**Primera versión**

1. Diseñe e implemente un sistema de software para calcular el precio de una opción de compra de manera secuencial (utilizando el algoritmo presentado anteriormente) con componentes SCA (usand FraSCAti).
2. El usuario de su sistema entrará los valores de entrada en un componente GUI, que no incluirá ningún tipo de cálculos. El sistema deberá realizar el cómputo secuencial de las simulaciones y mostrar en la misma GUI el intervalo de confianza donde se encuentra la prima de la opción de compra.
3. El sistema también deberá permitir recibir un archivo plano CSV con los valores de varias opciones call a las que se les deberá calcular la prima. Este archivo tendrá un call por línea, y las columnas incluirán: el nombre de la opción, S, r, E, σ, M, y la semilla del generador aleatorio.

Las soluciones deberán escribirse en un archivo de salida, con una línea por call, y las columnas: nombre de la opción, seed, M, valor de la acción subyacente promedio a término S(T), valor promedio del call, límite inferior del call y límite superior del valor del call, tiempo tomado por el cálculo de la prima de cada call.

Deberá utilizar una implementación de un generador pseudo aleatorio, que le permita inicializar una semilla (seed), de tal forma que pueda reproducir los resultados encontrados anteriormente si así lo requiriese. Pueden utilizar la clase java.util.Random con su método nextGaussian para tal efecto.

Deberá entregar en un archivo zip como sigue:

MonteCarlo-Secuencial-ApellidosIntegrantes.zip

* Archivo Word con la explicación de la solución, incluyendo un diagrama de deployment con notación SCA
* Archivo readme.txt con el nombre de los integrantes, instrucciones de compilación, instrucciones de ejecución de los componentes.
* Código fuente de la solución realizada y composite FraSCAti
* Archivo de test con 5 entradas para el mismo call (inspirarse de las acciones de la BVC <http://www.dataifx.com/mercado#lista>), sólo con el número de simulaciones Monte Carlo diferente (1000, 5000, 10000, 50000, 100000).

Incluir además 5 entradas para un call con las siguientes características: S=102, E=100, T=2 (años), σ=25%, r= 6.5%, siguiendo los mismos números de simulaciones anteriores.

* Archivo de salida correspondiente con los resultados del pricing de las 10 entradas.

No se recibirán trabajos por otro medio diferente a Moodle, sin excepción.

**Segunda versión**

Rediseñe la solución de la versión anterior para que el cálculo de las simulaciones Monte Carlo sea distribuible, además de escalable en tantos componentes FraSCAti como sea necesario.

Para propósitos prácticos, su solución debe distribuirse en 5 nodos de procesamiento.

Deberá entregar en un archivo zip como sigue:

MonteCarlo-Distribuido-ApellidosIntegrantes.zip

* Documentos
  + Archivo Word con la explicación de la solución, incluyendo un diagrama de deployment con notación SCA, especificando la estructura de distribución en hardware (SISD, SIMD, MISD, MIMD) y la estructura de memoria que usa su arquitectura (UMA, NUMA, NORMA).
  + Archivo readme.txt con el nombre de los integrantes, instrucciones de compilación, instrucciones de ejecución de los componentes.
  + Resultado de pruebas experimentales con varios nodos de procesamiento, según los datos de la primera versión
* TipoNodoProcesamiento1
  + Src
    - resources (archivos .composite FraSCAti, archivo readme con instrucciones para ejecutar)
    - java (código fuente de clases e interfaces Java de los composite del folder respectivo)
  + clases (código compilado de clases e interfaces Java, incluyendo .jars resultantes)
* …
* TipoNodoProcesamientoN
  + Src
    - resources (archivos .composite FraSCAti, archivo readme con instrucciones para ejecutar)
    - java (código fuente de clases e interfaces Java de los composite del folder respectivo)
  + clases (código compilado de clases e interfaces Java, incluyendo .jars resultantes)