# PROGRAMACION CONCURRENTE

## PRIMER PARCIAL

1. Suponga un juego donde hay 30 competidores. Cuando los jugadores llegan avisan al encargado, una vez que están los 30 el encargado del juego les entrega un numero aleatorio del 1 al 15 de tal manera que dos competidores tendrán el mismo numero. (Suponga que existe una función DarNumero() que devuelve en forma aleatoria un numero del 1 al 15, **el encargado no guarda el numero que les asigna a los competidores). Una vez que ya se entregaron** los 30 números, los competidores buscaran **concurrentemente** su compañero que tenga el mismo numero (tenga en cuenta que pueden empezar a buscar cuando todos los competidores tengan el numero **no antes**; Además la búsqueda de un jugador no interfiere con la búsqueda de otros que tengan distinto numero). Cuando los competidores se encuentran permanecen en una sala durante 15 minutos y dejan de jugar. Luego cada uno de los competidores avisa al encargado que termino de jugar y espera a que su compañero (el que tenia el mismo numero) también avise que finalizo para luego irse ambos; el encargado cuando llega el segundo competidor les devuelve a ambos el resultado que obtuvieron que es el orden en que se van. (Los primeros en irse, tendrán como resultado 1, los últimos 15). Para modelizar el tiempo utilice la función Delay(x) que produce un retardo de x minutos.

Modelice Con Semáforos

2. Se tienen N procesos que comparten el uso de una CPU.

Un proceso, cuando necesita utilizar la CPU, le pide el uso, le proporciona el trabajo que va a ejecutar y el tiempo que insume ejecutar el trabajo. Luego, el proceso se queda dormido hasta que la CPU haya finalizado de ejecutar su trabajo, momento en que es despertado y prosigue su ejecución normal.

La CPU funciona de la siguiente manera. Cada vez que el proceso i (i:1..N) pide CPU esta lo pone en el lugar isesimo de una lista de procesos esperando para ejecutar. La CPU recorre circularmente la lista de procesos esperando a ser ejecutados (del 1 al N) , si el proceso esta listo lo saca de la lista y lo ejecuta durante un lapso de 10 mls (o un lapso menor, si el tiempo de ejecución del proceso es menor a 10 mls). Una vez que ejecutó un proceso, si todavía le queda al proceso tiempo de ejecución lo vuelve a poner en su lugar en la lista y pasa al siguiente. Si el proceso terminó de ejecutar su trabajo, la CPU lo despierta para que continúe su ejecución y continúa con el siguiente proceso de la lista. Cuando la lista está vacía, la CPU no debe hacer nada, simplemente debe esperar a que algún proceso ingrese a la lista de procesos en espera de ejecución, para empezar a trabajar.

Tenga en cuenta que existe la función delay(x) que retarda un proceso durante x mls. La ejecución de un proceso puede ser modelizada utilizando esta función.

No haga suposiciones acerca de los tiempos de ejecución de los procesos

Modelice utilizando Monitores