Actividad 1.8: Tratamiento ficheros CVS – DAO properties – java.nio

A. Investigación sobre las clases del paquete java.nio.

Tratamiento de Ficheros utilizando las clases del paquete java.nio (en vez de las del paquete java.io): ¿Cuáles son las principales diferencias con las clases del paquete java.io para el tratamiento de ficheros?

Qué clases JAVA/Métodos están involucradas si queremos:

• Crear un fichero/directorio.
• Borrar un fichero/directorio.
• Renombrar un fichero/directorio.
• Mover un fichero/directorio.
• Abrir un fichero/directorio.
• Indicar que un fichero utilice un juego de caracteres concreto (asociado a la clase Charset)para un fichero (por ejemplo ISO8859-1) o uno de los juegos éstandard (ver clase StandardCharsets)
• Leer todas las líneas de un fichero (como lista de Strings).
• Escribir una línea, o líneas en un fichero.
• Añadir información a un fichero ya existente.
• Cerrar fichero

Respuesta:

Aunque se suelen comparar como dos alternativas para leer y escribir ficheros, conviene distinguir sus componentes.

Característica	**java.io**	**java.nio / java.nio.file**
Representación de rutas	File	Path
Operaciones sobre ficheros	Métodos de File y flujos de entrada/salida	Métodos estáticos de Files
Lectura de texto	FileReader, BufferedReader	Files.readAllLines(), Files.newBufferedReader()
Escritura de texto	FileWriter, BufferedWriter	Files.write(), Files.newBufferedWriter()
Codificación de caracteres	Se puede especificar mediante ciertas clases y constructores	Se especifica mediante Charset o StandardCharsets
Movimiento y sustitución de ficheros	renameTo() tiene limitaciones	Files.move() permite opciones como REPLACE_EXISTING
Otras capacidades	Flujos de bytes y caracteres	También ofrece canales, buffers y operaciones de E/S no bloqueante

Las clases JAVA/Métodos involucrados son

Operación solicitada	**Clase y método de NIO**	**Ejemplo**
Crear un fichero	Files.createFile()	Files.createFile(path);
Crear un directorio	Files.createDirectory()	Files.createDirectory(path);
Crear directorios y sus padres	Files.createDirectories()	Files.createDirectories(path);
Borrar un fichero o directorio vacío	Files.delete()	Files.delete(path);
Borrar si existe	Files.deleteIfExists()	Files.deleteIfExists(path);
Renombrar o mover	Files.move()	Files.move(origen, destino);
Abrir un fichero para lectura	Files.newBufferedReader()	Files.newBufferedReader(path, charset);
Abrir un fichero para escritura	Files.newBufferedWriter()	Files.newBufferedWriter(path, charset);
Abrir un directorio	Files.newDirectoryStream()	Files.newDirectoryStream(path);
Indicar una codificación	Charset / StandardCharsets	StandardCharsets.UTF_8
Leer todas las líneas	Files.readAllLines()	Files.readAllLines(path, charset);
Escribir líneas	Files.write()	Files.write(path, lineas, charset);
Añadir información al final	Files.write() con APPEND	Files.write(path, lineas, charset, StandardOpenOption.APPEND);
Cerrar recursos abiertos	close() o try-with-resources	try (...) { ... }