 1) (3 3)
  (1 4) (0 2) (1 0) (3 1) (4 3)
  (2 4) (0 3) (1 1) (3 0) (2 2))

```

- Con PDC-Todas, obtenemos todas las posibles soluciones para el recorrido del en un tablero de $n \times n$ e iniciando desde una posición en específico, los parámetros son los mismos que en *PDC-Sol*. Devuelve una matriz en donde cada fila es una solución (lista con pares ordenados). Ejemplo:

c. (PDC-Test n solución):

```
> (PDC-Test 5 (PDC-Sol 5 '(0 0)))
```

- Con esta función se muestra mediante una interfaz gráfica el recorrido del caballo en el tablero ingresado, recibe los mismos parámetros de *PDC-Test*. Ejemplo: