

Proyecto 1: Recursividad, algorítmica y medición de tiempos

- *iterativeFactorial (int n)*
- *recursiveFactorial (int n)*
- *iterativeFibonacci (int n)*
- *recursiveFibonacci (int n)*
- *iterativePow (int base, int exp)*
- *recursivePow (int base, int exp)*
- *recursivePow1 (int n)*
- *recursivePow2 (int n)*
- *recursivePow3 (int n)*
- *doNothing (int n)*
- *lineal ()*
- *quadratic (int n)*
- *cubic (int n)*
- *logarithmic (int n)*

Gráficos de Análisis Empírico de Eficiencia

Siguiendo las indicaciones del profesor, el alumno utilizará la función de hoja de cálculo para generar dos gráficos de rendimiento e imprimirlos en PDF, los cuales adjuntará al proyecto.

- **Gráfico A.** Incluirá la representación gráfica de las complejidades obtenidas tras la ejecución de los algoritmos *lineal*, *quadratic*, *cubic* y *logarithmic*.
Cada uno de los algoritmos se ejecutará con un SLEEP_TIME interno de 250 milisegundos, se ejecutarán cada uno 5 veces y con valores de carga que irán desde 1 hasta n (siendo n un valor a elegir por el alumno).
- **Gráfico B.** Incluirá la representación gráfica de las complejidades de todas las versiones del algoritmo de cálculo de potencia de 2.
Cada uno de los algoritmos se ejecutará con un SLEEP_TIME interno de 250 milisegundos, se ejecutarán cada uno 5 veces y con valores de carga que irán desde 1 hasta n (siendo n un valor a elegir por el alumno).

Consideraciones finales

- Todos los métodos diseñados deben ser evaluados por el alumno mediante una exhaustiva batería de pruebas haciendo uso de **JUnit**.

Fecha de Entrega

Una vez completado el proyecto, deberá subirse al Campus Virtual en las fechas que se designen. Pasado ese período no se aceptará la entrega del proyecto de ningún modo.