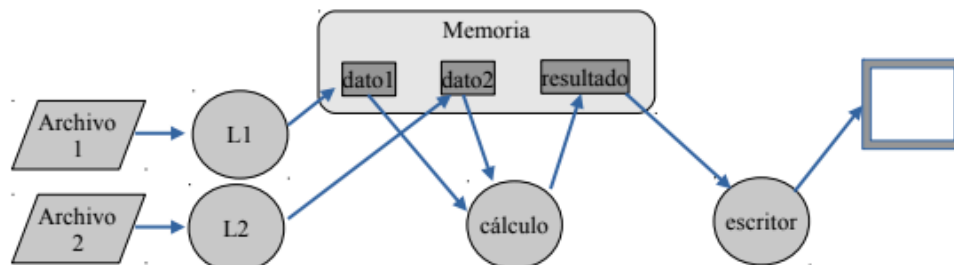


### Monitores: Lectores, cálculo y escritor.

**01/02)** En el siguiente esquema se muestran dos procesos  $L_1$  y  $L_2$ , que leen repetidamente datos de dos archivos, cada uno en forma independiente y lo colocan en una variable de memoria: **dato1** y **dato2** respectivamente. Luego, estas dos variables son leídas por un proceso: **cálculo**, que las suma entre sí y al resultado obtenido lo pone en la variable **resultado**. Esta variable es finalmente leída por un proceso **escritor** que lo vuelca a una pantalla.

Ud. deberá escribir un programa concurrente utilizando monitores para sincronizar los dos lectores, **cálculo** y **escritor**; de forma tal que no se pierda ningún dato en todo el proceso.



\* Especifique los supuestos que asuma (por ej. tipo de monitor)

**Incidencia: 50/100**

Pd: Especifique los supuestos que considere (por ejemplo, tipo de monitor).

### Mensajes. Transporte de token

**02/02)** Se pide que mediante un programa concurrente que utiliza mensajes para sincronizar, simule el comportamiento de  $n$  **Agentes** que deben transportar **dos tokens** en una disposición circular y en un único sentido (hacia la derecha). El modo en que funciona es el siguiente: cada Agente debe esperar a que le sean entregados los tokens por el Agente que está a su **izquierda**, una vez recibido, los entrega al Agente que está a su derecha y se queda esperando hasta que aquel lo haya recibido. Una vez que los entregó al Agente que está a su derecha, vuelve a esperar que el Agente a su izquierda le vuelva a entregar los tokens para así seguir indefinidamente repitiendo el proceso. Su solución debe además llevar la cuenta de la cantidad de entregas exitosas que han sufrido los tokens.