

Código para centrar un JFrame o JDialog

```
public principal() {  
    initComponents();  
    this.setLocationRelativeTo(null); //pone el JFrame centrado  
}
```

Código para limitar la introducción de 5 dígitos en el cuadro de texto Código.

```
private void txtCodigoKeyTyped(java.awt.event.KeyEvent evt) {  
    char TestChar=evt.getKeyChar();  
    if(!(Character.isDigit(TestChar)) || txtCodigo.getText().length() >= 5)  
    {  
        evt.consume();  
    }  
}
```

Código para limitar la introducción de 20 caracteres incluyendo espacios en el cuadro de texto Nombre

```
private void txtNombreKeyTyped(java.awt.event.KeyEvent evt) {  
    char TestChar=evt.getKeyChar();  
    if(!(Character.isLetter(TestChar)) && TestChar!=KeyEvent.VK_SPACE ||  
    txtNombre.getText().length() >= 20)  
    {  
        evt.consume();  
    }  
}
```

Código para limitar la introducción de 30 caracteres incluyendo espacios en el cuadro de texto Apellidos

```
private void txtApellidosKeyTyped(java.awt.event.KeyEvent evt) {  
    char TestChar=evt.getKeyChar();  
    if(!(Character.isLetter(TestChar)) && TestChar!=KeyEvent.VK_SPACE ||  
    txtApellidos.getText().length() >= 30)  
    {  
        evt.consume();  
    }  
}
```

Código para limitar la introducción de 30 caracteres de todo tipo en el cuadro de texto Direccion

```
private void txtDireccionKeyTyped(java.awt.event.KeyEvent evt) {  
    char TestChar=evt.getKeyChar();  
    if(txtDireccion.getText().length() >= 30)  
    {  
        evt.consume();  
    }  
}
```

Código para limitar la introducción de 9 dígitos en el cuadro de texto Teléfono

```
private void txtTelefonoKeyTyped(java.awt.event.KeyEvent evt) {  
    char TestChar=evt.getKeyChar();  
    if(!(Character.isDigit(TestChar)) || txtTelefono.getText().length() >= 9)  
    {  
        evt.consume();  
    }  
}
```

Código para controlar la fecha con una mascara ¡¡CUIDADO!! Hay que cambiar la máscara en el componente

1. En el componente donde se vaya a poner la fecha (Componente **Formatted Field**)
 - a. Propiedades
 - b. formatterFactory
 - c. Categoría -->Mask
 - d. Formato: ##/##/#### (dependerá del formato de salida que quieran darle, pueden ser guiones, puntos,...)

Este código se añade para que se ejecute CUANDO PIERDE EL FOCO (*focusLost*)

```
private void txtFechaAltaFocusLost(java.awt.event.FocusEvent evt) {  
    validarFecha();  
}
```

```
void validarFecha(){ //Declaro este método para reutilizar la validación de la fecha si quiero  
    try {  
        String fecha = txtFechaAlta.getText();  
        SimpleDateFormat formatoFecha = new SimpleDateFormat("dd/mm/yyyy",  
            Locale.getDefault());  
        formatoFecha.setLenient(false);  
        formatoFecha.parse(fecha);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null,"La fecha es incorrecta, por favor revise  
los datos introducidos");  
        txtFechaAlta.setText("");  
    }  
}
```

```

/*
    configuro lo que hace al hacer click con el raton sobre el boton Pavimento
    ESTE CODIGO ES SI SE USA UN COMBOBOX Y LO QUE HACE ES BORRAR TODOS LOS
    ELEMENTOS E IR AÑADIENDO ELEMENTOS QUE QUIERO QUE APAREZCAN
*/
private void btnPavimentoMouseClicked(java.awt.event.MouseEvent evt) {
    comboTipoReforma.removeAllItems();// borro lo que haya en los tipos de reformas
    comboTipoReforma.addItem("Estructura");// y voy añadiendo los items que correspondan
    comboTipoReforma.addItem("Cimentacion");
}

```

```

/*
    ESTE CODIGO ES POR SI TENGO QUE ACTIVAR UN CHECKBOTTOM. RESETEA TODO Y
    HABILITA LOS QUE NECESITE. Para reutilizar código, hago dos partes:
*/

```

// 1.Codigo del Boton de opción

```

private void rdoPavimentoActionPerformed(java.awt.event.ActionEvent evt) {
    //Limpiamos los check que pudieran estar seleccionados previamente
    limpiarServiciosSeleccionados();
    //Activamos las opciones que se pueden seleccionar
    chkCimentacion.setEnabled(true);
    chkEstructura.setEnabled(true);
}

```

// 2. Codigo de limpiarServicios

```

void limpiarServiciosSeleccionados(){
    Component[] componentesServicios = pnlServicios.getComponents() ;
    for (int i = 0; i < componentesServicios.length; i++)
    {
        ((JCheckBox) componentesServicios[i]).setSelected(false);
        ((JCheckBox) componentesServicios[i]).setEnabled(false);
    }
}

```

configuro cómo se comporta la elección de empleados dependiendo del encargado elegido.

```
*/
private void cmbEncargadosActionPerformed(java.awt.event.ActionEvent evt) {
    double tipoItem;
    if (cmbEncargados.getSelectedIndex()==0){//Si el encargado es: Empleados propios
        spinnerNumeroEmpleados.setEnabled(false);// las opciones de empleados no son
        visibles ni se habilitan
        spinnerNumeroEmpleados.setVisible(false);
        etiNumeroOperarios.setVisible(false);
    }
    if (cmbEncargados.getSelectedIndex()==1){//Si el item seleccionado es el segundo
        (Subcontrata)
        spinnerNumeroEmpleados.setEnabled(rootPaneCheckingEnabled);// las opciones son
        visibles y estan habilitadas
        spinnerNumeroEmpleados.setVisible(rootPaneCheckingEnabled);
        etiNumeroOperarios.setVisible(rootPaneCheckingEnabled);
    }

    if (cmbEncargados.getSelectedIndex()==2){//Si el encargado es AUTONOMO
        spinnerNumeroEmpleados.setEnabled(rootPaneCheckingEnabled);// las opciones son
        visibles y estan habilitadas
        spinnerNumeroEmpleados.setVisible(rootPaneCheckingEnabled);
        etiNumeroOperarios.setVisible(rootPaneCheckingEnabled);
    }
}
```

CODIGO DEL JFRAME (EL QUE SERÁ PANTALLA PRINCIPAL)

- Para abrir el JDialog

```
private void btnNuevaReformaActionPerformed(java.awt.event.ActionEvent evt) {  
    DialogoDatos reforma = new DialogoDatos (this, true); // true=MODAL, false=NO MODAL  
    reforma.setVisible(true);  
}
```

Si la ventana es **MODAL**, solo puedo actuar en el JDialog y no podré hacer nada con el JFrame.
Si la ventana es **NO MODAL**, poder actuar sobre el JFrame y el JDialog. Desde el JFrame podré abrir varios JDialog

- Para salir del programa

```
private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```