



Manual de Usuario

Contenidos

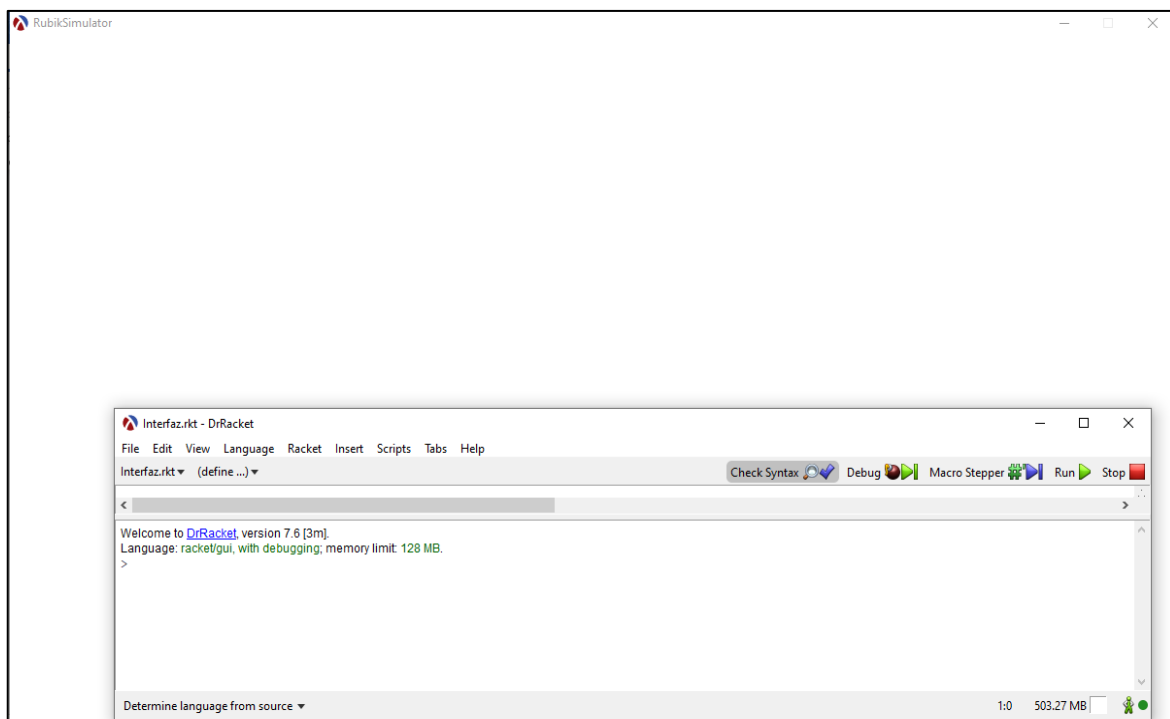
I.	Objetivos del sistema	3
II.	Inicio del programa.....	3
III.	Representación de los cubos	4
IV.	Representación de los movimientos del cubo	6
V.	Comandos necesarios	7
VI.	Movimiento y animación del cubo	8
VII.	Errores comunes	9

I. Objetivos del sistema

RubikSimulator permite la creación y la interacción con un cubo de Rubik virtual, simulando el comportamiento y movimiento de cada fila y columna por medio de comandos introducidos por consola.

II. Inicio del programa

Una vez se haya iniciado el programa se mostrará una ventana vacía y una caja de texto. Como se presenta a continuación:



La caja de texto de la parte inferior de la pantalla permitirá interactuar con el programa el cual reflejará los resultados en la ventana.

III. Representación de los cubos

Para lograr interactuar con un cubo se debe primero saber cómo representarlo e interpretarlo. RubikSimulator admite un rango de dimensiones del cubo que van desde un 3x3 hasta un 15x15.

Un cubo se debe representar como una lista de listas, cada una de estas listas representa una cara del cubo y cada lista dentro de esta representa cada fila. Cada letra representa una casilla de la fila y el color de la misma.

Siguiendo la siguiente estructura:

a – casilla color azul

r – casilla color rojo

y – casilla color amarillo

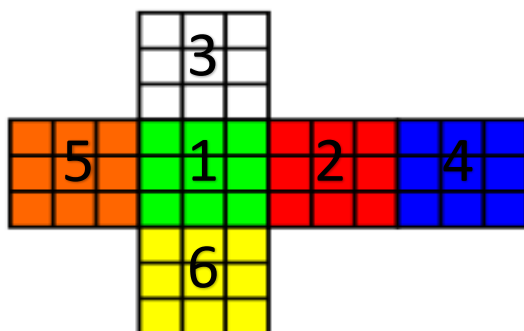
n – casilla color naranja

v – casilla color verde

b – casilla color blanco

Se debe tomar en cuenta que, si se desea crear un cubo $n \times n$, la lista que representa cada cara debe tener n cantidad de filas y las filas n cantidad de casillas.

El orden de las caras se detalla en la siguiente imagen, la primera cara seria la primera lista de la estructura, y así con el resto de caras y listas.



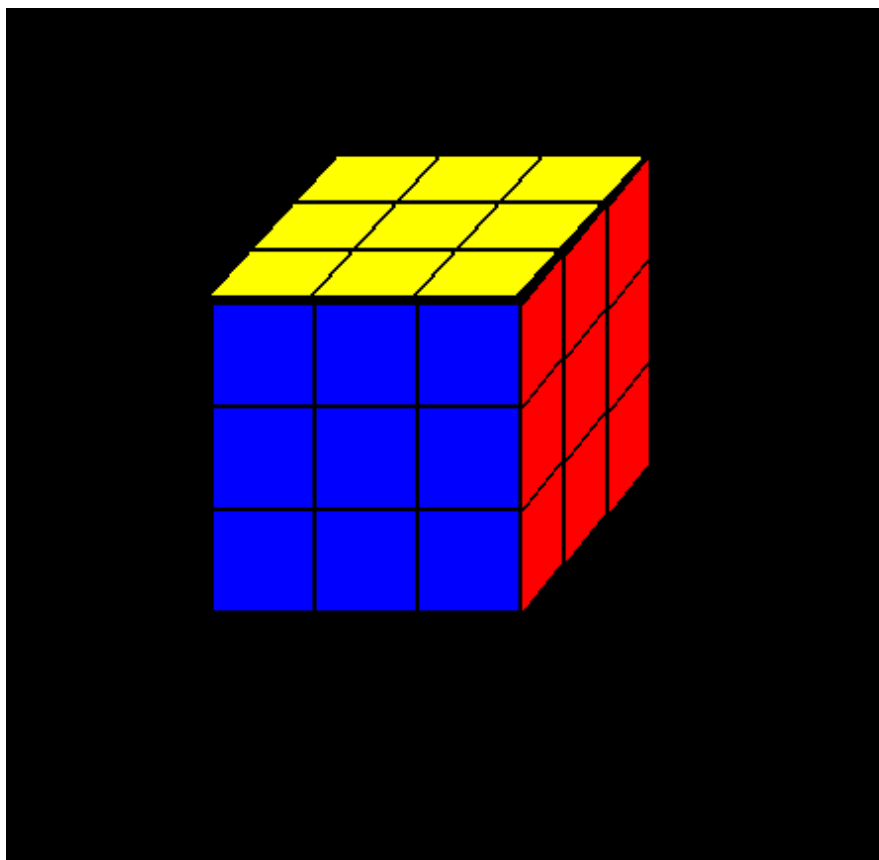
En el siguiente ejemplo se muestra la representación de un cubo 3x3:

Cara 1 **Cara 2, Fila 2**

```
'(((a a a) (a a a) (a a a)) ((r r r) (r r r) (r r r)) ((y y y) (y y y) (y y y))  
((n n n) (n n n) (n n n)) ((v v v) (v v v) (v v v)) ((b b b) (b b b) (b b b))))
```

Cara 5, Fila 1, Casilla 2 verde

Este cubo representado en la ventada del programa se vería así:



IV. Representación de los movimientos del cubo

Para lograr mover las columnas o filas del cubo se debe seguir una estructura similar a la creación del cubo, por medio de listas.

Por ejemplo, la siguiente estructura:

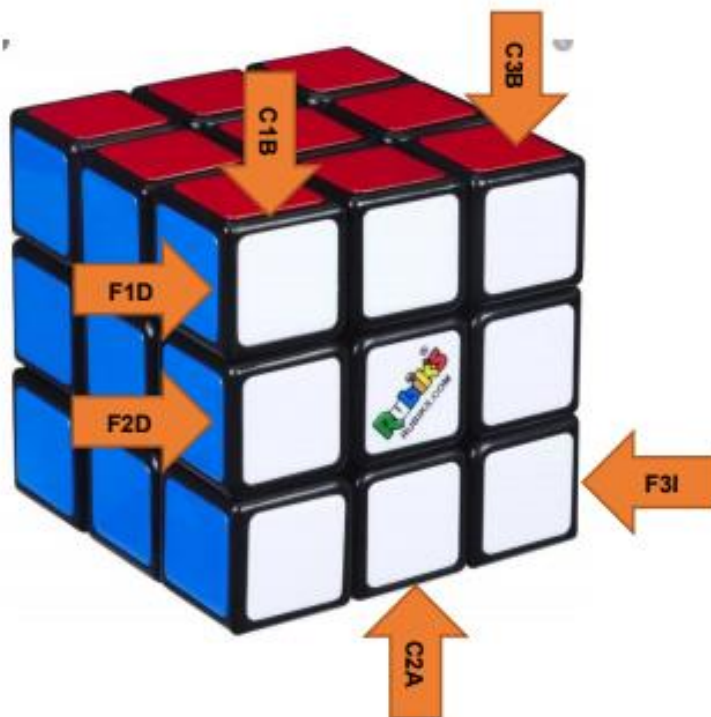
```
' ("F2D" "C3A" "F1I" "C2B")
```

Cada cadena de texto de la lista representa un movimiento del cubo, se debe separar con un espacio.

La primera letra de la cadena representará realizar un movimiento en una fila o columna, denotando a **F** como fila y **C** como columna.

La segunda letra representara el numero de fila o columna a mover y la última letra representa la dirección del movimiento, **D** e **I** para mover una fila a la derecha e izquierda respectivamente, **A** y **B** para mover una columna arriba o abajo.

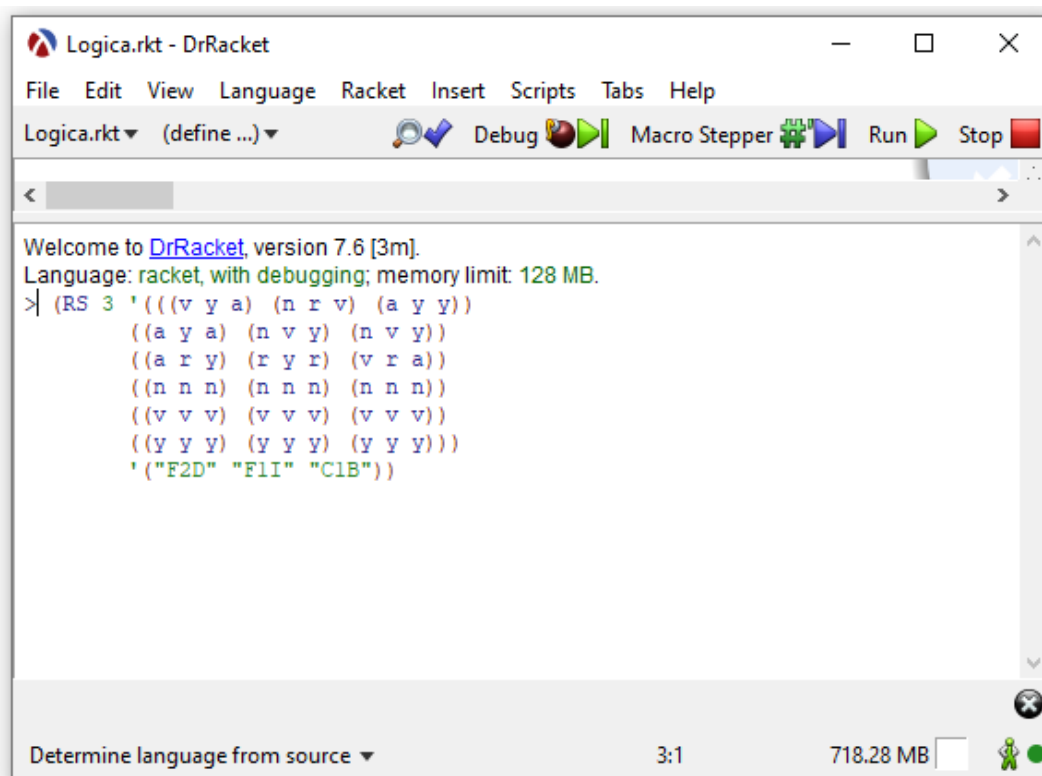
El resumen se aprecia en la siguiente imagen:



V. Comandos Necesarios

Para lograr crear e interactuar con los cubos creados se debe utilizar un único comando, este engloba tanto el cubo que se va a crear, su dimensión y los movimientos que se realizaran en el mismo.

Este comando y sus componentes se representan a continuación:



El componente RS indica al programa que se deben tomar las siguientes partes para crear e interactuar con el cubo, el numero siguiente representa la dimensión del mismo y las siguientes listas son el cubo a crear y el movimiento a realizar en el mismo.

Este comando debe escribirse en la caja de texto descrita en el apartado I. Una vez escrito el mismo se presiona el botón **Enter** y la ejecución del programa se iniciará.

Si se desea crear otro cubo o realizar otros movimientos se puede volver a enviar el comando editado las veces que sean.

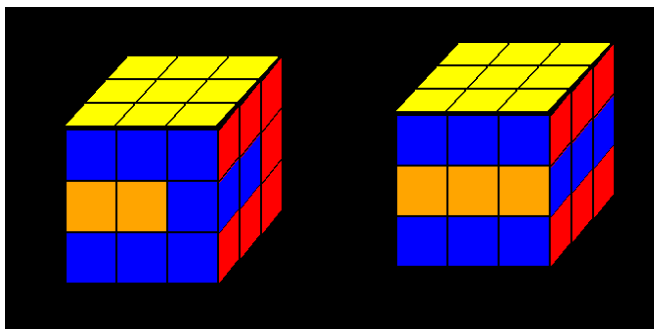
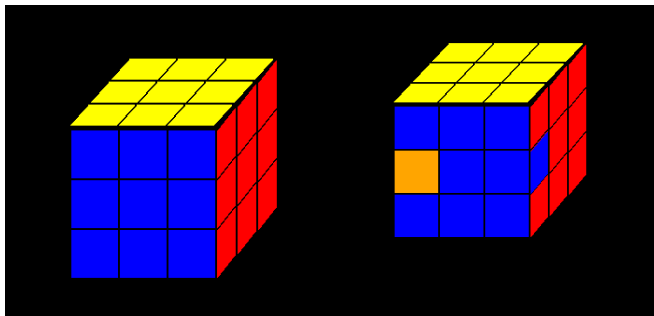
VI. Animación del cubo y respuesta en ventana

Una vez enviado el comando, en la ventana se creará el cubo y se irán realizando cada uno de los movimientos enviados uno por uno.

Se mostrará a continuación la animación con un cubo y un movimiento de ejemplo:

```
> (RS 3 '(((a a a) (a a a) (a a a))  
      ((r r r) (r r r) (r r r))  
      ((y y y) (y y y) (y y y))  
      ((n n n) (n n n) (n n n))  
      ((v v v) (v v v) (v v v))  
      ((b b b) (b b b) (b b b))) '("F2D"))
```

La animación se verá así:



VII. Errores Comunes

Existen ciertos errores comunes principalmente por la escritura del comando y de las estructuras de datos. El programa ante uno de estos informara cual es el origen del problema.

Por ejemplo, si se da una dimensión del cubo que no concuerda con la estructura del cubo, se informara de la siguiente forma:

```
> (RS 4 '(((a a a) (a a a) (a a a))
      ((r r r) (r r r) (r r r))
      ((y y y) (y y y) (y y y))
      ((n n n) (n n n) (n n n))
      ((v v v) (v v v) (v v v))
      ((b b b) (b b b) (b b b))) '("F2D"))
"numero de filas incorrecta o dimension del cubo incorrecta"
```

Si se envía un dato que no corresponde a un numero en el apartado de la dimensión se dará el siguiente error:

```
> (RS 'd '(((a a a) (a a a) (a a a))
      ((r r r) (r r r) (r r r))
      ((y y y) (y y y) (y y y))
      ((n n n) (n n n) (n n n))
      ((v v v) (v v v) (v v v))
      ((b b b) (b b b) (b b b))) '("F2D"))
"x debe ser un numero"
```


Si en la lista de los movimientos hay alguno incorrecto se informara así:

```
> (RS 3 '(((a a a) (a a a) (a a a))
      ((r r r) (r r r) (r r r))
      ((y y y) (y y y) (y y y))
      ((n n n) (n n n) (n n n))
      ((v v v) (v v v) (v v v))
      ((b b b) (b b b) (b b b))) '("R2D"))
"coloque F o C para el tipo de movimiento"
>
```

Existen algunos otros errores que se podrían dar, que no generan un mensaje como los anteriores. Por ejemplo:

Agregar ‘

```
> (RS 3 '(((a a a) (a a a) (a a a))
      ((r r r) (r r r) (r r r))
      ((y y y) (y y y) (y y y))
      ((n n n) (n n n) (n n n))
      ((v v v) (v v v) (v v v))
      ((b b b) (b b b) (b b b))) ("R2D"))
```

 application: not a procedure;
expected a procedure that can be applied to arguments
given: "R2D"
arguments...: [none]

El problema radica en la forma que se esta enviando la lista de movimientos, cada una de estas listas se representan con un apostrofe “ ’ ” al inicio de la misma.