


Aplicaciones para Ambientes Distribuidos

Laboratorio # 1 – Prof. Regis Rivera

Objetivo: Crear aplicaciones sencillas tipo consola en Java para aplicar primeros conceptos de concurrencia

Nota: Los nombres de archivos de cada aplicación están dados en el icono 

Laboratorio 1.1: Crear una salida de N números aplicando concurrencia

```
import java.lang.Math ;

class EjemploThread extends Thread {
    int numero;

    EjemploThread (int n) {
        numero = n;
    }

    public void run() {
        try {
            while (true) {
                System.out.println (numero);
                sleep((long)(1000*Math.random()));
            }
        }
        catch (InterruptedException e) { return; } // acaba este thread
    }

    public static void main (String args[]) {
        for (int i=0; i<10; i++)
            new EjemploThread(i).start();
    }
}
```



lab11.java

Aspectos interesantes:

- Explique la salida del programa
- Mencione cuantos hilos se ejecutaron
- ¿Qué haría si se quisiera agregar un segundo hilo al programa? Explique

Laboratorio 1.2: Ejecutar 3 tareas concurrentes



lab12.java

```
import java.util.concurrent.*;

public class EjemploConcurrenciaJava {
    public static void main(String[] args) {
        // Crear un ExecutorService para administrar hilos
        ExecutorService executor = Executors.newFixedThreadPool(3);

        // Crear tareas (Runnable) que se ejecutaran en paralelo
        Runnable tarea1 = () -> {
            for (int i = 0; i < 5; i++) {
                System.out.println("Tarea 1: " + i);
            }
        };
        Runnable tarea2 = () -> {
            for (int i = 0; i < 5; i++) {
                System.out.println("Tarea 2: " + i);
            }
        };
        Runnable tarea3 = () -> {
            for (int i = 0; i < 5; i++) {
                System.out.println("Tarea 3: " + i);
            }
        };
        // Ejecutar las tareas en hilos separados
        executor.execute(tarea1);
        executor.execute(tarea2);
        executor.execute(tarea3);

        // Cerrar el ExecutorService cuando ya no se necesite
        executor.shutdown();
    }
}
```

Aspectos interesantes:

- Explique la salida del programa
- ¿Cuántos hilos se ejecutan? ¿Por qué?
- ¿Qué haría si se quisiera agregar un hilo adicional al programa? Explique



lab13.java

```
public class NoSincronizada extends Thread {
    static int n = 1;

    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println(n);
            n++;
        }
    }

    public static void main(String args[]) {
        Thread thr1 = new NoSincronizada();
        Thread thr2 = new NoSincronizada();

        thr1.start();
        thr2.start();
    }
}
```

Aspectos interesantes:

- Explique la salida del programa
- ¿Cuál de las opciones de la imagen aparecen en su ejecución? ¿Por qué?

