

1) Path

- **Librería:** *NIO.2*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *java.nio.file*

Esta clase está pensada para reemplazar a la clase File. Al igual que ella, un Path representa la ruta de un archivo o un directorio del equipo. Dicha ruta puede estar expresada en forma absoluta (comienza por la letra de la unidad, o la barra /) o relativa (se refiere a una ruta respecto a un directorio).

Path
+ Path getFileName() + Path getParent() + Path resolve(Path p) + Path resolve(String s) + Path toRealPath() throws IOException + String toString()

- **getFileName:** Devuelve un objeto Path con el nombre del archivo.
- **getParent:** Devuelve un objeto Path con la ruta del directorio padre.
- **Primer método resolve:** Devuelve el objeto Path que resulta de pegar el Path que se pasa como parámetro, al Path representado por el objeto sobre el que se llama al método.
- **Segundo método resolve:** Hace lo mismo que el anterior, pero el parámetro viene dado como String en lugar de un Path.
- **toRealPath:** Convierte el Path representado el objeto (si es relativo) en un Path absoluto. Para ello, le concatena la ruta de la carpeta de la aplicación. En caso de que salga una ruta inexistente, se lanza una IOException
- **toString:** Devuelve un texto con la ruta del archivo o carpeta representado en el objeto.

2) Paths

- **Librería:** *NIO.2*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *java.nio.file*

Esta clase permite obtener objetos de la clase Path.

Paths
+ static Path get(String ruta)

- **get:** Devuelve un objeto Path a partir de la ruta que se pasa como parámetro. Se admite tanto una ruta absoluta (ej: c:\documentos\exámenes\1.doc) como una ruta relativa (exámenes\1.doc). Es posible usar la barra / o las barras \\ para escribir la ruta.

3) Files

- **Librería:** *NIO.2*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *java.nio.file*

Esta clase permite realizar operaciones habituales con ficheros y carpetas (copiar, mover, etc) que estén definidos mediante objetos Path.

Nota: En el IDE, al escribir muchos de sus métodos puede verse que se admite un parámetro más llamado CopyOption o LinkOption. Dicho parámetro puede dejarse sin escribir. Se recomienda llamar a los métodos tal y como viene en esta documentación:

Files
+ static long copy(Path origen, Path destino) throws IOException + static Path createDirectory(Path p) throws IOException + static void delete(Path p) throws IOException + static void move(Path origen, Path destino) throws IOException

- **copy:** Copia lo que haya en el Path de origen en el Path de destino y devuelve el número de bytes copiados. Si algo va mal, se lanza una IOException.
- **createDirectory:** Crea el directorio cuya ruta se pasa como parámetro. Devuelve un Path con el nuevo directorio creado.
- **delete:** Borra el archivo o directorio (vacío) que se pase como parámetro.
- **move:** Mueve lo que haya en el Path de origen en el Path de destino.