

Swing 14 – Plantillas

En esta práctica se manejarán las ventanas como si fueran plantillas, se usará el ejemplo de un cuestionario que presentará distintas preguntas.

Se comenzará descargando el proyecto SW14Base.zip, una vez descargado y cargado en el entorno de trabajo se comenzará haciendo una diferencia en la ventana de menú, para que tenga un comportamiento para un administrador (**admin,31749**) y para usuarios (**usrA,usrAAA**) (**usrB,usrBBB**).

MenuFrm.java

```
public MenuFrm() {
    initComponents();
    cargarDatos();
}
```

MenuFrm.java

```
private void cargarDatos(){
    DatosUsuario datos = new DatosUsuario();

    datos.nombreUsuario(Compartido.login);
    String descripcion = datos.descripcion(Compartido.tipoUsuario);

    lblNombre.setText(Compartido.nombreUsuario);
    lblDescripcion.setText(descripcion);

    if(Compartido.tipoUsuario.compareTo("admin")==0){

        btnOpcionA.setText("Consultar Resultados");
        btnOpcionB.setText("Otra Opción de Administrador");
        btnOpcionC.setVisible(false);

    }else{
        btnOpcionA.setText("Comenzar Prueba");
        btnOpcionB.setText("Otra Opción de Usuario 1");
        btnOpcionC.setText("Otra Opción de Usuario 2");
    }
}
```

Se ocultan ciertos botones y se les da una etiqueta dependiendo del tipo de usuario.

También se dará un comportamiento al presionar el botón dependiendo el tipo de usuarios

MenuFrm.java

```
private void btnOpcionAActionPerformed(java.awt.event.ActionEvent evt) {
    if(Compartido.tipoUsuario.compareTo("admin")==0){
        this.dispose();
        ConsultaResultadosFrm consulta = new ConsultaResultadosFrm();
        consulta.setVisible(true);
    }else{
        this.dispose();
        PreguntaFrm pregunta = new PreguntaFrm();
        pregunta.setVisible(true);
    }
}

private void btnOpcionBActionPerformed(java.awt.event.ActionEvent evt) {
    if(Compartido.tipoUsuario.compareTo("admin")==0){
        System.out.println("Otra opción del Administrador");
    }else{
        System.out.println("Otra opción del Usuario");
    }
}

private void btnOpcionCActionPerformed(java.awt.event.ActionEvent evt) {
    System.out.println("Otra opción solo del usuario");
}
```

Desplegando Preguntas

La ventana de pregunta se utilizará como una plantilla para desplegar cada una de las preguntas que se deben responder en el cuestionario.

PreguntaFrm.java

```
Pregunta pregunta;

public PreguntaFrm() {
    initComponents();
    datosCuestionario();
    cargaPregunta();
}
```

PreguntaFrm.java

```
private void datosCuestionario(){
    OperacionesCuestionario operaciones = new OperacionesCuestionario();
    Cuestionario cuestionario = operaciones.datosCuestionario();
}
```

```

lblTituloCuestionario.setText(cuestionario.getTituloCuestionario());
lblNombre.setText(Compartido.nombreUsuario);
}

```

PreguntaFrm.java

```

private void cargaPregunta(){
    int preguntaActual = Compartido.preguntaActual;
    OperacionesCuestionario operaciones = new OperacionesCuestionario();
    pregunta = operaciones.regresaPregunta(preguntaActual);
    lblTituloPregunta.setText(pregunta.getPregunta());
    lblPregunta.setText("Pregunta " + pregunta.getId() + " de "+Compartido.totalPreguntas);
    if(pregunta.getId()<=Compartido.totalPreguntas){
        btnAvanzar.setText("Siguiente");
    }
}

```

PreguntarFrm.java

```

private void btnAvanzarActionPerformed(java.awt.event.ActionEvent evt) {
    System.out.println("En la pregunta " + pregunta.getId() + " el usuario " + Compartido.login);
    Compartido.preguntaActual = pregunta.getId()+1;

    if(pregunta.getId() != Compartido.totalPreguntas){
        this.dispose();
        PreguntaFrm preguntaFrm = new PreguntaFrm();
        preguntaFrm.setVisible(true);
    }else{
        this.dispose();
        ResultadoFrm resultadoFrm = new ResultadoFrm();
        resultadoFrm.setVisible(true);
    }
}

```

Desplegando las Opciones de Combo Box

Se presentará el combo box con la lista de opciones para el caso de las preguntas que son de ese tipo.

PreguntaFrm.java

```

private void cargaPregunta(){
    int preguntaActual = Compartido.preguntaActual;
    OperacionesCuestionario operaciones = new OperacionesCuestionario();
    pregunta = operaciones.regresaPregunta(preguntaActual);
}

```

```

if(pregunta.getTipoPregunta().compareTo("CB")==0){
    construyeComboBox(pregunta);
}
lblTituloPregunta.setText(pregunta.getPregunta());
lblPregunta.setText("Pregunta " + pregunta.getId() + " de "+Compartido.totalPreguntas);
if(pregunta.getId()<=Compartido.totalPreguntas){
    btnAvanzar.setText("Siguiente");
}
}

```

PreguntaFrm.java

```

Pregunta pregunta;
JComboBox<String> listaRespuestas;

public PreguntaFrm() {
    initComponents();
    datosCuestionario();
    cargaPregunta();
}

```

PreguntaFrm.java

```

public void construyeComboBox(Pregunta pregunta){
    listaRespuestas = new JComboBox<String>();

    listaRespuestas.addItem("Selecciona tu respuesta");
    LinkedList<String>opciones = pregunta.getListaOpciones();
    for(int i=0;i<opciones.size();i++){
        listaRespuestas.addItem(opciones.get(i));
    }
    listaRespuestas.setSelectedIndex(0);

    double ancho = listaRespuestas.getPreferredSize().getWidth();
    int iancho = (int)ancho+100;

    listaRespuestas.setBounds(159, 81, iancho, 40);

    pnlContenedorRespuesta.add(listaRespuestas);
    listaRespuestas.setVisible(true);
}

```

Como no existía el componente, se debe crear, llenarlo, asignarle un tamaño y posición, agregarlo a un panel o frame y visualizarlo.

Leyendo la Respuesta del Combo Box

PreguntaFrm.java

```
private void btnAvanzarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String resultado="";  
  
    if(pregunta.getTipoPregunta().compareTo("CB")==0){  
        resultado = (String)listaRespuestas.getSelectedItem();  
    }  
    System.out.println("En la pregunta " + pregunta.getId() + " el usuario " + Compartido.login + "  
escogió " + resultado);  
    Compartido.preguntaActual = pregunta.getId()+1;  
  
    if(pregunta.getId() != Compartido.totalPreguntas){  
        this.dispose();  
        PreguntaFrm preguntaFrm = new PreguntaFrm();  
        preguntaFrm.setVisible(true);  
    }else{  
        this.dispose();  
        ResultadoFrm resultadoFrm = new ResultadoFrm();  
        resultadoFrm.setVisible(true);  
  
    }  
  
}
```

Desplegando las Opciones de Radio Button

PreguntaFrm.java

```
private void cargaPregunta(){  
    int preguntaActual = Compartido.preguntaActual;  
    OperacionesCuestionario operaciones = new OperacionesCuestionario();  
    pregunta = operaciones.regresaPregunta(preguntaActual);  
  
    if(pregunta.getTipoPregunta().compareTo("CB")==0){  
        construyeComboBox(pregunta);  
    }  
    if(pregunta.getTipoPregunta().compareTo("RB")==0){  
        construyeRadioButton(pregunta);  
    }  
}
```

```

lblTituloPregunta.setText(pregunta.getPregunta());
lblPregunta.setText("Pregunta " + pregunta.getId() + " de "+Compartido.totalPreguntas);
if(pregunta.getId()<=Compartido.totalPreguntas){
    btnAvanzar.setText("Siguiente");
}
}

```

PreguntaFrm.java

```

Pregunta pregunta;
JComboBox<String>listaRespuestas;
ButtonGroup btnGrp;

public PreguntaFrm() {
    initComponents();
    datosCuestionario();
    cargaPregunta();
}

```

PreguntaFrm.java

```

public void construyeRadioButton(Pregunta pregunta){

    btnGrp = new ButtonGroup();
    int inicio = 20;
    LinkedList<String>opciones = pregunta.getListaOpciones();
    for(int i=0;i<opciones.size();i++){
        JRadioButton btn = new JRadioButton();
        btn.setText(opciones.get(i));
        btnGrp.add(btn);
        btn.setBounds(100, inicio, 200, 40);
        inicio+=30;
        pnlContenedorRespuesta.add(btn);
        btn.setVisible(true);
        btn.setSelected(true);
    }
}

```

Leyendo las respuestas de un Radio Button

PreguntaFrm.java

```
private void btnAvanzarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String resultado="";  
  
    if(pregunta.getTipoPregunta().compareTo("CB")==0){  
        resultado = (String)listaRespuestas.getSelectedItem();  
  
    }  
    if(pregunta.getTipoPregunta().compareTo("RB")==0){  
        Enumeration<AbstractButton>elementos = btnGrp.getElements();  
        while(elementos.hasMoreElements()){  
            JRadioButton btnAux = (JRadioButton) elementos.nextElement();  
            if(btnAux.isSelected()){  
                resultado = btnAux.getText();  
            }  
        }  
    }  
  
    System.out.println("En la pregunta " + pregunta.getId() + " el usuario " + Compartido.login + "  
escogió " + resultado);  
    Compartido.preguntaActual = pregunta.getId()+1;  
  
    if(pregunta.getId() != Compartido.totalPreguntas){  
        this.dispose();  
        PreguntaFrm preguntaFrm = new PreguntaFrm();  
        preguntaFrm.setVisible(true);  
    }else{  
        this.dispose();  
        ResultadoFrm resultadoFrm = new ResultadoFrm();  
        resultadoFrm.setVisible(true);  
  
    }  
  
}
```