

# Tarea 7 – Ficheros de configuración

## Enunciado.

### 1. Responde a las siguientes preguntas relacionadas con los ficheros .properties:

#### a. ¿Por qué se utilizan los ficheros de propiedades o properties?

- Separación de código y configuración: Cambiar configuraciones sin cambiar el código fuente.
- Facilidad de mantenimiento: Más fácil modificar configuraciones en un archivo que en el código.
- Flexibilidad: Permite tener diferentes configuraciones para distintos entornos (desarrollo, pruebas, producción).

#### b. ¿Qué datos deben ser almacenados en un fichero de propiedades?

Se guarda la clave por la que filtraremos para buscar y el valor de esta (clave=valor). Por ejemplo:

Puerto=4200

#### c. ¿Qué datos no deben ser almacenados en un fichero properties?

Cualquier dato común que no deba estar en un archivo de configuración , aparte que no tenga el formato clave=valor por supuesto.

**d. Hemos visto en la teoría que el método `getProperty()` siempre devuelve un string. Imagina que tengo una propiedad llamada “`numUsersAllowed=5`”. Con el método `getProperty(“numUsersAllowed”)` obtendría un string con un “5”. ¿Cómo puedo hacerlo para transformar ese 5 en un entero?**

Pues tendríamos que usar `parseInt` (`Integer.parseInt`) para convertirlo en número con el contenido “valor” de la clave `numUsersAllowed`.

#### e. Además de ficheros de configuración tipo texto plano, ¿existe algún otro tipo?

Si, también existen los binarios que almacenan datos codificados y los archivos de texto estructurado como XML, JSON, etc...

**f. Dentro de la documentación de Java para la clase `Properties` (<https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html>) vemos que hay**

**dos métodos load: load(InputStream inStream) load(Reader reader) ¿Cuándo usaría uno u otro? Razona tu respuesta.**

Usaría load(InputStream) cuando el archivo .properties esté en la codificación antigua ISO-8859-1 (la que Java usaba por defecto) y los caracteres especiales estén escritos con \uXXXX.

En cambio, usaría load(Reader) cuando el archivo esté en UTF-8 (o cualquier otra codificación moderna), ya que el Reader me permite especificar esa codificación y leer correctamente los caracteres.

2. Crea un programa que cree, si no existe (la primera vez que se lanza la aplicación el fichero no existe), utilizando las clases que hemos visto hasta ahora y la clase Properties el siguiente fichero llamado application.properties.

El programa además cargará el fichero una sola vez y quedará cacheado para futuras referencias al mismo (el fichero de configuración se leerá y cargará en memoria solo una vez durante la ejecución del programa). La clase que carga el fichero .properties debe tener:

- a. Un método que devuelva el tamaño máximo de fichero permitido de servidor y se pueda llamar desde cualquier punto de la aplicación, sin tener que volver a leer el fichero, ya que está cacheado.
- b. Lo mismo que el punto anterior pero que devuelva si el servidor se debe lanzar o no (launcher).
- c. Un método que te permite cambiar cualquier propiedad en el fichero de configuración y añade un comentario cada vez que se modifica.
- d. Un método que muestre todas las propiedades que tiene el fichero y las saque al final de la ejecución del programa una vez ya se haya guardado el fichero.