

**CARRERA DE INGENIERIA DE SOFTWARE**

**PROGRAMACION ORIENTADA A OBJETOS**

**NOMBRE:** Sebastian Alvarez

**NRC:** 4024

**FECHA:** 18/07/2019

**INFORME DE “TallerMecanicoApp”**

1. **Problema**

La facturación de un taller mecánico requiere de un conjunto de clases que modelen las piezas que se utilizan en una reparación. Sobre cada PIEZA se debe reflejar su número de pieza, su nombre, su precio y el tiempo de mano de obra que se facturará por su instalación.

Para ciertas reparaciones se utilizan conjuntos de piezas denominados «kits» (aceite, tapa y filtro de aceite para un cambio de aceite). Los «kits» tienen su propio número, nombre y tiempo de mano de obra, pero su precio resulta de la suma de los precios de las piezas que lo componen menos un 10 % de descuento.

Considerando que nombre y número de pieza son muy estables, el precio de las piezas se actualiza con frecuencia y que las piezas que componen un «kit» pueden a su vez set «kits», elaborar el diagrama de clases de la solución propuesta para esta situación e implementar totalmente las clases que modelan los tipos de piezas.

Se pide:

* Realizar el patrón MVC
* Aplicar el patrón de diseño que crea conveniente para lograr un aplicativo efectivo

1. **Solución**

Para brindar una solución al problema se ha implementado el programa “TallerMecanicoApp”, el cual tiene las siguientes características:

* Consta de siete clases (Java Class), cuatro interfaces graficas (Formulario JFrame) y una clase principal (Java Main Class).
* El programa ha sido elaborado de tal manera que cumple con los principios SOLID y con dos patrones de diseño; Vista-Controlador y Fachada.
* Utiliza gestión de bases de datos a través de MySql con el sistema local WorkBrench con el uso de un esquema llamado “tallermecanico” y una tabla llamada “pieza”.
* Utiliza gestión de archivos planos para guardar datos de todas las compras a través del archivo “archivoCompras.txt”.
* Se ha añadido al proyecto la librería de tipo JAR “mysql-connector-java-8.0.16.jar”.

1. **Descripción**

El programa “TallerMecanicoApp” facilita la gestión de un taller mecánico en donde se tiene que vender piezas y/o kits, para lo cual se ha elaborado una interfaz menú que posibilita el uso del patrón de diseño “fachada”, el cual consta de cuatro opciones a elegir “Ingresar piezas”, “Vender pieza”, “Vender kit” y “Salir”, que direccionan a las diferentes interfaces del sistema.

Al presionar en “Ingresar piezas” se despliega una interfaz gráfica que permite ingresar, modificar, eliminar o buscar piezas, las cuales serán ingresadas o modificadas de la base de datos “tallermecanico” a través del controlador respectivo llamado “ControladorTaller”.

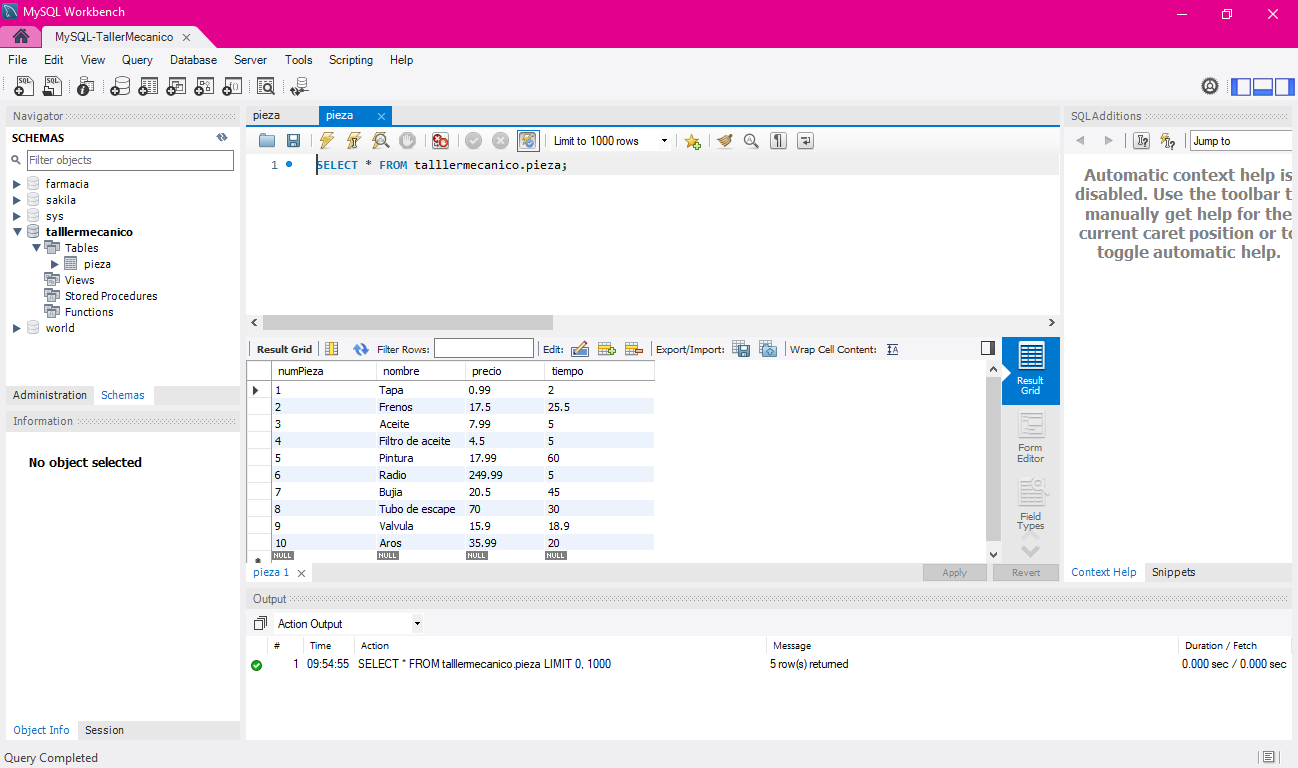
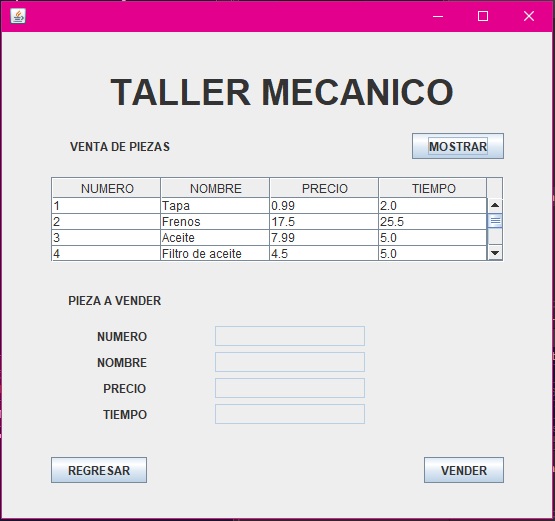
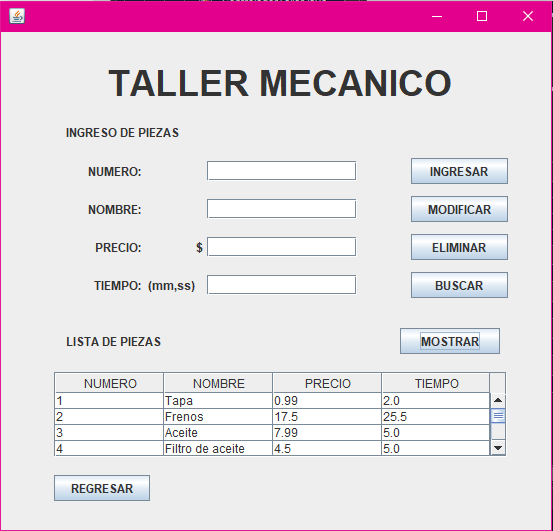
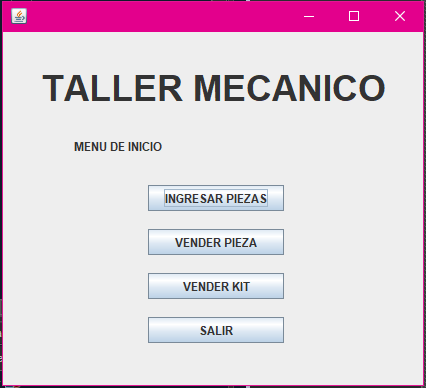
Al presionar en “Vender Pieza” se despliega una interfaz gráfica que permite seleccionar una pieza de una tabla que consta con las piezas registradas en la base de datos, al seleccionar una pieza se despliegan sus datos en txtlabels, al ser vendida la pieza se guardan sus datos en el archivo plano “archivoCompras.txt” y se elimina de la base de datos.

Al presionar en “Vender kit” se despliega una interfaz gráfica que permite seleccionar algunas piezas de una tabla, añadirle un nombre, numero, tiempo de instalación al kit por medio de txtlabels y el precio del kit se calcula automáticamente siendo la suma de los precios de las piezas seleccionadas menos un 10%, al venderse el kit se guarda su información en el archivo plano “archivoCompras.txt” y todas las piezas seleccionadas se eliminan de la base de datos.

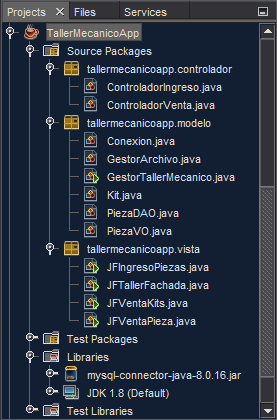
Al presionar “Salir” termina la ejecución del programa.

1. **Ejecución**

A continuación se presentan capturas de las interfaces gráficas y del registro en la base de datos para cada opción del menú principal.



1. **Visión del Proyecto**



1. **Código**

Debido a que existen 12 clases en el programa y copiar todo el código de las mismas seria exagerado, se ha escogido a una clase de cada package (Controlador, Modelo, Vista):

* ControladorVenta
* PiezaDAO
* JFVentaKits

**Clase ControladorVenta:**

package tallermecanicoapp.controlador;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.IOException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.DefaultListModel;

import javax.swing.ListModel;

import javax.swing.table.DefaultTableModel;

import tallermecanicoapp.modelo.\*;

import tallermecanicoapp.vista.JFVentaKits;

import tallermecanicoapp.vista.JFVentaPieza;

public class ControladorVenta implements ActionListener{

JFVentaPieza objVistaPieza = new JFVentaPieza();

JFVentaKits objVistaKit = new JFVentaKits();

PiezaVO objVO = new PiezaVO();

PiezaDAO objDAO = new PiezaDAO();

public ControladorVenta(JFVentaPieza objVistaPieza) {

this.objVistaPieza = objVistaPieza;

this.objVistaPieza.btnVender.addActionListener(this);

this.objVistaPieza.btnMostrar.addActionListener(this);

}

public ControladorVenta(JFVentaKits objVistaKit) {

this.objVistaKit = objVistaKit;

this.objVistaKit.btnVender.addActionListener(this);

this.objVistaKit.btnMostrar.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource() == objVistaPieza.btnVender){

int codigo = Integer.parseInt(objVistaPieza.txtNumero.getText());

String nombre = objVistaPieza.txtNombre.getText();

double precio = Double.parseDouble(objVistaPieza.txtPrecio.getText());

double tiempo = Double.parseDouble(objVistaPieza.txtTiempo.getText());

PiezaVO auxVO = new PiezaVO(codigo,nombre,precio,tiempo);

GestorArchivo ga = new GestorArchivo();

try {

ga.crearArchivo(auxVO);

} catch (IOException ex) {

Logger.getLogger(ControladorVenta.class.getName()).log(Level.SEVERE, null, ex);

}

objDAO.eliminarPieza(codigo);

llenarTablaPieza();

}else if(e.getSource() == objVistaKit.btnVender){

ArrayList<PiezaVO> arrPiezas = new ArrayList<>();

ListModel modelo = objVistaKit.listaCompra.getModel();

for(int i=0; i<modelo.getSize();i++){

String aux = (String)modelo.getElementAt(i);

String [] contacto = aux.split("/");

int numero = Integer.parseInt(contacto[0]);

String nombre = contacto[1];

double precio = Double.parseDouble(contacto[2]);

double tiempo = Double.parseDouble(contacto[3]);

PiezaVO auxVO = new PiezaVO(numero,nombre,precio,tiempo);

arrPiezas.add(auxVO);

}

Kit kit = new Kit();

kit.kit = arrPiezas;

kit.setNumKit(Integer.parseInt(objVistaKit.txtNumero.getText()));

kit.setNombre(objVistaKit.txtNombre.getText());

kit.setPrecio(Double.parseDouble(objVistaKit.txtPrecio.getText()));

kit.setTiempo(Double.parseDouble(objVistaKit.txtTiempo.getText()));

GestorArchivo ga = new GestorArchivo();

try {

ga.crearArchivo(kit);

} catch (IOException ex) {

Logger.getLogger(ControladorVenta.class.getName()).log(Level.SEVERE, null, ex);

}

vaciarCampos();

}else if(e.getSource() == objVistaPieza.btnMostrar){

llenarTablaPieza();

}else if(e.getSource() == objVistaKit.btnMostrar){

llenarTablaKit();

}

}

public void llenarTablaPieza(){

ArrayList<PiezaVO> arr = objDAO.mostrarPiezas();

DefaultTableModel modelo = new DefaultTableModel();

objVistaPieza.TablaPiezas.setModel(modelo);

modelo.addColumn("NUMERO");

modelo.addColumn("NOMBRE");

modelo.addColumn("PRECIO");

modelo.addColumn("TIEMPO");

Object[] columna = new Object[4];

for( int i=0; i<arr.size();i++){

columna[0] = arr.get(i).getNumPieza();

columna[1] = arr.get(i).getNombre();

columna[2] = arr.get(i).getPrecio();

columna[3] = arr.get(i).getTiempo();

modelo.addRow(columna);

}

}

public void llenarTablaKit(){

ArrayList<PiezaVO> arr = objDAO.mostrarPiezas();

DefaultTableModel modelo = new DefaultTableModel();

objVistaKit.TablaPiezas.setModel(modelo);

modelo.addColumn("NUMERO");

modelo.addColumn("NOMBRE");

modelo.addColumn("PRECIO");

modelo.addColumn("TIEMPO");

Object[] columna = new Object[4];

for( int i=0; i<arr.size();i++){

columna[0] = arr.get(i).getNumPieza();

columna[1] = arr.get(i).getNombre();

columna[2] = arr.get(i).getPrecio();

columna[3] = arr.get(i).getTiempo();

modelo.addRow(columna);

}

}

public void vaciarCampos(){

objVistaKit.txtNombre.setText("");

objVistaKit.txtNumero.setText("");

objVistaKit.txtPrecio.setText("");

objVistaKit.txtTiempo.setText("");

DefaultListModel listModel = new DefaultListModel();

objVistaKit.listaCompra.setModel(listModel);

}

}

**Clase PiezaDAO:**

package tallermecanicoapp.modelo;

import java.sql.\*;

import java.util.ArrayList;

public class PiezaDAO{

Conexion objConex = new Conexion();

public void insertarPieza(PiezaVO pieza){

Connection acceso = objConex.obtenerConexion();

String sql = "INSERT INTO pieza (numPieza,nombre,precio,tiempo) VALUES(?,?,?,?)";

try{

PreparedStatement ps = acceso.prepareStatement(sql);

ps.setInt(1, pieza.getNumPieza());

ps.setString(2, pieza.getNombre());

ps.setDouble(3, pieza.getPrecio());

ps.setDouble(4, pieza.getTiempo());

ps.executeUpdate();

}catch(SQLException e){

System.out.println(e);

}

}

public ArrayList<PiezaVO> mostrarPiezas(){

ArrayList<PiezaVO> arr = new ArrayList<>();

PiezaVO pieza;

try{

Connection acces = objConex.obtenerConexion();

PreparedStatement ps = acces.prepareStatement("SELECT \* from pieza");

ResultSet rs = ps.executeQuery();

while(rs.next()){

pieza = new PiezaVO();

pieza.setNumPieza(rs.getInt(1));

pieza.setNombre(rs.getString(2));

pieza.setPrecio(rs.getDouble(3));

pieza.setTiempo(rs.getDouble(4));

arr.add(pieza);

}

}catch(SQLException e){

System.out.println(e);

}

return arr;

}

public void modificarPieza(PiezaVO pieza){

Connection acceso = objConex.obtenerConexion();

String consulta="UPDATE pieza SET numPieza= ? ,nombre= ? , precio= ? , tiempo= ? WHERE numPieza= ? ";

try{

PreparedStatement ps = acceso.prepareStatement(consulta);

ps.setInt(1, pieza.getNumPieza());

ps.setString(2, pieza.getNombre());

ps.setDouble(3, pieza.getPrecio());

ps.setDouble(4, pieza.getTiempo());

ps.setInt(5, pieza.getNumPieza());

ps.executeUpdate();

}catch(SQLException e){

System.out.println(e);

}

}

public void eliminarPieza(int codigo){

try{

Statement estatuto = objConex.obtenerConexion().createStatement();

estatuto.executeUpdate("DELETE FROM pieza WHERE numPieza='"+codigo+"'");

estatuto.close();

}catch(SQLException e){

System.out.println(e);

}

}

public PiezaVO buscarPieza(int codigo){

String consulta="SELECT \* FROM pieza WHERE numPieza = '"+codigo+"'";

PiezaVO objVO = new PiezaVO();

try{

Statement ps = objConex.obtenerConexion().createStatement();

ResultSet rs = ps.executeQuery(consulta);

while(rs.next()){

objVO.setNumPieza(rs.getInt("numPieza"));

objVO.setNombre(rs.getString("nombre"));

objVO.setPrecio(rs.getDouble("precio"));

objVO.setTiempo(rs.getDouble("tiempo"));

}

}catch(SQLException e){

System.out.println(e);

}

return objVO;

}

}

**Clase JFVentaKits:**

package tallermecanicoapp.vista;

import java.util.ArrayList;

import javax.swing.DefaultListModel;

import javax.swing.table.DefaultTableModel;

import tallermecanicoapp.modelo.PiezaDAO;

import tallermecanicoapp.modelo.PiezaVO;

/\*\*

\*

\* @author SEBASTIAN ALVAREZ

\*/

public class JFVentaKits extends javax.swing.JFrame {

ArrayList<PiezaVO> arrVO = new ArrayList<>();

public JFVentaKits() {

initComponents();

this.setLocationRelativeTo(null);

DefaultTableModel modelo = new DefaultTableModel();

this.TablaPiezas.setModel(modelo);

modelo.addColumn("NUMERO");

modelo.addColumn("NOMBRE");

modelo.addColumn("PRECIO");

modelo.addColumn("TIEMPO");

this.txtPrecio.setEditable(false);

}

private void btnRegresarActionPerformed(java.awt.event.ActionEvent evt) {

JFTallerFachada inicio = new JFTallerFachada();

inicio.setVisible(true);

this.setVisible (false);

}

private void btnVenderActionPerformed(java.awt.event.ActionEvent evt) {

vaciarCampos();

}

private void actAñadirActionPerformed(java.awt.event.ActionEvent evt) {

PiezaDAO objDAO = new PiezaDAO();

int indice = this.TablaPiezas.getSelectedRow();

ArrayList<PiezaVO> arr = objDAO.mostrarPiezas();

PiezaVO aux = arr.get(indice);

arrVO.add(aux);

llenarLista(arrVO);

this.txtPrecio.setText(Double.toString(calcularPrecio()));

objDAO.eliminarPieza(aux.getNumPieza());

llenarTablaPieza();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(JFVentaKits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(JFVentaKits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(JFVentaKits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(JFVentaKits.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(() -> {

new JFVentaKits().setVisible(true);

});

}

public void vaciarCampos(){

this.txtNombre.setText("");

this.txtNumero.setText("");

this.txtPrecio.setText("");

this.txtTiempo.setText("");

}

public double calcularPrecio(){

double result = 0;

for(PiezaVO emp : arrVO){

result += emp.getPrecio();

}

return result - result\*0.1;

}

public void llenarLista(ArrayList<PiezaVO> arrVO){

DefaultListModel listModel = new DefaultListModel();

for(int i=0; i<arrVO.size(); i++) {

String aux = arrVO.get(i).getNumPieza() + "/" + arrVO.get(i).getNombre() +"/" + arrVO.get(i).getPrecio()

+"/" +arrVO.get(i).getTiempo();

listModel.add(i, aux);

}

this.listaCompra.setModel(listModel);

}

public void llenarTablaPieza(){

PiezaDAO objDAO = new PiezaDAO();

ArrayList<PiezaVO> arr = objDAO.mostrarPiezas();

DefaultTableModel modelo = new DefaultTableModel();

this.TablaPiezas.setModel(modelo);

modelo.addColumn("NUMERO");

modelo.addColumn("NOMBRE");

modelo.addColumn("PRECIO");

modelo.addColumn("TIEMPO");

Object[] columna = new Object[4];

for( int i=0; i<arr.size();i++){

columna[0] = arr.get(i).getNumPieza();

columna[1] = arr.get(i).getNombre();

columna[2] = arr.get(i).getPrecio();

columna[3] = arr.get(i).getTiempo();

modelo.addRow(columna);

}

}