



Cofinanciado por
la Unión Europea



MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Fondos Europeos



Con Scanner:

Otra forma común de leer archivos es usar la clase **Scanner**.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class LeerArchivoConScanner {
    public static void main(String[] args) {
        File archivo = new File("archivo.txt");
        try (Scanner scanner = new Scanner(archivo)) {
            while (scanner.hasNextLine()) {
                System.out.println(scanner.nextLine()); // Imprime cada línea
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

- **Scanner** puede ser útil para leer archivos línea por línea, palabras o números.

2. Escribir en Archivos en Java

Con FileWriter y BufferedWriter:

Al igual que al leer, al escribir archivos, se recomienda usar **BufferedWriter** para mejorar el rendimiento.

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class EscribirArchivo {
    public static void main(String[] args) {
        String contenido = "Este es un ejemplo de escritura en un archivo.";
        try (BufferedWriter writer = new BufferedWriter(new FileWriter("salida.txt"))) {
            writer.write(contenido); // Escribe el contenido
            writer.newLine();        // Añade una nueva línea (opcional)
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



REGISTRO NACIONAL DE ASOCIACIONES N°611922
DECLARADA ENTIDAD DE UTILIDAD PÚBLICA ESTATAL
AGENCIA DE COLOCACIÓN: ID 0100000017

CENTRO EN MÁLAGA
C/DOS ACERAS 23, 29012
MÁLAGA | (+34) 952 300 500
ARRABAL@ARRABALEMPLEO.ORG

CENTRO EN CÁDIZ
TR.º ALAMEDA DE SOLANO, 32, 11130
CHICLANA DE LA FRONTERA | (+34) 956 900 312
CHICLANA@ARRABALEMPLEO.ORG



Cofinanciado por
la Unión Europea



MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Fondos Europeos



Ayuda
en Acción
Impulsa Empleo Joven

En este ejemplo:

- **BufferedWriter** mejora la eficiencia al escribir grandes cantidades de datos.
- **write()** escribe una cadena de caracteres en el archivo.

Con **PrintWriter**:

Si necesitas escribir en un formato más flexible, **PrintWriter** puede ser útil. Permite escribir tanto cadenas como datos de tipo primitivo (int, double, etc.).

```
import java.io.FileNotFoundException;
import java.io.PrintWriter;

public class EscribirConPrintWriter {
    public static void main(String[] args) {
        try (PrintWriter writer = new PrintWriter("salida.txt")) {
            writer.println("Escribiendo con PrintWriter.");
            writer.println("Número: " + 123);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

- **PrintWriter** es conveniente porque automáticamente maneja la codificación de caracteres y el salto de línea.



REGISTRO NACIONAL DE ASOCIACIONES N°611922
DECLARADA ENTIDAD DE UTILIDAD PÚBLICA ESTATAL
AGENCIA DE COLOCACIÓN: ID 0100000017

CENTRO EN MÁLAGA
C/DOS ACERAS 23, 29012
MÁLAGA | (+34) 952 300 500
ARRABAL@ARRABALEMPLEO.ORG

CENTRO EN CÁDIZ
TR.º ALAMEDA DE SOLANO, 32, 11130
CHICLANA DE LA FRONTERA | (+34) 956 900 312
CHICLANA@ARRABALEMPLEO.ORG



Cofinanciado por
la Unión Europea



MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Fondos Europeos



3. Manejo de Archivos con java.nio.file

Desde **Java 7**, la clase **Paths** y el paquete **java.nio.file** ofrecen una forma más moderna y flexible de manejar archivos y directorios.

Leer un archivo con Files (de java.nio.file):

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;
import java.util.List;

public class LeerConNIO {
    public static void main(String[] args) {
        Path path = Paths.get("archivo.txt");
        try {
            List<String> lineas = Files.readAllLines(path); // Lee todas las líneas
            for (String linea : lineas) {
                System.out.println(linea);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- **Files.readAllLines(path)** lee todo el contenido de un archivo de texto y lo guarda en una lista de cadenas.
- El paquete **java.nio.file** es más eficiente y flexible en la manipulación de archivos y directorios.



REGISTRO NACIONAL DE ASOCIACIONES N°611922
DECLARADA ENTIDAD DE UTILIDAD PÚBLICA ESTATAL
AGENCIA DE COLOCACIÓN: ID 0100000017

CENTRO EN MÁLAGA
C/DOS ACERAS 23, 29012
MÁLAGA | (+34) 952 300 500
ARRABAL@ARRABALEMPLEO.ORG

CENTRO EN CÁDIZ
TR.ª ALAMEDA DE SOLANO, 32, 11130
CHICLANA DE LA FRONTERA | (+34) 956 900 312
CHICLANA@ARRABALEMPLEO.ORG



Cofinanciado por
la Unión Europea



MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Fondos Europeos



Junta de Andalucía
Fuerza de trabajo
América del Sur
REINICIANDO LA ECONOMÍA



Ayuda
en Acción
Impulsa Empleo Joven

Escribir en un archivo con Files:

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;

public class EscribirConNIO {
    public static void main(String[] args) {
        String contenido = "Este es un ejemplo con java.nio.file.";
        Path path = Paths.get("salida.txt");
        try {
            Files.write(path, contenido.getBytes()); // Escribe contenido en el archivo
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

- **Files.write(path, contenido.getBytes())** escribe contenido en el archivo en formato de bytes.

4. Manejo de Directorios

Puedes usar las clases de **java.nio.file** para trabajar con directorios, como crear, listar y eliminar directorios.

Crear un directorio:

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;

public class CrearDirectorio {
    public static void main(String[] args) {
        Path path = Paths.get("nuevo_directorio");
        try {
            Files.createDirectory(path); // Crea un nuevo directorio
            System.out.println("Directorio creado exitosamente.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



REGISTRO NACIONAL DE ASOCIACIONES N°611922
DECLARADA ENTIDAD DE UTILIDAD PÚBLICA ESTATAL
AGENCIA DE COLOCACIÓN: ID 0100000017

CENTRO EN MÁLAGA
C/DOS ACERAS 23, 29012
MÁLAGA | (+34) 952 300 500
ARRABAL@ARRABALEMPLEO.ORG

CENTRO EN CÁDIZ
TR.ª ALAMEDA DE SOLANO, 32, 11130
CHICLANA DE LA FRONTERA | (+34) 956 900 312
CHICLANA@ARRABALEMPLEO.ORG



Cofinanciado por
la Unión Europea



MINISTERIO
DE TRABAJO
Y ECONOMÍA SOCIAL



Fondos Europeos



Ayuda
en Acción
Impulsa Empleo Joven

Listar archivos en un directorio:

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;
import java.nio.file.DirectoryStream;

public class ListarArchivos {
    public static void main(String[] args) {
        Path path = Paths.get("directorio");
        try (DirectoryStream<Path> stream = Files.newDirectoryStream(path)) {
            for (Path entry : stream) {
                System.out.println(entry.getFileName());
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Resumen:

- **java.io** es el paquete tradicional para manejar archivos en Java, con clases como **FileReader**, **BufferedReader**, **FileWriter** y **BufferedWriter**.
- **java.nio.file** es una alternativa más moderna y eficiente introducida en Java 7, que permite un manejo más sencillo de archivos y directorios a través de las clases **Paths** y **Files**.
- **Manejo de directorios:** Con **java.nio.file**, puedes crear, listar y eliminar directorios con mayor facilidad.



REGISTRO NACIONAL DE ASOCIACIONES N°611922
DECLARADA ENTIDAD DE UTILIDAD PÚBLICA ESTATAL
AGENCIA DE COLOCACIÓN: ID 0100000017

CENTRO EN MÁLAGA
C/DOS ACERAS 23, 29012
MÁLAGA | (+34) 952 300 500
ARRABAL@ARRABALEMPLEO.ORG

CENTRO EN CÁDIZ
TR.º ALAMEDA DE SOLANO, 32, 11130
CHICLANA DE LA FRONTERA | (+34) 956 900 312
CHICLANA@ARRABALEMPLEO.ORG