|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Monitores** (métodos syncronized) sin comunicación entre hilos | **Monitores** + control manual de las condiciones = uso de métodos de **comunicación entre hilos** (wait y notify) | **Semáforos (regiones críticas/comunicación de hilos)** | Exchanger | CountDownLatch (cuenta atrás, sirve para operar desde el hilo principal con los resultados obtenidos de los hilos secundarios) | CyclicBarrier | Executor (Pool de hilos) |
| Problema del Lector-Escritor |  | x | X | X (creo) |  |  |  |
| Problema del Jardín | x |  |  |  |  |  |  |
| Servidor web - clientes | x |  |  |  |  |  |  |
| Productor - Consumidor |  |  |  | x |  |  |  |
| Sumas de números en una matriz |  |  |  |  | X (los hilos realizan la suma de las filas de la matriz) |  |  |
|  |  |  |  |  |  |  |  |

Notas importantes para diferenciar mejor cada solución:

* Si se opta por wait() y notify(): no es necesario usar yield()
* Si el hilo sólo necesita 1 "hueco" pones acquire() para acceder y release() para salir
* Si el hilo necesita más de 1 hueco (como el limpiador), entonces pones acquire (X) y release(X) siendo X el número de huecos