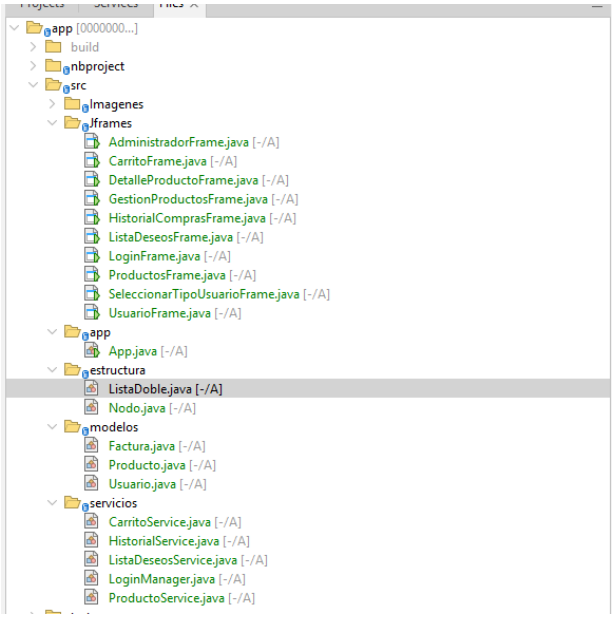
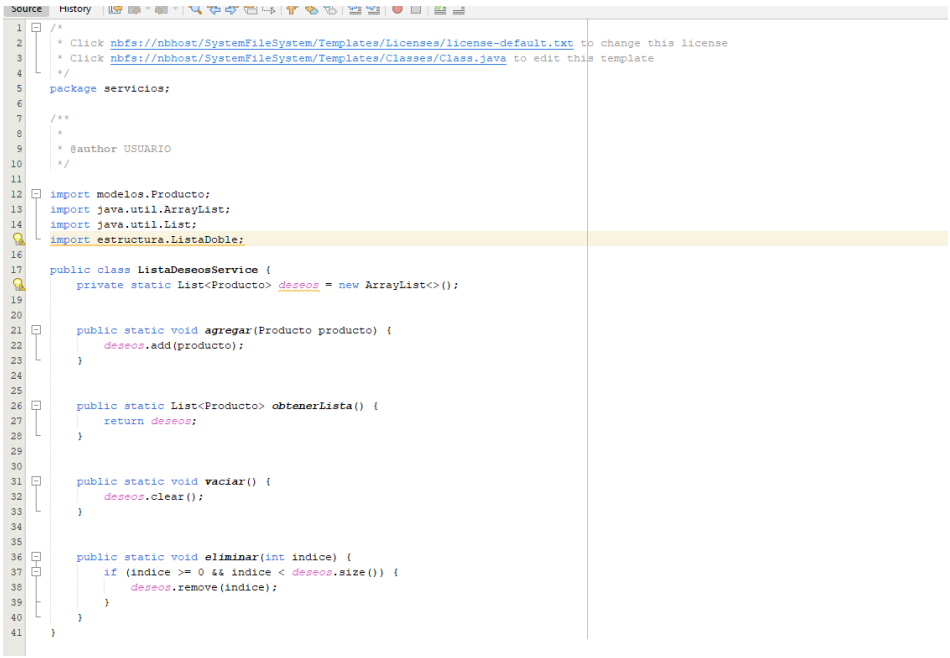
Organización de los paquetes y las clases del proyecto



Paquete Servicios



```
Source  History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package servicios;
6
7  /**
8   *
9   * @author USUARIO
10  */
11
12 import modelos.Producto;
13 import java.util.ArrayList;
14 import java.util.List;
15 import estructura.ListaDoble;
16
17 public class ListaDeseosService {
18     private static List<Producto> deseos = new ArrayList<>();
19
20
21     public static void agregar(Producto producto) {
22         deseos.add(producto);
23     }
24
25
26     public static List<Producto> obtenerLista() {
27         return deseos;
28     }
29
30
31     public static void vaciar() {
32         deseos.clear();
33     }
34
35
36     public static void eliminar(int indice) {
37         if (indice >= 0 && indice < deseos.size()) {
38             deseos.remove(indice);
39         }
40     }
41 }
```

La clase de Lista de deseos, producto y carrito de compra tienen métodos similares los cuales son agregar obtener lista , vaciar con la ligera diferencia que en la clase Producto hay un método que te permite editar.

```java
*/

import estructura.ListaDoble;
import modelos.Producto;
import java.time.LocalDate;
import java.util.ArrayList;
import java.util.List;

public class HistorialService {

    public static class Compra {
        private LocalDate fecha;
        private List<Producto> productos;
        private double total;

        public Compra(List<Producto> productos, double total) {
            this.fecha = LocalDate.now();
            this.productos = new ArrayList<>(productos);
            this.total = total;
        }

        public LocalDate getFecha() {
            return fecha;
        }

        public List<Producto> getProductos() {
            return productos;
        }

        public double getTotal() {
            return total;
        }
    }

    private static ListaDoble<Compra> historial = new ListaDoble<>();

    public static void registrarCompra(List<Producto> productos, double total) {
        historial.agregarAlFinal(new Compra(productos, total));
    }

    public static List<Compra> obtenerHistorial() {
        return historial.aListaJava();
    }

    public static void vaciarHistorial() {
        historial.vaciar();
    }
}
```

Historial de compras tenemos los métodos registrar compra , obtener historial de compras y vaciar historial aquí obtenemos todos los datos como la fecha , productos y total de la clase productos.

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package servicios;

/**
 *
 * @author USUARIO
 */

import modelos.Usuario;
import java.util.ArrayList;
import java.util.List;

public class LoginManager {

    private static final List<Usuario> usuarios = new ArrayList<>();

    static {
        usuarios.add(new Usuario("admin", "admin123"));
        usuarios.add(new Usuario("usuario", "user123"));
    }

    public static boolean validar(String nombre, String clave) {
        for (Usuario u : usuarios) {
            if (u.getUsuario().equals(nombre) && u.getContraseña().equals(clave)) {
                return true;
            }
        }
        return false;
    }

    public static boolean esAdmin(String nombre) {
        return nombre.equalsIgnoreCase("admin");
    }
}
```

La clase Login manager en esta usamos arraylist ya que es una actividad sencilla el de listar las contraseñas para admin y user. Usamos un boolean.

Paquete modelos

Este paquete tiene como función definir las estructuras de datos del sistema clases como usuario , factura y producto contienen los atributos , contructores y métodos básicos (getters/setters) que permiten almacenar, manipular y transferir información entre la lógica del programa y la interfaz gráfica.

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package modelos;

/**
 *
 * @author USUARIO
 */
public class Usuario {
    private String usuario;
    private String contraseña;

    public Usuario(String usuario, String contraseña) {
        this.usuario = usuario;
        this.contraseña = contraseña;
    }

    public String getUsuario() {
        return usuario;
    }

    public String getContraseña() {
        return contraseña;
    }
}
```

```
 * @author USUARIO
 */
import javax.swing.*;

public class Producto {
    private String nombre;
    private String precio;
    private String rutaImagen;
    private String descripcion;
    private int stock;

    public Producto(String nombre, String precio, String rutaImagen, String descripcion, int stock) {
        this.nombre = nombre;
        this.precio = precio;
        this.rutaImagen = rutaImagen;
        this.descripcion = descripcion;
        this.stock = stock;
    }


    public String getNombre() {
        return nombre;
    }

    public String getPrecio() {
        return precio;
    }

    public String getRutaImagen() {
        return rutaImagen;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public int getStock() {
        return stock;
    }


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

```
package modelos;

import java.util.Date;
import java.util.List;



public class Factura {
    private Date fecha;
    private double total;
    private List<Producto> productos;

    public Factura(Date fecha, double total, List<Producto> productos) {
        this.fecha = fecha;
        this.total = total;
        this.productos = productos;
    }

    public Date getFecha() {
        return fecha;
    }

    public double getTotal() {
        return total;
    }

    public List<Producto> getProductos() {
        return productos;
    }
}
```

Paquete estructura

Este se encarga de definir las estructuras de datos utilizadas en el sistema, como la lista doblemente enlazada y sus nodos. Estas estructuras permiten un mejor control sobre la organización, acceso y manipulación de los datos, haciendo operaciones como inserción, eliminación y recorrido eficiente de los elementos almacenados.

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

public class ListaDoble<T> {
    private Nodo<T> cabeza;
    private Nodo<T> cola;
    private int tamanio;

    public ListaDoble() {
        cabeza = null;
        cola = null;
        tamanio = 0;
    }

    public void agregarAlFinal(T dato) {
        Nodo<T> nuevo = new Nodo<>(dato);
        if (cabeza == null) {
            cabeza = cola = nuevo;
        } else {
            cola.siguiente = nuevo;
            nuevo.anterior = cola;
            cola = nuevo;
        }
        tamanio++;
    }

    public void eliminarPorIndice(int index) {
        if (index < 0 || index >= tamanio) return;
        Nodo<T> actual = cabeza;
        for (int i = 0; i < index; i++) {
            actual = actual.siguiente;
        }
        if (actual.anterior != null) actual.anterior.siguiente = actual.siguiente;
        else cabeza = actual.siguiente;

        if (actual.siguiente != null) actual.siguiente.anterior = actual.anterior;
        else cola = actual.anterior;

        tamanio--;
    }

    public T obtener(int index) {
        if (index < 0 || index >= tamanio) return null;
        Nodo<T> actual = cabeza;
        for (int i = 0; i < index; i++) {
            actual = actual.siguiente;
        }
        return actual.dato;
    }

    public int getTamanio() {
        return tamanio;
    }
```

```java
public T obtener(int index) {
    if (index < 0 || index >= tamanio) return null;
    Nodo<T> actual = cabeza;
    for (int i = 0; i < index; i++) {
        actual = actual.siguiente;
    }
    return actual.dato;
}

public int getTamanio() {
    return tamanio;
}

public void vaciar() {
    cabeza = cola = null;
    tamanio = 0;
}

public List<T> aListaJava() {
    List<T> lista = new ArrayList<>();
    Nodo<T> actual = cabeza;
    while (actual != null) {
        lista.add(actual.dato);
        actual = actual.siguiente;
    }
    return lista;
}
```