



Manual Técnico

ROBERTO GERVACIO EMILIO CANTÓN YANN LE LORIER

Manual Técnico

Índice

1. Introducción	2
2. La clase JFrame	2
2.1. Interfaz gráfica	2
2.2. Hacer visible la ventana gráfica	3
2.3. Construyendo la Ventana	3
2.3.1. Ventana de ingreso	3
2.3.2. Ventana Principal	5
2.3.3. Detalles	9
2.3.4. Footer	17
3. Activar Botones y TextFields	17
3.1. En la ventana Login (Método)	17
3.2. En la ventana Principal	20
4. La base de datos	20
5. Métodos	20
5.1. Desplegar los detalles del empleado	20
5.2. Habilitar/Deshabilitar las Ventanas	21
5.3. Validar Datos	22
5.4. Cálculo del ISR	22
5.5. Agregar a la Base de Datos	22
5.6. LectorCSV	24
5.7. BorrardeBD	24
5.8. Generar Reporte	24
5.9. escritorCSV	24

1. Introducción

Este programa está diseñado para una pequeña empresa que desea calcular nóminas, al ingresar datos, se genera un archivo en formato *.csv* que en realidad es la base de datos del programa.

2. La clase JFrame

2.1. Interfaz gráfica

Nuestra interfaz está diseñada con la extensión de *JFrame*. De aquí se declaran las variables que vienen en:

- *JPanel*
- *JButton*
- *TextField*
- *PasswordField*
- *JList* <*String*>
- *DefaultListModel* <*String*>
- *ScrollPane*

Al generar la ventana gráfica, se necesita definir cómo va organizada ésta última:

```
{
    private static final long serialVersionUID = 1L; //
        ver para que sirve pero quita una advertencia
    public static void main(String[] args) throws
        IOException
    {
        Interfaz ventanaGrafica = new Interfaz();
        ventanaGrafica.setVisible(true); // se abra
            la ventana en la ejecucion
    }
}
```

2.2. Hacer visible la ventana gráfica

Aquí vemos cómo corremos la ventana para hacerla visible:

```
public static void main(String[] args) throws
    IOException
{
    Interfaz ventanaGrafica = new Interfaz();
    ventanaGrafica.setVisible(true); // se abra
    la ventana en la ejecucion
}
```

2.3. Construyendo la Ventana

2.3.1. Ventana de ingreso

```
/* ===establecer propiedades de la ventana
== */
    setDefaultCloseOperation(
        WindowConstants.EXIT_ON_CLOSE);
    // para que se termine la
    execucion cuando se cierra
    setResizable(false); // desabilita
    la opcion de cambiar tamaño
    setSize(new java.awt.Dimension
        (1000, 550)); // Establecer
    tamaño
    setTitle("miSueldo"); // titulo de la
    ventana
    setLayout(new java.awt.
        GridBagLayout()); // para ordenar
    los elementos con coordenadas
/*
=====
*/

/* ===Armando la interfaz=== */
// Panel de Log In
    panelLogIn = new JPanel(); // panel
    derecho area log in
```

```

//panelLogIn.setBackground(new java
    .awt.Color(187, 72, 72)); //dar
    color
panelLogIn.setLayout(new java.awt.
    GridLayout(3,1,0,5)); // ordenar
    elementos (3 paneles)
    panelLogInAux = new JPanel
        (); // panel aux
        contendra al label Log
        in
        panelLogInAux.
            setLayout(new
                java.awt.
                BorderLayout());
            // posicionar la
            etiqueta
        panelLogInAux.add(
            new JLabel(" ____
                _____Log_
                in"), java.awt.
                BorderLayout.
                SOUTH);
panelLogInContent = new
    JPanel(); // tendra
    textfields
    panelLogInContent.
        add(new JLabel("
            Usuario"));
    txtUser = new
        JTextField(10);
    txtUser.
        setToolTipText("
            Ingrese su
            Usuario");
    panelLogInContent.
        add(txtUser);
    panelLogInContent.
        add(new JLabel("
            Contraseña"));
    txtPass = new

```

```

        JPasswordField
        (10);
        txtPass.
            setToolTipText("
            Ingrese _
            Contraseña");
        panelLogInContent.
            add(txtPass);
        panelLogInAux2 = new JPanel
            (); // tendra el boton de
            ingresar
            btnIngresar = new
                JButton("
                Ingresar");
        panelLogInAux2.add(
            btnIngresar);

        panelLogIn.add(panelLogInAux); //
            primer panel agregado
        panelLogIn.add(panelLogInContent);
            // segundo panel agregado
        panelLogIn.add(panelLogInAux2); //
            tercer panel agregado

```

2.3.2. Ventana Principal

```

        panelMain = new JPanel(); // panel
            de en medio
        panelMain.setBackground(new java.
            awt.Color(189, 195, 199)); //
            darle color
        panelMain.setLayout(new java.awt.
            BorderLayout()); // para ordenar
            panelMainTitle = new JPanel
                (); // panel donde esta
                lbl y botones
        panelMainTitle.
            setBackground(new java.
                awt.Color(189, 195, 199)
            );
            panelMainButtons

```

```

        = new
        JPanel()
        ;//panel
        para
        los
        botones
        dentro
        del
        maintitle

panelMainButtons
    .
    setBackground
    (new
    java.awt
    .Color
    (189,
    195,
    199));
    btnAgregar

    =

    new

    JButton
    (
    "
    Nuevo
    "
    )
    ;

    btnAgregar
    .
    setToolTipTex
    (
    "
    Crear
    "
    )

```

```

        nuevo
        {
            perfil
        }
    }
    ;

panelMainButtons
{
    add
    (
        btnAgregar
    )
    ;

btnReportesGrale
{
    =
    new
    JButton
    (
        "
        Generar
        Reportes
        "
    )
    ;

btnReportesGrale
{
    .
    setToolTipTex
    (
        "
        Genera
        el
        "
    )
}

```



```

reporte
┌
para
┌
todos
┌
los
┌
empleados
”
)
;

panelMainButtons
.
add
(
btnReportesGr
)
;

panelMainTitle.
setLayout(new
java.awt.
BorderLayout());
// para ordenar
lblTitulo =
new
JLabel(”
miSueldo
”);
lblTitulo.
setFont(
new java
.awt.
Font(””,
1, 30))
;
panelMainTitle.add(
lblTitulo ,java.awt.

```

```

        BorderLayout.NORTH); //
        para que el label quede
        hasta arriba
panelMainTitle.add(
    panelMainButtons); //
    para que se posicionen
    abajo del label
panelMainList = new JPanel
    (); //panel que contiene
    el Jlist
panelMainList.setLayout(new
    java.awt.BorderLayout()
    );
panelMainList.setBackground
    (new java.awt.Color(189,
    195, 199));

```

2.3.3. Detalles

```

panelDetails = new JPanel(); //
    panel de lado derecho
panelDetails.setBackground(new java
    .awt.Color(218,223,225));
panelDetails.setLayout(new java.awt
    .BorderLayout()); //para ordenar
    Norte Centro Sur
    panelDetailsButtons = new
        JPanel(); // panel con
        los botones editar y
        generar reporte
        btnEditar = new
            JButton("Editar"
            );
        btnEditar.
            setToolTipText("
            Modificar_datos_
            del_empleado_
            seleccionado");
        panelDetailsButtons
            .add(btnEditar);
        btnReporteInd = new

```

```

        JButton("
        Generar_Reporte"
        );
    btnReporteInd.
        setToolTipText("
        Generar_el_
        reporte_para_el_
        empleado_
        seleccionado");
    panelDetailsButtons
        .add(
            btnReporteInd);

    panelDetailsControles = new
        JPanel(); // panel con
        todos los datos
    panelDetailsControles.
        setLayout(new java.awt.
        GridLayout(16,4)); //
        para ordenar 10 filas x
        2 columnas
        panelDetailsControles
            .add(new JLabel(
                "Nombre:"));
    txtNombre = new
        JTextField(8);
    txtNombre.
        setToolTipText("
        ej._Fernando");
    panelDetailsControles
        .add(txtNombre);

    panelDetailsControles
        .add(new JLabel(
            "Apellido_
            Paterno:"));
    txtApp = new
        JTextField(8);
    txtApp.
        setToolTipText("

```

```

        ej. _Perez");
panelDetailsControles
    .add(txtApp);

panelDetailsControles
    .add(new JLabel(
        " Apellido _
        Materno:"));
txtApm = new
    JTextField(8);
txtApm.
    setToolTipText("
    ej. _Perez");
panelDetailsControles
    .add(txtApm);

panelDetailsControles
    .add(new JLabel(
        " Cargo:"));
txtCargo = new
    JTextField(8);
txtCargo.
    setToolTipText("
    Puesto _u_
    ocupacion _en _la _
    empresa");
panelDetailsControles
    .add(txtCargo);

panelDetailsControles
    .add(new JLabel(
        " Salario _Base _($
        ):"));
txtSueldo = new
    JTextField(8);
txtSueldo.
    setToolTipText("
    $$$");
panelDetailsControles
    .add(txtSueldo);

```

```

panelDetailsControles
    .add(new JLabel(
        "Fecha_de_
        ingreso:"));
txtFechaIngreso =
    new JTextField
        (8);
txtFechaIngreso.
    setToolTipText("
        dd/m/yyyy");
panelDetailsControles
    .add(
        txtFechaIngreso)
    ;

```

```

panelDetailsControles
    .add(new JLabel(
        "Numero_de_
        cuenta_de_nomina
        :"));
txtNominaNum = new
    JTextField(8);
txtNominaNum.
    setToolTipText("
        ej. _000");
panelDetailsControles
    .add(
        txtNominaNum);

```

```

panelDetailsControles
    .add(new JLabel(
        "Dias_Trabajados
        :"));
txtDiasTrabajdos =
    new JTextField
        (8);
txtDiasTrabajdos.
    setToolTipText("
        ej. _14");

```

```

panelDetailsControles
    .add(
        txtDiasTrabajdos
    );

```

```

panelDetailsControles
    .add(new JLabel(
        "Bonos: "));
txtBonos = new
    JTextField(8);
txtBonos.
    setToolTipText("
        Bonos_extra_$$$"
    );
panelDetailsControles
    .add(txtBonos);

```

```

panelDetailsControles
    .add(new JLabel(
        "Dias_feridos: "
    ));
txtFeriados = new
    JTextField(8);
txtFeriados.
    setToolTipText("
        Dias_feridos ,
        ej. 4");
panelDetailsControles
    .add(txtFeriados
    );

```

```

panelDetailsControles
    .add(new JLabel(
        "Horas_Extra: ")
    );
txtHorasExtra = new
    JTextField(8);
txtHorasExtra.
    setToolTipText("
        Cantidad_de_

```

```

        horas_trabajas_
        extra, _ej._3");
panelDetailsControles
        .add(
        txtHorasExtra);

```

```

panelDetailsControles
        .add(new JLabel(
        "Otros_
        Asignaciones:"))
        ;
txtAsignacionesOtros
        = new
        JTextField(8);
txtAsignacionesOtros
        .setToolTipText(
        "Otras_
        Asignaciones_que
        _se_deban_
        considerar_$$$")
        ;
panelDetailsControles
        .add(
        txtAsignacionesOtros
        );

```

```

panelDetailsControles
        .add(new JLabel(
        "IVA:"));
txtIVA = new
        JTextField(8);
txtIVA.
        setToolTipText("
        Deducciones_por_
        IVA_%");
panelDetailsControles
        .add(txtIVA);

```

```

panelDetailsControles
    .add(new JLabel(
        "ISR:"));
txtISR = new
    JTextField(8);
txtISR.
    setToolTipText("
        Deducciones por
        ISR %");
panelDetailsControles
    .add(txtISR);

```

```

panelDetailsControles
    .add(new JLabel(
        "Prestamos:"));
txtPrestamos = new
    JTextField(8);
txtPrestamos.
    setToolTipText("
        Deducciones en
        prestamos $$$");
panelDetailsControles
    .add(
        txtPrestamos);

```

```

panelDetailsControles
    .add(new JLabel(
        "Otras
        deducciones:"));
txtDeduccionesOtros
    = new
    JTextField(8);
txtDeduccionesOtros
    .setToolTipText(
        "Otras
        deducciones que
        se deben
        considerar $$$");
;
panelDetailsControles

```



```

        .add(
            txtDeduccionesOtros
        );

panelDetailsSaveCancel =
    new JPanel(); // panel
    que tendra boton de
    guardar
    btnGuardar = new
        JButton("Guardar
        ");
    btnGuardar.
        setToolTipText("
        Guardar los
        cambios");
    btnCancelar = new
        JButton("
        Cancelar");
    btnCancelar.
        setToolTipText("
        Ignorar Cambios"
        );
    panelDetailsSaveCancel
        .add(btnGuardar)
        ;
    panelDetailsSaveCancel
        .add(btnCancelar
        );

panelDetails.add(
    panelDetailsButtons, java.awt.
    BorderLayout.NORTH); //quede
    hasta arriba
panelDetails.add(
    panelDetailsControles); //cubra
    lo que sobre
panelDetails.add(
    panelDetailsSaveCancel, java.awt.
    BorderLayout.SOUTH); //que hasta
    abajo

```

2.3.4. Footer

```
panelFooter = new JPanel();
panelFooter.setBackground(new java.
    awt.Color(238, 238, 238));
panelFooter.setLayout(new java.awt.
    BorderLayout());//para ordenar
oeste centro este
    panelFooterHora = new
        JPanel();
    lblStatus = new JLabel("
        Esperando iniciar sesion"
    );
    panelFooterHora.add(
        lblStatus);
btnCerrar = new JButton("Cerrar");
btnCerrar.setToolTipText("Salir de
    miSuelo");
panelFooter.add(panelFooterHora,
    java.awt.BorderLayout.WEST);
panelFooter.add(btnCerrar, java.awt.
    BorderLayout.EAST);
```

3. Activar Botones y TextFields

3.1. En la ventana Login (Método)

```
public class IngresarAlSistema implements
    ActionListener
{
    public void actionPerformed(
       (ActionEvent e)
    {
        if(!session) // si no he
            iniciado
        {
            //intenta iniciar
            if(txtUser.getText
                ().equals("rob")
                &&
```

```

String.
    valueOf(
        txtPass.
            getPassword
            ( ) ).
    equals( "
123" ) )
//entra
{
    habilitaMainPanel
        ( );
    session = !
        session;
    txtUser.
        setEnabled
            ( false );
    txtPass.
        setText(
            " " );
    txtPass.
        setEnabled
            ( false );
    btnIngresar
        .setText
            ( " Salir "
            );
    lblStatus.
        setText(
            "
            CONECTADO
            " );
}
else // no entra
{
    JOptionPane
        .
        showMessageDialog
            ( null , "
            Datos _

```

```

        incorrectos
        ", "
        Error",
        JOptionPane
        .
        ERROR_MESSAGE
        );
        txtUser .
        setText (
        "");
        txtPass .
        setText (
        "");
    }
}
else // ya inicio la sesion
    , entonces va a cerrar
    session
{
    deshabilitaMainPanel
    ();
    deshabilitaPanelDetails
    ();
    session = !session;
    txtPass.setText("");
    ;
    txtUser.setText("")
    ;
    txtUser.setEnabled(
    true);
    txtPass.setEnabled(
    true);
    btnIngresar.setText
    (" Ingresar");
    lblStatus.setText("
    Esperando_inciar
    _sesion");
}
}
}

```

3.2. En la ventana Principal

A partir de la línea 485 podemos ver el método *EditarEmpleado*, que es el método que permite cambiar datos que se despliegan en la ventana derecha (Detalles). Este método llama a los métodos encargados de habilitar la edición de información.

4. La base de datos

Nuestra base de datos es un arreglo bidimensional (matriz de 100x10) en donde un renglón equivale a un perfil. Aquí el orden para cada empleado:

- Nombre
- Apellido Paterno
- Apellido materno
- Nómina
- Cargo
- Sueldo (80-...)
- Días trabajados
- Asignaciones
- Deducciones
- Fecha de ingreso (dd/m/aaaa)

Al cerrar el programa, *ie* al cerrar la sesión o cerrar la ventana, el arreglo se escribe en un archivo *bd.csv*. Al lanzar el programa una segunda vez, el archivo lee lo que ya estaba escrito en *bd.csv*. En caso de no existir un archivo en su carpeta en la que trabaja, se crea un nuevo *bd.csv*.

5. Métodos

5.1. Desplegar los detalles del empleado

```

        boolean session = false;
// Actualiza la lista para que se vean los usuarios que
// han sido agregados o que no se van los que se
// borrarón
public class ActualizaJList implements
    ActionListener
    {
        public void actionPerformed(
            ActionEvent e)
        {
            actualizaList();
        }
    }
// Iniciar Sesion
public class IngresarAlSistema implements
    ActionListener
    {
        public void actionPerformed(
            ActionEvent e)
        {
            if (!session) // si no he
                           iniciado
            {
                //intenta iniciar
                if (txtUser.getText
                    ().equals("rob")
                    &&
                    String.
                        valueOf(
                            txtPass.
                                getPassword
                                    ()).
                                equals("
                                    123"))
                    //entra
            {

```

5.2. Habilitar/Deshabilitar las Ventanas

En esta sección se llama a los métodos *habilitaPanelDetails()*, *limpiaTextFields()* y *deshabilitaMainpanel()*, así como los métodos *deshabilitaPanelDetails()* y *habil-*

taMainPanel(), se pueden apreciar desde la línea 553 a la línea 642.

5.3. Validar Datos

```
public boolean validacion() {  
  
    String nomb = txtNombre.getText(),  
        app = txtApp.getText(),  
        apm = txtApm.getText(),  
        fecha = txtFechaIngreso.  
            getText(),  
        cargo = txtCargo.getText()  
            (),  
        sldo = txtSueldo.getText()  
            (),  
        numNomina = txtNominaNum.  
            getText(),  
        dias = txtDiasTrabajados.  
            getText(),  
        horasExtra = txtHorasExtra.getText(),  
        bonos = txtBonos.getText(),  
        asignaciones =  
            txtAsignacionesOtros.  
            getText(),  
        isr = txtISR.getText(),  
        prestamos = txtPrestamos.getText(),  
        deducciones =  
            txtDeduccionesOtros.  
            getText();  
    String [] palabras = {nomb, app, apm, cargo,};  
    boolean noError = true;
```

5.4. Cálculo del ISR

De la línea 715 a la 769, se encuentra el método que calcula el ISR dependiendo del salario del empleado, este cálculo está de acuerdo al año 2017. Cualquier cambio puede ser efectuado simplemente, se trata de un método simple.

5.5. Agregar a la Base de Datos

El último paso de nuestro programa, que se encuentra a partir de la línea 777,

es simplemente meter los datos de cada empleado a nuestra matriz llamada *bd*:

```
bd[i][0] = aMayus(
    txtNombre.
    getText());
bd[i][1] = aMayus(
    txtApp.getText()
);
bd[i][2] = aMayus(
    txtApm.getText()
);
bd[i][3] =
    txtNominaNum.
    getText();
bd[i][4] = aMayus(
    txtCargo.getText
());
bd[i][5] =
    txtSueldo.
    getText();
bd[i][6] = txtDiasTrabajdos.getText();
bd[i][7] = txtHorasExtra.getText();
bd[i][8] = txtBonos.getText();
bd[i][9] = txtAsignacionesOtros.getText
();
bd[i][10] = Double.toString(ivaGlobal);
bd[i][11] = txtFeriados.getText();
bd[i][12] = txtPrestamos.getText();
bd[i][13] = txtDeduccionesOtros.getText
();
bd[i][14] =
    txtFechaIngreso.
    getText();
deshabilitaPanelDetails
();
habilitaMainPanel();
limpiaTextFields();
```

Una vez que termina de meter a la matriz, se hace llamar al método de escritura al archivo .csv

5.6. LectorCSV

5.7. BorrardeBD

Se encuentra en la línea 891, hasta la 907 podemos ver que las entradas de la matriz se están simplemente actualizando al valor de *null*, después manda llamar al método escritorCSV y el método que actualiza la lista de la ventana de nóminas (ActualizarList)

5.8. Generar Reporte

La verdadera escritura del archivo cuando agregamos un perfil sucede en la línea 963:

```
// Escribe al reporte
be.append(
    bd[i]
    ][0] + " , "
); //
Nombre
be.append(
    bd[i]
    ][1] + " , "
); // APP
be.append(
    bd[i]
    ][2] + " , "
); // APM
be.append(
    bd[i]
    ][3] + " , "
); // No.
Nomina
be.append(
    bd[i]
    ][4] + " , "
); //
Cargo
```

5.9. escritorCSV

Uno de los métodos más usados en este proyecto, va escribiendo con un *loop*

los diferentes datos de un perfil en particular, y salta una línea al final del loop

```
public void escritorCSV() {
    BufferedWriter bw = null;
try {
    d = "bd.csv";
    File archivoCSV = new File(d);
    bw = new BufferedWriter(new FileWriter(
        archivoCSV));
    for (int i = 0; i < bd.length; i++) {
        if (bd[i][0] == null) continue;

        bw.append(bd[i][0] + ",");
        bw.append(bd[i][1] + ",");
        bw.append(bd[i][2] + ",");
        bw.append(bd[i][3] + ",");
        bw.append(bd[i][4] + ",");
        bw.append(bd[i][5] + ",");
        bw.append(bd[i][6] + ",");
        bw.append(bd[i][7] + ",");
        bw.append(bd[i][8] + ",");
        bw.append(bd[i][9] + ",");
        bw.append(bd[i][10] + ",");
        bw.append(bd[i][11] + ",");
        bw.append(bd[i][12] + ",");
        bw.append(bd[i][13] + ",");
        bw.append(bd[i][14] + "\n");
        ;
    }
} catch (FileNotFoundException e1) {
    e1.printStackTrace();
} catch (IOException e1) {
    e1.printStackTrace();
} finally {
    if (bw != null) {
        try {
            bw.close();
        } catch (IOException e1) {
            e1.printStackTrace();
        }
    }
}
```

} }