



TSU en Software

Práctica 7: Escritura y lectura de archivos en Java

Desarrollo de Aplicaciones II

Carlos Araiza Dionicio

12-02-2021

Descripción

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, síncrono y trabaja bien con multiprocesos. Asimismo, cuenta con una memoria de buffer mayor a otros métodos.

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()`**.

Código

Código fuente:

<https://github.com/iCharlieAraiza/DDAII/tree/main/Pr%C3%A1ctica%207/out/production/Pr%C3%A1ctica%207/com/coding>

LectorArchivos.java

```
package com.coding;

import java.io.*;

public class LectorArchivos {

    String url;

    public LectorArchivos(String url) throws IOException {

        this.url = url;
    }

    public void crear(String texto) throws IOException {

        BufferedWriter bw = new BufferedWriter(new FileWriter(url) );

        bw.write(texto);

        bw.close();

    }

    public String leer() throws IOException {

        StringBuilder texto = new StringBuilder();

        String s;

        BufferedReader br = new BufferedReader( new FileReader(url) );

        while( (s = br.readLine()) != null){

            texto.append(s);

        }

        br.close();

        return texto.toString();

    }

}
```

Main.java

```
public static void main(String[] args) throws IOException {  
  
    LectorArchivos archivo = new LectorArchivos("./texto.txt");  
    archivo.crear("Hola! Esta es una prueba para comprobar que funciona. Segunda línea");  
    System.out.println( "Leer archivo: " + archivo.leer() );  
  
}
```

Referencias

- Java File I/O (Reading & Writing). (2020, 21 julio). [Vídeo]. YouTube.
https://www.youtube.com/watch?v=hgF21imQ_Is
- Java BufferedWriter Class - javatpoint. (2020). www.javatpoint.com.
<https://www.javatpoint.com/java-bufferedwriter-class>