**GUIA DIDACTICA 1**

**ESTRUCTURAS DE DATOS EXTERNAS (PERSISTENCIA)**

Son aquellas que radican en dispositivos de almacenamiento secundario, tales como: discos duros, memoria flash, discos de estado sólido, discos compactos, cintas magnéticas, etc. Debido a ello, reciben el nombre de “Archivo”.

Existen varias formas de organizar archivos, entre los más comunes tenemos a: Secuenciales, Randómicos y Binarios.

En cada una de estas organizaciones, se pueden efectuar acciones de lectura y escritura, mediante los siguientes procesos: Alta, Baja, Cambio, Listado y Búsqueda.

ALTA.- Significa agregar un nuevo registro o conjunto de caracteres, al archivo.

BAJA.- Significa eliminar un registro, existen dos tipos: lógico (el registro puede ser recuperado) y físico (el registro no puede ser recuperado).

CAMBIO.- Significa, actualizar campos de un registro, excepto el campo llave primaria.

LISTADO.- Significa, ver en pantalla o impresora, registro del archivo.

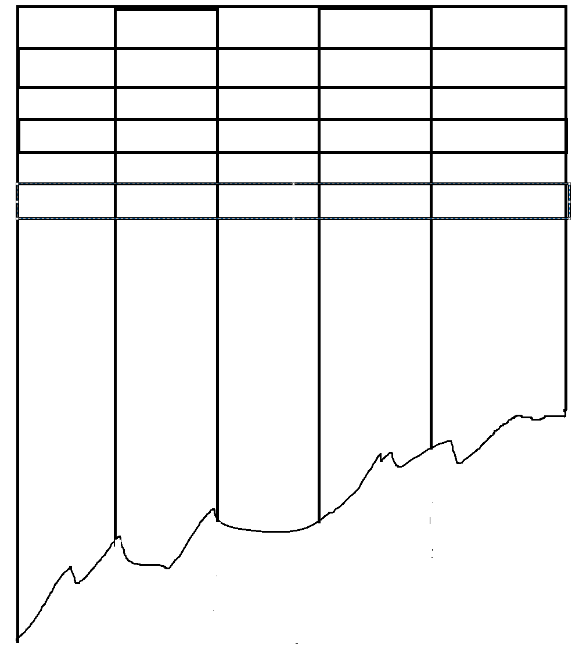
BÚSQUEDA.- Significa, ubicar algún registro en base a su campo llave primaria o en base a otro criterio.

VISTA GRÁFICA DE UN ARCHIVO

Las columnas son conocidas como: campos

Se lo puede ver como una matriz, con los siguientes componentes:

Campo llave



Las filas son conocidas como: registros

CAMPOS.- Son la columnas que contienen un determinado tipo de dato o información y está formada por un conjunto de caracteres.

CAMPO LLAVE.- Es aquel campo que permite identificar de manera única a algún registro.

REGISTROS.- Son las filas formadas por campos.

ARCHIVOS SECUENCIALES

Son estructuras organizadas como una cadena infinita de caracteres, por lo que debemos darnos modos para formar los registros y sus campos.

CARACTERISTICAS DE LOS ARCHIVOS SECUENCIALES

* Cada nuevo registro, siempre se graba o almacena al final del archivo.
* La lectura siempre se lo debe realizar desde el principio.
* Para detectar el final del archivo, se debe verificar si en el carácter leído salió -1 (menos uno).
* Este tipo de organización es recomendable cuando no hay mucha frecuencia de cambio o baja en la gestión.

ARCHIVOS SECUENCIALES EN JAVA

CLASES PARA ADMINISTRAR ARCHIVOS SECUENCIALES

FileOutputStream

Esta clase permite instanciar(crear) objetos, cuyos métodos permiten grabar datos en el archivo.

FORMA DE INSTANCIARLO

FileOutputStream nombreObjeto = new FileOutputStream***(”ruta y archivo”, tipo***);

Donde:

* ***“ruta y archivo”***.- es el camino del directorio donde se encuentra el archivo, donde se grabaran los datos enviados.
* ***tipo***.- Puede ser true o false, en caso de colocar true, los nuevos registros se irán agregando al final del archivo; en caso de colocar false, los anteriores registros serán eliminados y solo aparecerá el último registro grabado.

MÉTODOS DE FileOutputStream

* .write() .- Envía el dato al archivo.
* .close().- Cierra la comunicación con el archivo.

FileInputStream

Esta clase permite el recorrido y la lectura de registros del archivo.

FORMA DE INSTANCIARLO

FileInputStream nombreObjeto = new FileInputStream***(”ruta y archivo”***);

Donde:

* ***“ruta y archivo”***.- es el camino del directorio donde se encuentra el archivo a ser leido.

MÉTODOS DE FileInputStream

* .read() .- Lee carácter por carácter los registros del archivo.
* .close() .- Cierra la comunicación con el archivo.

EJEMPLO

Realizaremos un ejemplo, donde almacenaremos en cada registro un sólo campo, que será el nombre completo de una persona.

A continuación, les presento el código fuente del problema propuesto

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** javax.swing.JOptionPane;

**public** **class** UnoSec {

**public** **static** **void** main(String[] args) {

**int** opcion = 0;

**while**(opcion != 3) {

opcion = *menu*();

**switch**(opcion) {

**case** 1:

*alta*();

**break**;

**case** 2:

*lista*();

**break**;

}

}

}

**static** **int** menu() {

**int** ele=-1;

**while**(ele<1 || ele>3) {

String m = "MENU DE OPCIONES \n 1. Alta \n 2. Lista \n 3. Salir";

ele = Integer.*parseInt*(JOptionPane.*showInputDialog*(m));

}

**return** ele;

}

**static** **void** alta() {

**try** {

String data = JOptionPane.*showInputDialog*("Ingrese nombre completo: ");

FileOutputStream fout = **new** FileOutputStream("g:/daka/datos.txt",**true**);

**byte** cb[];

cb = data.getBytes();

fout.write(cb);

fout.write('\n');

fout.close();

System.***out***.println("el registro fue grabado exitosamente");

}**catch** (Exception e) {

System.***out***.println("Error" + e.getMessage());

}

}

**static** **void** lista() {

**try** {

FileInputStream fin = **new** FileInputStream("g:/daka/datos.txt");

System.***out***.println("Listado del archivo");

String regs = "";

String reg = "";

**int** c;

**while**( (c=fin.read()) != -1 ) {

**if**(c!='\n') {

reg = reg + (**char**)c;

}**else** {

regs = regs + reg + "\n";

reg = "";

}

}

JOptionPane.*showMessageDialog*(**null**, regs);

fin.close();

}**catch** (Exception e) {

System.***out***.println("Error" + e.getMessage());

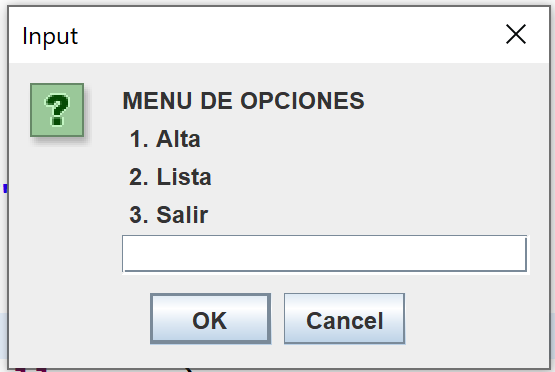
}

}

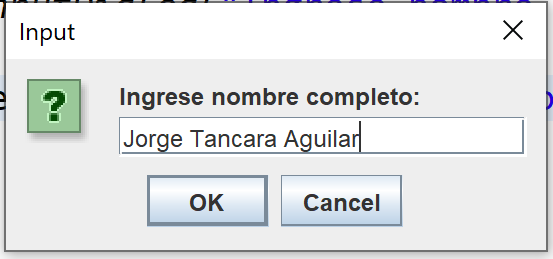
}

Ejecución del programa

Lo primero que se debe hacer es, crear la carpeta o la ruta en el disco duro, donde aparecerá el archivo, en el ejemplo yo uso la siguiente ruta: g:/daka/

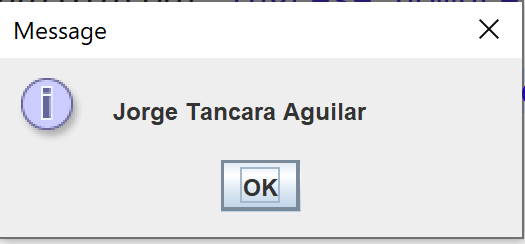


Opción uno, agregar un nuevo nombre (alta),



Para salir hacer clic sobre OK

Opción dos, ver el listado de nombres (lista),



Para salir hacer clic sobre OK