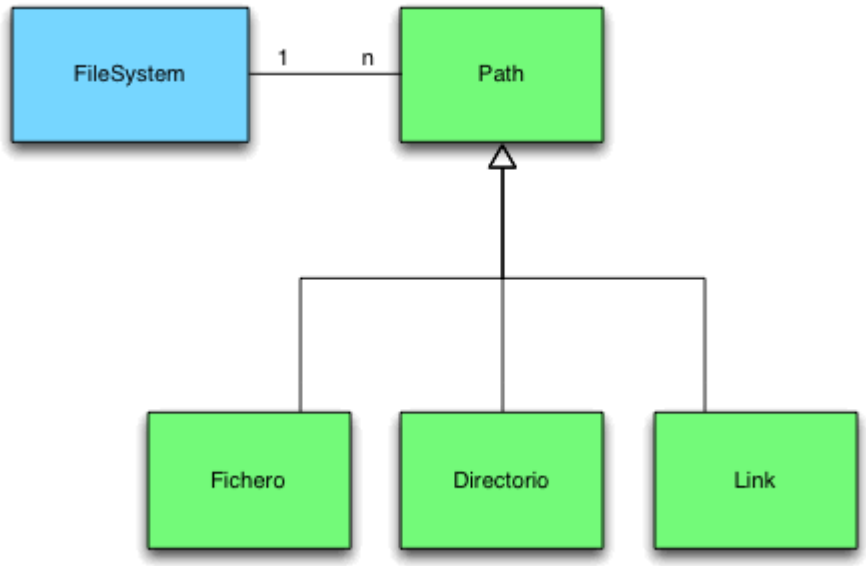
Hemos usado durante muchos años java.io para trabajar con ficheros en el mundo Java . Se trata de un API muy potente y flexible que nos permite realizar casi cualquier tipo de operación. Sin embargo es un API complicada de entender. Java NIO (Not Blocking IO) es un nuevo API disponible desde Java7 que nos permite mejorar el rendimiento así como simplificar el manejo de muchas cosas. Vamos a ver un ejemplo apoyándonos en FileSystem y Path de este nuevo API.

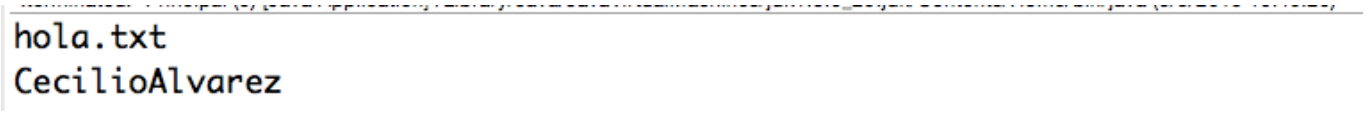


El concepto de FileSystem define un sistema de ficheros completo. Mientras que por otro lado el concepto de Path hace referencia a un directorio, fichero o link que tengamos dentro de nuestro sistema de ficheros. El siguiente código hace uso de FileSystem y Path para obtener el nombre de un fichero así como la carpeta padre en la que se encuentra ubicado.

```
FileSystem sistemaFicheros=FileSystems.getDefault();
Path
```

```
rutaFichero=sistemaFicheros.getPath("/Users/CecilioAlvarez/hola.txt");
System.out.println(rutaFichero.getFileName());
System.out.println(rutaFichero.getParent().getFileName());
```

El resultado será :



De forma similar podemos mostrar la lista de carpetas de la jerarquía de nuestro directorio:

```
Path
rutaDirectorio=sistemaFicheros.getPath("/Users/CecilioAlvarez/carpeta"
);
Iterator<Path> it=rutaDirectorio.iterator();
while(it.hasNext()) {

System.out.println(it.next().getFileName());
}
```



Por último podemos de una forma muy sencilla leer el contenido que hay en un Path (fichero) e imprimirlo por pantalla.

```
try {
List<String> texto=Files.readAllLines(rutaFichero);

for(String cadena:texto) {

System.out.println(cadena);
}

} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
```

```
hola desde un fichero
```

Como podemos ver es muy sencillo leer todas las lineas de texto que tiene un fichero y convertirla en una lista de cadenas a recorrer. El API de Java NIO nos aporta nuevos elementos para simplificar la programación del día a día.

Otros artículos relacionados: Entendiendo los constructores en Java ,El concepto de Java Interface ,Java Generics