****

**Programación orientada a objetos III**

Ingeniería en Desarrollo de Software

Unidad: 1. Archivos

Actividad. Evidencia de Aprendizaje

Francisco Manuel López García

Matrícula: ES1921010154

Profesor: Manuel Niño Navarrete

23 de octubre 2021

Índice

[**Introducción:** 3](#_Toc85860036)

[**Contenido:** 3](#_Toc85860037)

[*Investigaciones:* 3](#_Toc85860038)

[*Capturas de pantalla:* 5](#_Toc85860039)

[*Código fuente:* 12](#_Toc85860040)

[Clase Principal: 12](#_Toc85860041)

[Clase FramePrincipal: 13](#_Toc85860042)

[Clase Personal: 14](#_Toc85860043)

[Clase Inventario: 18](#_Toc85860044)

[*Análisis:* 21](#_Toc85860045)

[**Conclusión:** 22](#_Toc85860046)

[**Bibliografía:** 23](#_Toc85860047)

**Evidencia de Aprendizaje**

# **Introducción:**

Con esta actividad se busca reforzar la aplicación realizada en la actividad anterior, donde agregaremos métodos de manejos de archivos, para eliminar y renombrar archivos, por ello retomaremos los flujos de entrada y salida.  
Los datos fluyen a través de nuestra computadora desde una entrada hasta una salida/ Este flujo de datos se conoce como stream. Existe un flujo de entrada que se encarga de recibir los datos desde el exterior y un flujo de salida que envía los datos hacia los dispositivos o bases de datos.

El proceso de escritura o lectura se puede sintetizar en tres simples pasos:

1. Abrir flujo de datos
2. Siempre y cuando exista información, leer o escribir los datos
3. Cerrar el flujo de datos

En Java existen distintas clases para estos flujos, como son:

* InputStream: se encarga de establecer el flujo de bytes de entrada
* Reader: Es parecida a la anterior solo que en vez de bytes en este caso son char.
* OutputStream: Se encarga del flujo de escritura de bytes.
* Writer: Es la clase que se encarga del flujo de salida de caracteres.

Con estas clases podemos realizar al entrada y salida de datos, solo será necesario determinar si la entrada o salida será en bytes o char.

# **Contenido:**

## *Investigaciones:*

**Eliminar Fichero**  
Borrar un fichero con Java es una actividad bastante sencilla, ya que mediante la clase File, la cual nos abstrae de la manipulación de ficheros, se nos ofrece un método para llevar a cabo tal cometido.

Lo primero que deberemos de hacer es instanciar una clase File. Para instanciar esta clase deberemos de pasar al constructor el nombre del fichero sobre el que queremos trabajar.

*File fichero = new File("fichero.txt");*

Para borrar el fichero deberemos de invocar el método .delete() de la clase File. En caso de que se pueda realizar el borrado del fichero, dicho método devolverá true. En caso contrario devolverá false. Es por ello que deberemos de controlar su respuesta.

*if (fichero.delete())*

*System.out.println("El fichero ha sido borrado satisfactoriamente");*

*else*

*System.out.println("El fichero no puede ser borrado");*

**Renombrar Fichero**

Entre las muchas operativas que se pueden hacer sobre un fichero tenemos el renombrado de un fichero. Para ello la librería de Java, Java IO, nos ofrece un método sobre la clase File.

Si recordamos un poco la filosofía de la clase Java IO veremos que la clase File representa la abstracción de un fichero o directorio independientemente del sistema de ficheros que tengamos por debajo. Lo primero que tenemos que hacer es instanciar la clase File sobre el fichero que queramos realizar el renombrado. Para ello utilizamos el constructor de la clase File, como podemos ver en la siguiente línea de código:

*File f1 = new File("fichero1.txt");*

Lo siguiente será instanciar una nueva clase File con el nombre del fichero que queramos poner. Veamos la línea de código:

*File f2 = new File("fichero2.txt");*

Con las dos abstracciones de los ficheros solo nos quedará el ejecutar el método .renameTo(File) sobre el primer fichero. El método .renameTo(File) recibe como parámetro un objeto File con el nuevo nombre. El que nosotros hemos llamado f2

*boolean correcto = f1.renameTo(f2);*

El método .renameTo(File) devuelve un valor booleano indicando si se ha podido realizar el renombrado, o no. Así que para finalizar validaremos la variable booleana con el fin de dar información al usuario.

*if (correcto)*

*System.out.println("El renombrado ha sido correcto");*

*else*

*System.out.println("El renombrado no se ha podido realizar");*

## *Capturas de pantalla:*

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Inicio de la aplicación

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Modulo Personal-Altas

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Se ingresan datos y se pulsa guardar, se muestra una ventana que pide el nombre del archivo se ingresa “Prueba1”

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Se pulsa botón Buscar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Se selecciona el archivo antes creado “Prueba1”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
Nos muestra la información en el TextArea

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Pulsamos botón Renombrar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
Seleccionamos el archivo “Prueba1”

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente  
Nos pide el nuevo nombre en una ventana, con lo cual colocamos “PruebaNuevoNombre

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente  
Nos muestra una ventana que se renombro

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
Pulsamos el botón Eliminar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
Como podemos observar si se cambio el nombre del archivo, procedemos a seleccionarlo para eliminarlo

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Nos pregunta si estamos seguros de eliminarlos, pulsamos SI  
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
Nos muestra que se elimino

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
Como podemos observar el archivo ya no existe

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
Pulsamos el botón de Salir

## *Código fuente:*

### Clase Principal:

*package abarrotes;*

*public class Abarrotes {*

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

*framePrincipal fp; //Creación del objeto de tipo FramePrincipal que es el frame principal*

*fp = new framePrincipal();*

*fp.aparienciaTotal(); //Llamada a la funcion de la apariencia del frame principal*

*}*

*}*

### Clase FramePrincipal:

*package abarrotes;*

*//Se importan las librerias necesarias*

*import java.awt.BorderLayout;*

*import java.awt.event.ActionEvent;*

*import java.awt.event.WindowAdapter;*

*import java.awt.event.WindowEvent;*

*import javax.swing.JFrame;*

*import javax.swing.JLabel;*

*import javax.swing.JMenu;*

*import javax.swing.JMenuBar;*

*import javax.swing.JMenuItem;*

*import javax.swing.JPanel;*

*public class framePrincipal {*

*static JFrame frame = new JFrame("Abarrotes"); //Creación del frame*

*JPanel pnlInicio = new javax.swing.JPanel(); //Creación del panel de inicio*

*static Personal p = new Personal(); //Creación del objeto tipo Personal para utilizar las funciones de esta clase*

*static Inventario i = new Inventario(); //Creación del objeto tipo Inventario para utilizar las funciones de esta clase*

*public void limpiarFrame(){*

*frame.remove(p.pnlPers);*

*frame.remove(i.pnlInv);*

*frame.remove(pnlInicio);*

*}*

*public void aparienciaTotal(){*

*//Se inicializan las apariencias*

*p.aparienciaAltasPersonal();*

*i.aparienciaAltasInventario();*

*//Se inicializan los componentes del frame*

*pnlInicio.setBorder(javax.swing.BorderFactory.createTitledBorder("Abarrotes"));*

*JLabel ini = new javax.swing.JLabel();*

*ini.setText("Inicio");*

*pnlInicio.add(ini);*

*frame.setLocation(400,200);*

*frame.setVisible(true);*

*frame.setSize(600,500);*

*//Funcion para cerrar el programa con el evento WindowClosing*

*frame.addWindowListener(new WindowAdapter() {*

*@Override*

*public void windowClosing(WindowEvent e){*

*System.exit(0);*

*}*

*});*

*//Creando el Menu y su menuItem Inicio*

*JMenuBar menu = new javax.swing.JMenuBar();*

*JMenuItem Inicio = new javax.swing.JMenuItem("Inicio");*

*Inicio.addActionListener((ActionEvent e)-> {*

*if(e.getSource()==Inicio){*

*frame.setTitle("Abarrotes");*

*limpiarFrame();*

*frame.add(pnlInicio,BorderLayout.CENTER);*

*frame.repaint();*

*frame.validate();*

*}*

*});*

*//Creando el Menu Personal y el MenuItem Altas*

*JMenu Personal = new javax.swing.JMenu("Personal");*

*JMenuItem AltasPersonal = new javax.swing.JMenuItem("Altas");*

*Personal.add(AltasPersonal);*

*AltasPersonal.addActionListener((ActionEvent e) -> {*

*if(e.getSource()==AltasPersonal){*

*frame.setTitle("Abarrotes - Personal");*

*limpiarFrame();*

*frame.add(p.pnlPers,BorderLayout.CENTER);*

*frame.repaint();*

*frame.validate();*

*}*

*});*

*//Creando el Menu Inventario y el MenuItem Altas*

*JMenu Inventario = new javax.swing.JMenu("Inventario");*

*JMenuItem AltasInventario = new javax.swing.JMenuItem("Altas");*

*Inventario.add(AltasInventario);*

*AltasInventario.addActionListener((ActionEvent e) -> {*

*if(e.getSource()==AltasInventario){*

*frame.setTitle("Abarrotes - Inventario");*

*limpiarFrame();*

*frame.add(i.pnlInv,BorderLayout.CENTER);*

*frame.repaint();*

*frame.validate();*

*}*

*});*

*//Se añaden los componentes al menu y a su vez se añade el menu al frame*

*menu.add(Inicio);*

*menu.add(Personal);*

*menu.add(Inventario);*

*frame.add(menu, BorderLayout.PAGE\_START);*

*frame.add(pnlInicio,BorderLayout.CENTER);*

*frame.repaint();*

*frame.validate();*

*}*

*}*

### Clase Personal:

*package abarrotes;*

*//Se importan las librerias correspondientes*

*import java.awt.GridBagConstraints;*

*import java.awt.GridBagLayout;*

*import java.awt.Insets;*

*import java.awt.event.\*;*

*import java.io.BufferedReader;*

*import java.io.BufferedWriter;*

*import java.io.File;*

*import java.io.FileNotFoundException;*

*import java.io.FileReader;*

*import java.io.FileWriter;*

*import java.io.IOException;*

*import java.util.logging.Level;*

*import java.util.logging.Logger;*

*import javax.swing.\*;*

*public class Personal {*

*JPanel pnlPers = new javax.swing.JPanel();*

*JLabel lblNombre = new javax.swing.JLabel();*

*JTextField txtNombre = new javax.swing.JTextField();*

*JLabel lblEdad = new javax.swing.JLabel();*

*JTextField txtEdad = new javax.swing.JTextField();*

*JLabel lblTel = new javax.swing.JLabel();*

*JTextField txtTel = new javax.swing.JTextField();*

*JButton btnGuardar = new javax.swing.JButton();*

*JButton btnBuscar = new javax.swing.JButton();*

*JButton btnEliminar = new javax.swing.JButton();*

*JButton btnRenom = new javax.swing.JButton();*

*JButton btnSalir = new javax.swing.JButton();*

*JTextArea txtA = new javax.swing.JTextArea();*

*JScrollPane scroll = new JScrollPane();*

*public void aparienciaAltasPersonal(){*

*//Configuración de los componentes*

*pnlPers.setBorder(javax.swing.BorderFactory.createTitledBorder("Personal-Altas"));*

*lblNombre.setText("Ingresa el nombre: ");*

*lblEdad.setText("Ingresa la edad: ");*

*lblTel.setText("Ingresa el teléfono: ");*

*txtA.setBounds(10,50,200,200); // asignación de tamaño del TextArea*

*txtA.setColumns(50); //asignación del máximo de columnas del TextArea*

*txtA.setRows(20); //asignación del máximo de filas del TextArea*

*scroll.setViewportView(txtA); //se añade el TextArea a un scrollPane*

*btnGuardar.setText("Guardar");*

*//Función para el evento Clicked para llamar a la función Guardar*

*btnGuardar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Guardar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnBuscar.setText("Buscar");*

*//Función para el evento Clicked para llamar a la función Buscar*

*btnBuscar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Buscar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnEliminar.setText("Eliminar");*

*//Funcion para el evento Clicked para llamar a la función Eliminar*

*btnEliminar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Eliminar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnRenom.setText("Renombrar");*

*//Funcion para el evento Clicked para llamar a la función Renombrar*

*btnRenom.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Renombrar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnSalir.setText("Salir");*

*//Función para el evento Clicked para llamar a la función Salir*

*btnSalir.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*System.exit(0);*

*}*

*});*

*//Acomodo de los componentes en el Panel con GridBagLayaout*

*GridBagLayout gbl = new GridBagLayout();*

*GridBagConstraints gbc = new GridBagConstraints();*

*pnlPers.setLayout(gbl);*

*gbc.weightx = 1;*

*gbc.weighty = 1;*

*gbc.fill = GridBagConstraints.HORIZONTAL;*

*gbc.insets = new Insets(0,5,0,0);*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlPers.add(lblNombre, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlPers.add(txtNombre, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlPers.add(lblEdad, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlPers.add(txtEdad, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlPers.add(lblTel, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlPers.add(txtTel, gbc);*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlPers.add(btnGuardar, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlPers.add(btnBuscar, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlPers.add(btnEliminar, gbc);*

*gbc.gridwidth = GridBagConstraints.CENTER;*

*pnlPers.add(btnRenom, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlPers.add(btnSalir, gbc); gbc.ipady = 250;*

*pnlPers.add(scroll, gbc);*

*}*

*public void Guardar() throws FileNotFoundException, IOException{*

*String nombreArchivo = JOptionPane.showInputDialog(pnlPers,"Ingresa el nuevo nombre del archivo:","Guardar",1); //Se muestra ventana Emergente para pedir el nuevo nombre del archivo*

*String carpeta = System.getProperty("user.dir"); //Con esto se pone de ruta en donde se ejecuta el programa*

*String url = carpeta + "/" + nombreArchivo + ".txt";*

*FileWriter ubicacion = null;*

*try {*

*ubicacion = new FileWriter(url); //Se realiza un try de la ubicación del archivo*

*}catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*try{*

*BufferedWriter escribir = new BufferedWriter(ubicacion); // se abre el buffer de salida*

*escribir.write(txtNombre.getText()+" / "+txtEdad.getText()+" / "+txtTel.getText()); // se escribe en el archivo*

*escribir.close(); // se cierra el buffer*

*}catch(Exception e){*

*}*

*}*

*public void Buscar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*try{*

*FileReader fr = new FileReader(archivo); //Lectura del archivo*

*BufferedReader br = new BufferedReader(fr); //Creación del buffer de lectura*

*String texto = ""; //Variable que guardara todo el texto*

*String linea = ""; //Variale que guardara linea por linea*

*while(((linea=br.readLine())!=null)){ //Con esto se ejecutara y se almacenara*

*texto+=linea+"\n"; //las lineas y posteriormente se agregaran a la variable texto*

*}*

*txtA.setText(texto); //Se asigna el texto al textArea*

*}*

*catch(Exception e){*

*}*

*}*

*public void Eliminar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*if(JOptionPane.showConfirmDialog(pnlPers,"¿Seguro que quieres eliminar este archivo?","Eliminar",JOptionPane.YES\_NO\_OPTION)==0){ //Se muestra una ventana emergente para verificar si se eliminara ese archivo*

*if(archivo.delete())*

*JOptionPane.showMessageDialog(null,"Se elimino el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

*}*

*public void Renombrar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*String nombreArchivo = JOptionPane.showInputDialog(pnlPers,"Ingresa el nuevo nombre del archivo:","Guardar",1); //Se muestra ventana Emergente para pedir el nuevo nombre del archivo*

*String nvo = archivo.getParent()+ "\\" + nombreArchivo+ ".txt";*

*File a2 = new File(nvo); //Se crea un nuevo archivo de tipo File*

*if(archivo.renameTo(a2)) //Se Valida que se renombre el archivo por a2*

*JOptionPane.showMessageDialog(null,"Se renombro el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

*}*

### Clase Inventario:

*package abarrotes;*

*//Se importan las librerias correspondientes*

*import java.awt.GridBagConstraints;*

*import java.awt.GridBagLayout;*

*import java.awt.Insets;*

*import java.awt.event.\*;*

*import java.io.BufferedReader;*

*import java.io.BufferedWriter;*

*import java.io.File;*

*import java.io.FileNotFoundException;*

*import java.io.FileReader;*

*import java.io.FileWriter;*

*import java.io.IOException;*

*import java.util.logging.Level;*

*import java.util.logging.Logger;*

*import javax.swing.\*;*

*public class Inventario {*

*JPanel pnlInv = new javax.swing.JPanel();*

*JLabel lblMarca = new javax.swing.JLabel();*

*JTextField txtMarca = new javax.swing.JTextField();*

*JLabel lblProducto = new javax.swing.JLabel();*

*JTextField txtProducto = new javax.swing.JTextField();*

*JLabel lblPrecio = new javax.swing.JLabel();*

*JTextField txtPrecio = new javax.swing.JTextField();*

*JButton btnGuardar = new javax.swing.JButton();*

*JButton btnBuscar = new javax.swing.JButton();*

*JButton btnEliminar = new javax.swing.JButton();*

*JButton btnRenom = new javax.swing.JButton();*

*JButton btnSalir = new javax.swing.JButton();*

*JTextArea txtA = new javax.swing.JTextArea();*

*JScrollPane scroll = new JScrollPane();*

*public void aparienciaAltasInventario(){*

*//Configuración de los componentes*

*pnlInv.setBorder(javax.swing.BorderFactory.createTitledBorder("Inventario-Altas"));*

*lblMarca.setText("Ingresa el nombre de la marca: ");*

*lblProducto.setText("Ingresa el nombre del producto: ");*

*lblPrecio.setText("Ingresa el precio: ");*

*txtA.setBounds(10,50,200,200); // asignación de tamaño del TextArea*

*txtA.setColumns(50); // asignación del máximo de columnas del TextArea*

*txtA.setRows(20); // asignación del máximo de filas del TextArea*

*scroll.setViewportView(txtA); // se añade el TextArea a un scrollPane*

*btnGuardar.setText("Guardar");*

*//Función para el evento Clicked para llamar a la función Guardar*

*btnGuardar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Guardar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnBuscar.setText("Buscar");*

*//Función para el evento Clicked para llamar a la función Buscar*

*btnBuscar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Buscar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnEliminar.setText("Eliminar");*

*//Funcion para el evento Clicked para llamar a la función Eliminar*

*btnEliminar.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Eliminar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnRenom.setText("Renombrar");*

*//Funcion para el evento Clicked para llamar a la función Renombrar*

*btnRenom.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*try {*

*Renombrar();*

*} catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*}*

*});*

*btnSalir.setText("Salir");*

*//Función para el evento Clicked para llamar a la función Salir*

*btnSalir.addMouseListener(new MouseAdapter(){*

*@Override*

*public void mouseClicked(MouseEvent e){*

*System.exit(0);*

*}*

*});*

*//Acomodo de los componentes en el Panel con GridBagLayaout*

*GridBagLayout gbl = new GridBagLayout();*

*GridBagConstraints gbc = new GridBagConstraints();*

*pnlInv.setLayout(gbl);*

*gbc.weightx = 1;*

*gbc.weighty = 1;*

*gbc.fill = GridBagConstraints.HORIZONTAL;*

*gbc.insets = new Insets(0,5,0,0);*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlInv.add(lblMarca, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlInv.add(txtMarca, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlInv.add(lblProducto, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlInv.add(txtProducto, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlInv.add(lblPrecio, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlInv.add(txtPrecio, gbc);*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlInv.add(btnGuardar, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlInv.add(btnBuscar, gbc);*

*gbc.gridwidth = 1;*

*gbc.gridwidth = GridBagConstraints.WEST;*

*pnlInv.add(btnEliminar, gbc);*

*gbc.gridwidth = GridBagConstraints.CENTER;*

*pnlInv.add(btnRenom, gbc);*

*gbc.gridwidth = GridBagConstraints.REMAINDER;*

*pnlInv.add(btnSalir, gbc); gbc.ipady = 250;*

*pnlInv.add(scroll, gbc);*

*}*

*public void Guardar() throws FileNotFoundException, IOException{*

*String nombreArchivo = JOptionPane.showInputDialog(pnlInv,"Ingresa el nombre del archivo:","Guardar",1); //Se muestra ventana Emergente para pedir el nombre del archivo*

*String carpeta = System.getProperty("user.dir"); //Con esto se pone de ruta en donde se ejecuta el programa*

*String url = carpeta + "/" + nombreArchivo + ".txt";*

*FileWriter ubicacion = null;*

*try {*

*ubicacion = new FileWriter(url); //Se realiza un try de la ubicacion del archivo*

*}catch (IOException ex) {*

*Logger.getLogger(Personal.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*try{*

*BufferedWriter escribir = new BufferedWriter(ubicacion); // se abre el buffer de salida*

*escribir.write(txtMarca.getText()+" / "+txtProducto.getText()+" / "+txtPrecio.getText()); // se escribe en el archivo*

*escribir.close(); // se cierra el buffer*

*}catch(Exception e){*

*}*

*}*

*public void Buscar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*try{*

*FileReader fr = new FileReader(archivo); //Lectura del archivo*

*BufferedReader br = new BufferedReader(fr); //Creación del buffer de lectura*

*String texto = ""; //Variable que guardara todo el texto*

*String linea = ""; //Variale que guardara linea por linea*

*while(((linea=br.readLine())!=null)){ //Con esto se ejecutara y se almacenara*

*texto+=linea+"\n"; //las líneas y posteriormente se agregaran a la variable texto*

*}*

*txtA.setText(texto); //Se asigna el texto al textArea*

*}*

*catch(Exception e){*

*}*

*}*

*public void Eliminar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*if(JOptionPane.showConfirmDialog(pnlInv,"¿Seguro que quieres eliminar este archivo?","Eliminar",JOptionPane.YES\_NO\_OPTION)==0){ //Se muestra una ventana emergente para verificar si se eliminara ese archivo*

*if(archivo.delete())*

*JOptionPane.showMessageDialog(null,"Se elimino el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

*}*

*public void Renombrar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*String nombreArchivo = JOptionPane.showInputDialog(pnlInv,"Ingresa el nuevo nombre del archivo:","Guardar",1); //Se muestra ventana Emergente para pedir el nuevo nombre del archivo*

*String nvo = archivo.getParent()+ "\\" + nombreArchivo+ ".txt";*

*File a2 = new File(nvo); //Se crea un nuevo archivo de tipo File*

*if(archivo.renameTo(a2)) //Se Valida que se renombre el archivo por a2*

*JOptionPane.showMessageDialog(null,"Se renombro el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

*}*

## *Análisis:*

Se crearon dos métodos, uno para poder eliminar y otro para poder renombrar los archivos, los cuales son:

*public void Eliminar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*if(JOptionPane.showConfirmDialog(pnlInv,"¿Seguro que quieres eliminar este archivo?","Eliminar",JOptionPane.YES\_NO\_OPTION)==0){ //Se muestra una ventana emergente para verificar si se eliminara ese archivo*

*if(archivo.delete())*

*JOptionPane.showMessageDialog(null,"Se elimino el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

*}*

Como podemos observar en este método de Eliminar, se lee un archivo como anteriormente, con un If mostrando un cuadro de dialogo se pregunta si esta seguro de eliminar ese archivo, si la opción es si, se entra en otro If donde se valida si el archivo se pudo eliminar, si es así se muestra una ventana con el mensaje de que se eliminó, si no se muestra una ventana de error.

Nuestro segundo método:

*public void Renombrar() throws FileNotFoundException{*

*JFileChooser fc = new JFileChooser(); //Creación del objeto FileChooser para seleccionar archivos*

*fc.showOpenDialog(null);*

*File archivo = fc.getSelectedFile(); //Creación del objeto que guardara el archivo en memoria RAM*

*String nombreArchivo = JOptionPane.showInputDialog(pnlInv,"Ingresa el nuevo nombre del archivo:","Guardar",1); //Se muestra ventana Emergente para pedir el nuevo nombre del archivo*

*String nvo = archivo.getParent()+ "\\" + nombreArchivo+ ".txt";*

*File a2 = new File(nvo); //Se crea un nuevo archivo de tipo File*

*if(archivo.renameTo(a2)) //Se Valida que se renombre el archivo por a2*

*JOptionPane.showMessageDialog(null,"Se renombro el archivo");*

*else*

*JOptionPane.showMessageDialog(null,"Ocurrio un error");*

*}*

Como podemos observar en este método de Renombrar, se lee un archivo como anteriormente, en un cuadro de dialogo se pide el nuevo nombre, se crea un nuevo objeto de tipo File con el nuevo nombre, y se valida con un if y el método renameTo, donde renombramos el primer archivo con el segundo, si es correcto se muestra un mensaje de que se renombro, sino se muestra un error.

# **Conclusión:**

Con esta actividad pudimos entender como manejar los archivos, tanto en la creación, lectura, escritura, eliminación y renombramiento de este. Estos métodos nos ayudaran en un futuro para hacer que nuestros programas sean mas escalables, ya que, con estas funcionalidades, nuestras aplicaciones adquieren un valor a futuro, ya que podemos almacenar información y manipularla para poder hacer uso de ella.

Son métodos sencillos, pero que requieren de validaciones para que funcionen de manera óptima.

# **Bibliografía:**

* Froufe, A. (2009) Java 2 Manual de usuario y tutorial. (5ta Ed.). México: Alfaomega.
* García, J. (2000) Aprenda Java como si estuviera en primero. España: Tecnun.
* Joyanes, L. (2002) Java 2 Manual de programación. México: MC Graw Hill
* Martín, A. (2010) Programador certificado Java 2 curso práctico 3ra edición. México: Alfaomega.