# Davante

TAREA MÓDULO 2

Fernando Tejado Muñoz

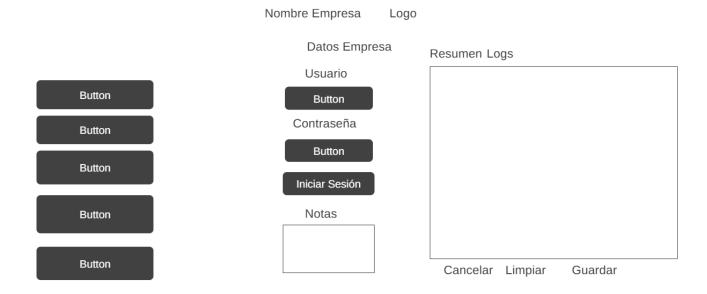


## Título del índice

Wireframe	2
Captura de pantalla de cada interfaz	3
Gestor Usuario	3
Diálogo	5
Analisis del Gestor Usuario	7
Analisis del Dialogo	10
Componentes Utilizados y Propiedades	10



# Wireframe



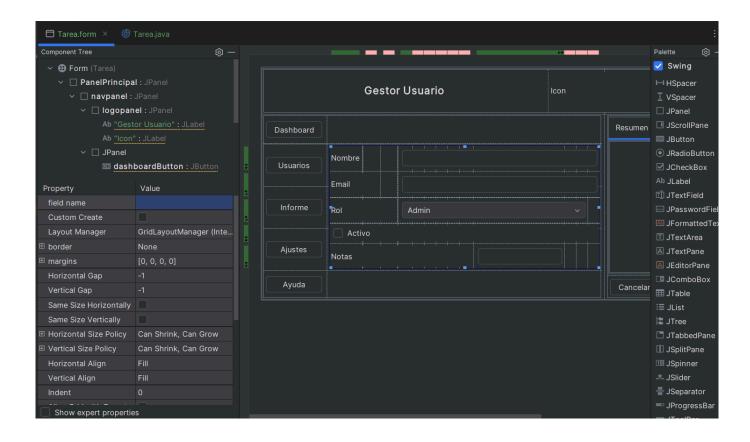


# Captura de pantalla de cada interfaz

#### **Gestor Usuario**

```
☐ Tarea.form
                3 > -
         public class Tarea { new*
             private JPanel PanelPrincipal; 2 usages
4
             private JPanel navpanel; 1usage
  private JPanel logopanel; 1usage
             private JButton dashboardButton; 1usage
  ОК
             private JButton usuariosButton; 1usage
  ОК
             private JButton informeButton; 1usage
  OK
             private JButton ajustesButton; 1usage
11 OK
             private JButton ayudaButton; 1usage
             private JTextField textField1; 1usage
             private JTextField textField2; 1usage
             private JComboBox comboBox1; 1usage
  ⋈
             private JCheckBox activoCheckBox; 1usage
             private JTextField textField3; 1usage
  OK
             private JButton cancelarButton; 1usage
  OK
             private JButton guardarButton; 1usage
  ОК
             private JButton limpiarButton; 1usage
20
             private JTabbedPane tabbedPane1; 1usage
  private JTabbedPane tabbedPane2; 1usage
             public static void main(String[] args) { new*
                 JFrame frame = new JFrame( title: "Tarea");
                 frame.setContentPane(new Tarea().PanelPrincipal);
                 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                 frame.pack();
                 frame.setVisible(true);
```



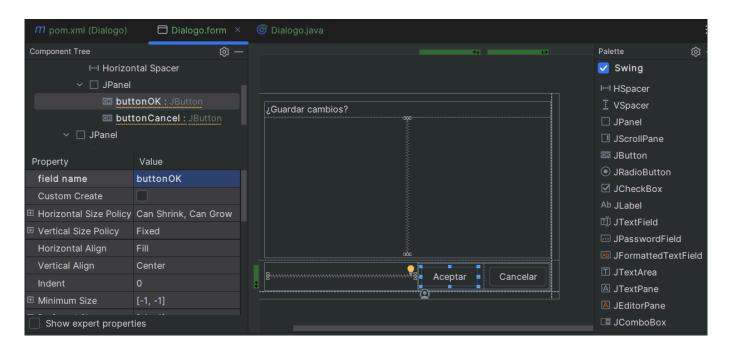


📤 Tarea					_		×
	Gestor Usuario		Icon				
Dashboard			Resumen	Logs			
Usuarios	Nombre						
Usuarios	Email						
	Ciliali						
Informe	Rol	Admin					
monne	1101						
	Activo						
Ajustes							
-	Notas						
				1			
Ayuda			Cancelar	Gı	uardar	Lim	piar



# Diálogo









#### Analisis del Gestor Usuario

El código de la clase Tarea parece ser una interfaz gráfica de usuario (GUI) diseñada con Swing para gestionar usuarios, con un panel principal que contiene varios componentes para navegación, entrada de datos y pestañas.

Componentes Utilizados y Propiedades

JPanel (PanelPrincipal, navpanel, logopanel)

Propiedades relevantes:

PanelPrincipal: Contenedor principal que se establece como contenido del JFrame (frame.setContentPane(new Tarea().PanelPrincipal)).

navpanel: Panel para la barra de navegación

logopanel: Panel para mostrar un logotipo o contenido relacionado.

Explicación:

JPanel es un contenedor genérico en Swing que se usa para agrupar otros componentes. En este caso, PanelPrincipal es el contenedor raíz de la ventana, mientras que navpanel y logopanel se utilizan para organizar la interfaz, posiblemente para una barra de navegación y un área de logotipo, respectivamente.

JButton (dashboardButton, usuariosButton, informeButton, ajustesButton, ayudaButton, cancelarButton, guardarButton, limpiarButton)

Propiedades relevantes:

Cada botón es una instancia de JButton

Explicación:

JButton representa un botón clickable en la interfaz. Los botones como dashboardButton, usuariosButton, etc., sugieren una barra de navegación para acceder a diferentes secciones (dashboard, gestión de usuarios, informes, ajustes, ayuda). Los botones cancelarButton, guardarButton y limpiarButton

JTextField (textField1, textField2, textField3)

Propiedades relevantes:







Campos de texto para entrada de usuario, sin configuraciones específicas visibles en el código.

Explicación:

JTextField permite al usuario ingresar texto. En el contexto de un gestor de usuarios, estos campos podrían usarse para ingresar datos como nombre, correo, ID, etc. La falta de nombres específicos (textField1, textField2, textField3) sugiere que son placeholders generados por un diseñador gráfico (como NetBeans GUI Builder).

JComboBox (comboBox1)

Propiedades relevantes:

Lista desplegable, sin configuraciones específicas visibles.

Explicación:

JComboBox muestra una lista desplegable de opciones. En un gestor de usuarios, podría usarse para seleccionar roles, departamentos o estados (por ejemplo, "Administrador", "Usuario", etc.).

JCheckBox (activoCheckBox)

Propiedades relevantes:

Casilla de verificación, probablemente para indicar un estado booleano (activo/inactivo).

Explicación:

JCheckBox permite al usuario seleccionar o deseleccionar una opción. En este contexto, activoCheckBox probablemente se usa para marcar si un usuario está activo o no.

JTabbedPane (tabbedPane1, tabbedPane2)

Propiedades relevantes:

Contenedores de pestañas, sin configuraciones específicas visibles.

Explicación:

JTabbedPane organiza contenido en pestañas, permitiendo al usuario alternar entre diferentes vistas. En un gestor de usuarios, podrían usarse para separar secciones como "Lista de usuarios", "Agregar usuario", o "Editar usuario".

**JFrame** 

8





#### Propiedades relevantes:

setContentPane(PanelPrincipal): Establece PanelPrincipal como el contenedor principal.

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE): Cierra la aplicación cuando se cierra la ventana.

pack(): Ajusta el tamaño de la ventana al contenido.

setVisible(true): Hace visible la ventana.

#### Explicación:

JFrame es la ventana principal de la aplicación. Aquí se configura para mostrar el PanelPrincipal y cerrar la aplicación al cerrar la ventana.



# Analisis del Dialogo

El código de la clase Dialogo representa un cuadro de diálogo modal en **Swing**, con botones para confirmar (OK) o cancelar (Cancel).

#### Componentes Utilizados y Propiedades

- 1. JDialog (contentPane)
  - Propiedades relevantes:
    - setContentPane(contentPane): Establece el panel principal del diálogo.
    - setModal(true): Configura el diálogo como modal, bloqueando la interacción con otras ventanas hasta que se cierre.
    - getRootPane().setDefaultButton(buttonOK): Establece buttonOK como el botón predeterminado (activado al presionar Enter).
    - setDefaultCloseOperation(DO\_NOTHING\_ON\_CLOSE): Evita el cierre automático al hacer clic en la cruz, delegando el manejo al método onCancel().
  - Explicación:
    - JDialog es una ventana secundaria utilizada para diálogos modales o no modales. En este caso, es un diálogo modal que bloquea la ventana principal hasta que el usuario interactúa con él (aceptar o cancelar).
- 2. JPanel (contentPane)
  - Propiedades relevantes:
    - registerKeyboardAction(...): Asocia la tecla ESC con la acción onCancel().
  - Explicación:
    - JPanel es el contenedor principal del diálogo, que contiene los botones buttonOK y buttonCancel. La configuración de la tecla ESC permite cerrar el diálogo al presionarla.
- 3. JButton (buttonOK, buttonCancel)
  - Propiedades relevantes:
    - buttonOK.addActionListener(...): Asocia un ActionListener que ejecuta onOK() al hacer clic.
    - buttonCancel.addActionListener(...): Asocia un ActionListener que ejecuta onCancel() al hacer clic.





getRootPane().setDefaultButton(buttonOK): Configura buttonOK como el botón predeterminado.

#### Explicación:

■ Los botones buttonOK y buttonCancel permiten al usuario confirmar o cancelar la acción del diálogo. Al hacer clic, ejecutan onOK() o onCancel(), respectivamente, que cierran el diálogo (dispose()).

#### 4. WindowAdapter

#### Propiedades relevantes:

■ addWindowListener(new WindowAdapter() {...}): Maneja el evento de cierre de la ventana (cruz) ejecutando onCancel().

#### Explicación:

WindowAdapter es una clase adaptadora que simplifica la implementación de WindowListener. Aquí se usa para capturar el evento de cierre de la ventana y ejecutar onCancel().

#### 5. KeyStroke y ActionListener (para ESC)

#### Propiedades relevantes:

- contentPane.registerKeyboardAction(...): Asocia la tecla ESC (KeyStroke.getKeyStroke(KeyEvent.VK\_ESCAPE, 0)) con la acción onCancel().
- Condición: JComponent.WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT.

#### Explicación:

■ Esta configuración permite que el diálogo se cierre al presionar ESC, ejecutando onCancel(). La condición asegura que la acción se active cuando el foco esté en cualquier componente dentro de contentPane.