

Bondi en Pandemia – Versión Semáforos.

Para este ejercicio, se me ocurrió desarrollarlo en primer lugar con semáforos, puesto que dan la versatilidad en el código de manejar cantidades, el planteo en sí, desde que desarrollamos el TP de BODES, no pude dejar de pensar en la concurrencia aplicada a casos cotidianos, es por eso que me basé en este caso y me pareció interesante la forma en que los semáforos dan esa ventaja.

En cuanto al código de esta versión:

Tenemos una clase “BONDI” que posee cierta capacidad, en cuanto a esto es completamente seteable y por lo tanto maneja la cantidad que queramos. No tiene mucho más.

Class Pasajero:

Será de quién se creen instancias y serán los “Pasajeros” (threads) que suban y bajen del “Bondi” (semáforo), con la particularidad de que en el código se encuentran 2 “while”, con uno se crean las instancias de pasajeros y son siempre los mismos los que suben y bajan del “Bondi”.

Habilitando el otro “while” una vez que descienden, estos desaparecen y se vuelven a crear nuevos “Pasajeros”.

Ventajas de la concurrencia: En este ejercicio, esta claro que las ventajas de la concurrencia apuntan a que el uso de “threads” simplifican la forma económica de correr el programa. El uso de “semáforos” evitan la condición de carrera, como hemos aprendido, y por lo tanto es un claro ejemplo de la versatilidad de los mismos.

Se utilizaron los siguientes links, como material de consulta:

<http://webdiis.unizar.es/~ezpeleta/lib/exe/fetch.php?media=misdatos:pc:10.pdf>

<http://www.laminfo.com/blog/archivos/ Teoria 1 Programacion Concurrente.pdf>

El desarrollo del problema y su solución es íntegramente idea de Sebastián Brandariz.

La culpa de que a partir de ahora no deje de buscar y generar nuevos desafíos que impliquen concurrencia, es del Profesor Sebastián Pedersen.