

## Ejercicio hilos propuesto 1: Hilos cooperantes

**n hilos de la misma clase.**

**Realizan la misma tarea y comparten acceso al mismo objeto.**

**Sincronización: necesito definir código sincronizado, para impedir que más de un hilo acceda a la vez al objeto /recurso compartido (la lista encadenada).**

La tarea a realizar es:

Procesar un **conjunto de ficheros** de texto con listas de nombres, para cada cadena leída en cada fichero calculamos su longitud y sus iniciales ( aquí suponemos un proceso más complejo con cada cadena del fichero, que es uno de los motivos que puede hacer que decidamos lanzar un hilo por cada fichero a procesar), y a continuación, almacenamos en una lista encadenada ordenada alfabéticamente la información.

En la lista almacenamos para cada cadena: la longitud de la cadena , las iniciales y la propia cadena.

Cada hilo lanzado realiza la misma tarea accede a un fichero de cadenas, lo lee secuencialmente e inserta un elemento en la lista encadenada. Por tanto diseñamos la clase a la que pertenecerán los hilos, todos hilos de la misma clase que tienen previsto recibir dos objetos del hilo primario.

*La **lista encadenada de elementos de tipo “persona”** es el **recurso compartido** por los n hilos arrancados.*

*No podemos permitir que dos hilos accedan al mismo tiempo a insertar un elemento en la lista.*

El hilo primario lanza tantos hilos como ficheros de texto tengamos.

El hilo primario recibe los ficheros en su lista de argumentos : cada argumento será la especificación en disco de cada fichero texto a procesar.

Procesa la lista de argumentos recibidos y para cada fichero recibido lanza un nuevo **hilo de la clase HiloLector**, pasándole el fichero y el mismo objeto (la referencia de la misma lista encadenada en memoria.

Podemos lanzar otro hilo demonio que repita una música en espera a la terminación del proceso.

Una vez realizado este proceso, el hilo primario puede implementar una operación join con respecto a sus hilos hijos lanzados.

Al final del proceso, cuando todos los hilos hayan terminado su ejecución, tendremos en memoria toda la información procesada y ordenada, **información elaborada con la ejecución simultánea y sincronizada de los n hilos lanzados**, en tonces el hilo primario vuelca la lista en un fichero de salida.

**Recurso compartido:** lista encadenada. Pasamos a cada hilo la misma lista encadenada.

**Fichero texto no es recurso compartido**, pasamos a cada hilo un fichero distinto.

Definimos en cada ejecución del proceso tantos hilos como ficheros de texto recibas en los argumentos.

Al final el hilo primario vuelca la lista en un fichero nuevo.