Tema01. MANEJO DE FICHEROS.

**Resultado de aprendizaje. R1.** Desarrolla aplicaciones que gestionan información **almacenada en ficheros** identificando el campo de aplicación de los mismos y utilizando clases específicas.

**Criterios de evaluación**

|  |
| --- |
| a) Se han utilizado clases para la **gestión de ficheros y directorios**  b) Se han valorado las ventajas y los inconvenientes de las distintas formas de acceso.  c) Se han utilizado las operaciones básicas para acceder a ficheros **binarios, texto, (aleatorios) y serialización de objetos.**  d) Se han utilizado clases para recuperar/almacenar información en un **fichero XML.**  e,f) Se han utilizado clases para convertir a otro formato información contenida en un fichero XML.  g) Se han previsto y gestionado las **excepciones**  h) Se han **probado y documentado** las aplicaciones desarrolladas |

1. **PRUEBAS.** Gestionar directorios/carpetas y ficheros/archivos 🡪 Clase File:
   1. Mostrar los ficheros (nombre del fichero, directorio padre, ruta, tamaño, si se puede leer y escribir en el fichero) y subdirectorios dentro de un directorio especificado.
      1. Mostrar solo los ficheros del directorio actual
      2. Mostrar solo los directorios del directorio actual.
   2. Se crea un objeto File ruta asociado al directorio c:/ficheros y un objeto File f asociado al fichero datos.txt que se encuentra en ese directorio. Si el fichero no existe se crea y si el directorio no existe se crea y a continuación se crea el fichero. Si el fichero existe se muestra el tamaño del mismo.
   3. Eliminar ficheros o carpetas, en este último caso si está vacío.
   4. Cambiar el nombre de fichero o directorio
   5. Mostrar ficheros de acuerdo a una extensión especificada.
   6. muestre todas las unidades de disco del ordenador indicando para cada una de ellas el tamaño disponible y el tamaño total.
   7. Programa que muestra el contenido del directorio raíz de la unidad actual de trabajo y de todos sus subdirectorios de forma recursiva. Para cada directorio se muestran primero los archivos y a continuación las carpetas que contienen de forma recursiva.
   8. Programa que permita buscar un archivo. Devuelva la ruta donde está.
2. **COMPONENTE: GFile**. Gestión de ficheros
   1. Método que devuelva un String que contenga una lista de ficheros (nombre y tamaño) especificando la ruta de partida y la extensión. Utilizar la interface FilenameFilter. Una línea por fichero. Solo ficheros. En caso de no existir ninguno devolverá null.
   2. Método que devuelva un String que contenga una lista de las unidades de disco del ordenador indicando para cada una de ellas el tamaño disponible y el tamaño total. Una línea por unidad. En caso de no existir ninguno devolverá null.
   3. Método que permita localizar o buscar un archivo en el árbol de directorio a partir de una ruta especificada. Devolverá un string con la ruta encontrada o bien null en caso de no encontrarla.
   4. [opcional]Localizar o buscar un archivo en el árbol de directorio a partir de una ruta especificada. Devolverá una colección de string con las rutas encontradas o bien una colección vacío en caso de no encontrarlo.
3. **COMPONENTE: IOTeclado.** Entrada/salida por teclado.
   1. Entrada de líneas de caracteres.
   2. Entrada de números enteros, float y double.
4. **COMPONENTE:** **StrTxt**.
   1. Método que cree un fichero texto a partir de un String.
   2. Método que devuelva un String con el contenido de un archivo texto especificado.
   3. Método que devuelva una colección de String a partir del contenido de un fichero texto utilizando un carácter separador (Probar con un script SQL)
5. **COMPONENTE: IOFile.**
   1. Método que guarde un objeto:Object en un archivo binario.
   2. Método que devuelva un objeto:Object recuperándolo de un archivo binario

Cada componente como mínimo tendrán estos métodos, pero puedes añadirle más.

Pruébalos y demuestra su funcionamiento antes de subirlos.

Documenta convenientemente cada uno de los métodos: el objetivo y modo de funcionamiento. No hace falta hacer un “testamento”. de cada uno de ellos

1. **COMPONENTE: UtilDOM.**
   1. Método que devuelve un Document a partir de un archivo XML.  
       XML2DOM(String CaminoAArchivoXml): Document
   2. Método que devuelva un String-XML a partir de un árbol   
      DOM2XML(Document doc): String
   3. Método que guarde un árbol DOM en un archive XML.  
      DOM2XML(Document doc, String nombreFichero)
   4. Método que devuelva un árbol DOM a partir de un String-XML.  
      string2DOM(String documentoXML): Document
   5. Método que muestre el contenido en consola de un documento.  
      mostrarDoc(Document doc)
   6. Método que cree y devuelva un árbol DOM vacío (Document).  
      crearDOM(): Document
   7. Método que cree el nodo raíz en el árbol DOM:  
      crearRaiz (Document doc, String etiquetaRaiz)
   8. Método que devuelva el elemento raíz  
      Element getRaiz(Document doc):Element
   9. Método para crear un elemento hijo en el documento DOM   
      crearElemento (String etiqueta, String valor, Element padre, Document doc)  
      crearElemento (String etiqueta, String valor, String atributo, String valorA, Element elemPadre, Document doc)
   10. Método que devuelva un nodo en encontrado a partir de una etiqueta y un valor en un documento.  
       buscarElemento(Document doc, String etiqueta, String valor): Node
   11. Método que cree un atributo y su valor en un elemento en un documento.  
       crearAtributo(String nombre, String valor, Element elem)
   12. Método que obtenga texto de un elemento.  
       String getTextoElemento(Element elem, String etiqueta)
2. **APLICACIÓN 1.** Gestión de departamentos y empleados sobre un fichero binario:
   1. Gestionar (A, B, M, C) departamentos de empleados. Un departamento está formado por varios empleados. Un empleado pertenece a un departamento.

**Dos planteamientos en el diseño de las clases para representar la información (unidireccionales):**  
Empleado (dni, nombre, fecha\_nacimiento, oficio, salario, comisión y departamento: Departamento)   
Departamento (num\_dep, nombre y la localidad).   
**O bien**   
Empleado (dni, nombre, fecha\_nacimiento, oficio, salario, comisión)   
Departamento (num\_dep, nombre, localidad, empleados:ArrayList<Empleado>).   
Se contralará que no existan departamentos ni empleados repetidos.

**Dos planteamientos de almacenamiento y por tanto de acceso:**   
1. Procesar los datos de los Departamentos/empleados subidos a un array de Departamentos/empleados. Guardar/recuperar el array de Departamentos/empleados.  
2. Procesar los datos de los empleados registro (Departamento) a registro (Departamento) de forma secuencial.

* 1. Guardar / recuperar datos en el fichero.
  2. Exportar los datos de los departamentos a fichero texto. (csv)
  3. Exportar los datos de los departamentos a un fichero XML (DOM)
  4. Importar al fichero binario los datos de un nuevo departamento con sus empleados contenido en un archivo xml. (DOM)  
     El proceso de importación quedará reflejado en un archivo log (texto)  
     --------------------
  5. Importar usando JAXB, …
  6. Exportar usando JAXB. Este proceso puede resultar un tanto laborioso. NO DESARROLLAR
  7. Actualizar el salario y comisión de los empleados de un departamento a partir de un fichero XML. (SAX)
     1. Si existe se actualizarán los con los nuevos valores.
     2. Si no existe se entenderá que se trata de un error.
     3. El proceso de actualización quedará reflejado en un archivo log (texto)
  8. Crear un archivo HTML a partir de los datos de los empleados y departamentos. (Diseño pendiente de concretar)