

# Prácticas Concurrencia y Distribución (16/17)

Arno Formella, Francisco Rodríguez Martínez,  
Javier Rodeiro Iglesias, David Olivieri

semana 27–31 marzo

Seguimos con la actividad de la metáfora de la lectura concurrente de un libro. Añadimos los siguientes requisitos a los ya existentes:

- Se garantiza que se lee cada página exactamente una vez.
- Los lectores leen las páginas en turnos, es decir, lector 1 lee página 1, lector 2 lee página 2, ..., lector  $n$  lee página  $n$ , y se empieza con lector 1 otra vez hasta que se haya acabado el libro.
- El libro gestiona internamente a cual de los lectores toca el turno de leer una página.

Aprovechando de la estructura principal del programa de los apartados anteriores, considera las tareas siguientes:

5. Modifica el método `Read()` de la clase `Book` para que devuelva un booleano verdadero si se ha podido leer una página, sino, es decir, que no haya más páginas, que devuelva falso.
6. Modifica el correspondiente recuento de las páginas leídas en el método `run()` de los hilos.
7. Aumenta la clase del libro para gestionar los turnos, por ejemplo, que sepa cuantos lectores hay, que haya un método `MyTurn(id)` que devuelve si el lector con la `id` debe leer, y que haya un método `NextTurn()` que prepara el turno para el siguiente lector.
8. Usa en el bucle del método `run()` de los lectores los métodos `wait()` and `notify()` respectivamente `notifyAll()` de tal manera que un hilo vuelve a esperar si no es su turno cuando le hayan despertado.
9. Realiza un gráfico de los tiempos de ejecución manteniendo el número de páginas por leer fijo pero incrementando el número de lectores. Para tal fin, no realices ninguna llamada a ningún `sleep()` en el programa.