

# PCTR Practica 9

Ignacio Rodríguez Pérez

Ejercicio 2: Cadena de producción

## 1 Código del monitor 1

```
public class monitorCadena_1
{
    int elementos = 0;

    public synchronized void introducir_matriz(int [][] matriz, int pos)
    {
        while (elementos >= 100)
            try
            {
                System.out.println(" Buffer 1 lleno (100 max)");
                wait( );
            } catch (InterruptedException e) { }
        //En la linea sgte no deberia haber espacios ,pero son para su mejor lectura
        ++ elementos;
        for (int i = 0; i < matriz.length; i++)
            for (int j = 0; j < matriz.length; j++)
                UsamonitorCadena.matrizIntermedia[pos][i][j] = matriz[i][j];
        System.out.println("Se introduce en el buffer1 en la posicion " + pos);
        notifyAll( );
    }

    public synchronized int [][] retirar_matriz(int pos)
    {
        while (elementos == 0)
            try
            {
                System.out.println(" Buffer 1 vacio");
                wait( );
            } catch (InterruptedException e) { }
        -- elementos;
        int [][] matriz = new int
        [UsamonitorCadena.matrizIntermedia[0].length]
        [UsamonitorCadena.matrizIntermedia[0].length];
        for (int i = 0; i < matriz.length; i++)
            for (int j = 0; j < matriz[0].length; j++)
                matriz[i][j] = UsamonitorCadena.matrizIntermedia[pos][i][j];
        notifyAll( );
        return matriz;
    }
}
```

## 2 Código del monitor 2

```

public class monitorCadena_2
{
    int elementos=0;

    public synchronized void introducir_matriz(int [][] matriz,int pos)
    {
        while (elementos == 50)
            try
            {
                System.out.println(" Buffer 2 lleno (50)");
                wait( );
            } catch (InterruptedException e) { }

        ++ elementos;

        for (int i = 0; i < matriz.length; i++)
            for (int j = 0; j < matriz.length; j++)
                UsamonitorCadena.matrizFinal[pos][i][j] = matriz[i][j];

        notifyAll();
    }

    public synchronized int [][] retirar_matriz(int pos)
    {
        while (elementos == 0)
            try
            {
                System.out.println(" Buffer 2 vacio");
                wait( );
            } catch (InterruptedException e) { }

        int [][] matriz = new int
        [UsamonitorCadena.matrizFinal[0].length]
        [UsamonitorCadena.matrizFinal[0].length];

        for (int i = 0; i < matriz.length; i++)
            for (int j = 0; j < matriz.length; j++)
                matriz[i][j] = UsamonitorCadena.matrizFinal[pos][i][j];

        -- elementos;
        notifyAll( );
        return matriz;
    }
}

```

### 3 Explicación

El primer monitor se puede considerar dos vectores; el primero recibe matrices aleatorias de forma concurrente, de modo que sólo las que se inicialicen más rápido serán almacenadas. Dichas matrices pasarán a otro vector, el cual se encarga de transponer dichas matrices.

Por otro lado, tenemos el segundo monitor que, además de tener también dos vectores, el primero de ellos hace de puente con el monitor anterior, ya que coincide con el segundo vector del monitor inicial. Por último, tenemos el segundo vector del segundo que permite imprimir el resultado que se pide en el enunciado(el producto de la diagonal principal de cada matriz).