

PROGRAMACIÓN ORIENTADA A OBJETOS

Docente: Ing. Freddy Infantes Quiroz
Correo: finfantesq@upao.edu.pe



LABORATORIO 14

Flujos y Archivos



Al término del Laboratorio enviar el Proyecto completo con el ejercicio guiado y los propuestos al

correo: finfantesq@upao.edu.pe

Asunto: [POO_Lab14_FlujosyArchivos](#)

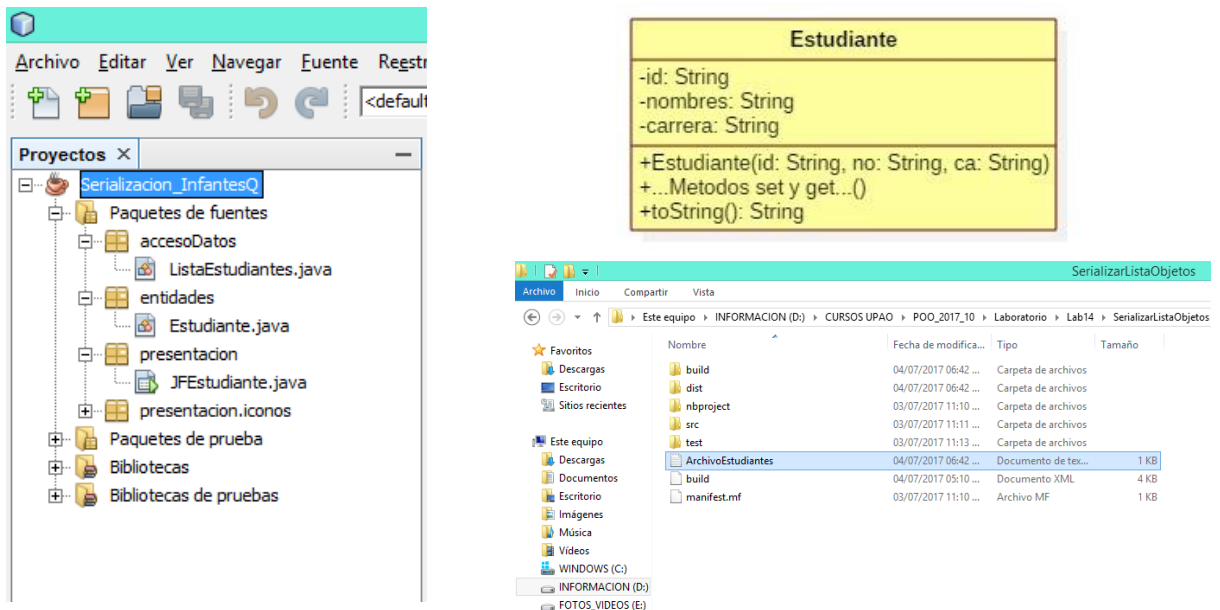
Contenido

Enfatizar los conceptos en:

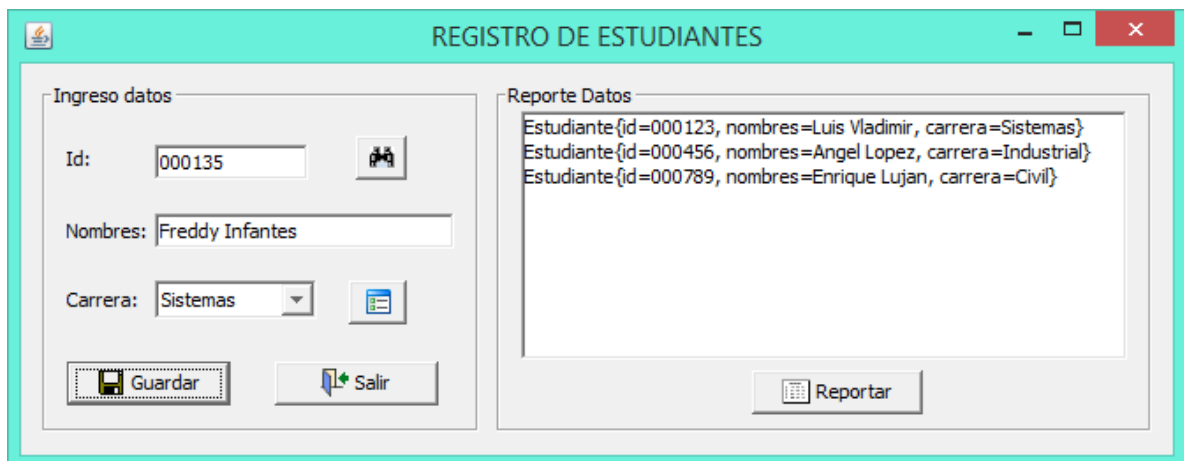
- Flujos y archivos
- Serializacion

EJERCICIO – GUIADO

1. Crear un **Proyecto** en Netbeans con el nombre: **Serializacion_InfantesQ**, y luego crear tres **paquetes** según la siguiente estructura.



2. Agregar un **JFrame** al paquete presentación llamado **JFEstudiante**

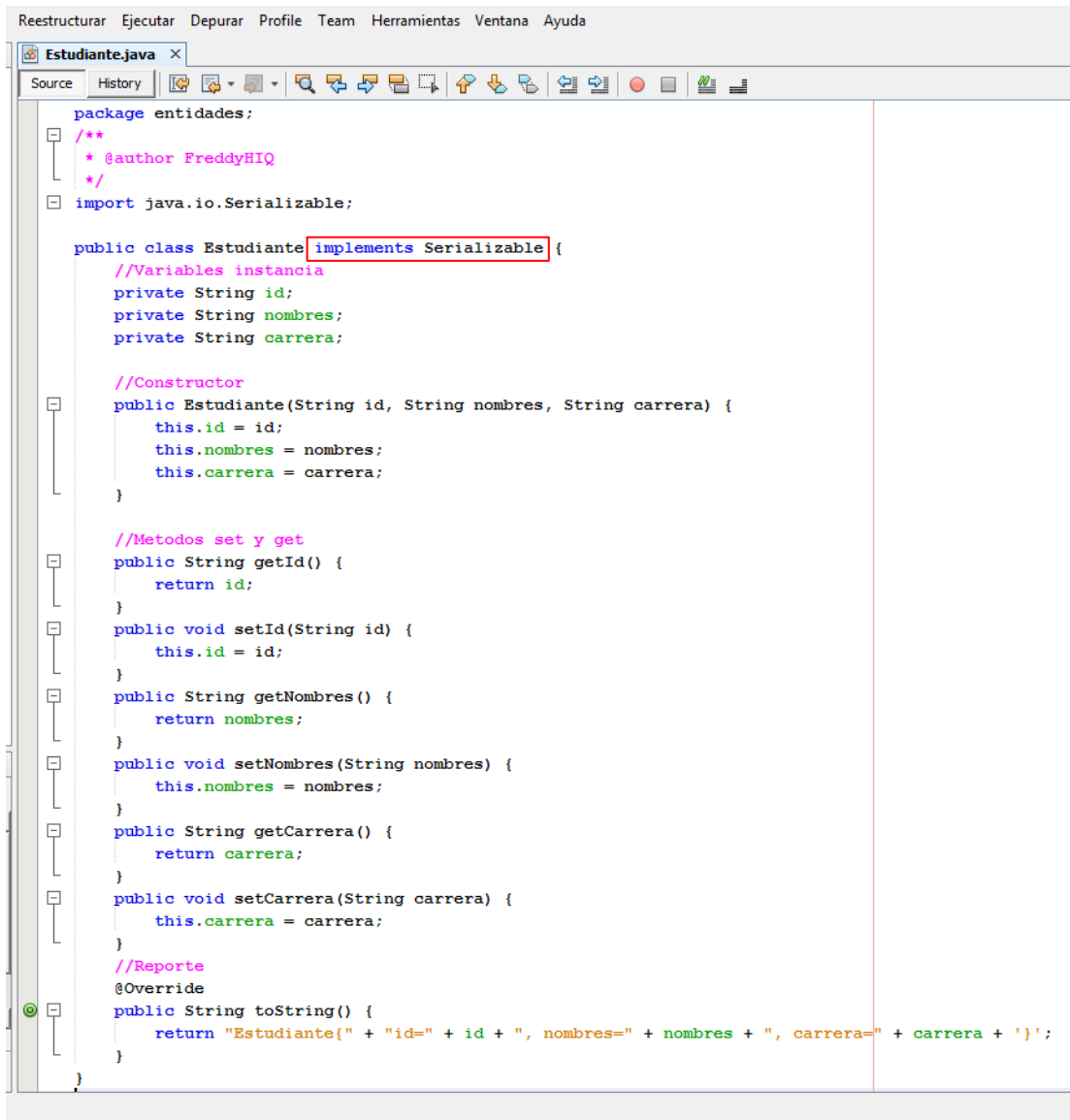


Asignar un nombre de variable a cada **componente** para identificarlo en la programación:

- txtId, txtNombres, cmbcarrera,
- btnBuscar, btnAgregar, btnGuardar, btnSalir, btnReportar

3. IMPLEMENTACIÓN DE CÓDIGO

a. Paquete entidades -> Estudiante



```
package entidades;

/**
 * @author FreddyHIQ
 */
import java.io.Serializable;

public class Estudiante implements Serializable {
    //Variables instancia
    private String id;
    private String nombres;
    private String carrera;

    //Constructor
    public Estudiante(String id, String nombres, String carrera) {
        this.id = id;
        this.nombres = nombres;
        this.carrera = carrera;
    }

    //Metodos set y get
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getNombres() {
        return nombres;
    }
    public void setNombres(String nombres) {
        this.nombres = nombres;
    }
    public String getCarrera() {
        return carrera;
    }
    public void setCarrera(String carrera) {
        this.carrera = carrera;
    }

    //Reporte
    @Override
    public String toString() {
        return "Estudiante{" + "id=" + id + ", nombres=" + nombres + ", carrera=" + carrera + '}';
    }
}
```

b. Paquete accesoDatos -> ListaEstudiantes

```

Regresar Ejecutar Depurar Profile Team Herramientas Ventana Ayuda
ListaEstudiantes.java
Source History

package accesoDatos;
/**
 * @author FreddyIQ
 */
import entidades.Estudiante; //importa a clase Estudiante de otro paquete
import java.io.*; //entrada y salida datos
import java.util.*; //importa colecciones
import javax.swing.JOptionPane;

public class ListaEstudiantes {
    //Declaración de variables de referencia
    Estudiante oE;
    ObjectOutputStream oup = null; //Escritura
    ObjectInputStream inp = null; //Lectura
    ArrayList<Estudiante> lstEst= null; //Coleccion
    File mf = null; //Archivo

    //Constructor
    public ListaEstudiantes() {
        //Se crea la coleccion y se especifica que almacene Estudiantes
        lstEst = new ArrayList<Estudiante>();
        //Crea el fichero de texto
        mf = new File("ArchivoEstudiantes.txt");
    }

    //*****Metodos de Mantenimiento de la Coleccion de Estudiantes*****
    public void agregarEst(Estudiante oE) {
        lstEst.add(oE); //Se agrega a la coleccion
    }

    public Estudiante buscarEstxId(String id){
        Estudiante oE=null;
        //Recorre la coleccion
        for (Estudiante e: lstEst){
            if(id.equals(e.getId()))
                oE = e;
        }
        return oE;
    }

    public Estudiante buscarEstxIndice(int i) {
        return lstEst.get(i); //Retorna el obj Estudiante
    }

    public void eliminarEst(Estudiante oE) {
        lstEst.remove(oE);
    }

    public int obtenerNEst() {
        return lstEst.size();
    }

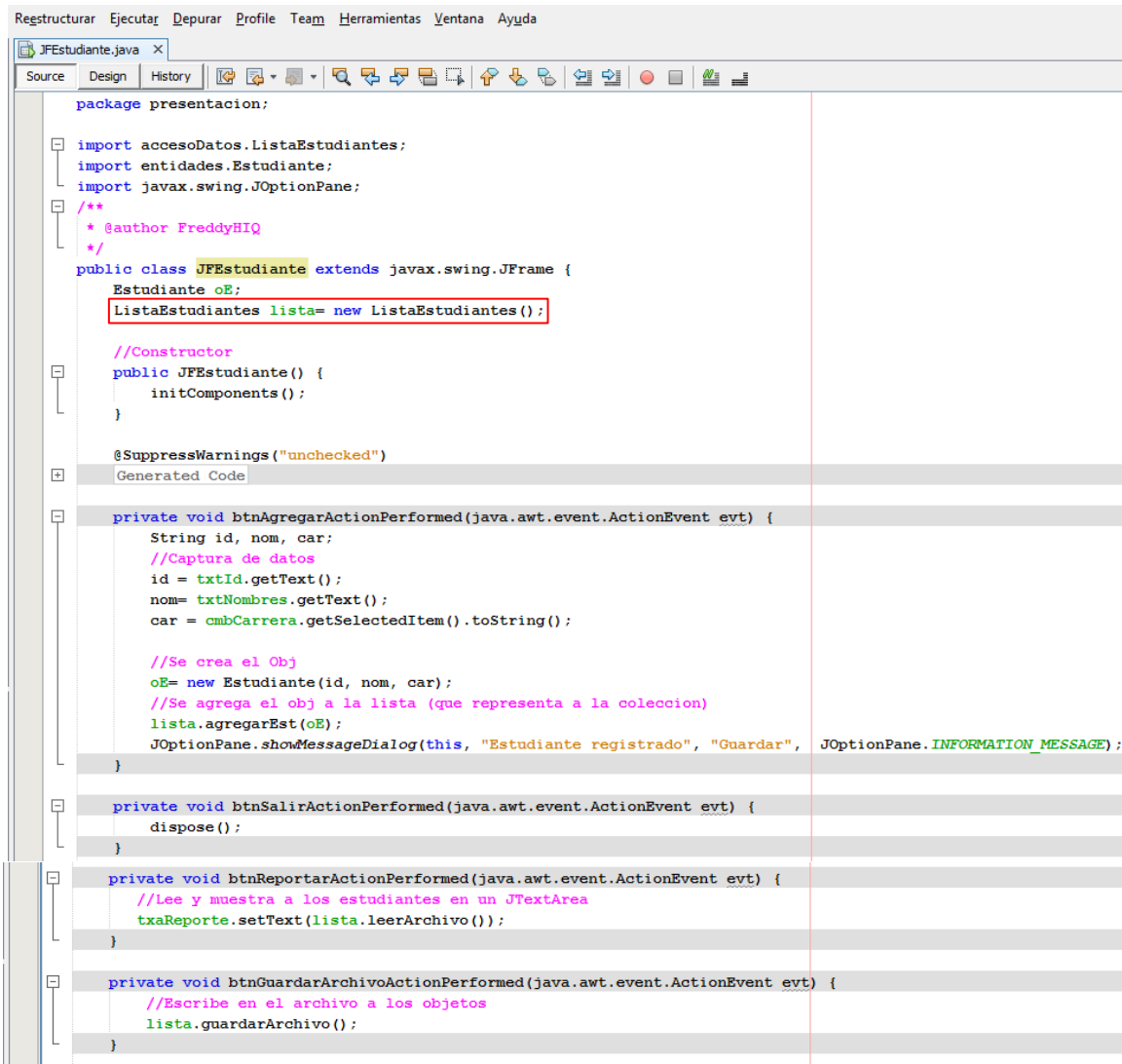
    public void modificarEst(int i, Estudiante oE) {
        lstEst.set(i, oE);
    }

    public String listarEst() {
        String cad="";
        //Recorre la coleccion
        for (Estudiante c: lstEst){
            cad= cad + c.toString() + "\n";
        }
        return cad; //Retorna la informacion de todas las Estudiantes
    }

    //*****Metodos de Acceso al Archivo de texto*****
    public void guardarArchivo() {
        try {
            //Permite la Serializacion
            oup = new ObjectOutputStream(new FileOutputStream(mf));
            //Escribe la coleccion en el archivo
            oup.writeObject(lstEst);
            //Cierra el archivo
            oup.close();
            JOptionPane.showMessageDialog(null, "Se Registró en Archivo", "Guardar Archivo", 1);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Error con serializacion", "Guardar Archivo", 0);
        }
    }

    public String leerArchivo() {
        try {
            //Permite la Deserializacion
            inp = new ObjectInputStream(new FileInputStream(mf));
            //Lee datos del archivo y lo pasa a una coleccion
            lstEst= (ArrayList<Estudiante>)inp.readObject();
            //Cierra el archivo
            inp.close();
        } catch (IOException | ClassNotFoundException ex) {
            JOptionPane.showMessageDialog(null, "Error con archivo", "Guardar Archivo", 0);
        }
        //Llama al método listarEst() y devuelve la lista de estudiantes en cadena
        return listarEst();
    }
}

```

c. Paquete accesoDatos -> JFEstudiante

```
package presentacion;

import accesoDatos.ListaEstudiantes;
import entidades.Estudiante;
import javax.swing.JOptionPane;

/**
 * @author FreddyHIQ
 */
public class JFEstudiante extends javax.swing.JFrame {
    Estudiante oE;
    ListaEstudiantes lista= new ListaEstudiantes();

    //Constructor
    public JFEstudiante() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void btnAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        String id, nom, car;
        //Captura de datos
        id = txtId.getText();
        nom= txtNombres.getText();
        car = cmbCarrera.getSelectedItem().toString();

        //Se crea el Obj
        oE= new Estudiante(id, nom, car);
        //Se agrega el obj a la lista (que representa a la coleccion)
        lista.agregarEst(oE);
        JOptionPane.showMessageDialog(this, "Estudiante registrado", "Guardar", JOptionPane.INFORMATION_MESSAGE);
    }

    private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
        dispose();
    }

    private void btnReportarActionPerformed(java.awt.event.ActionEvent evt) {
        //Lee y muestra a los estudiantes en un JTextArea
        txtaReporte.setText(lista.leerArchivo());
    }

    private void btnGuardarArchivoActionPerformed(java.awt.event.ActionEvent evt) {
        //Escribe en el archivo a los objetos
        lista.guardarArchivo();
    }
}
```

EJERCICIO PROPUESTO

Sobre el ejercicio anterior completar que el sistema permita:

- Buscar Estudiantes por Id.