

# Programación Concurrente en Java

## Laboratorio 2: FindDuplicates

J. Blanco, N. Wolovick

### Objetivos

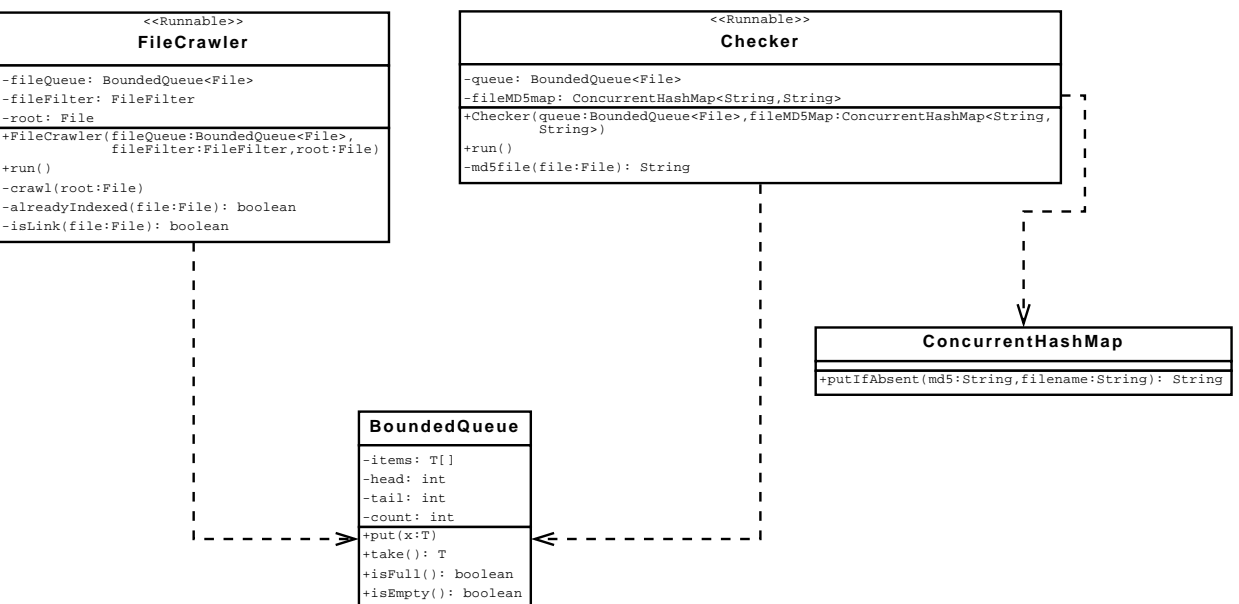
- Implementar *skip guardados* para asegurar las precondiciones de una Región Crítica Condicional.
- Aplicar el patrón productor/consumidor en un ejemplo real.

### Actividades

1. Leer y entender el código completo.  
Observar el uso de `Runtime.getRuntime().availableProcessors()` y la clase `ConcurrentHashMap`. Mirar con más detenimiento la clase `BoundedQueue`.
2. [25 pts] Probar el programa y ver que saltan las aserciones que son implicadas por el invariante de representación.  
Ayuda: sigue estando la clase `StoYield` para generar planificaciones más ricas.
3. [25 pts] Agregar en `put()` y `take()` las aserciones que indican las precondiciones que debe cumplir la cola para poder poner y tomar. Probar que efectivamente no se cumplen.
4. [50 pts] Sincronizar para que no salten las aserciones de “*state dependence*” (Capítulo 14 de JCIP) usando dos mecanismos:
  - a) Locks y condiciones **implícitas** a la clase `Object: wait, notify, notifyAll` (Sección 14.1.3 de JCIP).
  - b) Locks y condiciones **explícitas**: `Lock (ReentrantLock), Condition`.  
Usar **dos** variables de condición, una para dormir a los procesos que esperan por un lugar en la cola y otra para dormir a los procesos que están esperando un dato en la cola (Sección 14.4 de JCIP).

Generar dos clases nuevas (`BoundedQueue1.java`, `BoundedQueue2.java`) cada una en un archivo separado, de forma tal que haciendo `mv BoundedQueue1.java BoundedQueuee.java`, se pueda compilar y ejecutar `FindDuplicates`.

## Diagrama de Clases



## Actividades Extra

- [15 pts] ¿Es necesario el uso de un ciclo `while (!conditionPredicate())` como muestra el listado 14.7 de JCIIP? ¿Es posible reemplazar el `while` por un `if` sin tener problemas de *safety* o *liveness*? Explique tanto para 4a como para 4b.
- [20 pts] Rearmar el ejemplo para obtener *repetibilidad* (el `FileSystem` mete mucho ruido) y medir la diferencia de performance entre la variable de condición implícita y las dos explícitas para sincronizar `BoundedQueue`.