

## Actividad Obligatoria Práctica 9. Programación Concurrente.

Realizar este ejercicio como actividad autónoma del alumno si no diese tiempo a realizarla en el laboratorio de prácticas.

La actividad **deberá concluirse antes del próximo laboratorio** puesto que será utilizada.

### Enunciado

Se entrega un fichero llamado `codigo.alumnos.zip` con un proyecto de consola C# que trabaja con información de cotizaciones de Bitcoin extraídas de un sitio real. Este código obtiene un array de objetos `BitcoinValueData`, que contienen el valor de la criptomoneda Bitcoin en una fecha y hora determinadas. Con esta información, y partiendo del proyecto `vector.modulo` visto durante el laboratorio, se debe hacer lo siguiente:

- Implementar una aplicación que calcule de forma concurrente el nº de veces que el valor del Bitcoin está por encima de un valor prefijado (que se pasa como parámetro al Master) en el array de valores que se proporciona. El Master debe devolver el nº de veces que el valor del bitcoin ha sido mayor o igual que dicho valor suministrado. Por ejemplo, en el periodo en el que se suministran los datos, el Bitcoin ha estado por encima de los 7000 euros 2826 veces.
- Validar su correcta implementación, tanto secuencial como concurrente (múltiples hilos), haciendo uso de la herramienta de *testing* de Visual Studio.
- Medir el *context switching*, analizando cuando se obtiene el máximo beneficio al utilizar varios hilos y el punto en el que el coste es mayor que la solución secuencial.
- Llamar a `GC.Collect()` y `GC.WaitForFullGCComplete()` después de cada ejecución del algoritmo.
- Ejecutar la prueba dos veces para evitar los efectos de la compilación JIT en los resultados. Ejecutar en modo Release (no en modo Debug)
- En una hoja *Excel*, mostrar el gráfico de evolución de *context switching*, indicando los valores descritos en el punto anterior.

Utilice correctamente todos los elementos de programación aprendidos hasta ahora.