

Programación Concurrente y de Tiempo Real
Grado en Ingeniería Informática
Examen Final de Prácticas (**MODELO A**)
Septiembre de 2020

Apellidos:

Nombre:

Grupo (A ó B):

1. Notas

1. Escriba su nombre y apellidos con letra clara y legible en el espacio habilitado para ello.
2. Firme el documento en la esquina superior derecha de la primera hoja y escriba debajo su D.N.I.
3. Dispone de 2 horas para completar el ejercicio.
4. Puede utilizar el material bibliográfico (libros) y copia de API que estime convenientes. Se prohíbe el uso de apuntes, *pen-drives* y similares.
5. Entregue sus productos, utilizando la tarea de entrega disponible en el Campus Virtual, en un fichero (.rar, .zip) de nombre

Apellido1Apellido2Nombre

que contendrá una subcarpeta por cada enunciado del examen, la cual contendrá a su vez el conjunto de ficheros que den solución a ese enunciado en particular.

6. **No se aceptarán bajo ningún concepto entregas efectuadas por medios diferentes a la tarea de subida, ni fuera del plazo de examen establecido. Por tanto, debe asegurarse de entregar sus productos en el plazo establecido y mediante la tarea habilitada a tal efecto.**

2. Criterios de Corrección

1. El examen práctico se calificará de cero a diez puntos y ponderará en la calificación final al 30 % bajos los supuestos recogidos en la adenda a la ficha de la asignatura.
2. Las condiciones que una solución a un enunciado de examen práctico debe cumplir para considerarse correcta son:
 - a) Los ficheros subidos a través del Campus Virtual que conforman el examen práctico se ajustan al número, formato y nomenclatura de nombres explicitados por el profesor en el documento de examen.
 - b) El contenido de los ficheros es el especificado por el profesor en el documento de examen en orden a solucionar el enunciado en cuestión.
 - c) Los programas elaborados por el alumno, se pueden abrir y procesar con el compilador del lenguaje, y realizan un procesamiento técnicamente correcto, según el enunciado de que se trate.
 - d) Se entiende por un procesamiento técnicamente correcto a aquél código de programa que compila correctamente sin errores, cuya semántica dá soporte a la solución pedida, y que ha sido escrito de acuerdo a las convenciones de estilo y eficiencia habituales en programación

3. Enunciados

1. (Integración Numérica, regla del rectángulo, 5.5 puntos) Se desea conocer mediante una aproximación numérica el resultado de integrar una función real de variable real en un intervalo $[a, b]$, utilizando la «regla del rectángulo», descrita por la siguiente ecuación:

$$\int_a^b f(x) dx \approx (b - a)f(a)$$

Se pide escribir un programa en Java que implemente el cálculo mediante procesamiento paralelo de la aproximación numérica a la integral definida en $[a, b]$ de la función $f(x) = 3x^2 + 2$. Para ello deberá:

- Dividir el intervalo de integración en una partición con un número par de subintervalos, y aplicar la regla del rectángulo en cada uno de ellos, sumando finalmente los resultados parciales. El número de subintervalos se leerá por teclado, y los extremos a y b del intervalo principal, también.
- Efectuar la división del intervalo de integración de forma automatizada.
- Modelar sus tareas mediante las interfaz `Callable` y recoger los resultados parciales en una estructura de datos adecuada, utilizando la interfaz `Future`. El número de tareas `Callable` debe ser igual al número de intervalos de integración.
- Procesar sus tareas mediante un ejecutor del tipo que crea oportuno.

- El programa ofrecerá al terminar su ejecución el resultado de la aproximación de la integral (que debe ser coherente con el resultado de calcular la integral obteniendo una primitiva de la función y aplicando la regla de Barrow al intervalo escogido), el tiempo total de cálculo y el speedup.

Guarde su código en `intRectangulo.java`.

2. (Producto Interno con Paso de Mensajes, 4.5 puntos) Se desea calcular el producto interno de dos vectores de números reales de dimensión $n = 8$ utilizando una aproximación distribuida que emplee paso de mensajes en Java mediante el API MPJ-Express. Para ello deberá:

- Utilizar un esquema de procesos master-slave con un master y 8 slaves distribuidos.
- Leer los dos vectores a multiplicar en el proceso master, y efectuar el producto interno distribuido con los procesos slave.
- Imprimir, utilizando el proceso master, el resultado final del producto interno y el tiempo de ejecución total requerido.
- **Es obligatorio que incluya en su programa (al principio) un bloque de comentarios donde especifique las órdenes de línea de comandos necesarias para compilar y ejecutar su código.** Su programa será analizado utilizando esas órdenes.

Guarde su código en `innerProduct.java`.