

```

tiempo_actual = MINIMO TIEMPO DE LLEGADA DE TODOS LOS PROCESOS
cola_actual = COLA EN LA CUAL SE ENCUENTRA EL PROCESO CON TIEMPO = tiempo_actual
//IMPORTANTE: si existen varios procesos cuyo tiempo de llegada es tiempo_actual
//(tiempo= 0, por ejemplo) , cola_actual debera apuntar a la cola
//en la cual se encuentra el proceso de MAYOR prioridad con tiempo de llegada = tiempo_actual
proceso_actual = NULO

//Variable que cuenta los procesos que faltan por terminar
n = numero de procesos registrados que se encuentran en estado "iniciado" (todos)

//Inicialmente para todos los procesos, el tiempo que les falta por ejecutar
(tiempo restante de ejecucion) es igual a su tiempo de ejecucion

mientras (n > 0) //mientras exista un proceso sin terminar

    //Registrar primero llegadas de nuevos procesos
    si existen procesos en estado "iniciado"
        cuyo tiempo de llegada sea menor o igual a tiempo_actual
            pasar procesos a estado "listo"
            incluir estos procesos (en orden de
            llegada) en la cola de prioridad que le corresponda a cada uno
        fin si

    //Luego verificar si existia algun proceso en la CPU
    si proceso_actual != NULO
        cambiar estado de "ejecucion" a "listo"
        poner al proceso_actual en su cola correspondiente,
        de acuerdo con el algoritmo de planificacion de la cola
    fin si

    //Buscar el siguiente proceso a ejecutar en la cola actual
    //Primero debemos asegurarnos que la cola no este vacia,

```

```

//si es asi se debera pasar a la siguiente cola.
//IMPORTANTE: posible ciclo infinito si todas las colas estan
//vacias!!
mientras es_vacia (cola_actual)
    cola_actual = siguiente_colas()
    quantum = QUANTUM_DE_COLA_ACTUAL
fin_mientras
//POST: cola_actual no es vacia
//quantum : depende de cola_actual

proceso_actual = proceso al frente de cola_actual
//quitar proceso_actual de cola_actual
pasar proceso a estado "en ejecucion"

si proceso_actual != NULO //hay un proceso en la CPU!!
    si el quantum asignado a este proceso (depende de su prioridad)
    es menor a su tiempo restante de ejecucion,
        restar el quantum al tiempo restante de ejecucion del proceso
        para todos los procesos listos (en todas las colas)
            incrementar su tiempo de espera en el quantum que se le dio al proceso

        //EL PROCESO EJECUTO SU QUANTUM!

        actualizar el tiempo restante del proceso (correcto! restar el quantum!)

        //"Avanzar" en la simulacion...
        tiempo_actual = tiempo_actual + quantum

sino //El proceso termino en este quantum!!
    pasar el proceso a estado "terminado".
    para todos los procesos (en todas las colas) que se
    encuentren en estado de listo, incrementar su tiempo

```

```
de espera en el tiempo restante de ejecucion del proceso.
// "Avanzar" en la simulacion...
tiempo_actual = tiempo_actual + tiempo_restante_del_proceso
// El proceso no vuelve a su cola porque ya termino!!
tiempo_restante_del_proceso = 0
// Actualizar el numero de procesos terminados!!
n = n - 1
fin si
// Sin importar si el proceso termino o no, ya se le dio CPU a cola_actual.
// Se debe pasar a la siguiente cola de prioridad
// La CPU se asigna en forma Round-Robin para las colas
// No olvidar: si era la cola n, se debe pasar a la cola 1
cola_actual = siguiente_cola()
quantum = QUANTUM_DE_COLA_ACTUAL
fin si
fin mientras
```