

Implementar los siguientes ejercicios a partir del proyecto `vector.modulus` de la solución `threads`. Entregar una solución que incluya una aplicación de consola, que invoque los métodos pedidos y/o muestre el resultado de los cálculos realizados.

1. (2 puntos) Modificar las clases `Master` y `Worker` de modo que tengan dos atributos `vector1` y `vector2` del mismo tipo que el atributo de la clase original. Manteniendo el esquema de particionado de datos y el diseño `Master/Worker`, modificar todo lo necesario para transformar `ComputeModulus` en un método que calcule la distancia entre esos dos atributos en lugar del módulo del atributo de la clase original. La distancia se define como la raíz cuadrada de la suma de los cuadrados de las diferencias de las coordenadas, es decir para dos vectores x e y de tamaño n :

$$\sqrt{\sum_{i=0}^{n-1} (x_i - y_i)^2}$$

2. (3 puntos) En la clase `Master` implementar el método `ComputeDistanceF`, el cual usando `TPL` y `PLINQ` realiza el mismo cálculo que el método del ejercicio anterior.
3. (2.5 puntos) Calcular la varianza de un vector de `int` usando la siguiente estrategia. Se calculan $E(x)$ y $E(x^2)$ por separado, cada una para las posiciones pares e impares del vector. Las cuatro tareas se lanzan en paralelo usando `Invoke`. Al terminar se calcula $(E(x^2) - (E(x))^2)$.
4. (2.5 puntos) Calcular el histograma conjunto de los dos vectores usando la siguiente estrategia: lanzar en paralelo dos tareas, de modo que la primera calcula el histograma de los números del primer vector, la segunda lo mismo con los números del segundo vector. Cada una de estas tareas puede ser secuencial. Después acumular el valor de los dos histogramas usando `Parallel.For`: el número de veces que se repite cada número en total es el número de veces que se repite en el primer vector más el número de veces que se repite en el segundo vector.