

### Paso 7.-

Ahora vamos a hacer de forma concurrente el llenado de gasóleo y agua. Para ello crearemos dos threads por cada barco, cada uno haciendo una tarea. Estos threads se crearán dentro del método `run()` del barco petrolero. El barco ha de esperar a que terminen los dos nuevos threads. Cuando los dos threads han terminado su tarea, el barco puede continuar. Hay que lanzar estos threads con `Executor`. Observad que estos threads se lanzan desde un sitio distinto al `main`.

### Paso 8.-

En clases de teoría hemos visto diferentes clases y frameworks que nos permiten simplificar la creación y sincronización de threads (`CyclicBarrier`, `CountDownLatch`, `SynchronousQueue`, `Exchanger`, etc).

En este paso de la práctica se pide rehacer el proyecto Puerto Marítimo, pero usando las nuevas clases y frameworks aprendidos. La solución se simplificará en algunos puntos y entenderemos perfectamente por qué surgieron. Siéntete libre de usar las que quieras. Al menos deberías utilizar tres de ellas.

En la fecha que se indique deberás entregar dos proyectos diferentes:

1. un proyecto con todo lo que has hecho justo antes del paso 7.
2. un proyecto donde aparece la solución incluyendo el paso 7 y 8 (esto implica sustituir cosas hechas en los pasos anteriores por las nuevas)