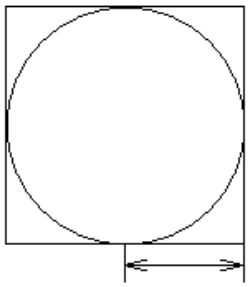
Programación de Sistemas y Concurrencia

Práctica Evaluable
Curso 2021-2022

Descripción del sistema

Queremos probar 2 métodos iterativos para estimar el valor de PI:

- Estimación de PI utilizando Montecarlo: supongamos que tenemos un círculo de tamaño R inscrito en un cuadrado como muestra la siguiente figura. Para estimar PI podemos ir generando puntos aleatorios dentro de las coordenadas del cuadrado. Algunos de estos puntos estarán también dentro del círculo y otros no. Para un punto (x,y), éste se encontrará dentro del círculo si $x^2+y^2<R^2$. El método para estimar PI se basa en la proporción de puntos que deberían estar dentro del círculo, de manera que finalmente, el valor de PI es 4 * (número de puntos dentro del círculo) / número total de puntos.

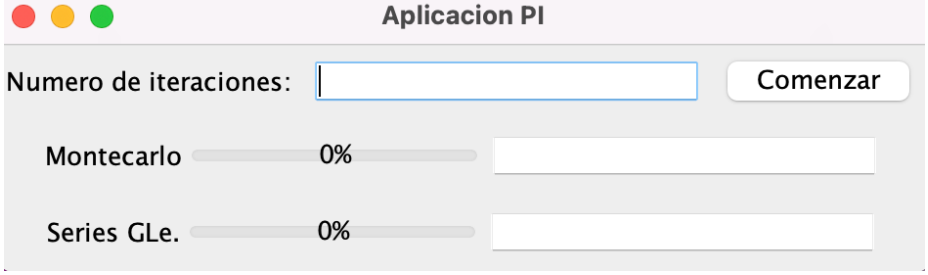


- Series infinitas de Gregory-Leibniz: En este caso, el valor de PI puede aproximarse utilizando series infinitas. La fórmula que debe aplicarse es:
  PI=(4/1)-(4/3)+(4/5)-(4/7)+(4/9)-(4/11)+(4/13)-(4/15)…
  Es decir, hay que hacer sumas y restas sucesivas con 4 como numerador y una serie ascendente de números impares.

Se pide realizar los siguientes ejercicios. En el campus virtual tienes la plantilla de la clase Panel que te servirán para empezar el proyecto. Recuerda que tienes que entregar cada uno de los ejercicios (no entregues solo la versión final) para que podamos evaluar los cambios llevados a cabo.

Ejercicio 1 (0.5 puntos): GUI con `SwingWorker` con `done`

1. Incluye en tu proyecto la clase `Panel`, que presentará un aspecto similar al que se muestra en la siguiente figura.



2. Crea una clase `Controlador` que implemente las interfaces necesarias para manejar los eventos del botón comenzar.
3. Crea una clase `WorkerMontecarlo` que aproxime el valor de PI usando el método de Montecarlo, generando tantos puntos aleatorios como el usuario indique el usuario en el campo de texto "Número de iteraciones". Devolverá el valor al final.
4. Crea una clase `WorkerSeries` que aproxime el valor de PI utilizando el método de series infinitas, generando tantos términos de la serie como indique el usuario en el campo de texto "Número de iteraciones". Devolverá el valor al final.
5. Crea una clase principal que construya una hebra dispatcher desde la que se cree la GUI.

Nota: En esta primera versión no es necesario controlar la barra de progreso.

Ejercicio 2 (0.3 puntos): GUI con `SwingWorker` con `publish` y barra de progreso.

1. Modifica las clases `WorkerMontecarlo` y `WorkerSeries` para que se vayan publicando las aproximaciones de PI a medida que se van generando. Además, los worker tienen que actualizar su progreso de manera que las dos barras cada barra de progreso muestre el porcentaje de trabajo realizado por cada worker.
2. Modifica la clase Controlador para que implemente las interfaces necesarias para manejar los eventos de cambio de progreso de los worker.

Ejercicio 3 (0.2 puntos): Realiza las modificaciones necesarias en la clase Panel, Controlador y los Workers para que sea posible cancelar los cálculos. Por ejemplo, puedes modificar el botón "Comenzar" para que pueda usarse para Cancelar la ejecución de los worker. También puedes optar por añadir un nuevo botón. Puedes añadir un pequeño retardo (por ejemplo, Thread.sleep(50)) para que los Workers tarden un poco en llevar a cabo su tarea y te dé tiempo a cancelarlos.