

Tema 9. Entrada/Salida

Módulo: Programación
Curso: 1º Desarrollo de Aplicaciones
Multiplataforma
IES Castelar (Badajoz)

java.nio

- En java.io encontramos clases para gestionar archivos con facilidad. La principal es File, que se ha venido usando desde las primeras versiones de java.

Java.nio

- El paquete `java.nio` mejora las funcionalidades de `java.io` con nuevas clases, cuyo uso se recomienda. Desde la versión 7 contamos con NIO.2, que mejora las posibilidades de `nio`.
- En la parte ficheros, contamos con buffers y canales que mejoran el rendimiento de los streams tradicionales, mejorando la velocidad de las operaciones de entrada y salida.

Path

- La clase Path es el eje de las funciones de gestión de archivos.
- <https://docs.oracle.com/javase/8/docs/api/java/nio/file/Path.html>
- Representa una ruta compuesta por archivos y directorios que identifican a un elemento, con sus separadores.

Path

Con el método `Paths.get()` tomamos una cadena y la convertimos en una ruta:

```
Path path = Paths.get("D:\\Prueba\\archivo.txt");  
System.out.println(path.toString());
```

Path

Información del path obtenido:

```
Path path = Paths.get(".\\Prueba\\entrada\\numeros.txt");  
System.out.println(path.getFileName()); //nombre del archivo o dir  
System.out.println(path.getName(0)); //elemento en posición 0 de la  
ruta  
System.out.println(path.getNameCount()); //numero de elementos  
System.out.println(path.subpath(0, 2)); //elementos de la ruta  
System.out.println(path.getParent()); // ruta padre del archivo  
System.out.println(path.getRoot()); //Raíz, null para las relativas
```

Path

Información del path obtenido:

```
Path path = Paths.get(".\\Prueba\\entrada\\numeros.txt");
```

```
System.out.println(path.getFileName()); //nombre del archivo o dir
```

```
System.out.println(path.getName(0)); //elemento en posición 0 de la ruta
```

```
System.out.println(path.getNameCount()); //numero de elementos
```

```
System.out.println(path.subpath(0, 2)); //elementos de la ruta
```

```
System.out.println(path.getParent()); // ruta padre del archivo
```

```
System.out.println(path.getRoot()); //Raíz, null para las relativas
```

```
Path absolute = path.toAbsolutePath(); //convierte en absoluto
```

Operaciones con ficheros

La clase `Files` opera con `Path` para realizar operaciones con sus métodos estáticos: `copy`, `createFile`, `createDirectory`, `exists`, `noExists`, `delete`, `find`, `getAttribute`, `isDirectory`, `isHidden`, `size`, etc.

Ejemplo:

```
Path path2 = Paths.get(".\\Prueba\\entrada\\otroarchivo.txt");

if (Files.exists(path2)) {
    System.out.println("El fichero existe");
} else {
    System.out.printf("El fichero %s no existe", path2.getFileName());
}
```


Files

- Podemos usar Files para leer y escribir en archivos, con write, readalllines o lines.

```
Path path = Paths.get("archivo.txt");  
    byte[] texto = "Hola a todos\n".getBytes();  
    Files.write(path, texto);  
    Files.write(path, "Otra línea  
más".getBytes(), StandardOpenOption.APPEND);  
    //Leemos  
    System.out.println(Files.readAllLines(path));
```

Tema 9. Entrada/Salida

Módulo: Programación
Curso: 1º Desarrollo de Aplicaciones
Multiplataforma
IES Castelar (Badajoz)