



**INSTITUTO POLITÉCNICO NACIONAL**

---

**ESCUELA SUPERIOR DE CÓMPUTO**

**ESTRUCTURAS DE DATOS**

**PRÁCTICA 3**

**PLANIFICACIÓN POR TURNO ROTARIO  
(ROUND ROBIN)**

**HERNÁNDEZ CASTELLANOS CÉSAR URIEL**

**MARTÍNEZ ISLAS MAURICIO JOEL**

**1CV7**

**05/03/2017**

**LUNA BENOSO BENJAMIN**



## INDICE

Resumen.....	3
Introducción.....	3
Experimentación y resultados.....	6
Planificación por turnos rotarios Round Robin. 1 .....	7
Planificación por turnos rotarios Round Robin 1 (Pseudocódigo) .....	26
Planificación por turnos rotarios Round Robin. 2 .....	31
Planificación por turnos rotarios Round Robin. 2 (Pseudocódigo.) .....	32
Referencias .....	34

## Resumen.

En el presente reporte se muestra la documentación de la práctica, cuya función principal es la siguiente:

Emular la planificación por turnos rotatorio (Round Robin), la aplicación tendrá las siguientes funcionalidades.

- Cantidad de procesos
- Mostrar tabla de procesos.
- Emular paso a paso.
- Emular completo.

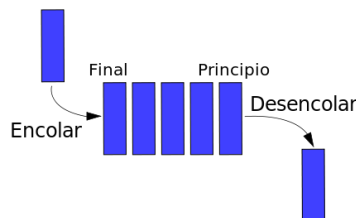
## Introducción.

Una computadora es una máquina con la capacidad de manipular un gran volumen de información. Por tanto, es de suma importancia comprender los conceptos de organización y manipulación de la información que se realiza mediante las estructuras de datos que no es más que la forma de organizar de una manera particular los datos en una computadora con el propósito de ser utilizado de manera eficiente.

Variadas estructuras de datos se adecuan para diferentes situaciones, algunas otras son altamente especializadas para ciertas tareas.

Las colas son dos de las estructuras más utilizadas debido a su amplio ámbito de aplicación. En esta práctica se estudiarán principalmente las colas, su utilización y su implementación básica en el lenguaje C.

Una cola es un conjunto ordenado de elementos del que se pueden suprimirse elementos de un extremo (llamado la parte delantera de la cola)



Representación de una cola.

A una cola se le denomina como una lista FIFO (El primero en entrar, el primero en salir) que es lo contrario de una pila la cual es una lista LIFO (El último en entrar, el primero en salir).

En la nuestra vida cotidiana encontramos innumerables ejemplos de colas, lo podemos ver en una caseta de peaje, en la fila del banco, etc.

Se le aplican tres operaciones fundamentales a una cola. La operación  $\text{insertar}(q,x)$ , inserta el elemento "x" en la parte posterior de la cola q y establece su contenido en x. La tercera operación  $\text{empty}(q)$  retorna falso o verdadero, dependiendo si la cola contiene elementos.

La operación insertar puede ejecutarse siempre, ya que no existe un limite en la cantidad de elementos que puedan contenerse en una cola. Pero, la operación remover solamente puede aplicarse si la cola no se encuentra vacía.

El resultado de intentar remover un elemento de una cola vacía se le ha denominado como subdesbordamiento, por otro lado la operación empty siempre se puede aplicar.

Operaciones básicas.

- Crear: Se crea la cola vacía.
- Encolar: (añadir, entrar, insertar): Se añade un elemento a la cola.
- Desencolar: (sacar, salir, eliminar): Se elimina el elemento frontal de la cola, es decir, el primer elemento que entró.
- Frente: (Consultar): Devuelve el elemento frontal de la cola, es decir el primero en ingresar a la cola.

Las colas se usan con frecuencias para simular situaciones del mundo real, como puede ser en el sector transporte u operaciones de investigación (etc), dónde los objetos, eventos o personas son almacenados en una cola como datos para su posterior procesamiento.



Cola de personas.

## Planificación por turnos rotarios (Round Robin)

Es uno de los algoritmos más antiguos, sencillos y equitativos en el reparto del CPU entre los procesos, muy válido para entornos de tiempo compartido. Cada proceso se le tiene asignado un cierto intervalo de ejecución, lo cual se le llama quantum o cuenta. Si el proceso se llega a agotar su quantum de tiempo, se procederá a elegir otro proceso para poder ocupar la CPU. Si el proceso se bloquea o termina antes de agotar su quantum también se alterna el uso de la CPU.

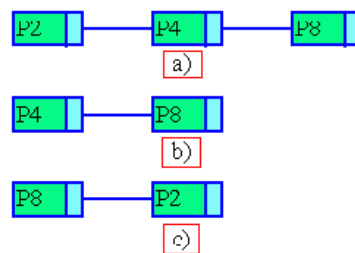


Figura 6.2 Lista de Procesos Preparados en Round Robin

RAFAGA	TIEMPO LLEGADA	REQUERIMIENTOS DE CPU(ms)
R1	0	16
R2	1	3
R3	2	2

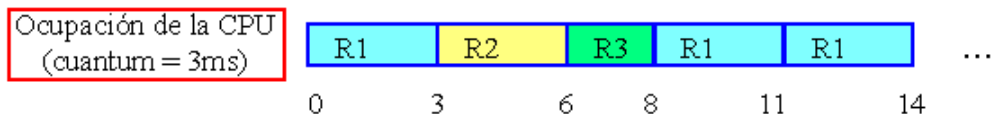


Figura 6.4 Ejemplo de Round-Robin

### Características.

- Periódicamente, se genera una interrupción de reloj.
- Se asigna un quantum de tiempo (10 – 100 ms) de igual duración a todos los procesos listos para ser ejecutados.

### Ventajas.

- Es equitativo.
- Sencillo de implementar.
- Las prioridades no cambian.
- Es un algoritmo muy predecible.

### Desventajas.

- Puede haber pospuesta indefinida.
- Un Quantum corto disminuye el rendimiento de la CPU.
- Un Quantum muy largo empobrece los tiempos de respuesta.

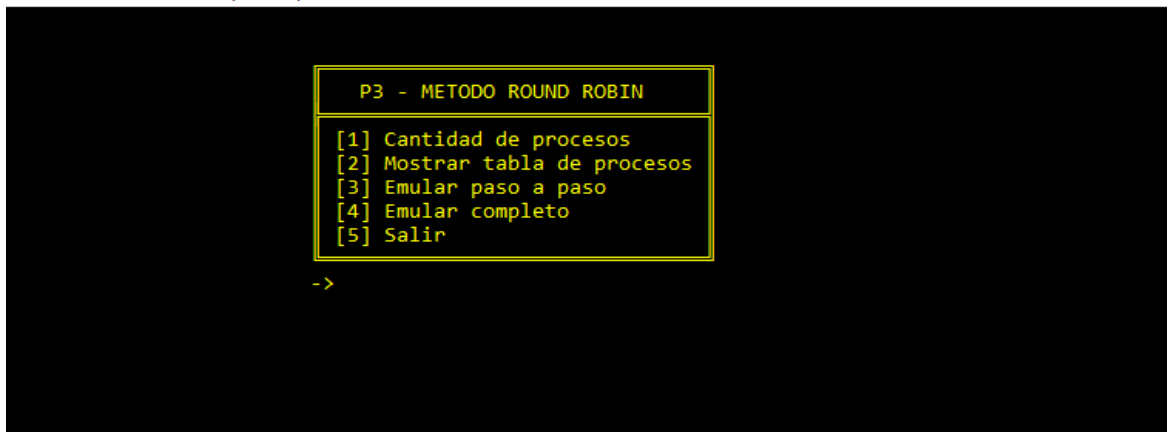
### **Experimentación y resultados.**

La práctica consiste en llevar a cabo una aplicación, la cual tiene como propósito la de emular la planificación por turnos rotario (Round Robin), el cual tendrá un menú con las siguientes opciones.

- Cantidad de procesos.
- Mostrar tabla de procesos
- Emular paso a paso.
- Emular completo.
- Salir

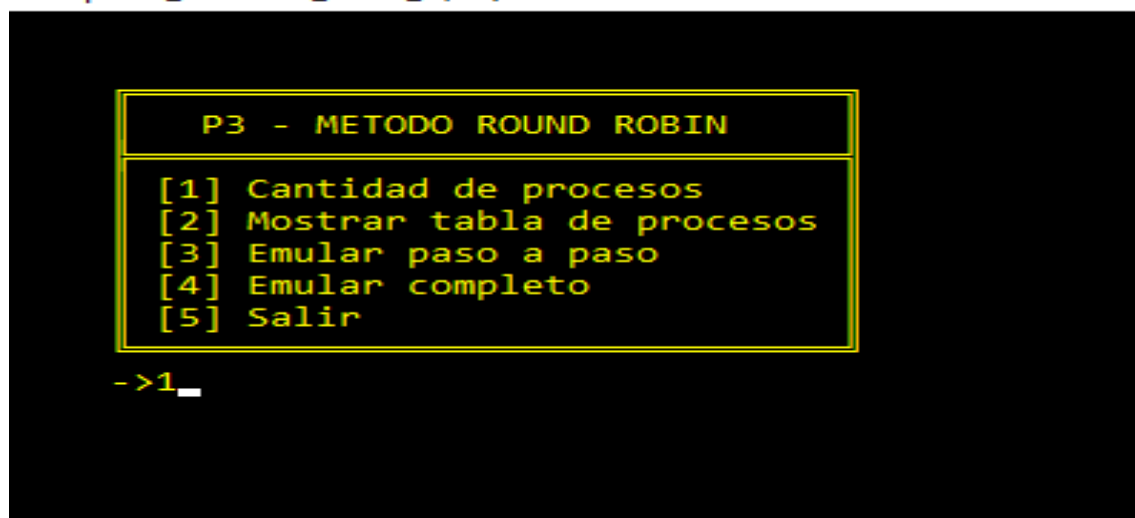
# Planificación por turnos rotarios Round Robin. 1

C:\Users\dxman\Desktop\champ\Pra\_3\Round\_robin\_1(a.c)\a.exe

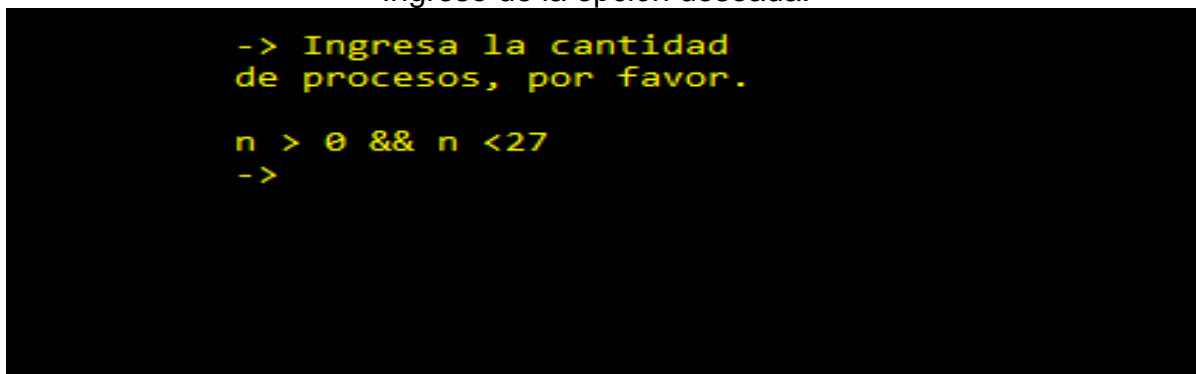


Solicitud de opción en el menú principal.

champ\Pra\_3\Round\_robin\_1(a.c)\a.exe



Ingreso de la opción deseada.



Solicitud del número de procesos entre un valor de 1 y 26.

hamp\Pra\_3\Round\_robin\_1(a.c)\a.exe

-> Ingresa la cantidad  
de procesos, por favor.

n > 0 && n < 27  
->10

Ingreso de un número de procesos.

\champ\Pra\_3\Round\_robin\_1(a.c)\a.exe

P3 - METODO ROUND ROBIN

[1] Cantidad de procesos  
[2] Mostrar tabla de procesos  
[3] Emular paso a paso  
[4] Emular completo  
[5] Salir

-> \_

Retorno al menú principal.

P3 - METODO ROUND ROBIN

[1] Cantidad de procesos  
[2] Mostrar tabla de procesos  
[3] Emular paso a paso  
[4] Emular completo  
[5] Salir

->2



Ingreso a la segunda opción del menú.

C:\Users\dxman\Desktop\champ\Pra\_3\Round\_robin\_1(a.c)\a.exe

TABLA DE PROCESOS		
A	1	18
B	2	18
C	3	10
D	4	14
E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15

Presione enter para retornar, por favor...

Visualización de la tabla de procesos.

P3 - METODO ROUND ROBIN	
[1]	Cantidad de procesos
[2]	Mostrar tabla de procesos
[3]	Emular paso a paso
[4]	Emular completo
[5]	Salir

->\_

Retorno al menú principal.

.champ\Pra\_3\Round\_robin\_1(a.c)\a.exe

```
P3 - METODO ROUND ROBIN

[1] Cantidad de procesos
[2] Mostrar tabla de procesos
[3] Emular paso a paso
[4] Emular completo
[5] Salir

->3_
```

Ingreso a la emulación paso por paso.

C:\Users\dxman\Desktop\champ\Pra\_3\Round\_robin\_1(a.c)\a.exe

```
QUANTUM = 3
```

Especificación de la cuenta seleccionada por la computadora de manera aleatoria.

TABLA DE PROCESOS		
A	1	18
B	2	18
C	3	10
D	4	14
E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15

1 - Impresión de la tabla inicial

### TABLA DE PROCESOS

B	2	18
C	3	10
D	4	14
E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15
A	1	15

2 - Disminuyendo A en 3 e ingresando nuevamente a la cola.

### TABLA DE PROCESOS

C	3	10
D	4	14
E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15

3 - Disminuyendo B en 3 e ingresando nuevamente a la cola.

### TABLA DE PROCESOS

D	4	14
E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7

4 - Disminuyendo C en 3 e ingresando nuevamente a la cola.

### TABLA DE PROCESOS

E	5	9
F	6	10
G	7	6
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7
D	4	11

5- Disminuyendo D en 3 e ingresando nuevamente a la cola.

### TABLA DE PROCESOS

F	6	10
G	7	6
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7
D	4	11
E	5	6

6- Disminuyendo E en 3 e ingresando nuevamente en la cola.

### TABLA DE PROCESOS

G	7	6
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7
D	4	11
E	5	6
F	6	7

7 - Disminuyendo F en 3 e ingresando nuevamente en la cola.

TABLA DE PROCESOS		
H	8	5
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7
D	4	11
E	5	6
F	6	7
G	7	3

8- Disminuyendo G en 3 e ingresando nuevamente en la cola.

TABLA DE PROCESOS		
I	9	3
J	10	15
A	1	15
B	2	15
C	3	7
D	4	11
E	5	6
F	6	7
G	7	3
H	8	2

El proceso I ha terminado

9 - Disminuyendo H en 3 e ingresando nuevamente en la cola.  
Culminación del proceso I, ya que  $3-3 = 0$ , por lo que no aparecerá en la siguiente tabla.

**TABLA DE PROCESOS**

J	10	15
A	1	15
B	2	15
C	3	7
D	4	11
E	5	6
F	6	7
G	7	3
H	8	2

10 - El proceso I ha culminado, por lo que no es ingresado en la cola.

**TABLA DE PROCESOS**

A	1	15
B	2	15
C	3	7
D	4	11
E	5	6
F	6	7
G	7	3
H	8	2
J	10	12

11 - Disminuyendo J en 3 e ingresando nuevamente en la cola.

**TABLA DE PROCESOS**

B	2	15
C	3	7
D	4	11
E	5	6
F	6	7
G	7	3
H	8	2
J	10	12
A	1	12

12 - Disminuyendo A en 3 e ingresando nuevamente en la cola.

**TABLA DE PROCESOS**

C	3	7
D	4	11
E	5	6
F	6	7
G	7	3
H	8	2
J	10	12
A	1	12
B	2	12

13 - Disminuyendo B en 3 e ingresando nuevamente en la cola.

**TABLA DE PROCESOS**

D	4	11
E	5	6
F	6	7
G	7	3
H	8	2
J	10	12
A	1	12
B	2	12
C	3	4

14 - Disminuyendo C en 3 e ingresando nuevamente en la cola.

**TABLA DE PROCESOS**

E	5	6
F	6	7
G	7	3
H	8	2
J	10	12
A	1	12
B	2	12
C	3	4
D	4	8

15 - Disminuyendo D en 3 e ingresando nuevamente en la cola.

TABLA DE PROCESOS		
F	6	7
G	7	3
H	8	2
J	10	12
A	1	12
B	2	12
C	3	4
D	4	8
E	5	3

16 - Disminuyendo E en 3 e ingresando nuevamente en la cola.

TABLA DE PROCESOS		
G	7	3
H	8	2
J	10	12
A	1	12
B	2	12
C	3	4
D	4	8
E	5	3
F	6	4

El proceso G ha terminado

17 - Disminuyendo F en 3 e ingresando nuevamente en la cola.

Culminación del proceso G, ya que  $3-3 = 0$ , por lo que no aparecerá en la siguiente tabla.



#### TABLA DE PROCESOS

H	8	2
J	10	12
A	1	12
B	2	12
C	3	4
D	4	8
E	5	3
F	6	4

El proceso H ha terminado

18 - Culminación del proceso G, ya que  $2-3 = -1$ , por lo que no aparecerá en la siguiente tabla.

#### TABLA DE PROCESOS

J	10	12
A	1	12
B	2	12
C	3	4
D	4	8
E	5	3
F	6	4

18- Sin el proceso H

#### TABLA DE PROCESOS

A	1	12
B	2	12
C	3	4
D	4	8
E	5	3
F	6	4
J	10	9

19 – Disminución del proceso J en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

B	2	12
C	3	4
D	4	8
E	5	3
F	6	4
J	10	9
A	1	9

20 – Disminución del proceso A en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

C	3	4
D	4	8
E	5	3
F	6	4
J	10	9
A	1	9
B	2	9

21 – Disminución del proceso B en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

D	4	8
E	5	3
F	6	4
J	10	9
A	1	9
B	2	9
C	3	1

22- Disminución del proceso C en 3 e ingreso a la cola.

#### TABLA DE PROCESOS

E	5	3
F	6	4
J	10	9
A	1	9
B	2	9
C	3	1
D	4	5

El proceso E ha terminado

23 – Disminución del proceso D en 3 e ingreso a la cola.

Culminación del proceso G, ya que  $2-3 = -1$ , por lo que no aparecerá en la siguiente tabla.

#### TABLA DE PROCESOS

F	6	4
J	10	9
A	1	9
B	2	9
C	3	1
D	4	5

24- Sin E

#### TABLA DE PROCESOS

J	10	9
A	1	9
B	2	9
C	3	1
D	4	5
F	6	1

25 - Disminución del proceso F en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

A	1	9
B	2	9
C	3	1
D	4	5
F	6	1
J	10	6

26 - Disminución del proceso J en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

B	2	9
C	3	1
D	4	5
F	6	1
J	10	6
A	1	6

27 – Disminución del proceso A en 3 e ingreso a la cola.

**TABLA DE PROCESOS**

C	3	1
D	4	5
F	6	1
J	10	6
A	1	6
B	2	6

**El proceso C ha terminado**

28 – Disminución del proceso B en 3 e ingreso a la cola.

Culminación del proceso C

### TABLA DE PROCESOS

D	4	5
F	6	1
J	10	6
A	1	6
B	2	6

29 – Sin C

### TABLA DE PROCESOS

F	6	1
J	10	6
A	1	6
B	2	6
D	4	2

El proceso F ha terminado

30 – Disminución en 3 del proceso D e ingreso a la cola.  
Culminación del proceso F.

### TABLA DE PROCESOS

J	10	6
A	1	6
B	2	6
D	4	2

31 – Sin F

#### TABLA DE PROCESOS

A	1	6
B	2	6
D	4	2
J	10	3

32 – Disminución del proceso J en 3 e ingreso a la cola.

#### TABLA DE PROCESOS

B	2	6
D	4	2
J	10	3
A	1	3

33 – Disminución del proceso A en 3 e ingreso a la cola.

#### TABLA DE PROCESOS

D	4	2
J	10	3
A	1	3
B	2	3

El proceso D ha terminado

34 – Disminución del proceso B en 3 e ingreso a la cola.  
Culminación del proceso D.

#### TABLA DE PROCESOS

J	10	3
A	1	3
B	2	3

El proceso J ha terminado

35- Culminación del proceso J.

TABLA DE PROCESOS

A	1	3
B	2	3

El proceso A ha terminado

36 – Culminación del proceso A

TABLA DE PROCESOS

B	2	3
---	---	---

El proceso B ha terminado

37 – Culminación del proceso B.

TABLA DE PROCESOS

B	2	3
---	---	---

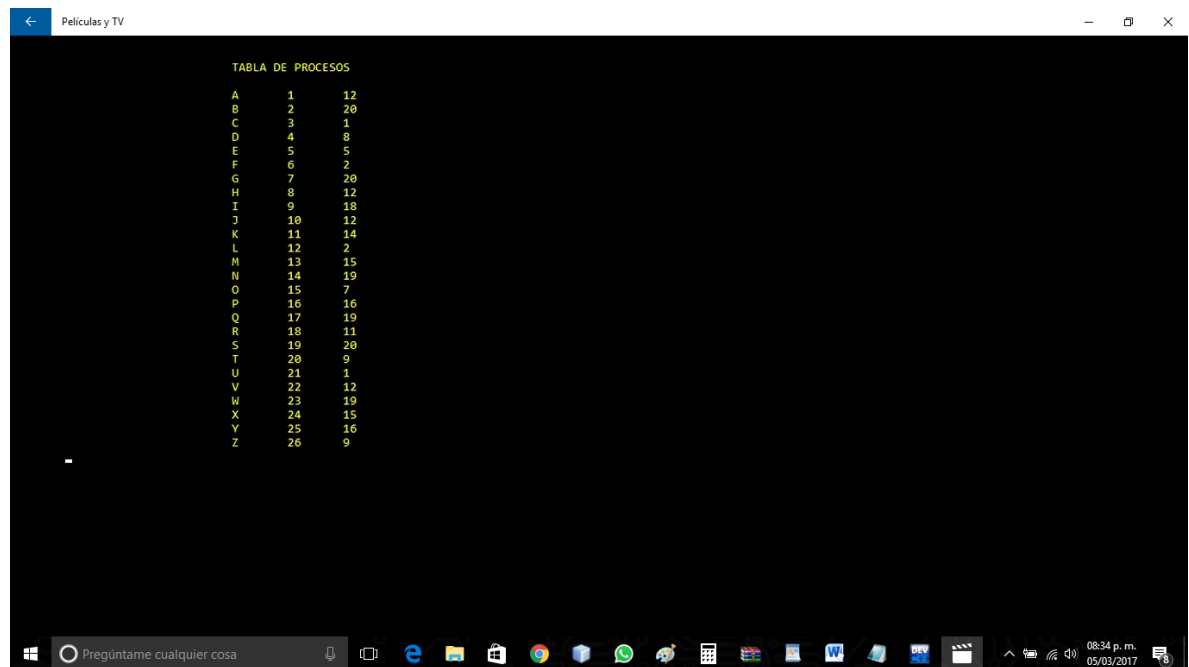
El proceso B ha terminado

Todos los procesos han terminado.

38 – Fin de la totalidad de los procesos.

Simulación completa.

Con el fin de tener una mejor apreciación de la simulación completa se ha anexado un video a este reporte, el cual muestra la simulación completa de 26 procesos, además de contar con otra simulación paso a paso en su interior.



Proceso	Prioridad	Tiempo
A	1	12
B	2	20
C	3	1
D	4	8
E	5	5
F	6	2
G	7	20
H	8	12
I	9	18
J	10	12
K	11	14
L	12	2
M	13	15
N	14	19
O	15	7
P	16	16
Q	17	19
R	18	11
S	19	20
T	20	9
U	21	1
V	22	12
W	23	19
X	24	15
Y	25	16
Z	26	9

Video.



Retorno al menú principal.



P3 - METODO ROUND ROBIN

- [1] Cantidad de procesos
- [2] Mostrar tabla de procesos
- [3] Emular paso a paso
- [4] Emular completo
- [5] Salir

->5\_

Ingreso de la opción deseada.

P3 - METODO ROUND ROBIN

- [1] Cantidad de procesos
- [2] Mostrar tabla de procesos
- [3] Emular paso a paso
- [4] Emular completo
- [5] Salir

->5

-----  
Process exited after 49.77 seconds with return value 4294967295  
Presione una tecla para continuar . . . \_

Fin del programa.

## Planificación por turnos rotarios Round Robin 1 (Pseudocódigo)

```
Entero num_procesos -> 0 , quantum ->0
Entero *num_ale;
tipoDato datos;
COLA COL
Inicio principal
  CrearCola(COL)
  menu_conlee()
Fin principal
Vacio mostrar_conenter (COLA COLU)
Inicio mostrar_conenter
  Carácter c
  COLA aux_2 -> COLU
  1 Si (num_procesos es distinto de cero) entonces
    Entero k -> num_procesos;
    Entero i -> 0, cont ->7
    Limpiarpantalla
    Imprimir quantum
    Dormir por un segundo
    Limpiarpantalla
    Imp_tabla(aux_2, num_procesos)

    Mientras(ColaVacia(aux_2) sea igual a cero) entonces
      Para i ->0 hasta i < k i en uno
        TABLA * x = quitarCola(aux_2)
        Si ( reque_cpu es mayor a 0 ) entonces
          Reque_cpu -> reque_cpu – quantum;
        Fin Si
        Si (reque_cpu es mayor a 0) entonces
          IngresarCola(aux_2, x)
        Si no
          K -> k -1
        Imprimir “El proceso ha terminado”
        Fin Si
        Obtener carácter desde consola
        Si c es igual a trece entonces
          Imp_tabla(aux_2, k)
      Fin Si
    Fin Para
  Fin Mientras
  Imprimir “Todos los procesos han terminado”
```

Si no  
Limpiar pantalla  
Imprimir "Ingresa el num de procesos primero"  
Obtener carácter  
Si (c es igual a 13)  
Limpiar pantalla  
menu\_conlee()  
Fin Si

Fin mostrar\_conenter

Vacio menú\_conlee  
Inicio menú\_conlee  
Entero opc -> 0  
Pintarletrasdeamarillo  
Menú()  
Leer\_opc  
Cont\_func(opc9  
Fin menú\_conlee

Inicio emular\_p\_a\_p  
Carácter c  
COLA aux\_2 -> COLU  
1 Si (num\_procesos es distinto de cero) entonces  
Entero k -> num\_procesos;  
Entero i -> 0, cont ->7  
Limpiar pantalla  
Imprimir quantum  
Dormir por un segundo  
Limpiar pantalla  
Imp\_tabla(aux\_2, num\_procesos)  
  
Mientras(ColaVacia(aux\_2) sea igual a cero) entonces  
Para i ->0 hasta i < k i en uno  
TABLA \* x = quitarCola(aux\_2)  
Si ( reque\_cpu es mayor a 0 ) entonces  
Reque\_cpu -> reque\_cpu – quantum;  
Fin Si  
Si (reque\_cpu es mayor a 0) entonces  
IngresarCola(aux\_2, x)  
Si no  
K -> k -1  
Imprimir "El proceso ha terminado"  
Fin Si  
Obtener carácter desde consola  
Dormir por mil milisegundos.  
Imp\_tabla(aux\_2, k)

```

Fin Para
Fin Mientras
Imprimir "Todos los procesos han terminado"

Si no
Limpiar pantalla
Imprimir "Ingresa el num de procesos primero"
Obtener carácter
Si (c es igual a 13)
Limpiar pantalla
menu_conlee()
Fin Si

Fin emular_p_a_p

Vacio imp_tabla(COLA aux, entero k)

Inicio imp_tabla
Entero i -> 0
COLA sin_md = aux
Si (k es diferente de cero) entonces
Imprimir "Tabla de procesos"
Para i-> hasta i<k i en uno
TABLA * y -> quitarCola(sin_md)
Imprimir elemento de tabla
Fin Para
Fin Si

Fin imp_tabla

Vacio cont_func (Entero opc)
Inicio cont_func

Switch (opc)
Caso 1
Cantidad_de_procesos()
Terminar
Caso 2
Mostrar_tab_proce()
Terminar
Caso 3
Mostrar_conenter()
Terminar
Caso 4
Emular_p_a_p(COL)
Terminar

```

Caso 5  
Salir  
Terminar

Omisión  
Limpiar pantalla  
Menú\_conlee()  
Fin Switch

Fin cont\_func

Vacio mostrar\_tab\_proce

Carácter c

Si (num\_procesos es distinto de cero) entonces

COLA sin\_md -> COL

Limpiar pantalla

Entero i -> 0, cont -> 7

Imprimir "Tabla de procesos"

Para i->0 hasta i< num\_procesos i en uno

TABLA \*x -> quitarCola(sin\_,md)

Imprimir elemento de la tabla

Fin para

Imprimir "Presione enter para retornar, por favor..."

Obtener carácter

Si (carácter es igual a trece) entonces

Limpiar pantalla

Menú\_conlee

Fin Si

En caso contrario

Limpiar pantalla

Imprimir "Ingresa un num de procesos primero, presiona ENTER para continuar..."

Obtener caracter

Si (c es igual a trece) entonces

Limpiar pantalla

Menú\_conlee()

Fin Si

Fin Si

Fin mostrar\_tab\_proce

Vacio cantidad\_de\_procesos

Limpiar pantalla

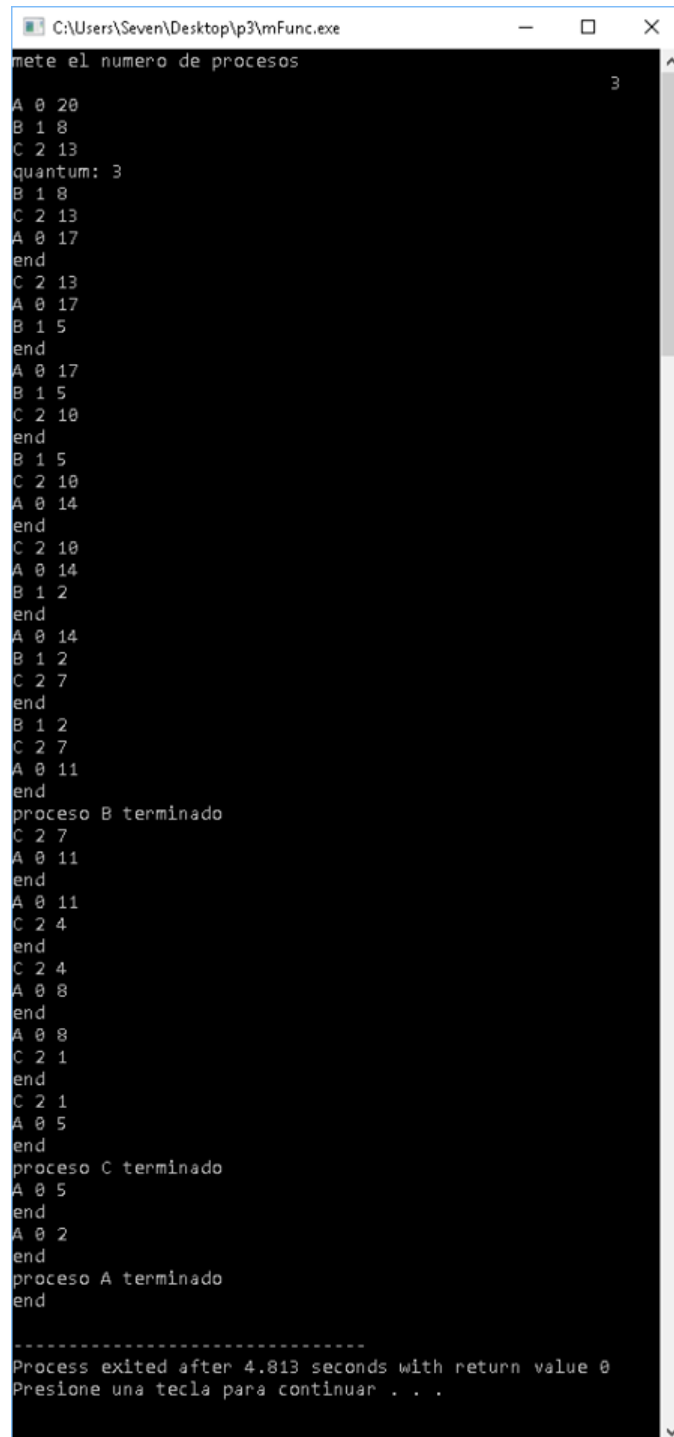
Semilla

Entero ca\_p, i ->0, y ->0

Imprimir "Ingresa la cantidad de procesos"

```
Leer y
Si (y está entra 1 y 26) entonces
    Num_procesos -> y
Si no
    Cantidad_de_procesos
Fin Si
Imprimir num_procesos
Asignar_memoria_dinámica a num_ale con base al num_procesos
Asignar_memoria_dinámica a datos con base al num_procesos
Carácter letra -> 'A'
Quantum -> numero aleatorio entre 3 y 5
Para (i->0 hasta i < num_procesos i en uno)
    Num_ale [en i] -> numero aleatorio entre 1 y 20
    Datos[ en i] letra -> letra
    Datos [en i] reque_cpu -> num_ale[en i]
    Datos[En i] tiempo_lle -> i+1
    Letra -> letra+1
    IngresarCola(COL, datos[en i])
Fin Para
Limpiar pantalla
Menú_conlee
Fin cantidad_de_procesos
```

## Planificación por turnos rotarios Round Robin. 2



```
C:\Users\Seven\Desktop\p3\mFunc.exe
mete el numero de procesos
3
A 0 20
B 1 8
C 2 13
quantum: 3
B 1 8
C 2 13
A 0 17
end
C 2 13
A 0 17
B 1 5
end
A 0 17
B 1 5
C 2 10
end
B 1 5
C 2 10
A 0 14
end
C 2 10
A 0 14
B 1 2
end
A 0 14
B 1 2
C 2 7
end
B 1 2
C 2 7
A 0 11
end
proceso B terminado
C 2 7
A 0 11
end
A 0 11
C 2 4
end
C 2 4
A 0 8
end
A 0 8
C 2 1
end
C 2 1
A 0 5
end
proceso C terminado
A 0 5
end
A 0 2
end
proceso A terminado
end

-----
Process exited after 4.813 seconds with return value 0
Presione una tecla para continuar . . .
```

Emulación de round robin, se aprecia la entrada de procesos, seguida de la asignación del quantum y los resultados de cada iteración del ciclo

## Planificación por turnos rotarios Round Robin. 2 (Pseudocódigo.)

//las unicas variables en donde se muestra el tipo de dato son en las TDA  
//pseudo de tipado debil

```
Funcion quant<- asignarQuantum ( )  
    quant = rand % 2 +3  
Fin Funcion
```

```
Funcion tam <- leerEntrada ( )  
    imprimir "mete el numero de procesos"  
    leer tam  
Fin Funcion
```

```
Funcion asignarMem ( COLA* c, tam )  
    tipoDato dato  
    Para i<-0 Hasta tam-1 Con Paso 1 Hacer  
        alocar memoria de dato  
        dato->nombre = 65 + i;  
        dato->tiempo = i;  
        dato->req = rand % 21;  
        ingresarCola(c, dato);  
    Fin Para  
Fin Funcion
```

```
Funcion correr ( COLA* c, quant )  
    tipoDato dato  
    imprimir "quantum: " + dato  
    mientras !colaVacía(c)  
        dato=quitarCola(c)  
        si dato->req-quant > 0 entonces  
            dato->req-=quant;  
            IngresarCola(c, dato)  
        sino  
            imprimir "proceso" + dato->nombre + "terminado"  
        FinSi  
        imprime(*c)  
        imprimir "fin"  
    FinMientras  
Fin Funcion
```

```
Funcion imprime ( COLA c )  
    tipoDato dato  
    i  
    si !colaVacía(&c) entonces
```



```

        mientras (c.frente + i)%TAM != c.fin%TAM hacer
            dato = c.CListaCola[c->frente]
            imprimir dato->nombre, dato->tiempo, dato->tiempo
            i++
        FinMientras
        dato = c.CListaCola[c->fin]
        imprimir dato->nombre, dato->tiempo, dato->tiempo
    FinSi
Fin Funcion

//este es el main
Algoritmo mFunc
    COLA c
    crearCola (&c)
    asignarMem(&c, leerEntrada)
    imprime(c)
    correr(&cm, asignarQuantum())
FinAlgoritmo
Conclusión.

```

Hernández Castellanos César Uriel.

El algoritmo de planificación de procesos de Round Robien es un algoritmo simple de implementar, en un sistema operativo, le es asignado a cada uno de los procesos una porción equitativa de tiempo y ordenada, lo que permite tratar a todos los procesos con una prioridad similar.

A pesar de ser un algoritmo de fácil implementación, se tuvieron algunos inconvenientes como el de poder ingresar/sacar TAD a una cola, lo que consumió más tiempo de lo necesario.

Martínez Islas Mauricio Joel.

No se si se pudo haber reducido el codigo de roundRobin, o siquiera si sea muy eficiente la manera en la que se implementó, pero pues de alguna manera funciona. Lo malo es que se cometió una falta de respeto al diseño y la manera en la que imprime la tabla no es la manera en la que viene especificada, pero simplemente se cometió esa falta por que se podía ahorrar líneas de código. También el código de Cola se tuvo que modificar por un problema que se tenía de lógica, pero pues logró salir adelante todo bien e incluso corre bien cuando TAM = n y el numero de procesos = n, obviamente esto no se obtuvo naturalmente, sino de un video informativo acerca de colas circulares.

## Referencias

- [1]Y. Langsam, M. Augenstein and A. Tenenbaum, *Estructuras de datos con C y C++*, 1st ed. Pearson Prentice Hall, 2000.
- [2]"6 Algoritmos de Planificación I", *Lsi.vc.ehu.es*, 2017. [Online]. Available: <http://lsi.vc.ehu.es/pablogn/docencia/manuales/SO/TemasSOuJaen/PLANIFICACIONDEPROCESOS/6AlgoritmosdePlanificacionI.htm>. [Accessed: 06- Mar- 2017].
- [3]"Data Structures - GeeksforGeeks", *GeeksforGeeks*, 2017. [Online]. Available: <http://www.geeksforgeeks.org/data-structures/>. [Accessed: 06- Mar- 2017].
- [4]"Data Structures - University of California, San Diego, Higher School of Economics | Coursera", *Coursera*, 2017. [Online]. Available: <https://es.coursera.org/learn/data-structures>. [Accessed: 06- Mar- 2017].
- [5]"Estructuras de datos - Colas", *C.conclase.net*, 2017. [Online]. Available: <http://c.conclase.net/edd/?cap=003>. [Accessed: 06- Mar- 2017].