Actividad 1.1: Configuración entorno y Tratamiento fichero.txt

A. Investigación sobre clases del paquete java.io

• ¿Para qué se utilizan las siguientes clases?
File -> Representa un archivo o directorio y permite consultar o modificar sus propiedades No lee ni escribe directamente el contenido.
FileReader -> Permite leer caracteres de un archivo de texto.
BufferedReader -> Añade un búfer a la lectura y permite leer texto de forma más eficiente, especialmente línea por línea mediante `readLine()`.
FileWriter -> Permite escribir caracteres en un archivo de texto.
BufferedWriter -> Añade un **búfer** a la escritura para hacerla más eficiente y permite escribir texto y saltos de línea.
PrintWriter -> Facilita la escritura de texto con formato, utilizando métodos como `print()`, `println()` y `printf()`.

• ¿Qué diferencia hay entre FileReader y BufferedReader? ¿Cuál es más eficiente en el tratamiento de información? ¿Por qué?

`FileReader`: lee directamente los caracteres de un archivo.
`BufferedReader`: utiliza un búfer para almacenar temporalmente los datos, haciendo la lectura más eficiente y permitiendo leer fácilmente línea por línea con `readLine()`.

`BufferedReader` es más eficiente, porque reduce el número de accesos directos al archivo al leer los datos en bloques mediante el búfer.

• Entre FileWriter y PrintWriter ¿Cuál de las dos clases lanza excepciones de tipo IOException?¿Cómo gestionará los errores la otra clase? ¿Cuál de las dos clases usarías para añadir/escribir texto al final de un fichero?

`FileWriter` puede lanzar excepciones de tipo `IOException`, por ejemplo, si no se puede abrir o escribir en el archivo.
`PrintWriter` no lanza `IOException` directamente. En caso de error, establece un estado interno que se puede comprobar con `checkError()`.
Para añadir texto al final de un fichero, usaría `FileWriter` con `append = true`, normalmente combinado con `PrintWriter`