

Programación de servicios y procesos - 2324

[Área personal](#) ▶ [Mis cursos](#) ▶ [FORMACIÓN PROFESIONAL 23/24](#) ▶ [CIFP CARLOS III](#) ▶
[CGS Desarrollo de Aplicaciones Multiplataforma 23/24](#) ▶ [Segundo\(DAM\)](#) ▶
[Programación de servicios y procesos - 2324](#) ▶ [1ª CONVOCATORIA](#) ▶
[Tarea para PSP06. \(hasta el 11/02/24 23:55\).](#)

Tarea para PSP06. (hasta el 11/02/24 23:55)

Tarea para PSP06.

Detalles de la tarea de esta unidad.

Enunciado.

Ejercicio 1. Expresiones regulares y Log4J

Tu programa debe realizar lo siguiente. Todo debe **validarse mediante expresiones regulares**:

- Solicita un nombre de fichero que termina en .txt o .csv que empiece por una letra minúscula y a continuación tenga como máximo 6 caracteres (mayúsculas, minúsculas, números o guión bajo _).
 - Válido: f.txt, fl.csv, fich_1.csv
 - No válido: f.pdf, mifichero.txt, Fich.csv
- En caso de que sea válido, abre el fichero sin comprobar si existe o no. Si el fichero no existe debes recoger la excepción y mostrarla por pantalla (más adelante te diré que la guardes también en el log). La excepción es FileNotFoundException.
- Si hemos podido abrir el fichero leeremos la información que contiene, en cada línea debe aparecer un dato en el siguiente orden que debes validar:
 - Primera línea: Clave de longitud de 6. Los 3 primeros caracteres son mayúsculas o minúsculas y los 3 siguientes números del 0 al 9.
 - Válido: mab123, Mab456, cLA123
 - No válido: miclave1, unaclavelarga123, otra_123
 - Segunda línea: Código que empieza por una letra A, B, C seguido de un guión -, y a continuación 5 números del 1 al 9.
 - Válido: A-12345, B-59991
 - No válido: D-00000, A-1, A.12345
 - Tercera línea: Un código postal español teniendo en cuenta que el código postal "más alto" debería ser 52999. Es decir, no existe un código postal en España que empiece por 53 o superior.
 - Válido: 00000, 30001
 - No válido: 54000, 99999

El programa debe tener un log utilizando la **librería Log4J**.

Este log se guardará en un fichero llamado log.html en formato HTML. Deberá generarse uno nuevo cada vez que se ejecute el programa.

Prepararemos en el programa los mensajes para guardar en el log lo siguiente:

- Error: Información de las excepciones.
- Info: Dato introducido por el usuario y las validaciones correctas.
- Warning: Validaciones incorrectas de los datos leídos.

Configuraremos el log para que únicamente muestre los mensajes a partir del nivel de Info.

Entrega un manual con diferentes ejecuciones del programa. Te adjunto dos ficheros, uno con todas las líneas válidas, y otro con todas las líneas inválidas.

Ejercicio 2. Firma y verificación del fichero .jar

Comienza este programa cuando termines el anterior. Lo que tienes que entregar de este ejercicio es un documento con todos los pasos realizados.

Genera el .jar con la librería de Log4J empaquetada. Tienes un documento en el aula virtual ([Generar un fichero .jar con dependencias](#)) dónde explico cómo hacer esto. Una vez generado el .jar comprueba que tu programa sigue funcionando correctamente.

Firma el fichero .jar.

Valida la firma del fichero firmado.

Ejecuta el fichero firmado.

Los pasos a seguir son los que vienen explicados en el documento "[Guía para firmar y verificar la firma de un fichero .jar](#)". Tienes también un video explicativo: "[Video Explicación de la firma y verificación de un .jar](#)".

El alias y nombre del certificado son las siglas de tu nombre y apellidos, en mi caso sería mjcp.

Cuando ejecutes el jar firmado, con un pantallazo dónde le introduces un nombre de fichero inválido y termina es suficiente. No tienes que volver a explicarlo todo. Es suficiente con ver que funciona.

Incluye el .jar firmado y sin firmar en la entrega.

Utiliza jdk 17.

Criterios de puntuación. Total 10 puntos.

Total 10 puntos.

Ejercicio 1. 6 puntos.

Ejercicio 2. 4 puntos.

Recursos necesarios para realizar la Tarea.

Ejercicio 1.- Sobre todo:

[Resumen Expresiones Regulares](#)

[Cheat Sheet Expresiones regulares](#)

[Proyecto Java Expresiones Regulares](#)

[Apuntes Resumen Log4J](#)

[Proyecto Log4J 2](#)

Ejercicio 2.- Sobre todo:

[Generar un fichero .jar con dependencias](#)

[Guía para firmar y verificar la firma de un fichero .jar](#)

[Video Explicación de la firma y verificación de un .jar](#)

Consejos y recomendaciones.

Lee la documentación y visualiza los vídeos.

Indicaciones de entrega.

Lo que debes entregar:

- Proyecto maven que pueda abrirse y ejecutarse tanto en IntelliJ o NetBeans bien comentado (incluye tus propios comentarios, no los de los proyectos de ejemplo). No es necesario Javadoc.
- Manual pdf para los dos ejercicios según lo indicado en el enunciado.
- Fichero .jar sin firmar.
- Fichero .jar firmado.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la quinta unidad del MP de PSP**, debería nombrar esta tarea como...

sanchez_manas_begona_PSP05_Tarea

Fecha de entrega: 11 de Febrero de 2024 a las 23:55

Fecha límite de entrega: 18 de Febrero de 2024 a las 23:55

 [f.txt](#)

29 de enero de 2024, 01:00

 [fl.txt](#)

29 de enero de 2024, 01:00

Estado de la entrega

Estado de la entrega	No entregado
Estado de la calificación	Sin calificar
Fecha de entrega	domingo, 11 de febrero de 2024, 23:55
Tiempo restante	9 horas 15 minutos
Última modificación	-
Comentarios de la entrega	+ Comentarios (0)

Agregar entrega

Todavía no has realizado una entrega.

Actividad previa

[◀ Tarea para PSP05. \(hasta el 28/01/24 23:55\)](#)

Ir a...

Siguiente actividad

[Tarea para PSP07 \(hasta el 29/02/24 23:55\) ▶](#)

Mantente en contacto

Dirección General de Formación Profesional e Innovación

[📁 Resumen de retención de datos](#)

[📱 Descargar la app para dispositivos móviles](#)