## Actividad 1.7: Tratamiento ficheros Formación. Uso de fichero ”properties”

## A. Investigación sobre el uso de los ficheros “properties” de configuración.

• Si se tienen muchos elementos en nuestro código que tiene valores constantes (rutas, separadores CSV, credenciales de base de datos claves de API, etc.) ¿investiga 4 formas distintas de gestionar estos elementos con valores constantes para un proyecto java “puro” (no es parte de un framework, ni carga dependencias como Maven/Gradle)?

• ¿Qué ventajas tiene usar un archivo .properties frente a escribir valores directamente como constantes en el código fuente?

- Mantenimiento y flexibilidad
- Internacionalización y Localización
- Seguridad y Gestión de Entornos
- Trabajo en Equipo y Roles

• ¿Qué tipo de valores son adecuados para guardarse en un archivo .properties, y cuáles no? ¿Por qué?

Valores adecuados serian parámetros de configuración general y de entorno que no son sensibles

• ¿Qué limitaciones tiene el uso de archivos .properties y qué otras alternativas modernas existen (por ejemplo, YAML, JSON, bases de datos o Spring @Value)?

- Falta de estructura jerárquica
- Soporte limitado para colecciones
- Problemas históricos de codificación
- Ausencia de tipos de datos

YAML: Sustituye a properties en entornos como Spring Boot
JSON:  Se usa cuando la configuracion es generada o leida automaticamente por otros sistemas o APIs
Bases de datos: Se usa en arquitecturas de microservicios

• ¿Cómo cargar desde Java un archivo .properties y acceder a sus valores? ¿Qué ocurre si una clave no existe?

Se usa la clase java.io.Properties con un InputStream