



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

M.I Tista García Edgar

Profesor:

Estructura de Datos y Algoritmos II

Asignatura:

05

Grupo:

10

No de Práctica(s):

Calzada Martinez Jonathan Omar

Integrante(s):

*No. de Equipo de
cómputo empleado:*

35

09

No. de Lista o Brigada:

2019-2

Semestre:

22 de abril de 2019

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo: El estudiante conocerá e identificará aspectos sobre los archivos, como las operaciones, el tipo de acceso y organización lógica.

Introducción: Los programas de la computadora son muy útiles y nos facilitan mucho la vida ya que nos ayudan a poder organizar u ordenar información, así como realizar cálculos de manera mucho más rápida que de manera manual, en muchas ocasiones necesitamos guardar esa información y no solo tener el resultado y para eso necesitamos la ayuda de un archivo.

Para poder acceder a determinada información en cualquier momento, se necesitará que ella esté depositada en soportes físicos los cuales la almacenan en forma permanente. Este es el caso de la memoria externa o auxiliar como ser disquete, disco duro, cinta magnética, etc.-, en las cuales, sin necesidad de estar conectadas a la corriente eléctrica, la información permanece allí. La forma de guardar los datos en estos dispositivos auxiliares es mediante unas estructuras llamadas archivos o ficheros.

Desarrollo:

- **Actividad 1). Bibliotecas en java.**

Se realizó la lista para poder trabajar con archivos / ficheros.

1) Las clases **BufferedReader y BufferedWriter** las podemos encontrar en java.io. Estas clases tienen la misma función que FileReader y FileWriter, leer y escribir en ficheros, pero BufferedReader y BufferedWriter optimizan estas funciones.

Se crean igual que FileReader y FileWriter, pero como parámetro insertaremos un objeto FileReader para BufferedReader y un objeto FileWriter para BufferedWriter.

2). **Java.io.File** se usa para obtener información sobre archivos y directorios. Además, la clase File permite crear y eliminar archivos y directorios.

3). Las clases **FileReader y FileWriter** permiten leer y escribir al igual que BufferedReader y BufferedWriter nos permiten leer y escribir en los ficheros. En mi caso yo utilice BufferedReader y BufferedWriter.

4). **Java.io.IOException** Una excepción es un evento producido en la ejecución de un programa que interrumpe el flujo normal de instrucciones. Por ejemplo, si manejamos ficheros, es posible que el fichero no exista en la ruta indicada, la sintaxis es correcta, pero en tiempo de ejecución no puede realizar la acción.

5). **Java.util.scanner.** Para poder leer datos del teclado nos apoyamos de esta clase. La clase Scanner de Java provee métodos para leer valores de entrada de varios tipos y está localizada en el paquete java.util. Los valores de entrada pueden venir de varias fuentes, incluyendo valores que se entren por el teclado o datos almacenados en un archivo.

★

```
|1). Creacion de un archivo si
|2). Sobreescritura de archivo
|3). Edicion de archivo
|4). Eliminación de archivo si
|5). Salir
```

• Actividad 3). Creación de un archivo.

Para poder crear un archivo nos ayudamos de la clase io.file;

Como realice un método void para poder realizar utilizarla la instanciamos en el case que se va a utilizar en este caso es el case 1: Para esto se manda la dirección y el nombre. Y creamos el archivo concatenando la dirección más el nombre y la extensión que en este caso es .txt y este lo llamamos en el archivo en donde se va a escribir.

```
void crearArchivo(String direccion,String nombre) throws IOException{
File archivoo =new File(direccion+"\\\\"+nombre+".txt");
System.out.println("Este es el archivo"+archivoo);
FileWriter archivo = new FileWriter (archivoo);
System.out.println(" ESCRIBA EL CONTENIDO");
String cont = sc.next();
archivo.write(cont);
archivo.close();
System.out.println(direccion);

File af = new File(nombre);
if (af.exists()) {
    // etcétera
}else{
    System.out.println(" El archivo se creo c
```

En la ejecución del programa. Se copia tal cual la dirección de donde se encuentra la carpeta en donde queremos que se guarde el archivo. Y en cuanto nos pida el nombre colocamos un nombre sin la extensión, en seguida nos pedirá un texto y colocamos el texto que queremos que se guarde en el archivo.

```
1
Ingresa la direccion: D:\Users\Jonathan\Desktop\2019-2\EDAI\practical0\Practic
Ingresa el nombre del archivo (sin extension): lucas
Este es el archivoD:\Users\Jonathan\Desktop\2019-2\EDAI\practical0\Practical0C
 ESCRIBA EL CONTENIDO
holamundo
D:\Users\Jonathan\Desktop\2019-2\EDAI\practical0\Practical0CalzadMartinez\src
El archivo se creo correctamente
```

• Actividad 4). Sobre-escritura de un archivo.

Para la sobre-escritura del archivo se debe crear un archivo esto debido a que hay la posibilidad de no querer sobrescribir el archivo recién hecho o que no se haya creado un archivo anteriormente, pero este guardado y pasando los mismos parámetros solicitados en el método anterior, con esto verificamos si existe el archivo o no, si existe se sobre escribe en el archivo, en caso de que no exista, se manda un mensaje a pantalla con la leyenda “El archivo no existe”.

```
void sobreEscribir(String direccion,String nombre) throws IOExcept:
File archi =new File(direccion+"\\\\"+nombre+".txt");

if (!archi.exists()) {
    System.out.println("El archivo data
} else {
    BufferedWriter bw = new BufferedWriter(new FileWriter(archi);
    System.out.println(" ESCRIBA EL CONTENIDO");
    String cont = sc.next();
    bw.write(cont);
    bw.close();
```

```
2
Ingresa la direccion: D:\Users\Jonathan\Desktop\2019-2\EDAI\practical0\Practic
Ingresa el nombre del archivo (sin extension): hey
 ESCRIBA EL CONTENIDO
hola
-----
[1]. Creacion de un archivo      |
[2]. Sobreescritura de archivo   |
[3]. Edicion de archivo         |
[4]. Eliminación de archivo     |
[5]. Salir                      |
-----
2
Ingresa la direccion: D:\Users\Jonathan\Desktop\2019-2\EDAI\practical0\Practic
Ingresa el nombre del archivo (sin extension): archi
El archivo data no existe.
-----
[1]. Creacion de un archivo      |
```

- **Actividad 5). Edición de un archivo.**

Se creó un método void en donde recibe el nombre, la dirección.

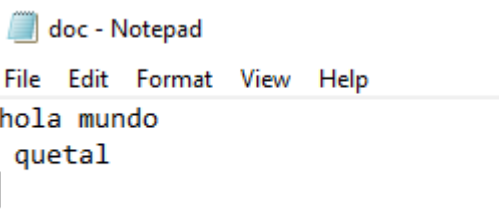
Se creó un archivo con los datos dados y se prueba si existe y en dado caso de que no se manda un mensaje diciendo que el archivo no existe, y si existe use la clase `BufferedWriter` para crear un objeto al que llame `bws` y pedí el contenido que se va a agregar, en seguida pedí que se guardara lo que ya estaba más lo agregado. Y al final lo imprime.

```
void editar(String direc,String nombre) throws IOException{
var archis = new File(direc+"\\ "+nombre+".txt");
    if (!archis.exists()) {
        System.out.println("El archivo data no existe");
    } else {
        FileWriter TextOut = new FileWriter(archis);
        System.out.println("Escriba lo que va a agregar:");
        String agrega = sc.next();
        TextOut.write(" "+agrega+"\n");
        TextOut.close();
    }
}
```

```

-----
|1). Creacion de un archivo      |
|2). Sobreescritura de archivo   |
|3). Edicion de archivo         |
|4). Eliminación de archivo     |
|5). Salir                      |
-----
3
Ingresa la direccion: D:\Users\Jonathan\Desktop\2019-2\
Ingresa el nombre del archivo (sin extension): doc
Escriba lo que va a agregar
quetal
-----

```



- **Actividad 6). Eliminación de un archivo.**

Este método fue sencillo. Al igual que en los pasados pedí el nombre y la dirección donde se aloja el archivo o fichero a eliminar y se verifica si existe o no, si existe solo utilizamos el método de la clase `file delete()` para poder borrar el fichero y mandamos un mensaje diciendo que le archivo fue eliminado.

```
void borrar (String direc,String nombre){

var sfichero = new File(direc+"\\ "+nombre+".txt");
    if (!sfichero.exists()) {
        System.out.println("El archivo data no existe");
    } else {
        sfichero.delete();
        System.out.println("El archivo data fue eliminado");
    }
}
```

```
Ingresa la direccion: D:\Users\Jonathan\Desktop\2019-2\EDAI\practica10\Pr:  
Ingresa el nombre del archivo (sin extension): doc  
El archivo data fue eliminado.  
-----
```

Conclusiones: En general los objetivos de la práctica se cumplieron ya que se pudo realizar con éxito la práctica. Solo está el detalle en el programa de que cuando se ingresa un espacio en el momento en que vamos a poner lo que va a ir dentro del archivo este indica un error.

Con esta práctica reaprendimos lo que ya habíamos visto para poder realizar el proyecto I. Para la realización de la lista de clases que nos permiten manipular archivos/ ficheros se utilizó el API de java.