Manual Técnico

El programa esta codificado 100% en el lengua Java y utilizando principalmente la interfaz grafica de Swing, utilizando Programación Orientada a Objetos y arreglos dinámicos para poder llevar a cabo los objetivos del proyecto.

librerías principalmente usadas:

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

```
import java.io.FileWriter;
import java.io.IOException;
```

```
import java.util.ArrayList;
```

Todos los Arrays utilizados para el manejo de datos durante la ejecución del programa:

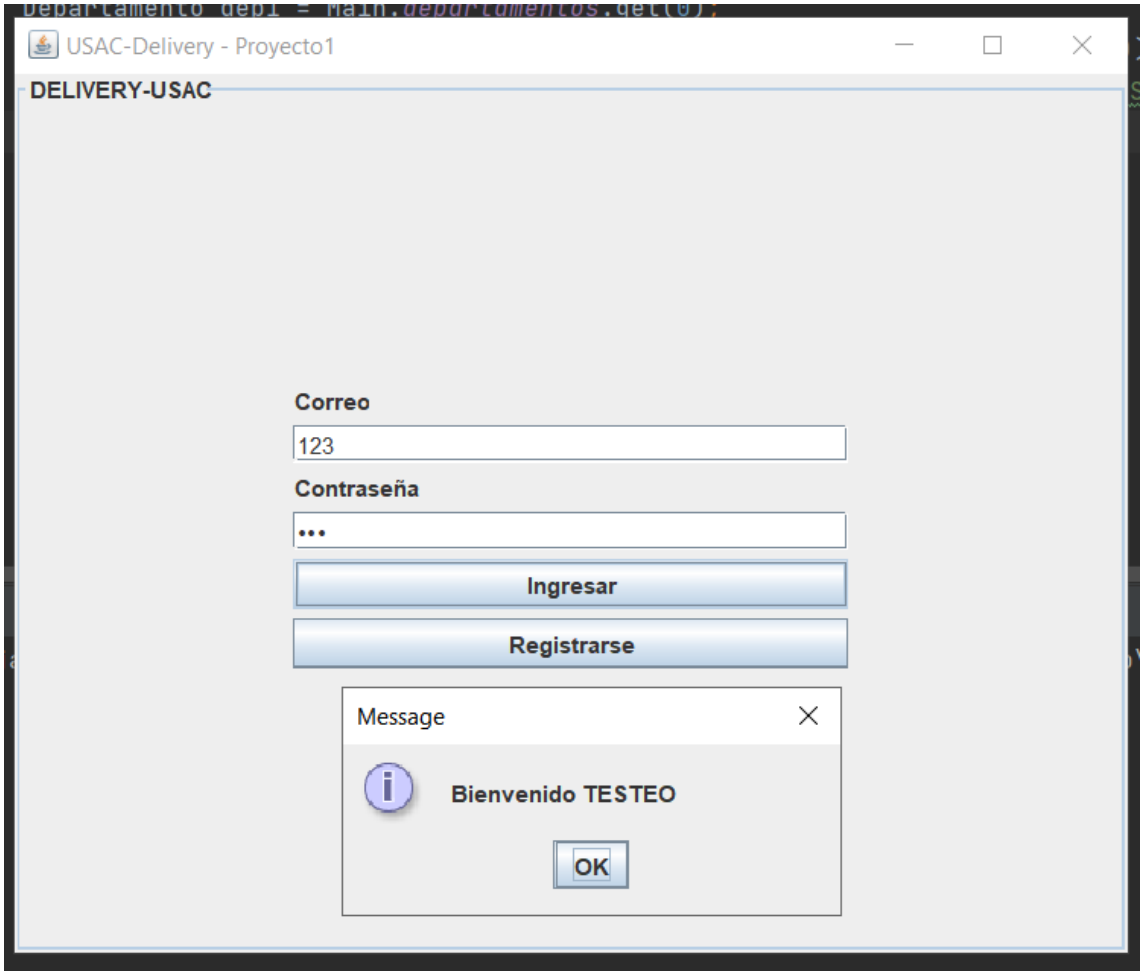
```
14 usages
public static ArrayList<Datos_usuarios> usuariosreg = new ArrayList<>();

public static ArrayList<Regiones> regiones = new ArrayList<>();
17 usages
public static ArrayList<Departamento> departamentos = new ArrayList<>();

public static ArrayList<Municipio> municipios = new ArrayList<>();
7 usages
public static ArrayList<Kiosko> kioskos = new ArrayList<>();
13 usages
public static ArrayList<Credic_Cards> tarjetas = new ArrayList<>();
18 usages
public static ArrayList<D_facturacion> datosfac = new ArrayList<>();
6 usages
public static ArrayList<Ventas> venta = new ArrayList<>();
```

Lista de clases utilizadas para implementar la programación orientada a objetos, utilizando herencia y abstracción .

El programa corre sobre un solo Jframe que es actualizado durante el transcurso para mostrar las diferentes ventanas que muestran toda la interfaz grafica

```
frame = new JFrame( title: "inicio");
frame.setContentPane(new inicio().panelogin);
frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
frame.pack();
frame.setVisible(true);
frame.setSize( width: 600, height: 500);
```

Se cuenta con un usuario especial para el testeo rápido de la funcionalidad del programa siendo el correo y contraseña <123>



Los datos son almacenados en la memoria dinámica utilizando arreglos y solo estan disponibles durante el proceso de ejecución del programa.

```
Main.municipios.add(new Municipio(cod.getCodigo(), Nom_municipio.getText(), cod));
JOptionPane.showMessageDialog(inicio.frame,  message: "Municipio Registrado!");
```

De la clase abstracta Ubicación se heredan sus atributos a las clases de Regiones, departamentos, municipios y kioskos, y a su vez estas funcionan usando herencia principalmente entre cada clase.

```java
public abstract class Ubicacion {

    4 usages
    private String codigo;
    4 usages
    private String nombre;

    4 usages  Luu-chan
    public Ubicacion(String codigo, String nombre) {
        this.codigo = codigo;
        this.nombre = nombre;
    }

    Luu-chan
    public String getCodigo() { return codigo; }

    Luu-chan
    public void setCodigo(String codigo) { this.codigo = codigo; }

    Luu-chan
    public String getNombre() { return nombre; }

    Luu-chan
    public void setNombre(String nombre) { this.nombre = nombre; }



    1 override  Luu-chan
    @Override
    public String toString() { return "Codigo: (" + codigo + ") - Nombre: " + nombre; }
}
```

Después de realizar un envió se puede descargar la factura y guía la cual se guardara en la carpeta raíz en donde se encuentra guardado el proyecto, se guardara en formato .html median el uso de FileWriter y escribiendo el código del archivo de la siguiente forma.

```
FileWriter fileWriter = null;
try {
    // Crear un nuevo archivo .html
    fileWriter = new FileWriter( fileName: "Guia.html");

    // Escribir contenido en el archivo
    fileWriter.write( str: "<html><head><title> Guia.html</title></head><body> \n");

    fileWriter.write( str: "<head> <title> </title> </head>\n" );
    fileWriter.write( str: "<h1>"+"Guia" +"</h1>");
    fileWriter.write( str: "<p>" + "----------------------------------------------------------------------------------------------------"+ "</p>");

    fileWriter.write( str: "<h3>" + "Datos del Cliente"+ "</h3>");

    fileWriter.write( str: "<p>" + "Envia: : " + ve.getDireccion()+ "</p>");
    fileWriter.write( str: "<p>" + "Origen: " + ve.getOrigen()+ "</p>");
    fileWriter.write( str: "<p>" + "Destino: " + ve.getDestino()+ "</p>");


    fileWriter.write( str: "<h3>"+ "Numero de Guia: "+ve.getId_paquete()+"</h3>");
    fileWriter.write( str: "<div>"
            + "<img src=\"cod_barras.svg\" alt=\"Código de barras\">\n"
            + "</div>" );

    fileWriter.write( str: "<p>" + "----------------------------------------------------------------------------------------------------"+ "</p>");
    fileWriter.write( str: "<p>" + "Guarde esto en un lugar seguro al momento de imprimirlo"+ "</p>");
    fileWriter.write( str: "<p>" + "La fecha de envio esta sujeta a cambios"+ "</p>");

    fileWriter.write(     str: "</body></html>");

    fileWriter.close();

    System.out.println("El archivo ha sido creado y escrito con éxito.");
```

Diagrama de Clases del Usuario

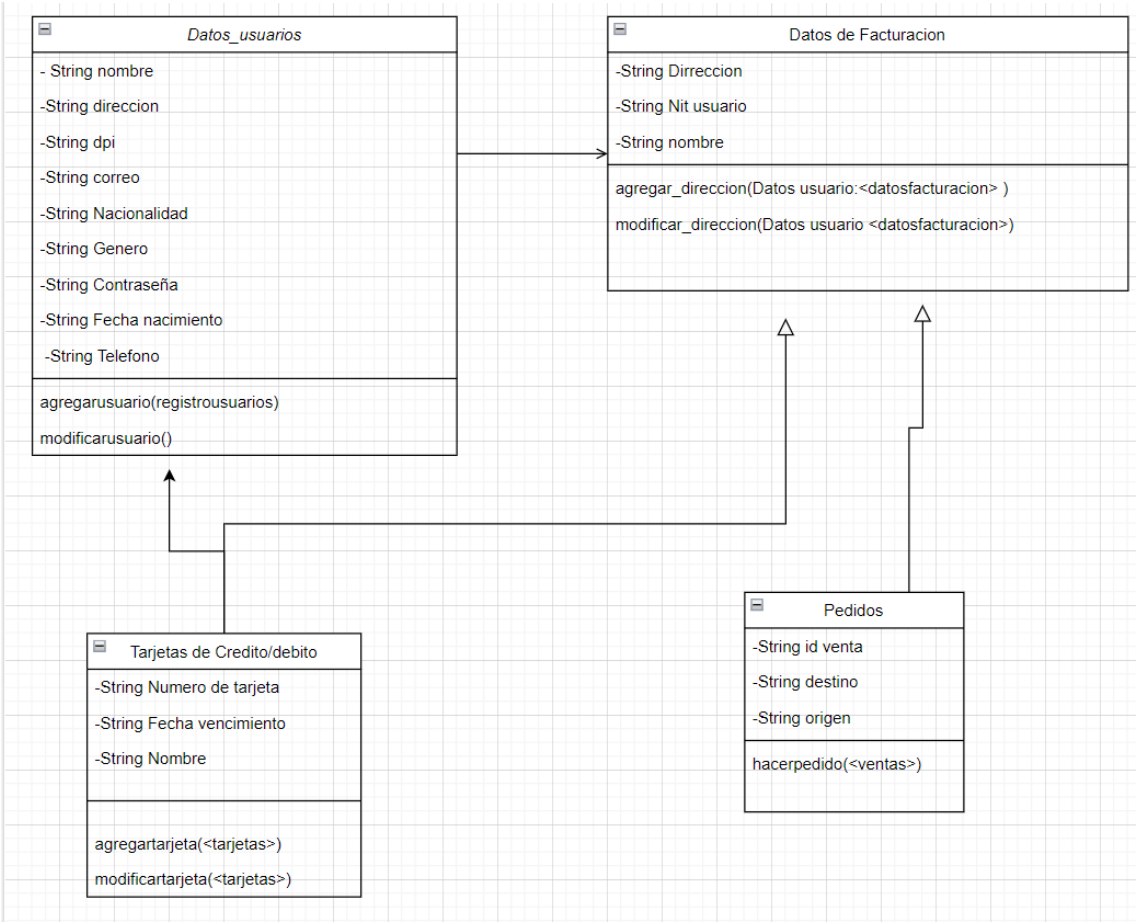


Diagrama de Clases de Ubicaciones