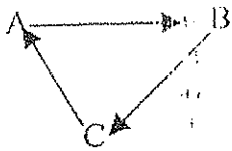


PROGRAMACIÓN CONCURRENTENTE

PRIMER RECUPERATORIO

1. Suponga que existen N barcos que recorren un canal llevando gente. Su recorrido es el siguiente:



A-B, B-C, C-A.

Los barcos parten desde A, van hasta B y luego van hasta C, y de C nuevamente vuelven a A. Luego de hacer 7 vueltas completas paran durante 1h a descansar y vuelven a salir.

No puede haber más de tres barcos haciendo el mismo recorrido simultáneamente.

Cuando un barco llega a un punto de parada, los pasajeros que quieren bajar lo hacen.

Una vez que bajaron todos el barco se fija si puede partir (teniendo en cuenta que no puede haber más de tres barcos haciendo el mismo recorrido), si no puede se queda esperando sin subir gente hasta poder partir.

Cuando se dan las condiciones de poder seguir permite subir gente durante 5' o hasta que se llene su capacidad. Suponga que cada barco tiene capacidad para 30 pasajeros.

Existen R pasajeros que conocen la parada en la que suben y en cual bajan. Cuando un pasajero llega a su punto de partida trata de subir en el primer barco que parta, cuando su barco llega a la parada destino se baja. Tenga en cuenta que el barco no conoce el destino de sus pasajeros.

Suponga que como mínimo cada recorrido del barco tarda 20'. Maximice la concurrencia.

Modelice con Pasajes de Mensajes Asincrónicos o Sincrónicos según crea conveniente.

2. 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 número aleatorio del 1 al 15 de tal manera que dos competidores tendrán el mismo número. (Suponga que existe una función `DarNumero()` que devuelve en forma aleatoria un número del 1 al 15, el encargado no guarda el número 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 número (tenga en cuenta que pueden empezar a buscar cuando todos los competidores tengan el número no antes; Además la búsqueda de un jugador no interfiere con la búsqueda de otros que tengan distinto número). 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 terminó de jugar y espera a que su compañero (el que tenía el mismo número) también avise que finalizó 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 Regiones Críticas.