

Swing 11 - Navegación con Menú

En ésta práctica se revisará la navegación con el uso de menú para las pantallas involucradas en la administración de Presentaciones en Congreso.

Se comenzará descargando el proyecto **NavegaMenuBase.zip** el cual contiene como elementos de tipo **JInternalFrameForm** para agregar una nueva Presentación (**NuevaPresentacion**) y un nuevo Congreso (**NuevoCongreso**), que no tienen funcionalidad alguna, así como un panel con las estadísticas generales.

Agregando el Menú

Una vez cargado el proyecto, se agregará la forma principal, un **JFrame** con nombre **FrmContenedorPrincipal**, al cual se le dará un tamaño de 800 x 600

Se agregará un elemento de menú (**Menu Bar**) el cual se renombrará a **menuSitio** y se editarán los textos por defecto para que digan "**Principal**" (**menuPrincipal**) y "**Presentaciones en Congreso**" (**menuPresentaciones**). A la opción de "**Presentaciones en Congreso**" se le agregarán dos tres sub menús dando clic derecho sobre ella, seleccionar **Add From Palette** y la opción de **Menu Item**.

Las tres opciones serán:

- Nueva Presentación (**opcNuevaPresentacion**)
- Mis Presentaciones (**opcMisPresentaciones**)
- Nuevo Congreso (**opcNuevoCongreso**)

Se agregará el elemento que permitirá ir visualizando los diferentes paneles internos, en este caso un elemento del tipo **Desktop Pane (dskPrincipal)** el cuál cubrirá el contenedor principal, se le cambiará el color de fondo a gris.

Agregando Navegación

Para la navegación, se le agregarán eventos a cada opción del menú en la que se quiera desplegar una pantalla diferente, en este caso a las tres del sub menú agregadas anteriormente.

FrmContenedorPrincipal.java

```
private void
opcNuevaPresentacionActionPerformed(java.awt.event.ActionEvent evt) {

}

private void
opcMisPresentacionesActionPerformed(java.awt.event.ActionEvent evt) {

}

private void
opcNuevoCongresoActionPerformed(java.awt.event.ActionEvent evt) {

}
```

Se cargará cada uno de los Frames internos según la opción seleccionada.

FrmContenedorPrincipal.java

```
private void
opcNuevaPresentacionActionPerformed(java.awt.event.ActionEvent evt) {
    NuevaPresentacion nuevaPresentacion = new NuevaPresentacion();
    dskPrincipal.add(nuevaPresentacion);
    nuevaPresentacion.setVisible(true);
}
private void
opcMisPresentacionesActionPerformed(java.awt.event.ActionEvent evt) {
    MisPresentaciones misPresentaciones = new MisPresentaciones();
    dskPrincipal.add(misPresentaciones);
    misPresentaciones.setVisible(true);
}
private void
opcNuevoCongresoActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoCongreso nuevoCongreso = new NuevoCongreso();
    dskPrincipal.add(nuevoCongreso);
    nuevoCongreso.setVisible(true);
}
```

Se desea que al cargar la aplicación, aparezca el panel principal, esto se configurará en el constructor.

FrmContenedorPrincipal.java

```
public FrmContenedorPrincipal() {
    initComponents();
    Principal principal = new Principal();
    dskPrincipal.add(principal);
    principal.setVisible(true);
}
```

También se quiere que al seleccionar la opción de Principal en el menú, se vuelva a presentar la pantalla inicial. Sin embargo, las acciones solo funcionan en elementos de sub menú, por lo que se agregará la opción de “*Inicio*” (*menuInicio*) como una opción del menú *Principal*.

FrmContenedorPrincipal.java

```
private void menuInicioActionPerformed(java.awt.event.ActionEvent evt) {
    Principal principal = new Principal();
    dskPrincipal.add(principal);
    principal.setVisible(true);
}
```

Mejorando la presentación.

Se maximizará tanto el contenedor principal como el resto de los elementos, primero se maximizará el contenedor principal.

FrmContenedorPrincipal.java

```
public void run() {  
    //new FrmContenedorPrincipal().setVisible(true);  
    FrmContenedorPrincipal contenedor = new  
FrmContenedorPrincipal();  
    contenedor.setExtendedState(MAXIMIZED_BOTH);  
    contenedor.setVisible(true);  
  
}
```

FrmContenedorPrincipal.java

```
private void  
opcNuevaPresentacionActionPerformed(java.awt.event.ActionEvent evt) {  
    NuevaPresentacion nuevaPresentacion = new NuevaPresentacion();  
    dskPrincipal.add(nuevaPresentacion);  
    try {  
        nuevaPresentacion.setMaximum(true);  
    } catch (PropertyVetoException ex) {  
  
Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE  
, null, ex);  
    }  
    nuevaPresentacion.setVisible(true);  
}  
  
private void  
opcMisPresentacionesActionPerformed(java.awt.event.ActionEvent evt) {  
    MisPresentaciones misPresentaciones = new MisPresentaciones();  
    dskPrincipal.add(misPresentaciones);  
    try {  
        misPresentaciones.setMaximum(true);  
    } catch (PropertyVetoException ex) {  
  
Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE  
, null, ex);  
    }  
    misPresentaciones.setVisible(true);  
}  
  
private void  
opcNuevoCongresoActionPerformed(java.awt.event.ActionEvent evt) {  
    NuevoCongreso nuevoCongreso = new NuevoCongreso();  
    dskPrincipal.add(nuevoCongreso);  
    try {  
        nuevoCongreso.setMaximum(true);  
    } catch (PropertyVetoException ex) {  
  
Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE  
, null, ex);  
    }  
    nuevoCongreso.setVisible(true);  
}
```

```

private void menuInicioActionPerformed(java.awt.event.ActionEvent
evt) {
    Principal principal = new Principal();
    dskPrincipal.add(principal);
    try {
        principal.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    principal.setVisible(true);
}

```

FrmContenedorPrincipal.java

```

public FrmContenedorPrincipal() {
    initComponents();
    Principal principal = new Principal();
    dskPrincipal.add(principal);
    try {
        principal.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    principal.setVisible(true);
}

```

Ocultando Frames

Si bien solo es visible un elemento en la ventana, los Frames se van acumulando, por lo que es deseable limpiarlos antes de cargar uno nuevo, de tal manera que solo haya uno en pantalla.

FrmContenedorPrincipal.java

```

private void
opcNuevaPresentacionActionPerformed(java.awt.event.ActionEvent evt) {
    NuevaPresentacion nuevaPresentacion = new NuevaPresentacion();
    dskPrincipal.removeAll();
    dskPrincipal.add(nuevaPresentacion);
    try {
        nuevaPresentacion.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    nuevaPresentacion.setVisible(true);
    dskPrincipal.repaint();
}

```

```

        private void
opcMisPresentacionesActionPerformed(java.awt.event.ActionEvent evt) {
    MisPresentaciones misPresentaciones = new MisPresentaciones();
    dskPrincipal.removeAll();
    dskPrincipal.add(misPresentaciones);
    try {
        misPresentaciones.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    misPresentaciones.setVisible(true);
    dskPrincipal.repaint();
}

        private void
opcNuevoCongresoActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoCongreso nuevoCongreso = new NuevoCongreso();
    dskPrincipal.removeAll();
    dskPrincipal.add(nuevoCongreso);
    try {
        nuevoCongreso.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    nuevoCongreso.setVisible(true);
    dskPrincipal.repaint();
}

        private void menuInicioActionPerformed(java.awt.event.ActionEvent
evt) {
    Principal principal = new Principal();
    dskPrincipal.removeAll();
    dskPrincipal.add(principal);
    try {
        principal.setMaximum(true);
    } catch (PropertyVetoException ex) {

Logger.getLogger(FrmContenedorPrincipal.class.getName()).log(Level.SEVERE
, null, ex);
    }
    principal.setVisible(true);
    dskPrincipal.repaint();
}

```

Con esto finaliza la práctica de Navegación con menú, posteriormente se trabajará con el ajuste de tamaños de pantallas y elementos.