

Manejo de ficheros 1  
Sergio David Vico Pineda  
David Pérez Pastor

Busca los métodos estudiados en clase, encuentra sus métodos y explica con las mínimas palabras posibles cada método:

- File
- FileReader
- FileWriter
- BuffererdReader
- BufferedWriter

## **File**

.canRead() -> Devuelve True o False si podría leer o no el archivo.

.canWrite() -> Devuelve True o False si podría escribir o no el archivo.

.createNewFile() -> Intenta crear un archivo, si lo consigue devuelve True si no False.

.delete() -> Intenta eliminar un archivo, si lo consigue devuelve True si no False.

.exists() -> Comprueba si un archivo existe, si existe devuelve True si no False.

.getName() -> Devuelve un String con el nombre del archivo.

.getAbsolutePath() -> Devuelve un String con la ruta absoluta del archivo.

.length() -> Devuelve un número (Long) con el tamaño del archivo en bytes.

.list() -> Devuelve un Array de Strings con los nombres de los archivos en un directorio.

.mkdir() -> Intenta crear una carpeta, si lo consigue devuelve True si no False.

## **FileReader**

Esta clase obtiene varios de sus métodos a través de herencia de Reader

.close() -> Para cerrar la instancia.

.getEncoding() -> Devuelve un String con el código de codificación del archivo.

.read() -> Lee un carácter del archivo, devuelve un Int con su código ASCII y devuelve '-1' cuando termina la línea.

.ready() -> Devuelve True o False si el archivo está listo para ser leído.

.skip(Long n) -> Se le pasa un numero de caracteres para ser omitidos y devuelve el número de caracteres omitidos.

Hay más métodos pero no los he entendido.

**FileWriter**

.write() -> Escribe lo que se le ha pasado por parámetros, y no devuelve nada.

.close() -> Para cerrar la instancia.

.getEncoding() -> Devuelve un String con el código de codificación que se va a escribir en el archivo.

Heredados

.append() -> Añade al final de un archivo existente lo que se le ha pasado por parámetros y no devuelve nada.

**BufferedReader**

.read() -> Lee un carácter del archivo, devuelve un Int con su código ASCII y devuelve '-1' cuando termina la línea.

.readLine() -> Lee una línea completa, y devuelve un String con el contenido si no hay más líneas devuelve null.

.ready() -> Devuelve True o False si el archivo está listo para ser leído.

.close() -> Para cerrar la instancia.

.skip(Long n) -> Se le pasa un numero de caracteres para ser omitidos y devuelve el número de caracteres omitidos.

Hay más métodos pero no los he entendido.

**BufferedWriter**

.write() -> Escribe lo que se le ha pasado por parámetros, y no devuelve nada.

.newLine() -> Escribe un salto de línea, y no devuelve nada.

.close() -> Para cerrar la instancia.

---

[File \(Java Platform SE 8 \) \(oracle.com\)](#)

[FileWriter \(Java Platform SE 8 \) \(oracle.com\)](#)

[FileReader \(Java Platform SE 8 \) \(oracle.com\)](#)

[BufferedReader \(Java Platform SE 8 \) \(oracle.com\)](#)

[BufferedWriter \(Java Platform SE 8 \) \(oracle.com\)](#)