
	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 caracteristicas 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: 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			
ACTIVIDADES DESARROLLADAS			
1. Revisar la teoría y conceptos de Java 8, 9 ,10, 11, 12			
<pre> public void actualizarTelefono(Telefono telefono){ var telefonoCambiar = listaTelefonos.stream().filter(t -> t.getCodigo() == telefono.getCodigo()).findFirst().get(); int index = listaTelefonos.indexOf(telefonoCambiar); listaTelefonos.remove(telefonoCambiar); listaTelefonos.add(index, telefono); } public void eliminarTelefono(int codigo){ var telefonoEliminar = listaTelefonos.stream().filter(telefono -> telefono.getCodigo() == codigo).findFirst().get(); listaTelefonos.remove(telefonoEliminar); } </pre> <p>Parte del código de la clase Persona del paquete modelo en donde se utiliza la palabra var para declarar alguna variable como podemos observar en el código es una de las nuevas características de java.</p> <p>A su vez también podemos observar en el mismo código el uso de Streams, Lambdas con la funcionalidad de buscar un teléfono requerido por lo que utilizamos un Stream().filter().</p> <p>Estas Dos nuevas operaciones de codificación de Java no son las únicas en todo el código, pero si son un pequeño ejemplo de su uso.</p> <pre> public Persona iniciarSesion(String correo, String contraseña) { List<Persona> copiaListaPersona; copiaListaPersona = (List<Persona>) List.copyOf(listaGenerica); return copiaListaPersona.stream().filter(pe -> pe.getCorreo().equalsIgnoreCase(correo) && pe.getContraseña().equalsIgnoreCase(contraseña)).findFirst().get(); } </pre> <p>Parte del código extraído de la clase controlador, ubicada en el paquete controlador podemos observar que para el método de iniciar sesión se utiliza uno de los nuevos métodos de java que</p>			

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

es el copyOf el cual nos permite hacer una copia no modificable de una lista es por ello que con la ayuda de ese método podemos recorrer la lista de personas buscando la correcta.

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.

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

import ec.edu.ups.modelo.Persona;
import java.util.*;

/**
 *
 * @author NANCY
 * @param <T>
 */
public abstract class Controlador<T> {

    private List<T> listaGenerica;

    public Controlador() {
        listaGenerica = new ArrayList<>();
    }

    public boolean create(T objeto) {
        if(validar(objeto)==true){
            return listaGenerica.add(objeto);
        }

        return false;
    }

    public T read(T comp){
        try{
            /*for (T t : listaGenerica) {
                if(comp.equals(t)){
                    return t;
                }
            }*/
            return listaGenerica.stream().filter(objeto->objeto.equals(comp)).findFirst().get();
        }catch(Exception ex){}
        return null;
    }

    public void update(T obj, T Eliminar) {
        int index = (listaGenerica.indexOf(Eliminar));
        listaGenerica.remove(index);
        listaGenerica.add(index, obj);
    }
}
```

```
}

public void delete(T objeto) {
    listaGenerica.remove(objeto);
    System.out.println(listaGenerica);
}

public boolean eliminar(T objeto){
    T objetoEliminar= read(objeto);
    return(objetoEliminar!=null)? listaGenerica.remove(objetoEliminar): false;
}

public List<T> findAll() {
    return listaGenerica;
}

public abstract boolean validar(T obj);

public int cargarCodigo(){
    if (findAll().size() > 0) {
        return findAll().size() + 1;
    } else {
        return 1;
    }
}

}
```

En el código de la clase controlador observamos como están realizados los métodos del CRUD los cuales ocupamos en todo el programa para poder crear, buscar, actualizar y eliminar datos de las listas creadas.

Cave recalcar que estos métodos están funcionando con una programación genérica por lo cual existe un solo controlador para crear las listas los dos objetos que en este caso son las personas y los teléfonos, también podemos observar algunos métodos de las nuevas versiones de Java.

Para después este controlador principal que es una clase abstracta instanciarlo en los controladores de persona y teléfono con el método abstracto de validar. También podemos observar que tenemos un método que genera códigos automáticamente tanto para la persona como para el teléfono denominado cargarCodigo.

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


```
public abstract boolean validar(T obj);
```

Código extraído del controlador donde inicia el método validar debido a que es un método abstracto

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

```
import ec.edu.ups.modelo.Persona;
```

```
import java.util.List;
```

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

```
/**
 *
 * @author NANCY
 */
public class ControladorPersona extends Controlador<Persona> {
```

```
    @Override
    public boolean validar(Persona obj) {
        if(obj.getCedula().length()== 10){
            return true;
        }
        return false;
    }
}
```

Modificación del código validar en la persona para validar a través de la cedula verificando si tiene 10 dígitos para así poder crear el usuario en caso de cumplirlo y manejar los errores.

```
public T read(T comp){
    try{
        /*for (T t : listaGenerica) {
            if(comp.equals(t)){
                return t;
            }
        }*/
        return listaGenerica.stream().filter(objeto-
>objeto.equals(comp)).findFirst().get();
    }catch(Exception ex){}
    return null;
}
```


Fragmento del código del controlador donde vemos como esta implementado un try y un catch para manejar los errores en caso de que se solicite buscar un objeto y no sea encontrado.

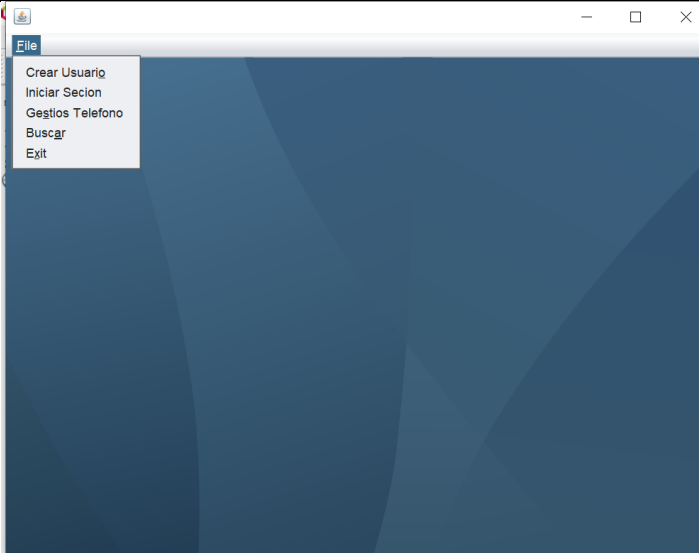
4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica

Link del repositorio en Git Hub: <https://github.com/RomelAvila2001/Practica-numero-2-Programacion-Aplicada/tree/master>

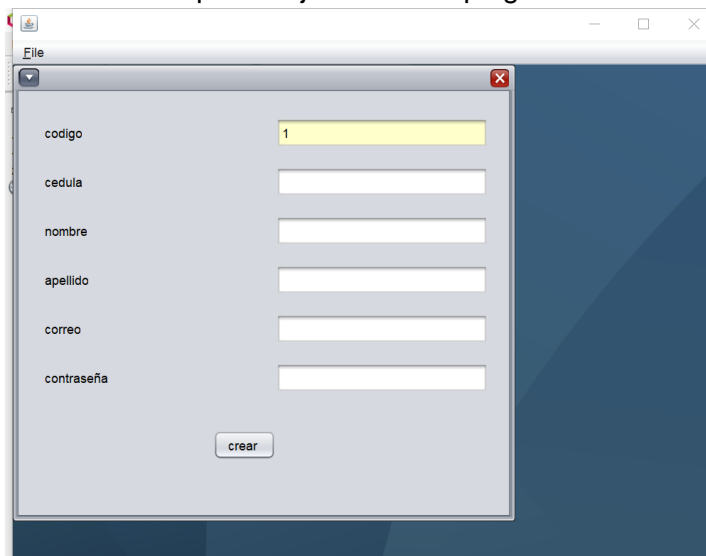
Aplicación Funcionando

Las capturas demuestran el correcto funcionamiento de la aplicación


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

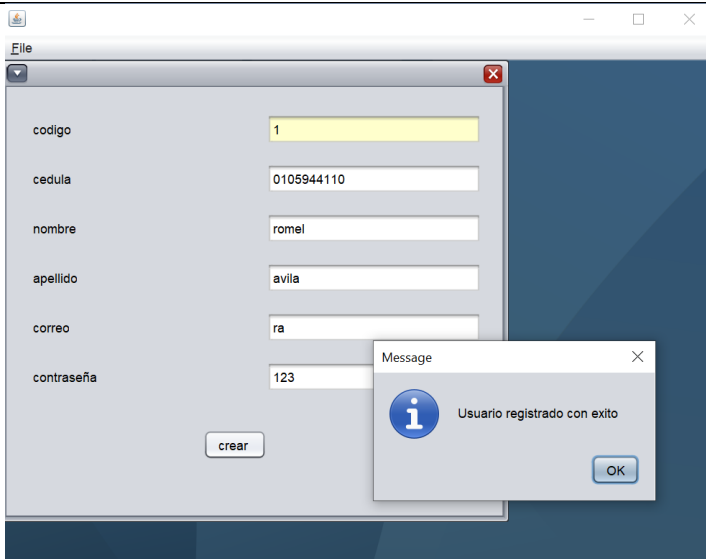


Ventana Principal en ejecución desplegando menú

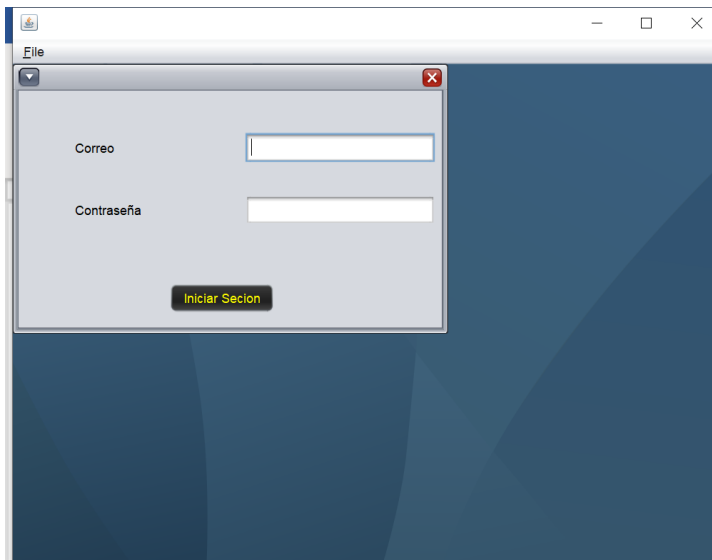


Ventana para crear el usuario


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

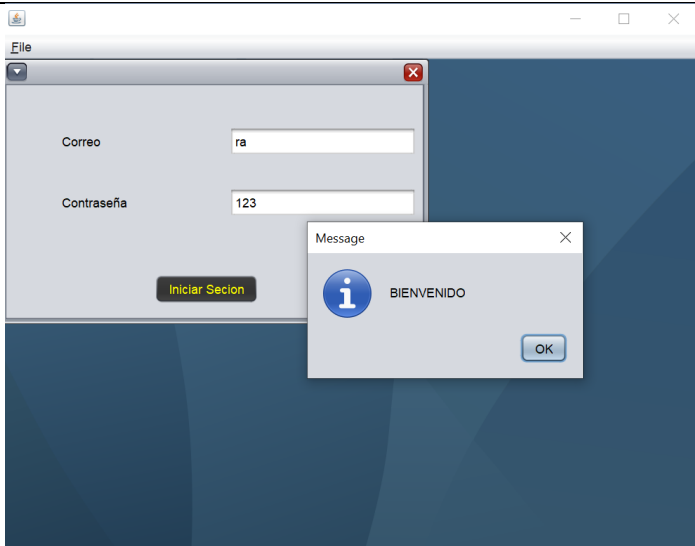


Creación de usuario

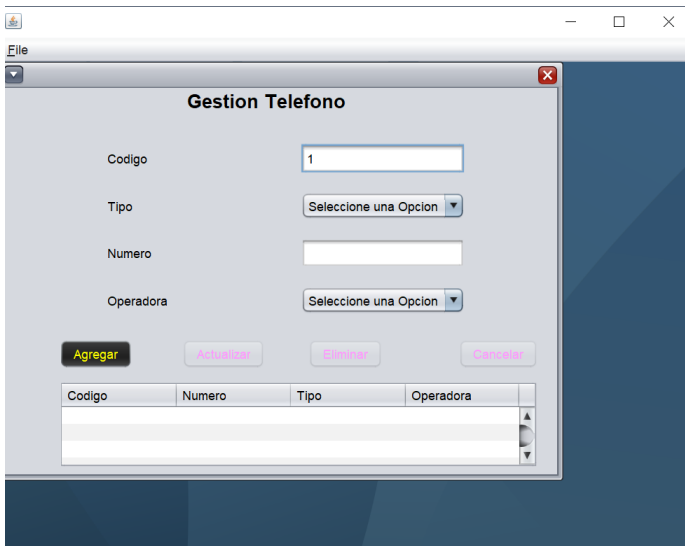


Ventana Iniciar Sesión

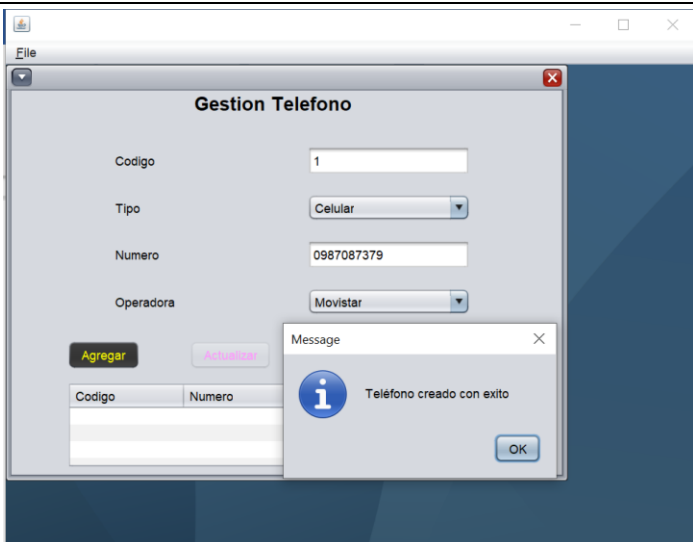
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



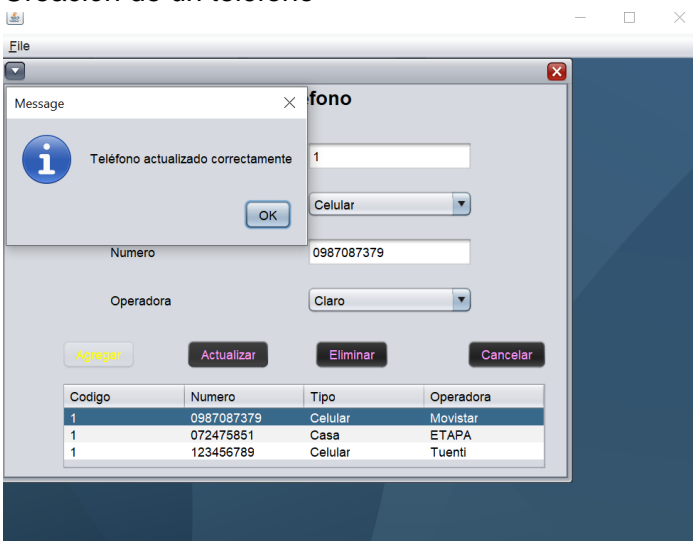
Sesión iniciada correctamente



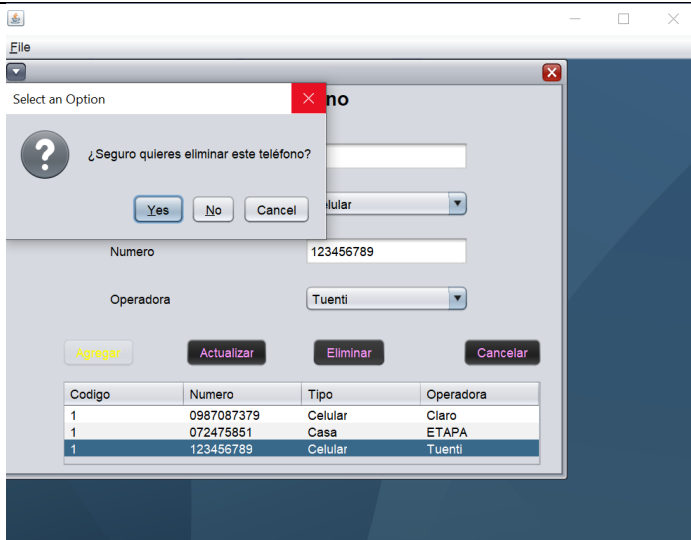
Ventana Gestión Telefono



Creación de un teléfono



Actualización de un Teléfono



Select an Option

¿Seguro quieres eliminar este teléfono?

Yes No Cancel

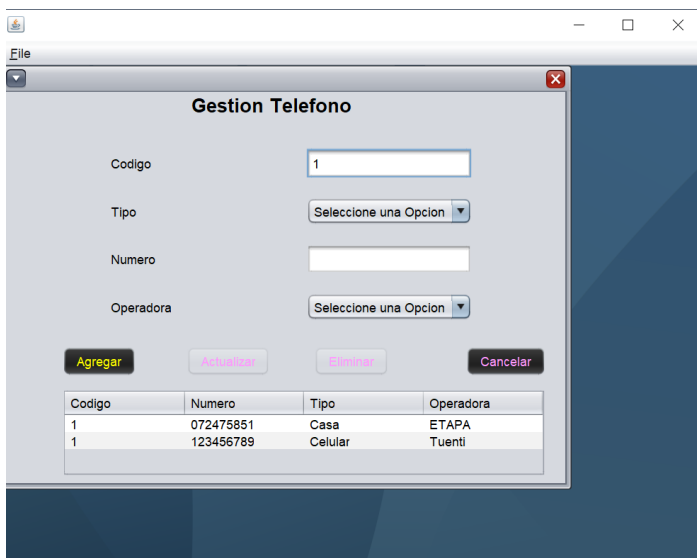
Numero 123456789

Operadora Tuenti

Agregar Actualizar Eliminar Cancelar

Codigo	Numero	Tipo	Operadora
1	0987087379	Celular	Claro
1	072475851	Casa	ETAPA
1	123456789	Celular	Tuenti

Eliminación de un teléfono



Gestion Telefono

Codigo 1

Tipo Seleccione una Opcion


Numero

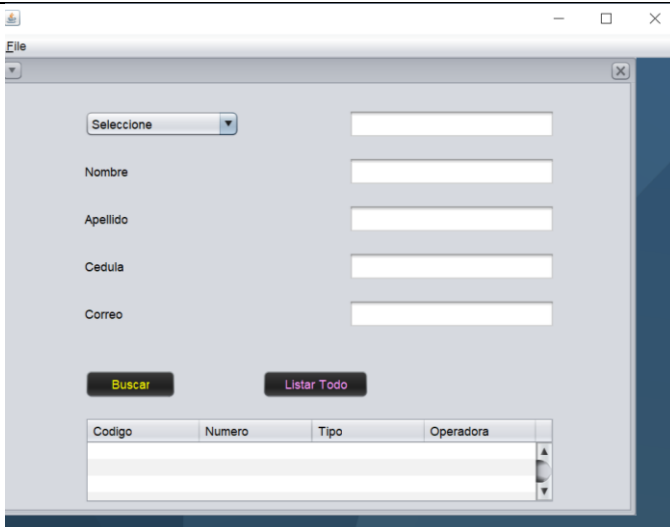
Operadora Seleccione una Opcion

Agregar Actualizar Eliminar Cancelar

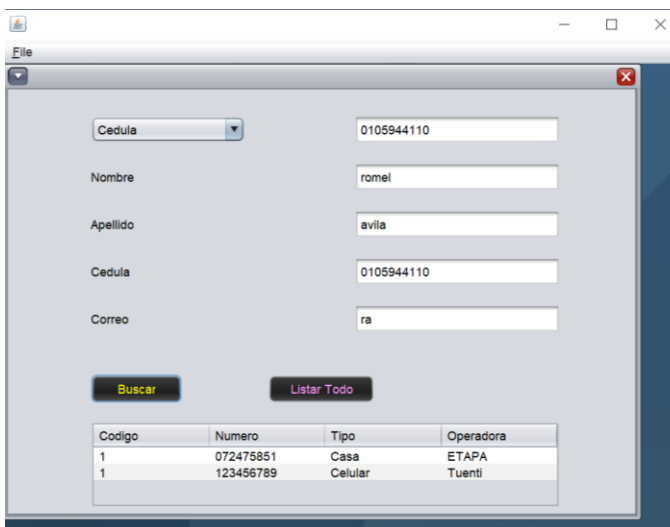
Codigo	Numero	Tipo	Operadora
1	072475851	Casa	ETAPA
1	123456789	Celular	Tuenti

Comprobación de la eliminación del teléfono

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




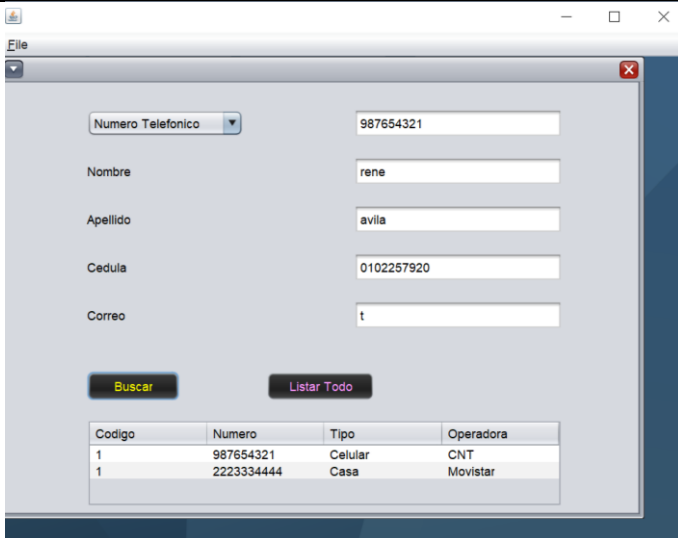
Ventana Buscar



Codigo	Numero	Tipo	Operadora
1	072475851	Casa	ETAPA
1	123456789	Celular	Tuenti

Buscando por Cedula

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



Numero Telefonico: 987654321

Nombre: rene

Apellido: avila

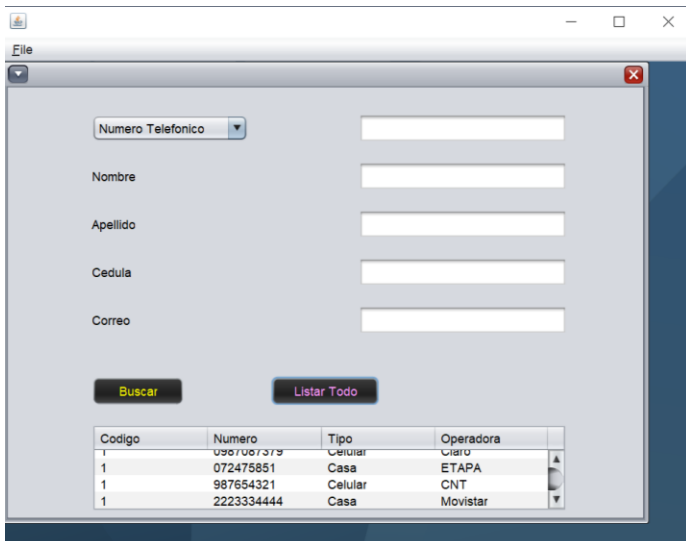
Cedula: 0102257920

Correo: t

Buscar Listar Todo

Codigo	Numero	Tipo	Operadora
1	987654321	Celular	CNT
1	2223334444	Casa	Movistar

Buscar por numero de teléfono para demostrar esto se creo otro usuario con otros numero de teléfono



Numero Telefonico:

Nombre:

Apellido:

Cedula:

Correo:

Buscar Listar Todo

Codigo	Numero	Tipo	Operadora
1	072475851	Casa	ETAPA
1	987654321	Celular	CNT
1	2223334444	Casa	Movistar


Listar todo aparecen todos los teléfonos de todos los usuarios

RESULTADO(S) OBTENIDO(S):

Se puede comprimir el código de la aplicación utilizando una programación genérica y los nuevos métodos que traen las nuevas versiones de java.

CONCLUSIONES:

En conclusión, al aplicar los nuevos métodos de las nuevas versiones de java profundizamos más nuestra técnica de programación

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

RECOMENDACIONES:

Profundizar un poco más sobre la programación genérica

Estudiantes: Romel Ávila

Firma:

