

Nombre

UNIDAD 8: Gestión de archivos

RA: 2

1. RA2 [b]

El objetivo de esta práctica es explorar y entender las diferencias fundamentales entre el manejo de ficheros de texto y ficheros binarios en el contexto de Java. Para ello, crearemos un programa que trabaje con información relacionada con el Día de la Paz.

Parte 1: Ficheros de Texto (PazTexto.java)

Crea una clase llamada PazTexto. Implementa un método para escribir información en un fichero de texto. Esta información podría incluir mensajes inspiradores o citas célebres relacionadas con la paz.

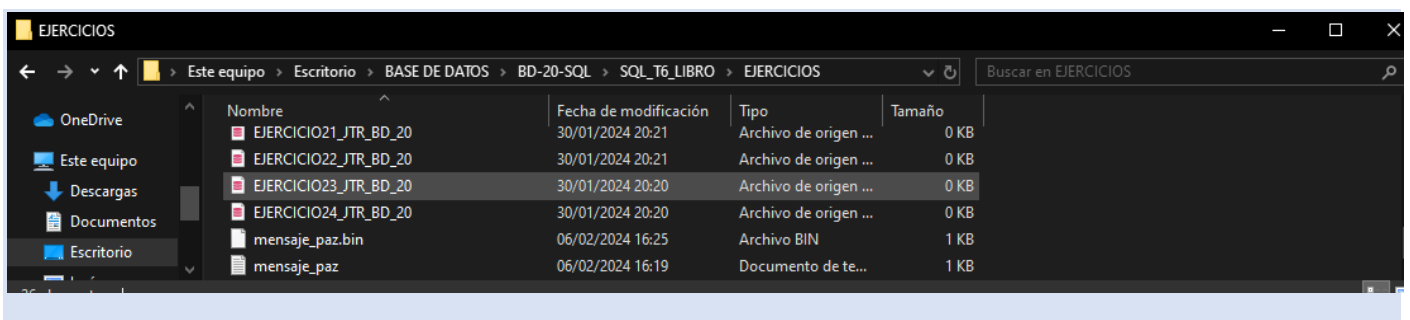
Implementa otro método para leer la información del fichero de texto y mostrarla en la consola.

Parte 2: Ficheros Binarios (PazBinario.java)

Crea una clase llamada PazBinario que haga lo mismo que la clase PazTexto, pero utilizando ficheros binarios en lugar de ficheros de texto. Utiliza `ObjectOutputStream` y `ObjectInputStream` para escribir y leer.

Incluye en ambas clases un método `main()` para probar las funcionalidades.

Prueba a escribir y leer información de las dos maneras, trata también de abrir los ficheros generados mediante el bloc de notas. Incluye capturas que muestren lo que has hecho y comenta brevemente las diferencias que has observado.



The screenshot shows an IDE with two windows. The left window, titled 'EJERCICIOS', displays the source code for 'PazTexto.java'. The code includes imports for Java I/O classes and implements two static methods: 'escribirEnArchivo' and 'leerArchivo'. The 'main' method initializes a message and filename, calls 'escribirEnArchivo', and then 'leerArchivo'. The right window, titled 'mensaje.paz: Bloc de notas', shows a text editor with the message: 'La paz comienza con una sonrisa. Donde hay amor, hay vida.'

```
1 import java.io.BufferedReader;
2 import java.io.BufferedWriter;
3 import java.io.FileReader;
4 import java.io.FileWriter;
5 import java.io.IOException;
6
7 public class PazTexto {
8
9     // Método para escribir información en un archivo de texto
10    public static void escribirEnArchivo(String mensaje, String nombreArchivo) {
11        try (BufferedWriter writer = new BufferedWriter(new FileWriter(nombreArchivo))) {
12            writer.write(mensaje);
13            System.out.println("Mensaje escrito en el archivo correctamente.");
14        } catch (IOException e) {
15            System.err.println("Error al escribir en el archivo: " + e.getMessage());
16        }
17    }
18
19    // Método para leer la información del archivo de texto y mostrarla en la consola
20    public static void leerArchivo(String nombreArchivo) {
21        try (BufferedReader reader = new BufferedReader(new FileReader(nombreArchivo))) {
22            String linea;
23            System.out.println("Contenido del archivo:");
24            while ((linea = reader.readLine()) != null) {
25                System.out.println(linea);
26            }
27        } catch (IOException e) {
28            System.err.println("Error al leer el archivo: " + e.getMessage());
29        }
30    }
31
32    public static void main(String[] args) {
33        String mensaje = "La paz comienza con una sonrisa.\nDonde hay amor, hay vida.\n";
34        String nombreArchivo = "mensaje_paz.txt";
35
36        // Escribir el mensaje en el archivo
37        escribirEnArchivo(mensaje, nombreArchivo);
38
39        // Leer y mostrar el contenido del archivo
40        leerArchivo(nombreArchivo);
41    }
42 }
```

The screenshot shows an IDE with two windows. The left window, titled 'EJERCICIOS', displays the source code for 'PazBinario.java'. The code includes imports for Java I/O classes and implements two static methods: 'escribirEnArchivo' and 'leerArchivo'. The 'main' method initializes a list of messages and a filename, calls 'escribirEnArchivo', and then 'leerArchivo'. The right window, titled 'mensaje.paz.bin', shows a hex editor view of the binary file. The first line of the hex editor shows the ASCII representation of the message: 'La paz comier'.

```
1 // Método para escribir información en un archivo binario
2 public static void escribirEnArchivo(List<String> mensajes, String nombreArchivo) {
3     try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(nombreArchivo))) {
4         out.writeObject(mensajes);
5         System.out.println("Mensajes escritos en el archivo binario correctamente.");
6     } catch (IOException e) {
7         System.err.println("Error al escribir en el archivo binario: " + e.getMessage());
8     }
9 }
10
11 // Método para leer la información del archivo binario y mostrarla en la consola
12 public static void leerArchivo(String nombreArchivo) {
13     try (ObjectInputStream in = new ObjectInputStream(new FileInputStream(nombreArchivo))) {
14         List<String> mensajes = (List<String>) in.readObject();
15         System.out.println("Contenido del archivo binario:");
16         for (String mensaje : mensajes) {
17             System.out.println(mensaje);
18         }
19     } catch (IOException | ClassNotFoundException e) {
20         System.err.println("Error al leer el archivo binario: " + e.getMessage());
21     }
22 }
23
24 public static void main(String[] args) {
25     List<String> mensajes = new ArrayList<>();
26     mensajes.add("La paz comienza con una sonrisa.");
27     mensajes.add("Donde hay amor, hay vida.");
28     String nombreArchivo = "mensaje_paz.bin";
29
30     // Escribir los mensajes en el archivo binario
31     escribirEnArchivo(mensajes, nombreArchivo);
32
33     // Leer y mostrar el contenido del archivo binario
34     leerArchivo(nombreArchivo);
35 }
```

La gran diferencia es que en la de texto yo si lo puedo entender y en el binario como se guardan en bytes pues los seres humanos no lo pueden entender dado que la máquina no lo interpreta en lenguaje humano sino en binario

Puedes consultar, por ejemplo, aquí:

https://chuidiang.org/index.php?title=Lectura_y_Escritura_de_Ficheros_en_Java

INSTRUCCIONES DE ENTREGA

Una vez terminado el documento, guárdalo/expórtalo como PDF y adjúntalo en la entrega de la tarea.

RÚBRICA

Cada criterio se evalúa sobre 1 punto para calificar la entrega sobre 4.

La tarea aporta al criterio de evaluación señalado una puntuación máxima de 1.

La puntuación de cada respuesta se establece en función de la siguiente rúbrica:

Criterio	Descripción	Puntuación
Programa texto	Incluye captura y es correcta.	1
	La captura incluida es parcialmente completa.	0.5
	La captura es incorrecta o no se aporta.	0
Programa binario	Incluye captura y es correcta.	1
	La captura incluida es parcialmente completa.	0.5
	La captura es incorrecta o no se aporta.	0
Documentación	Incluye capturas ordenadas y completas	1
	Las capturas son parcialmente completas o desordenadas.	0.5
	No hay capturas o no resuelven la pregunta.	0
Explicaciones	Incluye explicación	1
	La respuesta incluida es parcialmente completa.	0.5
	La respuesta es incorrecta o no aborda la pregunta.	0

Entrega tardía: actividad planteada para su realización en clase. Los días de retraso pueden tener una incidencia en la calificación de hasta un 10%: $\text{calificación} * [(9 + \text{días_restantes} / \text{días_plazo}) * 10]$
