IES CHAN DO MONTE
C.S. de Desarrollo de Aplicaciones Multiplataforma
Modulo Acceso a datos

ACTIVIDAD 1:

Objetivo:

- Utilizar las clases en Java para trabajar con el sistema de ficheros y directorios
- Manejar las excepciones y crear excepciones propias.

- Desarrollar una aplicación en Java que permita realizar distintas operaciones sobre el sistema de archivos.
- El **código se organizará en clases separadas para lógica de negocio y controlador/vista**, y utilizando excepciones propias para el manejo de errores.
- Vamos a implementar dos versiones de las mismas operaciones, una con la API clásica de Java (java.io) y otra con la API moderna (java.nio.file).**Estructura de la aplicación:**

```
src/
 ├─ excepciones/
 │    ├─ DirectorioNoExisteException.java
 │    ├─ NoEsDirectorioException.java
 │    └─ ArchivoNoExisteException.java
 ├─ servicio/
 │    ├─ OperacionesIO.java      // Versión con java.io.File
 │    └─ OperacionesNIO.java     // Versión con java.nio.file
 ├─ MainApp.java                 // Menú y controlador
```

Clases y responsabilidades

- **Clase OperacionesIO (versión con java.io.File):** Implementa la lógica usando la API clásica (File).
- **Clase OperacionesNIO (versión con java.nio.File):** usando las clases del paquete java.nio.file (Path, Paths, Files, FileVisitor, BasicFileAttributes, etc.).

Métodos a implementar en ambas versiones:

1. visualizarContenido(String ruta)

- Lista el contenido de un directorio (solo el nivel actual).
- Muestra nombre, tipo (<DIR> o <FICHERO>), tamaño en KB (solo ficheros) y fecha de última modificación.
- Los siguientes errores que se puedan producir se controlarán lanzado excepciones propias y son:
  - Si el parámetro introducido no existe, se visualizará un mensaje de error correspondiente.
  - Si el parámetro introducido no representa a un directorio, se visualizará un mensaje de error correspondiente.
- **Ejemplo**:

```
--- LISTANDO EL DIRECTORIO D:\ACCESO a DATOS\Unidades didácticas\Unidad 1-Gestión de fic
-|AnexoIUNIDAD1A-Excepciones.doc <FICHERO> 3.620 Kbytes  23/01/2012 01:09
-|AnexoIUNIDAD1A-Excepciones.pdf <FICHERO> 1.868 Kbytes  23/00/2012 01:34
-|Ejunidad1 <DIR>
-|Ejunidad1-Clase <DIR>
-|EjUnidad1-internet <DIR>
-|Ejunidad1-Teoria <DIR>
-|resume   ficheros.pdf <FICHERO> 38 Kbytes  20/12/2012 08:01
-|UNIDAD1A-Gestión de ficheros.doc <FICHERO> 3.897 Kbytes  19/16/2012 02:47
-|UNIDAD1A-Gestión de ficheros.pdf <FICHERO> 2.696 Kbytes  19/16/2012 02:19
-|UNIDAD1B-Gestión de ficheros XML.doc <FICHERO> 114 Kbytes  31/15/2012 03:55
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejemplo de error si introducimos un parámetro que no exista:

```
--- LISTANDO EL DIRECTORIO D:\ACCESO a DATOS\Unidades didácticas\Unidad 1-Gestión de fic
la ruta especificada no existe
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ejemplo de error si introducimos un parámetro que no es un directorio, por ejemplo: D:\Fotos\1_oscar.jpg"

```
--- LISTANDO EL DIRECTORIO D:\Fotos\1_oscar.jpg ---
la ruta no es un directorio
BUILD SUCCESSFUL (total time: 1 second)
```

2. recorrerRecursivo(String ruta)

- Lista el contenido de un directorio y todos sus subdirectorios con sangría.

```
-|Unidad 1-Gestión de ficheros <DIR>
-----|AnexoIUNIDAD1A-Excepciones.doc <FICHERO> 3.620 Kbytes  23/01/2012 01:09
-----|AnexoIUNIDAD1A-Excepciones.pdf <FICHERO> 1.868 Kbytes  23/00/2012 01:34
-----|Ejunidad1 <DIR>
---------|AplicacionAviso <DIR>
-------------|build <DIR>
-----------------|classes <DIR>
--------------------|.netbeans_automatic_build <FICHERO> 0 Kbytes  23/37/2012 08:55
--------------------|.netbeans_update_resources <FICHERO> 0 Kbytes  23/37/2012 08:55
--------------------|TPAviso.class <FICHERO> 1 Kbytes  23/39/2012 08:51
--------------------|TPAvisoPrueba.class <FICHERO> 0 Kbytes  23/37/2012 08:55
-------------|build.xml <FICHERO> 3 Kbytes  23/24/2012 06:04
-------------|manifest.mf <FICHERO> 0 Kbytes  23/24/2012 06:04
-------------|nbproject <DIR>
-----------------|build-impl.xml <FICHERO> 55 Kbytes  23/24/2012 06:04
-----------------|genfiles.properties <FICHERO> 0 Kbytes  23/24/2012 06:04
```

3.- filtrarPorExtension(String ruta, String extension)

Lista solo los archivos que tengan la extensión en el directorio indicado.

Parámetros:

- Una extensión (por ejemplo: .txt, .pdf, .jpg).
- Un directorio.

Errores para controlar:

Si la ruta no existe → DirectorioNoExisteException

Si la ruta no es un directorio → NoEsDirectorioException

Si no se encuentra ningún archivo con esa extensión → mostrar mensaje informativo.

```
Ruta: C:\MisDocumentos
Extensión: .pdf
```

alida:

Código ^

```
-|informe.pdf <FICHERO>  245 KB  12/09/2025 10:15:32
-|Trabajo\manual.pdf <FICHERO>  1.024 KB  01/09/2025 09:00:00
```

4.- filtrarPorExtensionYOrdenar(String ruta, String extension, boolean descendente)

Lista solo los archivos que tengan la extensión indicada en el directorio especificado y en todos sus subdirectorios (recursivo), ordenados alfabéticamente.

✓ Si descendente es true → orden de Z a A.
✓ Si descendente es false → orden de A a Z.
✓ La ordenación ignora mayúsculas y minúsculas.

Parámetros:

- String extension → extensión a buscar.
- String ruta → ruta del directorio donde buscar.
- boolean descendente → true para orden descendente, false para ascendente.

Errores a controlar:

Si la ruta no existe → DirectorioNoExisteException.

Si la ruta no es un directorio → NoEsDirectorioException.

Si no se encuentra ningún archivo con esa extensión → mostrar mensaje informativo.

5.- filtrarPorSubcadena(String ruta, String subcadena)

Lista todos los archivos cuyo nombre contenga la subcadena indicada, en el directorio especificado y en todos sus subdirectorios (recursivo).

La búsqueda ignora mayúsculas y minúsculas.

Parámetros:

- String subcadena → texto a buscar dentro del nombre de archivo.
- String ruta → ruta del directorio donde buscar.

Errores a controlar:

Si la ruta no existe → DirectorioNoExisteException.

Si la ruta no es un directorio → NoEsDirectorioException.

Si no se encuentra ningún archivo que contenga la subcadena → mostrar mensaje informativo.

6.- copiarArchivo(String origen, String destino)

- Copia un archivo desde una ruta de origen a una ruta de destino.
- Si el directorio destino no existe, lo crea.
- Si ya existe un archivo con el mismo nombre en destino, lo sobrescribe.

Parámetros:

- String origen → ruta completa del archivo a copiar.
- String destino → ruta completa donde se guardará la copia.

Errores a controlar:

- Si el archivo origen no existe → ArchivoNoExisteException.
- Si el origen es un directorio → mostrar mensaje de error.
- Si no se puede crear el directorio destino → mostrar mensaje de error.

7.- moverArchivo(String origen, String destino)

- Mueve un archivo desde una ruta de origen a una ruta de destino.
- Si el directorio destino no existe, lo crea.
- Si ya existe un archivo con el mismo nombre en destino, lo sobrescribe.

Parámetros:

- String origen → ruta completa del archivo a mover.
- String destino → ruta completa donde se moverá el archivo.

Errores a controlar:

- Si el archivo origen no existe → ArchivoNoExisteException.
- Si el origen es un directorio → mostrar mensaje de error.
- Si no se puede crear el directorio destino → mostrar mensaje de error.

8.- copiarDirectorio(String origen, String destino)

- Copia un directorio completo con todo su contenido (archivos y subdirectorios).
- Mantiene la estructura original.
- Si el directorio destino no existe, lo crea.
- Mostar información de los directorios y archivos copiados.

Parámetros:

- String origen → ruta del directorio a copiar.
- String destino → ruta donde se creará la copia.

Errores a controlar:

- Si el directorio origen no existe → DirectorioNoExisteException.
- Si el origen no es un directorio → NoEsDirectorioException.
- Si no se puede crear el directorio destino → mostrar mensaje de error.

9.- borrar(String ruta)

- Borra un archivo o un directorio.
- Si es un directorio, borra todo su contenido recursivamente y luego el propio directorio.
- Mostrar información de los archivos y directorios borrados.

Parámetros:

- String ruta → ruta del archivo o directorio a borrar.

Errores a controlar:

- Si la ruta no existe → ArchivoNoExisteException o DirectorioNoExisteException.
- Si no se puede borrar → mostrar mensaje de error.

En la clase MainApp:

- Se pedirá al usuario la información necesaria (ruta, extensión, subcadena, etc.).
- **Se incluirán ejemplos de llamada a cada método** para probar los métodos con la versión **IO** y también con la versión **NIO**.