



Nome: _____	Nota: _____
-------------	-------------

Modulo: Programación de Servicios e Procesos EXAME: Primeira Avaliación Data: 28/10/2025

ACLARACIONES

Todos los programas, al mostrar algo por pantalla, deberán hacerlo de forma legible y con comentarios que indiquen qué es cada cosa. De no hacerlo, no se obtendrá la calificación completa del ejercicio.

Un mal diseño tampoco obtendrá la calificación completa del ejercicio, es decir, nada de clases repetitivas. Recordad lo comentado en clase a este respecto.

Cada ejercicio debe estar en un proyecto distinto. Llama al profesor cuando acabes para entregarlo

ES FUNDAMENTAL LEER BIEN EL ENUNCIADO DE CADA EJERCICIO. LO QUE NO CUMPLA CON LO SOLICITADO NO SE VALORARÁ.

EL USO O CONSULTA DE MATERIAL/HERRAMIENTAS NO PERMITIDAS O COPIAR DE COMPAÑEROS, IMPLICA SUSPENDER EL EXAMEN.

Método de entrega:

A) Una vez terminado crear un repositorio en github y subir todos los proyectos.

Haced el repositorio privado y compartidlo con el profesor. Cuenta de github: diego-castelao

B) Entregar todo comprimido en un archivo en el aula virtual.

Las preguntas de teoría se pueden responder en un fichero markdown dentro del repositorio. Por ejemplo: “preguntas 3-4”.



Ejercicio 1. (3 puntos): Haz un programa que cree 3 hilos y los ejecute. Los hilos escriben 8 veces el número de iteración del bucle y su nombre. En cada iteración, después de escribir su nombre, se bloquean durante un tiempo y después vuelven a estar disponibles para su ejecución. Los hilos se deben comportar de manera secuencial inversa, es decir (esperar a que el tercer hilo acabe para que se ejecute el segundo y a su vez que el segundo acabe para que se ejecute el primero). El programa principal no acabará hasta que no terminen los 3 hilos, escribiendo un mensaje indicando que acabó.

Ejemplo de salida:

```
Soy el Hilo 3 - iteración: 1
Soy el Hilo 3 - iteración: 2
Soy el Hilo 3 - iteración: 3
Soy el Hilo 3 - iteración: 4
Soy el Hilo 3 - iteración: 5
Soy el Hilo 3 - iteración: 6
Soy el Hilo 3 - iteración: 7
Soy el Hilo 3 - iteración: 8
Soy el Hilo 2 - iteración: 1
Soy el Hilo 2 - iteración: 2
Soy el Hilo 2 - iteración: 3
Soy el Hilo 2 - iteración: 4
Soy el Hilo 2 - iteración: 5
Soy el Hilo 2 - iteración: 6
Soy el Hilo 2 - iteración: 7
Soy el Hilo 2 - iteración: 8
Soy el Hilo 1 - iteración: 1
Soy el Hilo 1 - iteración: 2
Soy el Hilo 1 - iteración: 3
Soy el Hilo 1 - iteración: 4
Soy el Hilo 1 - iteración: 5
Soy el Hilo 1 - iteración: 6
Soy el Hilo 1 - iteración: 7
Soy el Hilo 1 - iteración: 8
Programa principal terminado.
```



Alumna/o: _____

Nota: _____

Modulo: Programación de Servicios e Procesos EXAME: Primeira Avaliación Data: 28/10/2025

Ejercicio 2. (3 puntos): Crea un programa que utilice 4 hilos para contar el número total de dígitos pares en una cadena de texto y el número de ocurrencias de cada uno de ellos. Cada hilo será responsable de contar cuántas veces aparece uno de los cuatro dígitos pares (2, 4, 6, 8) en el texto. Todos los hilos se ejecutarán **concurrentemente** y actualizarán **una misma variable compartida que almacenará el TOTAL de dígitos pares encontrados**.

El texto puede ser introducido por teclado o leído de un fichero (NO ES NECESARIO HACER LAS DOS OPCIONES CON UNA ES SUFICIENTE).

Ejemplos de salidas

```
Introduce una cadena de texto (o escribe 'archivo' para leer de un fichero):  
12345678901234567890
```

```
Resultados:  
Total de dígitos pares encontrados: 8  
Frecuencia de cada dígito par:  
Dígito 2: 2  
Dígito 4: 2  
Dígito 6: 2  
Dígito 8: 2
```

```
Introduce una cadena de texto (o escribe 'archivo' para leer de un fichero):  
archivo  
Introduce el nombre del archivo:  
prueba.txt
```

```
Resultados:  
Total de dígitos pares encontrados: 13  
Frecuencia de cada dígito par:  
Dígito 2: 1  
Dígito 4: 4  
Dígito 6: 4  
Dígito 8: 4
```



Ejercicio 3. (2 puntos): Dado el siguiente código:

```
public class Ejercicio4Examen {
    public static void main(String[] args) {
        Thread hiloAltaPrioridad = new Thread(new Tarea(), "Hilo Alta Prioridad");
        Thread hiloBajaPrioridad = new Thread(new Tarea(), "Hilo Baja Prioridad");

        hiloAltaPrioridad.setPriority(Thread.MAX_PRIORITY);
        hiloBajaPrioridad.setPriority(Thread.MIN_PRIORITY);

        hiloBajaPrioridad.start();
        hiloAltaPrioridad.start();
    }

    static class Tarea implements Runnable {
        @Override
        public void run() {
            System.out.println(Thread.currentThread().getName() + " está ejecutándose.");
        }
    }
}
```

¿Qué salida producirá la ejecución de este código? ¿Por qué?

Justifica la respuesta

Ejercicio 4. (2 puntos): Revisa el archivo: Teoria4.java

¿Necesita alguna modificación o funciona correctamente? ¿Por qué?

Justifica la respuesta