

## Productors/Consumidors/i altres

En un entorn hi ha tres tipus de processos que comparteixen recursos:

1. Els processos **Productors** que produeixen seqüències de tipus enter. Aquestes seqüències estan formades per  $1 \leq s \leq S$  elements de dades i al final 2 elements de control, també enters (per tant una seqüència pot contenir un màxim d' $S + 2$  enters). El primer enter de control s'obté sumant els enters de dades que es troben a les posicions parells de la seqüència (considerant la primera posició com la posició 0) i el segon sumant els de les posicions senars (0 si no hi ha posicions senars). S'ha de dir que a vegades els càlculs dels elements de control són erronis.
2. Els **Verificadors** que s'encarreguen de verificar que els enters de control de les seqüències siguin correctes.
3. Els **Consumidors** que consumeixen els enters de dades de les seqüències que han estat donades per bones en el procés de verificació.

Per tal de sincronitzar tot aquest entorn es disposa d'un **Monitor** que proporciona als diferents tipus de procés els següents mètodes:

1. `public void put(int[] sq)`. Aquest mètode permet guardar una seqüència al monitor. En cas de no haver-hi espai suficient per a guardar tota la seqüència, el procés s'espera fins que aquest espai estigui disponible.
2. `public int verificar(int[] sq)`. Aquest mètode és invocat pels Verificadors i posa al paràmetre `sq` la seqüència que toca verificar (cada seqüència només es verifica un cop). Com que el verificador no sap la longitud de la seqüència que ha de verificar, aquesta s'afegeix a la posició 0 de la seqüència retornada, és a dir, `sq[0]=longitud` i les dades a verificar comencen a la posició 1 d'`sq`. Aquest mètode té un valor de retorn que identifica la seqüència.
3. `public void resVerificar(boolean ok, int id)`. Aquest mètode és invocat pels Verificadors per a informar del resultat de la verificació, a través del paràmetre booleà `ok`. Les dades (els valors de control no) de les seqüències verificades com a bones són acceptades i es posen a disposició dels Consumidors. Les seqüències verificades com a errònies són eliminades del sistema. El paràmetre `id` és l'identificador de la seqüència que s'ha verificat.

Es pot considerar (no és obligatori) que les seqüències verificades es processen (és a dir, s'accepten o s'eliminen) seguint l'ordre en que han estat lliurades a verificar (bàsicament perquè pot ajudar a simplificar la solució del problema).

4. `public void get(int[] sq, int offset, int length)`. Mètode invocat pels Consumidors per a demanar dades ja verificades i acceptades. Els `length (> 0)` enters de dades sol·licitats pel Consumidor es guarden a `sq`, a partir de la posició `offset`. Si no hi ha la quantitat de dades que sol·licita el Consumidor aquest s'ha d'esperar fins que hi sigui.

El punt més important a l'hora de resoldre aquest problema és decidir quina estructura de dades s'utilitza en el monitor per a guardar les dades que comparteixen els diferents processos.