

FICHEROS

Si trabajamos con mucho texto”ficheros de texto muy grandes” trabajamos con buffer:

```
public class ArchivoServicio {  
  
    public void crearArchivo(String nombre) {  
        File archivo = new File(nombre);  
  
        try (  
            FileWriter fw = new FileWriter(archivo, true);  
            BufferedWriter buffer = new BufferedWriter(fw);  
        ) {  
  
            buffer.append("Hola que tal amigos!\n")  
                .append("Todo bien? yo escribiendo en un archivo...\n")  
                .append("Hasta luego Lucas!\n");  
            // buffer.close();  
            System.out.println("El archivo se ha creado con éxito!");  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

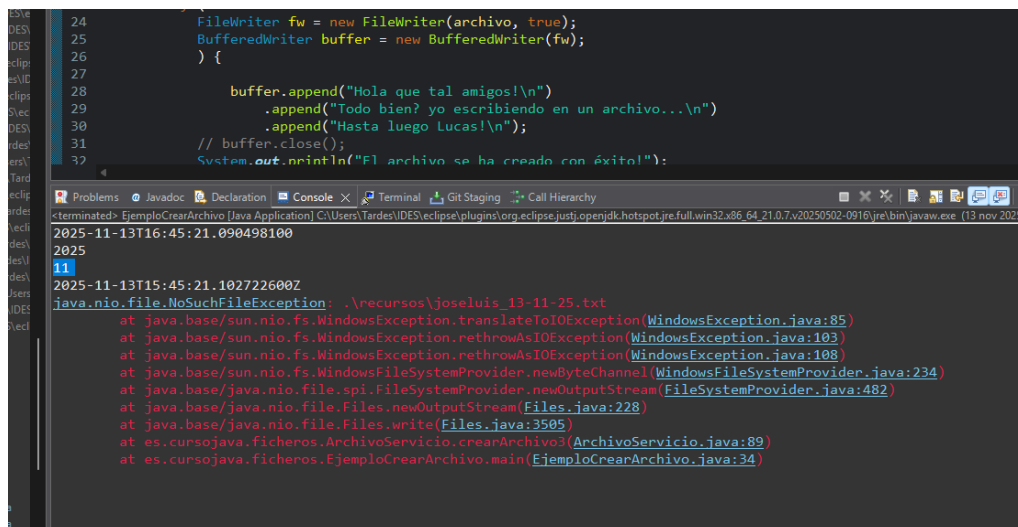
esto permite madar al fichero el texto por trozos, para no colapsar.

Tambien señalar que estoy usando try whit resources....

puedo meter en el try porque las dos clases implementa la interfaz closeable....

buffer.append añade al buffer los strings

PILA DE EXCEPCIONES: nos da la inf de las mierdas que están fucking mal, bro.



```
24     FileWriter fw = new FileWriter(archivo, true);  
25     BufferedWriter buffer = new BufferedWriter(fw);  
26     ) {  
27  
28         buffer.append("Hola que tal amigos!\n")  
29             .append("Todo bien? yo escribiendo en un archivo...\n")  
30             .append("Hasta luego Lucas!\n");  
31         // buffer.close();  
32         System.out.println("El archivo se ha creado con éxito!");  
33     }  
34 }
```

2025-11-13T16:45:21.090498100
2025-11-13T15:45:21.102722600Z
java.nio.file.NoSuchFileException: .\recursos\joseluis_13-11-25.txt
at java.base/sun.nio.fs.WindowsException.translateToIOException(WindowsException.java:85)
at java.base/sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:103)
at java.base/sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:108)
at java.base/sun.nio.fs.WindowsFileSystemProvider.newByteChannel(WindowsFileSystemProvider.java:234)
at java.base/java.nio.file.spi.FileSystemProvider.newOutputStream(FileSystemProvider.java:482)
at java.base/java.nio.file.Files.newOutputStream(Files.java:228)
at java.base/java.nio.file.Files.write(Files.java:3505)
at es.cursojava.ficheros.ArchivoServicio.crearArchivo3(ArchivoServicio.java:89)
at es.cursojava.ficheros.EjemploCrearArchivo.main(EjemploCrearArchivo.java:34)

A NOSOTROS LOS Q NOS VALEN SON LO QUE DICEN ES.CURSOJAVA.FICHEROS.....

OJO PARA QUE CREE UN FICHERO EL DIRECTORIO TIENE QUE EXISTIR EN RECUSOS EN MI PROYECTO.

METODO 1

Archivo.createNewFile no vale de nada sin el añadir el direcc o tenerlo añadido.

```
17
18 //nombre_yyyymmdd.txt
19 public class ArchivoServicio {
20
21     public void crearArchivo(String nombre) {
22         File archivo = new File(nombre);
23         System.out.println("si no existe.. lo creo...");
24         if (!archivo.exists()) {
25             try {
26                 archivo.createNewFile();
27             } catch (IOException e) {
28                 // TODO Auto-generated catch block
29                 e.printStackTrace();
30             }
31         }
32
33         try (FileWriter fw = new FileWriter(archivo, true); BufferedWriter buf
34
35             buffer.append("Hola que tal amigos!\n").append("Todo bien? yo escr
36                 .append("Hasta luego lucas!\n");
```

Problems Javadoc Declaration Console X Terminal Git Staging Call Hierarchy

<terminated> EjemploCrearArchivo [Java Application] C:\Users\Tardes\IDES\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86

2025-11-13T17:24:54.915464600
2025
11
2025-11-13T16:24:54.926535100Z
su no existe.. lo creo...
El archivo se ha creado con éxito!

METODO 2

Clase path y file ya podemos escribir de todo.

```
public void crearArchivo3(String nombrefichero) {
    Path ruta = Paths.get(nombrefichero);
    String contenido = "Hola, mundo!\nEste es un ejemplo usando java.nio.";
    try {
        Files.write(ruta, contenido.getBytes(StandardCharsets.UTF_8), StandardOpenOption.CREATE,
            StandardOpenOption.WRITE);
        System.out.println("Escritura completada.");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

VARARRGS SE LE PUEDE PASAR UN MONTON DE LA MISMA CLASE CLASE

```
public void test (String nombre, int ... numeros ) {  
    |  
}
```

AQUI LO APLICO, PUEDO METERLE A TEST NING NUMERO O UN MONTON DE NUMEROS

```
        System.out.println( "Se han completado. ",  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    test ("Hola",100,10,98,17);  
}  
  
public void test (String nombre, int ... numeros ) {  
    for (int i : numeros) {  
        ~
```

HYBERNATE

es un framware.

Esto es la mierda que estamos haciendo:
con jdbc

