

Programación de servicios y procesos

Práctica 1



NORMAS DE ENTREGA:

Día de activación: 10/12/2019

Plazo máximo de entrega: **19/01/2020, hasta las 23:55 horas**

FORMATO: Los enunciados se resolverán en proyectos independientes. Para ello utilizaremos el entorno NetBeans y el lenguaje Java.

La entrega se realizará en un documento comprimido conteniendo cada una de las carpetas de los diferentes proyectos. Las carpetas de los proyectos tendrán el nombre PracticaProgramacion1, PracticaProgramacion2... Si se adjuntara algún archivo explicativo se deben seguir las siguientes especificaciones: el texto justificado, letra verdana tamaño 11 para texto normal, letra verdana tamaño 14 para encabezados e interlineado 1,15.

El nombre del documento será **ANEXO_PSP_P1_Nombre_Apellido1_Apellido2.extensión** y se incluirá dentro del archivo comprimido.

El entregable cumplirá con el siguiente formato: **PSP_P1_Nombre_Apellido1_Apellido2.rar**

No se corregirán ni serán tenidas en cuenta aquellas prácticas que no estén entregadas en tiempo y forma indicados.

OBSERVACIONES: El profesorado únicamente resolverá las dudas que se notifiquen en la plataforma con 24 horas de antelación a la fecha y hora de entrega de la misma. Las dudas que se planteen fuera de ese plazo no podrán ser resueltas.

Programación de servicios y procesos

Práctica 1

ENUNCIADO 1: Se pretende realizar un programa el cual tiene un menú que ofrecerá una serie de opciones para controlar los hilos. En su arranque el programa creará tres ficheros. Cada fichero creará un fichero diferente Fichero1.txt, Fichero2.txt y Fichero3.txt. Se valorará que los hilos tengan nombre y prioridad asignada para poder crear primero el Fichero1.txt, luego Fichero2.txt y por último la Fichero3.txt todos en la misma carpeta del proyecto, y saquen un mensaje cada uno cuando acabe su tarea. Además de crear los ficheros cada hilo debe de escribir 100 líneas la frase "Soy el hilo nombre_del_hilo, mi prioridad es prioridad_hilo y estoy escribiendo el fichero nombre_fichero".

Una vez creados los ficheros y los datos se ejecutarán los siguientes procesos desde un menú:

- 1) Demonio que cuenta líneas. Este será el primer hilo que podemos lanzar, será un demonio que muestre cada 10 segundos un mensaje con el número de líneas que tiene cada fichero. Para ello estableceremos la propiedad del hilo `setDaemon(true)` y lo controlaremos con un bucle `do while`.
- 2) Demonio control inserciones. Hacer un segundo hilo basado en el primero. Este tendrá que controlar el número de líneas de Fichero1.txt y sacar un mensaje que indique cuando se ha insertado una nueva línea en el fichero. Se valorará que los hilos del primer apartado y el de este puedan ejecutarse simultáneamente.
- 3) Operaciones sincronizadas. Crear un sistema sincronizado de bloqueo y desbloqueo del código de un método que programaremos a parte y que se utilizará solo para este apartado. El metodo quedará bloqueado por un hilo y cuando se quiera realizar la escritura otro deberá desbloquearlo. Los hilos se lanzarán desde distintas opciones del menú. Cuando se desbloquee el método el fichero elegido para escribir una línea en él será Fichero1.txt.

ENUNCIADO 2: Se pretende realizar un programa servidor que nos permita gestionar las compras de billetes de tren. El servidor creará una matriz de 9x4 simulando un vagón. Los diferentes asientos estarán identificados por la fila y la columna en la que se encuentran. El servidor, una vez conectado el cliente, ofrecerá dos opciones, una abrir un chat para hablar con el servicio técnico y la otra la de reservar billetes.

La opción para reservar billete se hará de la siguiente manera. El servidor enviará una lista con cadenas, cada cadena almacenará la posición de un asiento en el formato 1,2 o 4,1 y se imprimirán todas en el cliente, si el cliente confirma una plaza escribiendo sus coordenadas el servidor hará la reserva y enviará una confirmación al cliente finalizando la comunicación después de mostrar el mensaje. Si el cliente escribe mal la plaza se volverán a mostrar las opciones del menú. En caso de estar el vagón lleno, el servidor enviará un mensaje de ocupado y finalizará la comunicación.

El servidor tiene que poder aceptar peticiones de varios clientes simultáneamente. Para atender la conexión de un cliente, será necesario la creación de un hilo para que tras el fin de la comunicación el servidor pueda continuar escuchando peticiones.

El servidor ha de ir guardando un log de las acciones del cliente que tiene conectado, guardará uno por cada cliente que tenga conectado y si el cliente envía la cadena /log desde su canal el servidor enviara el fichero de log. Esta última acción del programa es opcional, no será necesario programarla para superar la práctica.