
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Universidad Politécnica Salesiana

Vicerrectorado Docente

Código del Formato:	GUIA-PRL-001
Versión:	VF1.0
Elaborado por:	Directores de Área del Conocimiento Integrantes Consejo Académico
Fecha de elaboración:	2016/04/01
Revisado por:	Consejo Académico
Fecha de revisión:	2016/04/06
Aprobado por:	Lauro Fernando Pesántez Avilés Vicerrector Docente
Fecha de aprobación:	2016/14/06
Nivel de confidencialidad:	Interno

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Descripción General

Propósito


El propósito del presente documento es definir un estándar para elaborar documentación de guías de práctica de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana, con la finalidad de lograr una homogenización en la presentación de la información por parte del personal académico y técnico docente.

Alcance

El presente estándar será aplicado a toda la documentación referente a informes de prácticas de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana.

Formatos


- Formato de Guía de Práctica de Laboratorio / Talleres / Centros de Simulación – para Docentes
- Formato de Informe de Práctica de Laboratorio / Talleres / Centros de Simulación – para Estudiantes


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Clase Genéricas en Java	
OBJETIVO: Identificar los cambios importantes de Java Diseñar e Implementar las nuevas tecnicas de programación Entender la cada uno de las características nuevas en Java			
INSTRUCCIONES (Detallar las instrucciones que se dará al estudiante):	1. Revisar los conceptos fundamentales de Java		
	2. Establecer las características de Java en programación genérica		
	3. Implementar y diseñar los nuevos componentes de programación genérica		
	4. Realizar el informe respectivo según los datos solicitados.		
ACTIVIDADES POR DESARROLLAR (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Revisar la teoría y conceptos de Java 8, 9 ,10, 11, 12			
2. Diseñar e implementar las características de Java para generar una abstracción que permita realizar un CRUD,			
3. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo de programación genérica y ordenar una lista, buscar.			
4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica			
RESULTADO(S) OBTENIDO(S): Realizar procesos de investigación sobre los cambios importantes de Java Entender las aplicaciones de codificación de las nuevas características en base a la programación genérica Entender las funcionalidades adicionales de Java.			
CONCLUSIONES: Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.			
RECOMENDACIONES: Realizar el trabajo dentro del tiempo establecido.			

Docente / Técnico Docente: _____

Firma: _____

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	2	TÍTULO PRÁCTICA: Clase Genéricas en Java	
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> - Identificar los cambios importantes de Java - Diseñar e Implementar las nuevas técnicas de programación - Entender la cada uno de las características nuevas en Java - 			
ACTIVIDADES DESARROLLADAS			
<p>1. Revisar la teoría y conceptos de Java 8, 9 ,10, 11, 12</p> <p>En java de 8 para adelante se han agregado nuevas características las cuales nos ayudan a tener un código mucho más fácil de manipular y limpio, hemos utilizado las clases genéricas las cuales fueron agregadas en java 5.0, además de algunos métodos los cuales hemos revisado en clase y también hemos investigado por nuestra cuenta.</p> <p>¿Qué es una clase genérica?</p> <p>Las clases genéricas proporcionan los medios para describir el concepto de pila (o cualquier otra clase) en forma independiente de su tipo. Así podemos crear instancias de objetos con tipos específicos de la clase genérica.</p> <p>Algunos de los métodos que he utilizado:</p> <ul style="list-style-type: none"> - indexOf() = este método nos devuelve la posición mediante un entero de una cadena o un carácter. - contains() = este método nos sirve para comprobar si el String contiene otra subcadena. - Predicated = Una lambda Predicate en Java sirve para comprobar si una condición dada es verdadera o falsa. - stream() = Los Streams son una secuencia de elementos que soportan operaciones de agregación secuencial y paralela. - filter() = devuelve un listado que va desde cero a un listado que tiene como máximo el tamaño del listado de entrada. <p>¿Qué son las lambdas?</p> <p>Lambda es un intento que ha hecho Java en su versión 8 de incluir el paradigma de programación funcional, el cual es un paradigma diferente al de la programación imperativa. Está basado en un lenguaje matemático formal</p> <p>¿Para qué nos sirve las lambdas?</p> <p>lambda pueden ser utilizadas para contener funcionalidades que no necesitan ser nombradas y normalmente se utilizan en un tiempo corto. Algunos ejemplos notables incluyen las clausuras y la currificación.</p>			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

2. Diseñar e implementar las características de Java para generar una abstracción que permita realizar un CRUD,
4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica

Clase Persona

Estas clases son las encargadas de manipular gestionar y actualizar los datos. Es la parte lógica del programa.

```
package ec.edu.ups.modelo;

import java.util.Objects;

/**
 *
 * @author santi
 */
public class Persona {

    private String cedula;
    private String nombre;
    private String apellido;
    private String direccion;

    public Persona(String cedula, String nombre, String apellido, String direccion)
    {
        this.cedula = cedula;
        this.nombre = nombre;
        this.apellido = apellido;
        this.direccion = direccion;
    }

    public Persona(String cedula) {


        this.cedula = cedula;

    }

    public String getCedula() {
        return cedula;
    }

    public void setCedula(String cedula) {
        this.cedula = cedula;
    }

    public String getNombre() {
        return nombre;
    }
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getApellido() {
    return apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getDireccion() {
    return direccion;
}


public void setDireccion(String direccion) {
    this.direccion = direccion;
}

@Override
public int hashCode() {
    int hash = 5;
    hash = 47 * hash + Objects.hashCode(this.cedula);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Persona other = (Persona) obj;
    if (!Objects.equals(this.cedula, other.cedula)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Persona{" + "cedula=" + cedula + ", nombre=" + nombre + ",
apellido=" + apellido + ", direccion=" + direccion + '}';
}
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Clase Teléfono

```
package ec.edu.ups.modelo;

import java.util.Objects;

/**
 *
 * @author santi
 */
public class Telefono {

    private String numero;
    private String operadora;
    private String tipo;

    public Telefono(String numero, String operadora, String tipo) {
        this.numero = numero;
        this.operadora = operadora;
        this.tipo = tipo;
    }

    public Telefono(String numero) {

        this.numero = numero;
    }

    public String getNumero() {
        return numero;
    }


    public void setNumero(String numero) {
        this.numero = numero;
    }

    public String getOperadora() {
        return operadora;
    }

    public void setOperadora(String operadora) {
        this.operadora = operadora;
    }

    public String getTipo() {
        return tipo;
    }

    public void setTipo(String tipo) {
        this.tipo = tipo;
    }
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

@Override
public int hashCode() {
    int hash = 7;
    hash = 97 * hash + Objects.hashCode(this.numero);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Telefono other = (Telefono) obj;
    if (!Objects.equals(this.numero, other.numero)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Telefono{" + "numero=" + numero + ", operadora=" + operadora + ",
tipo=" + tipo + '}';
}
}

```

Clase Controlador Teléfono

Esta clase es la encargada de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista. Además de que hemos implementado los nuevos métodos de java.


```

package ec.edu.ups.controlador;

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import java.util.function.Predicate;

/**
 *
 * @author santi
 */
public class ControladorTelefono<T> {

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
//private C objeto; Indicando que es un objeto generico. vamos a decir mas
adelante que tipo de dato va
//a ser
private List<T> listaDeTelefono;

public ControladorTelefono() {

    listaDeTelefono = new ArrayList();

}

//Para hacer el método crear lo que hacemos es pasarle un objeto como parámetro y --
//este mismo añadirlo a nuestra lista.

public void create(T datos) {

    listaDeTelefono.add(datos);

}

/**
 *Para hacer el método read lo que hemos hecho es utilizar la ayuda tanto de lambdas
 *como de stream(), en el caso de lambdas hemos utilizada para pasarle como
 *parámetro el objeto T y hemos utilizado el stream para obtener un filter de la
 *lista y así con el método findFirst encontrar el primero que encuentre de toda la
 *lista
 */

public Optional<T> read(Predicate<T> t) {

    return this.listaDeTelefono.stream().filter(t).findFirst();

}


/**
 *Para realizar el método update lo que hemos hecho es pasarle dos objetos como
 *parámetros para que cuando utilicemos el método indexOf obtengamos la posición del
 *objeto que le queremos actualizar y así establecer en la lista la posición y los
 *datos nuevos que les vamos a asignar.
 */

public boolean update(T ob, T obN) {

    int posicion = listaDeTelefono.indexOf(ob);
    listaDeTelefono.set(posicion, obN);
    return false;

}

//Para realizar el método delete lo que haremos es pasarle como parámetro un objeto
//y preguntar si la lista contiene ese objeto entonces removemos el objeto.
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
public boolean delete(T ob) {
```

```
    if (listaDeTelefono.contains(ob)) {
        listaDeTelefono.remove(ob);
        return true;
    }
```

```
    return false;
```

```
}
```

//para hacer el método list simplemente lo que hacemos es pasarle el objeto a la lista <T> y retornamos nuestra lista que hayamos creado

```
public List<T> findAll() {
```

```
    return listaDeTelefono;
```

```
}
```

```
}
```

➤ Vistas

○ Vista Principal

```
package ec.edu.ups.vista;
import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.modelo.Persona;
import ec.edu.ups.modelo.Telefono;
```

```
/**
```

```
 *
```

```
 * @author santi
```


```
 */
```

```
public class VistaPrincipal extends javax.swing.JFrame {
```

```
    ControladorTelefono<Persona>controladorTelefono;
    ControladorTelefono<Telefono>controladorTelefono1;
    VistaGestionUsuario vistaGestionUsuario;
    VistaGestionTelefono vistaGestionTelefono;
```

```
public VistaPrincipal() {
    initComponents();
```

```
    controladorTelefono = new ControladorTelefono();
    controladorTelefono1 = new ControladorTelefono();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

vistaGestionUsuario = new VistaGestionUsuario(controladorTelefono);
vistaGestionTelefono = new VistaGestionTelefono(controladorTelefono1);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    desktopPane = new javax.swing.JDesktopPane();
    menuBar = new javax.swing.JMenuBar();
    FileInicio = new javax.swing.JMenu();
    itemGestionTelefono = new javax.swing.JMenuItem();
    itemGestionUsuario = new javax.swing.JMenuItem();
    itemSalir = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


    FileInicio.setForeground(new java.awt.Color(255, 51, 51));
    FileInicio.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/casa.png"))); // NOI18N
    FileInicio.setMnemonic('f');
    FileInicio.setText("Inicio");

    itemGestionTelefono.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/carpeta
(1).png"))); // NOI18N
    itemGestionTelefono.setMnemonic('o');
    itemGestionTelefono.setText("Gestion Telefono");
    itemGestionTelefono.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            itemGestionTelefonoActionPerformed(evt);
        }
    });
    FileInicio.add(itemGestionTelefono);

    itemGestionUsuario.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/sistema-de-
gestion-de-contenidos.png"))); // NOI18N
    itemGestionUsuario.setMnemonic('s');
    itemGestionUsuario.setText("Gestion Usuario");
    itemGestionUsuario.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            itemGestionUsuarioActionPerformed(evt);
        }
    });
    FileInicio.add(itemGestionUsuario);

    itemSalir.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/entrar.png"))); // NOI18N

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

itemSalir.setMnemonic('x');
itemSalir.setText("Salir");
itemSalir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        itemSalirActionPerformed(evt);
    }
});
FileInicio.add(itemSalir);

menuBar.add(FileInicio);

setJMenuBar(menuBar);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 342, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 273, Short.MAX_VALUE)
);

pack();
} // </editor-fold>

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

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

    desktopPane.add(vistaGestionUsuario);
    vistaGestionUsuario.setVisible(true);

}


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

    desktopPane.add(vistaGestionTelefono);
    vistaGestionTelefono.setVisible(true);

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(VistaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(VistaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(VistaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(VistaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VistaPrincipal().setVisible(true);
    }
});
}


// Variables declaration - do not modify
private javax.swing.JMenu FileInicio;
private javax.swing.JDesktopPane desktopPane;
private javax.swing.JMenuItem itemGestionTelefono;
private javax.swing.JMenuItem itemGestionUsuario;
private javax.swing.JMenuItem itemSalir;
private javax.swing.JMenuBar menuBar;
// End of variables declaration
}

    o Vista Gestion Usuario

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ec.edu.ups.vista;

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.modelo.Persona;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import javax.swing.JOptionPane;
import javax.swing.RowSorter.SortKey;
import javax.swing.SortOrder;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

/**
 *
 * @author santi
 */
public class VistaGestionUsuario extends javax.swing.JFrame {

    ControladorTelefono<Persona> controladorTelefono;

    public VistaGestionUsuario(ControladorTelefono controladorTelefono) {
        initComponents();

        this.controladorTelefono = controladorTelefono;
    }

    public void listarPersonas(Persona persona) {

        DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
        modelo.setRowCount(0);

        Object[] objeto = {persona.getCedula(), persona.getNombre(), persona.getApellido(),
            persona.getDireccion()};

        modelo.addRow(objeto);

        tblDatos.setModel(modelo);
    }

    public void datosDeLaTabla() {


        DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
        modelo.setRowCount(0);

        List<Persona> lista = controladorTelefono.findAll();

        TableRowSorter<TableModel> sorte = new TableRowSorter<>(modelo);
        tblDatos.setRowSorter(sorte);

        List<SortKey> sortKeys = new ArrayList<>();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

sortKeys.add(new SortKey(2, SortOrder.ASCENDING));
sorte.setSortKeys(sortKeys);

for (Persona persona : lista) {

    Object[] objeto = {persona.getCedula(), persona.getNombre(), persona.getApellido(),
        persona.getDireccion()};

    modelo.addRow(objeto);

    tblDatos.setModel(modelo);

}

}


/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    txtCedula = new javax.swing.JTextField();
    txtNombre = new javax.swing.JTextField();
    txtDireccion = new javax.swing.JTextField();
    btnActualizar = new javax.swing.JButton();
    btnEliminar = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    btnAtras = new javax.swing.JButton();
    jLabel7 = new javax.swing.JLabel();
    jLabel1 = new javax.swing.JLabel();
    txtBuscarPorCedula = new javax.swing.JTextField();
    btnBuscar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    txtApellido = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
    btnRegistrar = new javax.swing.JButton();
    btnLimpiar = new javax.swing.JButton();
    ListarTodo = new javax.swing.JButton();
    btnLimpiarTabla = new javax.swing.JButton();

    setClosable(true);
    setTitle("Gestion de Usuario\n");

    txtNombre.setText(" ");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtDireccion.setText(" ");

btnActualizar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/arrow_circle_double.png"))); // NOI18N
btnActualizar.setText("Actualizar");
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

btnEliminar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/borrar.png"))); //
NOI18N
btnEliminar.setText("Eliminar");
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarActionPerformed(evt);
    }
});

jLabel6.setFont(new java.awt.Font("Showcard Gothic", 1, 24)); // NOI18N
jLabel6.setText("Gestion de Usuario");

btnAtras.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/atras (1).png"))); // NOI18N
btnAtras.setText("Atras");
btnAtras.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAtrasActionPerformed(evt);
    }
});

jLabel7.setText("Apellido");


jLabel1.setText("Cedula");

btnBuscar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/find.png"))); // NOI18N
btnBuscar.setText("Buscar");
btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(evt);
    }
});

tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Cedula", "Nombre", "Apellido", "Direccion"
    }
)

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

));
tblDatos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        MouseClicked(evt);
    }
});
jScrollPane1.setViewportViewView(tblDatos);

jLabel2.setText("Cedula");

jLabel3.setText("Nombre");

txtApellido.setText(" ");

jLabel5.setText("Direccion");

btnRegistrar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/anadir.png"))); // NOI18N
btnRegistrar.setText("Registrar");
btnRegistrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRegistrarActionPerformed(evt);
    }
});

btnLimpiar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/limpiar.png"))); // NOI18N
btnLimpiar.setText("Limpiar");
btnLimpiar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLimpiarActionPerformed(evt);
    }
});

ListarTodo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/list.png"))); // NOI18N
ListarTodo.setText("Listar todo");
ListarTodo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ListarTodoActionPerformed(evt);
    }
});

btnLimpiarTabla.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/table (1).png"))); // NOI18N
btnLimpiarTabla.setText("Limpiar tabla");
btnLimpiarTabla.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLimpiarTablaActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```



```
.addComponent(txtApellido, javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(40, 40, 40))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
.addContainerGap(19, Short.MAX_VALUE)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel6)
.addComponent(btnLimpiarTabla))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtBuscarPorCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel1)
.addComponent(btnBuscar))
.addGap(20, 20, 20)
.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(30, 30, 30))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel2))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(txtNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel3))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel7)
.addComponent(txtApellido, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel5)
.addComponent(txtDireccion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(47, 47, 47)))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(btnEliminar)
.addComponent(btnAtras)
.addComponent(btnActualizar, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(btnRegistrar)
.addComponent(btnLimpiar)
```

```
.addComponent(ListarTodo, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(26, 26, 26))
);

pack();
} // </editor-fold>

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

    String cedula = txtCedula.getText();
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String direccion = txtDireccion.getText();

    if (cedula.isEmpty() || nombre.isEmpty() || apellido.isEmpty() || direccion.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Llene todos los campos");

    } else {

        Persona p = new Persona(cedula, nombre, apellido, direccion);

        controladorTelefono.update(p, p);

        if (p != null) {

            listarPersonas(p);

        }

    }

}

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

    String cedula = txtCedula.getText();

    int eliminar = JOptionPane.showConfirmDialog(this, "Esta seguro de eliminarlo");

    Persona p = new Persona(cedula);

    boolean cent = controladorTelefono.delete(p);


    if (cent && eliminar == JOptionPane.YES_OPTION) {

        datosDeLaTabla();

        JOptionPane.showMessageDialog(this, "Usuario eliminado");

    }

}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

}

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

    limpiar();
    this.dispose();

}

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

    String cedula = txtBuscarPorCedula.getText();

    if (cedula.isEmpty()) {

        limpiar();
        JOptionPane.showMessageDialog(this, "Llene el campo ");

    } else {

        Optional<Persona> persona = controladorTelefono.read(t -> cedula.equals(t.getCedula()));
        Persona p = persona.get();

        if (p != null) {

            listarPersonas(p);

        } else {

        }

    }

}

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

    String cedula = txtCedula.getText();
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String direccion = txtDireccion.getText();


    if (cedula.isEmpty() || nombre.isEmpty() || apellido.isEmpty() || direccion.isEmpty()) {

        JOptionPane.showMessageDialog(this, "Llene todos los campos");

    } else {

        Persona p = new Persona(cedula, nombre, apellido, direccion);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

controladorTelefono.create(p);

JOptionPane.showMessageDialog(this, "El usuario se ha registrado correctamente");

}

}

private void MouseClicked(java.awt.event.MouseEvent evt) {

    int fila = tblDatos.getSelectedRow();

    String cedula = (String) tblDatos.getValueAt(fila, 0);
    String nombre = (String) tblDatos.getValueAt(fila, 1);
    String apellido = (String) tblDatos.getValueAt(fila, 2);
    String direccion = (String) tblDatos.getValueAt(fila, 3);

    txtCedula.setText(cedula);
    txtNombre.setText(nombre);
    txtApellido.setText(apellido);
    txtDireccion.setText(direccion);

}

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

    limpiar();

}

private void ListarTodoActionPerformed(java.awt.event.ActionEvent evt) {
    datosDeLaTabla();
}

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

    DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
    modelo.setRowCount(0);
    for (int i = 0; i < tblDatos.getRowCount(); i++) {

        modelo.removeRow(i);
    }

    tblDatos.setModel(modelo);

}

public void limpiar() {

```

```
txtCedula.setText("");  
txtNombre.setText("");  
txtApellido.setText("");  
txtDireccion.setText("");
```

```
}
```


```
// Variables declaration - do not modify  
private javax.swing.JButton ListarTodo;  
private javax.swing.JButton btnActualizar;  
private javax.swing.JButton btnAtras;  
private javax.swing.JButton btnBuscar;  
private javax.swing.JButton btnEliminar;  
private javax.swing.JButton btnLimpiar;  
private javax.swing.JButton btnLimpiarTabla;  
private javax.swing.JButton btnRegistrar;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JTable tblDatos;  
private javax.swing.JTextField txtApellido;  
private javax.swing.JTextField txtBuscarPorCedula;  
private javax.swing.JTextField txtCedula;  
private javax.swing.JTextField txtDireccion;  
private javax.swing.JTextField txtNombre;  
// End of variables declaration  
}
```

○ Vista Gestion Telefono

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package ec.edu.ups.vista;
```

```
import ec.edu.ups.controlador.ControladorTelefono;  
import ec.edu.ups.modelo.Telefono;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Optional;  
import javax.swing.JOptionPane;  
import javax.swing.RowSorter.SortKey;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import javax.swing.SortOrder;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import javax.swing.table.TableRowSorter;

/**
 *
 * @author santi
 */
public class VistaGestionTelefono extends javax.swing.JFrame {

    /**
     * Creates new form VistaGestion
     */
    ControladorTelefono<Telefono> controladorTelefono;

    public VistaGestionTelefono(ControladorTelefono controladorTelefono) {
        initComponents();

        this.controladorTelefono = controladorTelefono;
    }

    /**
     *En este método estamos enviando los datos a la tabla y con el método TableRowSorter estamos realizando lo de ordenar
     *Creando una lista y pasándole el objeto TableRowSorter le decimos con que columna le queremos ordenar ya sea por el
     *numero la operadora y establecemos el sortKeys
     */
    public void datosDeLaTabla() {

        DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
        modelo.setRowCount(0);

        List<Telefono> lista = controladorTelefono.findAll();

        TableRowSorter<TableModel> sorte = new TableRowSorter<>(modelo);
        tblDatos.setRowSorter(sorte);


        List<SortKey> sortKeys = new ArrayList<>();
        sortKeys.add(new SortKey(0, SortOrder.ASCENDING));
        sorte.setSortKeys(sortKeys);

        for (Telefono telefono : lista) {

            Object[] objeto = {telefono.getNumero(), telefono.getOperadora(), telefono.getTipo()};
            modelo.addRow(objeto);
        }

        tblDatos.setModel(modelo);

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public void listarTabla(Telefono telefono) {

    DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
    modelo.setRowCount(0);

    Object[] objeto = {telefono.getNumero(), telefono.getOperadora(), telefono.getTipo()};
    modelo.addRow(objeto);

    tblDatos.setModel(modelo);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {


    jLabel1 = new javax.swing.JLabel();
    txtBuscarPorNumero = new javax.swing.JTextField();
    btnBuscar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblDatos = new javax.swing.JTable();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    txtNumero = new javax.swing.JTextField();
    txtOperadora = new javax.swing.JTextField();
    btnActualizar = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    btnEliminar = new javax.swing.JButton();
    btnAtras = new javax.swing.JButton();
    jLabel7 = new javax.swing.JLabel();
    txtTipo = new javax.swing.JTextField();
    btnRegistrar = new javax.swing.JButton();
    jButton1 = new javax.swing.JButton();
    btnListarTodo = new javax.swing.JButton();
    btnLimpiarTbla = new javax.swing.JButton();

    setClosable(true);
    setTitle("Gestion de Telefonos\n\n");

    jLabel1.setText("Numero");

    btnBuscar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/find.png"))); // NOI18N
    btnBuscar.setText("Buscar");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(evt);
    }
});

tblDatos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Numero", "Operadora", "Tipo"
    }
));
tblDatos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        ListaMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblDatos);

jLabel2.setText("Numero");

jLabel3.setText("Operadora");

txtOperadora.setText("");


btnActualizar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/arrow_circle_double.png"))); // NOI18N
btnActualizar.setText("Actualizar");
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

jLabel6.setFont(new java.awt.Font("Showcard Gothic", 1, 24)); // NOI18N
jLabel6.setText("Gestion de Telefonos");

btnEliminar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/borrar.png"))); //
NOI18N
btnEliminar.setText("Eliminar");
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarActionPerformed(evt);
    }
});

btnAtras.setIcon(new javax.swing.ImageIcon(getClass().getResource("/ec/ups/edu/imagenes/atras (1).png"))); // NOI18N
btnAtras.setText("Atras");
btnAtras.addActionListener(new java.awt.event.ActionListener() {


```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addComponent(btnEliminar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnAtras)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 67, Short.MAX_VALUE)
        .addComponent(btnListarTodo)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnRegistrar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(btnActualizar))
    .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 467,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel6)
                .addGroup(layout.createSequentialGroup())
                    .addComponent(jLabel1)
                    .addGap(18, 18, 18)
                    .addComponent(txtBuscarPorNumero, javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(btnBuscar))))
        .addGap(27, 27, 27)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(btnLimpiarTbla)
            .addGroup(layout.createSequentialGroup())
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel7)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jLabel2)))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(txtNumero)
                    .addComponent(txtOperadora)
                    .addComponent(txtTipo, javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(0, 0, Short.MAX_VALUE))
        .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addGap(21, 21, 21)
                    .addComponent(jLabel6))
                .addGroup(layout.createSequentialGroup())
                    .addGap(40, 40, 40)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


```

        .addComponent(btnLimpiarTbla)))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 16, Short.MAX_VALUE)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(txtBuscarPorNumero, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel1)
        .addComponent(btnBuscar))
    .addGap(21, 21, 21)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(40, 40, 40))
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtNumero, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel2))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtOperadora, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel3))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtTipo, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel7))
        .addGap(53, 53, 53)))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(btnActualizar, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(btnEliminar)
            .addComponent(btnAtras)
            .addComponent(btnRegistrar)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(btnListarTodo, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(26, 26, 26))
    );

    pack();
} // </editor-fold>

/**
 * Método Actualizar.
 */

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

*Tenemos como atributos el numero la operadora el tipo y realizamos una condición de que si los campos están vacíos no
 *pueda actualizar, creamos una instancia del Teléfono, llamamos el controlador y le pasamos el objeto y le pasamos al
 *método listar tabla.
 */

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

```
    String numero = txtNumero.getText();
    String operadora = txtOperadora.getText();
    String tipo = txtTipo.getText();
```

```
    if (numero.isEmpty() || operadora.isEmpty() || tipo.isEmpty()) {
```

```
        JOptionPane.showMessageDialog(this, "Llene los campos");
```

```
    } else {
```

```
        Telefono te = new Telefono(numero, operadora, tipo);
```

```
        controladorTelefono.update(te, te);
```

```
        if (te != null) {
```

```
            listarTabla(te);
```

```
        }
```

```
    }
```

```
}
```

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

```
    this.hide();
```

```
}
```

```
/**
```

*Para eliminar primero hacemos un condicional por si presiona el botón sin que haya puesto nada este nos dirá que ingrese el
 *número, instanciamos el teléfono y llamamos el controlador y al método delete y si el boolean es true entonces nos va a
 *eliminar el datos.

```
*/
```

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


```
    String numero = txtNumero.getText();
```

```
    int eliminar = JOptionPane.showConfirmDialog(this, "Estas seguro de eliminarlo");
```

```
    if (numero.isEmpty()) {
```

```
        JOptionPane.showMessageDialog(this, "Ingrese el numero");
```

```
    } else {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
Telefono t = new Telefono(numero);
```

```
boolean cent = controladorTelefono.delete(t);
```

```
if (cent && eliminar == JOptionPane.YES_OPTION) {
```

```
    datosDeLaTabla();
```

```
    JOptionPane.showMessageDialog(this, "dato eliminado");
```

```
}
```

```
}
```

```
}
```

```
/**
```

```
*Para buscar pasamos el número, llamamos al controlador y al método read utilizamos el lambda y le comparamos con el
```

```
*número y le llamamos y le guardamos dentro de la tabla listarTabla
```

```
*/.
```

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

```
    String numero = txtBuscarPorNumero.getText();
```

```
    if (numero.isEmpty()) {
```

```
        JOptionPane.showMessageDialog(this, "Llene el campo");
```

```
    } else {
```

```
        Optional<Telefono> telefono = controladorTelefono.read(t -> numero.equals(t.getNumero()));
```

```
        Telefono t = telefono.get();
```

```
        if (t != null) {
```

```
            listarTabla(t);
```

```
        }
```


```
    }
```

```
}
```

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

```
    String numero = txtNumero.getText();
```

```
    String operadora = txtOperadora.getText();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
String tipo = txtTipo.getText();

if (numero.isEmpty() || operadora.isEmpty() || tipo.isEmpty()) {

    JOptionPane.showMessageDialog(this, "Llene los campos");

} else {

    Telefono telefono = new Telefono(numero, operadora, tipo);

    controladorTelefono.create(telefono);

    JOptionPane.showMessageDialog(this, "Se ha registrado el telefono");

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    limpiar();

}

private void ListaMouseClicked(java.awt.event.MouseEvent evt) {

    int fila = tblDatos.getSelectedRow();

    String numero = (String) tblDatos.getValueAt(fila, 0);
    String operadora = (String) tblDatos.getValueAt(fila, 1);
    String tipo = (String) tblDatos.getValueAt(fila, 2);

    txtNumero.setText(numero);
    txtOperadora.setText(operadora);
    txtTipo.setText(tipo);


}

private void btnListarTodoActionPerformed(java.awt.event.ActionEvent evt) {
    datosDeLaTabla();
}

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

    DefaultTableModel modelo = (DefaultTableModel) tblDatos.getModel();
    modelo.setRowCount(0);
    for (int i = 0; i < modelo.getRowCount(); i++) {

        modelo.removeRow(i);
    }
}
```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

tblDatos.setModel(modelo);

}


public void limpiar() {

    txtNumero.setText("");
    txtOperadora.setText("");
    txtTipo.setText("");

}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnAtras;
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnEliminar;
private javax.swing.JButton btnLimpiarTbla;
private javax.swing.JButton btnListarTodo;
private javax.swing.JButton btnRegistrar;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblDatos;
private javax.swing.JTextField txtBuscarPorNumero;
private javax.swing.JTextField txtNumero;
private javax.swing.JTextField txtOperadora;
private javax.swing.JTextField txtTipo;
// End of variables declaration
}

```

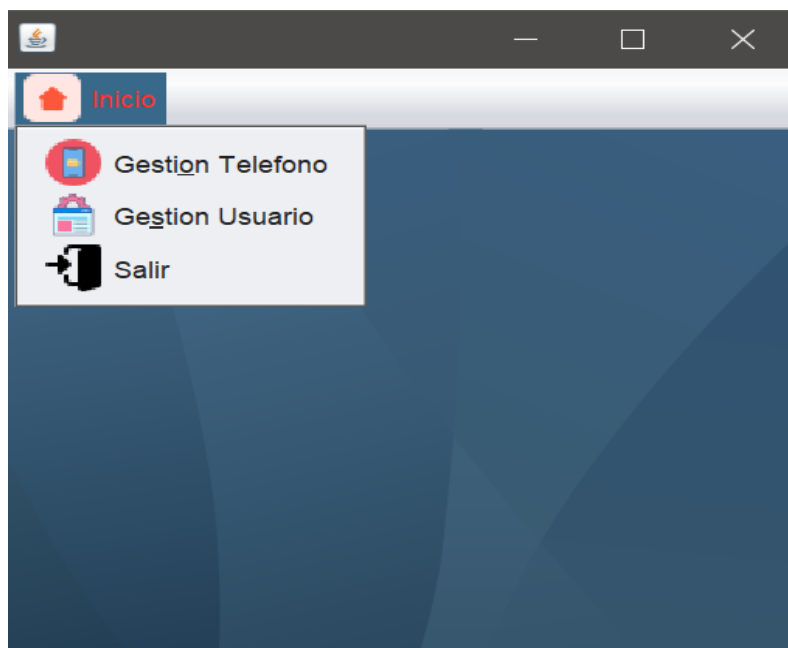
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

3. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo de programación genérica y ordenar una lista, buscar.

Aplicación creada con clase genérica.

🚦 Vista Principal de la aplicación.


En esta ventana podemos escoger si Gestionamos teléfonos o Gestionamos Usuarios o simplemente Salir.




Al presionar gestionar teléfonos o gestionar Usuarios nos salen estas interfaces en el cual podemos registrar, actualizar, eliminar y buscar el teléfono, además he agregado métodos para limpiar la tabla, limpiar los campos de texto y un botón que nos regresa a la ventana principal. A medida que registremos teléfonos o usuarios esta se ira ordenando respecto al número o al apellido.

🚦 Registrar

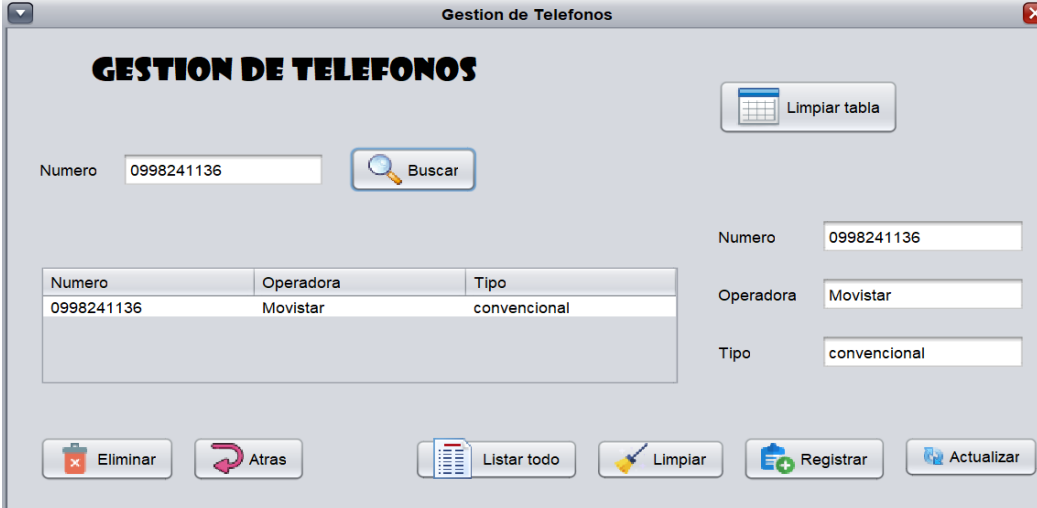
Para registrarnos ingresamos los datos en los campos del texto y pulsamos el botón registrar.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		




 Buscar

Para realizar una búsqueda simplemente en el campo de texto ingresamos el número a buscar y presionamos buscar el cual nos mostrara los datos en una tabla.




 Actualizar


Para actualizar podemos pulsar la fila que queremos actualizar en la tabla y esto nos enviara a los campos de texto los cuales simplemente cambiamos los datos que deseemos y le damos al botón actualizar.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Gestion de Telefonos







GESTION DE TELEFONOS

Numero:  Buscar

 Limpiar tabla

Numero	Operadora	Tipo
0998241136	Claro	convencional

Numero:
Operadora:
Tipo:

 Eliminar
 Atras
 Listar todo
 Limpiar
 Registrar
 Actualizar

 Eliminar

Para eliminar un teléfono lo que hacemos es seleccionar en la tabla de datos el teléfono que queramos a eliminar. Y nos saldrá una alerta de que si estamos seguros de eliminar el teléfono esto lo ponemos por si el usuario presiona el botón sin querer no se le elimine.

Gestion de Telefonos

GESTION DE TELEFONOS

Numero:  Buscar



Numero	Operadora	Tipo
0998241136	Claro	convencional

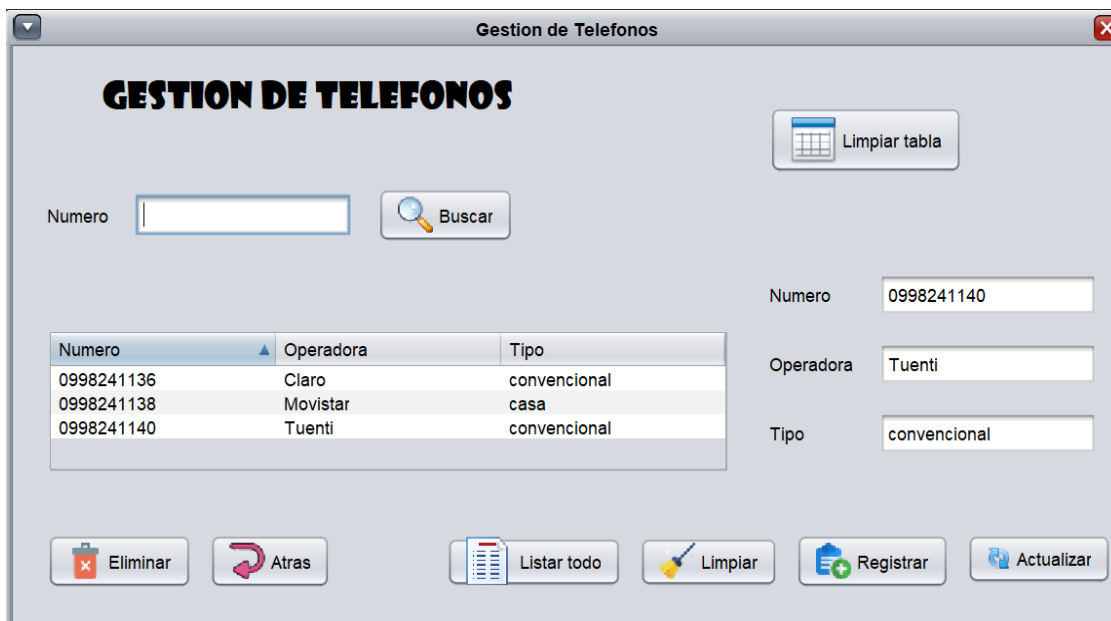
Operadora:
Tipo:

 Eliminar
 Atras
 Listar todo
 Limpiar
 Registrar
 Actualizar




 Listar

Para listar simplemente presionamos el botón listar todo y todos los teléfonos que hayamos registrado nos aparecerán en la tabla de datos además estos se irán ordenando en la tabla por el número del menor a mayor.



Numero	Operadora	Tipo
0998241136	Claro	convencional
0998241138	Movistar	casa
0998241140	Tuenti	convencional

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

➤ Vista de Gestión de Usuario.

Registrar

Para registrarnos ingresamos los datos en los campos del texto y pulsamos el botón registrar.




The screenshot shows the 'Gestion de Usuario' application window. On the left, there is a table with columns 'Cedula' and 'Nombre'. On the right, there are input fields for 'Cedula', 'Nombre', 'Apellido', and 'Direccion'. A 'Limpiar tabla' button is at the top right. A message box is displayed in the center, stating 'El usuario se ha registrado correctamente' with an 'OK' button. At the bottom, there are buttons for 'Eliminar', 'Atras', 'Listar todo', 'Limpiar', 'Registrar', and 'Actualizar'.

Buscar

Para realizar una búsqueda simplemente en el campo de texto ingresamos el número de cedula del usuario a buscar y presionamos buscar el cual nos mostrara los datos en la tabla.

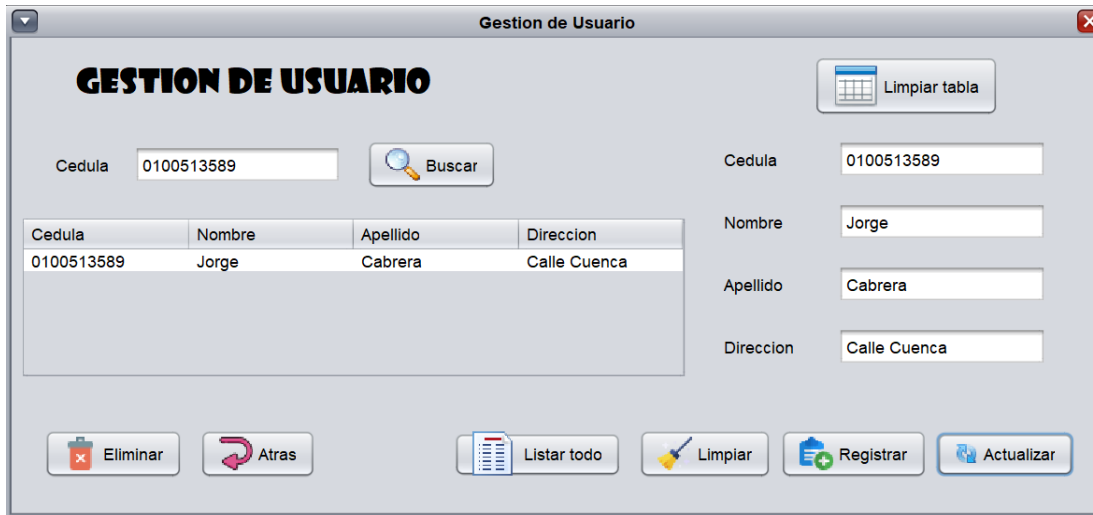


The screenshot shows the 'Gestion de Usuario' application window. The 'Cedula' input field on the left now contains the value '0100513589'. The 'Buscar' button is highlighted. The table on the left now displays the user's data: '0100513589', 'Santiago', 'Cabrera', and 'Calle Cuenca'. The right side of the window remains the same, with input fields for 'Cedula', 'Nombre', 'Apellido', and 'Direccion'.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


Actualizar

Para actualizar podemos pulsar la fila que queremos actualizar en la tabla y esto nos enviara a los campos de texto los cuales simplemente cambiamos los datos que deseemos y le damos al botón actualizar.



Gestion de Usuario

GESTION DE USUARIO

Cedula: 0100513589  Buscar

Cedula	Nombre	Apellido	Direccion
0100513589	Jorge	Cabrera	Calle Cuenca

Cedula: 0100513589
 Nombre: Jorge
 Apellido: Cabrera
 Direccion: Calle Cuenca

Eliminar

Para eliminar un usuario lo que hacemos es seleccionar en la tabla de datos el usuario que deseemos eliminar Y nos saldrá una alerta de que si estamos seguros de eliminar el usuario esto lo ponemos por si el usuario presiono el botón por error no se le elimine.



Gestion de Usuario

GESTION DE USUARIO

Cedula:  Buscar


Cedula	Nombre	Apellido	Direccion
0100513412	Pedro	Arias	Manuel Guillen
0100513123	Nicol	Bermeo	Simon Bolivar
0100513589	Jorge	Cabrera	Calle Cuenca

Cedula:
 Nombre:
 Apellido:
 Direccion:

Select an Option

 Esta seguro de eliminarlo

Apellido: Bermeo
 Direccion: Simon Bolivar

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

RESULTADO(S) OBTENIDO(S):

Realizar procesos de investigación sobre los cambios importantes de Java
Entender las aplicaciones de codificación de las nuevas características en base a la programación genérica
Entender las funcionalidades adicionales de Java.

CONCLUSIONES:

Como conclusión podemos decir que implementar las clases genéricas en el MVC implementando CRUD nos facilita mucho las cosas debido a que no debemos crear un controlador para cada clase como lo hacíamos antes si no que con un solo un controlador para diversas clases, además de la utilización de las lambdas y de los streams nos ayudan mucho a ahorrar código.

RECOMENDACIONES:

Recomiendo utilizar clases genéricas ya que nos ayudan a optimizar nuestro código al igual implementar lambdas y streams ya que con estos métodos podemos tener un código mucho más optimizado y limpio y utilizar los métodos que doy al inicio de este documento.

Nombre de estudiante: Jorge Santiago Cabrera Arias.

Firma de estudiante:



Escaneado con CamScanner