
	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 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:	1	TÍTULO PRÁCTICA: Clase Genéricas en Java	
OBJETIVO ALCANZADO: 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			
<ol style="list-style-type: none"> 1. Dentro de la práctica se realizó la arquitectura MVC la cual nos ayuda a separar en tres componentes los datos, la metodología y la interfaz gráfica de una aplicación, en este caso una aplicación que gestiona una agenda telefónica 2. En cuanto a los métodos CRUD se tomaron en cuenta conceptos de programación genérica en base a streams y lambdas que son características de la actualización del JDK 8 3. En cuanto a la codificación es la siguiente, comenzamos con la clase controlador <pre> package ec.edu.ups.controlador; import ec.edu.ups.modelo.Telefono; import ec.edu.ups.modelo.Usuario; import java.util.ArrayList; import java.util.Collection; import java.util.Iterator; import java.util.List; public class Controlador <E> { private List<E> listado; public Controlador() { listado= new ArrayList(); } public boolean crear (E obj){ return listado.add(obj); } public E buscar (E ob){ return this.listado.stream().filter(obj -> ob.equals(obj)).findFirst().get() ; } public Telefono buscarTelf (int id){ List <Telefono> listaT = (List <Telefono>) List.copyOf(listado); </pre>			

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

```

return (Telefono) listaT.stream().filter(Telefono ->
Telefono.getCodigo()==id).findFirst().get();

}
public Telefono buscarTelfUsu (String cedula){
    List <Telefono> listaT = (List <Telefono>) List.copyOf(listado);

    return (Telefono) listaT.stream().filter(Telefono ->
Telefono.getUsuario().getCedula().equals(cedula)).findFirst().get();

}

public Usuario buscarUsuarios (String cedula){

    List <Usuario> listaU = (List <Usuario>) List.copyOf(listado);
    return (Usuario) listaU.stream().filter(Usuario ->
Usuario.getCedula().equals(cedula)).findFirst().get();

}
public boolean eliminar(E obj) {
    return listado.remove(obj);
}

public void actualizar(E obj, E obj2) {

    int posicion = (listado.indexOf(obj2));
    listado.remove(posicion);

    listado.add(posicion, obj);

}


public Usuario iniciarSesion(String correo, String pass){
    List <Usuario> listaU = (List <Usuario>) List.copyOf(listado);
    return (Usuario) listaU.stream().filter(Usuario ->
Usuario.getCorreo().equals(correo)&&Usuario.getContrasena().equals(pass)).findFirst().get();
}

public List<E> findAll() {
    return listado;
}

public List<Telefono> telefonos (){
    List <Telefono> listaT= new ArrayList();
    Telefono telefono;
    Iterator i = listado.iterator();
    while(i.hasNext()){
        telefono = (Telefono)i.next();
        listaT.add(telefono);
    }

    return listaT;
}

```

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

```

}
public List<Usuario> usuarios() {
    List <Usuario> listaU= new ArrayList();
    Usuario usuario;
    Iterator i = listado.iterator();
    while(i.hasNext()){
        usuario = (Usuario)i.next();
        listaU.add(usuario);
    }

    return listaU;
}

public int generarId() {
    if(this.listado.size()>0){

        return (int) listado.size()-1;
    }

    return 1;
}
public List<E> getListado() {
    return listado;
}

public void setListado(List<E> listado) {
    this.listado = listado;
}
}

```

Dentro del paquete modelo se encuentran las clases Telefono y Usuario

```

package ec.edu.ups.modelo;

/**
 *
 * @author
 */
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;
}

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 = 5;
    hash = 37 * hash + this.codigo;
    return hash;
}
```

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

```

    }

    @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 "DATOS TELEFONO ** " + "codigo: " + codigo + " numero: " + numero + " tipo: "
+ tipo + " operadora: " + operadora + " **";
    }
}

```

La clase Usuario

```

package ec.edu.ups.modelo;

import java.util.Objects;


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

    /**
     */
    private int id;
    private String Cedula;
    private String nombre;
    private String apellido;
    private String correo;
    private String contrasena;

    public Usuario() {

    }
}

```


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

```

public Usuario(int id,String Cedula, String nombre, String apellido, String correo, String
contrasena) {
    this.setId(id);
    this.setCedula(Cedula);
    this.setNombre(nombre);
    this.setApellido(apellido);
    this.setCorreo(correo);
    this.setContrasena(contrasena);

}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

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() {
    return contrasena;
}

```

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

```

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

@Override
public int hashCode() {
    int hash = 5;
    hash = 89 * hash + this.id;
    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.id, other.id)) {
        return false;
    }
    return true;
}

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

```

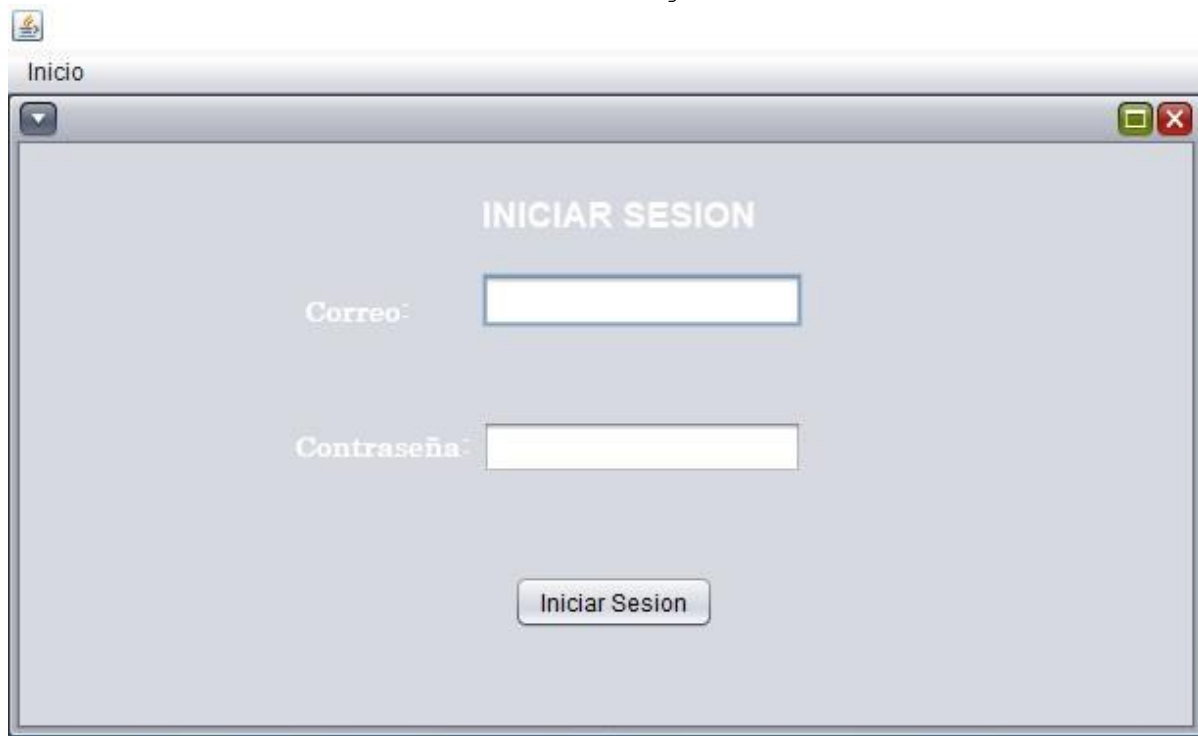
El diseño de las ventanas es el siguiente

Ventana Principal

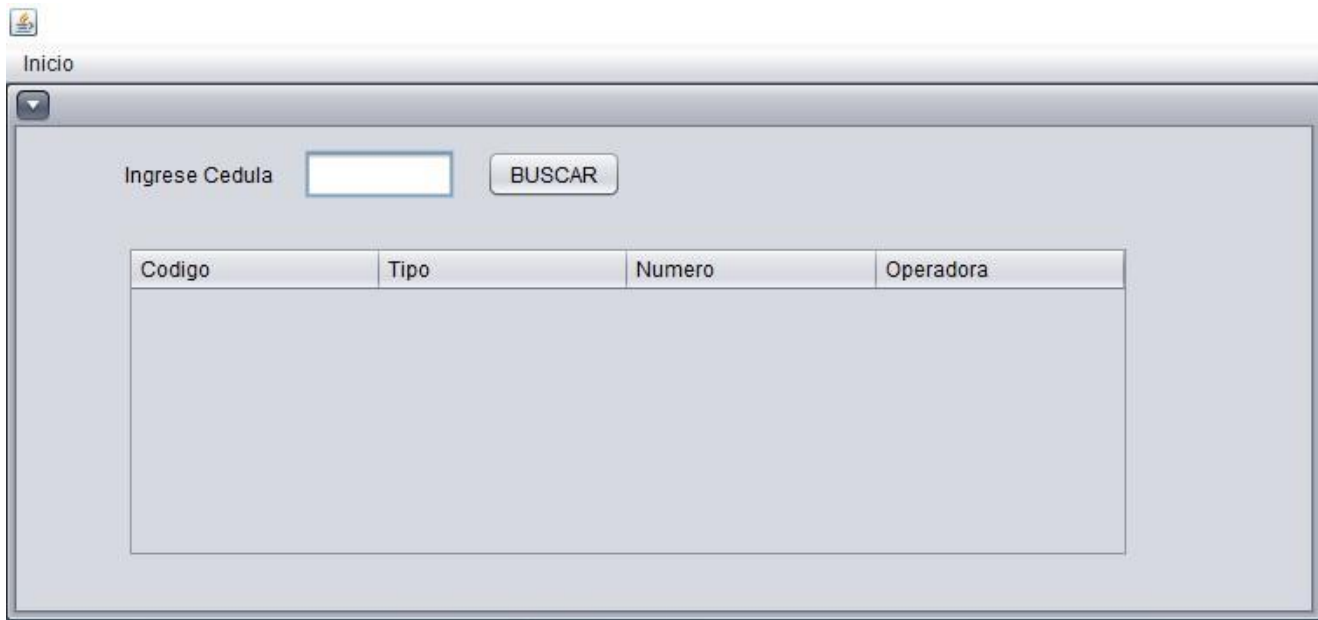
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Ventana Login



Ventana Registro de Usuario

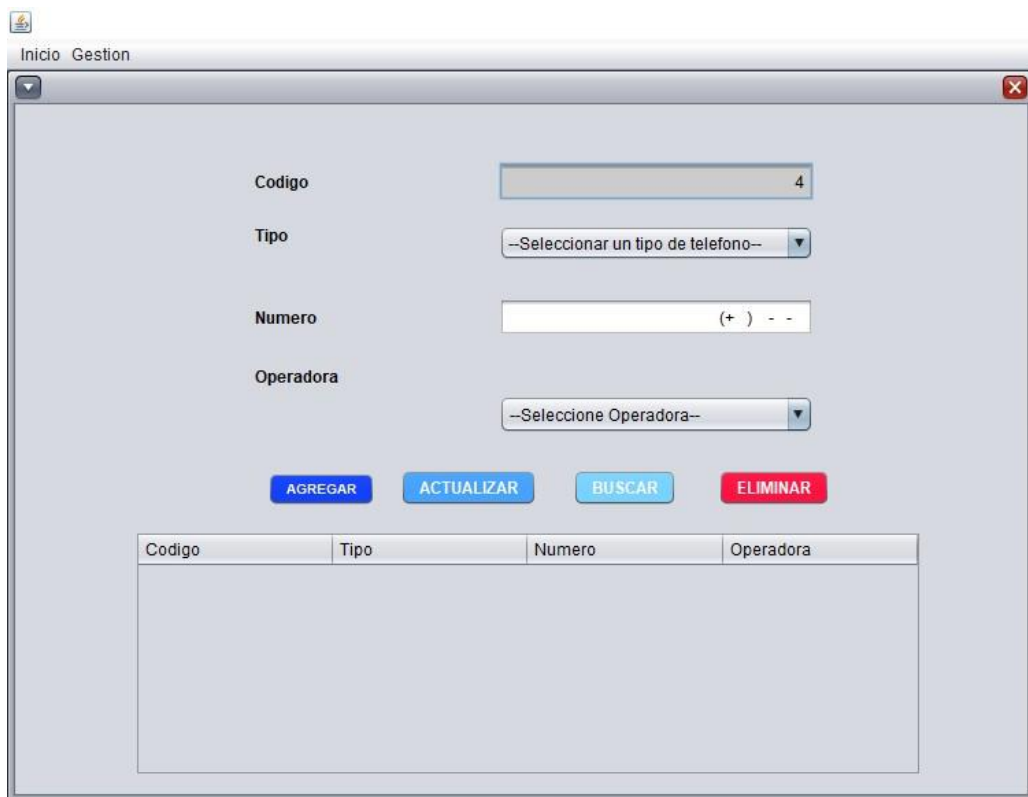


Inicio

Ingrese Cedula

Codigo	Tipo	Numero	Operadora
--------	------	--------	-----------

Ventana Gestión Telefono



Inicio Gestion


Codigo

Tipo

Numero

Operadora

Codigo	Tipo	Numero	Operadora
--------	------	--------	-----------

	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):

Dentro del ámbito de la programación genérica se pudo comprender acerca de los nuevos métodos que nos brinda las actualizaciones de java, en este caso los streams y lambdas tuvieron un papel muy importante en esta práctica, aparte de la arquitectura MVC que nos ayuda a organizar de una mejor manera nuestro proyecto

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.

Nombre de estudiante: JUAN JOSE CORDOVA CALLE

Firma de estudiante:

