

MANUAL DE USUARIO

FORMULARIO DE CREACIÓN MANUAL DE USUARIOS



The image shows a web browser window with the title "Formulario de Registro". The main heading inside the form is "FORMULARIO". Below this, there are five input fields, each preceded by a label: "Nombre :", "Apellidos :", "Dirección :", "Teléfono :", and "Notas :". The "Nombre :" field contains a single character, possibly a cursor. At the bottom of the form, there are two buttons: "Limpiar" and "Aceptar".

FORMULARIO	
Nombre :	<input type="text" value=" "/>
Apellidos :	<input type="text"/>
Dirección :	<input type="text"/>
Teléfono :	<input type="text"/>
Notas :	<input type="text"/>
<div><button>Limpiar</button><button>Aceptar</button></div>	

INDICE

1. INTRODUCCIÓN

2. REQUISITOS DEL SISTEMA

3. DESCRIPCIÓN GENERAL DE LA INTERFAZ

5. REGISTRO CORRECTO (BOTÓN ACEPTAR)

6. CAMPO DE “NOTAS (OPCIONAL)”

7. BOTÓN LIMPIAR

8. ESTILO Y DISEÑO NEOMÓRFICO PASTEL

9. ACCESIBILIDAD Y USABILIDAD

10. ERRORES FRECUENTES Y SOLUCIONES

11. CÓDIGO

1. INTRODUCCIÓN

Este manual explica el funcionamiento del formulario “**Creación manual de usuarios**”, desarrollado en **Java Swing** con un estilo visual **neomórfico en tonos rosa pastel y melocotón**.

El objetivo es registrar los datos personales de un usuario

Nombre, Apellidos, Dirección, Teléfono

y opcionalmente añadir **Notas** o comentarios, asegurando una **validación de datos obligatorios** y una **experiencia visual moderna y suave**.

2. REQUISITOS DEL SISTEMA

- **Java JDK 17** o superior.
- **Windows 10/11** (o equivalente).
- Ejecución desde el IDE (IntelliJ, Eclipse, etc.) o mediante:

`java -jar NeumorphicForm.jar`

- Resolución mínima: **1366×768** (para visualizar correctamente los efectos pastel).

3. DESCRIPCIÓN GENERAL DE LA INTERFAZ

La aplicación muestra una ventana titulada “**Formulario de Registro**”, centrada en pantalla, con un diseño **melocotón pastel y efecto 3D suave** (estilo *Soft UI*).

Elementos principales:

- **Campos de texto (obligatorios):**
 - Nombre
 - Apellidos
 - Dirección
 - Teléfono
- **Campo adicional (opcional):**
 - **Notas:** área multilínea para observaciones o comentarios.
- **Botones:**
 - **Aceptar** → valida y registra los datos.
 - **Limpiar** → borra todos los campos y restaura el formulario.
- **Diseño:**
 - Fondo principal: #FFE9E3
 - Relieve suave con sombras y luces cálidas
 - Fuente Segoe UI
 - Texto en color #5A4A4A (rosa oscuro contrastado)

Formulario de Registro

FORMULARIO

Nombre :

Apellidos :

Dirección :

Teléfono :

Notas :

Limpiar

Aceptar

Figura 1. Pantalla principal del formulario neomórfico pastel.

4. Reglas de validación de datos

El sistema valida todos los campos **obligatorios** antes de permitir el registro.

El campo “Notas” es **opcional** y no interviene en la validación.

CAMPO	REQUISITO	ACCIÓN SI FALTA CONTENIDO
Nombre	Obligatorio	Campo en rojo + mensaje de error

Apellidos	Obligatorio	Campo en rojo + mensaje de error
Dirección	Obligatorio	Campo en rojo + mensaje de error
Teléfono	Obligatorio; 7-15 dígitos, puede empezar con “+”	Campo en rojo + mensaje de error
Notas	Opcional	No se valida

Comportamiento del sistema:

- Si falta algún campo obligatorio → aparece un **cuadro de error** (JOptionPane) y los campos vacíos se marcan con **borde rojo**.
- Si todos los campos son correctos → se muestra un **mensaje de confirmación** con los datos introducidos y, si existen, las notas añadidas.
- El formulario se limpia automáticamente tras aceptar.

The screenshot shows a form titled 'FORMULARIO' with fields for 'Nombre', 'Apellidos', 'Dirección', 'Teléfono', and 'Notas'. The 'Nombre' field contains 'Génesis'. The 'Apellidos', 'Dirección', and 'Teléfono' fields are empty and have red borders. A 'Notas' field is also present. At the bottom are 'Limpiar' and 'Aceptar' buttons. An 'Error de validación' dialog box is open, listing three errors: 'El campo Apellidos es obligatorio.', 'El campo Dirección es obligatorio.', and 'El campo Teléfono es obligatorio.' The dialog has an 'OK' button.

Figura 2: Error de validación con campos vacíos.

The screenshot shows the same 'FORMULARIO' interface. The 'Nombre' field contains 'Genesis', 'Apellidos' contains 'Vaca', 'Dirección' contains 'La que sea', and 'Teléfono' contains '6a34'. The 'Notas' field is empty. The 'Limpiar' and 'Aceptar' buttons are at the bottom. An 'Error de validación' dialog box is open, showing one error: 'Teléfono no válido (7-15 dígitos, opcional '+').'. The dialog has an 'OK' button.

Figura 3: Error de formato del teléfono.

5. REGISTRO CORRECTO (BOTÓN ACEPTAR)

Cuando los datos son válidos:

1. El usuario pulsa **Aceptar** (o presiona **Enter**).
2. Aparece una ventana de confirmación con el resumen de todos los campos, incluido "Notas".
3. Si el campo de notas está vacío, se muestra el texto: **"(Sin notas)"**.
4. Tras la confirmación, el formulario se limpia para permitir nuevas entradas.

FORMULARIO

Nombre :

Genesis

Apellidos :

Vaca

Dirección :

La que sea

Teléfono :

+12561452

Notas :

Limpiar

Aceptar

The image shows a web form titled "FORMULARIO" with a pink background. The form contains five input fields: "Nombre :", "Apellidos :", "Dirección :", "Teléfono :", and "Notas :". A modal dialog box titled "Confirmación" is overlaid on the form, displaying the following information: "Formulario registrado correctamente:", "Nombre: Genesis", "Apellidos: Vaca", "Dirección: La que sea", "Teléfono: +12561452", and "Notas: (Sin notas)". The dialog has an "OK" button. At the bottom of the form, there are two buttons: "Limpiar" and "Aceptar".

Figura 4 y 5. Registro exitoso sin notas.

6. CAMPO DE “NOTAS (OPCIONAL)”

Este campo permite añadir información adicional sobre el usuario (por ejemplo, comentarios, observaciones o instrucciones especiales).

Su diseño mantiene el mismo estilo redondeado y pastel que los demás campos, pero con mayor altura para facilitar la escritura.

Características:

- Multilínea, con ajuste automático de texto.
- Fondo pastel con relieve suave.

- No es obligatorio (no genera errores si se deja vacío).
- Se incluye en el cuadro de confirmación final.

FORMULARIO

Nombre :

Genesis

Apellidos :

Vaca

Dirección :

La que sea

Teléfono :

+12345756

Notas :

Hola

Limpiar

Aceptar

The image shows a web form titled "FORMULARIO" with a pink background. The form contains five input fields: "Nombre :", "Apellidos :", "Dirección :", "Teléfono :", and "Notas :". A modal dialog box titled "Confirmación" is overlaid on the form, displaying the following information: "Formulario registrado correctamente:", "Nombre: Genesis", "Apellidos: Vaca", "Dirección: La que sea", "Teléfono: +12345756", and "Notas: Hola". The dialog has an "OK" button. At the bottom of the form, there are two buttons: "Limpiar" and "Aceptar".

Figura 6 y 7. Campo opcional de notas.

7. BOTÓN LIMPIAR

El botón **Limpiar** borra todos los campos, incluyendo el área de notas, y restaura los bordes originales.

También devuelve el **foco** al campo "Nombre".

Mantiene su efecto de presión neomórfico al hacer clic.

FORMULARIO

Nombre :

Apellidos :

Dirección :

Teléfono :

Notas :

Limpiar

Aceptar

Figura 8. Botón Limpiar restablece el formulario.

8. ESTILO Y DISEÑO NEOMÓRFICO PASTEL

Paleta de colores

Colores base (Rosa Melocotón Pastel)

ELEMENTO	COLOR	DESCRIPCIÓN
BASE	#FFE9E3	Fondo principal (melocotón claro)
ACCENT	#F9BEBE	Paneles y botones suaves
TEXT_DARK	#5A4A4A	Texto principal (rosa oscuro)
SHADOW_DARK	#E0B3B3	Sombra inferior del relieve
SHADOW_LIGHT	#FFFFFF	Luz superior cálida

OTRAS OPCIONES DE COLORES EN EL CÓDIGO:

Colores base (Rosa Pastel)

ELEMENTO	COLOR	DESCRIPCIÓN
BASE	#FAE8E8	Fondo principal (rosa empolvado suave)
ACCENT	#F3C6C6	Acento suave para botones o sombras interiores
TEXT_DARK	#5A4A4A	Texto gris rosado oscuro (contraste)
SHADOW_DARK	#E2BABA	Sombra inferior (tono rosado más profundo)
SHADOW_LIGHT	#FFFFFF	Luz superior cálida

FORMULARIO

Nombre :

Apellidos :

Dirección :

Teléfono :

Notas :

Limpiar

Aceptar

Figura 9. Efecto neomórfico Rosa Pastel.

Colores base (Rosa Pastel)

ELEMENTO	COLOR	DESCRIPCIÓN
BASE	#EAF0F5	Fondo principal (celeste suave)
ACCENT	#BBD4E7	Acento suave para botones o sombras interiores
TEXT_DARK	#364355	Texto gris azulado oscuro (contraste)
SHADOW_DARK	#B4C3D0	Sombra inferior (tono azulado más profundo)
SHADOW_LIGHT	#FFFFFF	Luz superior cálida

FORMULARIO

Nombre :

Apellidos :

Dirección :

Teléfono :

Notas :

Limpiar

Aceptar

Figura 10. Efecto neomórfico Azul Pastel.

9. ACCESIBILIDAD Y USABILIDAD

- **Tecla Enter** ejecuta el botón **Aceptar**.
- **Tooltips** informativos al pasar el ratón sobre cada campo.
- **Cursor interactivo** en los botones (mano al pasar).
- **Mensajes claros y centrados**.
- **Ventana adaptable** con todos los elementos centrados.

10. ERRORES FRECUENTES Y SOLUCIONES

SITUACIÓN	CAUSA	SOLUCIÓN
Campos marcados en rojo	Campos vacíos o incorrectos	Completa todos los datos y vuelve a pulsar Aceptar
Error en teléfono	Número corto o con letras	Introduce un número válido (ej. +34911122334)
No se abre la ventana	Java no instalado	Instalar JDK y ejecutar desde el IDE
Colores apagados	Modo oscuro activo	Cambiar al modo claro del sistema

11. CÓDIGO

```
package edu.thepower.desarrollointerfaces.u1;

/**
 * Crear un formulario con Swing que pida Nombre, Apellidos,
 * Dirección y Teléfono
 *
 * - Todos los campos son obligatorios
 *
 * - Botones: Aceptar y Limpiar
 *
 * - Validación con mensajes de error y marcado visual
 */

import javax.swing.*;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import java.awt.*;
import java.awt.event.ActionEvent;

/** Formulario con estilo Neomórfico pastel */

public class NeumorphicForm extends JFrame {

    /**
     * // Colores base (Azul)
     *
     * private static final Color BASE = new Color(0xEAF0F5);
     * // fondo pastel
     */
}
```



```

    *      private static final Color ACCENT = new
Color(0xBBD4E7);      // acento suave

    *      private static final Color TEXT_DARK = new
Color(0x364355);      // texto

    *      private static final Color SHADOW_DARK = new
Color(0xB4C3D0);      // sombra

    *      private static final Color SHADOW_LIGHT = new
Color(0xFFFFFFFF);    // luz

    */

/**

    *      // Colores base (rosa pastel)

    *      private static final Color BASE = new Color(0xFAE8E8);
// Fondo principal (rosa empolvado claro)

    *      private static final Color ACCENT = new
Color(0xF3C6C6);      // Acento suave para botones o sombras
interiores

    *      private static final Color TEXT_DARK = new
Color(0x5A4A4A);      // Texto gris rosado oscuro (contraste)

    *      private static final Color SHADOW_DARK = new
Color(0xE2BABA);      // Sombra inferior (tono rosado más profundo)

    *      private static final Color SHADOW_LIGHT = new
Color(0xFFFFFFFF);    // Luz superior (blanco suave)

    */

// Colores base (rosa melocotón pastel)

private static final Color BASE = new Color(0xFFE9E3);
private static final Color ACCENT = new Color(0xF9BEBE);
private static final Color TEXT_DARK = new Color(0x5A4A4A);
private static final Color SHADOW_DARK = new Color(0xE0B3B3);

```

```
private static final Color SHADOW_LIGHT = new
Color(0xFFFFFFFF);

// Campos

private RoundedField txtNombre;

private RoundedField txtApellidos;

private RoundedField txtDireccion;

private RoundedField txtTelefono;

private RoundedArea txtNotas; // Nuevo campo de texto
multilínea

// Botones

private NeoButton btnAceptar;

private NeoButton btnLimpiar;

// Borde por defecto para restaurar validación

private final javax.swing.border.Border defaultFieldBorder =
new EmptyBorder(8, 12, 8, 12);

public NeumorphicForm() {

    setTitle("Formulario de Registro");

    setDefaultCloseOperation(EXIT_ON_CLOSE);

    setSize(540, 500);

    setLocationRelativeTo(null);

    // Look & Feel y fuente

    try {
```

```

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName
());

    } catch (Exception ignored) {}

    UIManager.put("Label.font", new Font("Segoe UI",
Font.PLAIN, 14));

    // Panel raíz con fondo pastel

    NeoPanel root = new NeoPanel(BASE, 18);

    root.setLayout(new BorderLayout(0, 16));

    root.setBorder(new EmptyBorder(18, 18, 18, 18));

    setContentPane(root);

    // Título

    JLabel title = new JLabel("FORMULARIO",
SwingConstants.CENTER);

    title.setForeground(TEXT_DARK);

    title.setFont(new Font("Segoe UI Semibold", Font.BOLD,
22));

    root.add(title, BorderLayout.NORTH);

    // Panel del formulario

    NeoPanel form = new NeoPanel(ACCENT, 22);

    form.setLayout(new GridBagLayout());

    form.setBorder(new EmptyBorder(20, 22, 20, 22));

    root.add(form, BorderLayout.CENTER);

    GridBagConstraints c = new GridBagConstraints();

    c.insets = new Insets(8, 8, 8, 8);

```

```
c.anchor = GridBagConstraints.WEST;

c.fill = GridBagConstraints.HORIZONTAL;

c.weightx = 1;

// Etiquetas y campos

int row = 0;

txtNombre = addRow(form, c, row++, "Nombre :");

txtApellidos = addRow(form, c, row++, "Apellidos :");

txtDireccion = addRow(form, c, row++, "Dirección :");

txtTelefono = addRow(form, c, row++, "Teléfono :");

// Campo de texto multilínea para notas

c.gridx = 0;

c.gridy = row;

c.weightx = 0;

JLabel lblNotas = new JLabel("Notas :");

lblNotas.setForeground(TEXT_DARK);

form.add(lblNotas, c);

c.gridx = 1;

c.gridy = row;

c.weightx = 1;

txtNotas = new RoundedArea(5, BASE);

txtNotas.setToolTipText("Campo opcional para observaciones o comentarios.");

txtNotas.setPreferredSize(new Dimension(200, 70));

form.add(txtNotas, c);
```

```
// Botones

NeoPanel actions = new NeoPanel(ACCENT, 18);

actions.setLayout(new FlowLayout(FlowLayout.RIGHT, 12,
12));

btnLimpiar = new NeoButton("Limpiar");
btnAceptar = new NeoButton("Aceptar");
actions.add(btnLimpiar);
actions.add(btnAceptar);
root.add(actions, BorderLayout.SOUTH);

// Enter => Aceptar

getRootPane().setDefaultButton(btnAceptar);

// Acciones

btnAceptar.addActionListener(this::onAceptar);
btnLimpiar.addActionListener(e -> limpiar());

// Colores de texto

for (RoundedField f : new RoundedField[]{txtNombre,
txtApellidos, txtDireccion, txtTelefono}) {

    f.setForeground(TEXT_DARK);

    f.setBorder(defaultFieldBorder);

    f.setBackground(BASE);

}

txtNotas.setForeground(TEXT_DARK);

}
```

```
private RoundedField addRow(JPanel form, GridBagConstraints
c, int row, String label) {

    c.gridx = 0;

    c.gridy = row;

    c.weightx = 0;

    JLabel l = new JLabel(label);

    l.setForeground(TEXT_DARK);

    form.add(l, c);

    c.gridx = 1;

    c.gridy = row;

    c.weightx = 1;

    RoundedField field = new RoundedField(14, BASE);

    form.add(field, c);

    return field;

}

private void onAceptar(ActionEvent e) {

    resetBorders();

    String nombre = txtNombre.getText().trim();

    String apellidos = txtApellidos.getText().trim();

    String direccion = txtDireccion.getText().trim();

    String telefono = txtTelefono.getText().trim();

    String notas = txtNotas.getText().trim();

}
```

```
StringBuilder errs = new StringBuilder();

if (nombre.isEmpty()) {
    errs.append("• El campo Nombre es obligatorio.\n");
    markError(txtNombre);
}

if (apellidos.isEmpty()) {
    errs.append("• El campo Apellidos es obligatorio.\n");
    markError(txtApellidos);
}

if (direccion.isEmpty()) {
    errs.append("• El campo Dirección es obligatorio.\n");
    markError(txtDireccion);
}

if (telefono.isEmpty()) {
    errs.append("• El campo Teléfono es obligatorio.\n");
    markError(txtTelefono);
} else if (!telefono.matches("^\\+?\\d{7,15}$")) {
    errs.append("• Teléfono no válido (7-15 dígitos, opcional '+').\n");
    markError(txtTelefono);
}

if (errs.length() > 0) {
    JOptionPane.showMessageDialog(this, errs.toString(),
    "Error de validación", JOptionPane.ERROR_MESSAGE);
}
```

```
        return;
    }

    JOptionPane.showMessageDialog(

        this,

        "Formulario registrado correctamente:\n" +

            "Nombre: " + nombre + "\n" +

            "Apellidos: " + apellidos + "\n" +

            "Dirección: " + direccion + "\n" +

            "Teléfono: " + telefono + "\n" +

            "Notas: " + (notas.isEmpty() ? "(Sin"
notas)" : notas),

        "Confirmación",

        JOptionPane.INFORMATION_MESSAGE

    );

    limpiar();
}

private void limpiar() {

    txtNombre.setText("");

    txtApellidos.setText("");

    txtDireccion.setText("");

    txtTelefono.setText("");

    txtNotas.setText("");

    resetBorders();

    txtNombre.requestFocus();
}
```



```

    }

    private void markError(JTextField f) {
        f.setBorder(new LineBorder(new Color(0xE57373), 1));
    }

    private void resetBorders() {
        txtNombre.setBorder(defaultFieldBorder);
        txtApellidos.setBorder(defaultFieldBorder);
        txtDireccion.setBorder(defaultFieldBorder);
        txtTelefono.setBorder(defaultFieldBorder);
    }

    // ----- Componentes personalizados
    -----

    static class NeoPanel extends JPanel {
        private final Color base;
        private final int radius;

        NeoPanel(Color base, int radius) {
            this.base = base;
            this.radius = radius;
            setOpaque(false);
        }

        @Override

```

```

        protected void paintComponent(Graphics g) {

            Graphics2D g2 = (Graphics2D) g.create();

            g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

            int w = getWidth(), h = getHeight();

            g2.setColor(new Color(255, 255, 255, 180));

            g2.fillRoundRect(4, 4, w - 8, h - 8, radius,
radius);

            g2.setColor(new Color(180, 190, 200, 120));

            g2.fillRoundRect(6, 6, w - 12, h - 12, radius,
radius);

            g2.setColor(base);

            g2.fillRoundRect(6, 6, w - 12, h - 12, radius,
radius);

            g2.dispose();

            super.paintComponent(g);

        }

    }

    static class NeoButton extends JButton {

        NeoButton(String text) {

            super(text);

            setFocusPainted(false);

            setBorderPainted(false);

            setContentAreaFilled(false);

            setForeground(TEXT_DARK);

            setFont(new Font("Segoe UI", Font.PLAIN, 14));

            setPreferredSize(new Dimension(110, 36));

```

```

setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));

    }

    @Override
    protected void paintComponent(Graphics g) {
        Graphics2D g2 = (Graphics2D) g.create();

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        int w = getWidth(), h = getHeight(), r = 18;

        boolean pressed = getModel().isArmed() &&
getModel().isPressed();

        Color top = pressed ? new Color(0xE0B3B3) :
SHADOW_LIGHT;

        Color bottom = pressed ? SHADOW_LIGHT : SHADOW_DARK;

        g2.setColor(top);

        g2.fillRoundRect(2, 2, w - 4, h - 4, r, r);

        g2.setColor(bottom);

        g2.fillRoundRect(3, 3, w - 6, h - 6, r, r);

        g2.setColor(BASE);

        g2.fillRoundRect(3, 3, w - 6, h - 6, r, r);

        g2.setColor(TEXT_DARK);

        FontMetrics fm = g2.getFontMetrics();

        int tx = (w - fm.stringWidth(getText())) / 2;

        int ty = (h - fm.getHeight()) / 2 + fm.getAscent();

        g2.drawString(getText(), tx, ty);

        g2.dispose();

    }

```

```
}

static class RoundedField extends JTextField {

    private final int arc;

    private final Color base;

    RoundedField(int arc, Color base) {

        this.arc = arc;

        this.base = base;

        setOpaque(false);

        setBorder(new EmptyBorder(8, 12, 8, 12));

    }

    @Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2 = (Graphics2D) g.create();

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        int w = getWidth(), h = getHeight();

        g2.setColor(SHADOW_LIGHT);

        g2.fillRoundRect(1, 1, w - 2, h - 2, arc, arc);

        g2.setColor(SHADOW_DARK);

        g2.fillRoundRect(2, 2, w - 4, h - 4, arc, arc);

        g2.setColor(base);

        g2.fillRoundRect(2, 2, w - 4, h - 4, arc, arc);

        g2.dispose();

        super.paintComponent(g);

    }

}
```

```

    }

}

/** Campo multilínea para "Notas" con mismo estilo redondeado
 */

static class RoundedArea extends JTextArea {

    private final int arc;

    private final Color base;

    RoundedArea(int arc, Color base) {

        this.arc = arc;

        this.base = base;

        setOpaque(false);

        setBorder(new EmptyBorder(10, 12, 10, 12));

        setLineWrap(true);

        setWrapStyleWord(true);

    }

    @Override

    protected void paintComponent(Graphics g) {

        Graphics2D g2 = (Graphics2D) g.create();

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        int w = getWidth(), h = getHeight();

        g2.setColor(SHADOW_LIGHT);

        g2.fillRoundRect(1, 1, w - 2, h - 2, arc, arc);

        g2.setColor(SHADOW_DARK);

```

```
        g2.fillRoundRect(2, 2, w - 4, h - 4, arc, arc);

        g2.setColor(base);

        g2.fillRoundRect(2, 2, w - 4, h - 4, arc, arc);

        g2.dispose();

        super.paintComponent(g);
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new
NeumorphicForm().setVisible(true));
}
}
```