
	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: Reflexión en Java	
OBJETIVO: Identificar los cambios importantes de Java Diseñar e Implementar las nuevas tecnicas de programación Entender cada una 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 reflexión	
		3. Implementar y diseñar los nuevos componentes de reflexión	
		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, 13, 14, 15			
2. Diseñar e implementar las características de Java para generar la impresión de cualquier lista, de los modelos que tengan el campo id generar automaticamente.			
3. Probar y modificar el metodo validar para que nos permita utilizar excepciones, ademas de modificar el buscar para controlar el nullpointerexception.			
4. Realizar práctica codificando los codigos 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:	02	TÍTULO PRÁCTICA: Genérica Java	
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> • Diseñar una interfaz gráfica en la IDE NetBeans que pueda gestionar la creación, actualización, eliminación y búsqueda de teléfonos de un usuario que puede registrarse e iniciar sesión dentro de la misma interfaz que se debe desarrollar para la presentación de la aplicación. • Utilizar la ayuda de los conceptos y los métodos de un controlador directo para gestionar los métodos principales dentro de la interfaz, aplicar la teoría de las arquitecturas MVC y DAO para la gestión de los teléfonos y su respectiva acción además de la creación de usuarios y sus respectivas acciones. 			
ACTIVIDADES DESARROLLADAS			
1. Revisar la teoría y conceptos de Java 8, 9, 10, 11, 12, 13, 14, 15			
2. Diseñar e implementar las características de Java para generar la impresión de cualquier lista, de los modelos que tengan el campo id generar automáticamente. <ul style="list-style-type: none"> • Como podemos observar aquí tenemos el código correspondiente para implementar las listas en Java. <ul style="list-style-type: none"> - Listar Usuario <pre> package ec.edu.ups.vista; import ec.edu.ups.controlador.ControladorTelefono; import ec.edu.ups.controlador.ControladorUsuario; import ec.edu.ups.modelo.Telefono; import ec.edu.ups.modelo.Usuario; import java.util.Iterator; import java.util.Optional; import javax.swing.JOptionPane; import javax.swing.table.DefaultTableModel; /** * * @author paul_ */ public class ListarTelefonosUsuarios extends javax.swing.JInternalFrame { private ControladorTelefono controladorTelefono; private ControladorUsuario controladorUsuario; public ListarTelefonosUsuarios(ControladorTelefono controladorTelefono, ControladorUsuario controladorusuario) { this.controladorTelefono = controladorTelefono; this.controladorUsuario = controladorUsuario; initComponents(); } private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) { if (txtCedula == null) { </pre>			

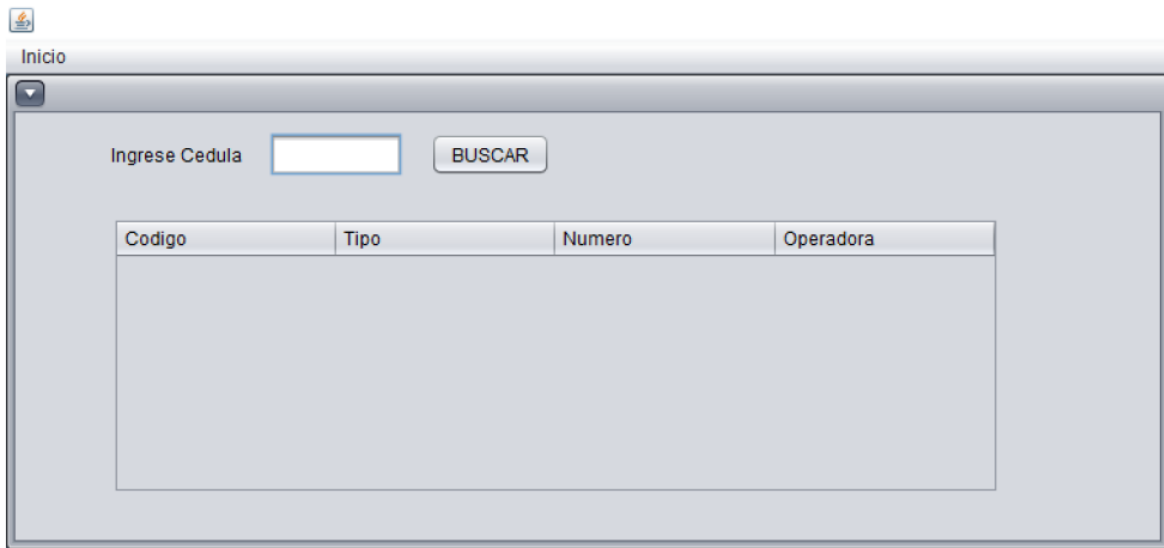
```

        JOptionPane.showMessageDialog(this, "Campo vacio");
    } else {
        cargarTelefonosTbl();
    }
}

public void cargarTelefonosTbl() {
    DefaultTableModel modelo = (DefaultTableModel) tblTelefonosU.getModel();
    Usuario comparar = new Usuario();
    comparar.setCedula(txtCedula.toString().trim());
    Optional<Usuario> usuario = controladorUsuario.buscar(comparar);
    modelo.setRowCount(0);
    for (Telefono t : controladorTelefono.listaTlfsUsers(usuario)) {
        Telefono telefono = (Telefono) t;
        Object[] rowData = {telefono.getCodigo(), telefono.getTipo(),
telefono.getNumero(), telefono.getOperadora()};
        modelo.addRow(rowData);
        tblTelefonosU.setModel(modelo);
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btnBuscar;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblTelefonosU;
private javax.swing.JTextField txtCedula;
// End of variables declaration
}

```




- Listar Telefonos

```

package ec.edu.ups.vista;

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.modelo.Telefono;
import javax.swing.table.DefaultTableModel;

```

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

```

/**
 *
 * @author paul_
 */
public class Listartelefonos extends javax.swing.JInternalFrame {

    private ControladorTelefono controladorTelefono;

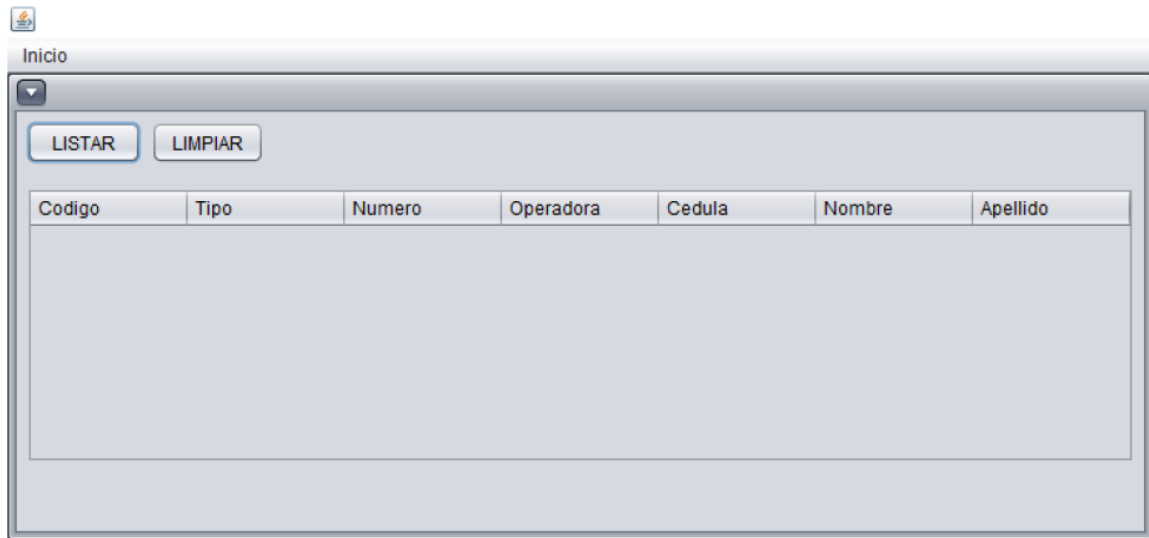
    public Listartelefonos(ControladorTelefono controladorTelefono) {
        this.controladorTelefono = controladorTelefono;

        initComponents();
    }
    private void btnListarActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonosUse.getModel();
        modelo.setRowCount(0);
        for (Telefono t : controladorTelefono.getList()) {
            Telefono telefono = (Telefono) t;
            Object[] rowData = {telefono.getCodigo(), telefono.getTipo(),
telefono.getNumero()};
            modelo.addRow(rowData);
            tblTelefonosUse.setModel(modelo);
        }
    }
    private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonosUse.getModel();
        modelo.setRowCount(0);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnLimpiar;
    private javax.swing.JButton btnListar;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable tblTelefonosUse;
    // End of variables declaration
}

```

[Interfaz gráfica de la ventana]



3. Probar y modificar el método validar para que nos permita utilizar excepciones, además de modificar el buscar para controlar el nullpointerexception.

- **Para realizar las validaciones y también hacer modificaciones implementamos los controladores.**

- **AbstractControlador**

```
package ec.edu.ups.controlador;

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


/**
 *
 * @author paul_
 */
public abstract class AbstractControlador<E> {

    private List<E> lista;

    public AbstractControlador() {
        lista = new ArrayList();
    }

    public boolean crear(E objeto) {

        if (validar(objeto) == true) {
            return lista.add(objeto);
        }
        return false;
    }
}
```

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

```

public Optional<E> buscar(E comparar) {
    return lista.stream().filter(objeto -> objeto.equals(comparar)).findFirst();
}

public int posicion(E objetoC) {
    for (int i = 0; i < lista.size(); i++) {
        E objetoL = lista.get(i);
        if (objetoL.equals(objetoC)) {
            return i;
        }
    }
    return -1;
}

public boolean eliminar(E objeto) {
    Optional<E> buscar = buscar(objeto);
    E objetoE = buscar.get();
    if (objetoE != null) {
        System.out.println("Verdadero");
        return lista.remove(objetoE);
    }
    System.out.println("Falso");
    return false;
}

public boolean actualizar(E objetoA) {
    int pos = posicion(objetoA);
    if (pos >= 0) {
        lista.set(pos, objetoA);
        System.out.println("TRUE");
        return true;
    }
    System.out.println("FALSE");
    return false;
}

public abstract boolean validar(E objeto);

public abstract int generarId();


public List<E> getLista() {
    return lista;
}

public void setLista(List<E> lista) {
    this.lista = lista;
}

```



```
}  
- Controlador Teléfono  
package ec.edu.ups.controlador;  
  
import ec.edu.ups.modelo.Telefono;  
import ec.edu.ups.modelo.Usuario;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Optional;  
  
/**  
 *  
 * @author paul_  
 */  
public class ControladorTelefono extends AbstractControlador<Telefono> {  
  
    @Override  
    public boolean validar(Telefono objeto) {  
        return true;  
    }  
  
    @Override  
    public int generarId() {  
        List<Telefono> temp = new ArrayList();  
        for (Telefono telefono : super.getList()) {  
            Telefono t = (Telefono) telefono;  
            temp.add(t);  
        }  
  
        if (temp.size() > 0 && temp != null) {  
            return temp.get(temp.size() - 1).getCodigo() + 1;  
        } else {  
            return 1;  
        }  
    }  
  
    public List<Telefono> listaTlfsUsers(Usuario usuario) {  
        List<Telefono> temp = new ArrayList();  
        for (Telefono telefono : super.getList()) {  
            Telefono t = (Telefono) telefono;  
            if (t.getUsuario().getCedula().equals(usuario.getCedula())) {  
                temp.add(t);  
            }  
        }  
  
        return temp;  
    }  
  
    public Iterable<Telefono> listaTlfsUsers(Optional<Usuario> usuario) {
```

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

```

        throw new UnsupportedOperationException("Not supported yet."); //To change
body of generated methods, choose Tools | Templates.
    }
}

- Controlador Usuario
package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Usuario;

/**
 *
 * @author paul_
 */
public class ControladorUsuario extends AbstractControlador<Usuario> {

    private Usuario usuario;


    public ControladorUsuario() {

    }

    @Override
    public boolean validar(Usuario usuario) {
        int suma = 0;
        String x = usuario.getCedula();
        if (x.length() == 9) {
            return false;
        } else {
            int a[] = new int[x.length() / 2];
            int b[] = new int[(x.length() / 2)];
            int c = 0;
            int d = 1;
            for (int i = 0; i < x.length() / 2; i++) {
                a[i] = Integer.parseInt(String.valueOf(x.charAt(c)));
                c = c + 2;
                if (i < (x.length() / 2) - 1) {
                    b[i] = Integer.parseInt(String.valueOf(x.charAt(d)));
                    d = d + 2;
                }
            }

            for (int i = 0; i < a.length; i++) {
                a[i] = a[i] * 2;
                if (a[i] > 9) {
                    a[i] = a[i] - 9;
                }
                suma = suma + a[i] + b[i];
            }
            int aux = suma / 10;
            int dec = (aux + 1) * 10;
            if ((dec - suma) == Integer.parseInt(String.valueOf(x.charAt(x.length()
- 1)))) {
                return true;
            } else if (suma % 10 == 0 && x.charAt(x.length() - 1) == '0') {

```

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

```

        return true;
    } else {
        return false;
    }
}

@Override
public int generarId() {
    return 0;
}

public boolean iniciarSesion(String correo, String pass) {

    for (Usuario usu : super.getLista()) {
        Usuario u = (Usuario) usu;
        if (u.getCorreo().equals(correo) && u.getContraseña().equals(pass)) {
            this.usuario = u;
            return true;
        }
    }
    return false;
}

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}

}

```

4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica.

- Para realizar la practica utilizamos los siguientes paquetes con sus respectivas clases:
- Ec.edu.ups.controlador

Abstract Controlador


```

package ec.edu.ups.controlador;

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

/**
 *
 * @author paul_
 */
public abstract class AbstractControlador<E> {

```

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

```

private List<E> lista;

public AbstractControlador() {
    lista = new ArrayList();
}

public boolean crear(E objeto) {

    if (validar(objeto) == true) {
        return lista.add(objeto);
    }
    return false;
}


public Optional<E> buscar(E comparar) {
    return lista.stream().filter(objeto -> objeto.equals(comparar)).findFirst();
}

public int posicion(E objetoC) {
    for (int i = 0; i < lista.size(); i++) {
        E objetoL = lista.get(i);
        if (objetoL.equals(objetoC)) {
            return i;
        }
    }
    return -1;
}

public boolean eliminar(E objeto) {
    Optional<E> buscar = buscar(objeto);
    E objetoE = buscar.get();
    if (objetoE != null) {
        System.out.println("Verdadero");
        return lista.remove(objetoE);
    }
    System.out.println("Falso");
    return false;
}

public boolean actualizar(E objetoA) {
    int pos = posicion(objetoA);
    if (pos >= 0) {
        lista.set(pos, objetoA);
        System.out.println("TRUE");
        return true;
    }
    System.out.println("FALSE");
}

```

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

```

        return false;

    }

    public abstract boolean validar(E objeto);

    public abstract int generarId();

    public List<E> getLista() {
        return lista;
    }

    public void setLista(List<E> lista) {
        this.lista = lista;
    }
}

    Controlador Telefono
package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Telefono;
import ec.edu.ups.modelo.Usuario;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;


/**
 *
 * @author paul_
 */
public class ControladorTelefono extends AbstractControlador<Telefono> {

    @Override
    public boolean validar(Telefono objeto) {
        return true;
    }

    @Override
    public int generarId() {
        List<Telefono> temp = new ArrayList();
        for (Telefono telefono : super.getLista()) {
            Telefono t = (Telefono) telefono;
            temp.add(t);
        }

        if (temp.size() > 0 && temp != null) {
            return temp.get(temp.size() - 1).getCodigo() + 1;
        } else {
            return 1;
        }
    }
}

```

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

```

public List<Telefono> listaTlfsUsers(Usuario usuario) {
    List<Telefono> temp = new ArrayList();
    for (Telefono telefono : super.getLista()) {
        Telefono t = (Telefono) telefono;
        if (t.getUsuario().getCedula().equals(usuario.getCedula())) {
            temp.add(t);
        }
    }

    return temp;
}

public Iterable<Telefono> listaTlfsUsers(Optional<Usuario> usuario) {
    throw new UnsupportedOperationException("Not supported yet."); //To change
body of generated methods, choose Tools | Templates.
}
}

Controlador Usuario
package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Usuario;

/**
 *
 * @author paul_
 */
public class ControladorUsuario extends AbstractControlador<Usuario> {

    private Usuario usuario;

    public ControladorUsuario() {
    }

    @Override
    public boolean validar(Usuario usuario) {
        int suma = 0;
        String x = usuario.getCedula();
        if (x.length() == 9) {
            return false;
        } else {
            int a[] = new int[x.length() / 2];
            int b[] = new int[(x.length() / 2)];
            int c = 0;
            int d = 1;
            for (int i = 0; i < x.length() / 2; i++) {
                a[i] = Integer.parseInt(String.valueOf(x.charAt(c)));
                c = c + 2;
                if (i < (x.length() / 2) - 1) {
                    b[i] = Integer.parseInt(String.valueOf(x.charAt(d)));
                    d = d + 2;
                }
            }
        }
    }
}

```

```
    }

    for (int i = 0; i < a.length; i++) {
        a[i] = a[i] * 2;
        if (a[i] > 9) {
            a[i] = a[i] - 9;
        }
        suma = suma + a[i] + b[i];
    }
    int aux = suma / 10;
    int dec = (aux + 1) * 10;
    if ((dec - suma) == Integer.parseInt(String.valueOf(x.charAt(x.length()
- 1)))) {
        return true;
    } else if (suma % 10 == 0 && x.charAt(x.length() - 1) == '0') {
        return true;
    } else {
        return false;
    }
}


@Override
public int generarId() {
    return 0;
}

public boolean iniciarSesion(String correo, String pass) {

    for (Usuario usu : super.getLista()) {
        Usuario u = (Usuario) usu;
        if (u.getCorreo().equals(correo) && u.getContraseña().equals(pass)) {
            this.usuario = u;
            return true;
        }
    }
    return false;
}

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}
}
```

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

- **Ec.edu.ups.modelo**

Telefono

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

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

    private int codigo;
    private String numero;
    private String tipo;
    private String operadora;
    private Usuario usuario;

    public Telefono() {
    }

    public Telefono(int codigo, String numero, String tipo, String operadora) {
        this.setCodigo(codigo);
        this.setNumero(numero);
        this.setTipo(tipo);
        this.setOperadora(operadora);
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }


    public String getNumero() {
        return numero;
    }

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

    public String getTipo() {
        return tipo;
    }

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

    public String getOperadora() {
        return operadora;
    }
}
```


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

```

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

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}

@Override
public int hashCode() {
    int hash = 7;
    hash = 13 * hash + this.codigo;
    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 (this.codigo != other.codigo) {
        return false;
    }
    return true;
}


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

    Usuario
package ec.edu.ups.modelo;

import java.util.Objects;

/**
 *
 * @author paul_
 */
public class Usuario {

```

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

```

private String Cedula;
private String nombre;
private String apellido;
private String correo;
private String contrasena;

public Usuario() {
}

public Usuario(String Cedula, String nombre, String apellido, String correo,
String contrasena) {

    this.setCedula(Cedula);
    this.setNombre(nombre);
    this.setApellido(apellido);
    this.setCorreo(correo);
    this.setContrasena(contrasena);

}

public String getCedula() {
    return Cedula;
}

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

public String getNombre() {
    return nombre;
}

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

public String getApellido() {
    return apellido;
}


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

public String getCorreo() {
    return correo;
}

public void setCorreo(String correo) {
    this.correo = correo;
}

public String getContrasena() {

```

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

```

        return contrasena;
    }

    public void setContrasena(String contrasena) {
        this.contrasena = contrasena;
    }

    @Override
    public int hashCode() {
        int hash = 7;
        hash = 41 * 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 Usuario other = (Usuario) obj;
        if (!Objects.equals(this.Cedula, other.Cedula)) {
            return false;
        }
        return true;
    }


    @Override
    public String toString() {
        return "Usuario{" + "Cedula=" + Cedula + ", nombre=" + nombre + ", apellido=" + apellido + ", correo=" + correo + ", contrasena=" + contrasena + '}';
    }
}

- Ec.edu.ups.vista
  Gestion Telefono
package ec.edu.ups.vista;

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.controlador.ControladorUsuario;
import ec.edu.ups.modelo.Telefono;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**

```

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

```

*
* @author paul_
*/
public final class GestionTelefono extends javax.swing.JInternalFrame {

    private List<String> operadoras;
    private ControladorTelefono controladorTelefono;
    private ControladorUsuario controladorUsuario;

    /**
     * Creates new form NewJInternalFrame
     *
     * @param controladorTelefono
     * @param controladorUsuario
     */
    public GestionTelefono(ControladorTelefono controladorTelefono,
ControladorUsuario controladorUsuario) {
        this.controladorTelefono = controladorTelefono;
        this.controladorUsuario = controladorUsuario;
        initComponents();

        cargarDatosOperadoras();

        // cargarTelefonosTbl();
    }


    public void cargarDatosOperadoras() {
        operadoras = new ArrayList<>();
        operadoras.add("Movistar");
        operadoras.add("Claro");
        operadoras.add("CNT");
        operadoras.add("Tuenti");
        operadoras.add("Etapa");
        cbxCargarDatos();
    }

    public void cbxCargarDatos() {
        DefaultComboBoxModel modelo = (DefaultComboBoxModel)
cbxOperadora.getModel();
        for (String operadora : operadoras) {
            modelo.addElement(operadora);
        }
    }

    public void limpiarTabla() {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonos.getModel();
        modelo.setRowCount(0);
    }

    public void cargarTelefonosTbl() {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonos.getModel();
        modelo.setRowCount(0);
        for (Telefono telefono : controladorTelefono.getListas()) {
            Telefono t = (Telefono) telefono;

```

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

```

        Object[] rowData = {telefono.getCodigo(), telefono.getTipo(),
telefono.getNumero(), telefono.getOperadora()};
        modelo.addRow(rowData);
        tblTelefonos.setModel(modelo);
    }

}


public void cargarSiguienteCodigo() {
    txtCodigo.setText(String.valueOf(controladorTelefono.generarId()));
}

public void Limpiar() {
    txtCodigo.setText("");
    txtFormatoNumero.setValue("");
    cbxTipo.setSelectedIndex(0);
    cbxOperadora.setSelectedIndex(0);
}

private void txtCodigoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cbxTipoActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String item = (String) cbxTipo.getSelectedItem();
        switch (item) {
            case "Casa":
                txtFormatoNumero.setFormatterFactory(
                    new javax.swing.text.DefaultFormatterFactory(
                        new javax.swing.text.MaskFormatter("(+593) #-
#####-####")
                    ));
                break;
            case "Movil":
                txtFormatoNumero.setFormatterFactory(
                    new javax.swing.text.DefaultFormatterFactory(
                        new javax.swing.text.MaskFormatter("(+593) #-
#####-####")
                    ));
                break;
            default:
                txtFormatoNumero.setFormatterFactory(
                    new javax.swing.text.DefaultFormatterFactory(
                        new javax.swing.text.MaskFormatter("(+593) #-
#####-####")
                    ));
                break;
        }
    } catch (java.text.ParseException ex) {
    }
}

```

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

```

private void txtFormatoNumeroActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void cbxOperadoraActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    Telefono t;
    t = new Telefono(Integer.parseInt(txtCodigo.getText().trim()),
cbxTipo.getSelectedItem().toString().trim(), txtFormatoNumero.getText().trim(),
cbxOperadora.getSelectedItem().toString().trim());
    t.setUsuario(controladorUsuario.getUsuario());
    if (controladorTelefono.crear(t)) {
        JOptionPane.showMessageDialog(this, "Telefono agregado exitosamente");
        Limpiar();
        cargarTelefonosTbl();
        cargarSiguienteCodigo();
    }

}

private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent
evt) {
    cargarSiguienteCodigo();
    cargarTelefonosTbl();
}

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

    Telefono t = new Telefono(Integer.parseInt(txtCodigo.getText()),
cbxTipo.getSelectedItem().toString().trim(), txtFormatoNumero.getText().trim(),
cbxOperadora.getSelectedItem().toString().trim());

    if (controladorTelefono.actualizar(t)) {


        JOptionPane.showMessageDialog(this, "Telefono actualizado
exitosamente");
        cargarTelefonosTbl();
        //this.dispose();
        Limpiar();
    }

}

private void tblTelefonosMouseClicked(java.awt.event.MouseEvent evt) {
    int index = tblTelefonos.getSelectedRow();

    int codigo = (int) tblTelefonos.getValueAt(index, 0);
    String numero = "" + tblTelefonos.getValueAt(index, 1);
    String tipo = "" + tblTelefonos.getValueAt(index, 2);
    String operadora = "" + tblTelefonos.getValueAt(index, 3);

```

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

```

txtCodigo.setText(String.valueOf(codigo));

cbxTipo.setSelectedItem(tipo.trim());
txtFormatoNumero.setValue(numero.trim());
cbxOperadora.setSelectedItem(operadora.trim());

}

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

    int d = JOptionPane.showConfirmDialog(this, "Esta seguro de eliminar el
telefono con codigo " + txtCodigo.getText());
    if (d == JOptionPane.YES_OPTION) {
        Telefono comparar = new Telefono();
        comparar.setCodigo(Integer.parseInt(txtCodigo.getText().trim()));
        Optional<Telefono> t = controladorTelefono.buscar(comparar);
        Telefono telefono = t.get();
        System.out.println("" + telefono);

        if (controladorTelefono.eliminar(telefono)) {
            JOptionPane.showMessageDialog(this, "Telefono eliminado
exitosamente");
            cargarTelefonosTbl();
        } else if (d == JOptionPane.NO_OPTION) {
            cargarSiguienteCodigo();
        }
        // this.dispose();
    }

}

@SuppressWarnings("empty-statement")
private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {


    String codigo = JOptionPane.showInputDialog(this, "Ingrese codigo del
telefono a buscar");
    if (codigo != null) {
        Telefono comparar = new Telefono();
        comparar.setCodigo(Integer.parseInt(codigo));
        Optional<Telefono> telefono = controladorTelefono.buscar(comparar);
        Telefono t = telefono.get();
        limpiarTabla();
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonos.getModel();
        modelo.setRowCount(0);

        Object[] rowData = {t.getCodigo(), t.getNumero(), t.getTipo(),
t.getOperadora()};
        modelo.addRow(rowData);
        tblTelefonos.setModel(modelo);
    } else {
        JOptionPane.showMessageDialog(this, "Parametro vacio");
    }

}

}

```

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

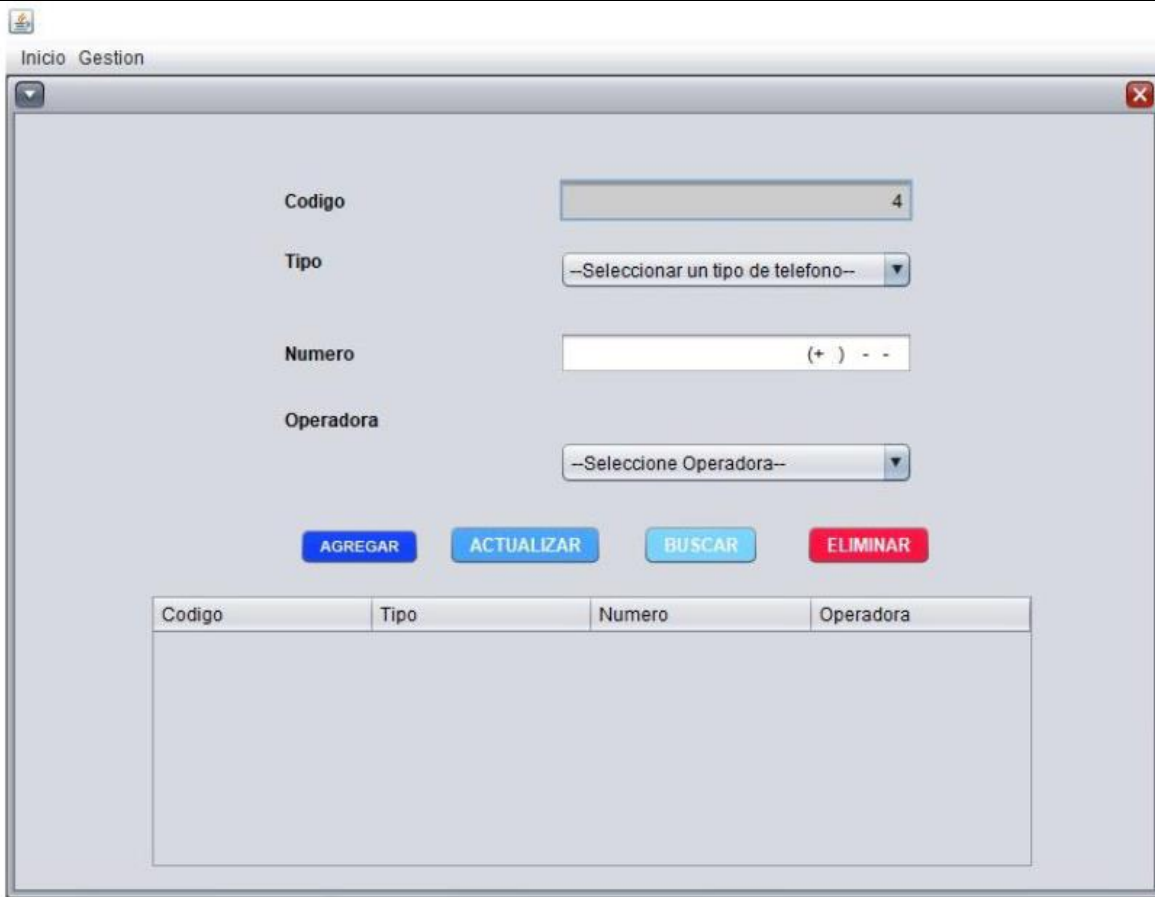
```

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

}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnActualizar1;
private javax.swing.JButton btnAgregar;
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnEliminar;
private javax.swing.JComboBox<String> cbxOperadora;
private javax.swing.JComboBox<String> cbxTipo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblTelefonos;
private javax.swing.JTextField txtCodigo;
private javax.swing.JFormattedTextField txtFormatoNumero;
// End of variables declaration
}

```

Listar Telefonos Usuarios

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

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.controlador.ControladorUsuario;
import ec.edu.ups.modelo.Telefono;
import ec.edu.ups.modelo.Usuario;
import java.util.Iterator;
import java.util.Optional;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author paul_
 */
public class ListarTelefonosUsuarios extends javax.swing.JInternalFrame {

    private ControladorTelefono controladorTelefono;
    private ControladorUsuario controladorUsuario;

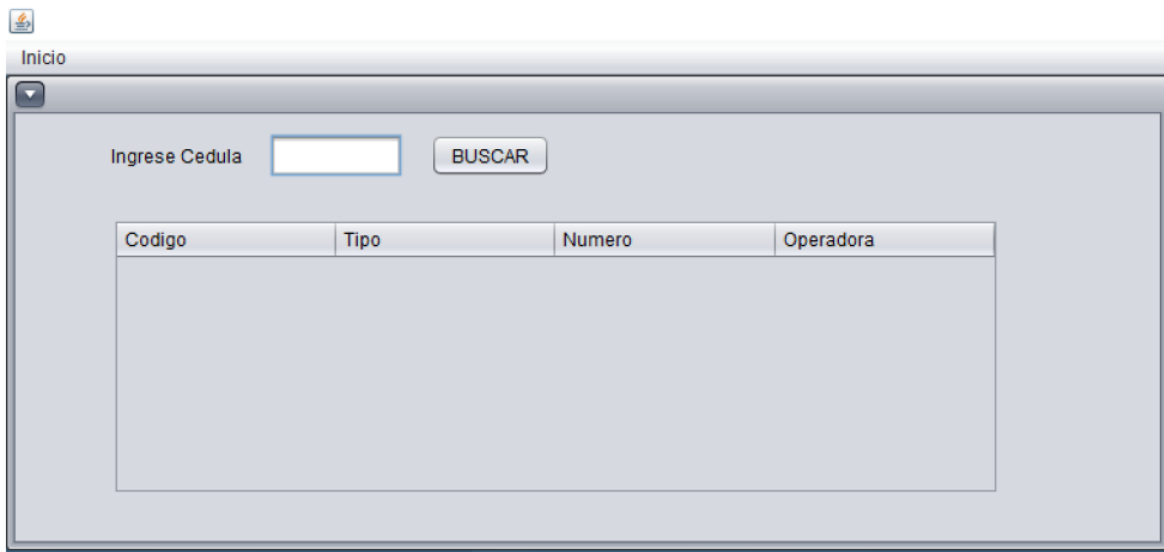
    public ListarTelefonosUsuarios(ControladorTelefono controladorTelefono,
    ControladorUsuario controladorusuario) {
        this.controladorTelefono = controladorTelefono;
    }
}
```


```

        this.controladorUsuario = controladorUsuario;
        initComponents();
    }
    private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        if (txtCedula == null) {
            JOptionPane.showMessageDialog(this, "Campo vacio");
        } else {
            cargarTelefonosTbl();
        }
    }

    public void cargarTelefonosTbl() {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonosU.getModel();
        Usuario comparar = new Usuario();
        comparar.setCedula(txtCedula.toString().trim());
        Optional<Usuario> usuario = controladorUsuario.buscar(comparar);
        modelo.setRowCount(0);
        for (Telefono t : controladorTelefono.listaTlfsUsers(usuario)) {
            Telefono telefono = (Telefono) t;
            Object[] rowData = {telefono.getCodigo(), telefono.getTipo(),
telefono.getNumero(), telefono.getOperadora()};
            modelo.addRow(rowData);
            tblTelefonosU.setModel(modelo);
        }
    }
    // Variables declaration - do not modify
    private javax.swing.JButton btnBuscar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable tblTelefonosU;
    private javax.swing.JTextField txtCedula;
    // 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		

Listar Telefonos

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

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.modelo.Telefono;
import javax.swing.table.DefaultTableModel;


/**
 *
 * @author paul_
 */
public class Listartelefonos extends javax.swing.JFrame {

    private ControladorTelefono controladorTelefono;

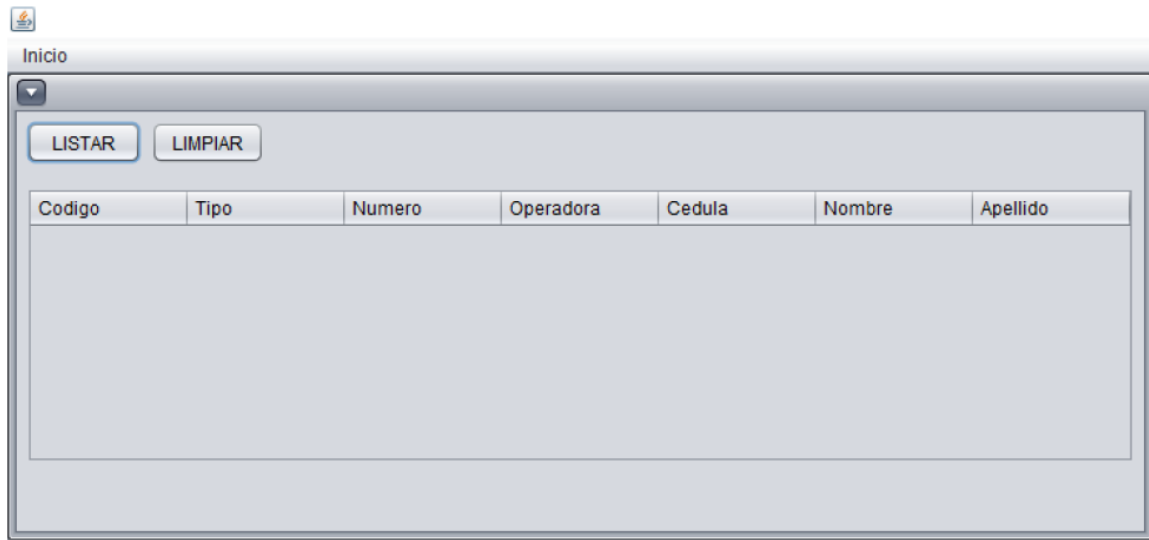
    public Listartelefonos(ControladorTelefono controladorTelefono) {
        this.controladorTelefono = controladorTelefono;

        initComponents();
    }
    private void btnListarActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonosUse.getModel();
        modelo.setRowCount(0);
        for (Telefono t : controladorTelefono.getLista()) {
            Telefono telefono = (Telefono) t;
            Object[] rowData = {telefono.getCodigo(), telefono.getTipo(),
telefono.getNumero()};
            modelo.addRow(rowData);
            tblTelefonosUse.setModel(modelo);
        }
    }
    private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {
        DefaultTableModel modelo = (DefaultTableModel) tblTelefonosUse.getModel();
        modelo.setRowCount(0);
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnLimpiar;
    private javax.swing.JButton btnListar;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable tblTelefonosUse;
    // 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		

[Interfaz gráfica de la ventana]



Registrar Usuario

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

import ec.edu.ups.controlador.ControladorUsuario;
import ec.edu.ups.modelo.Usuario;
import javax.swing.JOptionPane;

/**
 *
 * @author paul_
 */
public class RegistrarUsuario extends javax.swing.JInternalFrame {

    private ControladorUsuario controlador;

    /**
     * Creates new form GestionUsuario
     */
    public RegistrarUsuario(ControladorUsuario controlador) {
        initComponents();
        this.controlador = controlador;
    }

    private void txtGcedulaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }

    private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String pass = "";
        char[] pass1 = jGpr.getPassword();
        for (int i = 0; i < pass1.length; i++) {
```


```
        pass = pass + pass1[i];
    }
    Usuario usuario = new Usuario(txtGcedula.getText(), txtGnombre.getText(),
txtGapellido.getText(), txtGcorreo.getText(), pass);
    if (controlador.validar(usuario)) {
        controlador.crear(usuario);
        JOptionPane.showMessageDialog(this, "Usuario creado correctamente");
        Limpiar();
        this.dispose();

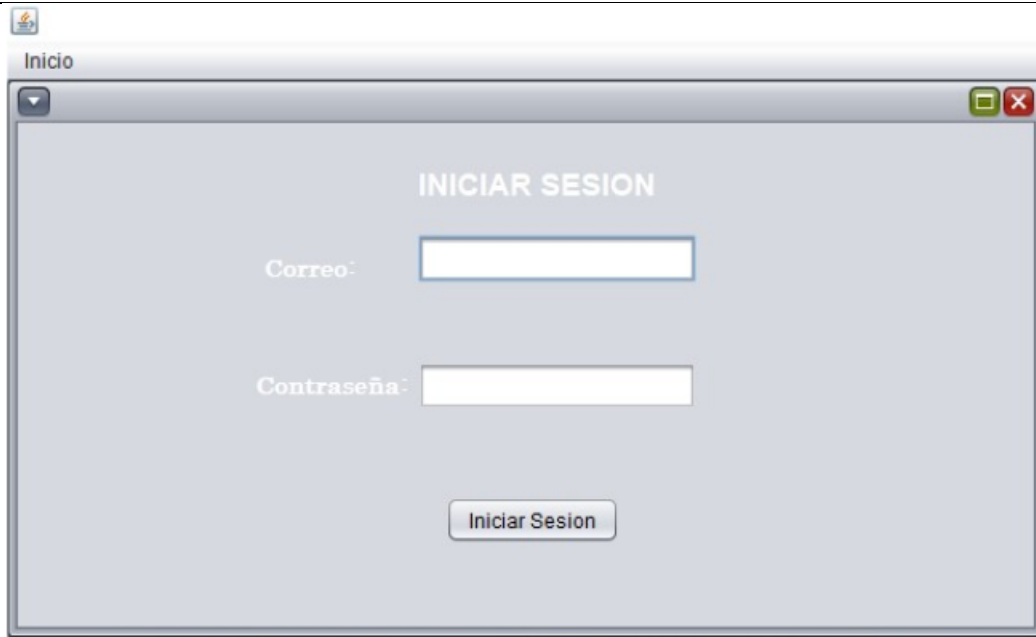
    } else {
        JOptionPane.showMessageDialog(this, "La cedula ingresada no existe");
        Limpiar();
    }
    ;

}

public void Limpiar() {
    txtGcedula.setText("");
    txtGnombre.setText("");
    txtGapellido.setText("");
    txtGcorreo.setText("");
    jGpr.setText("");

}
// Variables declaration - do not modify
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnGuardar;
private javax.swing.JPasswordField jGpr;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtGapellido;
private javax.swing.JTextField txtGcedula;
private javax.swing.JTextField txtGcorreo;
private javax.swing.JTextField txtGnombre;
// 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		



Ventana Iniciar Sesión


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

import ec.edu.ups.controlador.ControladorUsuario;
import ec.edu.ups.modelo.Usuario;
import javax.swing.JOptionPane;

/**
 *
 * @author paul_
 */
public class VentanaIniciarSesion extends javax.swing.JInternalFrame {

    private VentanaPrincipal ventanaPrincipal;
    private ControladorUsuario controladorUsuario;

    /**
     * Creates new form VentanaIniciarSesion
     */
    public VentanaIniciarSesion(ControladorUsuario controladorUsuario,
VentanaPrincipal ventanaPrincipal) {
        initComponents();
        this.ventanaPrincipal = ventanaPrincipal;
        this.controladorUsuario = controladorUsuario;
    }
    private void btnIniciarSesionActionPerformed(java.awt.event.ActionEvent evt) {
        String usuario = txtCorreo.getText();
        String pass = "";
        char[] pass1 = txtPass.getPassword();
        for (int i = 0; i < pass1.length; i++) {
            pass = pass + pass1[i];
        }
    }
}
```

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

```

if (controladorUsuario.iniciarSesion(usuario, pass)) {
    ventanaPrincipal.getIniciarSesionMenuItem().setVisible(false);
    ventanaPrincipal.getCerrarSesionMenu().setVisible(true);
    ventanaPrincipal.getGestionTelefonoMenu().setVisible(true);

    ventanaPrincipal.getSalirMenuItem().setVisible(false);
    ventanaPrincipal.getRegistrarUsuarioMenuItem().setVisible(false);
    ventanaPrincipal.getGestionMenu().setVisible(true);

    this.dispose();
    JOptionPane.showMessageDialog(this, "Inicio de sesion exitoso");
} else {


    JOptionPane.showMessageDialog(this, "Usuario o contrasena incorrecta ");
    Limpiar();
}

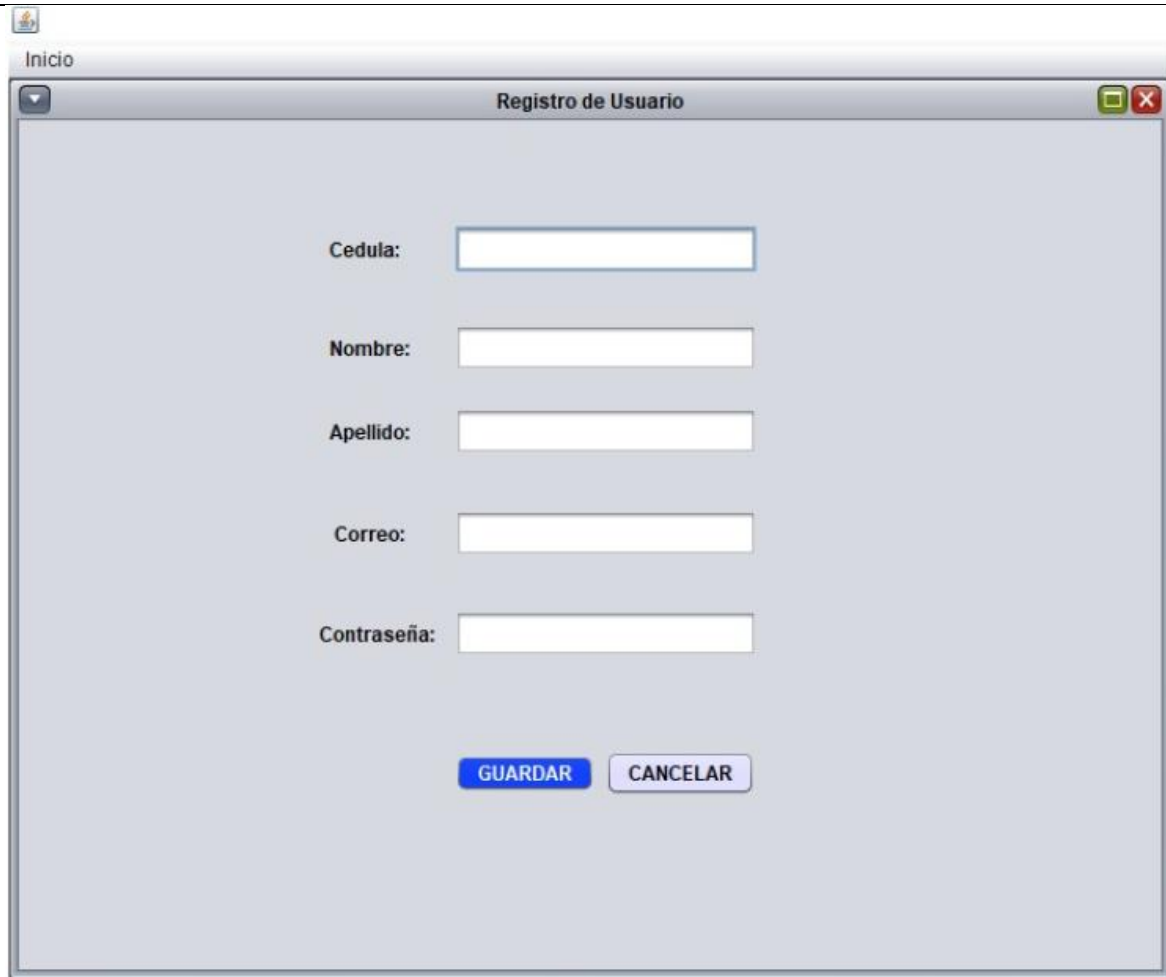
}

public void Limpiar() {
    txtCorreo.setText("");
    txtPass.setText("");
}

// Variables declaration - do not modify
private javax.swing.JButton btnIniciarSesion;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel lblContra;
private javax.swing.JLabel lblCorreo;
private javax.swing.JTextField txtCorreo;
private javax.swing.JPasswordField txtPass;
// 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		




Ventana Principal

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

import ec.edu.ups.controlador.ControladorTelefono;
import ec.edu.ups.controlador.ControladorUsuario;
import javax.swing.JMenu;
import javax.swing.JMenuItem;

/**
 *
 * @author paul_
 */
public class VentanaPrincipal extends javax.swing.JFrame {

    private VentanaIniciarSesion ventanaIniciarSesion;
    private RegistrarUsuario registrarUsuario;
    private GestionTelefono gestionTelefono;
    private ListarTelefonosUsuarios listarTlfUsuarios;
    private ListarTelefonos listarTelefonos;
    private ControladorUsuario controladorUsuario;
    private ControladorTelefono controladorTelefono;
```


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

```

/**
 * Creates new form VentanaPrincipal
 */
public VentanaPrincipal() {
    initComponents();
    gestionMenu.setVisible(false);
    CerrarSesionMenu.setVisible(false);
    controladorUsuario = new ControladorUsuario();
    controladorTelefono = new ControladorTelefono();

    ventanaIniciarSesion = new VentanaIniciarSesion(controladorUsuario, this);
    registrarUsuario = new RegistrarUsuario(controladorUsuario);

    gestionTelefono = new GestionTelefono(controladorTelefono,
controladorUsuario);
    listarTlfUsuarios = new ListarTelefonosUsuarios(controladorTelefono,
controladorUsuario);
    listarTelefonos = new Listartelefonos(controladorTelefono);

}

public JMenuItem getCerrarSesionMenu() {
    return CerrarSesionMenu;
}

public JMenuItem getIniciarSesionMenuItem() {
    return IniciarSesionMenuItem;
}

public JMenu getInicioMenu() {
    return InicioMenu;
}


public JMenuItem getRegistrarUsuarioMenuItem() {
    return RegistrarUsuarioMenuItem;
}

public JMenuItem getSalirMenuItem() {
    return SalirMenuItem;
}

public JMenu getGestionMenu() {
    return gestionMenu;
}

public JMenuItem getGestionTelefonoMenu() {
    return gestionTelefonoMenu;
}
private void SalirMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

```

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

```

private void RegistrarUsuarioMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
    desktopPane.add(registrarUsuario);
    registrarUsuario.setVisible(true);
    IniciarSesionMenuItem.setVisible(true);
    SalirMenuItem.setVisible(true);
    gestionMenu.setVisible(false);
    CerrarSesionMenu.setVisible(false);
}

private void IniciarSesionMenuItemActionPerformed(java.awt.event.ActionEvent
evt) {
    desktopPane.add(ventanaIniciarSesion);
    ventanaIniciarSesion.setVisible(true);
}


private void CerrarSesionMenuActionPerformed(java.awt.event.ActionEvent evt) {
    IniciarSesionMenuItem.setVisible(true);
    RegistrarUsuarioMenuItem.setVisible(true);
    gestionMenu.setVisible(false);
    SalirMenuItem.setVisible(true);
}

private void gestionTelefonoMenuActionPerformed(java.awt.event.ActionEvent evt)
{
    desktopPane.add(gestionTelefono);
    gestionTelefono.setVisible(true);
}

private void ListaTUsuariosActionPerformed(java.awt.event.ActionEvent evt) {
    desktopPane.add(listarTlfUsuarios);
    listarTlfUsuarios.setVisible(true);
    CerrarSesionMenu.setVisible(false);
}

private void ListarTelefonosMenuActionPerformed(java.awt.event.ActionEvent evt)
{
    desktopPane.add(listarTelefonos);
    listarTelefonos.setVisible(true);
    CerrarSesionMenu.setVisible(false);
}
// Variables declaration - do not modify
private javax.swing.JMenuItem CerrarSesionMenu;
private javax.swing.JMenuItem IniciarSesionMenuItem;
private javax.swing.JMenu InicioMenu;
private javax.swing.JMenuItem ListaTUsuarios;
private javax.swing.JMenuItem ListarTelefonosMenu;
private javax.swing.JMenuItem RegistrarUsuarioMenuItem;
private javax.swing.JMenuItem SalirMenuItem;
private javax.swing.JDesktopPane desktopPane;
private javax.swing.JMenu gestionMenu;
private javax.swing.JMenuItem gestionTelefonoMenu;
private javax.swing.JMenuBar menuBar;

```

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

```
// End of variables declaration
```

```
}
```



RESULTADO(S) OBTENIDO(S):

- Utilizar la ayuda de controladores y las herramientas de NetBeans que permiten la gestión de archivos dentro de la IDE que luego serán asignados a una ruta especificado por el usuario dentro del ordenador.
- Desarrollo de una aplicación que permita gestionar los teléfonos de un usuario que puede registrar e iniciar sesión dentro de la aplicación muy parecida a la práctica de Gestión de Teléfonos pero que ahora incluye una interfaz muy aceptable para el usuario que desee usarlo, además es muy fácil de manejar.

CONCLUSIONES:

- Una manera más eficaz de manejar este tipo de aplicaciones dentro de NetBeans es con el uso de un controlador dentro del proyecto en sí, porque dentro de este se crearían los métodos que luego van a ser instanciados a las ventanas donde serán usados y a la vez requeridos para su ayuda dentro del código.
- La aplicación de patrones de diseño y arquitecturas dentro de la aplicación creada son fundamentales y facilitan el trabajo a la vez que simplifican el código para que no venga a ser muy pesado y tenga una mayor aceptabilidad dentro de los ámbitos en los que se deseen proyectar su respectivo uso.

RECOMENDACIONES:

- Utilizar la ID NetBeans para este tipo de aplicaciones es muy recomendable, puesto que su variedad de diccionarios, métodos, patrones, nos ahorran mucho tiempo en cuanto a el desarrollo de la interfaz del mismo, o más bien el entorno que será visualizado por el usuario.
- Para que una vista de interfaz sea agradable a un lector se debe realizar un estudio de los colores que agraden a los usuarios para que de esta manera se pueda tener una respuesta factible cuando el o los usuarios utilicen la aplicación y den su visto bueno.

Nombre de estudiante:

PAUL ALEXANDER GUAPUCAL CARDENAS

Firma de estudiante:

