

## Trabajo Práctico N° 5

**Tema:** Interfaz Runnable

**Fecha Inicio:** 12/09/2024    **Fecha de Entrega:** 26/09/2024

### Actividades:

- 1) Desarrolle el punto 2 del TP N° 3 y 1.a del TP N° 4, implementando la interfaz Runnable.
- 2) Realice un programa que simule un estacionamiento con capacidad para 20 automóviles. Dicho estacionamiento posee dos entradas y dos salidas, un automóvil permanece un tiempo y luego abandona el lugar, con lo cual tendrá que ser cuidadoso para controlar el acceso a dicho estacionamiento de manera de bloquear a los autos que deseen ingresar cuando la capacidad se ha completado. Simule la E/S de 100 automóviles. El estacionamiento inicia vacío.
- 3) Dentro de la sección policiales del diario El Tribuno de Jujuy (<https://eltribunodejujuy.com/seccion/policiales>), seleccione los enlaces de 10 noticias y cárguelos en un array. Cree una tarea que tenga como atributo dicho array y desde el main() lance 10 hilos para ejecutar dicha tarea. Esta tarea lo que hará es tomar el ID del array de noticias pasado desde el main(), realizará una petición HTTP Request y obtendrá la nota. Dentro de cada nota existe una sección (<DIV></DIV>) llamado <div amp-access="mostrarNota"> </div>. Dentro de ese DIV está el cuerpo de la noticia, que es lo que nos interesa. Luego, a medida que vaya terminando cada hilo de ejecutarse, debe imprimir por pantalla el cuerpo de la noticia encontrado. Muestre el tiempo de ejecución.
- 4) Cree una rutina que realice la multiplicación de dos matrices ( $A \times B = C$ ) de forma concurrente. Para ello debe tener en cuenta lo siguiente:
  - a. La matriz **A** será de 20x15 y la matriz **B** será de 15x20, con lo cual la matriz **C** será de 20x20 previamente definida.
  - b. Ambas matrices deben inicializarse con números enteros aleatorios entre [5-15].
  - c. La matriz resultante (**C**) será el objeto a sincronizar.
  - d. Debe existir una clase (*CalculoConcurrente*) que implemente la interfaz Runnable que realice el cálculo de la multiplicación de los vectores ( $A_{[1][1-15]} \times B_{[1-15][1]}$ ), y cada resultado se irá ingresando en la matriz **C** según la conversión que se indica en el inciso “e”, por ello la necesidad de sincronizar los accesos a este objeto.
  - e. Además la clase *CalculoConcurrente* debe heredar de la clase *Operaciones* que posee un método para cálculo de raíces que es el siguiente:

```
public static double SumRootN(int root)
{
    double result = 0;
    for (int i = 0; i < 10000000; i++)
    {
        result += Math.exp(Math.log(i) / root);
    }
    return result;
}
```

- f. Debe mostrar ambas matrices al inicio y el resultado al final del proceso.
- g. Calcule y muestre el tiempo de ejecución del proceso.

**NOTA**

Los puntos obligatorios de este TP y que serán presentados por los grupos son: los puntos 3 y 4.