

Manual Técnico

Maria Paola Guadalupe Dávila Valenzuela
Laboratorio de Introducción a la programación y computación
Sección F
11/10/2023

La totalidad del código de este programa ha sido cuidadosamente elaborado en el lenguaje de programación Java. Se ha optado por hacer un extenso uso de la interfaz gráfica Swing, destacando la programación orientada a objetos, la implementación de arreglos dinámicos y el uso de hilos (Threads). Estas elecciones han sido importantes para alcanzar exitosamente los objetivos trazados en este proyecto.

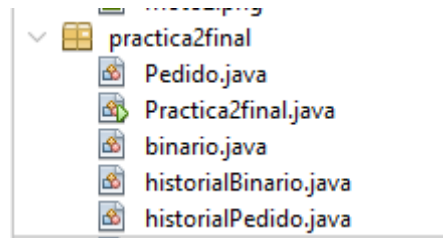
Estas fueron las librerías principalmente utilizadas:

```
5 package practica2final;
6 import java.awt.event.ActionEvent;
7 import java.awt.event.ActionListener;
8 import java.awt.event.MouseAdapter;
9 import java.awt.event.MouseEvent;
10 import javax.swing.text.AbstractDocument;
11 import javax.swing.text.AttributeSet;
12 import javax.swing.text.BadLocationException;
13 import javax.swing.text.DocumentFilter;
14 import java.time.LocalDateTime;
15 import java.time.format.DateTimeFormatter;
16 import java.util.ArrayList;
17 import javax.swing.JLabel;
18 import javax.swing.JOptionPane;
19 import javax.swing.JPanel;
20 import javax.swing.table.DefaultTableModel;
```

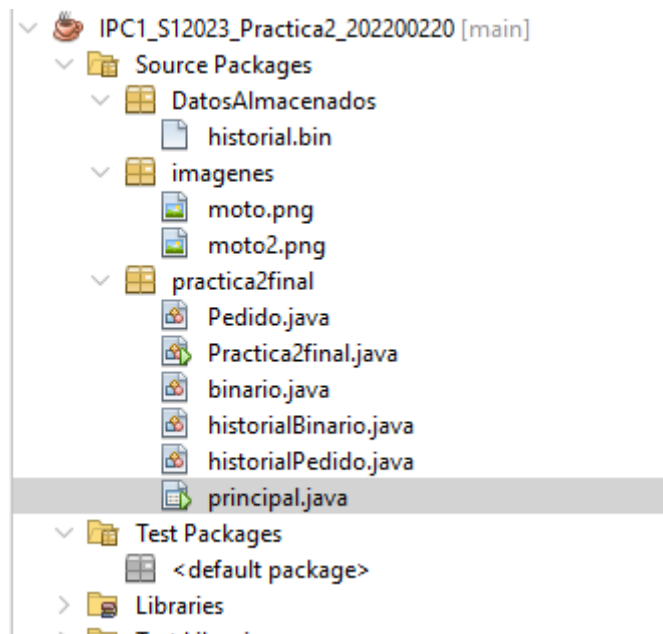
Todos los Arrays utilizados para el manejo de datos durante la ejecución del programa:

```
41 ArrayList<Pedido> ListaPedido = new ArrayList<>();
42 ArrayList<historialPedido> ListaHistorialPedidos = new ArrayList<>();
43
91 historialBinario HistorialBinario = new historialBinario();
92 ArrayList<historialPedido> historialPedido = HistorialBinario.cargarHistorial();
93 ActualizarHistorial();
```

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



El programa se ejecuta a través de varios JFrames que se van actualizando en tiempo real para mostrar las diferentes ventanas y funciones de la interfaz gráfica.



Hay una pestaña de pedidos donde se presenta una tabla de productos asignados y otra donde se crearán las ordenes para posteriormente ver su total y confirmar el pedido.

Los datos se guardan en archivos binarios utilizando arreglos y pueden estar disponibles incluso si el programa deja de ejecutarse.

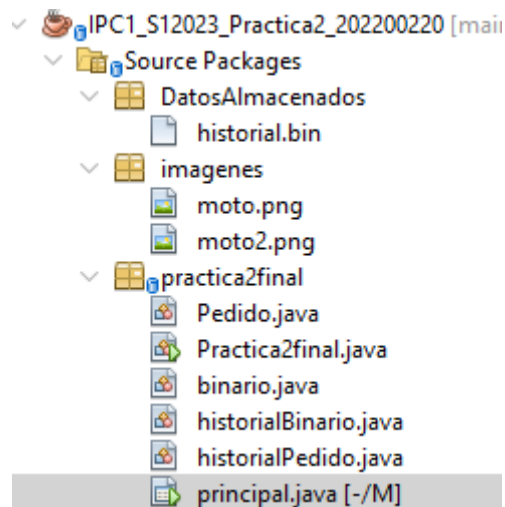
```

122 private void agregarProductoSeleccionado() {
123     int selectedRow = jTable1.getSelectedRow();
124     if (selectedRow >= 0) {
125         String producto = (String) sourceTableModel.getValueAt(row:selectedRow, column: 0);
126         String precioConQ = (String) sourceTableModel.getValueAt(row:selectedRow, column: 1);
127
128         destinationTableModel.addRow(new Object[]{producto, precioConQ});
129
130         double nuevoTotal = 0.0;
131         for (int i = 0; i < destinationTableModel.getRowCount(); i++) {
132             String precioProducto = (String) destinationTableModel.getValueAt(row:i, column: 1);
133             double precio = calcularTotal(precioConQ: precioProducto);
134             nuevoTotal += precio;
135         }
136
137         jTextField2.setText(String.format(format: "%.2f", args:nuevoTotal));
138     } else {
139         JOptionPane.showMessageDialog(parentComponent:this, message:"Selecciona un producto de la tabla de Productos.");
140     }
141 }

```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    EnlistarPedido();  
  
    String selectedVehicle = (String) jComboBox1.getSelectedItem();  
    String kmValue = jTextField1.getText();  
  
        if (selectedVehicle.equals(anObject: "Motocicleta 1")) {  
            jTextField3.setText(t: kmValue);  
        } else if (selectedVehicle.equals(anObject: "Motocicleta 2")) {  
            jTextField4.setText(t: kmValue);  
        } else if (selectedVehicle.equals(anObject: "Motocicleta 3")) {  
            jTextField5.setText(t: kmValue);  
        }  
    historialBinario HistorialBinario = new historialBinario();  
  
    ActualizarHistorial();  
  
}
```

Estos paquetes funcionan principalmente a través de la herencia entre las clases, lo que facilita la colaboración y el flujo de datos entre ellas



```
26 public final class principal extends javax.swing.JFrame {
27
28     private final DefaultTableModel sourceTableModel;
29     private final DefaultTableModel destinationTableModel;
30
31     LocalDateTime FechaActual = LocalDateTime.now();
32     LocalDateTime fechaFinal = FechaActual.plusMinutes(minutes:2);
33
34     DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern:"yyyy-MM-dd HH:mm:ss");
35     String fechaComoString = FechaActual.format(formatter);
36     String fechaFinalString = fechaFinal.format(formatter);
37
38     DefaultTableModel modelo = new DefaultTableModel();
39     DefaultTableModel modeloH = new DefaultTableModel();
40
41     ArrayList<Pedido> ListaPedido = new ArrayList<>();
42     ArrayList<historialPedido> ListaHistorialPedidos = new ArrayList<>();
43 }
```

```
49 public principal() {
50     initComponents();
51     this.setLocationRelativeTo(null);
52
53     sourceTableModel = (DefaultTableModel) jTable1.getModel();
54     destinationTableModel = (DefaultTableModel) jTable2.getModel();
55
56     jTable1.addMouseListener(new MouseAdapter() {
57         public void mouseClicked(MouseEvent e) {
58             int selectedRow = jTable1.getSelectedRow();
59             if (selectedRow >= 0) {
60                 String elemento = (String) sourceTableModel.getValueAt(row:selectedRow, column: 1);
61             }
62         }
63     });
64 }
```

Este programa puede hacer muchas cosas, entre ellas, animaciones con hilos. Los archivos se pueden guardar en la misma carpeta donde está el proyecto, junto con otros archivos similares.

```
712 public void EnviarPrimera() {
713     final int Max_x = 150 + (Integer.parseInt(s: jTextField3.getText()) * 40);
714     final int Min_x = 148;
715
716     Thread animation = new Thread(new Runnable() {
717         @Override
718         public void run() {
719             int x = 150;
720             int y = 75;
721             boolean checked = true;
722
723
724             while (true) {
725                 if (checked) {
726                     motool.setLocation(x, y);
727                     x = x + 10;
728                 } else {
729                     motool.setLocation(x, y);
730                     x = x - 10;
731                 }
732
733                 if (x > Max_x) {
734                     checked = false;
735                     x = 150 + (Integer.parseInt(s: jTextField3.getText()) * 40);
736                     if (x == 150 + (Integer.parseInt(s: jTextField3.getText()) * 40)) {
737                         }
738                 }
739
740                 if (x < Min_x) {
741                     if (x < Min_x) {
742                         x = 150;
743                         y = 75;
744                         jTextField3.setText(s: "");
745                         checked = false;
746                         x = 150 + (Integer.parseInt(s: jTextField3.getText()) * 40);
747                     }
748
749                     try {
750                         Thread.sleep(millis:200);
751                     } catch (Exception e) {
752                         }
753                 }
754             }
755         });
756     animation.start();
757 }
```