
	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: Examen Practico Java	
OBJETIVO: Identificar los cambios importantes de Java Diseñar e Implementar expresiones regulares 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 expresion regular.			
3. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo de programación genérica.			
4. Realizar práctica codificando los codigos de las nuevas características de Java y su uso dentro de un sistema escolar.			
Enunciado Se desea generar un sistema que me permita extraer infomación del internet a traves de expresiones regulares, esta informacion permitira vincular actividades desarrolladas del los niños con aplicaciones mobiles que permitan apoyar en el desarrollo de las actividades planteadas (https://play.google.com/store?hl=es&gl=US). Adicionalmente, se debe realizar un sistema de gestion de alumnos y actividades planificadas por curso, dentro de este sistema se debe realizar un procesos de administracion de usuarios los mismo que son los docentes de cada curso escolar, en este sentido solo debemos tener un administrador (Rector) el encargado de crear docentes y el curso que se le asigna. Ejemplo Rector: Docentes: 1. Diego Quisi 2. Vladimir Robles 3. Etc. Cursos: 1 de basica 2 de basica 3 basica			

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

Asignacion de Curso – Docente

1 Basica -> Diego Quisi

2 Basica -> Vladimir Robles

Dentro de cada curso el docente gestionara los estudiantes y las actividades planificadas para el curso, estas actividades tendra una opcion de buscar aplicaciones moviles dentro de la tiendas de play store, obtenidas desde el internet, dentro de esta información lo importante es mostrar el link y una descripción para ello deberán utilizar expresiones regulares.

Ejemplo Docentes:

Alumnos

1. Juan Perez

2. Maria Peralta

3. .

Actividades:

1. Suma de numeros -> Obtener aplicaciones moviles (Link y Titulo)
2. Resta de numeros -> Obtener aplicaciones moviles
3. Oraciones compuestas -> Obtener aplicaciones moviles
4. Etc.

Toda esta infomación sera almacenada dentro de archivos y deberan tener aplicado al menos una patron de diseño y las nuevas características de programación de Java 8 o superior.

Al finalizar, generar el informe de la practica en formato PDF y subir todo el proyecto incluido el informe al repositorio personal.

La fecha de entrega: 23:55 del 01 de diciembre del 2020.

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 y expresiones regulares.

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: Examen Practico Java	
OBJETIVO ALCANZADO: Identificar los cambios importantes de Java Diseñar e Implementar expresiones regulares Entender la cada uno de las características nuevas en Java			
ACTIVIDADES DESARROLLADAS			
<p>1. Diseñar e implementar las características de Java para generar una expresión regular.</p> <p>Las expresiones regulares son una cadena de búsqueda simplificadas para buscar en un texto cadenas similares, sirve para validar o encontrar la expresión que se desea.</p> <p>En este proyecto se aplicó la expresión regular para buscar la direcciones o URL de las aplicaciones que se deseaba, pero para las expresiones se necesita un patrón y un cuerpo, para recuperar el cuerpo de la página de Play Store se ocupó las siguientes líneas de código:</p> <pre>String s = "https://play.google.com/store/search?q=" + variable + "=apps&hl=es&gl=US"; URL url = new URL(s); URLConnection conexion = url.openConnection(); InputStream is = conexion.getInputStream(); BufferedReader bufer = new BufferedReader(new InputStreamReader(is)); String contenedor = ""; String linea = bufer.readLine(); while (null != linea) { contenedor += linea; linea = bufer.readLine(); } Pattern url1 = Pattern.compile("(href=\\\"(\\\\/store\\\\/apps\\\\/details)\\.\\.\\\"(id=)\\.\\.\\S*\\\"\\\""); Matcher matcher = url1.matcher(contenedor);</pre> <p>Una vez definido el cuerpo se planteó el patrón de la expresión regular el cual quedo de la siguiente forma:</p> <pre>Pattern url1 = Pattern.compile("(href=\\\"(\\\\/store\\\\/apps\\\\/details)\\.\\.\\\"(id=)\\.\\.\\S*\\\"\\\""); Matcher matcher = url1.matcher(contenedor);</pre> <p>Cuando ejecutamos el código e imprimimos los resultados por consola se obtuvo el siguiente resultado:</p> <pre>href="/store/apps/details?id=com.miniklerogreniyor.quiz.kids.math" href="/store/apps/details?id=com.miniklerogreniyor.quiz.kids.math" href="/store/apps/details?id=andrei.brusentcov.schoolcalculator.free" href="/store/apps/details?id=andrei.brusentcov.schoolcalculator.free" href="/store/apps/details?id=andrei.brusentcov.schoolcalculator.free" href="/store/apps/details?id=andrei.brusentcov.schoolcalculator.free" href="/store/apps/details?id=simplemath.math.games.subtractiongames.mathgames" href="/store/apps/details?id=simplemath.math.games.subtractiongames.mathgames" href="/store/apps/details?id=simplemath.math.games.subtractiongames.mathgames" href="/store/apps/details?id=simplemath.math.games.subtractiongames.mathgames" BUILD SUCCESSFUL (total time: 2 minutes 2 seconds)</pre>			

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

2. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo de programación genérica.

Para resolver el problema planteado se utilizó la clase genérica para contener los métodos comunes, esta clase es tipo abstracta para que se pueda heredar los métodos que la componen, en el caso tenía un método para generar las id mediante reflexión:

```
public int generarCodigo(List<T> lista){
    int minimo=-100;
    if(!lista.isEmpty()){
        for (T t : lista) {

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

                if(metodo.getName().equals("getCodigo")){
                    try {
                        String i = metodo.invoke(t, null)+"";
                        if(Integer.parseInt(i) > minimo){
                            minimo = Integer.parseInt(i);
                        }
                    } catch (IllegalAccessException ex) {
                        Logger.getLogger(AbstractControlador.class.getName()).log(Level.SEVERE, null, ex);
                    } catch (IllegalArgumentException ex) {
                        Logger.getLogger(AbstractControlador.class.getName()).log(Level.SEVERE, null, ex);
                    } catch (InvocationTargetException ex) {
                        Logger.getLogger(AbstractControlador.class.getName()).log(Level.SEVERE, null, ex);
                    }
                }
            }
        }
    }
    return minimo+1;
}
return 1;
```

Y los métodos que se heredan de la clase abstracta son los siguientes:

```
public abstract class AbstractControlador<T>{

    public abstract boolean create(T objeto);

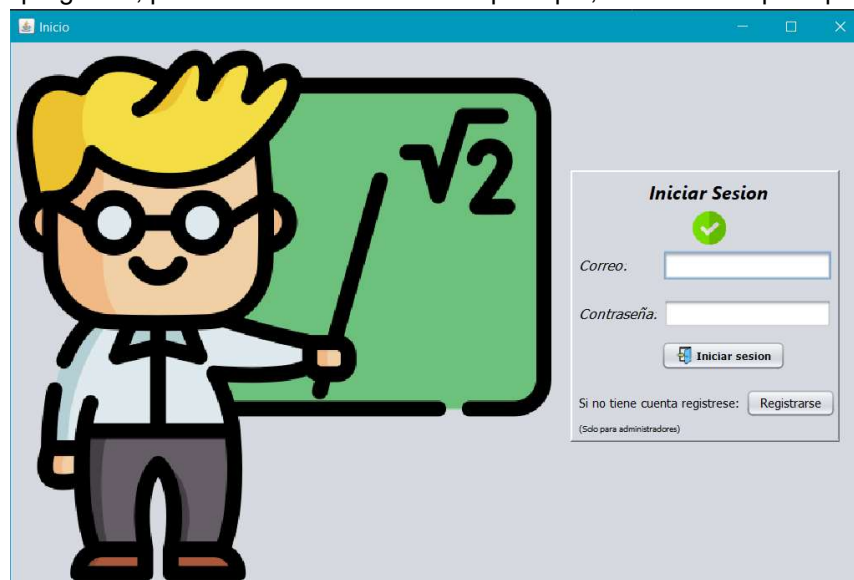
    public abstract T read(T objeto);


    public abstract boolean update(T objeto);

    public abstract boolean delite(T objeto);

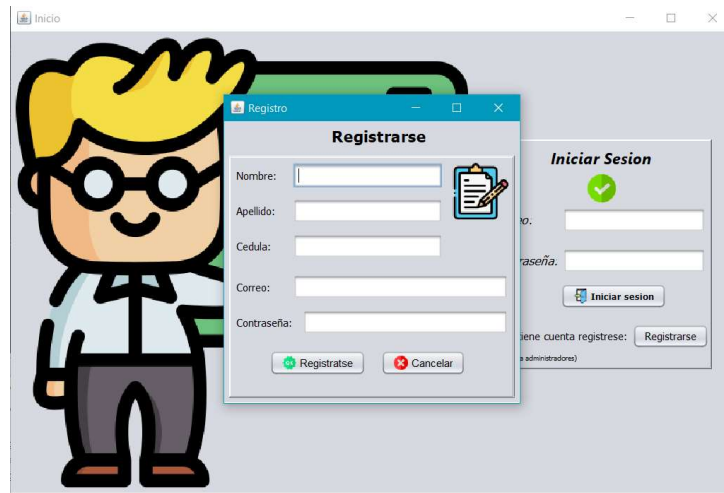
    public abstract List<T> findAll();
```

Cuando se ejecuta el programa, primero se visualiza la clase principal, en el caso la principal es el log in:

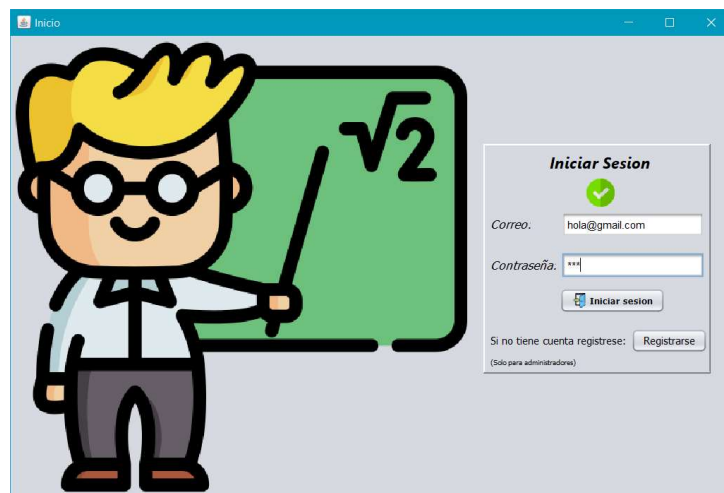


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

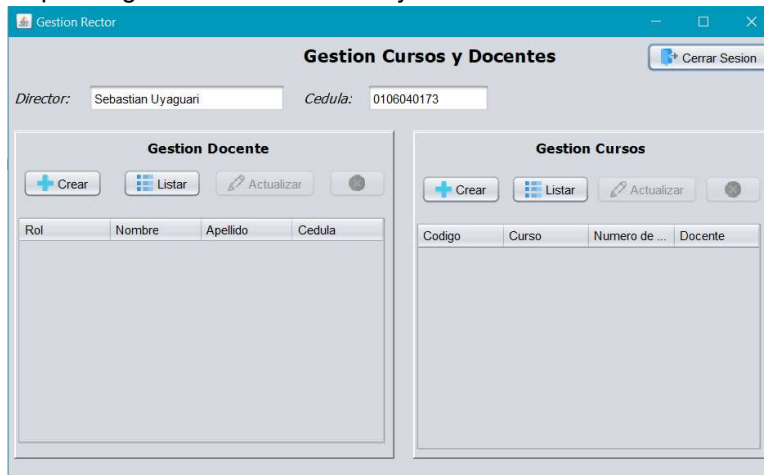
En el cual el usuario puede ingresar ya sea siendo Docente o sea Rector, en el caso que no está registrado puede registrarse como Administrador o Rector:




Una vez llenado los campos se aplasta el botón de registrarse y se puede ingresar su correo y contraseña en el login:

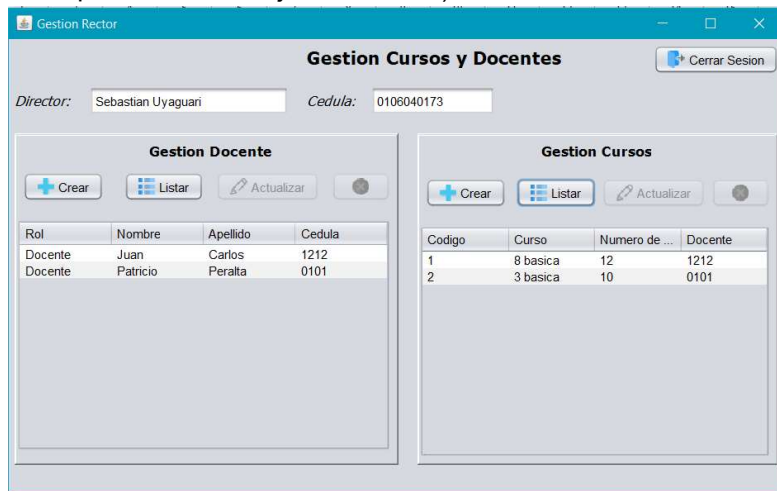


Cuando se inicia sesión se puede gestionar los docentes y los cursos:



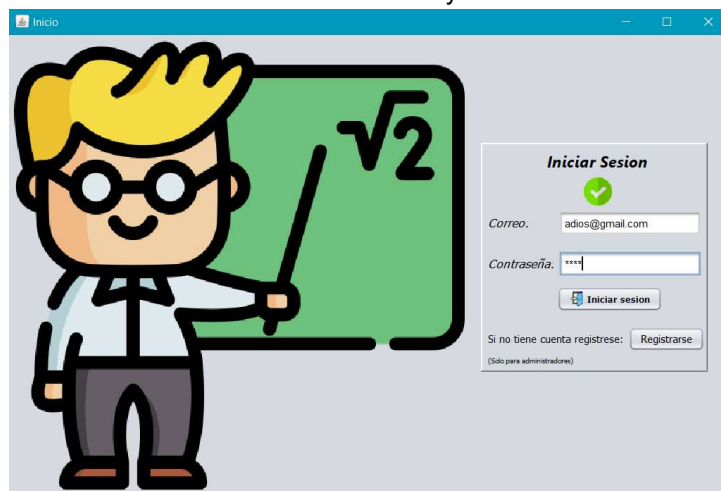
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

(Para actualizar se hace click primero en un objeto de la lista)



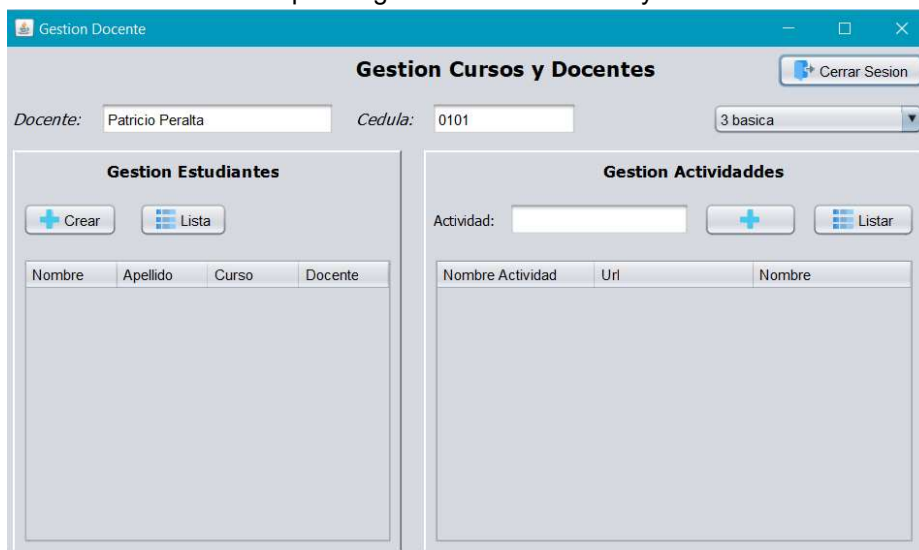
The interface shows the 'Gestion Rector' window with the title 'Gestion Cursos y Docentes'. It includes fields for 'Director' (Sebastian Uyaguari) and 'Cedula' (0106040173). There are two main sections: 'Gestion Docente' and 'Gestion Cursos'. Each section has buttons for '+ Crear', 'Listar', and 'Actualizar'. The 'Gestion Docente' table lists two teachers: Juan Carlos (Cedula 1212) and Patricio Peralta (Cedula 0101). The 'Gestion Cursos' table lists two courses: '8 basica' (Codigo 1, Docente 1212) and '3 basica' (Codigo 2, Docente 0101).

Una vez creado los docentes se cierra sesión de administrador y se inicia como Docente:




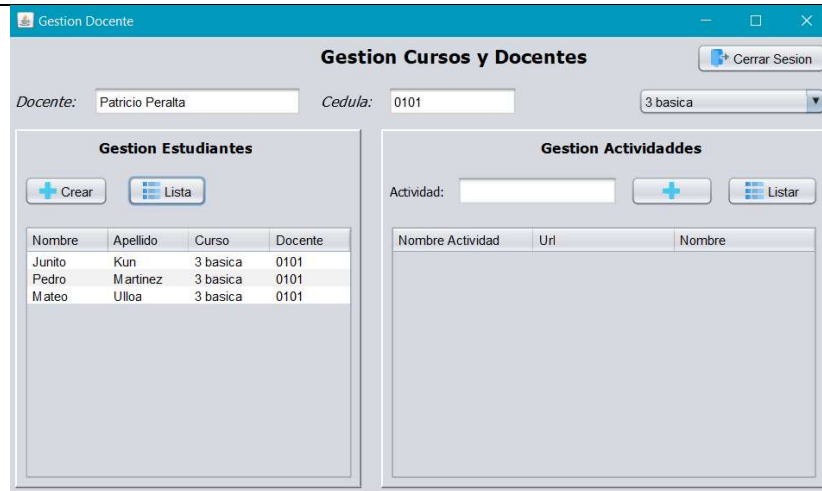
The 'Inicio' screen features a cartoon teacher character pointing at a greenboard with the equation $\sqrt{2}$. On the right, there is a login form titled 'Iniciar Sesión' with a green checkmark icon. It includes fields for 'Correo' (adlos@gmail.com) and 'Contraseña' (masked with asterisks). Below the fields are buttons for 'Iniciar sesion' and 'Registrarse'. A note at the bottom states 'Si no tiene cuenta registre: Registrarse (Solo para administradores)'.

Una vez iniciado sesión como docente se puede gestionar los alumnos y las actividades:



The 'Gestion Docente' window shows the 'Gestion Cursos y Docentes' section. It includes fields for 'Docente' (Patricio Peralta) and 'Cedula' (0101). There is a dropdown menu for 'Curso' set to '3 basica'. The interface is divided into two main sections: 'Gestion Estudiantes' and 'Gestion Actividades'. 'Gestion Estudiantes' has buttons for '+ Crear' and 'Lista', and a table with columns: Nombre, Apellido, Curso, and Docente. 'Gestion Actividades' has a field for 'Actividad', a '+ Crear' button, and a 'Listar' button, with a table below it having columns: Nombre Actividad, Url, and Nombre.

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



Nombre	Apellido	Curso	Docente
Junito	Kun	3 basica	0101
Pedro	Martinez	3 basica	0101
Mateo	Ulloa	3 basica	0101

Nombre Actividad	Uri	Nombre
------------------	-----	--------

Como patrón de Diseño se ocupó el singleton en todos los controladores para llevar una sola instancia en todas las clases que se necesite, ahorrándonos el trabajo de pasar como parámetros los controladores que se necesitarían para ejecutar el código:

```
public class ControladorActividades extends AbstractControlador<Actividad>{

    private static ControladorActividades instance = new ControladorActividades();

    public static ControladorActividades getIntance() {
        return instance;
    }
}
```

(Ejemplo clase Actividades)

Como base de datos se utilizó en el CRUD los archivos binarios permitiendo guardar modificar eliminar información de los archivos (Ejemplo CRUD de la clase Controlador Persona):

Método constructor de la clase controlador Persona:

```
/**
 * Estructura del archivo
 *
 * codigo: 4 bytes
 * rol: 7 bytes + 2 bytes
 * cedula: 10 bytes + 2 bytes
 * nombre: 25 bytes + 2 bytes
 * apellido: 25 bytes + 2 bytes
 * correo: 30 bytes + 2 bytes
 * contraseña: 20 bytes +2
 *
 * total: 133 bytes
 */
public ControladorPersona() {
    try {
        archivo = new RandomAccessFile("datos/persona.txt", "rw");
    } catch (FileNotFoundException ex) {
        System.out.println("Error Controlador persona: create");
    }
}
```

Método CRUD: create:


```
@Override
public boolean create(Persona objeto) {
    try {
        archivo.seek(archivo.length());
        archivo.writeInt(objeto.getCodigo());
        archivo.writeUTF(objeto.getRol());
        archivo.writeUTF(objeto.getCedula());
        archivo.writeUTF(objeto.getNombre());
        archivo.writeUTF(objeto.getApellido());
        archivo.writeUTF(objeto.getCorreo());
        archivo.writeUTF(objeto.getContraseña());
        return true;
    } catch (IOException ex) {
        System.out.println("Error COntrolador persona: create");
    }
    return false;
}
```

Método CRUD: read:

```
@Override
public Persona read(Persona objeto) {
    int salto = 0;
    try {
        while (salto < archivo.length()) {
            archivo.seek(salto+13);
            String cedula = archivo.readUTF();
            if(cedula.equals(objeto.getCedula())){
                archivo.seek(salto);
                return new Persona(archivo.readInt(), archivo.readUTF(),
                    archivo.readUTF(), archivo.readUTF(),
                    archivo.readUTF(), archivo.readUTF());
            }
            salto+=REGISTRO;
        }
    } catch (IOException ex) {
        System.out.println("Erro rcontrolador persona: read");
    }
    return null;
}
```

Método CRUD: update:

```
@Override
public boolean update(Persona objeto) {
    int salto = 0;
    try {
        while (salto < archivo.length()) {
            archivo.seek(salto);
            int codigo = archivo.readInt();
            if(codigo==objeto.getCodigo()){
                archivo.seek(salto);
                archivo.writeInt(objeto.getCodigo());
                archivo.writeUTF(objeto.getRol());
                archivo.writeUTF(objeto.getCedula());
                archivo.writeUTF(objeto.getNombre());
                archivo.writeUTF(objeto.getApellido());
                archivo.writeUTF(objeto.getCorreo());
                archivo.writeUTF(objeto.getContraseña());
                return true;
            }
            salto+=REGISTRO;
        }
    } catch (IOException ex) {
        System.out.println("Error controlador Persona: update");
    }
    return false;
}
```

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

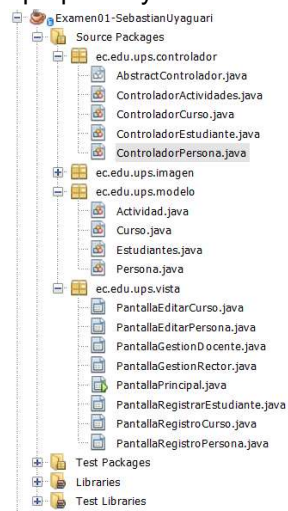
Método CRUD: delete:


```
@Override
public boolean delete(Persona objeto) {
    int salto = 0;
    try {
        while (salto < archivo.length()) {
            archivo.seek(salto);
            int codigo = archivo.readInt();
            if (codigo == objeto.getCodigo()) {
                archivo.seek(salto);
                archivo.writeInt(-100);
                archivo.writeUTF(String.format("%-" + 7 + "s", ""));
                archivo.writeUTF(String.format("%-" + 10 + "s", ""));
                archivo.writeUTF(String.format("%-" + 25 + "s", ""));
                archivo.writeUTF(String.format("%-" + 25 + "s", ""));
                archivo.writeUTF(String.format("%-" + 30 + "s", ""));
                archivo.writeUTF(String.format("%-" + 20 + "s", ""));
                return true;
            }
            salto += REGISTRO;
        }
    } catch (IOException ex) {
        System.out.println("Error controlador Persona: delete");
    }
    return false;
}
```

Método findAll:

```
@Override
public List<Persona> findAll() {
    List<Persona> lista = new ArrayList<>();
    int salto = 0;
    try {
        while (salto < archivo.length()) {
            archivo.seek(salto);
            int codigo = archivo.readInt();
            if (codigo > 0) {
                lista.add(new Persona(codigo, archivo.readUTF(),
                    archivo.readUTF(), archivo.readUTF(),
                    archivo.readUTF(), archivo.readUTF()));
            }
            salto += REGISTRO;
        }
    } catch (IOException ex) {
        System.out.println("Error controlador Persona findAll");
    }
    return lista;
}
```

El Proyecto completo contiene los siguientes paquetes y clases:



	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):

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 y expresiones regulares.
 Entender las funcionalidades adicionales de Java.

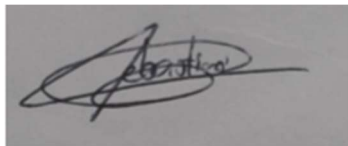
CONCLUSIONES:

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.

Nombre de estudiante: Sebastián Roberto Uyaguari Ramón



Firma de estudiante: