

Lectura y escritura de archivos

Hay diferentes formas de leer y escribir archivos de texto en Java, entre las más conocidas está `BufferedReader/BufferWriter` y `Scanner`.

En el caso de lectura de archivos, lo ideal es utilizar `BufferedReader` ya que es más eficiente, es síncrono y trabaja bien con multiprocesos. Asimismo, cuenta con una memoria de buffer mayor.

Implementación

1. Para crear archivos se puede utilizar la clase **`FileWriter`**, que permite escribir archivos utilizando la codificación de caracteres estándar que utilice el sistema operativo. Para construirlo, solo es necesario agregar la ruta que tendrá el archivo.
2. Para escribir el texto se utilizará **`BufferedWriter`**. Para inicializarlo es necesario pasar como parámetro un objeto de tipo **`FileWriter`**. Ya que esté el objeto, se pueden utilizar sus diferentes métodos para escribir, en este caso como ejemplo, solo se va a agregar una cadena de caracteres, por lo tanto se utiliza el método **`write()`**.
3. Para leer el archivo, la opción óptima es **`BufferedReader`** ya que es la más eficiente. Para crear un objeto de este tipo, primero que nada se debe crear una instancia de **`FileReader`**, que básicamente es el mismo proceso de **`FileWriter`**. Luego, este objeto se pasa como parámetro en el constructor de `BufferedReader` y ya se tiene instancia. Con este objeto se pueden leer línea por línea los elementos del archivo utilizando el método **`readLine()`**.