

	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		

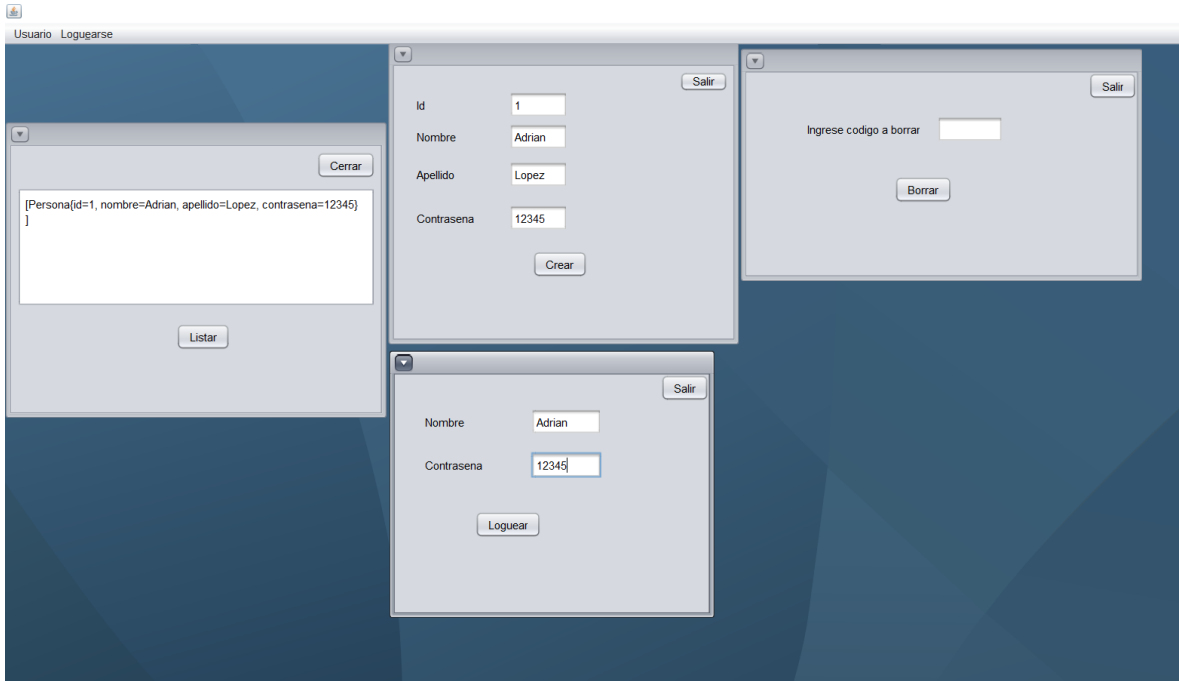
		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	3	TÍTULO PRÁCTICA: Reflexión en Java	
OBJETIVO ALCANZADO: Identificar los cambios importantes de Java Diseñar e Implementar las nuevas técnicas 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, 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.			
3. Probar y modificar el método validar para que nos permita utilizar excepciones, además de modificar el buscar para controlar el nullpointerexception.			
4. Realizar práctica codificando los códigos de las nuevas características de Java y su uso dentro de una agenda telefónica.			
5.			
6.			
N.			
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:			

Nombre de estudiante: Adrian Lopez

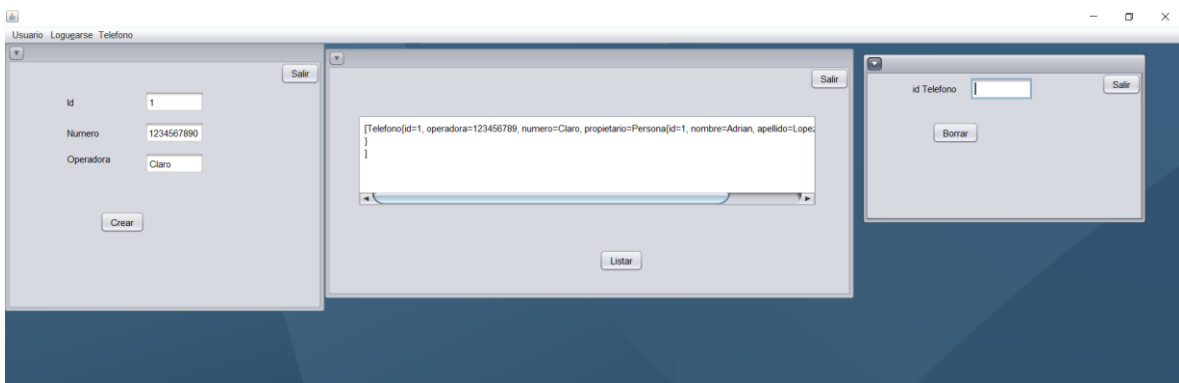
Firma de estudiante: _____


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

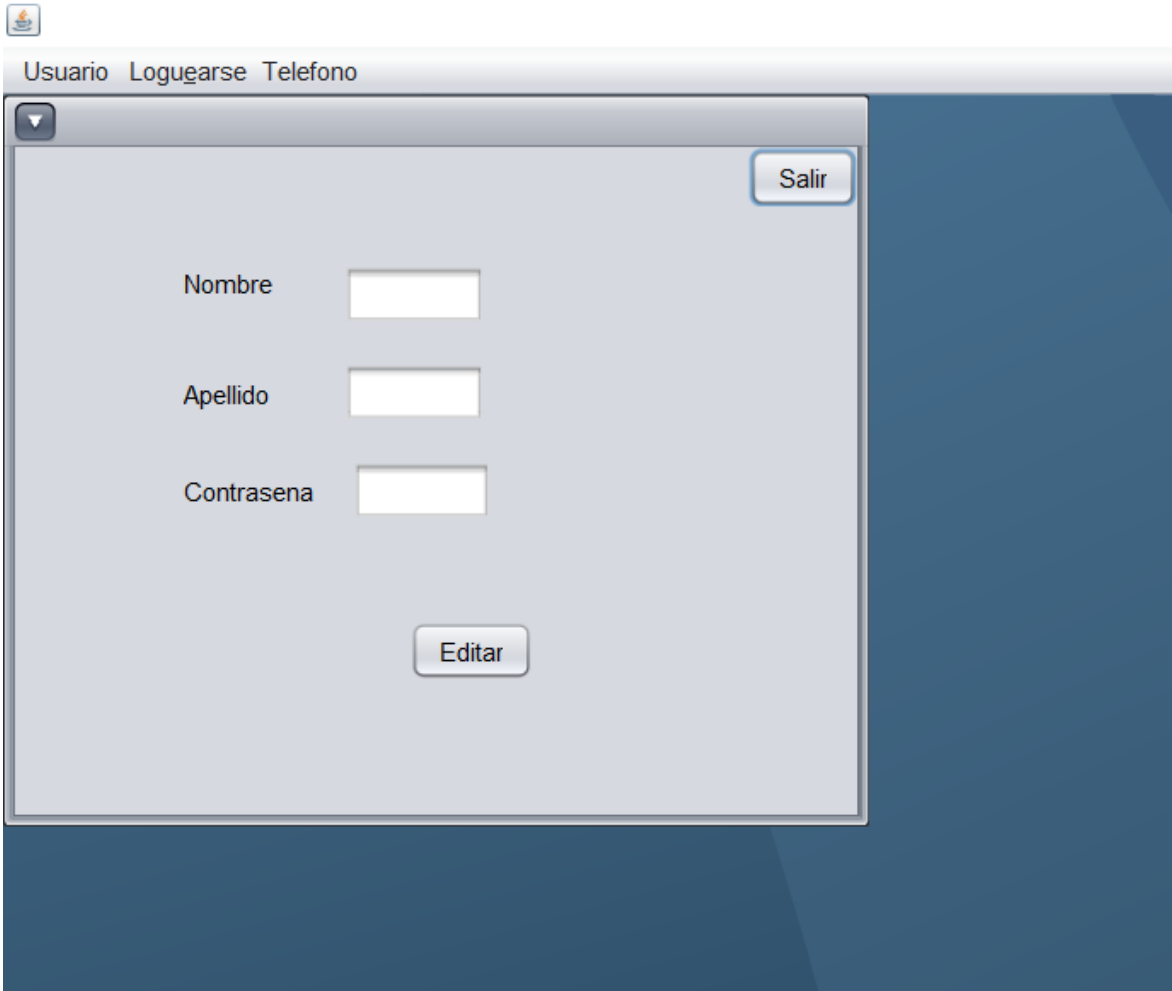
Se realizó una interfaz gráfica donde de principio deja crear borrar y listar Usuarios o loguear un usuario.



Después de loguearse con el nombre y contraseña dados por el usuario se va a habilitar el crear, editar, borrar y listar del teléfono; se agregará un teléfono al usuario y se listarán solo los teléfonos del mismo, además de habilitarse el editar del usuario.



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



Usuario Loguearse Telefono

Salir


Nombre

Apellido

Contraseña

Editar

Esta practica se hizo con clase genérica y se maneja un validar de usuario para que su contraseña sea mayor a 5 caracteres y el teléfono tenga solo 10 números

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

GuiaTelefonica - Apache NetBeans IDE 11.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects

- GuiaTelefonica
 - Source Packages
 - ups.edu.ec.controlador
 - AbstractControler.java
 - ControladorPersona.java
 - ControladorTelefono.java
 - ups.edu.ec.generica
 - ClassGenerica.java
 - ups.edu.ec.modelo
 - Persona.java
 - Telefono.java
 - ups.edu.ec.vista
 - PantallaActualizarTelefono.java
 - PantallaBorrarTelefono.java
 - PantallaBorrarUsuario.java
 - PantallaCrearTelefono.java
 - PantallaCrearUsuario.java
 - PantallaEditarTelefono.java
 - PantallaEditarUsuario.java
 - PantallaLoginUsuario.java
 - PantallaPrincipal.java
 - Test Packages
 - Dependencies
 - Java Dependencies
 - Project Files

AbstractControler.java

```

1 package ups.edu.ec.controlador;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public abstract class AbstractControler<E> {
7
8     public List<E> lista;
9     public AbstractControler(){
10         lista=new ArrayList();
11     }
12     public boolean crear(E objeto)
13     {
14         return lista.add(objeto);
15     }
16     public E buscar(E comparacion){
17         return lista.stream().filter(objeto->objeto.equals(comparacion)).findFirst().get();
18     }
19     public boolean eliminar(E objeto){
20         E objetoEliminar=buscar(objeto);
21         return (objetoEliminar!=null)? lista.remove(objetoEliminar):false;
22     }
23     public int BuscarPosicion(E comparacion){
24
25         for (int i = 0; i < lista.size(); i++) {
26             E objeto=lista.get(i);
27
28             if(objeto.equals(comparacion)){
29                 return i;
30             }
31         }
32         return -1;
33     }
34
35     public boolean actualizar(E objeto){
36

```