

MDVAL

DATOS GENERALES

Identificación del documento	ARQ-002_PANTALLAS			
Título del documento	Arquitectura de pantallas			
Proyecto	MDVAL			
Gerente				
Área demandante				
Director	Federico Martín Lara			
Responsables	Área	Responsable		
	Dirección	Federico Martín Lara		
Control de versionado	Versión	Fecha	Autor	Comentarios
	1.0	21/09/2021	Federico Martín Lara	Versión inicial

Contenido

Introducción.....	3
Clasificación de pantallas.....	4
Construcción de las pantallas.....	5
Definición de los métodos de inicialización de pantallas.....	7
Ejemplos de declaración de los distintos métodos.....	8

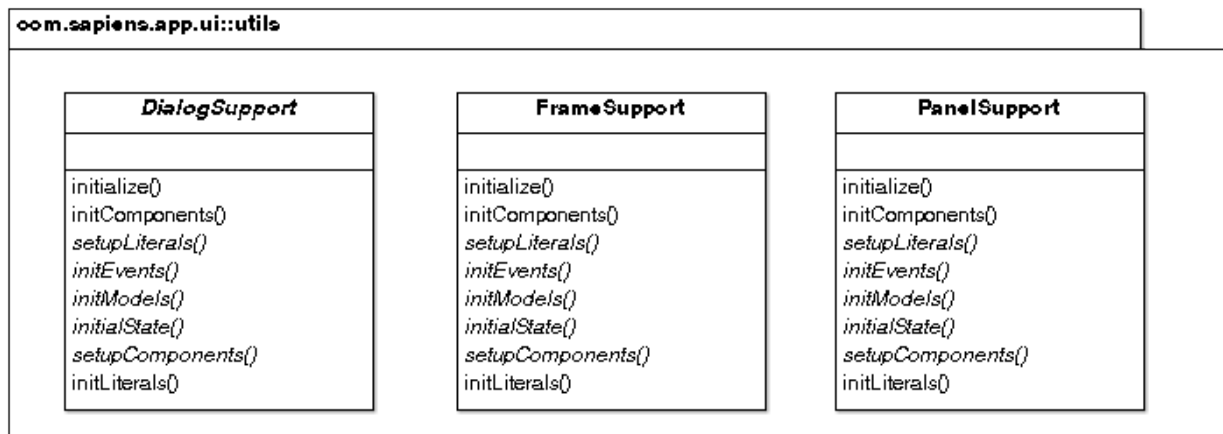
Introducción

El presente documento define la arquitectura genérica que siguen todas las pantallas en el momento de su construcción.

Clasificación de pantallas

En el aplicativo, para mostrar los distintos componentes funcionales e interactuar con ellos, se usan tres clases de pantallas o contenedores. Se detallan a continuación, mencionando también su clase gestora:

- **Frame o pantalla independiente**, muestra las distintas pantallas del aplicativo. Se pueden superponer unos sobre otros, independientemente de su funcionalidad. Su clase gestora es **FrameSupport**, con funcionalidades genéricas que serán útiles en las clases que hereden de ésta.
- **Cuadro de diálogo o popup**, muestra pantallas que requieren acciones como altas, borrados o modificaciones. Se activan mediante botones de las distintas pantallas independientes. Se sitúan por encima de las pantallas independientes y bloquean hasta que se realice la operación, usualmente con **Aceptar** o su rechazo con **Cancelar**. Su clase gestora es **DialogSupport**.
- **Paneles**. Dentro de los frames o cuadros de diálogo se pueden introducir paneles con funcionalidades específicas. Su clase gestora es **PanelSupport**.



Construcción de las pantallas

Todas las pantallas, independientemente de su clasificación, desde que se invocan hasta que se muestran, siguen una secuencia de inicialización bien definida (se implementa mediante el patrón **template-method**), la cual se cumple para todas. Se define la secuencia genérica que cumplen los tres tipos (FrameSupport, DialogSupport y PanelSupport):

1. Carga del logotipo. Si el componente no requiere de logo, por ejemplo PanelSupport, esta parte no existirá.
2. Construcción y colocación de los componentes, método **setupComponents**.
3. Carga de los literales y textos de los componentes, método **setupLiterals**.
4. Declaración de eventos (llamada a botones, etc), método **initEvents**.
5. Carga inicial de modelos (tablas, listas y combos), método **initModels**.
6. Invocación de listeners de carga, lista de listeners **OnloadListener**, método **onLoad** por cada listener.
7. Inicialización de componentes, método **initialState**.

Se muestra a continuación el proceso de inicialización:

```
/**
 * Proceso de inicialización
 */
private void initialize() {
    try {
        panelLogo = new PanelLogotipo("logotipo.png");
        panelLogo.setPreferredSize(new Dimension(286, 63));

        dateFormatter = new DateFormatter();
        onLoadListeners = new ArrayList<>();

        initComponents();
        initLiterals();
        initEvents();

        initModels();

        if (CollectionUtils.isEmpty(onLoadListeners)) {
            for (OnLoadListener l : onLoadListeners) {
                l.onLoad();
            }
        }

        initialState();
    } catch (IOException e) {
        log.warn("ERROR:", e);
    }
}
```

Inicialización para PanelSupport:

```
/**
 * Proceso de inicialización
 */
private void initialize() {
    try {
        initComponents();
        initLiterals();
        initEvents();

        initModels();
        initialState();
    } catch (IOException e) {
        log.warn("ERROR:", e);
    }
}
```

El método **setupComponents** se utiliza en el método initComponents:

```
/**
 * Encapsula la creación de componentes
 */
private void initComponents() {
    setupComponents();

    pack();
}
```

El método **setupLiterals** se utiliza en el método initLiterals:

```
/**
 * Encapsula la creación de los literales
 * @throws IOException
 */
private void initLiterals() throws IOException {
    literals = LiteralesSingleton.getInstance();

    setupLiterals();
}
```

Definición de los métodos de inicialización de pantallas

Los métodos mencionados en el apartado anterior deben ser redefinidos en las clases heredadas de los tres tipos de la clasificación. Se define a continuación el propósito de cada método:

- **initComponents**: Construye y coloca los componentes de la pantalla
- **setupLiterals**: Establece los textos a los componentes contruidos con el método anterior que lo necesiten. Los literales están definidos en **literals.properties**.
- **initEvents**: inicia los manejadores de eventos (listeners) que controlarán la pantalla. Se establecen listeners para botones, combos y tablas. Algunas tablas

disponen de un listener de selección, que maneja la selección de filas, en este caso se permite la selección de una única fila, y dispara las acciones especificadas para el evento.

- **initModels:** Establece los modelos para tablas y combos, de manera que puedan visualizar la información solicitada de forma correcta.
- **onLoad:** ejecuta los eventos de inicio de la ventana, en caso de que se tenga que hacer operaciones de acceso a bases de datos (a través de los servicios). Se pueden declarar varios eventos onLoad (uno por cada instancia de OnLoadListener que se le añada a la ventana, cuadro de diálogo o panel).
- **initialState:** informa los distintos componentes de la ventana que lo requieran, como cuadros de texto. También habilita y deshabilita o muestra y oculta botones según sea requerido en la creación de la pantalla.

Ejemplos de declaración de los distintos métodos

- Declaración de setupComponents

```
protected void setupComponents() {  
  
    txtGlosario = new JTextField();  
    jLabel1 = new JLabel();  
    btnBuscar = new JButton();  
    btnAlta = new JButton();  
    btnModificacion = new JButton();  
    btnSeleccionar = new JButton();  
    jLabel2 = new JLabel();  
}
```

MDVAL ARQ-002_PANTALLAS – Arquitectura de pantallas

```
JScrollPane jScrollPane1 = new JScrollPane(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,  
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);  
tblGlosarios = new TableSupport();  
  
setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);  
  
GroupLayout panelLogoLayout = new GroupLayout(panelLogo);  
panelLogo.setLayout(panelLogoLayout);  
panelLogoLayout.setHorizontalGroup(  
  
    panelLogoLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0, 286,  
Short.MAX_VALUE));  
panelLogoLayout.setVerticalGroup(  
  
    panelLogoLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGap(0, 63,  
Short.MAX_VALUE));  
  
txtGlosario.setMinimumSize(new Dimension(4, 27));  
txtGlosario.setPreferredSize(new Dimension(300, 27));  
  
jLabel1.setFont(new Font("Dialog", 1, 18)); // NOI18N  
  
btnAlta.setPreferredSize(new Dimension(130, 27));  
btnSeleccionar.setPreferredSize(new Dimension(130, 27));  
  
jScrollPane1.setViewportView(tblGlosarios);  
  
GroupLayout layout = new GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
  
    layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(GroupLayout.Alignment.TRAILING,  
        layout.createSequentialGroup()  
            .createSequentialGroup()  
                .addContainerGap()  
                .addGroup(layout.createParallelGroup(GroupLayout.  
                    GroupLayout.Alignment.TRAILING)  
                    .addComponent(jScrollPane1)  
                    .addGroup(GroupLayout.Alignment.LEADING,  
                        layout.createSequentialGroup()  
                            .createSequentialGroup()  
                                .addComponent(panelLogo, GroupLayout.PREFERRED_SIZE,  
                                    GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)  
                                .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)  
                                .addComponent(jLabel1).addGap(0, 0, Short.MAX_VALUE))  
                            .addGroup(layout.createSequentialGroup()  
                                .addGap(0, 0, Short.MAX_VALUE)  
                                .addComponent(btnAlta, GroupLayout.PREFERRED_SIZE,  
                                    GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)  
                                .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)  
                                .addComponent(btnModificacion).addGap(18, 18, 18)  
                                .addComponent(btnSeleccionar, GroupLayout.PREFERRED_SIZE,  
                                    GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))  
                            .addGroup(GroupLayout.Alignment.LEADING,  
                                layout.createSequentialGroup()  
                                    .createSequentialGroup().addGap(168, 168, 168)
```


MDVAL ARQ-002_PANTALLAS – Arquitectura de pantallas

```
.addComponent(jLabel2)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtGlosario, GroupLayout.PREFERRED_SIZE,
               GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 516,
               Short.MAX_VALUE)

.addComponent(btnBuscar)))
        .addContainerGap());
    layout.setVerticalGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup().addContainerGap()
            .addGroup(layout.createParallelGroup(GroupLayout.Alignment
ent.LEADING)
        .addComponent(panelLogo,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel1)
        .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELAT
        .addGroup(layout.createParallelGroup(GroupLayout.Alignm
ent.BASELINE)
        .addComponent(txtGlosario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2).addComponent(btnB
uscar))
        .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELAT
        .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELAT
        .addGroup(layout.createParallelGroup(GroupLayout.Alignm
ent.BASELINE)
        .addComponent(btnSeleccionar,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(btnModificacion).addCompon
        GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addContainerGap());
    }
```

- Declaración de setupLiterals

```
/**
 *
 */
protected void setupLiterals() {
    setTitle(literales.getLiteral("frmDefinicionGlosarios.titulo"));

    jLabel1.setText(literales.getLiteral("frmDefinicionGlosarios.titulo"));
    jLabel2.setText(literales.getLiteral("frmDefinicionGlosarios.glosario"));

    btnBuscar.setText(literales.getLiteral("frmDefinicionGlosarios.buscar"));
    btnAlta.setText(literales.getLiteral("frmDefinicionGlosarios.alta"));
    btnModificacion.setText(literales.getLiteral("frmDefinicionGlosarios.modificacion"));
    btnSeleccionar.setText(literales.getLiteral("frmDefinicionGlosarios.seleccionar"));
```

MDVAL ARQ-002_PANTALLAS – Arquitectura de pantallas

```
}
```

- Declaración de initEvents

```
/**
 *
 */
protected void initEvents() {
    frmDefinicionGlosariosListener = new FrmDefinicionGlosariosListener(this);
    ListSelectionListener listSelectionListener = new
FrmDefinicionGlosariosTableListener(this);

    btnBuscar.setActionCommand(Constants.FRM_DEFINICION_GLOSARIOS_BTN_BUSCAR);
    btnAlta.setActionCommand(Constants.FRM_DEFINICION_GLOSARIOS_BTN_ALTA);

    btnModificacion.setActionCommand(Constants.FRM_DEFINICION_GLOSARIOS_BTN_MODIFICACION);
    btnSeleccionar.setActionCommand(Constants.FRM_DEFINICION_GLOSARIOS_BTN_SELECCIONAR);

    btnBuscar.addActionListener(frmDefinicionGlosariosListener);
    btnAlta.addActionListener(frmDefinicionGlosariosListener);
    btnModificacion.addActionListener(frmDefinicionGlosariosListener);
    btnSeleccionar.addActionListener(frmDefinicionGlosariosListener);

    ListSelectionModel rowSM = tblGlosarios.getSelectionModel();
    rowSM.addListSelectionListener(listSelectionListener);
}
}
```

- Declaración de initModels

```
/**
 *
 */
@Override
protected void initModels() {
    tblGlosarios.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
    tblGlosarios.setDefaultRenderer(Integer.class, new IntegerRenderer());
    tblGlosarios.setDefaultRenderer(Date.class, new DateTimeRenderer());
    tblGlosarios.setDefaultRenderer(BigDecimal.class, new BigDecimalRenderer());
    tblGlosarios.setDefaultRenderer(String.class, new StringRenderer());

    Cabecera cabecera =
UIHelper.createCabeceraTabla(Constants.FRM_DEFINICION_GLOSARIOS_TABLA_GLOSARIOS_CABECERA);
    tblGlosarios.setModel(new
DefinicionGlosariosTableModel(cabecera.getColumnIdentifiers(), cabecera.getColumnClasses()));
}
}
```

- Declaración de initialState

```
/**
 *
 */
protected void initialState() {
    btnModificacion.setEnabled(Boolean.FALSE);
    btnSeleccionar.setEnabled(Boolean.FALSE);
}
```

}