

GUÍA DE LABORATORIO N° 01

Semana 03: Listas simplemente Enlazadas

INTRODUCCION

En el tratamiento de las listas simplemente enlazadas son aquellas que tienen una referencia al elemento siguiente. La diferencia entre una lista y una cola o pila es que los elementos de la lista pueden insertarse en cualquier lugar, logrando de esta forma que la lista quede ordenada. Cada elemento de la lista será una estructura auto referenciada, la cual está formada por los datos y un puntero a la estructura del mismo tipo.

I. OBJETIVOS

1. Desarrollo de ejercicios de Manejo de Cadenas usando Swing WT.
2. Codificar, compilar y ejecutar los ejercicios de aplicación.
3. Conocer la estructura de un programa visual en Java.
4. Se espera que el estudiante asocie los conocimientos nuevos con la "nueva plantilla" que se propone.

II. EQUIPOS Y MATERIALES

- ③ Computadora personal.
- ③ Programa NetBeans.
- ③ Notas de los ejercicios resueltos en la clase.

III. METODOLOGIA Y ACTIVIDADES

- Codificar los ejercicios desarrollados en el aula.
- Presentar avances y ejecución de cada uno de los ejercicios al docente o jefe de práctica encargado para la calificación correspondiente.
- Guardar la carpeta de sus archivos a sus memorias.
- Apagar el computador y dejarla en buen estado al retirarse del laboratorio dejar todo en orden.

IV. OBSERVACION

- 📌 El estudiante deberá crear una carpeta de trabajo con el nombre "Laboratorio_03ED", a donde deberá direccionar su proyecto a crear.
- 📌 Se recomienda que el estudiante haya planteado los ejercicios de la Guía para que pueda comprender las soluciones que se proponen.

V. EJERCICIOS.

Construya una aplicación que permita realizar el registro de empleados donde se podrá guardar, mostrar, consultar, actualizar y eliminar el registro de empleado. Para todas estas operaciones considere el ingreso del código del empleado

SOLUCION:

Paso 1: Construya el siguiente diseño:

LISTAS ENLAZADAS - Operaciones de Inserción, Búsqueda, Modificación y Eliminación

REGISTRO DE EMPLEADOS

CODIGO

NOMBRE

SUELDO

Guardar	Actualizar
Consultar	Eliminar
Reestaurar	Salir

N°	CODIGO	NOMBRES Y APELLIDOS	SUELDOS

Ahora váyase al editor de código y genere el siguiente código haga doble click sobre los botones y genere los métodos faltan:

```

package listas_simples;
import java.text.DecimalFormat;
import java.awt.Font;
import javax.swing.JOptionPane;

public class listas extends javax.swing.JFrame {
    //Declaracion de la Lista enlazada simple
    public class Nodo {
        String codigo;
        String nombre;
        String sueldo;
        Nodo sig;
        public Nodo(String cod, String nom,String suel)
        {
            codigo=cod;
            nombre=nom;
            sueldo=suel;
        }
    }
    //Declaracion de variables locales
    public Nodo ini,fin;
    public Nodo pFound;
    int num=0;
    public listas() {
        initComponents();
    }

```

Aquí se muestran los métodos que tiene que crear antes ejecutar los códigos

Instrucciones de los métodos utilizados en el aplicativo

```
void Encabezado()
{
    txaReporte.setFont(new Font("monospaced", Font.PLAIN, 12));
    txaReporte.setText("");
    //Insertando encabezados al TextArea
    txaReporte.append("");
    txaReporte.append("    N°    CODIGO    NOMBRES Y APELLIDOS    SUELDOS\n");
    txaReporte.append("-----\n");
}

void Habilitar()
{
    btnActualizar.setEnabled(true);
    btnEliminar.setEnabled(true);
    btnGuardar.setEnabled(false);
}

void Deshabilitar()
{
    btnActualizar.setEnabled(false);
    btnEliminar.setEnabled(false);
    btnGuardar.setEnabled(true);
}

void LimpiarEntradas()
{
    txtCodigo.setText("");
    txtNombre.setText("");
    txtSueldo.setText("");
    txtCodigo.requestFocus();
}

void VerDatos() {
    //Variable para recorrer la lista
    String cod,nom,s;
    Nodo aux=ini;
    num=0;
    //Colocando el Encabezado
    Encabezado();
    //Recorriendo la Lista
    while(aux!=null)
    {
        cod=aux.codigo;
        nom=aux.nombre;
        s=aux.sueldo;
        num++;
        String numera = String.valueOf(num);
        //modificando el tamaño de la numeracion con espacios en blanco a la izquierda
        for (int i = String.valueOf(num).length(); i < 5; i++) {
            numera = " " + numera;
        }
        //modificando el tamaño de la cadena codigo con espacios en blanco a la derecha
        for (int i = cod.length(); i < 12; i++) {
            cod = cod + " ";
        }
        //modificando el tamaño de la cadena nombre con espacios en blanco a la derecha
        for (int i = nom.length(); i < 28; i++) {
            nom = nom + " ";
        }
        //Le damos formato al sueldo solo con dos decimales
        DecimalFormat df2 = new DecimalFormat("####.00");
        s = df2.format(Double.valueOf(s));
        //modificando el tamaño de la cadena sueldo con espacios en blanco a la izquierdo
        for (int i = s.length(); i < 12; i++) {
            s = " " + s;
        }
        //Colocando la información en el TextArea
        txaReporte.append(numera + "    " + cod+nom + s + "\n");
        aux=aux.sig;
    }
}
```

Instrucciones en los métodos para las operaciones de la lista enlazada

```
Nodo Buscar(Nodo inicio,String cod)
{
    Nodo pos=inicio;
    //Recorriendo la lista para encontrar la informacion
    while(pos!=null && !cod.equalsIgnoreCase(pos.codigo))
        pos=pos.sig;
    return pos;
}

Nodo InsertaInicio(Nodo inicio,String cod,String nom,String suel)
{
    Nodo nuevo=new Nodo(cod,nom,suel);
    //Realizando los enlaces correspondientes
    nuevo.sig=inicio;
    inicio=nuevo;
    return inicio;
}

void Eliminar(Nodo actual)
{
    //Creando un puntero al nodo anterior del actual
    Nodo anterior=ini;
    //1. Buscando el nodo anterior
    while(anterior.sig!=actual && anterior.sig!=null)
    {
        anterior=anterior.sig;
    }
    // 2.- Elimina el nodo si existe
    if (actual != null)
    {
        if (anterior == actual) ini = actual.sig; // borrar el primero
        else anterior.sig = actual.sig; // borrar en otro sitio
    }
}
```

Instrucciones del botón Guardar

```
private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    //Capturando la informacion de los objetos
    String cod=txtCodigo.getText();
    String nom=txtNombre.getText().toUpperCase();
    String suel=txtSueldo.getText();
    //Creando el nodo de la Lista en memoria y colocando la informacion
    ini=InsertaInicio(ini,cod,nom,suel);
    LimpiarEntradas();
    //insertar(num,cod,nom,suel);
    VerDatos();
}
```

Instrucciones del Botón Actualizar

```
private void btnActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    //Colocando la informacion en el puntero pFound
    pFound.codigo=txtCodigo.getText();
    pFound.nombre=txtNombre.getText();
    pFound.sueldo=txtSueldo.getText();
    LimpiarEntradas();
    Deshabilitar();
    VerDatos();
}
```

Instrucciones del Botón Consultar

```
private void btnConsultarActionPerformed(java.awt.event.ActionEvent evt) {  
    String cod=txtCodigo.getText();  
    if(cod.equalsIgnoreCase("")) {  
        JOptionPane.showMessageDialog(this,"Ingrese un codigo por favor");  
    } else {  
  
        //Llamada a la funcion que retorna la posicion del dato buscado  
        pFound=Buscar(ini,cod);  
        //Verificando el puntero pFound para mostrar la inf. buscada  
        if(pFound!=null) {  
            txtNombre.setText(pFound.nombre);  
            txtSueldo.setText(pFound.sueldo);  
            //Habilitamos los objetos para eliminar y actualizar  
            Habilitar();  
  
        } else {  
            JOptionPane.showMessageDialog(this,"El código: "+cod + ", no esta en la Lista..");  
        }  
    }  
}
```

Instrucciones del Botón Eliminar

```
private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    Eliminar(pFound);  
  
private void btnReestaurarActionPerformed(java.awt.event.ActionEvent evt) {  
    LimpiarEntradas();  
    Deshabilitar();  
}  
  
Deshabilitar();  
}
```

Instrucciones del Botón Restaurar

Instrucciones del Evento Opened del Window

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
    //Colocando el encabezado  
    Encabezado();  
    //Deshabilitando objetos  
    Deshabilitar();  
}
```

Luego de terminar Shift+F6 y ejecutarse el aplicativo deberá verse de la siguiente manera:

LISTAS ENLAZADAS - Operaciones de Inserción, Búsqueda, Modificación y Eliminación

REGISTRO DE EMPLEADOS

CODIGO

NOMBRE

SUELDO

Guardar	Actualizar
Consultar	Eliminar
Reestaurar	Salir

N°	CODIGO	NOMBRES Y APELLIDOS	SUELDOS
1	1200	YOLANDA GALECIO	8411,00
2	8444	MILUSKA SAAVEDRA	4112,00
3	5222	JOSE ALAIN	841,00
4	9875	CARLOS SAGASTEGUI	1600,00

EJERCICIO PROPUESTO

Construya un programa que guarde en una lista simple enlazada, la siguiente información de los empleados de una Compañía: Código, Nombre, Tipo de Contrato (A plazo fijo, Servicios No personales y Service), sueldo, monto asignado por movilidad y minutos de tardanza y permita mostrar la siguiente información:

- a) Número de empleados con más de una 15 minutos de tardanzas.
- b) Nombre del empleado con el mayor tiempo de tardanzas y que tenga un tipo de contrato por Services.
- c) El mayor monto de movilidad asignado a un empleado con contrato a Plazo Fijo con un sueldo menor a 1500 soles.

Su programa también tendrá la opción de consultar, actualizar y eliminar registros.