



Tarea de Evaluación 3t

Ramón y Reyes



Boceto inicial app para Maricarmen

Aetas historia

Acceso

Administrador

Investigador

Visitante

Aetas historia

Administrador

Regístrate

Accede

Aetas historia

Regístrate

Nombre

Apellidos

Correo electrónico

Aetas historia

**Añadir
documento**

**Actualizar
documento**

**Eliminar
documento**

**Buscar
Documento**

Aetas Historia

Búsqueda



Inicio

Documentos

Autores

Año

Colección

Nombre del documento

Nombre del Autor

Descripción

Año

Coleccion

Formato

Aetas Historia

Investigador

Regístrate

**Accede con tu
correo de usuario**

Aetas Historia

Regístrate

Nombre

Apellidos

Correo electrónico

Aetas Historia

Visitante

Regístrate

**Accede con tu
correo electrónico**

Aetas Historia

Accede con tu correo
electrónico

Correo

Contraseña

Aetas Historia

Búsqueda



Inicio

Documentos

Autores

Año

Colección

Nombre del documento

Nombre del Autor

Descripción

Año

Coleccion

Formato

Aetas Historia

Búsqueda



Inicio

Documentos

Autores

Epocas históricas

Categoría



Documento 1



Documento 2



Documento 3



Documento 4



Documento 5



Documento 6



Documento 7



Documento 8



Documento 9



Documento 10

Aetas Historia

Búsqueda



2023-1815



Inicio

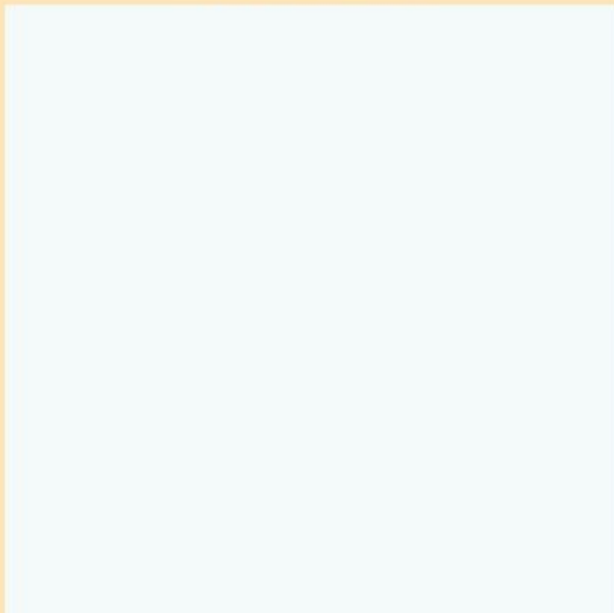
Documentos

Autores

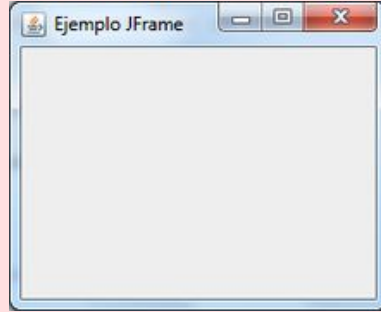
Epocas históricas

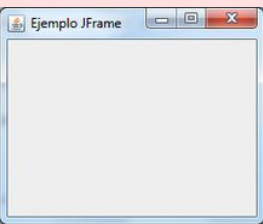
Categoría

Documento 7

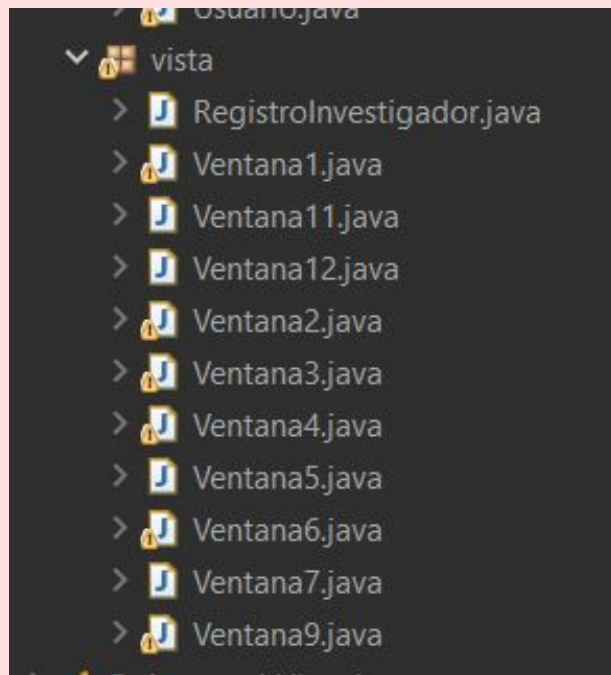


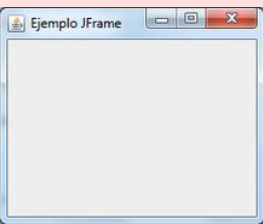
CAMBIO NOMBRES DE VENTANA



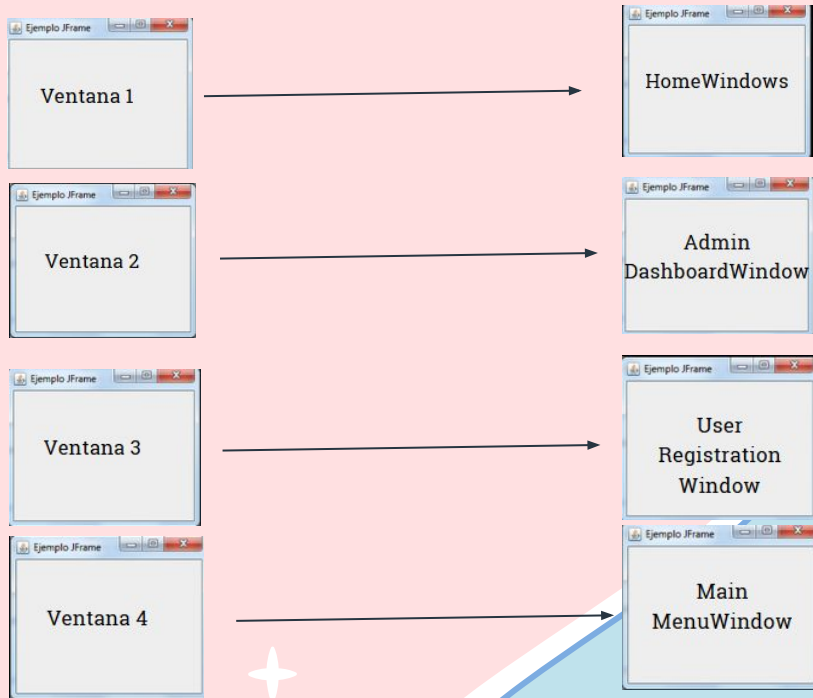


CAMBIO NOMBRES DE VENTANA

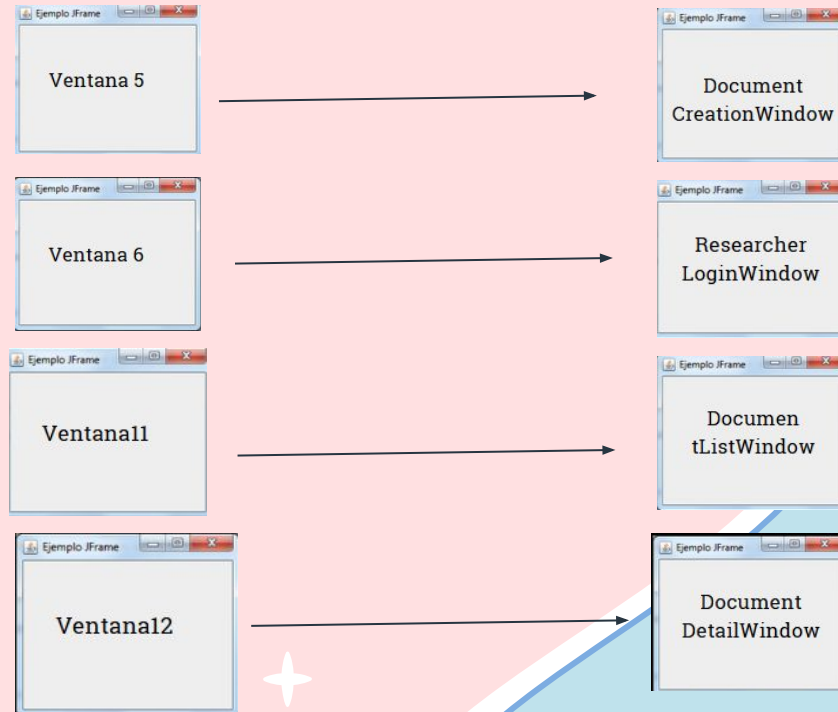


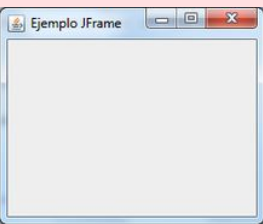


CAMBIO NOMBRES DE VENTANA




CAMBIO NOMBRES DE VENTANA






CAMBIO NOMBRES DE VENTANA

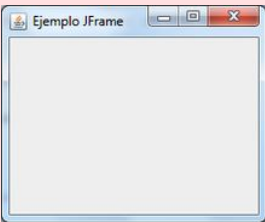


```
Uso.java
vista
├── RegistroInvestigador.java
├── Ventana1.java
├── Ventana11.java
├── Ventana12.java
├── Ventana2.java
├── Ventana3.java
├── Ventana4.java
├── Ventana5.java
├── Ventana6.java
├── Ventana7.java
└── Ventana9.java
```



```
vista
├── AdminDashboardWindow.java
├── DocumentCreationWindow.java
├── DocumentDetailWindow.java
├── DocumentListWindow.java
├── HomeWindow.java
├── InvestigatorRegistrationWindow.java
├── LoginWindow.java
├── MainMenuWindow.java
├── RegistroInvestigador.java
├── ResearcherLoginWindow.java
└── UserRegistrationWindow.java
```





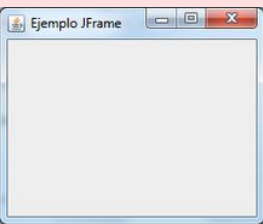
Comentarios

Antes

Sin nada de
comentarios

Ahora

Comentarios bien
estructurados



Comentarios

```
new LoginWindow(conexion).setVisible(true);
dispose();
});
contentPane.add(btnAccede);

// Botón "Retroceso" → regresa a HomeWindow
JButton btnRetroceso = new JButton("Retroceso");
aplicarEstiloBoton(btnRetroceso);
btnRetroceso.setBounds(10, 332, 139, 62);
btnRetroceso.addActionListener(e -> {
    new HomeWindow(conexion).setVisible(true);
    dispose();
});
contentPane.add(btnRetroceso);

// -----
// 4.6) BLOQUE: Imagen representativa
// -----

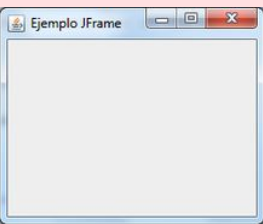
ImageIcon iconoOriginal = new ImageIcon(
    getClass().getResource("/Fotos/administrador.png")
);
Image imgRed = iconoOriginal.getImage()
    .getScaledInstance(150, 160, Image.SCALE_SMOOTH);
JLabel foto1 = new JLabel(new ImageIcon(imgRed));
foto1.setBounds(95, 10, 150, 160);
foto1.setHorizontalAlignment(SwingConstants.CENTER);
contentPane.add(foto1);

// -----
```

```
JLabel lblImg = new JLabel(new ImageIcon(img));
lblImg.setBounds(550, 220, 200, 200);
contentPane.add(lblImg);
}
} else {
    // Bloc mensaje visitante
    JLabel lblVis = new JLabel("Bienvenido, visitante sin logeo");
    lblVis.setForeground(Color.WHITE);
    lblVis.setFont(new Font("Times New Roman", Font.BOLD, 24));
    lblVis.setBounds(200, 200, 400, 50);
    contentPane.add(lblVis);
}

// Bloc Botón Volver
JButton btnVolver = new JButton("Volver");
btnVolver.setForeground(Color.WHITE);
btnVolver.setBackground(Color.GRAY);
btnVolver.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnVolver.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnVolver.setBounds(600, 420, 158, 38);
btnVolver.addActionListener(e -> {
    new DocumentListWindow(conexion).setVisible(true);
    dispose();
});
contentPane.add(btnVolver);

// Bloc Fondo atrás
contentPane.setComponentZOrder(fotoFondo, contentPane.getComponentCount() - 1);
}
```



Comentarios

```
JLabel lblImg = new JLabel(new ImageIcon(img));
lblImg.setBounds(550, 220, 200, 200);
contentPane.add(lblImg);
}
} else {
    // Bloc mensaje visitante
    JLabel lblVis = new JLabel("Bienvenido, visitante sin logeo");
    lblVis.setForeground(Color.WHITE);
    lblVis.setFont(new Font("Times New Roman", Font.BOLD, 24));
    lblVis.setBounds(200, 200, 400, 50);
    contentPane.add(lblVis);
}

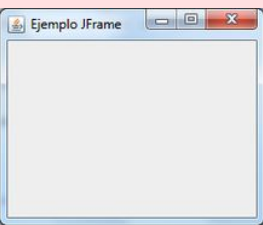
// Bloc Botón Volver
JButton btnVolver = new JButton("Volver");
btnVolver.setForeground(Color.WHITE);
btnVolver.setBackground(Color.GRAY);
btnVolver.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnVolver.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnVolver.setBounds(600, 420, 158, 38);
btnVolver.addActionListener(e -> {
    new DocumentListWindow(conexion).setVisible(true);
    dispose();
});
contentPane.add(btnVolver);

// Bloc Fondo atrás
contentPane.setComponentZOrder(fotoFondo, contentPane.getComponentCount() - 1);
}
```

```
});
if (resp == JOptionPane.YES_OPTION) {
    if (conexion.deleteDocument(sel)) {
        JOptionPane.showMessageDialog(this, "Documento eliminado.");
        model.removeElement(sel);
        titulos.remove(sel);
    } else {
        JOptionPane.showMessageDialog(this, "Error al eliminar.");
    }
}
});

// 11) Botón Volver al menú principal
JButton btnVolver = new JButton("Volver");
btnVolver.setForeground(Color.WHITE);
btnVolver.setBackground(Color.GRAY);
btnVolver.setFont(new Font("Times New Roman", Font.BOLD, 16));
btnVolver.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnVolver.setBounds(10, 480, 120, 40);
btnVolver.addActionListener(e -> {
    new MainMenuWindow(conexion).setVisible(true);
    dispose();
});
contentPane.add(btnVolver);

// 12) Fondo al fondo del z-order
contentPane.setComponentZOrder(fotoFondo, contentPane.getComponentCount() - 1);
}
```



Comentarios

```
1 // src/vista/HomeWindow.java
2
3 package vista;
4
5 import java.awt.*;
6 import javax.swing.*;
7 import javax.swing.border.EmptyBorder;
8 import modelo.ConexionMySQL;
9
10 /**
11  * Ventana principal de selección de rol:
12  * - Administrador
13  * - Investigador
14  * - Visitante (sin logeo)
15  */
16 public class HomeWindow extends JFrame {
17     private static final long serialVersionUID = 1L;
18
19     /** Conexión centralizada a la base de datos */
20     private final ConexionMySQL conexion;
21     /** Panel principal donde pondremos todos los componentes */
22     private JPanel contentPane;
23
24     /**
25      * Constructor.
26      * @param conexion Instancia de conexión a la base de datos
27      */
28     public HomeWindow(ConexionMySQL conexion) {
29         this.conexion = conexion;
30
31         // Configuración básica de la ventana
```

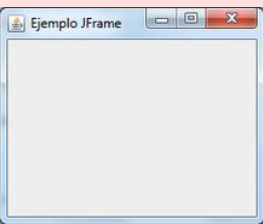
```
/**
 * Ventana para registrar un nuevo investigador.
 */
public class InvestigatorRegistrationWindow extends JFrame {

    private static final long serialVersionUID = 1L;

    /** Panel principal donde van todos los componentes */
    private JPanel contentPane;
    /** Campo de texto para el nombre */
    private JTextField txtNombre;
    /** Campo de texto para los apellidos */
    private JTextField txtApellidos;
    /** Campo de texto para el correo electrónico */
    private JTextField txtCorreo;

    /**
     * Constructor.
     * Monta la interfaz para que un investigador introduzca sus datos.
     */
    public InvestigatorRegistrationWindow() {
        // 4.1 Configuración básica de la ventana
        setTitle("Aetas Historia - Registro Investigador");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 785, 440);

        // 4.2 Creación y configuración del panel principal
        contentPane = new JPanel(null);
        contentPane.setBackground(new Color(139, 69, 19));
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

Comentarios

```
/**
 * Ventana de login genérico: comprueba usuario y contraseña contra la BD.
 */
```

```
public class LoginWindow extends JFrame {
    private static final long serialVersionUID = 1L;

    /** Conexión a la base de datos MySQL */
    private final ConexionMySQL conexion;
    /** Panel principal donde se colocan todos los componentes */
    private JPanel contentPane;
    /** Campo de texto para el nombre de usuario */
    private JTextField txtUsuario;
    /** Campo de texto para la contraseña */
    private JPasswordField txtPassword;

    /**
     * Punto de entrada para pruebas independientes.
     */
    public static void main(String[] args) {
        ConexionMySQL conexion = new ConexionMySQL(
            "localhost", "3306", "root", "", "aetas_historia"
        );
        EventQueue.invokeLater() -> {
            LoginWindow frame = new LoginWindow(conexion);
            frame.setVisible(true);
        });
    }
}
```

```
/**
 * Constructor.
 * @param conexion Instancia de conexión a la base de datos
```

```
// src/vista/MainMenuWindow.java

package vista;

import java.awt.*;

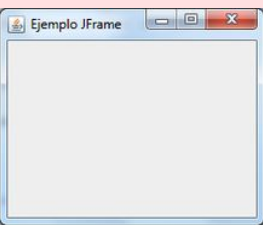
/**
 * Ventana principal tras login: permite gestionar documentos.
 */
public class MainMenuWindow extends JFrame {
    private static final long serialVersionUID = 1L;

    /** Conexión centralizada a la base de datos */
    private final ConexionMySQL conexion;
    private JPanel contentPane;

    /**
     * Constructor.
     * @param conexion Instancia de conexión a la base de datos
     */
    public MainMenuWindow(ConexionMySQL conexion) {
        this.conexion = conexion;

        // 4.1) Configuración básica de la ventana
        setTitle("Aetas Historia - Gestión de Documentos");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 785, 440);

        // 4.2) Panel principal sin layout manager
        contentPane = new JPanel(null);
```

Comentarios

```
istroInve... x HomeWindow.java MainViewWind... Investigato... LoginWindow...
// src/vista/RegistroInvestigador.java

package vista;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import modelo.ConexionMySQL;

/**
 * Ventana para registrar un investigador,
 * llamada desde ResearcherLoginWindow.
 */
public class RegistroInvestigador extends JFrame {

    private static final long serialVersionUID = 1L;
    private final ConexionMySQL conexion;
    private JPanel contentPane;
    private JTextField rellenar_Nombre;
    private JTextField rellenar_Apellidos;
    private JTextField rellenar_CorreoElectronico;

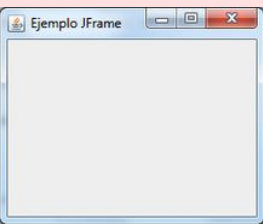
    /**
     * Constructor.
     * @param conexion instancia de conexión a la BD
     */
    public RegistroInvestigador(ConexionMySQL conexion) {
        this.conexion = conexion;

        setTitle("Ventana de Registro de Investigadores");
    }
}
```

```
estiloBoton(btnAccede);
btnAccede.setBounds(226, 257, 282, 51);
btnAccede.addActionListener(e -> {
    new LoginWindow(conexion).setVisible(true);
    dispose();
});
contentPane.add(btnAccede);

// 5) Imagen de fondo
JLabel fotoFondo = new JLabel(new ImageIcon(
    getClass().getResource("/Fotos/biblioteca.png")
));
fotoFondo.setBounds(-112, 0, 902, 558);
contentPane.add(fotoFondo);
contentPane.setComponentZOrder(
    fotoFondo, contentPane.getComponentCount() - 1
);

/** Método auxiliar para dar estilo uniforme a los botones. */
private void estiloBoton(JButton b) {
    b.setForeground(Color.WHITE);
    b.setFont(new Font("Times New Roman", Font.BOLD, 16));
    b.setBackground(Color.GRAY);
    b.setFocusPainted(false);
    b.setBorderPainted(false);
    b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
}
```



Comentarios

```
* @param conexion Instancia de Conexion a la base de datos
*/
public UserRegistrationWindow(ConexionMySQL conexion) {
    this.conexion = conexion;

    // Configuración de la ventana
    setTitle("Aetas Historia - Registro");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 785, 440);

    // Panel principal sin layout manager
    contentPane = new JPanel(null);
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setBackground(new Color(139, 69, 19));
    setContentPane(contentPane);

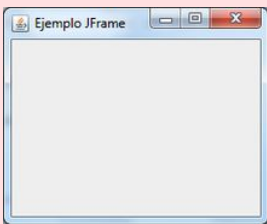
    // Títulos
    JLabel titulo1 = new JLabel("Aetas Historia");
    titulo1.setForeground(Color.WHITE);
    titulo1.setFont(new Font("Times New Roman", Font.BOLD, 36));
    titulo1.setBounds(246, 22, 300, 60);
    contentPane.add(titulo1);

    JLabel titulo2 = new JLabel("Regístrate");
    titulo2.setForeground(Color.WHITE);
    titulo2.setFont(new Font("Times New Roman", Font.BOLD, 36));
    titulo2.setBounds(275, 78, 300, 60);
    contentPane.add(titulo2);

    // Etiquetas de campo
```

```
// Lógica de registro al pulsar
btnRegistrate.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String nombre = rellenar_Nombre.getText().trim();
        String password = rellenar_Apellidos.getText().trim();
        String email = rellenar_CorreoElectronico.getText().trim();

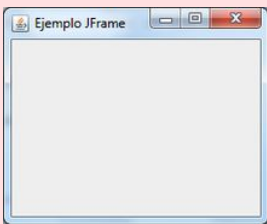
        if (nombre.isEmpty() || password.isEmpty() || email.isEmpty()) {
            JOptionPane.showMessageDialog(
                UserRegistrationWindow.this,
                "Por favor, rellena todos los campos.",
                "Datos inválidos",
                JOptionPane.ERROR_MESSAGE
            );
            return;
        }
        if (!email.contains("@")) {
            JOptionPane.showMessageDialog(
                UserRegistrationWindow.this,
                "El correo debe contener '@'.",
                "Error en correo",
                JOptionPane.ERROR_MESSAGE
            );
            return;
        }
    }
});
```



Codigo reorganizado

Bloques aplicados a TODAS las ventanas

1. **Configuración de la ventana** - ``setTitle`, `setBounds``
2. **Panel principal** - ``contentPane`` sin layout
3. **Fondo visual** - Imagen colocada y enviada al fondo
4. **Títulos** - ``JLabel`` con nombre de la ventana
5. **Etiquetas y campos** - ``JLabel`` + ``JTextField``, etc.
6. **Botones** - Acciones visibles (``JButton``)
7. **Lógica** - ``addActionListener()`` y validaciones
8. **Fondo al fondo** - ``setComponentZOrder(...)``



Codigo reorganizado

Aplicado en cada clase

- **HomeWindow:** Título → botones de rol → fondo
- **AdminDashboardWindow:** Título → botones → fondo
- **ResearcherLoginWindow:** Título → botones → fondo
- **RegistroInvestigador:** Título → etiquetas → campos → botón → fondo
- **LoginWindow:** Usuario/Contraseña → botón → fondo
- **UserRegistrationWindow:** Título → campos → botón → validación → fondo
- **MainMenuWindow:** Título → botones de documentos → fondo
- **DocumentCreationWindow:** Campos → botón crear → fondo
- **DocumentDetailWindow:** Datos del documento → imagen → botón volver → fondo
- **DocumentListWindow:** Título → buscador → lista → botones → fondo

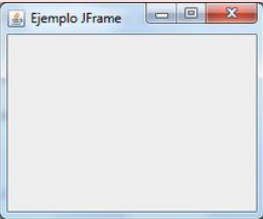
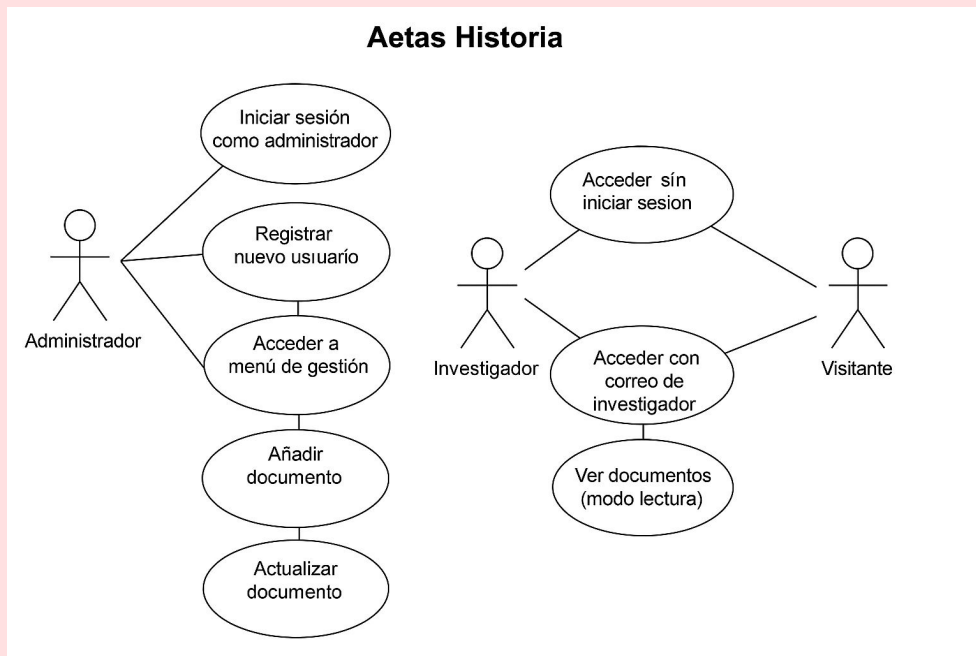
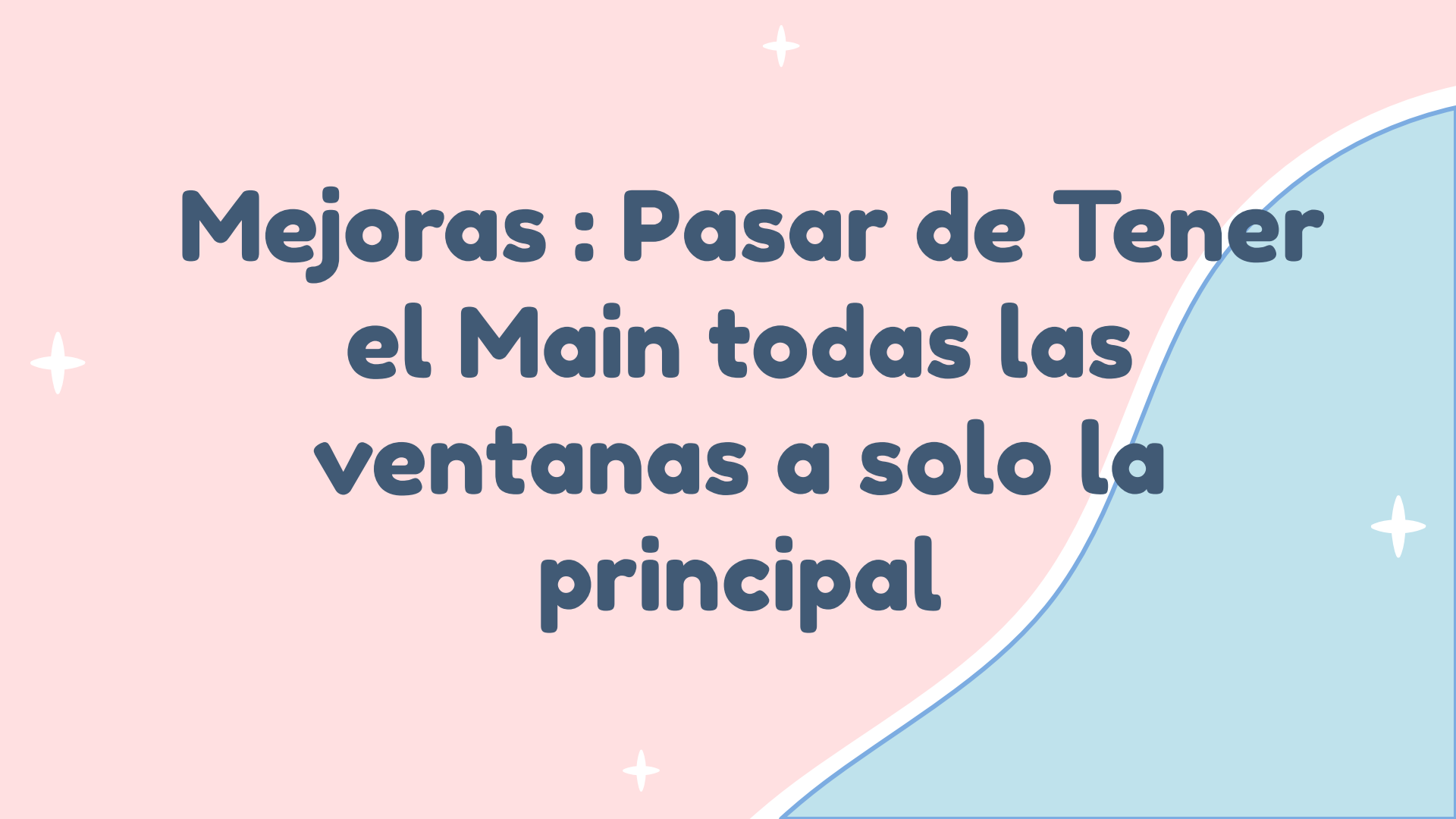


Diagrama entidad relacion





**Mejoras : Pasar de Tener
el Main todas las
ventanas a solo la
principal**

Refactorización

AdminDashboardWindow.java

```
// --- BLOQUE 7: Método auxiliar para aplicar estilo a los botones ---
private void aplicarEstiloBoton(JButton b) {
    b.setForeground(Color.WHITE);
    b.setFont(new Font("Times New Roman", Font.BOLD, 16));
    b.setBackground(Color.GRAY);
    b.setFocusPainted(false);
    b.setBorderPainted(false);
    b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
}
```

DocumentCreationWindow.java
DocumentDetailWindow.java
DocumentListWindow.java

```
new DocumentListWindow(conexion).setVisible(true);
dispose();
});
contentPane.add(btnVolver);

// Fondo al fondo
contentPane.setComponentZOrder(fotoFondo, contentPane.getComponentCount() - 1);
}

// --- REFACTORIZACIÓN: método auxiliar para etiquetas ---
private void addLabel(String texto, int x, int y) {
    JLabel lbl = new JLabel(texto);
    lbl.setForeground(Color.WHITE);
    lbl.setFont(new Font("Times New Roman", Font.BOLD, 18));
    lbl.setBounds(x, y, 140, 30);
    contentPane.add(lbl);
}

// --- REFACTORIZACIÓN: método auxiliar para valores de texto ---
private void addValue(String valor, int x, int y) {
    JLabel val = new JLabel(valor);
    val.setForeground(Color.WHITE);
    val.setFont(new Font("Times New Roman", Font.PLAIN, 16));
    val.setBounds(x, y, 600, 30);
    contentPane.add(val);
}
```


Refactorización

DocumentListWindow.java

LoginWindow.java

```
b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
return b;
}

// --- REFACTORIZACIÓN: Acción eliminar separada ---
private void eliminarDocumento(JList<String> list, ConexionMySQL conexion) {
    String sel = list.getSelectedValue();
    if (sel == null) {
        JOptionPane.showMessageDialog(this, "Selecciona un documento primero.");
        return;
    }

    int resp = JOptionPane.showConfirmDialog(
        this,
        "¿Eliminar '" + sel + "'?",
        "Confirmar eliminación",
        JOptionPane.YES_NO_OPTION
    );

    if (resp == JOptionPane.YES_OPTION) {
        if (conexion.deleteDocument(sel)) {
            JOptionPane.showMessageDialog(this, "Documento eliminado.");
            model.removeElement(sel);
            titulos.remove(sel);
        } else {
            JOptionPane.showMessageDialog(this, "Error al eliminar.");
        }
    }
}
}
```

```
// REFACTORIZACIÓN: Clase auxiliar para componentes comunes (botones)
static class UIComponents {
    public static JButton boton(String texto) {
        JButton b = new JButton(texto);
        b.setForeground(Color.WHITE);
        b.setFont(new Font("Times New Roman", Font.BOLD, 16));
        b.setBackground(Color.GRAY);
        b.setFocusPainted(false);
        b.setBorderPainted(false);
        b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        return b;
    }
}
```


Refactorización Botones

RegistroInvestigador.java


RegistroInvestigador.java

ResearcherLoginWindow.java

```
// REFACTORIZACIÓN: Clase auxiliar para crear botones con estilo común
static class UIComponents {
    public static JButton boton(String texto) {
        JButton b = new JButton(texto);
        b.setForeground(Color.WHITE);
        b.setFont(new Font("Times New Roman", Font.BOLD, 16));
        b.setBackground(Color.GRAY);
        b.setFocusPainted(false);
        b.setBorderPainted(false);
        b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        return b;
    }
}
```

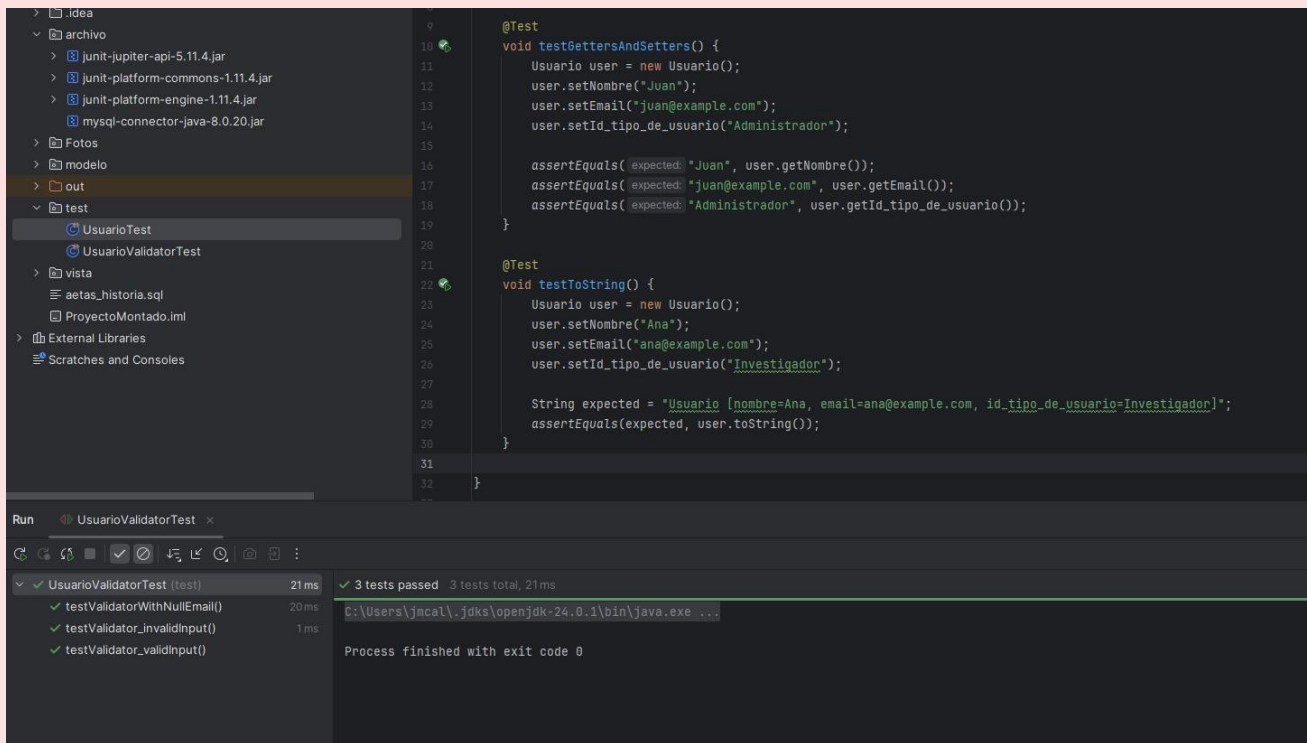
```
// REFACTORIZACIÓN: Clase auxiliar para crear botones con estilo común
static class UIComponents {
    public static JButton boton(String texto) {
        JButton b = new JButton(texto);
        b.setForeground(Color.WHITE);
        b.setFont(new Font("Times New Roman", Font.BOLD, 16));
        b.setBackground(Color.GRAY);
        b.setFocusPainted(false);
        b.setBorderPainted(false);
        b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        return b;
    }
}
```

Refactorización Botones

>  UserRegistrationWindow.java

```
// REFACTORIZACIÓN: Clase auxiliar para crear botones con estilo común
static class UIComponents {
    public static JButton boton(String texto) {
        JButton b = new JButton(texto);
        b.setForeground(Color.WHITE);
        b.setFont(new Font("Times New Roman", Font.BOLD, 16));
        b.setBackground(Color.GRAY);
        b.setFocusPainted(false);
        b.setBorderPainted(false);
        b.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        return b;
    }
}
```

Pruebas Unitarias



The screenshot displays an IDE with a project structure on the left, a code editor in the center, and a run console at the bottom.

Project Structure:

- .idea
- archivo
 - JUnit-jupiter-api-5.11.4.jar
 - junit-platform-commons-1.11.4.jar
 - junit-platform-engine-1.11.4.jar
 - mysql-connector-java-8.0.20.jar
- Fotos
- modelo
- out
- test
 - UsuarioTest
 - UsuarioValidatorTest
- vista
 - aetas_historia.sql
 - ProyectoMontado.iml
- External Libraries
- Scratches and Consoles

Code Editor:

```
9
10
11 @Test
12 void testGettersAndSetters() {
13     Usuario user = new Usuario();
14     user.setNombre("Juan");
15     user.setEmail("juan@example.com");
16     user.setId_tipo_de_usuario("Administrador");
17
18     assertEquals(expected: "Juan", user.getNombre());
19     assertEquals(expected: "juan@example.com", user.getEmail());
20     assertEquals(expected: "Administrador", user.getId_tipo_de_usuario());
21 }
22
23 @Test
24 void testToString() {
25     Usuario user = new Usuario();
26     user.setNombre("Ana");
27     user.setEmail("ana@example.com");
28     user.setId_tipo_de_usuario("Investigador");
29
30     String expected = "Usuario [nombre=Ana, email=ana@example.com, id_tipo_de_usuario=Investigador]";
31     assertEquals(expected, user.toString());
32 }
33 }
```

Run Console:

Run UsuarioValidatorTest x

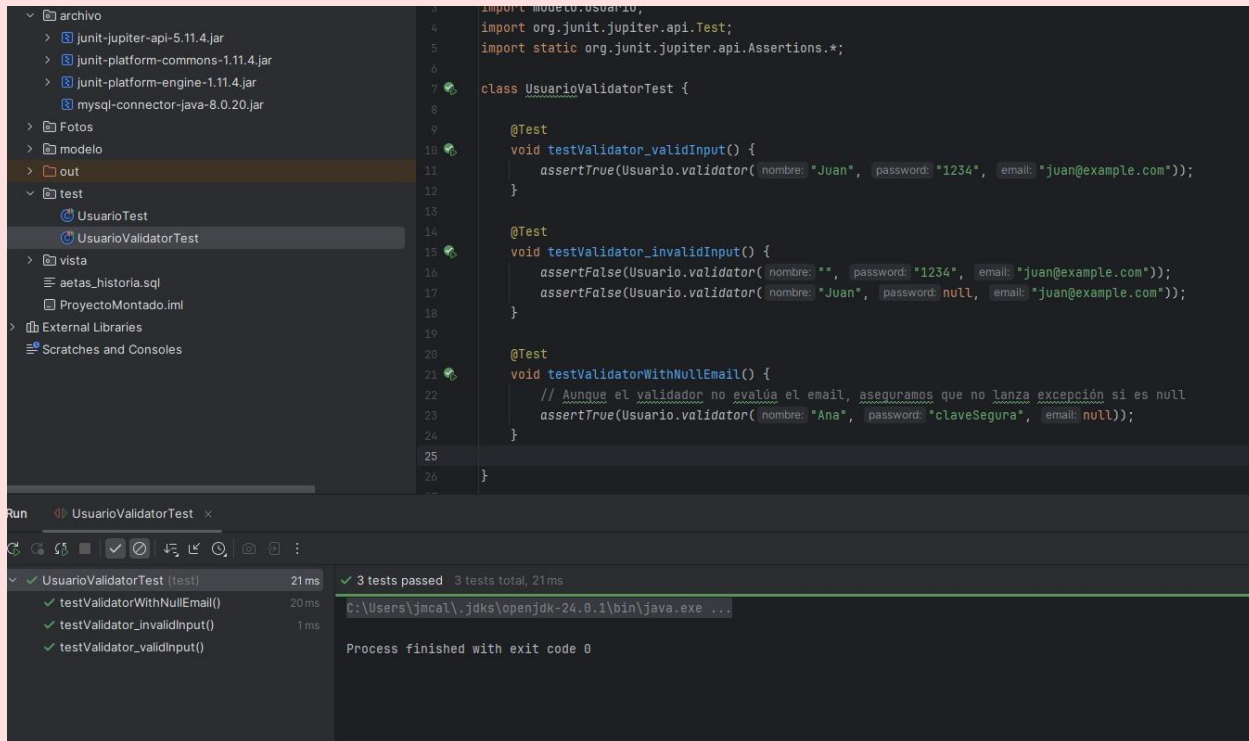
3 tests passed 3 tests total, 21ms

C:\Users\jmc\j\jdk\openjdk-24.0.1\bin\java.exe ...

Process finished with exit code 0

Test Name	Duration
testValidatorWithNullEmail()	21 ms
testValidator_invalidInput()	20 ms
testValidator_validInput()	1 ms

Pruebas Unitarias



The screenshot displays an IDE with a project structure on the left, a code editor in the center, and a run console at the bottom.

Project Structure (Left):

- archivo
 - junit-jupiter-api-5.11.4.jar
 - junit-platform-commons-1.11.4.jar
 - junit-platform-engine-1.11.4.jar
 - mysql-connector-java-8.0.20.jar
- Fotos
- modelo
- out
- test
 - UsuarioTest
 - UsuarioValidatorTest
- vista
- aetas_historia.sql
- ProyectoMontado.iml
- External Libraries
- Scratches and Consoles

Code Editor (Center):

```
import modelo.Usuario;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.*;  
  
class UsuarioValidatorTest {  
  
    @Test  
    void testValidator_validInput() {  
        assertTrue(Usuario.validator( nombre: "Juan", password: "1234", email: "juan@example.com"));  
    }  
  
    @Test  
    void testValidator_invalidInput() {  
        assertFalse(Usuario.validator( nombre: "", password: "1234", email: "juan@example.com"));  
        assertFalse(Usuario.validator( nombre: "Juan", password: null, email: "juan@example.com"));  
    }  
  
    @Test  
    void testValidatorWithNullEmail() {  
        // Aunque el validator no evalúa el email, aseguramos que no lanza excepción si es null  
        assertTrue(Usuario.validator( nombre: "Ana", password: "claveSegura", email: null));  
    }  
}
```

Run Console (Bottom):

Run UsuarioValidatorTest x

✓ 3 tests passed 3 tests total, 21ms

Test Method	Duration
testValidatorWithNullEmail()	20 ms
testValidator_invalidInput()	1 ms
testValidator_validInput()	1 ms

C:\Users\jmc\jdk\openjdk-24.0.1\bin\java.exe ...

Process finished with exit code 0