

Examen PSP: Hilos, Sincronización y Sockets. Tipo 1.

- Entregar un proyecto de IntelliJ IDEA exportado en zip con **un módulo por cada uno de los ejercicios**.
- Asegúrate de que los proyectos **compilan**.
- Utiliza el **JDK 17**.

Ejercicio 1. Hilos (3 ptos)

Escribe un programa que mediante el uso de la clase `Thread` y la clase `URL` permita descargar **en paralelo** los siguientes ficheros:

- <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/InputStream.html>
- <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/OutputStream.html>
- <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/Flushable.html>
- <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/io/Closeable.html>

Consideraciones

1. Utiliza el **paquete** `com.[nombreapellido].psp.hilos`
2. Se deben descargar los ficheros utilizando las clases apropiadas para leer y escribir **en modo texto**.
3. La descarga debe realizarse en paralelo, no de forma secuencial.
4. Se deben **cerrar los recursos** utilizados.
5. Se debe gestionar correctamente el **ciclo de vida de los hilos**.
6. Se deben guardar los ficheros descargados en la carpeta `downloads` del usuario, cada fichero se guardará con el nombre del fichero html, por ejemplo, para el primer fichero será `InputStream.html` (puedes utilizar los métodos `substring` y `lastIndexOf` de la clase `String`).

Ejercicio 2. Sincronización (4 ptos)

Utiliza las clases `Thread` y `Semaphore` para escribir un programa que simule el funcionamiento de una máquina tragaperras (SlotMachine). La máquina tragaperras está compuesta de una serie de tambores (Drum) que giran durante un tiempo aleatorio y generan, cada uno, un valor aleatorio.

La máquina tragaperras debe esperar a que los tambores terminen de girar, obtiene el valor que ha generado cada tambor, los concatena y saca por pantalla el valor generado.

Los tambores, una vez han generado el valor aleatorio, deben esperar a que la máquina tragaperras les permita girar de nuevo.

Este ciclo se repite una serie predeterminada de veces.

Una posible salida del programa, en la que tenemos 5 tambores generando valores aleatorios desde el 0 al 5 y en el que se repite el juego 3 veces sería:

```
Slots value 10341
Slots value 15421
Slots value 25013
Drum done
Drum done
Drum done
Drum done
Drum done
Slot machine done
```

Programa las clases

- **Drum** en la que se puede configurar por constructor el tiempo máximo de giro y el valor máximo que genera.
- **SlotMachine** en la que se puede configurar el número de tambores, el tiempo máximo que estos giran, el valor máximo que estos generan y el número de veces que se repite el juego.
- Sincroniza estas clases mediante el uso de semáforos.
- Utiliza el **paquete** `com.[nombreapellido].psp.sincronizacion`

Ejercicio 3. Sockets (3 ptos)

Escribe un servidor **multihilo** que permita a un cliente consultar la hora o la fecha actuales.

- Cuando el servidor reciba un mensaje comenzando por **t**, debe responder con el resultado de llamar a `LocalTime.now()`
- Cuando el servidor reciba un mensaje comenzando por **d**, debe responder con el resultado de llamar a `LocalDate.now()`
- En cualquier otro caso, debe responder con la cadena `type t for time, d for date`

Escribe un cliente que permita interactuar de forma asíncrona con los datos que proporciona el usuario y con los datos recibidos desde el servidor.

Utiliza el **paquete** `com.[nombreapellido].psp.sockets`