**Guía Manejo de Archivos Secuenciales**

**Asignatura: Programación II**

**Docente: José Souza Ordenes**

La presente guía hace uso de archivos secuenciales grabando y recuperando registros lógicos.

Registro Lógico: conjunto de campos relacionados lógicamente entre sí. Los campos pueden ser de diferente

tipo de datos.

Archivo: Conjunto de registros lógicos con los mismos tipos de datos.

Ejemplo de registro lógico:

1. Antecedente de cliente: Rut, Nombre, Dirección, Celular.
2. Control de Asistencia: Rut, Nombre, Fecha, Hora\_Entrada, Hora\_Salida.
3. Datos Persona: (int clave, String nombre, byte edad, float estatura)

(Se podría cambiar: int edad, int estatura)

**Campos del registro lógico del archivo:**

* **int clave**
* **String nombre**
* **byte edad**
* **float estatura**

package archivossecuenciales;

/\*\*

\* @author jose

\*/

import java. util.Scanner;

import java.io.\*;

public class ArchivosSecuenciales {

**public static boolean verificarExistencia(String nombreArchivo){**

**//Este método solo se encarga de verificar por un archivo; indica si existe o no.**

**File archivo = new File(nombreArchivo);**

**if(archivo.exists())**

**return true;**

**else**

**return false;**

**}**

**public static boolean crearArchivo(String nombreArchivo){**

**//Este método solo crear el archivo**

**try{**

**File archivo = new File(nombreArchivo);**

**archivo.createNewFile();**

**return true;**

**}**

**catch(IOException crearArchivo){**

**System.out.print("Ocurrió un error al crear el archivo\n");**

**return false;**

**}**

**}**

**public static boolean insertarDatos(String nombreArchivo, int clave,String nombre,byte edad, float estatura){**

**try{**

**//Apertura de streams que permitirá la comunicacion de la aplicación,**

**// indican que la aplicación mandara datos a disco.**

**// out y Dout son simplemente instancias; estos pueden ser nombrados de cualquier forma**

**FileOutputStream out= new FileOutputStream(nombreArchivo,true);**

**DataOutputStream Dout= new DataOutputStream(out);**

**//valores a insertar**

**Dout.writeInt(clave);**

**Dout.writeUTF(nombre);**

**Dout.writeByte(edad);**

**Dout.writeFloat(estatura);**

**return true;**

**}**

**catch(IOException Entrada){**

**System.out.print("\nOcurrio una falla al ingresar los datos al archivo\n");**

**return false;**

**}**

**}**

public static void leerDelArchivo(String nombreArchivo){

try{

//Apertura de streams que permitiran la comunicacion de la aplicacion

//con el sistema; indican que la aplicacion extraera datos de disco.

//in y Din son simplemente instancias; estos pueden ser nombrados

//de cualquier forma

FileInputStream in= new FileInputStream(nombreArchivo);

DataInputStream Din=new DataInputStream(in);

//La lectura se debe hacer de forma ordenada, el primer dato que se ingreso fue un entero, el primer dato

//a leer será un entero. Y así con los demás datos.

/\* Se puede asignar la lectura a una variable del tipo correspondiente o bien se puede realizar la impresión

de forma directa. \*/

try{

do{

System.out.print("\n Cliente: ");

System.out.print("\n"+Din.readInt());

System.out.print("\n"+Din.readUTF());

System.out.print("\n"+Din.readByte());

System.out.print("\n"+Din.readFloat());

}while(true);

}

catch(EOFException err\_file){

System.out.print("\nOcurrio una falla al extraer los datos del archivo, está vacío");

}

}

catch(IOException extraccion){

System.out.print("\nOcurrio una falla al extraer los datos del archivo");

}

}

**public static void consultaIndividual(String nombreArchivo, int clave){**

**int cve=0;**

**String nombre=null;**

**byte edad=0;**

**float estatura=0.0f;**

**try{**

**FileInputStream in= new FileInputStream(nombreArchivo);**

**DataInputStream Din=new DataInputStream(in);**

**try{**

**do{**

**cve=Din.readInt();**

**nombre=Din.readUTF();**

**edad=Din.readByte();**

**estatura=Din.readFloat();**

**if(cve==clave){**

**System.out.print("\nLa clave registrada es: "+cve);**

**System.out.print("\nEl nombre registrado es: "+nombre);**

**System.out.print("\nLa edad registrada es: "+edad);**

**System.out.print("\nLa estatura registrada es: "+estatura);**

**}**

**}while(cve!=clave);**

**}**

**catch(EOFException f){**

**System.out.print("La clave buscada no se encuentra en el archivo\n");**

**}**

**}**

**catch(IOException Entrada){**

**System.out.print("\nOcurrio una falla al extraer los datos del archivo\n");**

**}**

**}**

**public static boolean solicitudDatos(String nombreArchivo){**

**Scanner sc = new Scanner(System.in);**

**String nombre=null;**

**int clave=0;**

**byte edad=0;**

**float estatura=0.0f;**

**System.out.print("\nIndique una clave: ");**

**clave=sc.nextInt();**

**System.out.print("\nIndique su nombre: ");**

**sc.nextLine(); //limpiar buffer**

**nombre=sc.nextLine();**

**System.out.print("\nIndique su edad: ");**

**edad=sc.nextByte();**

**System.out.print("\n Indique su estatura: ");**

**estatura=sc.nextFloat();**

**if(insertarDatos(nombreArchivo, clave, nombre, edad, estatura)== true)**

**return true;**

**else**

**return false;**

**}**

**// Programa Principal**

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String nombreArchivo=null;

byte opt=0;//variable que permitirá controlar el menú

**do**{

System.out.println("\n\t\t\tMENU DE SELECCION");

System.out.println("\t\tELIJA UNA DE LAS SIGUIENTES OPCIONES");

System.out.println("1)VERIFICAR UN ARCHIVO (VER SI EXISTE UN ARCHIVO)");

System.out.println("2)CREAR UN ARCHIVO");

System.out.println("3)INSERTAR DATOS A UN ARCHIVO");

System.out.println("4)CONSULTA DE ARCHIVO");

System.out.println("5)CONSULTA INDIVIDUAL");

System.out.println("6)SALIR");

opt=sc.nextByte();

switch(opt){

case 1:

System.out.print("Indique el nombre del archivo (con su extension) a buscar: ");

nombreArchivo=sc.next();

if(verificarExistencia(nombreArchivo)== true)

System.out.print("\nEl archivo ya existe");

else

System.out.print("\nEl archivo no existe");

break;

case 2:

System.out.print("Indique el nombre del archivo (con su extension) a crear: ");

nombreArchivo=sc.next();

if(crearArchivo(nombreArchivo)==true)

System.out.print("\n El archivo ha sido creado satisfactoriamente");

break;

case 3:

System.out.print("Indique el nombre del archivo (con su extensión) en el que insertara datos: ");

nombreArchivo=sc.next();

if(solicitudDatos(nombreArchivo)==true)

System.out.print("Los datos han sido insertados correctamente ");

break;

case 4:

System.out.print("Indique el nombre del archivo con su extension a visualizar: ");

nombreArchivo=sc.next();

leerDelArchivo(nombreArchivo);

break;

case 5:

int clave=0;

System.out.print("Indique el nombre del archivo con su extension a visualizar: ");

nombreArchivo=sc.next();

System.out.print("Indique la clave a buscar: ");

clave=sc.nextInt();

consultaIndividual(nombreArchivo,clave);

break;

}

}**while(opt!=6);**

}

}

**Ejercicio: Propuesto**

Crear un archivo factura con los siguientes campos:

* F olio (String)
* Fecha (String)
* Cliente (String)
* Descripcion (String)
* Neto (int)
* Iva (int) Campo Calculado Iva = Neto \* 0.19
* Total (int) Campo Calculado Total = Neto + Iva
* Estado\_pago (String) Campo lógico:

pagado = ”si”

no pagado = “no”

Validar el ingreso de pago

**Además se requiere:**

* Listar el archivo completo
* Aceptar un Cliente ,

Si existe:

Listar todas las facturas que tiene.

Si no existe

Desplegar mensaje de error

* Listar todas las facturas no pagadas, Contar y sumar el total adeudado.
* Listar todas las facturas pagadas y listar el total de pagos efectuados.