

### 1.Creación de un fichero:

```
try {
    // Creamos el objeto que encapsula el fichero
    File fichero = new File("/home/directorio/miFichero.txt");
    // A partir del objeto File creamos el fichero físicamente
    if (fichero.createNewFile())
        System.out.println("El fichero se ha creado correctamente");
    else
        System.out.println("No ha podido ser creado el fichero");
} catch (Exception ioe) {
    ioe.getMessage();
}
```

### 2.Borrado de un fichero:

```
// Borrar fichero
File fichero = new File("/home/directorio/miFichero.txt");
if (fichero.exists ())
    fichero.delete();
```

### 3.Creación de uno o varios directorios:

```
try{
    // Declaración de variables
    String directorio = "/home/directorio";
    String varios = "carpeta1/carpeta2/carpeta3";
    // Crear un directorio
    boolean exito = (new File(directorio)).mkdir();
    if (exito)
        System.out.println("Directorio: " + directorio + " creado");
    // Crear varios directorios
    exito = (new File(varios)).mkdirs();
    if (exito)
        System.out.println("Directorios: " + varios + " creados");
} catch (Exception e){
    System.err.println("Error: " + e.getMessage());
}
```

**4.Borrado de ficheros:** hay que borrar cada uno de los ficheros y directorios que este contenga. Al poder almacenar otros directorios, se podría recorrer recursivamente el directorio para ir borrando todos los ficheros.

```
// listamos el contenido de un directorio:
String directorio = "/home/directorio";
File [] ficheros = new File(directorio).listFiles ();
for (int i = 0; i < ficheros.length; i++){
    // Comprobamos si es un directorio
    if (ficheros[i].isDirectory()){
        // Listar los ficheros y eliminarlos uno a uno
        //TODO
        //Eliminar el directorio: se puede borrar importando la librería
        FileUtils y después escribir
        FileUtils.deleteDirectory(new File(destination));
    }else{
        ficheros[i].delete();
    }
}
```

# Existencia y listado de ficheros y carpetas

Mediante la clase **File**, podemos ver si un fichero cualquiera, digamos por ejemplo texto.txt, existe o no. Para ello, nos valemos del método **exists()**.

A continuación, mostramos un ejemplo del código:

```
String path = "/home/directorio";
File f = new File(path);
if (!f.exists()){
    System.out.println("No existe el fichero");
}
else{
    System.out.println("Existe el fichero");
    // Obtenemos el listado de elementos
    String[] listaArchivos = f.list();
    // La cargamos en el JList
    for(int i = 0; i < listaArchivos.length; i++){
        // Pintamos el nombre del archivo
        System.out.println(listaArchivos[i]);
    }
}
```

## Listar ficheros de una carpeta, filtrando

```
import java.io.File;
import java.io FilenameFilter;
public class Filtrar implements FilenameFilter {
    String extension;
    // Constructor
    Filtrar(String extension){
        this.extension = extension;
    }
    public boolean accept(File dir, String name){
        return name.endsWith(extension);
    }
    public static void main(String[] args) {
        try {
            // Obtendremos el listado de los archivos de ese directorio
            File fichero=new File("c:\\datos\\.");
            String[] listadeArchivos = fichero.list();

            // Filtraremos por los de extension .txt
            listadeArchivos = fichero.list(new Filtrar(".txt"));
        }
    }
}
```

```

// Comprobamos el número de archivos en el listado
int numarchivos = listadeArchivos.length ;

// Si no hay ninguno lo avisamos por consola
if (numarchivos < 1)
    System.out.println("No hay archivos que listar");

// Y si hay, escribimos su nombre por consola.
else{
    for(int conta = 0; conta < listadeArchivos.length; conta++)
        System.out.println(listadeArchivos[conta]);
}
} catch (Exception ex) {
    System.out.println("Error al buscar en la ruta indicada");
}
}
}

```

## Mapa conceptual

