
	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


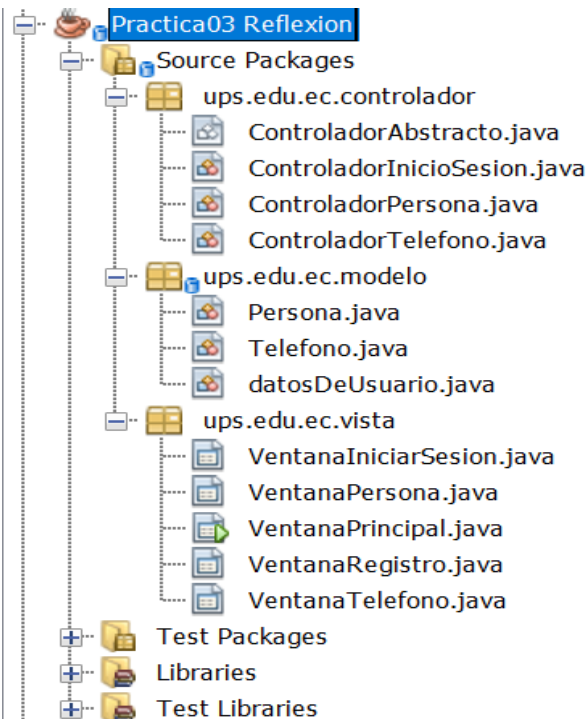
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		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:		ASIGNATURA:
NRO. PRÁCTICA:		TÍTULO PRÁCTICA:
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> - 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		
<p>1. Revisar la teoría y conceptos de Java 8, 9 ,10, 11, 12, 13, 14, 15</p> <p>Hemos trabajado e implementado varios métodos y además del patrón de arquitectura MVC el cual nos ayuda a tener un código mucho más organizado al igual este se separa en capas que son modelo, vista, controlador.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;">  </div>		
Clases genéricas: <p>Hemos implementado las clases genéricas que nos ayudan a reutilizar código, esto nos permite tener un código mucho más compacto.</p>		

```
/**
 *
 * @author santi
 */
public abstract class ControladorAbstracto<T> {
    . . . . .
}
```

Reflexión en java.

Hemos implementado la reflexión para imprimir cualquier lista que nosotros deseemos pero este método no es muy utilizado por su complejidad y su amplio código.

```
public static void imprimirCualquierLista(List listar) {
    for (Object obj : listar) {
        Method[] metodos = obj.getClass().getMethods();
        for (Method metodo : metodos) {
            if (metodo.getName().equals("getId")) {
                try {
                    int cadena = (Integer) metodo.invoke(obj, null);
                    System.out.println(cadena);
                } catch (IllegalAccessException | IllegalArgumentException
                    | InvocationTargetException ex) {
                    java.util.logging.Logger.getLogger(ControladorAbstracto.class.getName()).
                        log(java.util.logging.Level.SEVERE, null, ex);
                }
            }
        }
    }
}
```

Patrón de diseño singleton

Este patrón permite tener un solo objeto instanciado, al crear el controlador debe ser privado para que este no pueda ser instanciado en otra clase.

```
/**
 *
 * @author santi
 */
public class ControladorInicioSesion {

    private static ControladorInicioSesion instancia = new ControladorInicioSesion();

    private List lista = new ArrayList();

    private ControladorInicioSesion() {
    }
}
```

Se llamaría a otra clase de esta manera.

```
controladorInicioSesion = ControladorInicioSesion.getInstancia();
```

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.

- Impresión de cualquier lista

Se ha implementado reflexión para generar la impresión de la lista

```
public static void imprimirCualquierLista(List listar) {

    for (Object obj : listar) {

        Method[] metodos = obj.getClass().getMethods();
        for (Method metodo : metodos) {

            if (metodo.getName().equals("getId")) {

                try {
                    int cadena = (Integer) metodo.invoke(obj, null);
                    System.out.println(cadena);
                } catch (IllegalAccessException | IllegalArgumentException
                        | InvocationTargetException ex) {
                    java.util.logging.Logger.getLogger(ControladorAbstracto.class.getName()).
                        log(java.util.logging.Level.SEVERE, null, ex);
                }
            }
        }
    }
}
```

Una vez hayamos creado este método lo vamos a implementar en la interfaz donde hemos llamado a la lista para que se genere en una tabla de datos.

```
controladorPersona.imprimirCualquierLista(listaDePersona);
```

Para generar el id hemos implementado un método abstracto para que así herede tanto la clase controlador persona como la clase Teléfono.

```
public abstract int generarId();
```

Al igual le sobrescribimos el método que nos va a permitir que el id se genere automáticamente y vaya aumentando según el usuario registre datos.

```
@Override
public int generarId() {

    List<Persona> lista = findAll();
    System.out.println(lista);
    int cont = 0;
    if (lista.size() > 0) {


        for (Persona persona : lista) {

            int cent = persona.getId();

            if (cent > cont) {

                cont = cent;
            }
        }
        return cont + 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		

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

Para controlar el nullpointerexception he usado las excepciones try catch, ya que nos bota un error a la hora de buscar y con esto lo he resuelto sin problemas.

```
public T read(T objeto) {
    try {
        return lista.stream().filter(t -> t.equals(objeto)).findFirst().get();
    } catch (NullPointerException ex) {
        ex.printStackTrace();
        System.out.println("Error de lectura y escritura");
    }
    return null;
}
```


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

- Clase controlador abstracto

```
/*
 * 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 ups.edu.ec.controlador;

import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
import java.util.ArrayList;
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 santi
*/
public abstract class ControladorAbstracto<T> {

    private List<T> lista;

    public ControladorAbstracto() {

        lista = new ArrayList();

    }

    public boolean create(T objeto) {

        if (validar(objeto)) {

            lista.add(objeto);

        }

        return false;

    }

    public T read(T objeto) {

        try {

            return lista.stream().filter(t -> t.equals(objeto)).findFirst().get();

        } catch (NullPointerException ex) {

            ex.printStackTrace();
            System.out.println("Error de lectura y escritura");

        }

        return null;

    }

    public boolean update(T objeto) {

        int posicion = lista.indexOf(objeto);
        if (validar(objeto)) {
            lista.set(posicion, objeto);
            return true;
        }
        return false;


    }

    public boolean delete(T objeto) {

        if (lista.contains(objeto)) {
            lista.remove(objeto);
        }

    }
}

```

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

```

        return true;

    }
    return false;
}

public abstract int generarId();

public static void imprimirCualquierLista(List listar) {

    for (Object obj : listar) {

        Method[] metodos = obj.getClass().getMethods();
        for (Method metodo : metodos) {

            if (metodo.getName().equals("getId")) {

                try {
                    int cadena = (Integer) metodo.invoke(obj, null);
                    System.out.println(cadena);
                } catch (IllegalAccessException | IllegalArgumentException
                    | InvocationTargetException ex) {

java.util.logging.Logger.getLogger(ControladorAbstracto.class.getName()).
                    log(java.util.logging.Level.SEVERE, null, ex);

                }

            }

        }

    }

}

public abstract boolean validar(T objeto);

public List<T> findAll() {

    return lista;

}

}


```

- Clase Controlador Persona

```

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

```

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

```

* and open the template in the editor.
*/
package ups.edu.ec.controlador;

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

/**
 *
 * @author santi
 */
public class ControladorPersona extends ControladorAbstracto<Persona> {

    @Override
    public boolean validar(Persona objeto) {

        if (objeto.getNombre().equals(objeto.getApellido())) {

            return false;

        } else {

            return true;

        }

    }

    @Override
    public int generarId() {

        List<Persona> lista = findAll();
        System.out.println(lista);
        int cont = 0;
        if (lista.size() > 0) {

            for (Persona persona : lista) {

                int cent = persona.getId();

                if (cent > cont) {

                    cont = cent;

                }

            }
            return cont + 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		

- Clase controlador 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 ups.edu.ec.controlador;

import java.util.List;
import ups.edu.ec.modelo.Telefono;

/**
 *
 * @author santi
 */
public class ControladorTelefono extends ControladorAbstracto<Telefono> {

    @Override
    public boolean validar(Telefono objeto) {

        return true;

    }

    @Override
    public int generarId() {

        List<Telefono> lista = findAll();
        System.out.println(lista);

        int cont = 0;

        if (lista.size() > 0) {

            for (Telefono telefono : lista) {

                int cent = telefono.getId();

                if (cent > cont) {

                    cont = cent;

                }

            }

            return cont + 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		

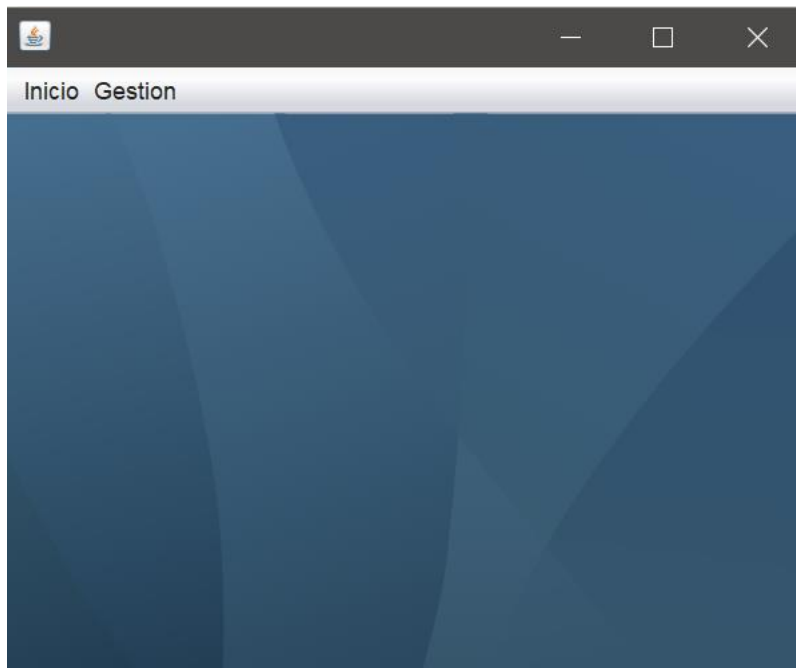
}

}

}


Resultados de la aplicacion

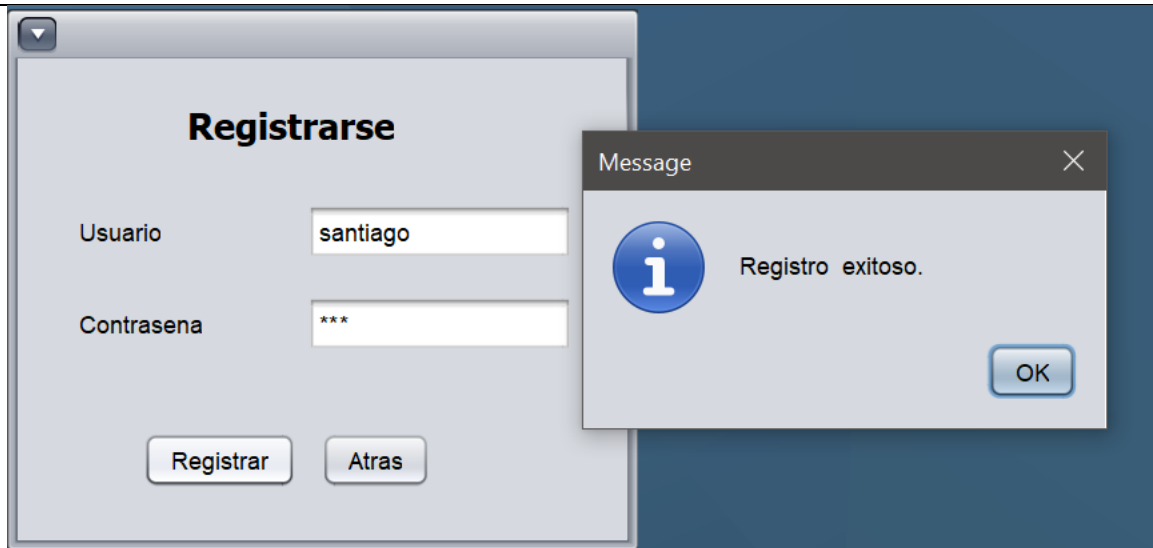
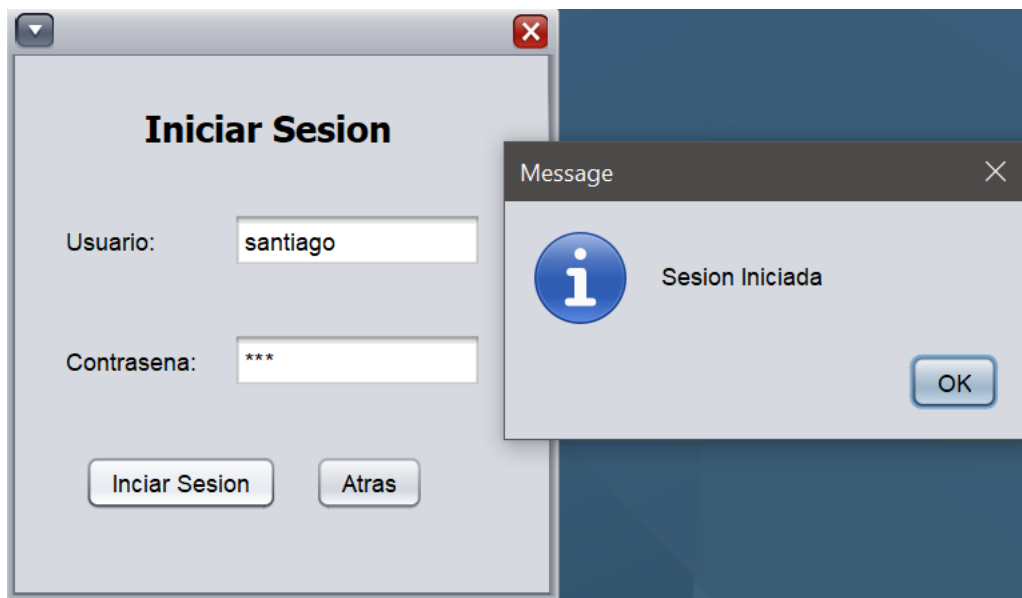
- Ventana Principal de la aplicación aquí podemos escoger entre múltiples opciones



- **Ventana Registrar Usuario**

El usuario deberá registrarse he iniciar sesión para poder acceder a los demas característica del programa

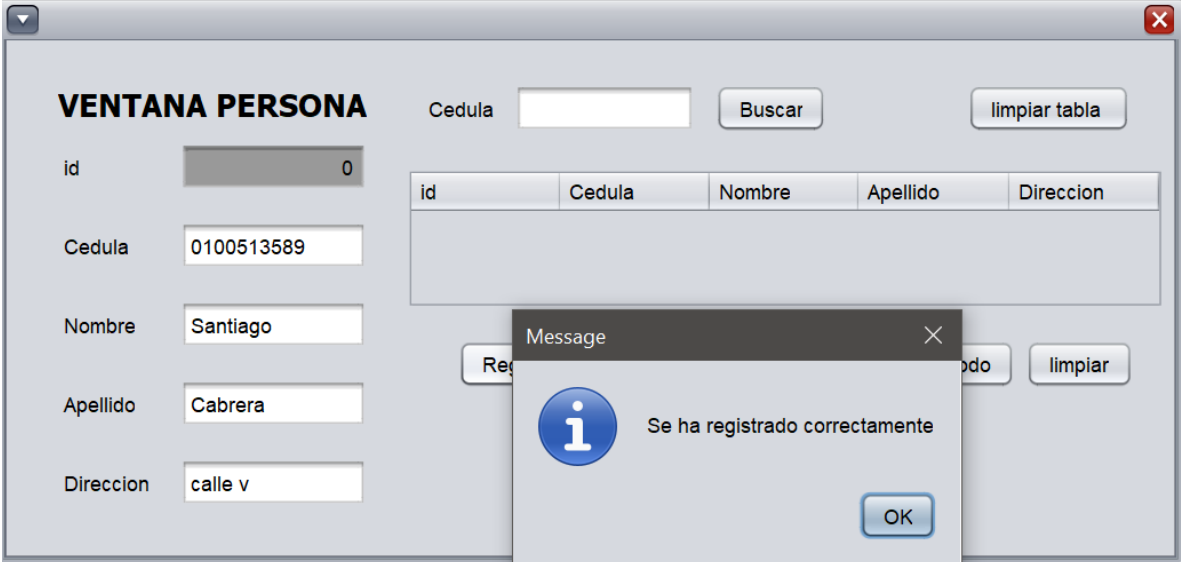
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Ventana Persona

En esta ventana el usuario podrá crear, actualizar, read, delete (crud) aquí se ha implementado la programación aprendida tanto en clases como en horas fuera de ellas.

- Al registrar nos saldrá un aviso de que se ha registrado correctamente o un aviso que nos diga que no ha llenado todos los campos.



VENTANA PERSONA

Cedula

id

Cedula


Nombre

Apellido


Direccion

id	Cedula	Nombre	Apellido	Direccion

Message

 Se ha registrado correctamente

En este caso estamos buscando por número de cedula por que se nos hace mucho más práctico para el usuario El dato que busquemos se nos mostrara en el jTable además de que se va ordenando por el apellido.



VENTANA PERSONA

Cedula

id

Cedula

Nombre

Apellido

Direccion

id	Cedula	Nombre	Apellido	Direccion
1	0100513589	Santiago	Cabrera	calle v

- Listar

Al pulsar listar todo este nos enviara los datos que el usuario haya registrado

VENTANA PERSONA

id:

Cedula:

Nombre:

Apellido:

Direccion:

Cedula:

id	Cedula	Nombre	Apellido	Direccion
1	0100513589	Santiago	Cabrera	calle v
2	0100513589	Nicolas	Cabrera	calle v

- Actualizar

Para actualizar los datos lo que debemos realizar es darle click en la tabla y los datos de inmediato se nos irán a los campos de texto donde tu podrás cambiarlos y presionar el botón actualizar.

VENTANA PERSONA

id:

Cedula:

Nombre:


Apellido:

Direccion:

Cedula:

id	Cedula	Nombre	Apellido	Direccion
2	0100513589	Nicolas	Cabrera	calle v
3	0100513589	Nicol	Cabrera	calle v

Es prácticamente lo mismo en el caso de las ventanas de teléfono con el breve detalle de que buscamos con el id.

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

CONCLUSIONES:

Este trabajo nos ha ayudado a recordar cosas vistas en anteriores ciclos y a implementar cosas nuevas que se nos han facilitado aplicarlas sobre este tipo de aplicación.

RECOMENDACIONES:

Recomiendo realizar más prácticas de este tipo debido a que nos ayuda en nuestro aprendizaje y a reforzar y buscar información sobre lo nuevo que nos ofrece java.

Nombre de estudiante: Jorge Santiago Cabrera Arias

Firma de estudiante:



Escaneado con CamScanner