

# Aclaraciones y Fe de Erratas

En este documento se pretende recalcar algunos aspectos del trabajo práctico que por la frecuencia de las consultas comprendemos que puedan requerir de una aclaración adicional.



## Error en el enunciado sobre notificación de rechazo de solicitud de recurso

Donde dice:

**Control de recursos totales y asignados:** Cuando un personaje solicita un recurso, [...] le responderá con una confirmación, o el rechazo (en este caso el personaje le notificará al *orquestador* el bloqueo).



Debe decir:

**Control de recursos totales y asignados:** Cuando un personaje solicita un recurso, [...] le responderá con una confirmación, o el rechazo (en este caso el personaje le notificará al *planificador del nivel* el bloqueo).

## Error en el enunciado del proceso Koopa

Donde dice:

El mismo recibe por parámetro el nombre del archivo que contiene una lista de solicitudes [...] y luego pregunta por pantalla el tamaño del segmento a crear.

Debe decir:

El programa debe ser ejecutado pasando como primer argumento el nombre del archivo que contiene una lista de solicitudes (./koopa archivo.lst). Al iniciar, este pregunta por pantalla el tamaño del segmento de memoria a crear y el algoritmo designado al grupo.

## La lista de solicitudes de ejemplo no coincide con el resultado

La operación "BORRAR:C" no debería estar en el listado de operaciones

Se debe poder configurar un tiempo de espera entre cada movimiento

**permitido por cada Planificador.**

Antes de enviar un mensaje de movimiento permitido cada hilo Planificador debe hacer una pausa configurable (usleep() o similar). De esta manera uno puede visualmente seguir la ejecución del proceso.

**El alcance del trabajo práctico no debe considerar la posibilidad que en un plan de objetivos por nivel se soliciten dos instancias del mismo recurso de forma consecutiva.**

Ejemplo, donde F es un recurso flor y M un recurso moneda:  
Un lista de objetivos “FFM” es un error pero “FMFM” es posible.

**El alcance del trabajo práctico no debe considerar la posibilidad de que un personaje ingrese a un nivel que no pueda completar.**

Ejemplo: Un personaje ingresa a un nivel con el objetivo de tomar en total tres instancias del recurso Flor y en el nivel a) no hay caja de recursos flor; o b) el total de los recursos de la caja Flor está configurado con un valor inferior a tres.

**Se considera que un personaje utilizó un quantum cuando realizó un movimiento.**

Operaciones como consultar la ubicación de un recurso o la propia solicitud de un recurso son acciones anteriores o posteriores a un movimiento y no se contabilizan como que usan quantum.

**Al solicitar un recurso el personaje interrumpe su ejecución**

Un personaje que solicita un recurso deja el lugar de movimiento al próximo personaje indefectiblemente, sin importar si había utilizado todos sus quantums o no.

**El único parámetro alterable en tiempo de ejecución es la cantidad de quantums del Planificador**

Si el Proceso Plataforma detectara que su archivo de configuración fue alterado durante la ejecución del programa, deberá volver a leerlo solo para actualizar su valor de quantum

## **Sobre la implementación de libmemoria.so**

El contenido de las particiones se debe guardar únicamente en el segmento reservado, y en ninguna otra parte de la memoria.

En la lista de particiones devuelta, el campo "dato" debe apuntar a alguna parte de ese segmento. Esta lista tiene que ser creada específicamente para el momento en que se invoque a "particiones", es decir, no debe ser la misma que su estructura administrativa, ya que será liberada luego de imprimir el listado.

No hace falta implementar ningún algoritmo de compactación.