**UCSM Escuela Profesional de**

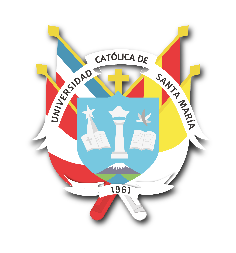
**Ingeniería de Sistemas**

**Octubre - 2024**

Informe de Prácticas

**Lenguaje de Programación III**

**COMPUTACIÓN EN RED I**



**Práctica N°09: Manejo de Archivos en Java**

Elaborado por:

Cusirramos Chiri, Santiago Jesús



**GRUPO N° 04**

**PRÁCTICAS DE LENGUAJE DE PROGRAMACION**

Presentado por:

|  |  |  |
| --- | --- | --- |
| 60783993 | CUSIRRAMOS CHIRI, SANTIAGO JESUS | 100% |
|  |  |  |

**RECONOCIMIENTOS**

El autor de este trabajo reconoce con gratitud a los creadores de los lenguajes JAVA y otras personalidades y autores de libros de programación Bjarne Stroustrup, Dennis Ritchie, Herb Sutter, Herb Sutter, James Gosling, James Gosling, Brian Kernighan, Brian Kernighan, Ken Thompson.

**PALABRAS CLAVES**

F

**ÍNDICE**

[1. RESÚMEN 1](#_Toc180650882)

[2. INTRODUCCIÓN 1](#_Toc180650883)

[3. MARCO TEÓRICO 1](#_Toc180650884)

[4. ACTIVIDADES 12](#_Toc180650885)

[4.1 Actividad 1 12](#_Toc180650886)

[4.2 Actividad 2 14](#_Toc180650887)

[4.3 Actividad 3 15](#_Toc180650888)

[4.4 Actividad 4 16](#_Toc180650889)

[4.5 Actividad 5 17](#_Toc180650890)

[4.6 Actividad 6 19](#_Toc180650891)

[4.7 Actividad 7 21](#_Toc180650892)

[5. EJERCICIOS 23](#_Toc180650893)

[5.1 Ejercicio 1 y 2 23](#_Toc180650894)

[5.2 Ejercicio 3 27](#_Toc180650895)

[5.3 Ejercicio 4 31](#_Toc180650896)

[6. CONCLUSIONES DE LA PRÁCTICA: 33](#_Toc180650897)

[7. CUESTIONARIO 33](#_Toc180650898)

[8. BIBLIOGRAFÍA 35](#_Toc180650899)

# RESÚMEN

E.

# INTRODUCCIÓN

E.

# MARCO TEÓRICO

L

# ACTIVIDADES

## Actividad 1

Según el código visto en el marco teórico implementen tres ventanas que utilicen diferentes tipos de administradores de esquemas FlowLayout, BorderLayout y GridLayout.

Para la implementación de cada uno de los administradores, creen ventanas con títulos diferentes, agréguenles diferentes funcionalidades según los temas vistos en el marco teórico.

Hagan capturas de pantalla por cada implementación de administrador de esquema de tal forma que se evidencie cómo se organiza el contenido y las interacciones que se puede realizar en cada ventana.

**FlowLayOut.**

|  |
| --- |
| package Guia9.Actividad**;** import javax.swing.JFrame**;** import javax.swing.JLabel**;** import javax.swing.JPanel**;** import javax.swing.JTextField**;** import javax.swing.JPasswordField**;** import javax.swing.JButton**;** import java.awt.Color**;** import java.awt.event.ActionEvent**;** import java.awt.event.ActionListener**;** import java.awt.FlowLayout**;** public class LoginFlowLayout {  public static void main(String[] args) {  LoginFrame loginP = new LoginFrame()**;** loginP.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*)**;** loginP.setSize(**400, 200**)**;** loginP.setVisible(true)**;** } }  class LoginFrame extends JFrame {  private JPanel contentPane = new JPanel()**;** private JLabel eti\_user = new JLabel("Ingresar tu correo: ")**;** private JTextField txtUsuario = new JTextField(**20**)**;** private JLabel eti\_contrasenia = new JLabel("Ingresar tu contrasenia: ")**;** private JPasswordField txtPassword = new JPasswordField(**20**)**;** private JButton btnLogin = new JButton("Login")**;** JLabel salida = new JLabel("")**;** public LoginFrame() {  super("Login")**;** contentPane.setLayout(new FlowLayout())**;** contentPane.setBackground(Color.*LIGHT\_GRAY*)**;** contentPane.add(eti\_user)**;** contentPane.add(txtUsuario)**;** contentPane.add(eti\_contrasenia)**;** contentPane.add(txtPassword)**;** contentPane.add(btnLogin)**;** contentPane.add(salida)**;** btnLogin.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String user = txtUsuario.getText()**;** String contrasenia = txtPassword.getText()**;** if(user.trim().isEmpty() || contrasenia.trim().isEmpty()) {  salida.setText("Usuario o Contrasenia ingresado incorrectamente")**;** }  if(!user.trim().isEmpty() || !contrasenia.trim().isEmpty()) {  salida.setText("Ingreso")**;** }  }  })**;** setContentPane(contentPane)**;** } } |

**BorderLayOut y GridLayout.**

|  |
| --- |
| package Guia9.Actividad**;** import javax.swing.JFrame**;** import javax.swing.JLabel**;** import javax.swing.JPanel**;** import javax.swing.JTextField**;** import javax.swing.JPasswordField**;** import javax.swing.JButton**;** import java.awt.GridLayout**;** import javax.swing.SwingConstants**;** import java.awt.BorderLayout**;** import java.awt.event.ActionEvent**;** import java.awt.event.ActionListener**;** public class LoginFlowLayout {  public static void main(String[] args) {  LoginFrame loginP = new LoginFrame()**;** loginP.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*)**;** loginP.setSize(**500, 200**)**;** loginP.setVisible(true)**;** } }  class LoginFrame extends JFrame {  private JPanel contentPane = new JPanel()**;** private JPanel inputPane = new JPanel(new GridLayout(**2, 2, 5, 5**))**;** private JLabel eti\_user = new JLabel("Ingresar tu correo: ")**;** private JTextField txtUsuario = new JTextField(**20**)**;** private JLabel eti\_contrasenia = new JLabel("Ingresar tu contrasenia: ")**;** private JPasswordField txtPassword = new JPasswordField(**20**)**;** private JButton btnLogin = new JButton("Login")**;** JLabel salida = new JLabel(""**,** SwingConstants.*CENTER*)**;** public LoginFrame() {  super("Login")**;** inputPane.add(eti\_user)**;** inputPane.add(txtUsuario)**;** inputPane.add(eti\_contrasenia)**;** inputPane.add(txtPassword)**;** contentPane.add(inputPane**,** BorderLayout.*CENTER*)**;** contentPane.add(btnLogin**,** BorderLayout.*SOUTH*)**;** contentPane.add(salida**,** BorderLayout.*NORTH*)**;** btnLogin.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String user = txtUsuario.getText()**;** String contrasenia = txtPassword.getText()**;** if(user.trim().isEmpty() || contrasenia.trim().isEmpty()) {  salida.setText("Usuario o Contrasenia ingresado incorrectamente")**;** }  if(!user.trim().isEmpty() || !contrasenia.trim().isEmpty()) {  salida.setText("Ingreso")**;** }  }  })**;** setContentPane(contentPane)**;** } } |

# EJERCICIOS

## Ejercicio 1

Investigue en el IDE de su preferencia cómo se puede utilizar herramientas visuales que permitan crear componentes GUI Swing a través de operaciones “soltar y arrastrar” realizadas con el mouse de tal manera que sólo se tenga que agregar código a las clases y manejadores ya creados de manera automática por el IDE. Explique brevemente lo siguiente:

1. Como es el proceso de creación de una GUI utilizando herramientas de “soltar y arrastrar” y haga capturas de pantalla que evidencien el mismo.
2. Cuáles serían algunas ventajas y desventajas de crear una GUI desde código y de hacerlo con una herramienta de “soltar y arrastrar”

Sa

## Ejercicio 2

Implemente una pequeña aplicación que simule el ingreso de datos para la compra de pasajes de una empresa de transporte terrestre. Para esto deberá utilizar todos los conceptos y componentes ya vistos. Una vez que se ingresen todos los datos del pasajero, se deberá mostrar un resumen de sus datos en un cuadro de Dialogo luego de presionar un botón. A continuación, se muestran cómo se utilizarían los componentes para ingresar un tipo de información específica:

Tabla

Descripción generada automáticamente

# CONCLUSIONES DE LA PRÁCTICA:

1. El .

# CUESTIONARIO

* **¿?**

. (2024, UCSM)

# BIBLIOGRAFÍA

D. F. Silva, R. M. (2021). "A Study on the Performance of Java Virtual Machine Garbage Collectors. *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 1*(35), 23-31.

Hortsmann, C. (2009). *Big Java.* San José: Pearson.

Hortsmann, C. S. (2019). *Big Java: Early Objects.* Hoboken: John Wiley & Sons.

Oracle. (1994). *The Java™ Tutorials*. (Oracle) Recuperado el 28 de 07 de 2024, de https://docs.oracle.com/javase/tutorial/index.html

Guia Practica 9 (Ing. Karim, Ing. Montesinos, Ing Iquira, Ing Eveling) - UCSM

# Git

Enlace 🡪