

Gestión de Ficheros

Tema 1. Acceso a datos

Objetivos

- Conocer diferencias en la gestión de ficheros de texto y binarios
- Diferenciar entre acceso secuencial y acceso aleatorio
- Clases de Java para el acceso a ficheros.
- Comprender el mecanismo de buffering
- Acceder correctamente a los contenidos de los archivos de texto
- Persistencia de objetos en ficheros

¿Qué es un archivo?

- Conjunto de bytes que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene.
- Es la unidad de almacenamiento básica del sistema operativo.
- Dependiendo del contenido pueden ser binarios o de texto.

Clase File

<https://docs.oracle.com/javase/8/docs/api/java/io/File.html>

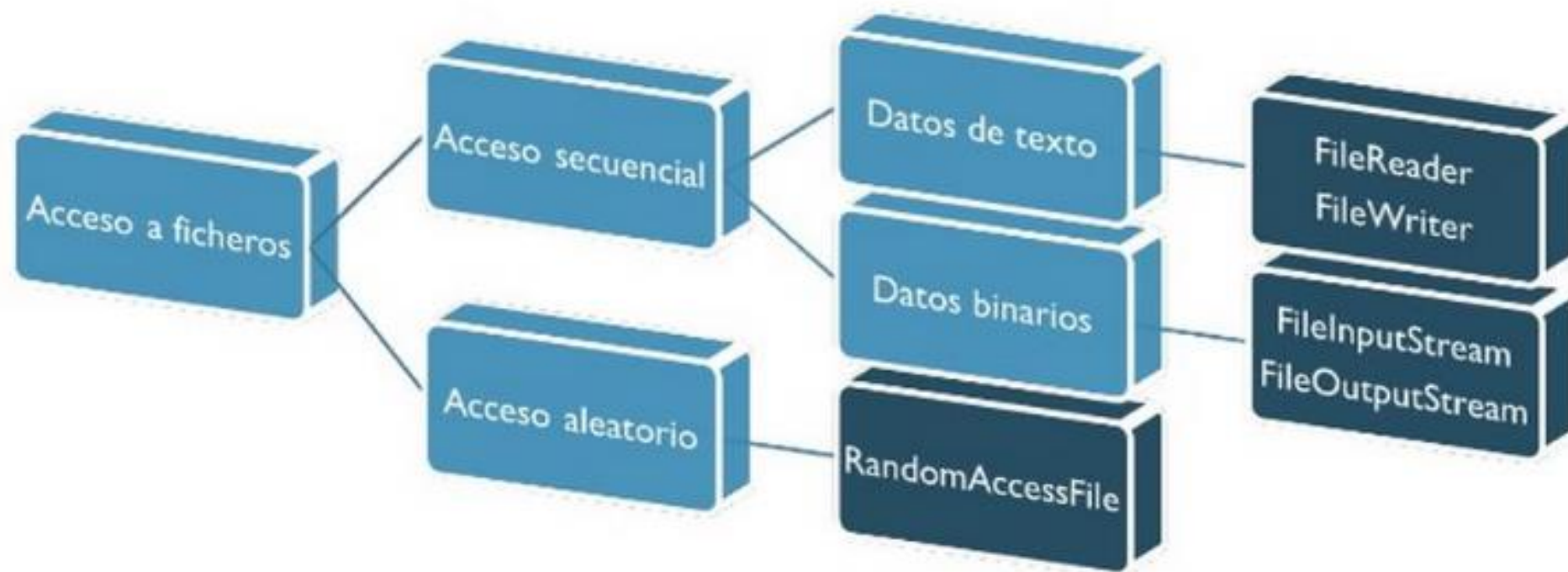
- Es la clase encargada de gestionar los archivos y directorios a nivel de sistema operativo.
- Constructor: **File(String pathname)**
- Métodos principales:
List(), listFiles(), getName(), createFile(), length(),
isFile()...

Excepciones de la clase File

<https://docs.oracle.com/javase/8/docs/api/java/io/IOException.html>

- `IOException`: Error al acceder al archivo
- `NullPointerException`: Error en el constructor
- `SecurityException`: El sistema deniega alguna operación por seguridad

Accesos a ficheros en Java

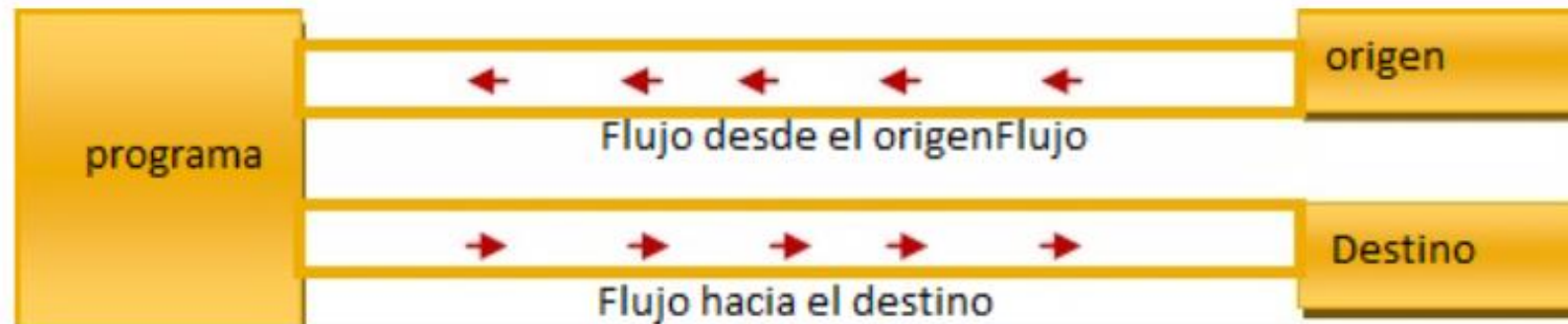


Flujos de datos

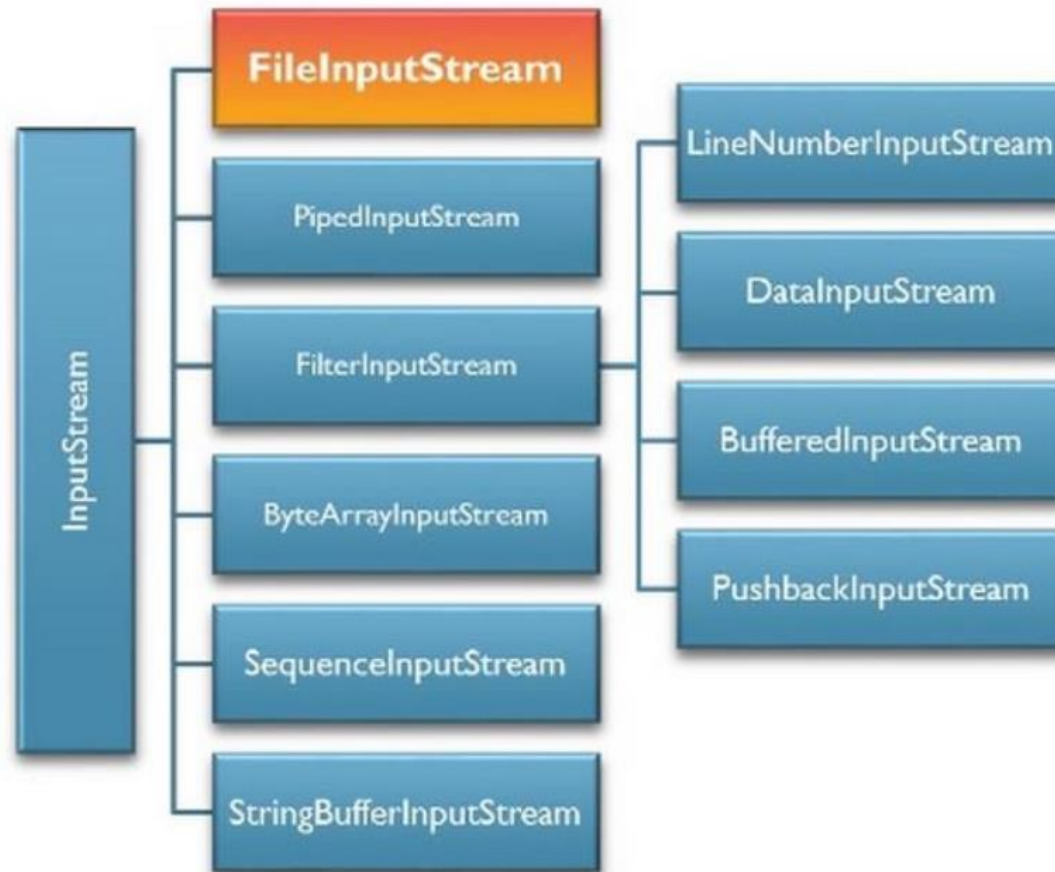
- Un flujo es un objeto que hace de intermediario entre un programa y la fuente o el destino de la información.
- Esto permite que el programa pueda leer o escribir información en el flujo sin importarle el origen o destino de la información ni el tipo de datos

Flujos básicos

- Bytes (archivos binarios)
- Caracteres (archivos de texto)

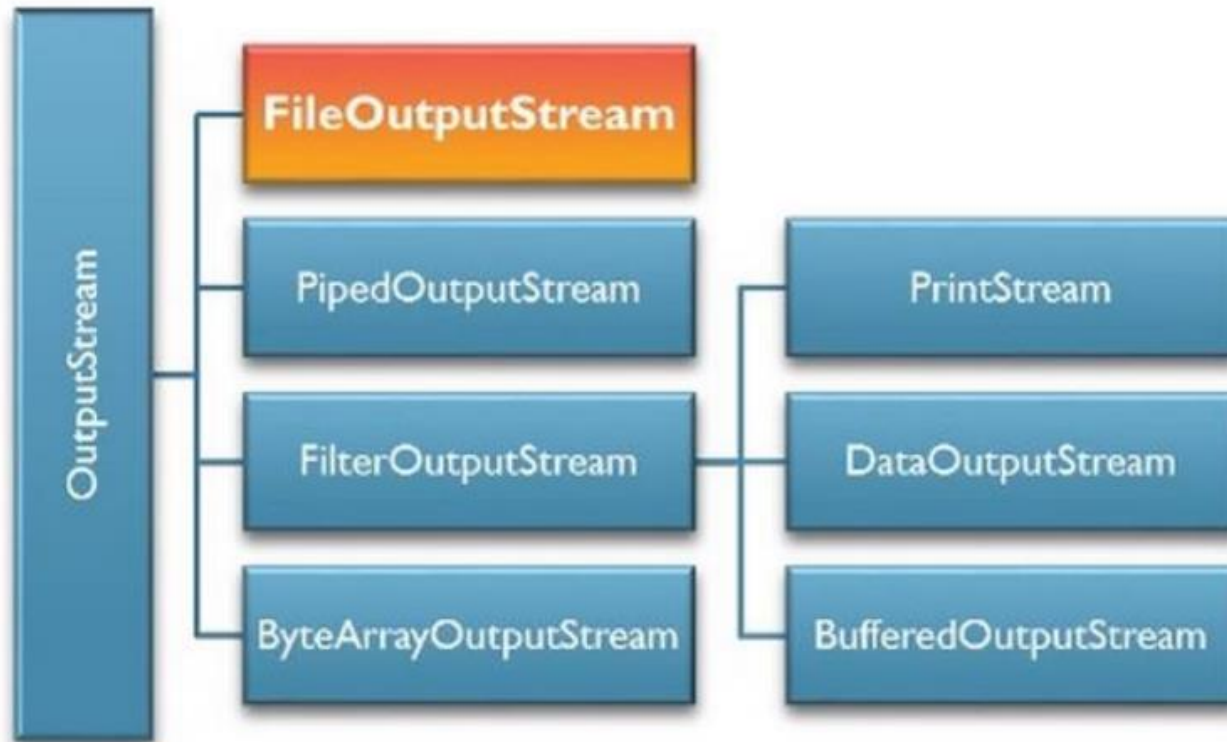


Clase InputStream



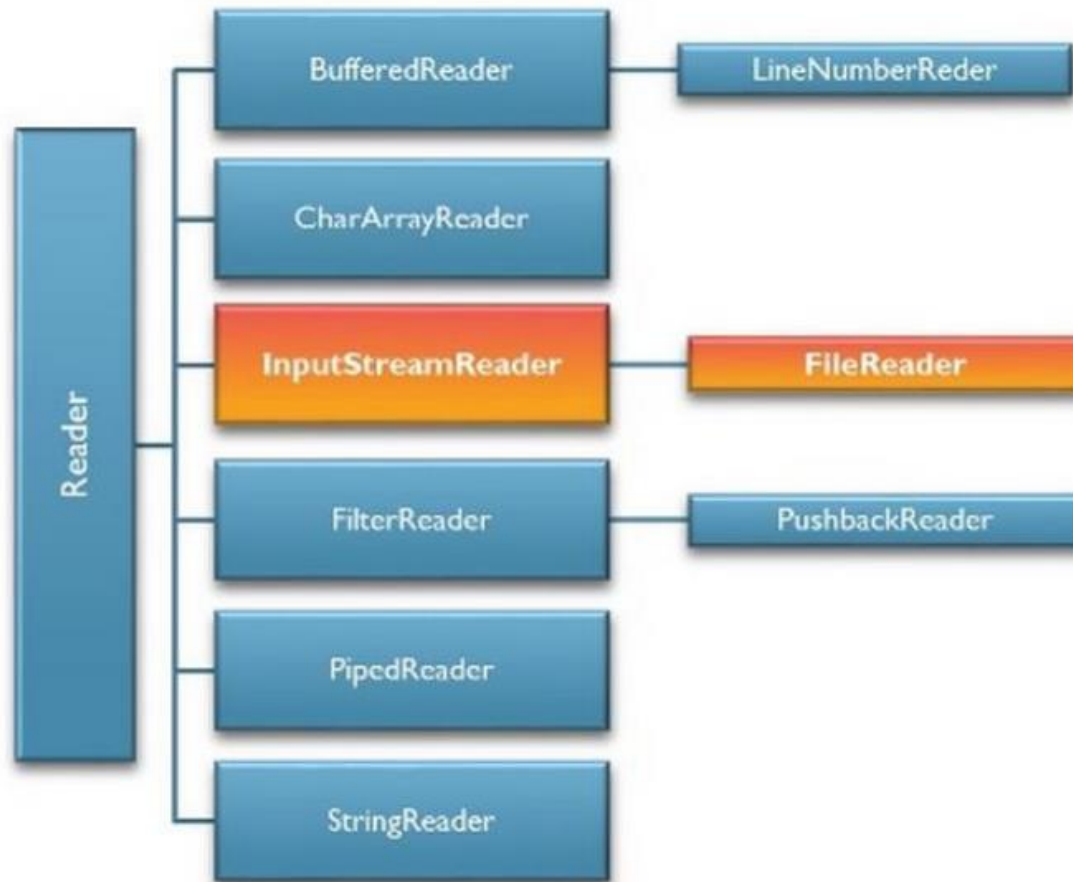
- `InputStream`
- `FileInputStream`
- `PipedInputStream`
- `ByteArrayInputStream`

Clase OutputStream



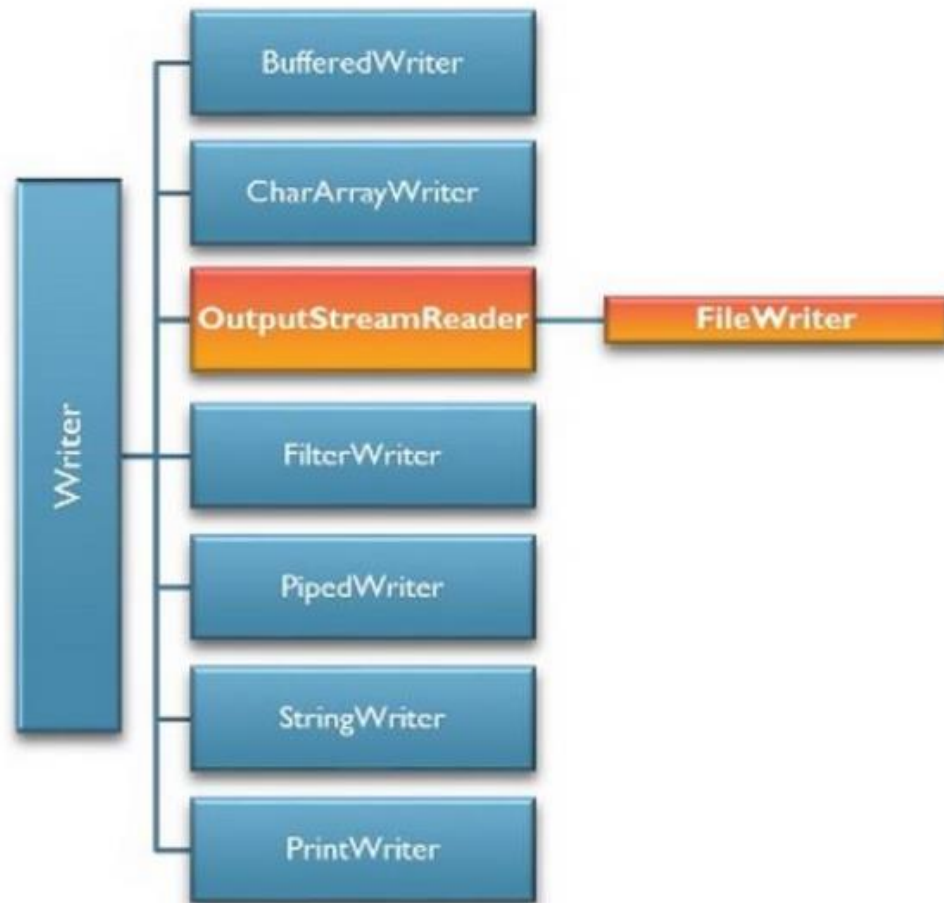
- **OutputStream**
- `FileOutputStream`
- `PipedOutputStream`
- `ByteArrayOutputStream`

Clase Reader



- Reader
- FileReader
- PipedReader
- ByteArrayReader

Clase Writer



- `Writer`
- `FileWriter`
- `PipedWriter`
- `ByteArrayWriter`

Lectura binaria de un archivo

```
try {  
    FileInputStream in = new FileInputStream(orig);  
  
    int b;  
  
    while ((b = in.read()) > 0) { }  
  
    in.close();  
} catch (IOException ioe){  
    ioe.printStackTrace();  
}
```

Lectura de un archivo de texto

```
try( FileReader entrada = new FileReader("pom.xml")){  
    int dato;  
    while( (dato = entrada.read()) != -1 ){  
  
        }  
} catch (IOException ex) { }
```

Classes con buffering

- Permiten agilizar el acceso a los ficheros, muy especialmente los de texto
- Usa las clases `BufferedReader` y `BufferedWriter`
- Permite métodos como `readLine(...)` `newLine(...)`

```
BufferedReader bfr = new BufferedReader(new FileReader("f.txt"));
```

Operaciones de escritura

- `.write(...)`
- `.append(...)`
- `.newLine(...)`

Operaciones de lectura

- `.read(...)`
- `.skip(...)`
- `.readLine(...)`

Clase Scanner

<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>

Optimiza la lectura de ficheros de texto con datos formateados y permite usar expresiones regulares.

Métodos:

`nextInt(), nextFloat()..., hasNext()`

```
Scanner sc = new Scanner(new File("myNumbers"));  
Scanner teclado = new Scanner(in);
```

Ejemplos de acceso a archivos

- Lectura de un archivo de texto y mostrarlo en pantalla.
- Leer un archivo de texto y convertirlo en otro archivo que tenga todo el texto en mayúsculas.
- Guardar en un archivo de texto las líneas introducidas por el usuario por teclado.

Clase Properties

<https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html>

La clase Properties permite almacenar la configuración de un programa de Java de forma sencilla.

```
Properties propiedades = new Properties();  
propiedades.load(new FileInputStream("archivo.cfg"));  
String valor = propiedades.getProperty("clave");
```

Clase Properties

<https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html>

```
1  #cambio de versión
2  #Sat Sep 26 20:30:10 CEST 2020
3  date=2016-11-12
4  Zip=Enabled
5  name=TestaPeal
6  CantModified=True
7  version=4.4.4
```

Persistencia de objetos

<https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>

Para almacenar objetos en archivos se usan las clases `ObjectInputStream` y `ObjectOutputStream`.

Para ello las clases de los objetos han de implementar el interfaz `Serializable`, y todos sus atributos también deben serlo.

Persistencia de objetos

<https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html>

Excepciones que hay que tratar:

- IOException
- ClassNotFoundException
- ObjectOutputStreamException

```
FileInputStream f1 = new FileInputStream("media.obj");  
ObjectInputStream entrada = new ObjectInputStream(f1);
```

Escribir un ArrayList en un archivo

Generar un ArrayList de 20 enteros aleatorios y almacenarlos en un archivo.

Leer el objeto del archivo y mostrarlo por pantalla.

Apache Commons CSV

```
<dependency>  
  <groupId>org.apache.commons</groupId>  
  <artifactId>commons-csv</artifactId>  
  <version>1.4</version>  
</dependency>
```

Apache Commons CSV

```
try (FileReader in = new FileReader(archivo)) {  
    Iterable<CSVRecord> records;  
    records = CSVFormat.DEFAULT.parse(in);  
  
    for (CSVRecord record : records) {  
        System.out.println( record.get("nombre") );  
    }  
}
```

Apache Commons CSV

```
try (FileWriter out = new FileWriter(archivo, true);  
    CSVPrinter printer = new CSVPrinter(out, CSVFormat.DEFAULT);  
    Scanner sc = new Scanner(System.in)) {  
  
    System.out.println("Añadir una entrada:");  
    printer.printRecord(sc.next(), sc.next(), sc.next());  
  
} catch (IOException ex) { }
```

Leer un archivo de texto y crear un archivo csv con el histograma de todas las palabras de más de longitud mayor de 2 caracteres.

Enlaces de interés

- <http://docs.oracle.com/javase/7/docs/api/java/io/File.html>
- <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/archivos/flujo.htm>
- <https://www.youtube.com/watch?v=4TFi4tyBQGw>
- https://www.youtube.com/watch?v=Ng0_7uZyloA
- <http://www.javahispano.org/storage/contenidos/serializacion.pdf>
- http://www.mclibre.org/consultar/xml/lecciones/xml_que_es.html
- <http://howtodoinjava.com/2014/07/30/dom-vs-sax-parser-in-java/>
- <https://www.mclibre.org/consultar/xml/lecciones/xml-xpath.html>