## **Semaphores y Thread Executors**

## P3

En el segundo apartado de dicha entrega se implementa, a partir de un código proporcionado, una suma de matrices usando ThreadPools.

Crear u destruir Threads es una tarea bastante costosa. Por esto si vamos a realizar tareas cortas y repetitivas es mas eficiente crear un ThreadPoolExecutor al que se le pueden ir pasando tareas que debe ejecutar.

Para ello el código proporcionado aplica un algoritmo que divide en 4 cuadrantes las matrices, siempre que el tamaño sea una potencia de 2, y las suma.

Para que una tarea pueda devolver un resultado debe implementar la interfaz Callable<E>. Dicha interfaz permite lanzar una tarea que devuelve un dato en forma de Future<E>. Para lanzar la tarea se utiliza el método Future.sumit(new Task()). En dicho Future<E> podemos consultar el estado de la tarea con métodos como isDone() (está terminada) o get() (resultado que devuelve).

La principal limitación del método proporcionado es la obligatoriedad a proporcionar matrices cuadradas de dimensión potencia de 2. Podria generalizarse adaptándola a matrices de cualquier tamaño.