

CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Examen Práctico 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. Enunciado:</p> <p>Se desea generar un sistema que me permita extraer información del internet a través de expresiones regulares, esta información permitirá vincular actividades desarrolladas de los niños con aplicaciones móviles que permitan apoyar en el desarrollo de las actividades planteadas (https://play.google.com/store?hl=es&gl=US).</p> <p>Adicionalmente, se debe realizar un sistema de gestión de alumnos y actividades planificadas por curso, dentro de este sistema se debe realizar un proceso de administración 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.</p> <p>Ejemplo Rector:</p> <p>Docentes:</p> <ol style="list-style-type: none">1. Diego Quisi2. Vladimir Robles3. Etc. <p>Cursos:</p> <p>1 de básica</p> <p>2 de básica</p> <p>3 básica</p> <p>Asignación de Curso – Docente</p> <p>1 Básica -> Diego Quisi</p> <p>2 Básica -> Vladimir Robles</p> <p>Dentro de cada curso el docente gestionará los estudiantes y las actividades planificadas para el curso, estas actividades tendrán una opción de buscar aplicaciones móviles dentro de la tienda 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.</p> <p>Ejemplo Docentes:</p> <p>Alumnos</p> <ol style="list-style-type: none">1. Juan Perez2. Maria Peralta3. . <p>Actividades:</p> <ol style="list-style-type: none">1. Suma de números -> Obtener aplicaciones móviles (Link y Titulo)2. Resta de números -> Obtener aplicaciones móviles3. Oraciones compuestas -> Obtener aplicaciones móviles4. Etc.		

ec.edu.ups.controlador :

Controlador:

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

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

/**
 *
 * @author Usuario
 * @param <E>
 */
public abstract class Controlador<E> {

    private String ruta;
    private List<E> lista;

    public Controlador(String ruta) {
        lista = new ArrayList();
        this.ruta = ruta;
    }

    public List<E> getLista() {
        return lista;
    }

    public void setLista(List<E> lista) {
        this.lista = lista;
    }
}
```

```

public void cargarDatos() throws ClassNotFoundException, IOException {
    ObjectInputStream datos = null;
    try {
        File f = new File(ruta);
        FileInputStream a = new FileInputStream(f);
        datos = new ObjectInputStream(a);

        lista = (List<E>) datos.readObject();

    } catch (IOException e) {

    }

}

public void guardarDatos(String ruta) throws IOException {
    ObjectOutputStream datos = null;
    File f = new File(ruta);
    FileOutputStream archivo = new FileOutputStream(f);
    datos = new ObjectOutputStream(archivo);
    datos.writeObject(lista);

}

public boolean crear(E obj) {

    if (validar(obj) == true) {
        return lista.add(obj);
    }
    return false;

}

public Optional<E> buscar(E comparar) {
    return lista.stream().filter(objeto -> objeto.equals(comparar)).findFirst();

}

```

```

public int posicion(E objeto) {
    for (int i = 0; i < lista.size(); i++) {
        E objetoL = lista.get(i);
        if (objetoL.equals(objeto)) {
            return i;
        }
    }
    return -1;
}

```

```

public boolean eliminar(E objeto) {
    Optional<E> buscar = buscar(objeto);
    E objetoE = buscar.get();
    if (objetoE != null) {
        System.out.println("Verdadero");
        return lista.remove(objetoE);
    }
    System.out.println("Falso");
    return false;
}

```

```

public boolean actualizar(E objeto2) {
    int pos = posicion(objeto2);
    if (pos >= 0) {
        lista.set(pos, objeto2);
        System.out.println("TRUE");
        return true;
    }
    System.out.println("FALSE");
    return false;
}

```

```

public abstract boolean validar(E objeto);

public abstract int generarId();

```

ControladorActividad:

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

import ec.edu.ups.modelo.Actividad;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
/**
 *
 * @author Usuario
 */
public class ControladorActividad extends Controlador<Actividad> {

    private Pattern patron;
    private Matcher corpus;

    public ControladorActividad(String ruta) {
        super(ruta);
    }

    @Override
    public boolean validar(Actividad objeto) {
        return true;
    }
}
```

```

@Override
public int generarId() {
    List<Actividad> temp = new ArrayList();
    for (Actividad a : super.getLista()) {
        Actividad m = (Actividad) a;
        temp.add(m);

    }

    if (temp.size() > 0 && temp != null) {
        return temp.get(temp.size() - 1).getCodigo() + 1;
    } else {
        return 1;
    }

}

public void ingresarRegex(String regex) {
    patron = Pattern.compile(regex);
}

public Set<String> obtenerUrl(String texto) {

    Set<String> resultado = new HashSet();
    corpus = patron.matcher(texto);

    while (corpus.find()) {
        resultado.add(corpus.group(0));
    }
    return resultado;

}

public List<Actividad> actividades() {

    List<Actividad> lista = new ArrayList();
    Actividad actividad;
    Iterator i = super.getLista().iterator();
    while (i.hasNext()) {
        actividad = (Actividad) i.next();
        lista.add(actividad);

    }
    return lista;

}

```

ControladorAlumno:

```

package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Alumno;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/**
 *
 * @author Usuario
 */
public class ControladorAlumno extends Controlador<Alumno> {

    private ControladorPersona control;

    public ControladorAlumno(ControladorPersona control, String ruta) {
        super(ruta);
        this.control = control;
    }

    @Override
    public boolean validar(Alumno objeto) {
        return control.validar(objeto);
    }

    @Override
    public int generarId() {
        return 0;
    }

    public List<Alumno> alumnos() {

        List<Alumno> lista = new ArrayList();
        Alumno alumno;
        Iterator i = super.getList().iterator();
        while (i.hasNext()) {
            alumno = (Alumno) i.next();
            lista.add(alumno);
        }
        return lista;
    }
}

```

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

import ec.edu.ups.modeloCurso;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
/**
 *
 * @author Usuario
 */
public class ControladorCurso extends Controlador<Curso> {

    public ControladorCurso(String ruta) {
        super(ruta);
    }

    @Override
    public boolean validar(Curso objeto) {
        return true;
    }

}
```



```

@Override
public int generarId() {
    List<Curso> temp = new ArrayList();
    for (Curso a : super.getList()) {
        Curso m = (Curso) a;
        temp.add(m);
    }

    if (temp.size() > 0 && temp != null) {
        return temp.get(temp.size() - 1).getCodigo() + 1;
    } else {
        return 1;
    }
}

public List<Curso> cursos() {

    List<Curso> lista = new ArrayList();
    Curso curso;
    Iterator i = super.getList().iterator();
    while (i.hasNext()) {
        curso = (Curso) i.next();
        lista.add(curso);
    }
    return lista;
}

```

ControladorDocente:

```

package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Docente;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
/**
 *
 * @author Usuario
 */
public class ControladorDocente extends Controlador<Docente> {

    private ControladorPersona control;

    public ControladorDocente(ControladorPersona control, String ruta) {
        super(ruta);
        this.control = control;
    }

    @Override
    public int generarId() {
        return 0;
    }

    public List<Docente> docentes() {

        List<Docente> lista = new ArrayList();
        Docente docente;
        Iterator i = super.getList().iterator();
        while (i.hasNext()) {
            docente = (Docente) i.next();
            lista.add(docente);
        }
        return lista;
    }

    @Override
    public boolean validar(Docente objeto) {
        return control.validar(objeto);
    }
}

```

ControladorPersona:

```

package ec.edu.ups.controlador;

import ec.edu.ups.modelo.Persona;
/**
 *
 * @author Usuario
 */
public class ControladorPersona extends Controlador<Persona> {

    public ControladorPersona(String ruta) {
        super(ruta);
    }

    @Override
    public boolean validar(Persona objeto) {
        int suma = 0;
        String x = objeto.getCedula();
        if (x.length() == 9) {
            return false;
        } else {
            int a[] = new int[x.length() / 2];
            int b[] = new int[(x.length() / 2)];
            int c = 0;
            int d = 1;
            for (int i = 0; i < x.length() / 2; i++) {
                a[i] = Integer.parseInt(String.valueOf(x.charAt(c)));
                c = c + 2;
                if (i < (x.length() / 2) - 1) {
                    b[i] = Integer.parseInt(String.valueOf(x.charAt(d)));
                    d = d + 2;
                }
            }

            for (int i = 0; i < a.length; i++) {
                a[i] = a[i] * 2;
                if (a[i] > 9) {
                    a[i] = a[i] - 9;
                }
                suma = suma + a[i] + b[i];
            }
            int aux = suma / 10;
            int dec = (aux + 1) * 10;
            if ((dec - suma) == Integer.parseInt(String.valueOf(x.charAt(x.length() - 1)))) {
                return true;
            } else if (suma % 10 == 0 && x.charAt(x.length() - 1) == '0') {
                return true;
            } else {
                return false;
            }
        }
    }

    @Override
    public int generarId() {
        return 0;
    }
}

```

ControladorUsuario:

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

import ec.edu.ups.modelo.Docente;
import ec.edu.ups.modelo.Usuario;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/**
 *
 * @author Usuario
 */
public class ControladorUsuario extends Controlador<Usuario> {

    private Usuario usuario;

    public ControladorUsuario(String ruta) {
        super(ruta);
    }

    @Override
    public boolean validar(Usuario objeto) {
        return true;
    }

    @Override
    public int generarId() {
        return 0;
    }
}
```

```

public List<Usuario> usuarios() {

    List<Usuario> lista = new ArrayList();
    Usuario user;
    Iterator i = super.getList().iterator();
    while (i.hasNext()) {
        user = (Usuario) i.next();
        lista.add(user);
    }
    return lista;
}

public Usuario getUsuario() {
    return usuario;
}

public boolean iniciarSesion(String correo, String pass) {
    for (Usuario user : super.getList()) {
        Usuario u = (Usuario) user;
        if (u.getCorreo().equals(correo) && u.getPass().equals(pass)) {
            this.usuario = u;
            return true;
        }
    }
    return false;
}

public Usuario buscarUser(Docente docente) {

    for (Usuario us : super.getList()) {
        Usuario user = (Usuario) us;
        if (docente.equals(user.getDocente())) {
            this.usuario = user;
        }
    }
    return null;
}

```

ec.edu.ups.modelo :

Actividad:

```
package ec.edu.ups.modelo;

import java.io.Serializable;

/**
 * @author Usuario
 */
public class Actividad implements Serializable {

    private int codigo;
    private String titulo;
    private String descripcion;
    private String link;
    private Curso curso;

    public Actividad(int codigo, String titulo, String descripcion, String link, Curso curso) {
        this.codigo = codigo;
        this.titulo = titulo;
        this.descripcion = descripcion;
        this.link = link;
        this.curso = curso;
    }

    public Curso getCurso() {
        return curso;
    }

    public void setCurso(Curso curso) {
        this.curso = curso;
    }
}
```

```
public int getCodigo() {  
    return codigo;  
}  
  
public void setCodigo(int codigo) {  
    this.codigo = codigo;  
}  
  
public String getTitulo() {  
    return titulo;  
}  
  
public void setTitulo(String titulo) {  
    this.titulo = titulo;  
}  
  
public String getDescripcion() {  
    return descripcion;  
}  
  
public void setDescripcion(String descripcion) {  
    this.descripcion = descripcion;  
}  
  
public String getLink() {  
    return link;  
}  
  
public void setLink(String link) {  
    this.link = link;  
}
```

```

@Override
public int hashCode() {
    int hash = 3;
    hash = 97 * hash + this.codigo;
    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 Actividad other = (Actividad) obj;
    if (this.codigo != other.codigo) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("Actividad{codigo=").append(codigo);
    sb.append(", titulo=").append(titulo);
    sb.append(", descripcion=").append(descripcion);
    sb.append(", link=").append(link);
    sb.append('}');
    return sb.toString();
}

```

Alumno:


```

package ec.edu.ups.modelo;

/**
 *
 * @author Usuario
 */
public class Alumno extends Persona{
    private Curso curso;
    private String aula;

    public Alumno(String cedula, String nombre, String apellido,
        int edad, String direccion, Curso curso, String aula) {
        super(cedula, nombre, apellido, edad, direccion);
        this.curso = curso;
        this.aula = aula;
    }

    public Curso getCurso() {
        return curso;
    }

    public void setCurso(Curso curso) {
        this.curso = curso;
    }

    public String getAula() {
        return aula;
    }

    public void setAula(String aula) {
        this.aula = aula;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Alumno{curso=").append(curso);
        sb.append(", aula=").append(aula);
        sb.append('}');
        return sb.toString();
    }
}

```

Curso:

```
import java.io.Serializable;

/**
 *
 * @author Usuario
 */
public class Curso implements Serializable {

    private int codigo;
    private String nombre;
    private String seccion;

    public Curso(int codigo, String nombre, String seccion) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.seccion = seccion;
    }

    public Curso() {
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

```

public String getSeccion() {
    return seccion;
}

public void setSeccion(String seccion) {
    this.seccion = seccion;
}

@Override
public int hashCode() {
    int hash = 3;
    hash = 29 * hash + this.codigo;
    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 Curso other = (Curso) obj;
    if (this.codigo != other.codigo) {
        return false;
    }
    return true;
}

```

Docente:

```
package ec.edu.ups.modelo;
```

```
/**
```

```
 *
```

```
 * @author Usuario
```

```
 */
```

```
public class Docente extends Persona {
```

```
    private String titulo;
```

```
    private String tipo;
```

```
    private Usuario usuario;
```

```
    private Curso curso;
```

```
    public Docente(String cedula, String nombre, String apellido,
```

```
        int edad, String direccion) {
```

```
        super(cedula, nombre, apellido, edad, direccion);
```

```
    }
```

```
    public Docente(String cedula, String nombre, String apellido,
```

```
        int edad, String direccion, String titulo, String tipo) {
```

```
        super(cedula, nombre, apellido, edad, direccion);
```

```
        this.titulo = titulo;
```

```
        this.tipo = tipo;
```

```
    }
```

```
    public String getTitulo() {
```

```
        return titulo;
```

```
    }
```

```
    public void setTitulo(String titulo) {
```

```
        this.titulo = titulo;
```

```
    }
```

```

public String getTipo() {
    return tipo;
}

public void setTipo(String tipo) {
    this.tipo = tipo;
}

public Curso getCurso() {
    return curso;
}

public void setCurso(Curso curso) {
    this.curso = curso;
}

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("Docente{titulo=").append(titulo);
    sb.append(", tipo=").append(tipo);
    sb.append(", curso=").append(curso);
    sb.append('}');
    return sb.toString();
}

```

Persona:

```
package ec.edu.ups.modelo;

import java.io.Serializable;
import java.util.Objects;
/**
 *
 * @author Usuario
 */
public class Persona implements Serializable {

    private String cedula;
    private String nombre;
    private String apellido;
    private int edad;
    private String direccion;

    public Persona(String cedula, String nombre, String apellido,
        int edad, String direccion) {
        this.cedula = cedula;
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;

        this.direccion = direccion;
    }

    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 int getEdad() {  
    return edad;  
}  
  
public void setEdad(int edad) {  
    this.edad = edad;  
}  
  
public String getDireccion() {  
    return direccion;  
}  
  
public void setDireccion(String direccion) {  
    this.direccion = direccion;  
}
```

```

@Override
public int hashCode() {
    int hash = 7;
    hash = 79 * hash + Objects.hashCode(this.cedula);
    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 Persona other = (Persona) obj;
    if (!Objects.equals(this.cedula, other.cedula)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();
    sb.append("Persona{cedula=").append(cedula);
    sb.append(", nombre=").append(nombre);
    sb.append(", apellido=").append(apellido);
    sb.append(", edad=").append(edad);

    sb.append(", direccion=").append(direccion);
    sb.append('}');
    return sb.toString();
}

```

Usuario:


```
package ec.edu.ups.modelo;

import java.io.Serializable;

/**
 * @author user
 */
public class Usuario implements Serializable {

    private Docente docente;
    private String cargo;
    private String correo;
    private String contra;

    public Usuario(Docente docente, String cargo, String correo, String contra) {
        this.docente = docente;
        this.cargo = cargo;
        this.correo = correo;
        this.contra = contra;
    }

    public Usuario() {

    }

    public String getCargo() {
        return cargo;
    }

    public void setCargo(String cargo) {
        this.cargo = cargo;
    }
}
```

```

public Docente getDocente() {
    return docente;
}

public void setDocente(Docente docente) {
    this.docente = docente;
}

public String getCorreo() {
    return correo;
}

public void setCorreo(String correo) {
    this.correo = correo;
}

public String getContra() {
    return contra;
}

public void setContra(String contra) {
    this.contra = contra;
}

@Override
public String toString() {
    return "Usuario{" + "docente=" + docente + ", cargo=" +
        cargo + ", correo=" + correo + ", contra=" + contra + '}';
}

```

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: Nelson Paul Ortega Segarra

Firma de estudiante:

A handwritten signature in black ink on a yellow background. The signature is a stylized, cursive letter 'D' with a vertical stroke through the center, resembling a '10' or a 'D' with a vertical line.