

**INSTITUTO SUPERIOR TECNOLOGICO PRIVADO CIBERTEC**

**CARRERAS PROFESIONALES**

**SISTEMA DE INGRESO, ATENCIÓN Y HOSPEDAJE**

**ALGORITMIA Y ESTRUCTURA DE DATOS**

**Profesor:**

Carlos Rolando Mendo Bravo

**SEGUNDO CICLO**

**SECCIÓN T1AM**

**SEMESTRE 2019 – 02**

**Coordinador:**

Richard Saul Rodas Pita

**Integrantes:**

Luis Gustavo Torpoco Vega

Cesa Augusto Medina Ortiz

Bruno Joaquín Coronado Bustamane

Andres Fabian Mondoñedo Diaz

Moises Aldair Gonzales Montes

**CARRERAS PROFESIONALES**

**CIBERTEC**

**MIRAFLORES, 2019**

#### Índice

|  |  |
| --- | --- |
| Introducción……………………...3 |  |
| Justificación……………….……...4 |  |
| Objetivos……………………….…5 Entregables…………………………………………7 |  |
| Definición………………………… |  |
| Conclusiones…………………… |  |
| Recomendaciones……………… |  |
| Bibliografía…………………….… |  |
|  |  |

#### Introducción

El presente proyecto se refiere al servicio de hospedaje, desarrollado en lenguaje Java con ayuda de el entorno Eclipse.

Lo que implica: La recolección de datos para su posterior proceso, con la finalidad de ser eficiente en sus procesos respectivos.

Dentro de el proceso de alquiler para que el usuario pueda alquilar eficientemente, se utilizarán diversas cajas de diálogos, en las cuales se tiene:

Mantenimiento

-Socio

-Producto

-Bungalow

Registro

-Ingreso

-Consumo

-Hospedaje

Pago

-Ingresos y consumos

-Hospedajes

Reporte

-Ingresos y consumos pendientes

-Ingresos y consumos pagados

-Hospedajes pendientes

-Hospedajes pagados

Boleta

-Boletas

Dentro de las limitaciones de este programa se tiene:

La no existencia del botón “Imprimir” dentro de la caja de dialogo boleta, el cual daría al cliente un formato físico de la boleta o comprobante pago.

#### Justificación

1.-Aplicabilidad:

-En las empresas es acelerar el proceso de servicio y recaudar asi la información de alquileres generados, optimizando informes y minimizando errores.

-Brindarle al cliente información sobre la calidad del servicio, especificaciones sobre este, etc.

-En la sociedad la aplicabilidad de este programa en pequeñas y grandes empresas daría virtud del desarrollo tecnológico en cuanto a programación orientada a objetos.

2.-Como la solución contribuye positivamente en la mejora del proceso:

-Acelera el proceso de la venta del servicio.

-Minimiza los errores.

-Satisfacción por parte del cliente .

3.-Beneficiarios Directos:

-Las empresas que brindan servicios de alojamiento.

-Los programadores quienes comersializan el programa como producto final.

4.-Beneficiarios Indirectos:

-El cliente..

#### Objetivos

* Por un lado el objetivo del proyecto es eficientizar el proceso de venta y registro .
* Por el otro es mantener una forma organizada de las operaciones de pago, reporte y boleta de venta.

MENU PRINCIPAL

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuItem;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.UIManager;

**public** **class** Proyecto **extends** JFrame **implements** ActionListener {

**private** JLabel lblFondo;

**private** JPanel contentPane;

**private** JMenuBar menuBar;

**private** JMenu mnNewMenu;

**private** JMenu mnNewMenu\_1;

**private** JMenu mnNewMenu\_2;

**private** JMenu mnNewMenu\_3;

**private** JMenu mnNewMenu\_4;

**private** JMenuItem mntmSocio;

**private** JMenuItem mntmProducto;

**private** JMenuItem mntmBungalow;

**private** JMenuItem mntmIngreso;

**private** JMenuItem mntmConsum;

**private** JMenuItem mntmNewMenuItem;

**private** JMenuItem mntmNewMenuItem\_1;

**private** JMenuItem mntmHospedajes;

**private** JMenuItem mntmNewMenuItem\_2;

**private** JMenuItem mntmNewMenuItem\_3;

**private** JMenuItem mntmNewMenuItem\_4;

**private** JMenuItem mntmHospedajesPagados;

**private** JMenuItem mntmBoletas;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

**try** {

UIManager.*setLookAndFeel*("javax.swing.plaf.nimbus.NimbusLookAndFeel");

} **catch** (Throwable e) {

e.printStackTrace();

}

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() {

setResizable(**false**);

setTitle("HOTEL HILTON® >>> TheNyGys ® >>");

setIconImage(**new** ImageIcon("imagenes/logo.png").getImage());

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setSize(748, 560);

**this**.setLocationRelativeTo(**null**);

menuBar = **new** JMenuBar();

setJMenuBar(menuBar);

mnNewMenu = **new** JMenu("Mantenimiento ");

menuBar.add(mnNewMenu);

mntmSocio = **new** JMenuItem("Socio");

mntmSocio.addActionListener(**this**);

mnNewMenu.add(mntmSocio);

mntmProducto = **new** JMenuItem("Producto");

mntmProducto.addActionListener(**this**);

mnNewMenu.add(mntmProducto);

mntmBungalow = **new** JMenuItem("Bungalow");

mntmBungalow.addActionListener(**this**);

mnNewMenu.add(mntmBungalow);

mnNewMenu\_1 = **new** JMenu("Registro");

menuBar.add(mnNewMenu\_1);

mntmIngreso = **new** JMenuItem("Ingreso");

mntmIngreso.addActionListener(**this**);

mnNewMenu\_1.add(mntmIngreso);

mntmConsum = **new** JMenuItem("Consumo");

mntmConsum.addActionListener(**this**);

mnNewMenu\_1.add(mntmConsum);

mntmNewMenuItem = **new** JMenuItem("Hospedaje");

mntmNewMenuItem.addActionListener(**this**);

mnNewMenu\_1.add(mntmNewMenuItem);

mnNewMenu\_2 = **new** JMenu("Pago");

menuBar.add(mnNewMenu\_2);

mntmNewMenuItem\_1 = **new** JMenuItem("Ingresos y Consumos");

mntmNewMenuItem\_1.addActionListener(**this**);

mnNewMenu\_2.add(mntmNewMenuItem\_1);

mntmHospedajes = **new** JMenuItem("Hospedajes");

mntmHospedajes.addActionListener(**this**);

mnNewMenu\_2.add(mntmHospedajes);

mnNewMenu\_3 = **new** JMenu("Reporte");

menuBar.add(mnNewMenu\_3);

mntmNewMenuItem\_2 = **new** JMenuItem("Ingresos y Consumos pendientes");

mntmNewMenuItem\_2.addActionListener(**this**);

mnNewMenu\_3.add(mntmNewMenuItem\_2);

mntmNewMenuItem\_3 = **new** JMenuItem("Ingresos y Consumos Pagados");

mntmNewMenuItem\_3.addActionListener(**this**);

mnNewMenu\_3.add(mntmNewMenuItem\_3);

mntmNewMenuItem\_4 = **new** JMenuItem("Hospedajes Pendientes");

mntmNewMenuItem\_4.addActionListener(**this**);

mnNewMenu\_3.add(mntmNewMenuItem\_4);

mntmHospedajesPagados = **new** JMenuItem("Hospedajes Pagados");

mntmHospedajesPagados.addActionListener(**this**);

mnNewMenu\_3.add(mntmHospedajesPagados);

mnNewMenu\_4 = **new** JMenu("Boleta");

menuBar.add(mnNewMenu\_4);

mntmBoletas = **new** JMenuItem("Boletas");

mntmBoletas.addActionListener(**this**);

mnNewMenu\_4.add(mntmBoletas);

contentPane = **new** JPanel();

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

setContentPane(contentPane);

lblFondo = **new** JLabel(**new** ImageIcon("imagenes/The-Muraka.jpg"));

lblFondo.setBounds(0, 0, 750, 500);

getContentPane().add(lblFondo);

contentPane.setLayout(**null**);

}

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == mntmBoletas) {

actionPerformedMntmBoletas(arg0);

}

**if** (arg0.getSource() == mntmHospedajesPagados) {

actionPerformedMntmHospedajesPagados(arg0);

}

**if** (arg0.getSource() == mntmNewMenuItem\_4) {

actionPerformedMntmNewMenuItem\_4(arg0);

}

**if** (arg0.getSource() == mntmNewMenuItem\_3) {

actionPerformedMntmNewMenuItem\_3(arg0);

}

**if** (arg0.getSource() == mntmNewMenuItem\_2) {

actionPerformedMntmNewMenuItem\_2(arg0);

}

**if** (arg0.getSource() == mntmHospedajes) {

actionPerformedMntmHospedajes(arg0);

}

**if** (arg0.getSource() == mntmNewMenuItem\_1) {

actionPerformedMntmNewMenuItem\_1(arg0);

}

**if** (arg0.getSource() == mntmNewMenuItem) {

actionPerformedMntmNewMenuItem(arg0);

}

**if** (arg0.getSource() == mntmConsum) {

actionPerformedMntmConsum(arg0);

}

**if** (arg0.getSource() == mntmIngreso) {

actionPerformedMntmIngreso(arg0);

}

**if** (arg0.getSource() == mntmBungalow) {

actionPerformedMntmBungalow(arg0);

}

**if** (arg0.getSource() == mntmProducto) {

actionPerformedMntmProducto(arg0);

}

**if** (arg0.getSource() == mntmSocio) {

actionPerformedMntmSocio(arg0);

}

}

**protected** **void** actionPerformedMntmSocio(ActionEvent arg0) {

JdSocio jds=**new** JdSocio();

jds.setLocationRelativeTo(**this**);

jds.setVisible(**true**);

}

**protected** **void** actionPerformedMntmProducto(ActionEvent arg0) {

JdProducto jdp=**new** JdProducto();

jdp.setLocationRelativeTo(**this**);

jdp.setVisible(**true**);

}

**protected** **void** actionPerformedMntmBungalow(ActionEvent arg0) {

JdBungalow jdb=**new** JdBungalow();

jdb.setLocationRelativeTo(**this**);

jdb.setVisible(**true**);

}

**protected** **void** actionPerformedMntmIngreso(ActionEvent arg0) {

JdIngreso jdi=**new** JdIngreso();

jdi.setLocationRelativeTo(**this**);

jdi.setVisible(**true**);

}

**protected** **void** actionPerformedMntmConsum(ActionEvent arg0) {

JdConsumo jdc=**new** JdConsumo();

jdc.setLocationRelativeTo(**this**);

jdc.setVisible(**true**);

}

**protected** **void** actionPerformedMntmNewMenuItem(ActionEvent arg0) {

JdHospedaje jdh=**new** JdHospedaje();

jdh.setLocationRelativeTo(**this**);

jdh.setVisible(**true**);

}

**protected** **void** actionPerformedMntmNewMenuItem\_1(ActionEvent arg0) {

JdIngresosyConsumos jdiyc=**new** JdIngresosyConsumos();

jdiyc.setLocationRelativeTo(**this**);

jdiyc.setVisible(**true**);

}

**protected** **void** actionPerformedMntmHospedajes(ActionEvent arg0) {

JdHospedajes jdhs=**new** JdHospedajes();

jdhs.setLocationRelativeTo(**this**);

jdhs.setVisible(**true**);

}

**protected** **void** actionPerformedMntmNewMenuItem\_2(ActionEvent arg0) {

JdIngresosyConsumosPendientes jdiycp=**new** JdIngresosyConsumosPendientes();

jdiycp.setLocationRelativeTo(**this**);

jdiycp.setVisible(**true**);

}

**protected** **void** actionPerformedMntmNewMenuItem\_3(ActionEvent arg0) {

JdIngresosyConsumosPagados jdiycpa =**new** JdIngresosyConsumosPagados();

jdiycpa.setLocationRelativeTo(**this**);

jdiycpa.setVisible(**true**);

}

**protected** **void** actionPerformedMntmNewMenuItem\_4(ActionEvent arg0) {

JdHospedajesPendientes jdhp =**new** JdHospedajesPendientes();

jdhp.setLocationRelativeTo(**this**);

jdhp.setVisible(**true**);

}

**protected** **void** actionPerformedMntmHospedajesPagados(ActionEvent arg0) {

JdHospedajesPagados jdhpa =**new** JdHospedajesPagados();

jdhpa.setLocationRelativeTo(**this**);

jdhpa.setVisible(**true**);

}

**protected** **void** actionPerformedMntmBoletas(ActionEvent arg0) {

JdBoletas jdb =**new** JdBoletas();

jdb.setLocationRelativeTo(**this**);

jdb.setVisible(**true**);

}

}

MANTENIMIENTO

SOCIO

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.table.TableColumnModel;

**import** arreglos.ArregloSocio;

**import** clases.Socio;

**import** javax.swing.JComboBox;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JButton;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.DefaultComboBoxModel;

**import** javax.swing.ImageIcon;

**public** **class** JdSocio **extends** JDialog **implements** ActionListener {

**private** JLabel lblImgSocio;

**private** JLabel lblNewLabel;

**private** JTextField txtCodigo;

**private** JComboBox<String> cboOperacion;

**private** JLabel lblNombre;

**private** JTextField txtNombre;

**private** JLabel lblApellidos;

**private** JTextField txtApellido;

**private** JLabel lblDni;

**private** JTextField txtDni;

**private** JTextField txtTelefono;

**private** JLabel lblTelefono;

**private** JScrollPane scrollPane;

**private** JTable tblSocio;

**private** JButton btnOk;

**private** DefaultTableModel modelo;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdSocio dialog = **new** JdSocio();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdSocio() {

setTitle("Mantenimiento||Socio");

setBounds(100, 100, 700, 400);

getContentPane().setLayout(**null**);

lblImgSocio = **new** JLabel();

lblImgSocio.setIcon(**new** ImageIcon("imagenes/socio.png"));

lblImgSocio.setBounds(412, 42, 88, 89);

getContentPane().add(lblImgSocio);

lblNewLabel = **new** JLabel("Codigo");

lblNewLabel.setBounds(427, 13, 56, 16);

getContentPane().add(lblNewLabel);

txtCodigo = **new** JTextField();

txtCodigo.setEditable(**false**);

txtCodigo.setBounds(475, 10, 94, 22);

getContentPane().add(txtCodigo);

txtCodigo.setColumns(10);

cboOperacion = **new** JComboBox<String>();

cboOperacion.setModel(**new** DefaultComboBoxModel<String>(**new** String[] { "Adicionar", "Consultar", "Modificar", "Eliminar" }));

cboOperacion.addActionListener(**this**);

cboOperacion.setBounds(581, 10, 91, 22);

getContentPane().add(cboOperacion);

lblNombre = **new** JLabel("Nombre");

lblNombre.setBounds(12, 13, 56, 16);

getContentPane().add(lblNombre);

txtNombre = **new** JTextField();

txtNombre.setBounds(80, 10, 116, 22);

getContentPane().add(txtNombre);

txtNombre.setColumns(10);

lblApellidos = **new** JLabel("Apellidos");

lblApellidos.setBounds(12, 48, 56, 16);

getContentPane().add(lblApellidos);

txtApellido = **new** JTextField();

txtApellido.setBounds(80, 45, 116, 22);

getContentPane().add(txtApellido);

txtApellido.setColumns(10);

lblDni = **new** JLabel("Dni");

lblDni.setBounds(12, 77, 56, 16);

getContentPane().add(lblDni);

txtDni = **new** JTextField();

txtDni.setBounds(80, 74, 116, 22);

getContentPane().add(txtDni);

txtDni.setColumns(10);

txtTelefono = **new** JTextField();

txtTelefono.setBounds(80, 109, 116, 22);

getContentPane().add(txtTelefono);

txtTelefono.setColumns(10);

lblTelefono = **new** JLabel("Telefono");

lblTelefono.setBounds(12, 112, 56, 16);

getContentPane().add(lblTelefono);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 141, 655, 298);

getContentPane().add(scrollPane);

tblSocio = **new** JTable();

tblSocio.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tblSocio);

btnOk = **new** JButton("OK");

btnOk.addActionListener(**this**);

btnOk.setBounds(232, 48, 97, 25);

getContentPane().add(btnOk);

modelo = **new** DefaultTableModel();

modelo.addColumn("CODIGO");

modelo.addColumn("NOMBRE");

modelo.addColumn("APELLIDO");

modelo.addColumn("DNI");

modelo.addColumn("TELEFONO");

tblSocio.setModel(modelo);

btnOk.setEnabled(**false**);

ajustarAnchoColumnas();

listar();

}

// Declaracion Global

ArregloSocio as = **new** ArregloSocio();

// CODIGO A CONSULTAR

**int** codigoConsultado = as.obtener(0).getCodigoSocio();

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == btnOk) {

actionPerformedBtnOk(arg0);

}

**if** (arg0.getSource() == cboOperacion) {

actionPerformedCboOperacion(arg0);

}

}

**protected** **void** actionPerformedCboOperacion(ActionEvent arg0) {

btnOk.setEnabled(**true**);

**int** posOperacion = leerPosOperacion();

**switch** (posOperacion) {

**case** 0:

**if** (as.tamaño() == 0)

btnOk.setEnabled(**true**);

txtCodigo.setEditable(**false**);

txtCodigo.setText("" + as.codigoCorrelativo());

txtNombre.setText("");

txtApellido.setText("");

txtDni.setText("");

txtTelefono.setText("");

txtNombre.requestFocus();

HabilitarEntradas(**true**);

txtNombre.requestFocus();

**break**;

**case** 1:

**if** (as.tamaño() > 0)

txtCodigo.setEditable(**true**);

txtCodigo.setText("" + codigoConsultado);

HabilitarEntradas(**false**);

consultarSocio();

txtCodigo.requestFocus();

**break**;

**case** 2:

txtCodigo.setEditable(**false**);

txtCodigo.setText("" + codigoConsultado);

HabilitarEntradas(**true**);

consultarSocio();

txtNombre.requestFocus();

**break**;

**case** 3:

txtCodigo.setEditable(**false**);

txtCodigo.setText("" + codigoConsultado);

HabilitarEntradas(**false**);

consultarSocio();

}

}

**protected** **void** actionPerformedBtnOk(ActionEvent arg0) {

**int** posOperacion = leerPosOperacion();

**switch** (posOperacion) {

**case** 0:

adicionarSocio();

**break**;

**case** 1:

consultarSocio();

**break**;

**case** 2:

modificarSocio();

**break**;

**case** 3:

eliminarSocio();

**break**;

}

}

**void** adicionarSocio() {

btnOk.setEnabled(**true**);

**int** codigo = leerCodigo();

String nombre = leerNombre();

String apellido = leerApellido();

String dni = leerDni();

**int** telefono = leerTelefono();

Socio nuevo = **new** Socio(codigo, nombre, apellido, dni, telefono);

as.adicionar(nuevo);

listar();

txtCodigo.setText("" + as.codigoCorrelativo());

txtNombre.setText("");

txtApellido.setText("");

txtDni.setText("");

txtTelefono.setText("");

}

**void** consultarSocio() {

**try** {

Socio x = as.buscar(leerCodigo());

**if** (x != **null**) {

codigoConsultado = x.getCodigoSocio();

txtNombre.setText(x.getNombres());

txtApellido.setText(x.getApellidos());

txtDni.setText(x.getDni());

txtTelefono.setText("" + x.getTelefono());

txtCodigo.requestFocus();

}

**else**

error("El codigo" + leerCodigo() + "no existe", txtCodigo);

}

**catch** (Exception e) {

error("INGRESE CODIGO A CONSULTAR", txtCodigo);

}

}

**void** modificarSocio() {

btnOk.setEnabled(**true**);

Socio x = as.buscar(codigoConsultado);

String nombres = leerNombre();

**if** (nombres.length() > 0) {

String apellidos = leerApellido();

String dni = leerDni();

**try** {

**int** telefono = leerTelefono();

x.setNombres(nombres);

x.setApellidos(apellidos);

x.setDni(dni);

x.setTelefono(telefono);

listar();

}

**catch** (Exception e) {

error("INGRESE EL TELEFONO CORRECTO", txtTelefono);

}

}

**else**

error("IGRESE EL NOMBR CORRECTO", txtNombre);

}

**void** eliminarSocio() {

btnOk.setEnabled(**true**);

**int** ok = confirmar("¿Desea eliminar el registro?");

**if** (ok == 0) {

Socio x = as.buscar(codigoConsultado);

as.eliminar(x);

listar();

**if** (as.tamaño() > 0) {

codigoConsultado = as.obtener(0).getCodigoSocio();

txtCodigo.setText("" + codigoConsultado);

consultarSocio();

}

**else** {

codigoConsultado = as.codigoCorrelativo();

cboOperacion.setSelectedIndex(0);

}

}

}

// METODOS VOID (SIN PARAMETROS)

**void** ajustarAnchoColumnas() {

TableColumnModel tcm = tblSocio.getColumnModel();

tcm.getColumn(0).setPreferredWidth(anchoColumna(8)); // codigo

tcm.getColumn(1).setPreferredWidth(anchoColumna(18)); // nombre

tcm.getColumn(2).setPreferredWidth(anchoColumna(10)); // apellido

tcm.getColumn(3).setPreferredWidth(anchoColumna(10)); // dni

tcm.getColumn(4).setPreferredWidth(anchoColumna(15)); // telefono

}

**void** listar() {

Socio x;

modelo.setRowCount(0);

**for** (**int** i = 0; i < as.tamaño(); i++) {

x = as.obtener(i);

Object[] fila = { x.getCodigoSocio(), x.getNombres(), x.getApellidos(), x.getDni(), x.getTelefono() };

modelo.addRow(fila);

}

}

// METODOS VOID CON PARAMETROS//

**void** HabilitarEntradas(**boolean** sino) {

txtNombre.setEditable(sino);

txtDni.setEditable(sino);

txtApellido.setEditable(sino);

txtTelefono.setEditable(sino);

}

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Informacion", 0);

}

**void** error(String s, JTextField txt) {

mensaje(s);

txt.setText("");

txt.requestFocus();

}

// METODOS QUE RETORNAN VALOR (SIN PARAMETRO)

**int** leerCodigo() {

**return** Integer.*parseInt*(txtCodigo.getText().trim());

}

String leerNombre() {

**return** txtNombre.getText().trim();

}

String leerApellido() {

**return** txtApellido.getText().trim();

}

String leerDni() {

**return** txtDni.getText().trim();

}

**int** leerTelefono() {

**return** Integer.*parseInt*(txtTelefono.getText().trim());

}

**int** leerPosOperacion() {

**return** cboOperacion.getSelectedIndex();

}

/// METODOS QUE RETORNAN VALOR

**int** anchoColumna(**int** porcentaje) {

**return** porcentaje \* scrollPane.getWidth() / 100;

}

String ajustar(**double** numero) {

**return** String.*format*("5.2f", numero);

}

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

}

PRODUCTO**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.table.TableColumnModel;

**import** arreglos.ArregloProducto;

**import** clases.Producto;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.DefaultComboBoxModel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**public** **class** JdProducto **extends** JDialog **implements** ActionListener {

**private** JLabel lblProducto;

**private** JLabel lblDetalle;

**private** JLabel lblPrecio;

**private** JLabel lblStock;

**private** JTextField txtCodigo;

**private** JTextField txtDetalle;

**private** JTextField txtPrecio;

**private** JTextField txtStock;

**private** JButton btnOk;

**private** JComboBox<String> cboOperacion;

**private** JScrollPane scrollPane;

**private** JTable tblProducto;

**private** DefaultTableModel modelo;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdProducto dialog = **new** JdProducto();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdProducto() {

setBounds(100, 100, 549, 408);

setTitle("Mantenimiento||Producto");

getContentPane().setLayout(**null**);

lblProducto = **new** JLabel("Codigo");

lblProducto.setBounds(12, 13, 56, 16);

getContentPane().add(lblProducto);

lblDetalle = **new** JLabel("Detalle");

lblDetalle.setBounds(12, 42, 56, 16);

getContentPane().add(lblDetalle);

lblPrecio = **new** JLabel("Precio");

lblPrecio.setBounds(12, 71, 56, 16);

getContentPane().add(lblPrecio);

lblStock = **new** JLabel("Stock");

lblStock.setBounds(12, 100, 56, 16);

getContentPane().add(lblStock);

txtCodigo = **new** JTextField();

txtCodigo.setEditable(**false**);

txtCodigo.setBounds(60, 10, 87, 22);

getContentPane().add(txtCodigo);

txtCodigo.setColumns(10);

txtDetalle = **new** JTextField();

txtDetalle.setBounds(60, 42, 116, 22);

getContentPane().add(txtDetalle);

txtDetalle.setColumns(10);

txtPrecio = **new** JTextField();

txtPrecio.setBounds(60, 71, 116, 22);

getContentPane().add(txtPrecio);

txtPrecio.setColumns(10);

txtStock = **new** JTextField();

txtStock.setBounds(60, 97, 116, 22);

getContentPane().add(txtStock);

txtStock.setColumns(10);

btnOk = **new** JButton("OK");

btnOk.addActionListener(**this**);

btnOk.setBounds(203, 42, 97, 25);

getContentPane().add(btnOk);

cboOperacion = **new** JComboBox<String>();

cboOperacion.addActionListener(**this**);

cboOperacion.setModel(**new** DefaultComboBoxModel<String>(**new** String[] {"Adicionar", "Consultar", "Modificar", "Eliminar"}));

cboOperacion.setBounds(340, 10, 147, 22);

getContentPane().add(cboOperacion);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 137, 507, 211);

getContentPane().add(scrollPane);

tblProducto = **new** JTable();

tblProducto.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tblProducto);

modelo=**new** DefaultTableModel();

modelo.addColumn("Codigo");

modelo.addColumn("Detalle");

modelo.addColumn("Precio");

modelo.addColumn("Stock");

tblProducto.setModel(modelo);

btnOk.setEnabled(**false**);

ajustarAnchoColumnas();

listar();

}

//DECLARACION GLOBAL

ArregloProducto ap=**new** ArregloProducto();

//Codigo Consultar

**int** codigoConsultado=ap.obtener(0).getCodigoProducto();

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == cboOperacion) {

actionPerformedCboOperacion(arg0);

}

**if** (arg0.getSource() == btnOk) {

actionPerformedBtnOk(arg0);

}

}

**protected** **void** actionPerformedBtnOk(ActionEvent arg0) {

**int** posOperacion=leerPosOperacion();

**switch**(posOperacion){

**case** 0:

adicionarProducto();

**break**;

**case** 1:

consultarProducto();

**break**;

**case** 2:

modificarProducto();

**break**;

**case** 3:

eliminarProducto();

**break**;

}

}

**protected** **void** actionPerformedCboOperacion(ActionEvent arg0) {

**int** posOperacion=leerPosOperacion();

**switch**(posOperacion){

**case** 0:

**if**(ap.tamaño()==0)

btnOk.setEnabled(**true**);

txtCodigo.setEditable(**false**);

txtCodigo.setText(""+ap.codigoCorrelativo());

txtDetalle.setText("");

txtPrecio.setText("");

txtStock.setText("");

txtDetalle.requestFocus();

HabilitarEntrada(**true**);

txtDetalle.requestFocus();

**break**;

**case** 1:

**if**(ap.tamaño()>0)

txtCodigo.setEditable(**true**);

txtCodigo.setText(""+codigoConsultado);

HabilitarEntrada(**false**);

consultarProducto();

txtCodigo.requestFocus();

**break**;

**case** 2:

txtCodigo.setEditable(**false**);

txtCodigo.setText(""+codigoConsultado);

HabilitarEntrada(**true**);

consultarProducto();

txtDetalle.requestFocus();

**break**;

**case** 3:

txtCodigo.setEditable(**false**);

txtCodigo.setText(""+codigoConsultado);

HabilitarEntrada(**false**);

consultarProducto();

**break**;

}

}

//METODOS VOID SIN PARAMETOS

**void** ajustarAnchoColumnas(){

TableColumnModel tcm=tblProducto.getColumnModel();

tcm.getColumn(0).setPreferredWidth(anchoColumna( 8)); // codigo

tcm.getColumn(1).setPreferredWidth(anchoColumna(18)); // nombre

tcm.getColumn(2).setPreferredWidth(anchoColumna(10)); // apellido

}

**void** listar(){

Producto x;

modelo.setRowCount(0);

**for**(**int** i=0;i<ap.tamaño();i++){

x=ap.obtener(i);

Object[]fila={x.getCodigoProducto(),

x.getDetalle(),

x.getPrecioUnitario(),

x.getStock()};

modelo.addRow(fila);

}

}

**void** adicionarProducto(){

btnOk.setEnabled(**true**);

**int** codigoProducto=leerCodigoProducto();

**int** stock=leerStock();

**double** preciounitario=leerPrecioUnitario();

String detalle=leerDetalle();

Producto nuevo=**new** Producto(codigoProducto, detalle, preciounitario, stock);

ap.adicionar(nuevo);

listar();

txtCodigo.setText(""+ap.codigoCorrelativo());

txtDetalle.setText("");

txtPrecio.setText("");

txtStock.setText("");

}

**void** consultarProducto(){

btnOk.setEnabled(**true**);

**try**{

Producto x=ap.buscar(leerCodigoProducto());

**if**(x!=**null**){

codigoConsultado=x.getCodigoProducto();

txtDetalle.setText(x.getDetalle());

txtPrecio.setText(""+x.getPrecioUnitario());

txtStock.setText(""+x.getStock());

txtCodigo.requestFocus();

}**else**

error("El codigo"+ leerCodigoProducto()+"no existe",txtCodigo);

}**catch**(Exception e){

error("INGRESE CODIGO A CONSULTAR",txtCodigo);

}

}

**void** modificarProducto(){

btnOk.setEnabled(**true**);

Producto x=ap.buscar(codigoConsultado);

String detalle=leerDetalle();

**double** preciounitario=leerPrecioUnitario();

**int** stock=leerStock();

x.setDetalle(detalle);

x.setPrecioUnitario(preciounitario);

x.setStock(stock);

listar();

}

**void** eliminarProducto(){

btnOk.setEnabled(**true**);

**int** ok=confirmar("¿Desea eliminar el registro?");

**if**(ok==0){

Producto x=ap.buscar(codigoConsultado);

ap.eliminar(x);

listar();

**if**(ap.tamaño()>0){

codigoConsultado=ap.obtener(0).getCodigoProducto();

txtCodigo.setText(""+codigoConsultado);

consultarProducto();

}

**else**{

codigoConsultado=ap.codigoCorrelativo();

cboOperacion.setSelectedIndex(0);

}

}

}

//METODOS VOID CON PARAMETROS

**void** HabilitarEntrada(**boolean** sino){

txtDetalle.setEditable(sino);

txtPrecio.setEditable(sino);

txtStock.setEditable(sino);

}

**void** mensaje(String s){

JOptionPane.*showMessageDialog*(**this**,s,"Informacion",0);

}

**void** error(String s,JTextField txt){

mensaje(s);

txt.setText("");

txt.requestFocus();

}

//METODOS QUE RETORNAN VALOR (SIN PARAMETRO)

**int** leerCodigoProducto(){

**return** Integer.*parseInt*(txtCodigo.getText().trim());

}

**double** leerPrecioUnitario(){

**return** Double.*parseDouble*(txtPrecio.getText().trim());

}

String leerDetalle(){

**return** txtDetalle.getText().trim();

}

**int** leerStock(){

**return** Integer.*parseInt*(txtStock.getText().trim());

}

**int** leerPosOperacion(){

**return** cboOperacion.getSelectedIndex();

}

//METODOS QUE RETORNAN VALOR

**int** anchoColumna(**int** porcentaje){

**return** porcentaje\*scrollPane.getWidth()/100;

}

String ajustar(**double** numero){

**return** String.*format*("5.2",numero);

}

**int** confirmar(String s){

**return** JOptionPane.*showConfirmDialog*(**this**,s,"Alerta",0,1,**null**);

}

}

BUNGALOW

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.table.TableColumnModel;

**import** arreglos.ArregloBungalow;

**import** clases.Bungalow;

**import** libreria.Lib;

**import** javax.swing.JButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.JComboBox;

**import** javax.swing.DefaultComboBoxModel;

**public** **class** JdBungalow **extends** JDialog **implements** ActionListener {

**private** JLabel lblNumeroBungalow;

**private** JLabel lblCategoria;

**private** JLabel lblPrecioXDia;

**private** JLabel lblEstado;

**private** JTextField txtNumBun;

**private** JTextField txtPrecioPorDia;

**private** JButton btnAceptar;

**private** JButton btnAdicionar;

**private** JButton btnModificar;

**private** JButton btnEliminar;

**private** JScrollPane scrollPane;

**private** JTable tblbung;

**private** JButton btnConsultar;

**private** DefaultTableModel modelo;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdBungalow dialog = **new** JdBungalow();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdBungalow() {

setTitle("Mantenimiento||Bungalow");

setBounds(100, 100, 641, 474);

getContentPane().setLayout(**null**);

lblNumeroBungalow = **new** JLabel("Numero Bungalow");

lblNumeroBungalow.setBounds(12, 13, 118, 16);

getContentPane().add(lblNumeroBungalow);

lblCategoria = **new** JLabel("Categoria");

lblCategoria.setBounds(12, 42, 56, 16);

getContentPane().add(lblCategoria);

lblPrecioXDia = **new** JLabel("Precio/Dia");

lblPrecioXDia.setBounds(12, 71, 79, 16);

getContentPane().add(lblPrecioXDia);

lblEstado = **new** JLabel("Estado");

lblEstado.setBounds(12, 100, 56, 16);

getContentPane().add(lblEstado);

txtNumBun = **new** JTextField();

txtNumBun.setBounds(124, 10, 116, 22);

getContentPane().add(txtNumBun);

txtNumBun.setColumns(10);

txtPrecioPorDia = **new** JTextField();

txtPrecioPorDia.setBounds(124, 68, 116, 22);

getContentPane().add(txtPrecioPorDia);

txtPrecioPorDia.setColumns(10);

btnAceptar = **new** JButton("Aceptar");

btnAceptar.addActionListener(**this**);

btnAceptar.setBounds(339, 96, 97, 25);

getContentPane().add(btnAceptar);

btnAdicionar = **new** JButton("Adicionar");

btnAdicionar.addActionListener(**this**);

btnAdicionar.setBounds(496, 9, 97, 25);

getContentPane().add(btnAdicionar);

btnModificar = **new** JButton("Modificar");

btnModificar.addActionListener(**this**);

btnModificar.setBounds(496, 38, 97, 25);

getContentPane().add(btnModificar);

btnEliminar = **new** JButton("Eliminar");

btnEliminar.addActionListener(**this**);

btnEliminar.setBounds(496, 96, 97, 25);

getContentPane().add(btnEliminar);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 141, 599, 273);

getContentPane().add(scrollPane);

tblbung = **new** JTable();

tblbung.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tblbung);

btnConsultar = **new** JButton("Consultar");

btnConsultar.addActionListener(**this**);

btnConsultar.setBounds(496, 67, 97, 25);

getContentPane().add(btnConsultar);

modelo = **new** DefaultTableModel();

modelo.addColumn("NUMERO BUNGALOW");

modelo.addColumn("CATEGORIA");

modelo.addColumn("ESTADO");

modelo.addColumn("PRECIO/DIA");

tblbung.setModel(modelo);

cboCategoria = **new** JComboBox<String>();

cboCategoria.setModel(**new** DefaultComboBoxModel<String>(Lib.*CategoriaBungalow*));

cboCategoria.setBounds(124, 40, 118, 16);

getContentPane().add(cboCategoria);

cboEstado = **new** JComboBox<String>();

cboEstado.setModel(**new** DefaultComboBoxModel<String>(Lib.*estadosBungalow*));

cboEstado.setBounds(124, 98, 116, 18);

getContentPane().add(cboEstado);

habilitarEntradas(**false**);

ajustarAnchoColumnas();

listar();

editarFila();

}

// DEC GLOBAL

ArregloBungalow ab = **new** ArregloBungalow();

**int** numeroConsultado = ab.obtener(0).getNumeroBungalow();

**private** JComboBox<String> cboCategoria;

**private** JComboBox<String> cboEstado;

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == btnAceptar) {

actionPerformedBtnAceptar(arg0);

}

**if** (arg0.getSource() == btnConsultar) {

actionPerformedBtnConsultar(arg0);

}

**if** (arg0.getSource() == btnEliminar) {

actionPerformedBtnEliminar(arg0);

}

**if** (arg0.getSource() == btnModificar) {

actionPerformedBtnModificar(arg0);

}

**if** (arg0.getSource() == btnAdicionar) {

actionPerformedBtnAdicionar(arg0);

}

}

**protected** **void** actionPerformedBtnAdicionar(ActionEvent arg0) {

btnAdicionar.setEnabled(**false**);

btnModificar.setEnabled(**true**);

btnAceptar.setEnabled(**true**);

btnEliminar.setEnabled(**true**);

habilitarEntradas(**true**);

limpieza();

txtPrecioPorDia.requestFocus();

}

**protected** **void** actionPerformedBtnModificar(ActionEvent arg0) {

btnAdicionar.setEnabled(**true**);

btnModificar.setEnabled(**false**);

**if** (ab.tamaño() == 0) {

btnAceptar.setEnabled(**false**);

habilitarEntradas(**false**);

mensaje("No existen Bungalows");

}

**else** {

editarFila();

btnAceptar.setEnabled(**true**);

habilitarEntradas(**true**);

txtNumBun.requestFocus();

}

}

**protected** **void** actionPerformedBtnEliminar(ActionEvent arg0) {

btnAdicionar.setEnabled(**true**);

btnModificar.setEnabled(**true**);

btnAceptar.setEnabled(**false**);

**if** (ab.tamaño() == 0)

mensaje("No existen Bungalows");

**else** {

editarFila();

habilitarEntradas(**false**);

**int** ok = confirmar("¿ Desea eliminar el registro ?");

**if** (ok == 0) {

ab.eliminar(ab.buscar(leerNumBun()));

listar();

editarFila();

}

}

}

**protected** **void** actionPerformedBtnConsultar(ActionEvent arg0) {

Bungalow x = ab.buscar(leerNumBun());

numeroConsultado = x.getNumeroBungalow();

txtPrecioPorDia.setText("" + x.getPrecioPorDia());

cboCategoria.setSelectedIndex(x.getCategoria());

cboEstado.setSelectedIndex(x.getEstado());

txtNumBun.requestFocus();

}

**protected** **void** actionPerformedBtnAceptar(ActionEvent arg0) {

**int** numeroBungalow = leerNumBun();

**if** (numeroBungalow >= 0) {

**int** categoria = leerCategoria();

**if** (categoria >= 0)

**try** {

**double** precioPorDia = leerPrecioxDia();

**int** estado = leerEstado();

**if** (btnAdicionar.isEnabled() == **false**) {

Bungalow nuevo = **new** Bungalow(numeroBungalow, categoria, precioPorDia, estado);

ab.adicionar(nuevo);

btnAdicionar.setEnabled(**true**);

}

**if** (btnModificar.isEnabled() == **false**) {

Bungalow x = ab.buscar(numeroBungalow);

x.setNumeroBungalow(numeroBungalow);

x.setPrecioPorDia(precioPorDia);

x.setEstado(estado);

x.setCategoria(categoria);

btnModificar.setEnabled(**true**);

}

listar();

habilitarEntradas(**false**);

}

**catch** (Exception x) {

error("Ingrese Precio/Dia correcto", txtPrecioPorDia);

}

}

**else**

error("Ingrese Numero de Bungalow correcto", txtNumBun);

}

////////////

// Métodos tipo void (sin parámetros)

**void** ajustarAnchoColumnas() {

TableColumnModel tcm = tblbung.getColumnModel();

tcm.getColumn(0).setPreferredWidth(anchoColumna(20)); // Numero Bungalow

tcm.getColumn(1).setPreferredWidth(anchoColumna(20)); // Categoria

tcm.getColumn(2).setPreferredWidth(anchoColumna(20)); // Precio/Dia

tcm.getColumn(3).setPreferredWidth(anchoColumna(20)); // Estado

}

**void** listar() {

**int** posFila = 0;

**if** (modelo.getRowCount() > 0)

posFila = tblbung.getSelectedRow();

**if** (modelo.getRowCount() == ab.tamaño() - 1)

posFila = ab.tamaño() - 1;

**if** (posFila == ab.tamaño())

posFila--;

modelo.setRowCount(0);

Bungalow b;

**for** (**int** i = 0; i < ab.tamaño(); i++) {

b = ab.obtener(i);

Object[] fila = { b.getNumeroBungalow(), enTextoCategoria(b.getCategoria()), enTextoEstado(b.getEstado()), b.getPrecioPorDia() };

modelo.addRow(fila);

}

**if** (ab.tamaño() > 0)

tblbung.getSelectionModel().setSelectionInterval(posFila, posFila);

}

**void** editarFila() {

**if** (ab.tamaño() == 0)

limpieza();

**else** {

Bungalow b = ab.obtener(tblbung.getSelectedRow());

txtNumBun.setText("" + b.getNumeroBungalow());

cboCategoria.setSelectedIndex(b.getCategoria());

cboEstado.setSelectedIndex(b.getEstado());

txtPrecioPorDia.setText("" + b.getPrecioPorDia());

}

}

**void** limpieza() {

txtNumBun.setText("" + ab.numeroCorrelativo());

cboCategoria.setSelectedIndex(0);

;

cboEstado.setSelectedIndex(0);

txtPrecioPorDia.setText("");

}

// Métodos tipo void (con parámetros)

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 0);

}

**void** error(String s, JTextField txt) {

mensaje(s);

txt.setText("");

txt.requestFocus();

}

**void** habilitarEntradas(**boolean** sino) {

txtPrecioPorDia.setEditable(sino);

}

**void** habilitarBotones(**boolean** sino) {

btnAdicionar.setEnabled(sino);

btnModificar.setEnabled(sino);

}

// Métodos que retornan valor (sin parámetros)

**int** leerNumBun() {

**return** Integer.*parseInt*(txtNumBun.getText().trim());

}

**int** leerEstado() {

**return** cboEstado.getSelectedIndex();

}

**int** leerCategoria() {

**return** cboCategoria.getSelectedIndex();

}

**double** leerPrecioxDia() {

**return** Double.*parseDouble*(txtPrecioPorDia.getText().trim());

}

// Métodos que retornan valor (con parámetros)

String enTextoCategoria(**int** i) {

**return** cboCategoria.getItemAt(i);

}

String enTextoEstado(**int** i) {

**return** cboEstado.getItemAt(i);

}

**int** anchoColumna(**int** porcentaje) {

**return** porcentaje \* scrollPane.getWidth() / 100;

}

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

}

REGISTRO

INGRESOS

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.table.TableColumnModel;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloSocio;

**import** clases.Ingresos;

**import** clases.Socio;

**import** libreria.Lib;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** java.awt.event.ActionListener;

**import** java.util.Calendar;

**import** java.util.GregorianCalendar;

**import** java.awt.event.ActionEvent;

**public** **class** JdIngreso **extends** JDialog **implements** ActionListener {

**private** JLabel lblIngreso;

**private** JTextField txtCodIngreso;

**private** JLabel lblSocio;

**private** JTextField txtInvitados;

**private** JLabel lblInvitados;

**private** JButton btnIngresar;

**private** JScrollPane scrollPane;

**private** JTable tblIngreso;

**private** DefaultTableModel modelo;

**private** JComboBox<String> cboCodigosSocios;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdIngreso dialog = **new** JdIngreso();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdIngreso() {

setTitle("Registro||Ingreso");

setBounds(100, 100, 660, 458);

getContentPane().setLayout(**null**);

lblIngreso = **new** JLabel("Ingreso:");

lblIngreso.setBounds(12, 11, 56, 16);

getContentPane().add(lblIngreso);

txtCodIngreso = **new** JTextField();

txtCodIngreso.setEditable(**false**);

txtCodIngreso.setText("40001");

txtCodIngreso.setBounds(129, 8, 79, 22);

getContentPane().add(txtCodIngreso);

txtCodIngreso.setColumns(10);

lblSocio = **new** JLabel("Socio:");

lblSocio.setBounds(12, 46, 56, 16);

getContentPane().add(lblSocio);

txtInvitados = **new** JTextField();

txtInvitados.setBounds(129, 82, 79, 22);

getContentPane().add(txtInvitados);

txtInvitados.setColumns(10);

lblInvitados = **new** JLabel("Numero Invitados:");

lblInvitados.setBounds(10, 85, 105, 16);

getContentPane().add(lblInvitados);

btnIngresar = **new** JButton("Ingresar");

btnIngresar.addActionListener(**this**);

btnIngresar.setBounds(508, 43, 122, 22);

getContentPane().add(btnIngresar);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 128, 618, 270);

getContentPane().add(scrollPane);

tblIngreso = **new** JTable();

tblIngreso.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tblIngreso);

modelo=**new** DefaultTableModel();

modelo.addColumn("Codigo Ingreso");

modelo.addColumn("Codigo Socio");

modelo.addColumn("Fecha Ingreso");

modelo.addColumn("Hora Ingreso");

modelo.addColumn("Invitados");

modelo.addColumn("Costo");

modelo.addColumn("Estado");

tblIngreso.setModel(modelo);

cboCodigosSocios = **new** JComboBox<String>();

cboCodigosSocios.setBounds(129, 41, 79, 30);

getContentPane().add(cboCodigosSocios);

txtCodIngreso.setText(""+ai.codigoCorrelativo());

colocarCodigosPacientes();

ajustarAnchoColumnas();

listar();

}

ArregloIngresos ai=**new** ArregloIngresos();

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == btnIngresar) {

actionPerformedBtnIngresar(arg0);

}

}

**protected** **void** actionPerformedBtnIngresar(ActionEvent arg0) {

**int** codigoIngreso=leerCodigoIngreso();

**int** codigoSocio=leerCodigoSocio();

String fechaIngreso=fechaActual();

String horaIngreso=horaActual();

**int** numeroInvitados=leerNumeroInvitados();

**double** costoIngreso=CostoInvitado();

Ingresos nuevo =**new** Ingresos(codigoIngreso, codigoSocio, fechaIngreso, horaIngreso, numeroInvitados, costoIngreso, 0);

ai.adicionar(nuevo);

listar();

txtCodIngreso.setText(""+ai.codigoCorrelativo());

txtInvitados.setText("");

}

//metodos void sin parametros

**void** colocarCodigosPacientes() {

ArregloSocio as = **new** ArregloSocio();

Socio s;

**for** (**int** i=0; i<as.tamaño(); i++) {

s = as.obtener(i);

**if** (ai.procedeCodigoSocio(s.getCodigoSocio()))

cboCodigosSocios.addItem("" + s.getCodigoSocio());

}

}

**void** listar(){

Ingresos x;

modelo.setRowCount(0);

**for**(**int** i=0;i<ai.tamaño();i++){

x=ai.obtener(i);

Object[]fila={ x.getcodigoIngreso(),

x.getcodigoSocio(),

x.getfechaIngreso(),

x.gethoraIngreso(),

x.getnumeroInvitados(),

x.getcostoIngreso(),

Lib.*estadosIngreso*[x.getEstado()]};

modelo.addRow(fila);

}

}

**void** ajustarAnchoColumnas(){

TableColumnModel tcm=tblIngreso.getColumnModel();

tcm.getColumn(0).setPreferredWidth(anchoColumna( 8));

tcm.getColumn(1).setPreferredWidth(anchoColumna(18));

tcm.getColumn(2).setPreferredWidth(anchoColumna(10));

tcm.getColumn(3).setPreferredWidth(anchoColumna(10));

tcm.getColumn(4).setPreferredWidth(anchoColumna(15));

tcm.getColumn(5).setPreferredWidth(anchoColumna(15));

tcm.getColumn(6).setPreferredWidth(anchoColumna(15));

}

//METODOS QUE RETORNAN VALOR SIN PARAMETRO

**int** leerCodigoIngreso(){

**return** Integer.*parseInt*(txtCodIngreso.getText().trim());

}

**int** leerNumeroInvitados(){

**return** Integer.*parseInt*(txtInvitados.getText().trim());

}

**double** CostoInvitado(){

**return** leerNumeroInvitados()\*25;

}

**int** leerCodigoSocio(){

**return** Integer.*parseInt*(cboCodigosSocios.getSelectedItem().toString());

}

String fechaActual() {

**int** dd, mm, aa;

Calendar c = **new** GregorianCalendar();

dd = c.get(Calendar.***DAY\_OF\_MONTH***);

mm = c.get(Calendar.***MONTH***) + 1;

aa = c.get(Calendar.***YEAR***);

**return** ajustar(dd) + "/" + ajustar(mm) + "/" + aa;

}

String horaActual() {

**int** hh, mm, ss;

Calendar c = **new** GregorianCalendar();

hh = c.get(Calendar.***HOUR\_OF\_DAY***);

mm = c.get(Calendar.***MINUTE***);

ss = c.get(Calendar.***SECOND***);

**return** ajustar(hh) + ":" + ajustar(mm) + ":" + ajustar(ss);

}

//METODOS QUE RETORNAN VALOR

**int** anchoColumna(**int** porcentaje) {

**return** porcentaje \* scrollPane.getWidth() / 100;

}

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

String ajustar(**int** numero) {

**return** String.*format*("%02d", numero);

}

}

CONSUMO

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.table.TableColumnModel;

**import** arreglos.ArregloConsumo;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloProducto;

**import** arreglos.ArregloSocio;

**import** clases.Consumo;

**import** clases.Ingresos;

**import** clases.Producto;

**import** clases.Socio;

**import** libreria.Fecha;

**import** libreria.Lib;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.MouseListener;

**public** **class** JdConsumo **extends** JDialog **implements** ActionListener, MouseListener {

**private** JLabel lblCodigo;

**private** JTextField txtIngreso;

**private** JLabel lblSocio;

**private** JTextField txtSocio;

**private** JLabel lblProducto;

**private** JComboBox<String> cboCodigoProducto;

**private** JButton btnQuitar;

**private** JComboBox<String> cboProducto;

**private** JButton btnAgregar;

**private** JScrollPane scrollPane;

**private** JTable tblIngreso;

**private** JComboBox<String> cboCodigoSocio;

**private** DefaultTableModel modelo;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdConsumo dialog = **new** JdConsumo();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdConsumo() {

setTitle("Registro||Consumo");

setBounds(100, 100, 792, 499);

getContentPane().setLayout(**null**);

lblCodigo = **new** JLabel("Codigo");

lblCodigo.setBounds(12, 13, 56, 16);

getContentPane().add(lblCodigo);

txtIngreso = **new** JTextField();

txtIngreso.setEditable(**false**);

txtIngreso.setBounds(72, 10, 116, 22);

getContentPane().add(txtIngreso);

txtIngreso.setColumns(10);

lblSocio = **new** JLabel("Socio");

lblSocio.addMouseListener(**this**);

lblSocio.setBounds(12, 42, 56, 16);

getContentPane().add(lblSocio);

txtSocio = **new** JTextField();

txtSocio.setEditable(**false**);

txtSocio.setBounds(72, 39, 116, 22);

getContentPane().add(txtSocio);

txtSocio.setColumns(10);

lblProducto = **new** JLabel("Producto");

lblProducto.setBounds(12, 79, 56, 16);

getContentPane().add(lblProducto);

cboCodigoProducto = **new** JComboBox<String>();

cboCodigoProducto.setBounds(72, 76, 104, 22);

getContentPane().add(cboCodigoProducto);

btnQuitar = **new** JButton("Quitar");

btnQuitar.addActionListener(**this**);

btnQuitar.setBounds(420, 75, 97, 25);

getContentPane().add(btnQuitar);

cboProducto = **new** JComboBox<String>();

cboProducto.setBounds(311, 76, 97, 22);

getContentPane().add(cboProducto);

btnAgregar = **new** JButton("Agregar");

btnAgregar.addActionListener(**this**);

btnAgregar.setBounds(188, 75, 97, 25);

getContentPane().add(btnAgregar);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 144, 750, 295);

getContentPane().add(scrollPane);

tblIngreso = **new** JTable();

tblIngreso.addMouseListener(**this**);

tblIngreso.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tblIngreso);

cboCodigoSocio = **new** JComboBox<String>();

cboCodigoSocio.addActionListener(**this**);

cboCodigoSocio.setBounds(200, 39, 85, 22);

getContentPane().add(cboCodigoSocio);

colocarCodigosSocio();

modelo = **new** DefaultTableModel();

modelo.addColumn("Código");

modelo.addColumn("Socio");

modelo.addColumn("Invitados");

modelo.addColumn("Fecha Ingreso");

modelo.addColumn("Hora Ingreso");

modelo.addColumn("Total pagar");

modelo.addColumn(" Estado");

tblIngreso.setModel(modelo);

txtIngreso.setEditable(**false**);

txtSocio.setEditable(**false**);

ajustarAnchoColumnas();

listar();

editarFila();

}

//Declaración global

ArregloIngresos ai= **new** ArregloIngresos();

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == btnAgregar) {

actionPerformedBtnAgregar(e);

}

**if** (e.getSource() == btnQuitar) {

actionPerformedBtnQuitar(e);

}

**if** (e.getSource() == cboCodigoSocio) {

actionPerformedCboCodigoSocio(e);

}

}

**protected** **void** actionPerformedBtnQuitar(ActionEvent e) {

Ingresos i = ai.buscar(leerCodigoIngreso());

**if** (i.getEstado() == 0)

**try** {

**int** codProducto = leerCodigoProductoRemover();

**int** ok = confirmar("¿ Quitar producto del consumo ?");

**if** (ok == 0) {

ArregloConsumo ac = **new** ArregloConsumo(leerCodigoConsumo());

Consumo c = ac.buscar(codProducto);

ac.eliminar(c);

ac.grabarConsumo();

ArregloProducto ap = **new** ArregloProducto();

Producto p = ap.buscar(codProducto);

p.setStock(p.getStock() + c.getCantidad());

ap.actualizarDatos();

i.setcostoIngreso(redondear(i.getcostoIngreso() - c.getCantidad()\*c.getPrecioUnitario()));

ai.actualizarDatos();

cboCodigoProducto.addItem(cboProducto.getSelectedItem().toString());

cboProducto.removeItem(cboProducto.getSelectedItem());

listar();

}

}

**catch** (Exception e1) {

mensaje("El Socio no cuenta con productos .");

}

**else**

mensaje("No se puede quitar productos porque el consumo está pagado");

}

**protected** **void** actionPerformedBtnAgregar(ActionEvent e) {

Ingresos i = ai.buscar(leerCodigoIngreso());

**if** (i.getEstado() == 0)

**try** {

**int** codigoProducto = leerCodigoProducto();

**int** ok = confirmar(obtenerDatosProducto(codigoProducto), "¿ Añadir ?");

**if** (ok == 0)

**try** {

**int** cantidad = Integer.*parseInt*(confirmarIngreso("Cantidad"));

ArregloProducto ap = **new** ArregloProducto();

Producto p = ap.buscar(codigoProducto);

**if** (cantidad <= p.getStock()) {

ArregloConsumo ac = **new** ArregloConsumo(leerCodigoConsumo());

ac.adicionar(**new** Consumo(codigoProducto, cantidad, p.getPrecioUnitario()));;

ac.grabarConsumo();

p.setStock(p.getStock() - cantidad);

ac.grabarConsumo();

i.setcostoIngreso(i.getcostoIngreso() + cantidad\*p.getPrecioUnitario());;

ai.grabarIngreso();;

cboProducto.addItem(cboCodigoProducto.getSelectedItem().toString());

cboCodigoProducto.removeItem(cboCodigoProducto.getSelectedItem());

listar();

}

**else**

mensaje("Sólo hay " + p.getStock() + " unidades en stock");

}

**catch** (Exception e1) {

mensaje("Ingrese CANTIDAD correcta");

}

}

**catch** (Exception e1) {

mensaje("El consumo del Socio tiene todos los productos");

}

**else**

mensaje("No se puede agregar Productos el Ingreso ya esta pagado");

}

// Métodos tipo void (sin parámetros)

**void** colocarCodigosSocio() {

ArregloSocio as = **new** ArregloSocio();

Socio s;

**for** (**int** i=0; i<as.tamaño(); i++) {

s = as.obtener(i);

cboCodigoSocio.addItem("" + s.getCodigoSocio());

}

}

**void** ajustarAnchoColumnas() {

TableColumnModel tcm = tblIngreso.getColumnModel();

tcm.getColumn(0).setPreferredWidth(anchoColumna(13)); // codigoAtencion

tcm.getColumn(1).setPreferredWidth(anchoColumna(13)); // codigoPaciente

tcm.getColumn(2).setPreferredWidth(anchoColumna(13)); // invitados

tcm.getColumn(3).setPreferredWidth(anchoColumna(22)); // fechaAtencion

tcm.getColumn(4).setPreferredWidth(anchoColumna(13)); // horaAtencion

tcm.getColumn(5).setPreferredWidth(anchoColumna(13)); // totalPagar

tcm.getColumn(6).setPreferredWidth(anchoColumna(13)); // estado

}

**void** distribuirCodigosProductos() {

ArregloConsumo ac = **new** ArregloConsumo(leerCodigoConsumo());

Consumo c;

cboProducto.removeAllItems();

**for** (**int** i=0; i<ac.tamaño(); i++) {

c = ac.obtener(i);

cboProducto.addItem("" + c.getCodigoProducto());

}

ArregloProducto ap = **new** ArregloProducto();

Producto p;

cboCodigoProducto.removeAllItems();

**for** (**int** i=0; i<ap.tamaño(); i++) {

p = ap.obtener(i);

**if** (ac.buscar(p.getCodigoProducto()) == **null**)

cboCodigoProducto.addItem("" + p.getCodigoProducto());

}

}

**void** listar() {

**int** posFila = 0;

**if** (modelo.getRowCount() > 0)

posFila = tblIngreso.getSelectedRow();

**if** (modelo.getRowCount() == ai.tamaño() - 1)

posFila = ai.tamaño() - 1;

**if** (posFila == ai.tamaño())

posFila --;

modelo.setRowCount(0);

Ingresos i;

**for** (**int** j=0; j<ai.tamaño(); j++) {

i = ai.obtener(j);

Object[] fila = { i.getcodigoIngreso(),

i.getcodigoSocio(),

i.getnumeroInvitados(),

Fecha.*enTextoFecha*(i.getfechaIngreso()),

i.gethoraIngreso(),

i.getcostoIngreso(),

Lib.*estadosIngreso*[i.getEstado()] };

modelo.addRow(fila);

}

**if** (ai.tamaño() > 0)

tblIngreso.getSelectionModel().setSelectionInterval(posFila, posFila);

}

**void** editarFila() {

**if** (ai.tamaño() == 0)

txtIngreso.setText("" + ai.codigoCorrelativo());

**else** {

Ingresos i = ai.obtener(tblIngreso.getSelectedRow());

txtIngreso.setText("" + i.getcodigoIngreso());

txtSocio.setText("" + i.getcodigoSocio());

distribuirCodigosProductos();

}

}

// Métodos que retornan valor (con parámetros)

**int** anchoColumna(**int** porcentaje) {

**return** porcentaje \* scrollPane.getWidth() / 100;

}

**int** confirmar(String s1, String s2) {

**return** JOptionPane.*showConfirmDialog*(**this**, s1, s2, 0, 1, **null**);

}

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

String confirmarIngreso(String s) {

**return** JOptionPane.*showInputDialog*(**this**, "", s, 3);

}

// Métodos que retornan valor (sin parámetros)

String obtenerDatosSocio() {

ArregloSocio as = **new** ArregloSocio();

Socio s = as.buscar(leerCodigoSocio());

**return** "Nombres : " + s.getNombres() + "\n" +

"Apellidos : " + s.getApellidos();

}

**double** redondear(**double** real) {

**return** Math.*round*(real \* 100) / 100.0;

}

**int** leerCodigoSocio(){

**return** Integer.*parseInt*( txtSocio.getText());

}

**int** leerCodigoIngreso(){

**return** Integer.*parseInt*(txtIngreso.getText());

}

String leerCodigoConsumo(){

**return** "" + leerCodigoIngreso();

}

**int** leerCodigoProducto() {

**return** Integer.*parseInt*(cboCodigoProducto.getSelectedItem().toString());

}

**int** leerCodigoProductoRemover(){

**return** Integer.*parseInt*(cboProducto.getSelectedItem().toString());

}

String obtenerDatosProducto(**int** codigoProducto) {

ArregloProducto ap = **new** ArregloProducto();

Producto p = ap.buscar(codigoProducto);

**return** "Detalle : " + p.getDetalle() + "\n" +

"Precio unitario : S/ " + p.getPrecioUnitario();

}

//Métodos tipo void(con parámetros)

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 1);

}

**protected** **void** actionPerformedCboCodigoSocio(ActionEvent e) {

txtSocio.setText(""+cboCodigoSocio.getSelectedItem());

}

**public** **void** mouseClicked(MouseEvent e) {

**if** (e.getSource() == lblSocio) {

mouseClickedLblSocio(e);

}

**if** (e.getSource() == tblIngreso) {

mouseClickedTblIngreso(e);

}

}

**public** **void** mouseEntered(MouseEvent e) {

}

**public** **void** mouseExited(MouseEvent e) {

}

**public** **void** mousePressed(MouseEvent e) {

}

**public** **void** mouseReleased(MouseEvent e) {

}

**protected** **void** mouseClickedTblIngreso(MouseEvent e) {

editarFila();

}

**protected** **void** mouseClickedLblSocio(MouseEvent e) {

mensaje(obtenerDatosSocio());

}

}

HOSPEDAJE

**package** GUI;

**import** libreria.\*;

**import** clases.\*;

**import** arreglos.\*;

**import** java.awt.BorderLayout;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.table.DefaultTableModel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

**import** javax.swing.JComboBox;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** java.awt.Font;

**import** java.awt.event.MouseListener;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.KeyListener;

**import** java.awt.event.KeyEvent;

**public** **class** JdHospedaje **extends** JDialog **implements** ActionListener, MouseListener, KeyListener {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **final** JPanel contentPanel = **new** JPanel();

**private** JScrollPane scrollPane;

**private** JTable tbhospedaje;

**private** JLabel lblHospedaje;

**private** JLabel lblIngreso;

**private** JLabel lblBungalow;

**private** JTextField txtHospedaje;

**private** JComboBox<String> cboCodIngreso;

**private** JComboBox<String> cboBungalow;

**private** JButton btnProceder;

**private** JButton btnHospedar;

**private** DefaultTableModel modelo;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

**try** {

JdHospedaje dialog = **new** JdHospedaje();

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

dialog.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* Create the dialog.

\*/

**public** JdHospedaje() {

setTitle("Registro||Hospedaje");

setBounds(100, 100, 772, 570);

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

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

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

contentPanel.setLayout(**null**);

scrollPane = **new** JScrollPane();

scrollPane.setToolTipText("");

scrollPane.setBounds(10, 125, 736, 385);

contentPanel.add(scrollPane);

tbhospedaje = **new** JTable();

tbhospedaje.addKeyListener(**this**);

tbhospedaje.addMouseListener(**this**);

tbhospedaje.setFillsViewportHeight(**true**);

scrollPane.setViewportView(tbhospedaje);

modelo=**new** DefaultTableModel();

modelo = **new** DefaultTableModel();

modelo.addColumn("Código hospedaje");

modelo.addColumn("Código Ingreso");

modelo.addColumn("Nro bungalow");

modelo.addColumn("Fecha de salida");

modelo.addColumn("Hora de salida");

modelo.addColumn("Costo de hospedaje");

modelo.addColumn("Estado");

tbhospedaje.setModel(modelo);

lblHospedaje = **new** JLabel("Hospedaje");

lblHospedaje.setFont(**new** Font("Dialog", Font.***PLAIN***, 12));

lblHospedaje.setBounds(10, 19, 80, 20);

contentPanel.add(lblHospedaje);

lblIngreso = **new** JLabel("Ingreso");

lblIngreso.setFont(**new** Font("Dialog", Font.***BOLD***, 12));

lblIngreso.setBounds(214, 22, 46, 14);

contentPanel.add(lblIngreso);

lblBungalow = **new** JLabel("Bungalow");

lblBungalow.setFont(**new** Font("Dialog", Font.***BOLD***, 12));

lblBungalow.setBounds(10, 50, 61, 14);

contentPanel.add(lblBungalow);

txtHospedaje = **new** JTextField();

txtHospedaje.setEditable(**false**);

txtHospedaje.setBounds(100, 20, 86, 20);

contentPanel.add(txtHospedaje);

txtHospedaje.setColumns(10);

cboCodIngreso = **new** JComboBox<String>();

cboCodIngreso.setBounds(287, 21, 86, 22);

contentPanel.add(cboCodIngreso);

visualizarIngreso();

cboBungalow = **new** JComboBox<String>();

cboBungalow.setBounds(100, 48, 87, 20);

contentPanel.add(cboBungalow);

visualizarNroBungalow();

btnProceder = **new** JButton("Proceder");

btnProceder.setFont(**new** Font("Dialog", Font.***PLAIN***, 12));

btnProceder.addActionListener(**this**);

btnProceder.setBounds(383, 18, 86, 28);

contentPanel.add(btnProceder);

txtHospedaje.setText("" + ah.codigoCorrelativo());

btnHospedar = **new** JButton("Hospedar");

btnHospedar.addActionListener(**this**);

btnHospedar.setFont(**new** Font("Dialog", Font.***PLAIN***, 12));

btnHospedar.setBounds(649, 11, 97, 90);

contentPanel.add(btnHospedar);

listar();

editarFila();

}

ArregloIngresos ai = **new** ArregloIngresos();

ArregloBungalow ab = **new** ArregloBungalow();

ArregloHospedaje ah = **new** ArregloHospedaje();

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == btnHospedar) {

actionPerformedBtnHospedar(arg0);

}

**if** (arg0.getSource() == btnProceder) {

actionPerformedBtnProceder(arg0);

}

}

**protected** **void** actionPerformedBtnProceder(ActionEvent arg0) {

**int** codigoHospedaje = leerCodigoHospedaje();

**int** codigoIngreso = leerCodigoIngreso();

**int** numeroBungalow = leerNumeroBungalow();

**int** ok = confirmar("Desea Hospedar");

**if**(ok == 0){

Bungalow b = ab.buscar(numeroBungalow);

Hospedajes nuevo = **new** Hospedajes(codigoHospedaje, codigoIngreso, numeroBungalow,"", "", 0, 0);

ah.adicionar(nuevo);

ah.actualizarDatos();

b.setEstado(1);

ab.actualizarDatos();

cboCodIngreso.removeItem(cboCodIngreso.getSelectedItem());

cboBungalow.removeItem(cboBungalow.getSelectedItem());

}

listar();

editarFila();

}

**protected** **void** actionPerformedBtnHospedar(ActionEvent arg0) {

**if**(cboCodIngreso.getSelectedIndex()<0)

mensaje("Todos los socios están hospedados");

**else**

**if**(cboBungalow.getSelectedIndex()<0)

mensaje("Todos los Bungalows están ocupados");

**else**{

txtHospedaje.setText("" + ah.codigoCorrelativo());

}

}

**void** listar(){

**int** posfila = 0;

**if** (modelo.getRowCount() > 0)

posfila = tbhospedaje.getSelectedRow();

**if** (modelo.getRowCount() == ah.tamaño() - 1)

posfila = ah.tamaño() - 1;

**if** (posfila == ah.tamaño())

posfila --;

modelo.setRowCount(0);

Hospedajes h;

**for**(**int** i=0;i< ah.tamaño();i++){

h=ah.obtener(i);

Object[] lista={ h.getCodigoHospedaje(),

h.getCodigoIngreso(),

h.getnumeroBungalow(),

h.getfechaSalida(),

h.gethoraSalida(),

h.getcostoHospedaje(),

Lib.*estadosHospedaje*[h.getEstado()]};

modelo.addRow(lista);

}

**if** (ai.tamaño() > 0)

tbhospedaje.getSelectionModel().setSelectionInterval(posfila, posfila);

}

**void** editarFila() {

**if** (ah.tamaño() == 0)

txtHospedaje.setText("" + ah.codigoCorrelativo());

**else** {

Hospedajes h = ah.obtener(tbhospedaje.getSelectedRow());

txtHospedaje.setText("" + h.getCodigoHospedaje());

}

}

**void** visualizarNroBungalow(){

ArregloBungalow ab = **new** ArregloBungalow();

Bungalow b;

**for** (**int** i=0; i<ab.tamaño(); i++) {

b = ab.obtener(i);

**if** (b.getEstado() == 0)

cboBungalow.addItem("" + b.getNumeroBungalow());

}

}

**void** visualizarIngreso(){

Ingresos i;

**for**(**int** j=0; j< ai.tamaño(); j++){

i = ai.obtener(j);

**if**(ah.procedeCodigoIngreso(i.getcodigoIngreso()))

cboCodIngreso.addItem("" + i.getcodigoIngreso());

}

}

**int** leerCodigoHospedaje(){

**return** Integer.*parseInt*(txtHospedaje.getText().trim());

}

**int** leerCodigoIngreso(){

**return** Integer.*parseInt*(cboCodIngreso.getSelectedItem().toString());

}

**int** leerNumeroBungalow(){

**return** Integer.*parseInt*(cboBungalow.getSelectedItem().toString());

}

// Métodos que retornan valor (con parámetros)

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 1);

}

**int** anchoColumna(**int** porcentaje) {

**return** porcentaje \* scrollPane.getWidth() / 100;

}

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

**int** confirmar(String s1, String s2) {

**return** JOptionPane.*showConfirmDialog*(**this**, s1, s2, 0, 1, **null**);

}

**public** **void** mouseClicked(MouseEvent arg0) {

**if** (arg0.getSource() == tbhospedaje) {

mouseClickedTbhospedaje(arg0);

}

}

**public** **void** mouseEntered(MouseEvent arg0) {

}

**public** **void** mouseExited(MouseEvent arg0) {

}

**public** **void** mousePressed(MouseEvent arg0) {

}

**public** **void** mouseReleased(MouseEvent arg0) {

}

**protected** **void** mouseClickedTbhospedaje(MouseEvent arg0) {

}

**public** **void** keyPressed(KeyEvent arg0) {

}

**public** **void** keyReleased(KeyEvent arg0) {

**if** (arg0.getSource() == tbhospedaje) {

keyReleasedTbhospedaje(arg0);

}

}

**public** **void** keyTyped(KeyEvent arg0) {

}

**protected** **void** keyReleasedTbhospedaje(KeyEvent arg0) {

arg0.consume();

editarFila();

}

}

PAGO

INGRESOS Y CONSUMOS

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextArea;

**import** arreglos.ArregloConsumo;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloProducto;

**import** arreglos.ArregloSocio;

**import** clases.Consumo;

**import** clases.Ingresos;

**import** clases.Producto;

**import** clases.Socio;

**import** java.awt.Font;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** java.awt.event.ActionListener;

**import** java.util.Locale;

**import** java.awt.event.ActionEvent;

**public** **class** JdIngresosyConsumos **extends** JDialog **implements** ActionListener {

**private** JLabel lblCodigoIngreso;

**private** JTextArea txtS;

**private** JButton btnPagar;

**private** JComboBox<Object> cboCodigoIngreso;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdIngresosyConsumos dialog = **new** JdIngresosyConsumos();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdIngresosyConsumos() {

setBounds(100, 100, 677, 427);

setTitle("Pago||Ingresos y Consumos");

getContentPane().setLayout(**null**);

lblCodigoIngreso = **new** JLabel("Codigo");

lblCodigoIngreso.setBounds(12, 13, 110, 23);

getContentPane().add(lblCodigoIngreso);

txtS = **new** JTextArea();

txtS.setFont(**new** Font("Monospaced", Font.***PLAIN***, 15));

txtS.setBounds(12, 49, 635, 318);

getContentPane().add(txtS);

btnPagar = **new** JButton("Pagar");

btnPagar.addActionListener(**this**);

btnPagar.setBounds(530, 12, 97, 25);

getContentPane().add(btnPagar);

cboCodigoIngreso = **new** JComboBox<Object>();

cboCodigoIngreso.addActionListener(**this**);

cboCodigoIngreso.setBounds(88, 13, 167, 22);

getContentPane().add(cboCodigoIngreso);

colocarCodigosConsultas();

}

ArregloIngresos ai=**new** ArregloIngresos();

ArregloSocio as=**new** ArregloSocio();

**public** **void** actionPerformed(ActionEvent arg0) {

**if** (arg0.getSource() == cboCodigoIngreso) {

actionPerformedCboCodigoIngreso(arg0);

}

**if** (arg0.getSource() == btnPagar) {

actionPerformedBtnPagar(arg0);

}

}

**protected** **void** actionPerformedCboCodigoIngreso(ActionEvent arg0) {

txtS.setText("");

**if**(cboCodigoIngreso.getSelectedIndex()>=0)

listar();

}

**protected** **void** actionPerformedBtnPagar(ActionEvent arg0) {

**if**(cboCodigoIngreso.getSelectedIndex()>=0){

**int** ok=confirmar("¿Desea pagar Atencion?");

**if**(ok==0){

Ingresos i =ai.buscar(leerCodigoIngreso());

i.setEstado(1);

ai.actualizarDatos();

cboCodigoIngreso.removeItem(cboCodigoIngreso.getSelectedItem());

}

}

**else** mensaje("No existen pagos pendientes");

}

// Métodos tipo void (sin parámetros)

**void** colocarCodigosConsultas() {

Ingresos a;

**for** (**int** i=0; i<ai.tamaño(); i++) {

a = ai.obtener(i);

**if** (a.getEstado() == 0)

cboCodigoIngreso.addItem("" + a.getcodigoIngreso());

}

}

**void** imprimir(){

imprimir("");

}

**void** listar(){

Ingresos in=ai.buscar(leerCodigoIngreso());

Socio s=as.buscar(in.getcodigoSocio());

imprimir("Socio : "+s.getCodigoSocio());

imprimir("Nombres : "+s.getNombres());

imprimir("Apellidos : "+s.getApellidos());

imprimir();

ArregloConsumo ac=**new** ArregloConsumo(""+leerCodigoIngreso());

Consumo c;

imprimir("1)Consumo");

imprimir(" Cantidad Precio Importe");

**double** pago;

**for**(**int** i=0;i<ac.tamaño();i++){

c=ac.obtener(i);

pago=c.getCantidad()\*c.getPrecioUnitario();

ArregloProducto ap=**new** ArregloProducto();

Producto p=ap.buscar(c.getCodigoProducto());

imprimir(" "+formato(p.getDetalle())+formato(c.getCantidad())+formato(c.getPrecioUnitario())+formato(pago));

}

imprimir();

imprimir("TOTAL A PAGAR S/ "+formato(in.getcostoIngreso()));

}

//METODOS TIPO VOID (CON PARAMETROS)

**void** imprimir(String s){

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

}

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 1);

}

//Metodos que retornan valor (sin parametros)

**int** leerCodigoIngreso(){

**return** Integer.*parseInt*(cboCodigoIngreso.getSelectedItem().toString());

}

//metodos que retornan valor(Con parametros)

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

String formato(String cadena) {

**return** String.*format*("%-15s", cadena);

}

String formato(**int** entero) {

**return** String.*format*("%-10d", entero);

}

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

HOSPEDAJE

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JScrollPane;

**import** arreglos.\*;

**import** clases.\*;

**import** libreria.Fecha;

**import** libreria.Lib;

**import** javax.swing.JTextArea;

**import** java.awt.event.ActionListener;

**import** java.util.Locale;

**import** java.awt.event.ActionEvent;

**public** **class** JdHospedajes **extends** JDialog **implements** ActionListener {

**private** JLabel lblHospedajes;

**private** JComboBox<String> cboCodigoHospedaje;

**private** JButton btnPagar;

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdHospedajes dialog = **new** JdHospedajes();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdHospedajes() {

setBounds(100, 100, 734, 439);

setTitle("Pago||Hospedajes");

getContentPane().setLayout(**null**);

lblHospedajes = **new** JLabel("Hospedajes");

lblHospedajes.setBounds(12, 13, 81, 16);

getContentPane().add(lblHospedajes);

cboCodigoHospedaje = **new** JComboBox<String>();

cboCodigoHospedaje.addActionListener(**this**);

cboCodigoHospedaje.setBounds(105, 13, 125, 22);

getContentPane().add(cboCodigoHospedaje);

btnPagar = **new** JButton("Pagar");

btnPagar.addActionListener(**this**);

btnPagar.setBounds(590, 9, 97, 25);

getContentPane().add(btnPagar);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(22, 53, 671, 311);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

scrollPane.setViewportView(txtS);

colocarCodigosHospedaje();

}

// Declaración global

ArregloSocio as=**new** ArregloSocio();

ArregloIngresos ai=**new** ArregloIngresos();

ArregloHospedaje ah = **new** ArregloHospedaje();

String fechaSalida, horaSalida;

**double** costoHospedaje;

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == btnPagar) {

actionPerformedBtnPagar(e);

}

**if** (e.getSource() == cboCodigoHospedaje) {

actionPerformedCboCodigoHospedajes(e);

}

}

**protected** **void** actionPerformedCboCodigoHospedajes(ActionEvent e) {

//txtS.setText("");

**if** (cboCodigoHospedaje.getSelectedIndex() >= 0)

listar();

}

**protected** **void** actionPerformedBtnPagar(ActionEvent e) {

**if** (cboCodigoHospedaje.getSelectedIndex() >= 0) {

**int** ok = confirmar("¿ Desea pagar Hospedaje ?");

**if** (ok == 0) {

Hospedajes h = ah.buscar(leerCodigoHospedaje());

h.setEstado(1);

h.setfechaSalida(fechaSalida);

h.sethoraSalida(horaSalida);

h.setcostoHospedaje(costoHospedaje);

ah.actualizarDatos();

ArregloBungalow ab = **new** ArregloBungalow();

Bungalow b = ab.buscar(h.getnumeroBungalow());

b.setEstado(0);

ab.actualizarDatos();;

cboCodigoHospedaje.removeItem(cboCodigoHospedaje.getSelectedItem());

}

}

**else**

mensaje("No existen Hospedajes pendientes de pago");

}

// Métodos tipo void (sin parámetros)

**void** colocarCodigosHospedaje() {

Hospedajes h;

**for** (**int** j=0; j<ah.tamaño(); j++) {

h = ah.obtener(j);

**if** (h.getEstado() == 0)

cboCodigoHospedaje.addItem("" + h.getCodigoHospedaje());

}

}

**void** imprimir() {

imprimir("");

}

**void** listar() {

Hospedajes h=ah.buscar(leerCodigoHospedaje());

Ingresos i=ai.buscar(h.getCodigoIngreso());

Socio s=as.buscar(i.getcodigoSocio());

imprimir("Socio : " + s.getCodigoSocio());

imprimir("Nombres : " + s.getNombres());

imprimir("Apellidos : " + s.getApellidos());

imprimir();

imprimir("Bungalow : " + h.getnumeroBungalow());

ArregloBungalow ab = **new** ArregloBungalow();

Bungalow b = ab.buscar(h.getnumeroBungalow());

imprimir("Categoría : " + Lib.*CategoriaBungalow*[b.getCategoria()]);

imprimir();

imprimir("Ingreso : " + i.getfechaIngreso()

+ " - " + i.gethoraIngreso());

fechaSalida = Fecha.*fechaActual*();

horaSalida = Fecha.*horaActual*();

imprimir("Salida : " + fechaSalida

+ " - " + horaSalida);

**int** dias = Fecha.*diasTranscurridos*(i.getfechaIngreso(), fechaSalida);

costoHospedaje =h.getcostoHospedaje() + dias \* b.getPrecioPorDia();

imprimir();

imprimir("1) COSTO DE HOSPEDAJE S/ " + formato(h.getCodigoHospedaje()));

imprimir();

imprimir("2) PRECIO POR DÍA S/ " + formato(b.getPrecioPorDia()));

imprimir();

imprimir("3) DÍAS TRANSCURRIDOS " + dias);

imprimir();

imprimir("4) TOTAL A PAGAR S/ " + formato(costoHospedaje));

}

// Métodos tipo void (con parámetros)

**void** imprimir(String s) {

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

}

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 1);

}

// Métodos que retornan valor (sin parámetros)

**int** leerCodigoHospedaje() {

**return** Integer.*parseInt*(cboCodigoHospedaje.getSelectedItem().toString());

}

// Métodos que retornan valor (con parámetros)

**int** confirmar(String s) {

**return** JOptionPane.*showConfirmDialog*(**this**, s, "Alerta", 0, 1, **null**);

}

String formato(String cadena) {

**return** String.*format*("%-15s", cadena);

}

String formato(**int** entero) {

**return** String.*format*("%-10d", entero);

}

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

REPORTE

INGRESOS Y CONSUMOS PENDIENTES

**package** GUI;

**import** clases.\*;

**import** arreglos.\*;

**import** java.awt.EventQueue;

**import** java.awt.Font;

**import** java.util.Locale;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**public** **class** JdIngresosyConsumosPendientes **extends** JDialog {

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdIngresosyConsumosPendientes dialog = **new** JdIngresosyConsumosPendientes();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdIngresosyConsumosPendientes() {

setResizable(**false**);

setTitle("Reporte | Ingresos y Consumos Pendientes");

setBounds(100, 100, 700, 400);

getContentPane().setLayout(**null**);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 15, 675, 345);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

txtS.setFont(**new** Font("Monospaced", Font.***PLAIN***, 15));

scrollPane.setViewportView(txtS);

listar();

}

// Métodos tipo void (sin parámetros)

**void** imprimir() {

imprimir("");

}

**void** listar() {

ArregloIngresos ai = **new** ArregloIngresos();

ArregloSocio as = **new** ArregloSocio();

Ingresos i;

Socio s;

txtS.setText("");

**for** (**int** j=0; j<ai.tamaño(); j++) {

i = ai.obtener(j);

**if** (i.getEstado() == 0) {

s = as.buscar(i.getcodigoSocio());

imprimir("Código de Ingreso : " + i.getcodigoIngreso());

imprimir("Código de Socio : " + i.getcodigoSocio());

imprimir("Nombres : " + s.getNombres());

imprimir("Apellidos : " + s.getApellidos());

imprimir("Fecha de ingreso : " + i.getfechaIngreso()

+ " - " + i.gethoraIngreso());

imprimir();

ArregloConsumo ac=**new** ArregloConsumo(""+i.getcodigoIngreso());

Consumo c;

imprimir(" Cantidad Precio Importe");

**double** pago;

**for**(**int** a=0;a<ac.tamaño();a++){

c=ac.obtener(a);

pago=c.getCantidad()\*c.getPrecioUnitario();

ArregloProducto ap=**new** ArregloProducto();

Producto p=ap.buscar(c.getCodigoProducto());

imprimir(" "+formato(p.getDetalle())+formato(c.getCantidad())+formato(c.getPrecioUnitario())+formato(pago));

imprimir("TOTAL A PAGAR S/ "+formato(pago));

}

imprimir();

imprimir();

}

}

}

// Métodos tipo void (con parámetros)

**void** imprimir(String s) {

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

}

String formato(String cadena) {

**return** String.*format*("%-15s", cadena);

}

String formato(**int** entero) {

**return** String.*format*("%-10d", entero);

}

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

**void** mensaje(String s) {

JOptionPane.*showMessageDialog*(**this**, s, "Información", 1);

}

}

INGRESO Y CONSUMOS PAGADOS

**package** GUI;

**import** java.awt.EventQueue;

**import** java.util.Locale;

**import** clases.\*;

**import** arreglos.\*;

**import** javax.swing.JDialog;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**public** **class** JdIngresosyConsumosPagados **extends** JDialog {

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdIngresosyConsumosPagados dialog = **new** JdIngresosyConsumosPagados();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdIngresosyConsumosPagados() {

setBounds(100, 100, 759, 456);

setTitle("Reporte||Ingresos y Consumos Pagados");

getContentPane().setLayout(**null**);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 13, 717, 383);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

scrollPane.setViewportView(txtS);

listar();

}

// Métodos tipo void (sin parámetros)

**void** imprimir() {

imprimir("");

}

**void** listar() {

ArregloHospedaje ah=**new** ArregloHospedaje();

ArregloIngresos ai=**new** ArregloIngresos();

ArregloSocio as =**new** ArregloSocio();

Hospedajes h;

Ingresos i;

Socio s;

txtS.setText("");

**for**(**int** j=0;j<ah.tamaño();j++){

h=ah.obtener(j);

**if**(h.getEstado()==1){

i=ai.buscar(h.getCodigoIngreso());

s=as.buscar(i.getcodigoSocio());

imprimir("Código de Hospedaje : " + h.getCodigoHospedaje());

imprimir("Código de Socio : " + s.getCodigoSocio());

imprimir("Nombres : " + s.getNombres());

imprimir("Apellidos : " + s.getApellidos());

imprimir("Fecha de ingreso : " + i.getfechaIngreso()

+ " - " + i.gethoraIngreso());

imprimir("Costo Total del hospedaje :"+h.getcostoHospedaje());

imprimir();

}

}

}

// Métodos tipo void (con parámetros)

**void** imprimir(String s) {

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

}

// Métodos que retornan valor (con parámetros)

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

HOSPEDAJES PENDIENTES

**package** GUI;

**import** java.awt.EventQueue;

**import** java.util.Locale;

**import** javax.swing.JDialog;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** arreglos.ArregloHospedaje;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloSocio;

**import** clases.Hospedajes;

**import** clases.Ingresos;

**import** clases.Socio;

**public** **class** JdHospedajesPendientes **extends** JDialog {

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdHospedajesPendientes dialog = **new** JdHospedajesPendientes();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdHospedajesPendientes() {

setBounds(100, 100, 776, 475);

setTitle("Reporte||Hospedajes Pendientes");

getContentPane().setLayout(**null**);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 13, 734, 402);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

scrollPane.setViewportView(txtS);

listar();

}

// Métodos tipo void (sin parámetros)

**void** imprimir() {

imprimir("");

}

**void** listar() {

ArregloHospedaje ah=**new** ArregloHospedaje();

ArregloIngresos ai=**new** ArregloIngresos();

ArregloSocio as =**new** ArregloSocio();

Hospedajes h;

Ingresos i;

Socio s;

txtS.setText("");

**for**(**int** j=0;j<ah.tamaño();j++){

h=ah.obtener(j);

**if**(h.getEstado()==0){

i=ai.buscar(h.getCodigoIngreso());

s=as.buscar(i.getcodigoSocio());

imprimir("Código de Hospedaje : " + h.getCodigoHospedaje());

imprimir("Código de Socio : " + s.getCodigoSocio());

imprimir("Nombres : " + s.getNombres());

imprimir("Apellidos : " + s.getApellidos());

imprimir("Fecha de ingreso : " + i.getfechaIngreso()

+ " - " + i.gethoraIngreso());

imprimir("Costo Total del hospedaje :"+h.getcostoHospedaje());

imprimir();

}

}

}

// Métodos tipo void (con parámetros)

**void** imprimir(String s) {

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

}

// Métodos que retornan valor (con parámetros)

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

HOSPEDAJES PAGADOS

**package** GUI;

**import** java.awt.EventQueue;

**import** java.util.Locale;

**import** javax.swing.JDialog;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** arreglos.ArregloBungalow;

**import** arreglos.ArregloHospedaje;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloSocio;

**import** clases.Bungalow;

**import** clases.Hospedajes;

**import** clases.Ingresos;

**import** clases.Socio;

**import** libreria.Lib;

**public** **class** JdHospedajesPagados **extends** JDialog {

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdHospedajesPagados dialog = **new** JdHospedajesPagados();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdHospedajesPagados() {

setBounds(100, 100, 738, 448);

setTitle("Reporte||Hospedajes Pagados");

getContentPane().setLayout(**null**);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 13, 696, 375);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

scrollPane.setViewportView(txtS);

listar();

}

// Métodos tipo void (sin parámetros)

**void** imprimir() {

imprimir("");

}

**void** listar() {

ArregloHospedaje ah= **new** ArregloHospedaje();

ArregloSocio as= **new** ArregloSocio();

ArregloBungalow ab = **new** ArregloBungalow();

ArregloIngresos ai=**new** ArregloIngresos();

Hospedajes h;

Socio s;

Bungalow b;

Ingresos i;

txtS.setText("");

**for** (**int** j=0; j<ah.tamaño(); j++) {

h= ah.obtener(j);

**if** (h.getEstado() == 1) {

i=ai.buscar(h.getCodigoIngreso());

s = as.buscar(i.getcodigoSocio());

b = ab.buscar(h.getnumeroBungalow());

imprimir("--------------------------------------");

imprimir("Código de Hospedaje : " + h.getCodigoHospedaje());

imprimir("Código de Socio : " + s.getCodigoSocio());

imprimir("Nombres : " + s.getNombres());

imprimir("Apellidos : " + s.getApellidos());

imprimir("Número de bungalow : " + h.getnumeroBungalow());

imprimir("Categoría : " + Lib.*CategoriaBungalow*[b.getCategoria()]);

imprimir("Precio por día : S/ " + formato(b.getPrecioPorDia()));

imprimir("Fecha de ingreso : " + i.getfechaIngreso()

+ " - " + i.gethoraIngreso());

imprimir("----------------------------------------");

imprimir("Costo total :"+h.getcostoHospedaje());

}

}

}

// Métodos tipo void (con parámetros)

**void** imprimir(String s) {

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

}

// Métodos que retornan valor (con parámetros)

String formato(**double** real) {

**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

BOLETAS

BOLETAS

**package** GUI;

**import** java.awt.EventQueue;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** arreglos.ArregloBoleta;

**import** arreglos.ArregloConsumo;

**import** arreglos.ArregloHospedaje;

**import** arreglos.ArregloIngresos;

**import** arreglos.ArregloProducto;

**import** arreglos.ArregloSocio;

**import** clases.Boleta;

**import** clases.Consumo;

**import** clases.Hospedajes;

**import** clases.Ingresos;

**import** clases.Producto;

**import** clases.Socio;

**import** java.awt.event.ActionListener;

**import** java.util.Calendar;

**import** java.util.GregorianCalendar;

**import** java.util.Locale;

**import** java.awt.event.ActionEvent;

**import** javax.swing.JTextField;

**public** **class** JdBoletas **extends** JDialog **implements** ActionListener {

**private** JLabel lblCodigoIngreso;

**private** JComboBox<String> cboCodigoIngresoP;

**private** JButton btnBuscar;

**private** JScrollPane scrollPane;

**private** JTextArea txtS;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

JdBoletas dialog = **new** JdBoletas();

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

dialog.setVisible(**true**);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the dialog.

\*/

**public** JdBoletas() {

setBounds(100, 100, 614, 732);

setTitle("Boleta||Boletas");

getContentPane().setLayout(**null**);

lblCodigoIngreso = **new** JLabel("INGRESO");

lblCodigoIngreso.setBounds(12, 69, 119, 16);

getContentPane().add(lblCodigoIngreso);

cboCodigoIngresoP = **new** JComboBox<String>();

cboCodigoIngresoP.setBounds(99, 66, 140, 22);

getContentPane().add(cboCodigoIngresoP);

btnBuscar = **new** JButton("Buscar");

btnBuscar.addActionListener(**this**);

btnBuscar.setBounds(454, 33, 97, 25);

getContentPane().add(btnBuscar);

scrollPane = **new** JScrollPane();

scrollPane.setBounds(12, 101, 572, 571);

getContentPane().add(scrollPane);

txtS = **new** JTextArea();

scrollPane.setViewportView(txtS);

lblBoleta = **new** JLabel("Boleta : ");

lblBoleta.setBounds(12, 13, 56, 16);

getContentPane().add(lblBoleta);

txtBoleta = **new** JTextField();

txtBoleta.setEditable(**false**);

txtBoleta.setBounds(64, 10, 116, 22);

getContentPane().add(txtBoleta);

txtBoleta.setColumns(10);

colocarCodigosingresos();

txtBoleta.setText("60001");

}

ArregloIngresos ai=**new** ArregloIngresos();

ArregloBoleta ab=**new** ArregloBoleta();

ArregloSocio as=**new** ArregloSocio();

ArregloHospedaje ah=**new** ArregloHospedaje();

Ingresos in;

Socio s;

Boleta x;

Hospedajes h;

**double** pagoTotal;

**private** JLabel lblBoleta;

**private** JTextField txtBoleta;

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == btnBuscar) {

actionPerformedBtnBuscar(e);

}

}

**protected** **void** actionPerformedBtnBuscar(ActionEvent e) {

**int** codigoBoleta =leerCodigoboleta();

**int** codigoIngreso=leerCodigo();

Boleta nuevo = **new** Boleta(codigoBoleta, codigoIngreso, pagoTotal);

ab.adicionar(nuevo);

listar();

}

**void** listar(){

**double** pagoTotal;

x=ab.buscar(leerCodigoboleta());

in = ai.buscar(leerCodigo());

s=as.buscar(in.getcodigoSocio());

txtS.setText("");

imprimirEncabezado();

imprimir(" CODIGO BOLETA : "+leerCodigoboleta() );

imprimir("CODIGO SOCIO : "+in.getcodigoSocio());

imprimir("NOMBRE DEL SOCIO : "+s.getNombres());

imprimir("APELLIDOS DEL SOCIO : "+ s.getApellidos());

imprimir("DNI DEL SOCIO : "+s.getDni());

imprimir("------------------------------------------------");

imprimir("CODIGO INGRESO : "+in.getcodigoIngreso());

imprimir("FECHA DE INGRESO : "+in.getfechaIngreso());

imprimir("HORA DE INGRESO : "+in.gethoraIngreso());

imprimir("NUMERO DE INVITADOS : "+in.getnumeroInvitados());

imprimir("-------------------------------------------------");

ArregloConsumo ac=**new** ArregloConsumo(""+leerCodigo());

Consumo c;

**double** pago = 0;

**for**(**int** i=0;i<ac.tamaño();i++){

c=ac.obtener(i);

pago=c.getCantidad()\*c.getPrecioUnitario();

ArregloProducto ap=**new** ArregloProducto();

Producto p=ap.buscar(c.getCodigoProducto());

imprimir(" Cantidad Precio Importe");

imprimir(" "+formato(p.getDetalle())+formato(c.getCantidad())+formato(c.getPrecioUnitario())+formato(pago));

imprimir("Pago de Producto"+ pago);

}

imprimir("---------------------------------------------------");

imprimir("Pago de ingreso :"+in.getcostoIngreso());

imprimir("-----------------------------------------------------");

pagoTotal=pago+in.getcostoIngreso();

imprimir("PAGO TOTAL : "+pagoTotal);

}

**int** leerCodigo(){

**return** Integer.*parseInt*(cboCodigoIngresoP.getSelectedItem().toString());

}

**int** leerCodigoboleta(){

**return** Integer.*parseInt*(txtBoleta.getText());

}

**void** colocarCodigosingresos() {

Ingresos i;

**for** (**int** j=0; j<ai.tamaño(); j++) {

i = ai.obtener(j);

**if** (i.getEstado() == 1)

cboCodigoIngresoP.addItem("" + i.getcodigoIngreso());

}

}

**void** imprimirEncabezado(){

txtS.setText("");

imprimir(" HOTEL HILTON - BOLETA DE PAGO");

imprimir(" DEL " + fechaBoleta() + " - " + horaBoleta());

imprimir(" ===================================");

}

**void** imprimir(String s) {

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

}

String fechaBoleta() {

Calendar calendario = **new** GregorianCalendar();

**return** String.*format*("%02d/",calendario.get(Calendar.***DAY\_OF\_MONTH***)) +

String.*format*("%02d/",calendario.get(Calendar.***MONTH***) + 1) +

calendario.get(Calendar.***YEAR***);

}

String horaBoleta() {

Calendar calendario = **new** GregorianCalendar();

**return** String.*format*("%02d:",calendario.get(Calendar.***HOUR\_OF\_DAY***)) +

String.*format*("%02d:",calendario.get(Calendar.***MINUTE***)) +

String.*format*("%02d",calendario.get(Calendar.***SECOND***));

}

String formato(String cadena) {

**return** String.*format*("%-15s", cadena);

}

String formato(**int** entero) {

**return** String.*format*("%-10d", entero);

}

String formato(**double** real) {

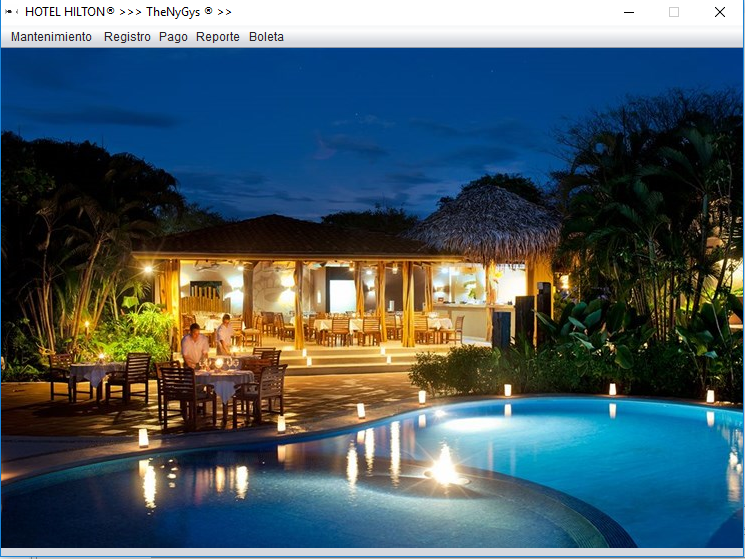
**return** String.*format*(Locale.***US***, "%-10.2f", real);

}

}

#### Definición

* Menú principal:

Donde podemos apreciar que contiene las siguientes cajas de dialogo.

a)Mantenimiento

Socio

Producto

Bungalow

b) Registro

Ingreso

Consumo

Hospedaje

c) Pago

Ingresos y consumos

Hospedajes

d) Reporte

Ingresos y consumos pendientes

Ingresos y consumos pagados

Hospedajes pendientes

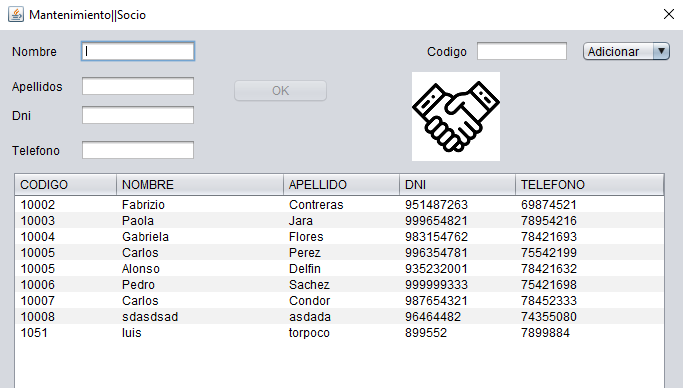
Hospedajes pagados

e) Boleta

Boletas

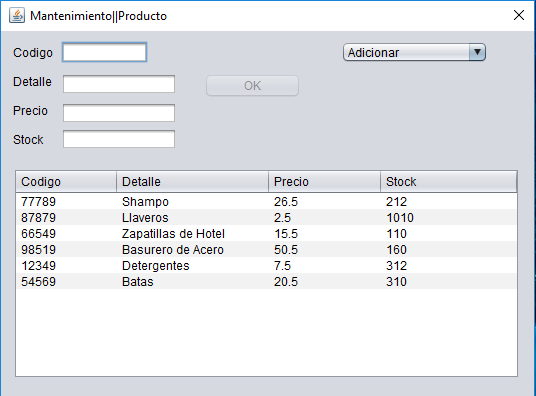
* Mantenimiento

Socio

Se podrá Adicionara,Modificara,Eliminar, Consulatar datos del Socio mediante el código.

Producto

Se podrá Adicionara,Modificara,Eliminar, Consulatar datos del Producto .



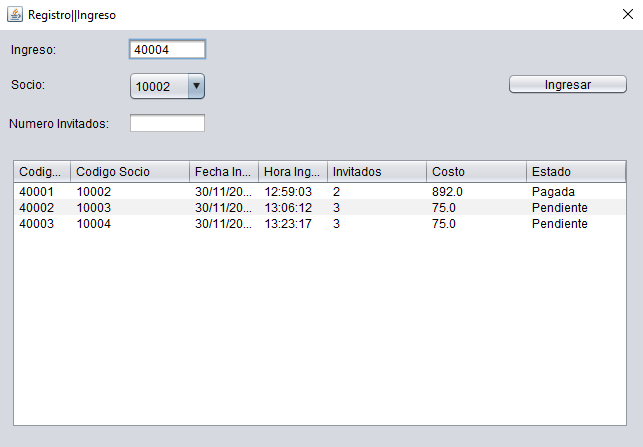
Bungalow

Se podrá Adicionara,Modificara,Eliminar, Consulatar datos del Bungalow .

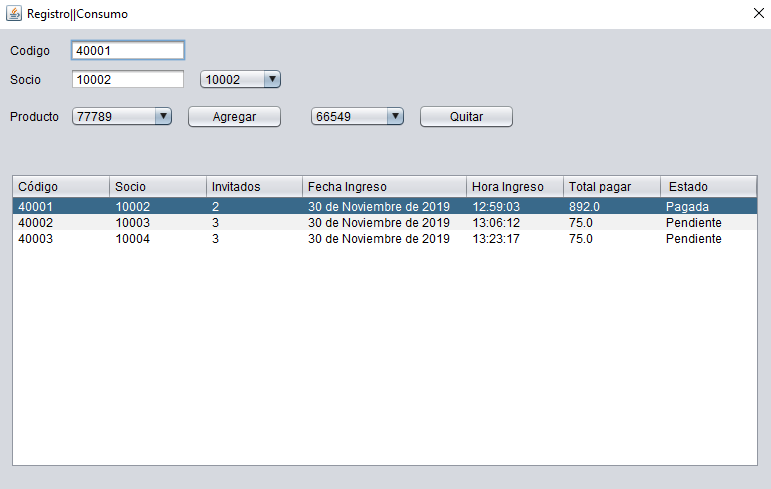


* Registro

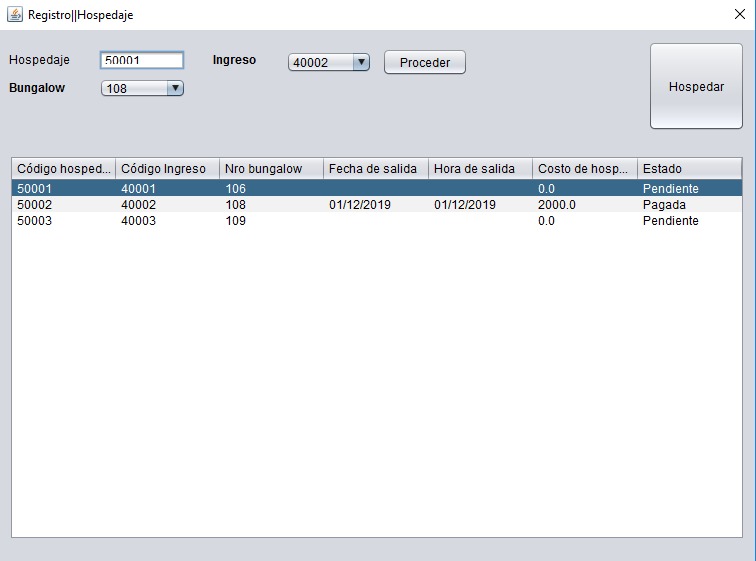
Ingreso

Se ingresaran el numero de invitados ,se pondrá el estado en pendiente y se tomar la fecha y hora del sistema.

Consumo

Se ubicara un socio cuyo estado de ingreso sea pendientey se registrara el producto con su total a pagar.

Hospedaje

Se generara un código de hospedaje.

- Escoger uno de los bungalows libres y fijar el estado del bungalow en 1 (Ocupado).

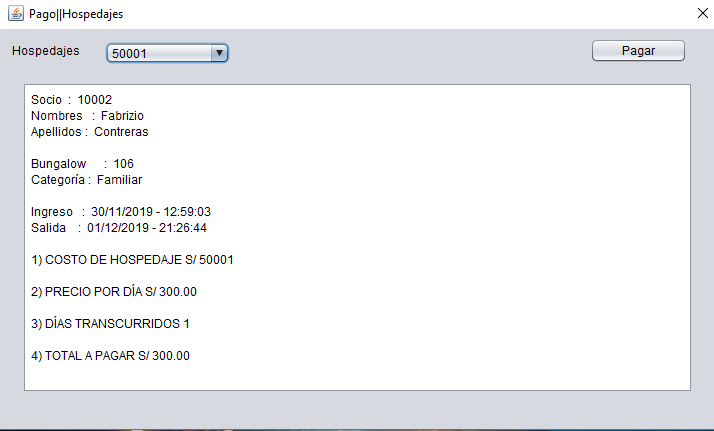
- Fijar el estado del hospedaje en pendiente.

- Un socio no podrá hospedarse en más de un bungalow a la vez.

* Pago

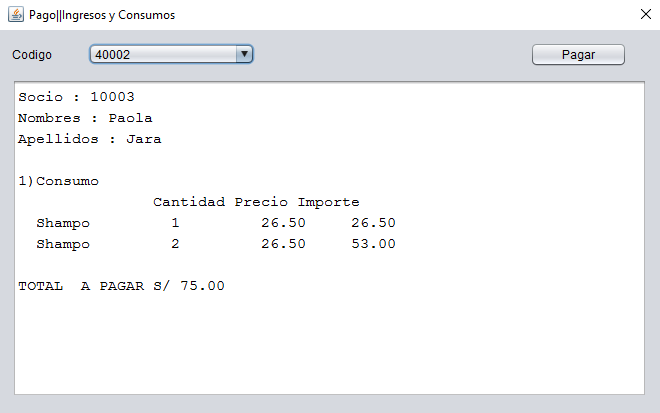
Hospedaje

Mediante el código de Hospedaje se listara el código del socio y datos del mismo para su posterior pago



Ingresos y Consumos

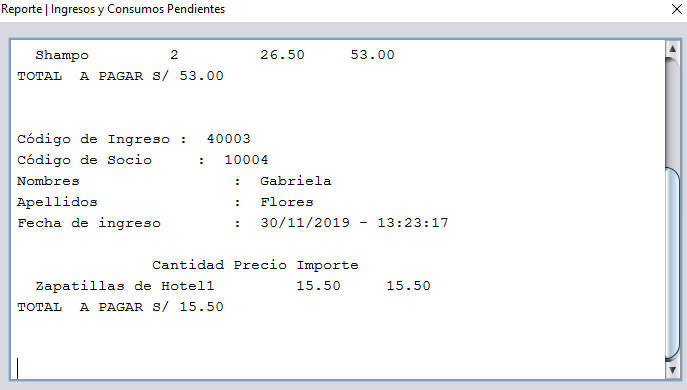
Mediante un código de ingreso se listara el código del socio y datos del mismo para su posterior pago



* Reporte

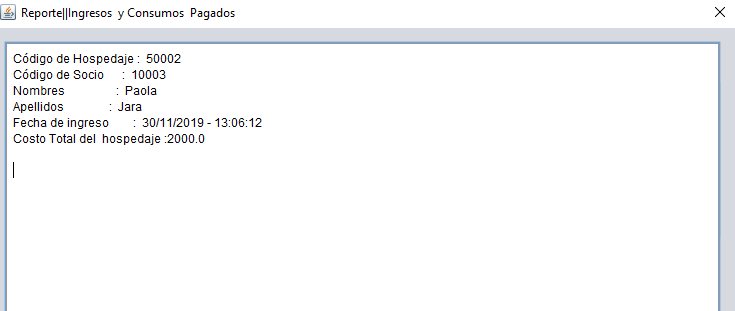
Ingreso y Consumos pendientes

Se reportara el importe total de los productos consumidos pendientes

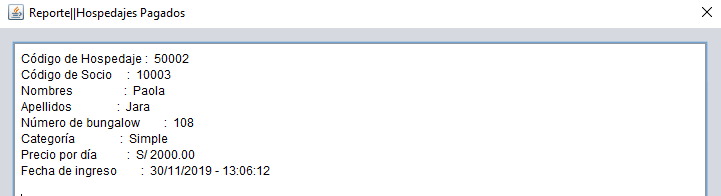


Ingresos y consumos pagados

Se listaran los ingresos y consumos pagados

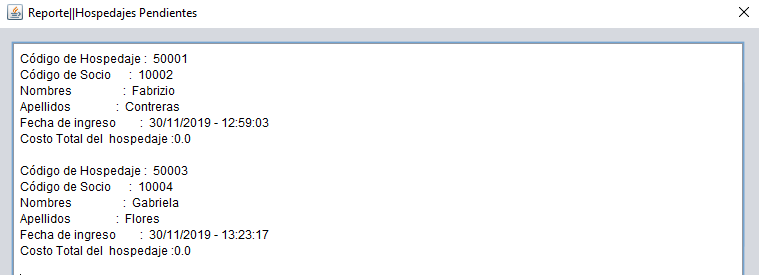


Hospedajes Pagados

Ser listara los datos con respecto a los hospedajes pagados

Hospedajes Pendientes

Ser listara los datos con respecto a los hospedajes pendientes



#### Boleta

#### Se podrá objervar una boleta de venta mediante un código de socio

#### 

#### Conclusiones

* Primero se concluye que el uso de este programa ,optimiza mucha el tiempo de reserva y registro.
* Tambiee que el programa necesitara una siguiente versión en la cual incluya algunos aspectos como el botón Imprimir.
* Por ultimo para la correcta programación del código se debe dirigir, adecuadamente, las partes de este, a un grupo organizado.

#### Recomendaciones

Se recomienda al grupo de personas que quiera desarrollar un proyecto similar a este:

* Delegue las partes del proyecto de forma ordenada.
* Al realizar el código utilicen métodos más eficientes ya que esto mejora el proceso del programa.
* Con respecto a las cajas de diálogo, que evolucionen y mejoren al beneficio del empleado quien la va a manejar , para que de esta forma tenga una rapida curva de aprendizaje con el programa .En síntesis que sean cada vez mas comprensibles y fáciles de entender (intuitivas).

#### Bibliografía

* <https://www.tutorialesprogramacionya.com/javaya/detalleconcepto.php?punto=3&codigo=76&inicio=0>
* <https://cibertec.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_1295855_1&content_id=_6514841_1>
* <https://cibertec.blackboard.com/bbcswebdav/pid-6514838-dt-content-rid-37329029_1/institution/PECI01%20-%20CIBERTEC/CIBERTEC/MASTER/PECI01.CIBERTEC.MASTER.AC.2326/Manual_Introducci%C3%B3n_a_la_Algoritmia.pdf>
* <https://cibertec.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_1295855_1&content_id=_6514842_1>