

**INSTITUTO SUPERIOR TECNOLOGICO PRIVADO CIBERTEC**

**CARRERAS PROFESIONALES**

**SISTEMA DE VENTA DE DISCOS DUROS**

**INTRODUCCION A LA ALGORITMIA**

**Profesor:**

**Carlos Rolando Mendo Bravo**

**PRIMER CICLO**

**SECCIÓN T1BM**

**SEMESTRE 2021 – 01**

**Coordinador:**

**- Bryan Enrique Grados Robles**

**Integrantes:**

**- Jhan Pier Teran Luna**

**- Sebastián Espinoza Espinoza**

**- Gustavo Marcelo Romero Bedoya**

**CARRERAS PROFESIONALES**

**CIBERTEC**

**MIRAFLORES-BREÑA, 2021**

#### ÍNDICE:

|  |  |
| --- | --- |
| Introducción |  |
| Justificación |  |
| Objetivos |  |
| Definición |  |
| Entregables |  |
| Conclusiones |  |
| Recomendaciones |  |
| Bibliografía |  |

**CARRERAS PROFESIONALES**

**CIBERTEC**

**MIRAFLORES-BREÑA, 2021**

#### Introducción

El Presente proyecto fue desarrollado como herramienta de trabajo para aquellos pequeños negocios o aquellos que recién comienzan, ya que al tener desconocimiento en este ámbito empresarial puede causar perdidas por la mala contabilidad ya que con esta herramienta va a permitir de una manera más efectiva y precisa realizar las ventas, generar reportes y poder modificar los productos, limitándose al almacenamiento de la información solo hasta el cierre del programa.

#### Justificación

El Programa va a permitir una transacción con mayor velocidad y sin error de cálculo o registro, permitiendo que el flujo de ventas sea mayor y a la vez la satisfacción de los consumidores con la empresa. Además de un cambio rápido de la información de los productos, ya sea tamaño, precio, modelos y capacidad al igual se va a poder generar un reporte de las ventas realizadas en el día para mantener una supervisión constante en las ventas

* Los beneficiarios directos:

Cliente

* Los Beneficiarios indirectos:

Empresa de disco duros

#### Objetivos

* Mejorar la Velocidad de ventas y cambios en los productos en un 50% en un plazo de una semana.

* Elevar las ventas a travéz de la satisfacción de los clientes y ofertas en productos con un mayor número de ventas según los reportes del programa, para alcanzar un aumento del 20% en el plazo de un mes.

#### Definición

• Mantenimiento: Consultar Disco Duro

Esta ventana permite consultar el precio, modelo, ancho, alto, fondo y capacidad de cada disco duro según el modelo que seleccionemos e la lista despegable de marca, el botón cerrar va a permitir el cierre de la ventana. Los datos de los discos duros en esta ventana no son editables.

Aplicación

Descripción generada automáticamente con confianza baja

• Mantenimiento: Modificar Disco Duro

Esta ventana permite cambiar los valores de precio, modelo, ancho, alto, fondo y capacidad de cada disco duro cambiando los datos de manera global referente a cada disco duro, se puede cambiar la modelo que desea al editar a través de la lista despegable.

Al presionar el botón grabar se va a realizar los cambios de las variables modificadas con respecto a la marca seleccionada y se procede al cerrado de la ventana.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

• Mantenimiento: Listar Disco Duro

Esta Ventana muestra la lista de disco duros que contiene el programa, asi como la modelo precio, alto, ancho, profundidad y modelo de cada disco duro, esta lista cambia de acuerdo a lo realizado en la ventana modificar disco duro.

Interfaz de usuario gráfica

Descripción generada automáticamente

• Ventas: Vender

Esta ventana la utilizamos para vender los productos seleccionando la marca a vender ya que el precio aparece de manera automática con respecto a la marca seleccionada, se ingresa la cantidad y se procede a presionar el botón vender luego de validar el numero aparece una pequeña ventana anunciando la venta exitosa que cerrarse aparecerá una boleta de venta en la parte inferior de la ventana de ventas señalando el modelo del disco duro, precio unitario, cantidad de unidades adquiridas, importe de compra, importe descuento, obsequio, importe a pagar y correspondiente por la cantidad de cada modelo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

• Ventas: Generar Reportes

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteEsta ventana nos muestra las ventas que se han realizado por modelo, mostrando el modelo, la cantidad de ventas realizadas, la cantidad de unidades vendidas, el importe total vendido y la cuota optima. Además, se pueden mostrar reportes de marcas que han alcanzado una venta optima, la relación al precio promedio, además del precio promedio, mayor y menor.

• Configuración: Porcentajes de Descuento

En esta ventana se puede modificar los porcentajes de descuento que son aplicados a las ventas según el número de unidades compras. Luego de presionar el botón aceptar automáticamente se cambian las variables y se cierra la ventana.

Interfaz de usuario gráfica

Descripción generada automáticamente

• Configuración: Obsequio

En configurar obsequio se cambia la cantidad mínima de productos para recibir un obsequio y de igual forma el obsequio a recibir, luego de presionar el botón aceptar se procede a la validación de los datos ingresados y se realiza el cambio de ser valido, posteriormente se cierra la ventana automáticamente, de ser invalido algún dato aparece una ventana de advertencia.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

• Configuración: Cantidad Óptima

En ventana permite cambiar la cantidad optima a vender de cada producto, de igual forma se realiza una validación al presionar el botón aceptar y luego se procede al cerrado automático de resultar invalido algún dato se pasará a notificar al usuario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

• Configuración: Cuota Diaria

En ventana permite establecer una cuota diaria de ventas y poder obtener una mejor estadística en tus ventas, de igual forma se realiza una validación al presionar el botón aceptar y luego se procede al cerrado automático de resultar invalido algún dato se pasará a notificar al usuario.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

• Ayuda: Acerca de Tienda

Esta ventana se muestra el nombre del programa, la versión, los autores y luego el botón cerrar.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### Entregables

**Guis Proyecto:**

public class Proyecto extends JFrame implements ActionListener {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JMenuBar menuTienda;

private JMenu mnArchivo;

private JMenu mnMantenimiento;

private JMenu mnVentas;

private JMenu mnConfiguracion;

private JMenu mnAyuda;

private JMenuItem mntmSalir;

private JMenuItem mntmConsultarDisco;

private JMenuItem mntmModificarDisco;

private JMenuItem mntmListarDisco;

private JMenuItem mntmVender;

private JMenuItem mntmGenerarReportes;

private JMenuItem mntmConfigurarDescuentos;

private JMenuItem mntmConfigurarObsequio;

private JMenuItem mntmCantidadOptima;

private JMenuItem mntmCuotaDiaria;

private JMenuItem mntmAcercaDeLaTienda;

// Datos mínimos del primer disco duro

public static String modelo0 = "Western Digital My Cloud Home 6TB";

public static double precio0 = 1499.0;

public static double fondo0 = 14.0;

public static double ancho0 = 5.3;

public static double alto0 = 17.5;

public static double capacidad0 = 6;

// Datos mínimos del segundo disco duro

public static String modelo1 = "Western Digital Elements 4TB";

public static double precio1 = 555.0;

public static double ancho1 = 11.5;

public static double alto1 = 8.2;

public static double fondo1 = 2.1;

public static double capacidad1 = 4;

// Datos mínimos del tercer disco duro

public static String modelo2 = "Toshiba Canvio Basics 4TB";

public static double precio2 = 499.0;

public static double ancho2 = 14.1;

public static double alto2 = 11.1;

public static double fondo2 = 3.8;

public static double capacidad2 = 4;

// Datos mínimos del cuarto disco duro

public static String modelo3 = "Seagate Expansion 4TB";

public static double precio3 = 619.0;

public static double ancho3 = 11.7;

public static double alto3 = 8.0;

public static double fondo3 = 2.1;

public static double capacidad3 = 4;

// Datos mínimos del quinto disco duro

public static String modelo4 = "Seagate Backup Plus Hub 10TB";

public static double precio4 = 1099.0;

public static double ancho4 = 19.8;

public static double alto4 = 11.8;

public static double fondo4 = 4.1;

public static double capacidad4 = 10;

// Porcentajes de descuento

public static double porcentaje1 = 7.5;

public static double porcentaje2 = 10.0;

public static double porcentaje3 = 12.5;

public static double porcentaje4 = 15.0;

// Obsequios

public static String tipoObsequio = "Lapicero";

public static int obsequioCantidad1 = 2;

public static int obsequioCantidad2 = 3;

public static int obsequioCantidad3 = 4;

// Cantidad óptima de unidades vendidas por día

public static int cantidadOptima = 10;

// Cuota diaria

public static double cuotaDiaria = 30000;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

Proyecto frame = new Proyecto();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Proyecto() {

setTitle("Tienda 1.0");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 600, 400);

menuTienda = new JMenuBar();

setJMenuBar(menuTienda);

mnArchivo = new JMenu("Archivo");

menuTienda.add(mnArchivo);

mntmSalir = new JMenuItem("Salir");

mntmSalir.addActionListener(this);

mnArchivo.add(mntmSalir);

mnMantenimiento = new JMenu("Mantenimiento");

menuTienda.add(mnMantenimiento);

mntmConsultarDisco = new JMenuItem("Consultar Disco Duro");

mntmConsultarDisco.addActionListener(this);

mnMantenimiento.add(mntmConsultarDisco);

mntmModificarDisco = new JMenuItem("Modificar Disco Duro");

mntmModificarDisco.addActionListener(this);

mnMantenimiento.add(mntmModificarDisco);

mntmListarDisco = new JMenuItem("Listar Disco Duro");

mntmListarDisco.addActionListener(this);

mnMantenimiento.add(mntmListarDisco);

mnVentas = new JMenu("Ventas");

menuTienda.add(mnVentas);

mntmVender = new JMenuItem("Vender");

mntmVender.addActionListener(this);

mnVentas.add(mntmVender);

mntmGenerarReportes = new JMenuItem("Generar Reportes");

mntmGenerarReportes.addActionListener(this);

mnVentas.add(mntmGenerarReportes);

mnConfiguracion = new JMenu("Configuraci\u00F3n");

menuTienda.add(mnConfiguracion);

mntmConfigurarDescuentos = new JMenuItem("Configurar Descuentos");

mntmConfigurarDescuentos.addActionListener(this);

mnConfiguracion.add(mntmConfigurarDescuentos);

mntmConfigurarObsequio = new JMenuItem("Configurar Obsequio");

mntmConfigurarObsequio.addActionListener(this);

mnConfiguracion.add(mntmConfigurarObsequio);

mntmCantidadOptima = new JMenuItem("Cantidad \u00D3ptima");

mntmCantidadOptima.addActionListener(this);

mnConfiguracion.add(mntmCantidadOptima);

mntmCuotaDiaria = new JMenuItem("Cuota Diaria");

mntmCuotaDiaria.addActionListener(this);

mnConfiguracion.add(mntmCuotaDiaria);

mnAyuda = new JMenu("Ayuda");

menuTienda.add(mnAyuda);

mntmAcercaDeLaTienda = new JMenuItem("Acerca de Tienda");

mntmAcercaDeLaTienda.addActionListener(this);

mnAyuda.add(mntmAcercaDeLaTienda);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == mntmCuotaDiaria) {

actionPerformedMntmCuotaDiaria(e);

}

if (e.getSource() == mntmCantidadOptima) {

actionPerformedMntmCantidadOptima(e);

}

if (e.getSource() == mntmConfigurarObsequio) {

actionPerformedMntmConfigurarObsequio(e);

}

if (e.getSource() == mntmConfigurarDescuentos) {

actionPerformedMntmConfigurarDescuentos(e);

}

if (e.getSource() == mntmGenerarReportes) {

actionPerformedMntmGenerarReportes(e);

}

if (e.getSource() == mntmVender) {

actionPerformedMntmVender(e);

}

if (e.getSource() == mntmListarDisco) {

actionPerformedMntmListarDisco(e);

}

if (e.getSource() == mntmModificarDisco) {

actionPerformedMntmModificarDisco(e);

}

if (e.getSource() == mntmConsultarDisco) {

actionPerformedMntmConsultarDisco(e);

}

if (e.getSource() == mntmAcercaDeLaTienda) {

actionPerformedMntmAcercaDeLaTienda(e);

}

if (e.getSource() == mntmSalir) {

actionPerformedMntmSalir(e);

}

}

// Menú Archivo > Salir

protected void actionPerformedMntmSalir(ActionEvent e) {

dispose();

}

// Menú Mantenimiento > Consultar Disco Duro

protected void actionPerformedMntmConsultarDisco(ActionEvent e) {

DialogoConsultarDisco cdd = new DialogoConsultarDisco();

cdd.setLocationRelativeTo(this);

cdd.setVisible(true);

}

// Menú Mantenimiento > Listar Disco Duro

protected void actionPerformedMntmListarDisco(ActionEvent e) {

DialogoListarDisco ldd = new DialogoListarDisco();

ldd.setLocationRelativeTo(this);

ldd.setVisible(true);

}

// Menú Mantenimiento > Modificar Disco Duro

protected void actionPerformedMntmModificarDisco(ActionEvent e) {

DialogoModificarDisco mdd = new DialogoModificarDisco();

mdd.setLocationRelativeTo(this);

mdd.setVisible(true);

}

// Menú Ventas > Vender

protected void actionPerformedMntmVender(ActionEvent e) {

DialogoVender ven = new DialogoVender();

ven.setLocationRelativeTo(this);

ven.setVisible(true);

}

// Menú Ventas > Generar Reportes

protected void actionPerformedMntmGenerarReportes(ActionEvent e) {

DialogoGenerarReportes grv = new DialogoGenerarReportes();

grv.setLocationRelativeTo(this);

grv.setVisible(true);

}

// Menú Configuraración > Configurar Descuentos

protected void actionPerformedMntmConfigurarDescuentos(ActionEvent e) {

DialogoConfigurarDescuentos cdt = new DialogoConfigurarDescuentos();

cdt.setLocationRelativeTo(this);

cdt.setVisible(true);

}

// Menú Configuración > Configurar Obsequios

protected void actionPerformedMntmConfigurarObsequio(ActionEvent e) {

DialogoConfigurarObsequios cob = new DialogoConfigurarObsequios();

cob.setLocationRelativeTo(this);

cob.setVisible(true);

}

// Menú Configuración > Configurar Cantidad Óptima

protected void actionPerformedMntmCantidadOptima(ActionEvent e) {

DialogoConfigurarCantidadOptima cco = new DialogoConfigurarCantidadOptima();

cco.setLocationRelativeTo(this);

cco.setVisible(true);

}

// Menú Configuración > Configurar Cuota Diaria

protected void actionPerformedMntmCuotaDiaria(ActionEvent e) {

DialogoConfigurarCuotaDiaria ccd = new DialogoConfigurarCuotaDiaria();

ccd.setLocationRelativeTo(this);

ccd.setVisible(true);

}

// Menú Ayuda > Acerca de la Tienda

protected void actionPerformedMntmAcercaDeLaTienda(ActionEvent e) {

DialogoAcercaTienda dat = new DialogoAcercaTienda();

dat.setLocationRelativeTo(this);

dat.setVisible(true);

}

**Guis Dialogo Acerca de la tienda:**

public class DialogoAcercaTienda extends JDialog implements ActionListener {

private JButton btnCerrar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoAcercaTienda dialog = new DialogoAcercaTienda();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoAcercaTienda() {

setBounds(100, 100, 450, 300);

getContentPane().setLayout(null);

{

JLabel lblTienda = new JLabel("Tienda 1.0");

lblTienda.setFont(new Font("Tahoma", Font.BOLD, 23));

lblTienda.setBounds(161, 11, 129, 36);

getContentPane().add(lblTienda);

}

{

JSeparator separator = new JSeparator();

separator.setBounds(10, 58, 414, 2);

getContentPane().add(separator);

}

{

JLabel lblAutores = new JLabel("Autores");

lblAutores.setFont(new Font("Tahoma", Font.BOLD, 13));

lblAutores.setBounds(196, 71, 53, 14);

getContentPane().add(lblAutores);

}

{

JLabel lblBryanGrados = new JLabel("Bryan Grados");

lblBryanGrados.setHorizontalAlignment(SwingConstants.CENTER);

lblBryanGrados.setFont(new Font("Tahoma", Font.PLAIN, 13));

lblBryanGrados.setBounds(182, 112, 81, 14);

getContentPane().add(lblBryanGrados);

}

{

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(174, 227, 89, 23);

getContentPane().add(btnCerrar);

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnCerrar) {

actionPerformedBtnNewButton(e);

}

}

protected void actionPerformedBtnNewButton(ActionEvent e) {

dispose();

}

}

**Guis Configuracion:**

* **Dialogo configurar cantidad optima:**

public class DialogoConfigurarCantidadOptima extends JDialog implements ActionListener {

private JTextField txtCantidadOptima;

private JButton btnCancelar;

private JButton btnAceptar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoConfigurarCantidadOptima dialog = new DialogoConfigurarCantidadOptima();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoConfigurarCantidadOptima() {

setTitle("Configurar | Cantidad \u00D3ptima");

setBounds(100, 100, 450, 116);

getContentPane().setLayout(null);

{

JLabel lblCantidadOptima = new JLabel("Cantidad \u00D3ptima de unidades vendidas");

lblCantidadOptima.setFont(new Font("Tahoma", Font.PLAIN, 11));

lblCantidadOptima.setBounds(10, 11, 219, 14);

getContentPane().add(lblCantidadOptima);

}

{

txtCantidadOptima = new JTextField(Proyecto.cantidadOptima + "");

txtCantidadOptima.setBounds(215, 8, 86, 20);

getContentPane().add(txtCantidadOptima);

txtCantidadOptima.setColumns(10);

}

{

btnAceptar = new JButton("Aceptar");

btnAceptar.addActionListener(this);

btnAceptar.setBounds(335, 7, 89, 23);

getContentPane().add(btnAceptar);

}

{

btnCancelar = new JButton("Cancelar");

btnCancelar.addActionListener(this);

btnCancelar.setBounds(335, 32, 89, 23);

getContentPane().add(btnCancelar);

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnAceptar) {

actionPerformedBtnAceptar(e);

}

if (e.getSource() == btnCancelar) {

actionPerformedBtnCancelar(e);

}

}

protected void actionPerformedBtnAceptar(ActionEvent e) {

grabarDatos();

}

protected void actionPerformedBtnCancelar(ActionEvent e) {

dispose();

}

void grabarDatos() {

try {

int cdOptima = leerNum(txtCantidadOptima);

Proyecto.cantidadOptima = cdOptima;

dispose();

}

catch (Exception x) {

validacion("valor", txtCantidadOptima);

}

}

void validacion(String s, JTextField txt) {

mensaje("Ingrese un " + s + " correcto", "Error", 0);

txt.setText("");

txt.requestFocus();

}

void mensaje(String s1, String s2, int i) {

JOptionPane.showMessageDialog(this, s1, s2, i);

}

int leerNum(JTextField txt) {

return Integer.parseInt(txt.getText());

}

}

* **Dialogo configurar cuota diaria:**

public class DialogoConfigurarCuotaDiaria extends JDialog implements ActionListener {

private final JPanel contentPanel = new JPanel();

private JTextField txtCuotaDiaria;

private JButton btnCancelar;

private JButton btnAceptar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoConfigurarCuotaDiaria dialog = new DialogoConfigurarCuotaDiaria();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoConfigurarCuotaDiaria() {

setTitle("Configurar | Cuota Diaria");

setBounds(100, 100, 450, 105);

getContentPane().setLayout(new BorderLayout());

contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));

getContentPane().add(contentPanel, BorderLayout.CENTER);

contentPanel.setLayout(null);

{

JLabel lblCuotaDiaria = new JLabel("Cuota diaria esperada (S/.)");

lblCuotaDiaria.setBounds(10, 11, 160, 14);

contentPanel.add(lblCuotaDiaria);

}

{

txtCuotaDiaria = new JTextField(Proyecto.cuotaDiaria + "");

txtCuotaDiaria.setBounds(180, 8, 86, 20);

contentPanel.add(txtCuotaDiaria);

txtCuotaDiaria.setColumns(10);

}

{

btnAceptar = new JButton("Aceptar");

btnAceptar.addActionListener(this);

btnAceptar.setBounds(335, 7, 89, 23);

contentPanel.add(btnAceptar);

}

{

btnCancelar = new JButton("Cancelar");

btnCancelar.addActionListener(this);

btnCancelar.setBounds(335, 35, 89, 23);

contentPanel.add(btnCancelar);

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnAceptar) {

actionPerformedBtnAceptar(e);

}

if (e.getSource() == btnCancelar) {

actionPerformedBtnCancelar(e);

}

}

protected void actionPerformedBtnAceptar(ActionEvent e) {

grabarDatos();

}

protected void actionPerformedBtnCancelar(ActionEvent e) {

dispose();

}

void grabarDatos() {

try {

double ctDiaria = leerNum(txtCuotaDiaria);

Proyecto.cuotaDiaria = ctDiaria;

dispose();

}

catch (Exception x) {

validacion("valor" , txtCuotaDiaria);

}

}

void validacion(String s, JTextField txt) {

mensaje("Ingrese un " + s + " correcto", "Error", 0);

txt.setText("");

txt.requestFocus();

}

void mensaje(String s1, String s2, int i) {

JOptionPane.showMessageDialog(this, s1, s2, i);

}

double leerNum(JTextField txt) {

return Double.parseDouble(txt.getText());

}

}

* **Dialogo configurar descuentos:**

public class DialogoConfigurarDescuentos extends JDialog implements ActionListener {

private JTextField txtPorcentaje1;

private JTextField txtPorcentaje2;

private JTextField txtPorcentaje3;

private JTextField txtPorcentaje4;

private JButton btnCancelar;

private JButton btnAceptar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoConfigurarDescuentos dialog = new DialogoConfigurarDescuentos();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoConfigurarDescuentos() {

setTitle("Configurar | Descuentos");

setBounds(100, 100, 389, 160);

getContentPane().setLayout(null);

{

JLabel lblPorcentaje1 = new JLabel("1 a 5 unidades");

lblPorcentaje1.setBounds(10, 11, 100, 14);

getContentPane().add(lblPorcentaje1);

}

{

JLabel lblPorcentaje2 = new JLabel("6 a 10 unidades");

lblPorcentaje2.setBounds(10, 36, 100, 14);

getContentPane().add(lblPorcentaje2);

}

{

JLabel lblPorcentaje3 = new JLabel("11 a 15 unidades");

lblPorcentaje3.setBounds(10, 61, 100, 14);

getContentPane().add(lblPorcentaje3);

}

{

JLabel lblPorcentaje4 = new JLabel("M\u00E1s de 15 unidades");

lblPorcentaje4.setBounds(10, 86, 122, 14);

getContentPane().add(lblPorcentaje4);

}

{

txtPorcentaje1 = new JTextField(Proyecto.porcentaje1 + "");

txtPorcentaje1.setBounds(140, 8, 63, 20);

getContentPane().add(txtPorcentaje1);

txtPorcentaje1.setColumns(10);

}

{

txtPorcentaje2 = new JTextField(Proyecto.porcentaje2 + "");

txtPorcentaje2.setColumns(10);

txtPorcentaje2.setBounds(140, 33, 63, 20);

getContentPane().add(txtPorcentaje2);

}

{

txtPorcentaje3 = new JTextField(Proyecto.porcentaje3 + "");

txtPorcentaje3.setColumns(10);

txtPorcentaje3.setBounds(140, 58, 63, 20);

getContentPane().add(txtPorcentaje3);

}

{

txtPorcentaje4 = new JTextField(Proyecto.porcentaje4 + "");

txtPorcentaje4.setColumns(10);

txtPorcentaje4.setBounds(140, 83, 63, 20);

getContentPane().add(txtPorcentaje4);

}

{

JLabel lblPorc4 = new JLabel("%");

lblPorc4.setBounds(213, 82, 46, 23);

getContentPane().add(lblPorc4);

}

{

JLabel lblPorc3 = new JLabel("%");

lblPorc3.setBounds(213, 57, 46, 23);

getContentPane().add(lblPorc3);

}

{

JLabel lblPorc2 = new JLabel("%");

lblPorc2.setBounds(213, 32, 46, 23);

getContentPane().add(lblPorc2);

}

{

JLabel lblPorc1 = new JLabel("%");

lblPorc1.setBounds(213, 7, 46, 23);

getContentPane().add(lblPorc1);

}

{

btnAceptar = new JButton("Aceptar");

btnAceptar.addActionListener(this);

btnAceptar.setBounds(278, 7, 89, 23);

getContentPane().add(btnAceptar);

}

{

btnCancelar = new JButton("Cancelar");

btnCancelar.addActionListener(this);

btnCancelar.setBounds(278, 32, 89, 23);

getContentPane().add(btnCancelar);

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnAceptar) {

actionPerformedBtnAceptar(e);

}

if (e.getSource() == btnCancelar) {

actionPerformedBtnCancelar(e);

}

}

protected void actionPerformedBtnAceptar(ActionEvent e) {

grabarDatos();

}

protected void actionPerformedBtnCancelar(ActionEvent e) {

dispose();

}

void grabarDatos() {

try {

double porcen1 = leerNum(txtPorcentaje1);

try {

double porcen2 = leerNum(txtPorcentaje2);

try {

double porcen3 = leerNum(txtPorcentaje3);

try {

double porcen4 = leerNum(txtPorcentaje4);

Proyecto.porcentaje1 = porcen1;

Proyecto.porcentaje2 = porcen2;

Proyecto.porcentaje3 = porcen3;

Proyecto.porcentaje4 = porcen4;

dispose();

}

catch (Exception x) {

validacion("Nro° 4" , txtPorcentaje4);

}

}

catch (Exception x) {

validacion("Nro° 3" , txtPorcentaje3);

}

}

catch (Exception x) {

validacion("Nro° 2" , txtPorcentaje2);

}

}

catch (Exception x) {

validacion("Nro° 1" , txtPorcentaje1);

}

}

void validacion(String s, JTextField txt) {

mensaje("Valor " + s + " incorrecto", "Error", 0);

txt.setText("");

txt.requestFocus();

}

void mensaje(String s1, String s2, int i) {

JOptionPane.showMessageDialog(this, s1, s2, i);

}

double leerNum(JTextField txt) {

return Double.parseDouble(txt.getText());

}

}

* **Dialogo configurar obsequio:**

public class DialogoConfigurarObsequios extends JDialog implements ActionListener {

private JTextField txtTipoObsequio;

private JLabel lblObsequioCantidad1;

private JLabel lblObsequioCantidad2;

private JLabel lblObsequioCantidad3;

private JTextField txtObsequioCantidad1;

private JTextField txtObsequioCantidad2;

private JTextField txtObsequioCantidad3;

private JButton btnAceptar;

private JButton btnCancelar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoConfigurarObsequios dialog = new DialogoConfigurarObsequios();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoConfigurarObsequios() {

setTitle("Configurar | Obsequios");

setBounds(100, 100, 450, 151);

getContentPane().setLayout(null);

{

JLabel lblTipoObsequio = new JLabel("Tipo de Obsequio");

lblTipoObsequio.setBounds(10, 11, 109, 14);

getContentPane().add(lblTipoObsequio);

}

txtTipoObsequio = new JTextField(Proyecto.tipoObsequio);

txtTipoObsequio.setBounds(166, 8, 86, 20);

getContentPane().add(txtTipoObsequio);

txtTipoObsequio.setColumns(10);

lblObsequioCantidad1 = new JLabel("1 a 5 unidades");

lblObsequioCantidad1.setBounds(10, 36, 109, 14);

getContentPane().add(lblObsequioCantidad1);

lblObsequioCantidad2 = new JLabel("6 a 10 unidades");

lblObsequioCantidad2.setBounds(10, 61, 109, 14);

getContentPane().add(lblObsequioCantidad2);

lblObsequioCantidad3 = new JLabel("11 a m\u00E1s unidades");

lblObsequioCantidad3.setBounds(10, 86, 109, 14);

getContentPane().add(lblObsequioCantidad3);

txtObsequioCantidad1 = new JTextField(Proyecto.obsequioCantidad1 + "");

txtObsequioCantidad1.setColumns(10);

txtObsequioCantidad1.setBounds(166, 33, 86, 20);

getContentPane().add(txtObsequioCantidad1);

txtObsequioCantidad2 = new JTextField(Proyecto.obsequioCantidad2 + "");

txtObsequioCantidad2.setColumns(10);

txtObsequioCantidad2.setBounds(166, 58, 86, 20);

getContentPane().add(txtObsequioCantidad2);

txtObsequioCantidad3 = new JTextField(Proyecto.obsequioCantidad3 + "");

txtObsequioCantidad3.setColumns(10);

txtObsequioCantidad3.setBounds(166, 83, 86, 20);

getContentPane().add(txtObsequioCantidad3);

btnAceptar = new JButton("Aceptar");

btnAceptar.addActionListener(this);

btnAceptar.setBounds(335, 7, 89, 23);

getContentPane().add(btnAceptar);

btnCancelar = new JButton("Cancelar");

btnCancelar.addActionListener(this);

btnCancelar.setBounds(335, 36, 89, 23);

getContentPane().add(btnCancelar);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnAceptar) {

actionPerformedBtnAceptar(e);

}

if (e.getSource() == btnCancelar) {

actionPerformedBtnCancelar(e);

}

}

protected void actionPerformedBtnAceptar(ActionEvent e) {

grabarDatos();

}

protected void actionPerformedBtnCancelar(ActionEvent e) {

dispose();

}

void grabarDatos() {

String tipObs = leerDat(txtTipoObsequio);

if (tipObs.length() > 0)

try {

int de1\_5 = leerNum(txtObsequioCantidad1);

try {

int de6\_10 = leerNum(txtObsequioCantidad2);

try {

int de11\_ = leerNum(txtObsequioCantidad3);

Proyecto.tipoObsequio = tipObs;

Proyecto.obsequioCantidad1 = de1\_5;

Proyecto.obsequioCantidad2 = de6\_10;

Proyecto.obsequioCantidad3 = de11\_;

dispose();

}

catch (Exception x) {

validacion("valor" , txtObsequioCantidad3);

}

}

catch (Exception x) {

validacion("valor" , txtObsequioCantidad2);

}

}

catch (Exception x) {

validacion("valor" , txtObsequioCantidad1);

}

else

validacion("nombre" , txtTipoObsequio);

}

void validacion(String s, JTextField txt) {

mensaje("Ingrese un " + s + " correcto", "Error", 0);

txt.setText("");

txt.requestFocus();

}

void mensaje(String s1, String s2, int i) {

JOptionPane.showMessageDialog(this, s1, s2, i);

}

int leerNum(JTextField txt) {

return Integer.parseInt(txt.getText());

}

String leerDat(JTextField txt) {

return String.format(txt.getText());

}

}

**Guis Mantenimiento:**

* **Dialogo consultar disco :**

public class DialogoConsultarDisco extends JDialog implements ActionListener {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JComboBox cboModelo;

private JTextField txtPrecio;

private JTextField txtAncho;

private JTextField txtAlto;

private JTextField txtFondo;

private JTextField txtCapacidad;

private JButton btnCerrar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoConsultarDisco dialog = new DialogoConsultarDisco();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoConsultarDisco() {

setTitle("Mantenimiento | Consultar Disco Duro");

setBounds(100, 100, 500, 200);

getContentPane().setLayout(null);

{

JLabel lblModelo = new JLabel("Modelo");

lblModelo.setBounds(10, 11, 89, 14);

getContentPane().add(lblModelo);

}

{

JLabel lblPrecio = new JLabel("Precio (S/.)");

lblPrecio.setBounds(10, 36, 89, 14);

getContentPane().add(lblPrecio);

}

{

JLabel lblAncho = new JLabel("Ancho (cm)");

lblAncho.setBounds(10, 61, 89, 14);

getContentPane().add(lblAncho);

}

{

JLabel lblAlto = new JLabel("Alto (cm)");

lblAlto.setBounds(10, 86, 89, 14);

getContentPane().add(lblAlto);

}

{

JLabel lblFondo = new JLabel("Fondo (cm)");

lblFondo.setBounds(10, 111, 89, 14);

getContentPane().add(lblFondo);

}

{

JLabel lblCapacidad = new JLabel("Capacidad (TB)");

lblCapacidad.setBounds(10, 136, 89, 14);

getContentPane().add(lblCapacidad);

}

cboModelo = new JComboBox();

cboModelo.addActionListener(this);

cboModelo.setModel(new DefaultComboBoxModel(new String[] {"Western Digital My Cloud Home 6TB", "Western Digital Elements 4TB", "Toshiba Canvio Basics 4TB", "Seagate Expansion 4TB", "Seagate Backup Plus Hub 10TB"}));

cboModelo.setBounds(123, 7, 230, 22);

getContentPane().add(cboModelo);

txtPrecio = new JTextField(Proyecto.precio0 + "");

txtPrecio.setEditable(false);

txtPrecio.setBounds(123, 33, 230, 20);

getContentPane().add(txtPrecio);

txtPrecio.setColumns(10);

txtAncho = new JTextField(Proyecto.ancho0 + "");

txtAncho.setEditable(false);

txtAncho.setBounds(123, 58, 230, 20);

getContentPane().add(txtAncho);

txtAncho.setColumns(10);

txtAlto = new JTextField(Proyecto.alto0 + "");

txtAlto.setEditable(false);

txtAlto.setBounds(123, 83, 230, 20);

getContentPane().add(txtAlto);

txtAlto.setColumns(10);

txtFondo = new JTextField(Proyecto.fondo0 + "");

txtFondo.setEditable(false);

txtFondo.setBounds(123, 108, 230, 20);

getContentPane().add(txtFondo);

txtFondo.setColumns(10);

txtCapacidad = new JTextField(Proyecto.capacidad0 + "");

txtCapacidad.setEditable(false);

txtCapacidad.setBounds(123, 133, 230, 20);

getContentPane().add(txtCapacidad);

txtCapacidad.setColumns(10);

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(385, 7, 89, 23);

getContentPane().add(btnCerrar);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cboModelo) {

actionPerformedCboModelo(e);

}

if (e.getSource() == btnCerrar) {

actionPerformedBtnCerrar(e);

}

}

protected void actionPerformedCboModelo(ActionEvent e) {

int posModelo = cboModelo.getSelectedIndex();

switch (posModelo) {

case 0:

txtPrecio.setText(Proyecto.precio0 + "");

txtAncho.setText(Proyecto.ancho0 + "");

txtAlto.setText(Proyecto.alto0 + "");

txtFondo.setText(Proyecto.fondo0 + "");

txtCapacidad.setText(Proyecto.capacidad0 + "");

break;

case 1:

txtPrecio.setText(Proyecto.precio1 + "");

txtAncho.setText(Proyecto.ancho1 + "");

txtAlto.setText(Proyecto.alto1 + "");

txtFondo.setText(Proyecto.fondo1 + "");

txtCapacidad.setText(Proyecto.capacidad1 + "");

break;

case 2:

txtPrecio.setText(Proyecto.precio2 + "");

txtAncho.setText(Proyecto.ancho2 + "");

txtAlto.setText(Proyecto.alto2 + "");

txtFondo.setText(Proyecto.fondo2 + "");

txtCapacidad.setText(Proyecto.capacidad2 + "");

break;

case 3:

txtPrecio.setText(Proyecto.precio3 + "");

txtAncho.setText(Proyecto.ancho3 + "");

txtAlto.setText(Proyecto.alto3 + "");

txtFondo.setText(Proyecto.fondo3 + "");

txtCapacidad.setText(Proyecto.capacidad3 + "");

break;

default:

txtPrecio.setText(Proyecto.precio4 + "");

txtAncho.setText(Proyecto.ancho4 + "");

txtAlto.setText(Proyecto.alto4 + "");

txtFondo.setText(Proyecto.fondo4 + "");

txtCapacidad.setText(Proyecto.capacidad4 + "");

}

}

protected void actionPerformedBtnCerrar(ActionEvent e) {

dispose();

}

}

* **Dialogo listar disco:**

public class DialogoListarDisco extends JDialog implements ActionListener {

private JButton btnCerrar;

private JButton btnListar;

private JScrollPane scrollPane;

private JTextArea txtS;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoListarDisco dialog = new DialogoListarDisco();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoListarDisco() {

setTitle("Mantenimiento | Listado de Discos Duros ");

setBounds(100, 100, 550, 523);

getContentPane().setLayout(null);

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(171, 437, 89, 23);

getContentPane().add(btnCerrar);

btnListar = new JButton("Listar");

btnListar.addActionListener(this);

btnListar.setBounds(270, 437, 89, 23);

getContentPane().add(btnListar);

scrollPane = new JScrollPane();

scrollPane.setBounds(10, 11, 514, 415);

getContentPane().add(scrollPane);

txtS = new JTextArea();

txtS.setFont(new Font("Monospaced", Font.PLAIN, 13));

scrollPane.setViewportView(txtS);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnListar) {

actionPerformedBtnListar(e);

}

if (e.getSource() == btnCerrar) {

actionPerformedBtnCerrar(e);

}

}

protected void actionPerformedBtnListar(ActionEvent e) {

listado();

}

protected void actionPerformedBtnCerrar(ActionEvent e) {

dispose();

}

void listado() {

imprimir("LISTADO DE DISCOS DUROS");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo0);

imprimir("Precio : " + "S/ " + Proyecto.precio0);

imprimir("Ancho : " + Proyecto.ancho0 + "cm" );

imprimir("Alto : " + Proyecto.alto0 + "cm");

imprimir("Profundidad : " + Proyecto.fondo0 + "cm");

imprimir("Capacidad : " + Proyecto.capacidad0 + "TB");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo1);

imprimir("Precio : " + "S/ " + Proyecto.precio1);

imprimir("Ancho : " + Proyecto.ancho1 + "cm" );

imprimir("Alto : " + Proyecto.alto1 + "cm");

imprimir("Profundidad : " + Proyecto.fondo1 + "cm");

imprimir("Capacidad : " + Proyecto.capacidad1 + "TB");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo2);

imprimir("Precio : " + "S/ " + Proyecto.precio2);

imprimir("Ancho : " + Proyecto.ancho2 + "cm" );

imprimir("Alto : " + Proyecto.alto2 + "cm");

imprimir("Profundidad : " + Proyecto.fondo2 + "cm");

imprimir("Capacidad : " + Proyecto.capacidad2 + "TB");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo3);

imprimir("Precio : " + "S/ " + Proyecto.precio3);

imprimir("Ancho : " + Proyecto.ancho3 + "cm" );

imprimir("Alto : " + Proyecto.alto3 + "cm");

imprimir("Profundidad : " + Proyecto.fondo3 + "cm");

imprimir("Capacidad : " + Proyecto.capacidad3 + "TB");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo4);

imprimir("Precio : " + "S/ " + Proyecto.precio4);

imprimir("Ancho : " + Proyecto.ancho4 + "cm" );

imprimir("Alto : " + Proyecto.alto4 + "cm");

imprimir("Profundidad : " + Proyecto.fondo4 + "cm");

imprimir("Capacidad : " + Proyecto.capacidad4 + "TB");

imprimir("");

}

void imprimir(String s) {

txtS.append(s + "\n");

}

}

* **Dialogo modificar disco:**

public class DialogoModificarDisco extends JDialog implements ActionListener {

private JTextField txtPrecio;

private JTextField txtAncho;

private JTextField txtAlto;

private JTextField txtFondo;

private JTextField txtCapacidad;

private JComboBox cboModelo;

private JButton btnGrabar;

private JButton btnCerrar;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoModificarDisco dialog = new DialogoModificarDisco();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoModificarDisco() {

setTitle("Mantenimiento | Modificar Disco Duro");

setBounds(100, 100, 500, 200);

getContentPane().setLayout(null);

{

JLabel lblModelo = new JLabel("Modelo");

lblModelo.setBounds(10, 11, 89, 14);

getContentPane().add(lblModelo);

}

{

JLabel lblPrecio = new JLabel("Precio (S/.)");

lblPrecio.setBounds(10, 36, 89, 14);

getContentPane().add(lblPrecio);

}

{

JLabel lblAncho = new JLabel("Ancho (cm)");

lblAncho.setBounds(10, 61, 89, 14);

getContentPane().add(lblAncho);

}

{

JLabel lblAlto = new JLabel("Alto (cm)");

lblAlto.setBounds(10, 86, 89, 14);

getContentPane().add(lblAlto);

}

{

JLabel lblFondo = new JLabel("Fondo (cm)");

lblFondo.setBounds(10, 111, 89, 14);

getContentPane().add(lblFondo);

}

{

JLabel lblCapacidad = new JLabel("Capacidad (TB)");

lblCapacidad.setBounds(10, 136, 89, 14);

getContentPane().add(lblCapacidad);

}

{

cboModelo = new JComboBox();

cboModelo.addActionListener(this);

cboModelo.setModel(new DefaultComboBoxModel(new String[] {"Western Digital My Cloud Home 6TB", "Western Digital Elements 4TB", "Toshiba Canvio Basics 4TB", "Seagate Expansion 4TB", "Seagate Backup Plus Hub 10TB"}));

cboModelo.setBounds(142, 7, 218, 22);

getContentPane().add(cboModelo);

}

{

txtPrecio = new JTextField(Proyecto.precio0 + "");

txtPrecio.setBounds(142, 33, 218, 20);

getContentPane().add(txtPrecio);

txtPrecio.setColumns(10);

}

{

txtAncho = new JTextField(Proyecto.ancho0 + "");

txtAncho.setColumns(10);

txtAncho.setBounds(142, 58, 218, 20);

getContentPane().add(txtAncho);

}

{

txtAlto = new JTextField(Proyecto.alto0 + "");

txtAlto.setColumns(10);

txtAlto.setBounds(142, 83, 218, 20);

getContentPane().add(txtAlto);

}

{

txtFondo = new JTextField(Proyecto.fondo0 + "");

txtFondo.setColumns(10);

txtFondo.setBounds(142, 108, 218, 20);

getContentPane().add(txtFondo);

}

{

txtCapacidad = new JTextField(Proyecto.capacidad0 + "");

txtCapacidad.setColumns(10);

txtCapacidad.setBounds(142, 133, 218, 20);

getContentPane().add(txtCapacidad);

}

{

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(385, 32, 89, 23);

getContentPane().add(btnCerrar);

}

{

btnGrabar = new JButton("Grabar");

btnGrabar.addActionListener(this);

btnGrabar.setBounds(385, 7, 89, 23);

getContentPane().add(btnGrabar);

}

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnCerrar) {

actionPerformedBtnCerrar(e);

}

if (e.getSource() == btnGrabar) {

actionPerformedBtnGrabar(e);

}

if (e.getSource() == cboModelo) {

actionPerformedComboBox(e);

}

}

protected void actionPerformedComboBox(ActionEvent e) {

listaTexto();

}

protected void actionPerformedBtnGrabar(ActionEvent e) {

grabarTexto();

}

protected void actionPerformedBtnCerrar(ActionEvent e) {

dispose();

}

void listaTexto() {

switch (cboModelo.getSelectedIndex()) {

case 0:

txtPrecio.setText(Proyecto.precio0 + "");

txtAncho.setText(Proyecto.ancho0 + "");

txtAlto.setText(Proyecto.alto0 + "");

txtFondo.setText(Proyecto.fondo0 + "");

txtCapacidad.setText(Proyecto.capacidad0 + "");

break;

case 1:

txtPrecio.setText(Proyecto.precio1 + "");

txtAncho.setText(Proyecto.ancho1 + "");

txtAlto.setText(Proyecto.alto1 + "");

txtFondo.setText(Proyecto.fondo1 + "");

txtCapacidad.setText(Proyecto.capacidad1 + "");

break;

case 2:

txtPrecio.setText(Proyecto.precio2 + "");

txtAncho.setText(Proyecto.ancho2 + "");

txtAlto.setText(Proyecto.alto2 + "");

txtFondo.setText(Proyecto.fondo2 + "");

txtCapacidad.setText(Proyecto.capacidad2 + "");

break;

case 3:

txtPrecio.setText(Proyecto.precio3 + "");

txtAncho.setText(Proyecto.ancho3 + "");

txtAlto.setText(Proyecto.alto3 + "");

txtFondo.setText(Proyecto.fondo3 + "");

txtCapacidad.setText(Proyecto.capacidad3 + "");

break;

default:

txtPrecio.setText(Proyecto.precio4 + "");

txtAncho.setText(Proyecto.ancho4 + "");

txtAlto.setText(Proyecto.alto4 + "");

txtFondo.setText(Proyecto.fondo4 + "");

txtCapacidad.setText(Proyecto.capacidad4 + "");

}

}

void grabarTexto() {

try {

double precio = leerNum(txtPrecio);

try {

double ancho = leerNum(txtAncho);

try {

double alto = leerNum(txtAlto);

try {

double fondo = leerNum(txtFondo);

try {

double capacidad = leerNum(txtCapacidad);

switch (cboModelo.getSelectedIndex()) {

case 0:

Proyecto.precio0 = precio;

Proyecto.ancho0 = ancho;

Proyecto.alto0 = alto;

Proyecto.fondo0 = fondo;

Proyecto.capacidad0 = capacidad;

break;

case 1:

Proyecto.precio1 = precio;

Proyecto.ancho1 = ancho;

Proyecto.alto1 = alto;

Proyecto.fondo1 = fondo;

Proyecto.capacidad1 = capacidad;

break;

case 2:

Proyecto.precio2 = precio;

Proyecto.ancho2 = ancho;

Proyecto.alto2 = alto;

Proyecto.fondo2 = fondo;

Proyecto.capacidad2 = capacidad;

break;

case 3:

Proyecto.precio3 = precio;

Proyecto.ancho3 = ancho;

Proyecto.alto3 = alto;

Proyecto.fondo3 = fondo;

Proyecto.capacidad3 = capacidad;

break;

default:

Proyecto.precio4 = precio;

Proyecto.ancho4 = ancho;

Proyecto.alto4 = alto;

Proyecto.fondo4 = fondo;

Proyecto.capacidad4 = capacidad;

}

dispose();

}

catch (Exception x) {

validacion("Capacidad", txtCapacidad);

}

}

catch (Exception x) {

validacion("Fondo", txtFondo);

}

}

catch (Exception x) {

validacion("Alto", txtAlto);

}

}

catch (Exception x) {

validacion("Ancho", txtAncho);

}

}

catch (Exception x) {

validacion("Precio", txtPrecio);

}

}

void validacion(String s, JTextField txt) {

mensaje("Ingrese " + s + " correcto", "Error", 0);

txt.setText("");

txt.requestFocus();

}

void mensaje(String s1, String s2, int i) {

JOptionPane.showMessageDialog(this, s1, s2, i);

}

double leerNum(JTextField txt) {

return Double.parseDouble(txt.getText());

}

}

**Guis ventas :**

* **Dialogo reporte de ventas :**

public class DialogoGenerarReportes extends JDialog implements ActionListener {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private final JPanel contentPanel = new JPanel();

private JButton btnCerrar;

private JComboBox cboReportes;

private JScrollPane scrollPane;

private JTextArea txtS;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoGenerarReportes dialog = new DialogoGenerarReportes();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoGenerarReportes() {

setTitle("Generar Reportes");

setBounds(100, 100, 600, 300);

getContentPane().setLayout(new BorderLayout());

contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));

getContentPane().add(contentPanel, BorderLayout.CENTER);

contentPanel.setLayout(null);

{

JLabel lbltipoReporte = new JLabel("Tipo de Reporte");

lbltipoReporte.setBounds(10, 15, 102, 14);

contentPanel.add(lbltipoReporte);

}

{

cboReportes = new JComboBox();

cboReportes.addActionListener(this);

cboReportes.setModel(new DefaultComboBoxModel(new String[] {"Ventas por modelo", "Raz\u00F3n de precios en relaci\u00F3n al precio mayor", "Estad\u00EDstica sobre la cantidad \u00F3ptima", "Estad\u00EDstica sobre el precio"}));

cboReportes.setBounds(108, 11, 261, 22);

contentPanel.add(cboReportes);

}

{

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(485, 11, 89, 23);

contentPanel.add(btnCerrar);

}

scrollPane = new JScrollPane();

scrollPane.setBounds(10, 57, 564, 193);

contentPanel.add(scrollPane);

txtS = new JTextArea();

txtS.setFont(new Font("Monospaced", Font.PLAIN, 13));

scrollPane.setViewportView(txtS);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cboReportes) {

actionPerformedCboReportes(e);

}

if (e.getSource() == btnCerrar) {

actionPerformedBtnCerrar(e);

}

}

protected void actionPerformedCboReportes(ActionEvent e) {

txtS.setText("");

switch (cboReportes.getSelectedIndex()) {

case 0:

ventaPorModelo(); break;

case 1:

razonesPrecios(); break;

case 2:

cantidadOptima(); break;

default:

promedios();

}

}

void ventaPorModelo() {

//Variables

double aporteModelo0 = ((DialogoVender.impAcumuladoModelo0\*100)/Proyecto.cuotaDiaria);

double aporteModelo1 = ((DialogoVender.impAcumuladoModelo1\*100)/Proyecto.cuotaDiaria);

double aporteModelo2 = ((DialogoVender.impAcumuladoModelo2\*100)/Proyecto.cuotaDiaria);

double aporteModelo3 = ((DialogoVender.impAcumuladoModelo3\*100)/Proyecto.cuotaDiaria);

double aporteModelo4 = ((DialogoVender.impAcumuladoModelo4\*100)/Proyecto.cuotaDiaria);

//Impresión

imprimir("VENTAS POR MODELO");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo0);

imprimir("Cantidad de ventas : " + DialogoVender.cantidadVentasModelo0);

imprimir("Cantidad de unidades vendidas : " + DialogoVender.discosVendidosModelo0);

imprimir("Importe total vendido : S/." + DialogoVender.impAcumuladoModelo0);

imprimir("Aporte a la cuota diaria : " + Math.round(aporteModelo0\*100.0)/100.0 + "%");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo1);

imprimir("Cantidad de ventas : " + DialogoVender.cantidadVentasModelo1);

imprimir("Cantidad de unidades vendidas : " + DialogoVender.discosVendidosModelo1);

imprimir("Importe total vendido : S/." + DialogoVender.impAcumuladoModelo1);

imprimir("Aporte a la cuota diaria : " + Math.round(aporteModelo1\*100.0)/100.0 + "%");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo2);

imprimir("Cantidad de ventas : " + DialogoVender.cantidadVentasModelo2);

imprimir("Cantidad de unidades vendidas : " + DialogoVender.discosVendidosModelo2);

imprimir("Importe total vendido : S/." + DialogoVender.impAcumuladoModelo2);

imprimir("Aporte a la cuota diaria : " + Math.round(aporteModelo2\*100.0)/100.0 + "%");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo3);

imprimir("Cantidad de ventas : " + DialogoVender.cantidadVentasModelo3);

imprimir("Cantidad de unidades vendidas : " + DialogoVender.discosVendidosModelo3);

imprimir("Importe total vendido : S/." + DialogoVender.impAcumuladoModelo3);

imprimir("Aporte a la cuota diaria : " + Math.round(aporteModelo3\*100.0)/100.0 + "%");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo4);

imprimir("Cantidad de ventas : " + DialogoVender.cantidadVentasModelo4);

imprimir("Cantidad de unidades vendidas : " + DialogoVender.discosVendidosModelo4);

imprimir("Importe total vendido : S/." + Math.round(DialogoVender.impAcumuladoModelo4\*100.0)/100.0);

imprimir("Aporte a la cuota diaria : " + Math.round(aporteModelo4\*100.0)/100.0 + "%");

imprimir("");

}

//---

void razonesPrecios() {

double precioMayor;

precioMayor = calcularMaximo(Proyecto.precio0, Proyecto.precio1, Proyecto.precio2, Proyecto.precio3, Proyecto.precio4);

double razonModelo0 = ((Proyecto.precio0/precioMayor) \* 1);

double razonModelo1 = ((Proyecto.precio1/precioMayor) \* 1);

double razonModelo2 = ((Proyecto.precio2/precioMayor) \* 1);

double razonModelo3 = ((Proyecto.precio3/precioMayor) \* 1);

double razonModelo4 = ((Proyecto.precio4/precioMayor) \* 1);

BigDecimal bd0 = new BigDecimal(razonModelo0).setScale(2, RoundingMode.DOWN);

double razonMod0 = bd0.doubleValue();

BigDecimal bd1 = new BigDecimal(razonModelo1).setScale(2, RoundingMode.DOWN);

double razonMod1 = bd1.doubleValue();

BigDecimal bd2 = new BigDecimal(razonModelo2).setScale(2, RoundingMode.DOWN);

double razonMod2 = bd2.doubleValue();

BigDecimal bd3 = new BigDecimal(razonModelo3).setScale(2, RoundingMode.DOWN);

double razonMod3 = bd3.doubleValue();

BigDecimal bd4 = new BigDecimal(razonModelo4).setScale(2, RoundingMode.DOWN);

double razonMod4 = bd4.doubleValue();

imprimir("RAZÓN DE PRECIOS EN RELACIÓN AL PRECIO MAYOR");

imprimir("");

imprimir("Modelo : " + Proyecto.modelo0);

imprimir("Precio : " + Proyecto.precio0);

imprimir("Razón : " + razonMod0);

imprimir("");

imprimir("Modelo : " + Proyecto.modelo1);

imprimir("Precio : " + Proyecto.precio1);

imprimir("Razón : " + razonMod1);

imprimir("");

imprimir("Modelo : " + Proyecto.modelo2);

imprimir("Precio : " + Proyecto.precio2);

imprimir("Razón : " + razonMod2);

imprimir("");

imprimir("Modelo : " + Proyecto.modelo3);

imprimir("Precio : " + Proyecto.precio3);

imprimir("Razón : " + razonMod3);

imprimir("");

imprimir("Modelo : " + Proyecto.modelo4);

imprimir("Precio : " + Proyecto.precio4);

imprimir("Razón : " + razonMod4);

}

//---

private int superoCantidad = 0;

private int igualoCantidad = 0;

private int menorCantidad = 0;

void validarVentas (int unidadesVendidas) {

if(unidadesVendidas == Proyecto.cantidadOptima)

igualoCantidad ++;

else

if(unidadesVendidas >= Proyecto.cantidadOptima)

superoCantidad ++;

else

menorCantidad++;

}

void ReinicioContador() {

superoCantidad = 0;

igualoCantidad = 0;

menorCantidad = 0;

}

void cantidadOptima () {

ReinicioContador();

validarVentas(DialogoVender.discosVendidosModelo0);

validarVentas(DialogoVender.discosVendidosModelo1);

validarVentas(DialogoVender.discosVendidosModelo2);

validarVentas(DialogoVender.discosVendidosModelo3);

validarVentas(DialogoVender.discosVendidosModelo4);

imprimir("ESTADÍSTICA SOBRE LA CANTIDAD ÓPTIMA");

imprimir("");

imprimir("Cantidad de modelos que superaron la cantidad óptima : " + superoCantidad);

imprimir("Cantidad de modelos que no superaron la cantidad óptima : " + menorCantidad);

imprimir("Cantidad de modelos que igualaron la cantidad óptima : " + igualoCantidad);

}

//-----

void promedios () {

double precioPromedio, precioMinimo, precioMaximo;

precioPromedio = calcularPromedio(Proyecto.precio0, Proyecto.precio1, Proyecto.precio2, Proyecto.precio3, Proyecto.precio4);

precioMaximo = calcularMaximo(Proyecto.precio0, Proyecto.precio1, Proyecto.precio2, Proyecto.precio3, Proyecto.precio4);

precioMinimo = calcularMinimo(Proyecto.precio0, Proyecto.precio1, Proyecto.precio2, Proyecto.precio3, Proyecto.precio4);

imprimir("ESTADÍSTICA SOBRE EL PRECIO");

imprimir("");

imprimir("Precio promedio : S/. " + precioPromedio);

imprimir("Precio mayor : S/. " + precioMaximo);

imprimir("Precio menor : S/. " + precioMinimo);

}

double calcularPromedio(double a, double b, double c, double d, double e) {

return (a + b + c + d + e) / 5;

}

double calcularMinimo (double a, double b, double c, double d, double e) {

double minimo = a;

if (minimo > b)

minimo = b;

if (minimo > c)

minimo = c;

if (minimo > d)

minimo = d;

if (minimo > e)

minimo = e;

return minimo;

}

double calcularMaximo (double a, double b, double c, double d, double e) {

double maximo = a;

if (maximo < b)

maximo = b;

if (maximo < c)

maximo = c;

if (maximo < d)

maximo = d;

if (maximo < e)

maximo = e;

return maximo;

}

//---

void imprimir(String s) {

txtS.append(s + "\n");

}

protected void actionPerformedBtnCerrar(ActionEvent e) {

dispose();

}

}

* **Dialogo vender :**

public class DialogoVender extends JDialog implements ActionListener {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private final JPanel contentPanel = new JPanel();

private JLabel lblModelo;

private JComboBox cboModelo;

private JLabel lblPrecio;

private JTextField txtPrecio;

private JLabel lblCantidad;

private JTextField txtCantidad;

private JButton btnVender;

private JButton btnCerrar;

private JScrollPane scrollPane;

private JTextArea txtS;

private double aPagar;

public static int

cantidadVentasModelo0 = 0,

cantidadVentasModelo1 = 0,

cantidadVentasModelo2 = 0,

cantidadVentasModelo3 = 0,

cantidadVentasModelo4 = 0,

discosVendidosModelo0 = 0,

discosVendidosModelo1 = 0,

discosVendidosModelo2 = 0,

discosVendidosModelo3 = 0,

discosVendidosModelo4 = 0;

public static double

impAcumuladoModelo0 = 0.0,

impAcumuladoModelo1 = 0.0,

impAcumuladoModelo2 = 0.0,

impAcumuladoModelo3 = 0.0,

impAcumuladoModelo4 = 0.0;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

try {

DialogoVender dialog = new DialogoVender();

dialog.setDefaultCloseOperation(JDialog.DISPOSE\_ON\_CLOSE);

dialog.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

public DialogoVender() {

setTitle("Vender");

setBounds(100, 100, 500, 400);

getContentPane().setLayout(new BorderLayout());

contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));

getContentPane().add(contentPanel, BorderLayout.CENTER);

contentPanel.setLayout(null);

lblModelo = new JLabel("Modelo");

lblModelo.setBounds(10, 11, 65, 14);

contentPanel.add(lblModelo);

cboModelo = new JComboBox();

cboModelo.addActionListener(this);

cboModelo.setModel(new DefaultComboBoxModel(new String[] {"Western Digital My Cloud Home 6TB", "Western Digital Elements 4TB", "Toshiba Canvio Basics 4TB", "Seagate Expansion 4TB", "Seagate Backup Plus Hub 10TB"}));

cboModelo.setBounds(85, 7, 241, 22);

contentPanel.add(cboModelo);

lblPrecio = new JLabel("Precio (S/.)");

lblPrecio.setBounds(10, 43, 65, 14);

contentPanel.add(lblPrecio);

txtPrecio = new JTextField(Proyecto.precio0 + "");

txtPrecio.setEditable(false);

txtPrecio.setBounds(85, 40, 241, 20);

contentPanel.add(txtPrecio);

txtPrecio.setColumns(10);

lblCantidad = new JLabel("Cantidad");

lblCantidad.setBounds(10, 68, 65, 14);

contentPanel.add(lblCantidad);

txtCantidad = new JTextField();

txtCantidad.setColumns(10);

txtCantidad.setBounds(85, 71, 241, 20);

contentPanel.add(txtCantidad);

btnVender = new JButton("Vender");

btnVender.addActionListener(this);

btnVender.setBounds(385, 7, 89, 23);

contentPanel.add(btnVender);

btnCerrar = new JButton("Cerrar");

btnCerrar.addActionListener(this);

btnCerrar.setBounds(385, 39, 89, 23);

contentPanel.add(btnCerrar);

scrollPane = new JScrollPane();

scrollPane.setBounds(10, 109, 464, 241);

contentPanel.add(scrollPane);

txtS = new JTextArea();

txtS.setFont(new Font("Monospaced", Font.PLAIN, 13));

scrollPane.setViewportView(txtS);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cboModelo) {

actionPerformedCboModelo(e);

}

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

actionPerformedBtnVender(e);

}

if (e.getSource() == btnCerrar) {

actionPerformedBtnCerrar(e);

}

}

private int contador = 1;

private int acumuladorVentas = 1;

private double porcentaje = 0;

protected void actionPerformedBtnVender(ActionEvent e) {

//Variables

int obsequio, cantidad;

double precio, importeCompra, descuento;

String modelo;

cantidad = leerCantidad();

//Selección

switch (cboModelo.getSelectedIndex()) {

case 0:

modelo = Proyecto.modelo0;

precio = Proyecto.precio0;

importeCompra = precio \* cantidad;

break;

case 1:

modelo = Proyecto.modelo1;

precio = Proyecto.precio1;

importeCompra = precio \* cantidad;

break;

case 2:

modelo = Proyecto.modelo2;

precio = Proyecto.precio2;

importeCompra = precio \* cantidad;

break;

case 3:

modelo = Proyecto.modelo3;

precio = Proyecto.precio3;

importeCompra = precio \* cantidad;

break;

default:

modelo = Proyecto.modelo4;

precio = Proyecto.precio4;

importeCompra = precio \* cantidad;

}

//Procesos

descuento = calcularDescuento(cantidad, importeCompra);

aPagar = calcularImpCompra(importeCompra, descuento);

obsequio = calcularObsequios(cantidad);

calcularImpTotalGeneral(cantidad);

//Resultados

imprimirResultados(modelo, precio, cantidad, importeCompra, descuento, obsequio);

}

double calcularDescuento(int can, double impCompra) {

double des;

//Descuento

if (can <= 5)

des = Proyecto.porcentaje1/100 \* impCompra;

else

if (can <= 10)

des = Proyecto.porcentaje2/100 \* impCompra;

else

if (can <= 15)

des = Proyecto.porcentaje3/100 \* impCompra;

else

des = Proyecto.porcentaje4/100 \* impCompra;

return des;

}

double calcularImpCompra(double impCompra, double des) {

double aPag;

//A pagar

return aPag = impCompra - des;

}

int calcularObsequios(int can) {

int obs;

//Obsequios

if (can <= 5)

obs = Proyecto.obsequioCantidad1 \* can;

else

if (can <= 10)

obs = Proyecto.obsequioCantidad2 \* can;

else

obs = Proyecto.obsequioCantidad3 \* can;

return obs;

}

void calcularImpTotalGeneral(int can) {

//Alerta > ImporteTotalGeneral Acumulado

switch (cboModelo.getSelectedIndex()) {

case 0:

cantidadVentasModelo0 ++;

discosVendidosModelo0 += can;

impAcumuladoModelo0 += aPagar;

break;

case 1:

cantidadVentasModelo1 ++;

discosVendidosModelo1 += can;

impAcumuladoModelo1 += aPagar;

break;

case 2:

cantidadVentasModelo2 ++;

discosVendidosModelo2 += can;

impAcumuladoModelo2 += aPagar;

break;

case 3:

cantidadVentasModelo3 ++;

discosVendidosModelo3 += can;

impAcumuladoModelo3 += aPagar;

break;

default:

cantidadVentasModelo4 ++;

discosVendidosModelo4 += can;

impAcumuladoModelo4 += aPagar;

}

double importeAcumuladoGeneral =

impAcumuladoModelo0 +

impAcumuladoModelo1 +

impAcumuladoModelo2 +

impAcumuladoModelo3 +

impAcumuladoModelo4;

//Alerta > Porcentaje

porcentaje = ((importeAcumuladoGeneral\*100)/Proyecto.cuotaDiaria);

//Alerta > Contador

if (contador != 5 ) {

contador ++;

acumuladorVentas ++;

}

else {

mensaje("Venta Nro. " + acumuladorVentas + "\n" + "Importe total general acumulado: " + Math.round(importeAcumuladoGeneral\*100.0)/100.0 + "\n" + "Porcentaje de la cuota diara: " + Math.round(porcentaje\*100.0)/100.0 + "%");

contador = 0;

}

}

void imprimirResultados(String mod, double pre, int can, double impCompra, double des, int obs) {

//Imprimir

txtS.setText("");

imprimir("BOLETA DE VENTA" + "\n");

imprimir("Modelo : " + mod);

imprimir("Precio : S/. " + pre);

imprimir("Cantidad Adquirida : " + can);

imprimir("Importe Compra : S/. " + impCompra);

imprimir("Importe Descuento : S/. " + des);

imprimir("Importe Pagar : S/. " + aPagar);

imprimir("Tipo Obsequio : " + Proyecto.tipoObsequio);

imprimir("Unidades obsequiadas : " + obs);

}

protected void actionPerformedBtnCerrar(ActionEvent e) {

dispose();

}

protected void actionPerformedCboModelo(ActionEvent e) {

int posModelo = cboModelo.getSelectedIndex();

switch (posModelo) {

case 0:

txtPrecio.setText(Proyecto.precio0 + "");

break;

case 1:

txtPrecio.setText(Proyecto.precio1 + "");

break;

case 2:

txtPrecio.setText(Proyecto.precio2 + "");

break;

case 3:

txtPrecio.setText(Proyecto.precio3 + "");

break;

default:

txtPrecio.setText(Proyecto.precio4 + "");

}

}

void imprimir(String s) {

txtS.append(s + "\n");

}

int leerCantidad() {

return Integer.parseInt(txtCantidad.getText());

}

void mensaje(String s) {

JOptionPane.showMessageDialog(this, s, "Tienda", 1);

}

}

#### Conclusiones

* El sistema es eficiente
* Fácil de comprender y utilizar
* La configuración es comprensible para el usuario

#### Recomendaciones

* Como recomendación esto lo hemos hecho para todos los negocios que están en una etapa inicial y quieren una herramienta para mejorar la velocidad de ventas y más aspectos del negocio. Además, es fácil de comprender y utilizar, logrando que sea accesible para todo público.

#### Bibliografía

Guías para aplicar formulas Math en Java:

* [https://www.delftstack.com/es/howto/java/how-to-round-a-double-to-two-decimal-places-in-java/#:~:text=El%20m%C3%A9todo%20Math.,(double%20\*%20100.0)%2F100.0%20](https://www.delftstack.com/es/howto/java/how-to-round-a-double-to-two-decimal-places-in-java/#:~:text=El%20m%C3%A9todo%20Math.,(double%20*%20100.0)%2F100.0%20).

* <https://docs.oracle.com/javase/7/docs/api/java/math/RoundingMode.html>