**PRÁCTICA 3er TRIMESTRE**

**“Base de Datos. Gestión de Empresa"**

**Alumno:** Sergio Martínez Más. **DNI:** 34.039.949-H.

**Asignatura:** Programación (1er Curso Técnico Superior D.A.M.)

**Fecha de entrega:** 03 de jUNIO de 2.013.

|  |
| --- |
| **ÍNDICE** |

1. **Base de datos**
2. **Puente de acceso**
3. **Códigos**
4. **Manual de uso e instalación. Pruebas de funcionamiento**
5. **Creación del un archivo ejecutable y su instalación**

|  |
| --- |
| **1. BASE DE DATOS.**  **Nombre de la base de datos: practica\_hotelClinic**  **Usuarios:**  **Nombre usuario: administrador Contraseña: 1111**  **Nombre usuario: gerente Contraseña: 2222**  **Nombre usuario: usuario Contraseña: 3333** |
|  |
| CREATE DATABASE IF NOT EXISTS `practica\_hotelclinic` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;  USE `practica\_hotelclinic`;  -- MySQL dump 10.13 Distrib 5.5.16, for Win32 (x86)  --  -- Host: localhost Database: practica\_hotelclinic  -- ------------------------------------------------------  -- Server version 5.5.30  /\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;  /\*!40101 SET NAMES utf8 \*/;  /\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;  /\*!40103 SET TIME\_ZONE='+00:00' \*/;  /\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;  /\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;  /\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;  /\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;  --  -- Table structure for table `articulos`  --  DROP TABLE IF EXISTS `articulos`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `articulos` (  `IdArticulo` varchar(10) NOT NULL DEFAULT '',  `NombreArticulo` varchar(45) DEFAULT NULL,  `DescripcionArticulo` varchar(100) DEFAULT NULL,  `PrecioCompraArticulo` float(10,2) DEFAULT NULL,  `PrecioVentaArticulo` float(10,1) DEFAULT NULL,  `IdIVAFK` varchar(10) DEFAULT NULL,  PRIMARY KEY (`IdArticulo`),  KEY `IdIVAFK` (`IdIVAFK`),  CONSTRAINT `articulos\_ibfk\_1` FOREIGN KEY (`IdIVAFK`) REFERENCES `tipos\_iva` (`IdIVA`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `articulos`  --  LOCK TABLES `articulos` WRITE;  /\*!40000 ALTER TABLE `articulos` DISABLE KEYS \*/;  INSERT INTO `articulos` VALUES ('col150','Colchon de 150\*200\*35','Colchon viscoelastico',210.00,325.0,'iva8'),('col80','Colchon de 80\*180\*20','Colchon viscoelastico ',175.00,225.0,'iva30'),('col90','Colchon de 90\*200\*30','Colchon viscoelastico ',160.00,210.0,'iva25'),('tapi150','Base de 150\*200 b','Base tapizada relleno',155.00,100.0,'iva21'),('tapi80','Base de 80\*180 beig','Base tapizada relleno',165.00,195.0,'iva21');  /\*!40000 ALTER TABLE `articulos` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `clientes`  --  DROP TABLE IF EXISTS `clientes`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `clientes` (  `CifCliente` varchar(9) NOT NULL DEFAULT '',  `NombreCliente` varchar(45) DEFAULT NULL,  `DireccionCliente` varchar(45) DEFAULT NULL,  `CiudadCliente` varchar(45) DEFAULT NULL,  `CpCliente` int(11) DEFAULT NULL,  `TelefonoCliente` int(11) DEFAULT NULL,  PRIMARY KEY (`CifCliente`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `clientes`  --  LOCK TABLES `clientes` WRITE;  /\*!40000 ALTER TABLE `clientes` DISABLE KEYS \*/;  INSERT INTO `clientes` VALUES ('40000000B','Colchon. Almeria','Calle central 21','Almeria',33335,878787878),('41255458B','MaxiColchon S.L.','Avenida Salud 21','Cadiz',25254,111000111),('45545858B','FactoryColchon','Calle de Alcala 1','Malaga',54800,11111111),('78545896B','Colchones blanco','Calle Pajarito 28','Huelva',87879,111754111),('85858574A','Colchonrelax','Calle Sol 25','Sevilla',10000,645857214);  /\*!40000 ALTER TABLE `clientes` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `facturas\_clientes`  --  DROP TABLE IF EXISTS `facturas\_clientes`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `facturas\_clientes` (  `IdFacturaCliente` varchar(10) NOT NULL DEFAULT '',  `FechaFacturaCliente` date DEFAULT NULL,  `BaseImponibleFacturaCliente` float(10,2) DEFAULT NULL,  `IvaFacturaCliente` float(10,2) DEFAULT NULL,  `TotalFacturaCliente` float(10,2) DEFAULT NULL,  `CifClienteFK` varchar(9) DEFAULT NULL,  PRIMARY KEY (`IdFacturaCliente`),  KEY `CifClienteFK` (`CifClienteFK`),  CONSTRAINT `facturas\_clientes\_ibfk\_1` FOREIGN KEY (`CifClienteFK`) REFERENCES `clientes` (`CifCliente`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `facturas\_clientes`  --  LOCK TABLES `facturas\_clientes` WRITE;  /\*!40000 ALTER TABLE `facturas\_clientes` DISABLE KEYS \*/;  /\*!40000 ALTER TABLE `facturas\_clientes` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `facturas\_proveedores`  --  DROP TABLE IF EXISTS `facturas\_proveedores`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `facturas\_proveedores` (  `IdFacturaProveedor` varchar(10) NOT NULL DEFAULT '',  `FechaFacturaProveedor` date DEFAULT NULL,  `BaseImponibleFacturaProveedor` float(10,2) DEFAULT NULL,  `IvaFacturaProveedor` float(10,2) DEFAULT NULL,  `TotalFacturaProveedor` float(10,2) DEFAULT NULL,  `CifProveedorFK` varchar(9) DEFAULT NULL,  PRIMARY KEY (`IdFacturaProveedor`),  KEY `CifProveedorFK` (`CifProveedorFK`),  CONSTRAINT `facturas\_proveedores\_ibfk\_1` FOREIGN KEY (`CifProveedorFK`) REFERENCES `proveedores` (`CifProveedor`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `facturas\_proveedores`  --  LOCK TABLES `facturas\_proveedores` WRITE;  /\*!40000 ALTER TABLE `facturas\_proveedores` DISABLE KEYS \*/;  /\*!40000 ALTER TABLE `facturas\_proveedores` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `lineas\_facturas\_clientes`  --  DROP TABLE IF EXISTS `lineas\_facturas\_clientes`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `lineas\_facturas\_clientes` (  `IdArticuloFK` varchar(10) NOT NULL DEFAULT '',  `IdFacturaClienteFK` varchar(10) NOT NULL DEFAULT '',  `UdsArticulos` int(11) DEFAULT NULL,  PRIMARY KEY (`IdArticuloFK`,`IdFacturaClienteFK`),  KEY `IdFacturaClienteFK` (`IdFacturaClienteFK`),  CONSTRAINT `lineas\_facturas\_clientes\_ibfk\_1` FOREIGN KEY (`IdArticuloFK`) REFERENCES `articulos` (`IdArticulo`),  CONSTRAINT `lineas\_facturas\_clientes\_ibfk\_2` FOREIGN KEY (`IdFacturaClienteFK`) REFERENCES `facturas\_clientes` (`IdFacturaCliente`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `lineas\_facturas\_clientes`  --  LOCK TABLES `lineas\_facturas\_clientes` WRITE;  /\*!40000 ALTER TABLE `lineas\_facturas\_clientes` DISABLE KEYS \*/;  /\*!40000 ALTER TABLE `lineas\_facturas\_clientes` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `lineas\_facturas\_proveedores`  --  DROP TABLE IF EXISTS `lineas\_facturas\_proveedores`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `lineas\_facturas\_proveedores` (  `IdArticuloFK` varchar(10) NOT NULL DEFAULT '',  `IdFacturaProveedorFK` varchar(10) NOT NULL DEFAULT '',  `UdsArticulos` int(11) DEFAULT NULL,  PRIMARY KEY (`IdArticuloFK`,`IdFacturaProveedorFK`),  KEY `IdFacturaProveedorFK` (`IdFacturaProveedorFK`),  CONSTRAINT `lineas\_facturas\_proveedores\_ibfk\_1` FOREIGN KEY (`IdArticuloFK`) REFERENCES `articulos` (`IdArticulo`),  CONSTRAINT `lineas\_facturas\_proveedores\_ibfk\_2` FOREIGN KEY (`IdFacturaProveedorFK`) REFERENCES `facturas\_proveedores` (`IdFacturaProveedor`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `lineas\_facturas\_proveedores`  --  LOCK TABLES `lineas\_facturas\_proveedores` WRITE;  /\*!40000 ALTER TABLE `lineas\_facturas\_proveedores` DISABLE KEYS \*/;  /\*!40000 ALTER TABLE `lineas\_facturas\_proveedores` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `proveedores`  --  DROP TABLE IF EXISTS `proveedores`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `proveedores` (  `CifProveedor` varchar(9) NOT NULL DEFAULT '',  `NombreProveedor` varchar(45) DEFAULT NULL,  `DireccionProveedor` varchar(45) DEFAULT NULL,  `CiudadProveedor` varchar(45) DEFAULT NULL,  `CpProveedor` int(11) DEFAULT NULL,  `TelefonoProveedor` int(11) DEFAULT NULL,  PRIMARY KEY (`CifProveedor`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `proveedores`  --  LOCK TABLES `proveedores` WRITE;  /\*!40000 ALTER TABLE `proveedores` DISABLE KEYS \*/;  INSERT INTO `proveedores` VALUES ('41255000F','Sum. de descan.','Calle Madrid 21','Zaragoza',85478,23254879),('41856980D','Colchones S.A.','Calle General 54','Madrid',11111,555555555),('75655000F','Colchones And.','Calle Central 50','Sevilla',41525,452325784),('77855070Y','ImnoColchon S.A.','Calle Sevilla 74','Malaga',78965,123456789),('85255000P','S. de aluminio S.A.','Calle Barcelo 87','Toledo',45632,55555555);  /\*!40000 ALTER TABLE `proveedores` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `tipos\_iva`  --  DROP TABLE IF EXISTS `tipos\_iva`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `tipos\_iva` (  `IdIVA` varchar(10) NOT NULL DEFAULT '',  `DescripcionIVA` varchar(45) DEFAULT NULL,  `CantidadIVA` float(4,2) DEFAULT NULL,  PRIMARY KEY (`IdIVA`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `tipos\_iva`  --  LOCK TABLES `tipos\_iva` WRITE;  /\*!40000 ALTER TABLE `tipos\_iva` DISABLE KEYS \*/;  INSERT INTO `tipos\_iva` VALUES ('iva10','Iva Reducido',10.00),('iva21','Iva Genera',21.00),('iva25','Iva Especial',25.00),('iva30','Iva S.Especial',30.00),('iva8','Iva S.Reducido',8.00);  /\*!40000 ALTER TABLE `tipos\_iva` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Table structure for table `usuarios`  --  DROP TABLE IF EXISTS `usuarios`;  /\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;  /\*!40101 SET character\_set\_client = utf8 \*/;  CREATE TABLE `usuarios` (  `idUsuario` varchar(9) DEFAULT NULL,  `nombreUsuario` varchar(20) DEFAULT NULL,  `contrasenaUsuario` varchar(12) DEFAULT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  /\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;  --  -- Dumping data for table `usuarios`  --  LOCK TABLES `usuarios` WRITE;  /\*!40000 ALTER TABLE `usuarios` DISABLE KEYS \*/;  INSERT INTO `usuarios` VALUES ('user1','administrador','1111'),('user2','gerente','2222'),('user3','usuario','3333');  /\*!40000 ALTER TABLE `usuarios` ENABLE KEYS \*/;  UNLOCK TABLES;  --  -- Dumping routines for database 'practica\_hotelclinic'  --  /\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;  /\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;  /\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;  /\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;  /\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;  /\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;  /\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;  /\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;  -- Dump completed on 2013-05-28 22:31:44 |

|  |
| --- |
| **2. PUENTE DE ACCESO** |
|  |
| A continuación de adjunta imagen de los datos empleados en el puente. La contraseña será la que se tenga en cada equipo como root.  **Data Source Name: hotelClinic** |

|  |
| --- |
| **3. CODIGOS** |

|  |
| --- |
| **panelAcceso (main)** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.Frame;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class panelAcceso extends Frame implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONTROL DE ACCESO");  Label nombreUsuario = new Label("Nombre de Usuario (máximo 20 carácteres)");  Label contrasena = new Label("Contraseña (máximo 9 carácteres)");  Label informacion = new Label("Algún campo es incorrecto !!");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  //TextFiled  TextField textoNombreUsuario = new TextField("Nombre de usuario...",20);  TextField textoContrasena = new TextField("Contraseña...",9);  //Button  Button entrar = new Button ("Entrar");  Button salir = new Button ("Salir");  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion = new Dialog (this,"Información",true);  //creacion objetos paneles adminstrador, gerente y usuario  static panelAcceso acceso = new panelAcceso();    //datos de usuarios  String nUsuarios[] = new String [3];  String cUsuarios[] = new String [3];    public panelAcceso()//constructor  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  nombreUsuario.setFont(fuenteLabelNegrita);  contrasena.setFont(fuenteLabelNegrita);  dialogInformacion.setBackground(Color.red);  dialogInformacion.setForeground(Color.white);  dialogInformacion.setFont(fuenteLabelNegrita);  logo.setForeground(Color.blue);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  //escuchadores  addWindowListener(this);  salir.addActionListener(this);  entrar.addActionListener(this);  dialogInformacion.addWindowListener(this);  //gestion cuadro de dialogo  dialogInformacion.setLayout(new FlowLayout());  dialogInformacion.setSize(200,100);  dialogInformacion.setResizable(false);  dialogInformacion.setLocation(250,250);  dialogInformacion.add(informacion);  setTitle("CONTROL DE ACCESO");  setVisible(true);  setSize(500,350);//horizontal,vertical  setResizable(false);  setLocation(120,120);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.CENTER;  add(LbTitulo,gbc);  //situacion label nombre usuario  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  add(nombreUsuario,gbc);  //situacion txt usuario  gbc.gridx=1;  gbc.gridy=1;  add(textoNombreUsuario,gbc);  //situacion label contraseña usuario  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(contrasena,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  textoContrasena.setEchoChar('\*');  add(textoContrasena,gbc);  //botoneS  gbc.gridx=0;  gbc.gridy=3;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(entrar,gbc);  gbc.gridx=1;  gbc.gridy=3;  add(salir,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=4;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);    //consulta datos de usuarios a la BD  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT nombreUsuario,contrasenaUsuario FROM usuarios");  //Recorrer los registros obtenidos  int i=0;  while(rs.next())  {  nUsuarios[i] = rs.getString("nombreUsuario");  cUsuarios[i] = rs.getString("contrasenaUsuario");  i++;  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public static void main(String[] args)  {  acceso.setVisible(true);  }  public void actionPerformed(ActionEvent ap)  {  if(salir.equals(ap.getSource()))  {  System.exit(0);  }  if(entrar.equals(ap.getSource()))  {  if ((nUsuarios[0].equals(textoNombreUsuario.getText())) && (cUsuarios[0].equals(textoContrasena.getText())))  {  panelAdministrador administrador = new panelAdministrador();  acceso.dispose();    }  else if ((nUsuarios[1].equals(textoNombreUsuario.getText())) && (cUsuarios[1].equals(textoContrasena.getText())))  {  panelGerente gerente = new panelGerente();  acceso.dispose();  }  else if ((nUsuarios[2].equals(textoNombreUsuario.getText())) && (cUsuarios[2].equals(textoContrasena.getText())))  {  panelUsuario usuario = new panelUsuario();  acceso.dispose();  }  else  {  dialogInformacion.setVisible(true);  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion.equals(wcg.getSource()))  {  dialogInformacion.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **panelUsuario** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.Frame;  import java.awt.Graphics;  import java.awt.Image;  import java.awt.Label;  import java.awt.Menu;  import java.awt.MenuBar;  import java.awt.MenuItem;  import java.awt.PrintJob;  import java.awt.TextArea;  import java.awt.Toolkit;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.File;  import java.io.IOException;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  import javax.swing.text.Style;  public class panelUsuario extends Frame implements WindowListener,ActionListener  {  // Declarar el objeto Image  Image imagen;  // Declarar el objeto Toolkit para manejo de imágenes  Toolkit herramienta;  //fuentes  Font fuenteGeneral = new Font("Arial", Font.PLAIN, 12);  Font fuenteLabel = new Font("Arial", Font.BOLD, 12);  //label  Label informacionSeleccione = new Label("Seleccione alguna opción !!");  Label informacionImpresionCancelada = new Label("Impresion Cancelada !!");  //menu  MenuBar barra = new MenuBar();  Menu facturasClientes = new Menu("Facturas de Clientes");  Menu facturasProveedores = new Menu("Facturas de Proveedores");  MenuItem altaFacturaCliente = new MenuItem ("Alta de Factura Cliente");  MenuItem bajaFacturaCliente = new MenuItem ("Baja de Factura Cliente");  MenuItem modificacionFacturaCliente = new MenuItem ("Modificación de Factura Cliente");  MenuItem consultaFacturaCliente = new MenuItem ("Consulta de Factura Cliente");  MenuItem imprimirFacturasCliente = new MenuItem ("Imprimir Facturas de Clientes");  MenuItem altaFacturaProveedor = new MenuItem ("Alta de Factura Proveedor");  MenuItem bajaFacturaProveedor = new MenuItem ("Baja de Factura Proveedor");  MenuItem modificacionFacturaProveedor = new MenuItem ("Modificación de Factura Proveedor");  MenuItem consultaFacturaProveedor = new MenuItem ("Consulta de Factura Proveedor");  MenuItem imprimirFacturasProveedor = new MenuItem ("Imprimir Facturas de Proveedores");  //ventana de dialogo  Dialog dialogInformacionSeleccione = new Dialog (this,"Información",true);  public panelUsuario()//constructor  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  // Establecer el método de trabajo con imágenes  herramienta = getToolkit();  // Especificar la ruta de la imagen  imagen = herramienta.getImage("img/hcUsuario.jpg");  setTitle("PANEL DE USUARIO");  setVisible(true);  setSize(900,800);//horizontal,vertical  setResizable(false);  setLocation(120,120);  //añadir menu  setMenuBar(barra);  barra.add(facturasClientes);  barra.add(facturasProveedores);  facturasClientes.add(altaFacturaCliente);  facturasClientes.add(bajaFacturaCliente);  facturasClientes.add(modificacionFacturaCliente);  facturasClientes.add(consultaFacturaCliente);  facturasClientes.add(imprimirFacturasCliente);  facturasProveedores.add(altaFacturaProveedor);  facturasProveedores.add(bajaFacturaProveedor);  facturasProveedores.add(modificacionFacturaProveedor);  facturasProveedores.add(consultaFacturaProveedor);  facturasProveedores.add(imprimirFacturasProveedor);  dialogInformacionSeleccione.setBackground(new Color (255, 97, 3));  dialogInformacionSeleccione.setForeground(Color.white);  dialogInformacionSeleccione.setFont(fuenteLabel);  //escuchadores  addWindowListener(this);  dialogInformacionSeleccione.addWindowListener(this);  altaFacturaCliente.addActionListener(this);  bajaFacturaCliente.addActionListener(this);  modificacionFacturaCliente.addActionListener(this);  consultaFacturaCliente.addActionListener(this);  imprimirFacturasCliente.addActionListener(this);  altaFacturaProveedor.addActionListener(this);  bajaFacturaProveedor.addActionListener(this);  modificacionFacturaProveedor.addActionListener(this);  consultaFacturaProveedor.addActionListener(this);  imprimirFacturasProveedor.addActionListener(this);  //gestion cuadro de dialogo  dialogInformacionSeleccione.setLayout(new FlowLayout());  dialogInformacionSeleccione.setSize(200,100);  dialogInformacionSeleccione.setResizable(false);  dialogInformacionSeleccione.setLocation(120,120);  dialogInformacionSeleccione.add(informacionSeleccione);  }  public void paint(Graphics g)  {  // Dibujar la imagen  g.drawImage(imagen,200,150,this);  }  public void actionPerformed(ActionEvent ap)  {  //facturas clientes  if(altaFacturaCliente.equals(ap.getSource()))  {  altaFacturaCliente alFactCli = new altaFacturaCliente(null);  }  if(bajaFacturaCliente.equals(ap.getSource()))  {  bajaFacturaCliente baFactCli = new bajaFacturaCliente(null);  }  if(modificacionFacturaCliente.equals(ap.getSource()))  {  modificacionFacturaCliente moFactCli = new modificacionFacturaCliente(null);  }  if(consultaFacturaCliente.equals(ap.getSource()))  {  consultaFacturaCliente coFactCli = new consultaFacturaCliente();  }  if(imprimirFacturasCliente.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob = this.getToolkit().getPrintJob(this, "", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg = pjob.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg.drawString("RESUMEN FACTURAS DE CLIENTES", 150, 50);  pg.setFont(new Font("Arial",Font.BOLD,12));  pg.drawString("Código", 80, 100);  pg.drawString("Fecha", 155, 100);  pg.drawString("Base", 230, 100);  pg.drawString("IVA", 290, 100);  pg.drawString("Total", 340, 100);  pg.drawString("CIF", 420, 100);  pg.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_clientes");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaCliente"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaCliente"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaCliente"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaCliente"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaCliente"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifClienteFK");  pg.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //facturas proveedores  if(altaFacturaProveedor.equals(ap.getSource()))  {  altaFacturaProveedor alFactProv = new altaFacturaProveedor(null);  }  if(bajaFacturaProveedor.equals(ap.getSource()))  {  bajaFacturaProveedor baFactProv = new bajaFacturaProveedor(null);  }  if(modificacionFacturaProveedor.equals(ap.getSource()))  {  modificacionFacturaProveedor moFactProv = new modificacionFacturaProveedor(null);  }  if(consultaFacturaProveedor.equals(ap.getSource()))  {  consultaFacturaProveedor coFactProv = new consultaFacturaProveedor();  }  if(imprimirFacturasProveedor.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob = this.getToolkit().getPrintJob(this, "", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg = pjob.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg.drawString("RESUMEN FACTURAS DE PROVEEDORES", 130, 50);  pg.setFont(new Font("Arial",Font.BOLD,12));  pg.drawString("Código", 80, 100);  pg.drawString("Fecha", 160, 100);  pg.drawString("Base", 240, 100);  pg.drawString("IVA", 310, 100);  pg.drawString("Total", 360, 100);  pg.drawString("CIF", 435, 100);  pg.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_proveedores");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaProveedor"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaProveedor"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaProveedor"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaProveedor"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaProveedor"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifProveedorFK");  pg.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  panelAcceso acceso = new panelAcceso();  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **panelGerente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.Frame;  import java.awt.Graphics;  import java.awt.Image;  import java.awt.Label;  import java.awt.Menu;  import java.awt.MenuBar;  import java.awt.MenuItem;  import java.awt.PrintJob;  import java.awt.Toolkit;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class panelGerente extends Frame implements WindowListener,ActionListener  {  // Declarar el objeto Image  Image imagen;  // Declarar el objeto Toolkit para manejo de imágenes  Toolkit herramienta;  //label  Label informacion = new Label("Seleccione alguna opción !!");  //fuentes  Font fuenteGeneral = new Font("Arial", Font.PLAIN, 12);  Font fuenteLabel = new Font("Arial", Font.BOLD, 12);  //menu  MenuBar barra = new MenuBar();  Menu articulos = new Menu("Artículos");  Menu clientes = new Menu("Clientes");  Menu proveedores = new Menu("Proveedores");  Menu facturasClientes = new Menu("Facturas de Clientes");  Menu facturasProveedores = new Menu("Facturas de Proveedores");  MenuItem altaArticulo = new MenuItem ("Alta de Artículo");  MenuItem bajaArticulo = new MenuItem ("Baja de Artículo");  MenuItem modificacionArticulo = new MenuItem ("Modificación de Artículo");  MenuItem consultaArticulo = new MenuItem ("Consulta de Artículo");  MenuItem imprimirArticulos = new MenuItem ("Imprimir Artículos");  MenuItem altaCliente = new MenuItem ("Alta de Cliente");  MenuItem bajaCliente = new MenuItem ("Baja de Cliente");  MenuItem modificacionCliente = new MenuItem ("Modificación de Cliente");  MenuItem consultaCliente = new MenuItem ("Consulta de Cliente");  MenuItem imprimirClientes = new MenuItem ("Imprimir Clientes");  MenuItem altaProveedor = new MenuItem ("Alta de Proveedor");  MenuItem bajaProveedor = new MenuItem ("Baja de Proveedor");  MenuItem modificacionProveedor = new MenuItem ("Modificación de Proveedor");  MenuItem consultaProveedor = new MenuItem ("Consulta de Proveedor");  MenuItem imprimirProveedores = new MenuItem ("Imprimir Proveedores");  MenuItem altaFacturaCliente = new MenuItem ("Alta de Factura Cliente");  MenuItem bajaFacturaCliente = new MenuItem ("Baja de Factura Cliente");  MenuItem modificacionFacturaCliente = new MenuItem ("Modificación de Factura Cliente");  MenuItem consultaFacturaCliente = new MenuItem ("Consulta de Factura Cliente");  MenuItem imprimirFacturasCliente = new MenuItem ("Imprimir Facturas de Clientes");  MenuItem altaFacturaProveedor = new MenuItem ("Alta de Factura Proveedor");  MenuItem bajaFacturaProveedor = new MenuItem ("Baja de Factura Proveedor");  MenuItem modificacionFacturaProveedor = new MenuItem ("Modificación de Factura Proveedor");  MenuItem consultaFacturaProveedor = new MenuItem ("Consulta de Factura Proveedor");  MenuItem imprimirFacturasProveedor = new MenuItem ("Imprimir Facturas de Proveedores");  //ventana de dialogo  Dialog dialogInformacion = new Dialog (this,"Información",true);  public panelGerente()//constructor  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  // Establecer el método de trabajo con imágenes  herramienta = getToolkit();  // Especificar la ruta de la imagen  imagen = herramienta.getImage("img/hcGerente.jpg");  setTitle("PANEL DE GERENTE");  setVisible(true);  setSize(900,800);//horizontal,vertical  setResizable(false);  setLocation(120,120);  //añadir menu  setMenuBar(barra);  barra.add(articulos);  barra.add(clientes);  barra.add(proveedores);  barra.add(facturasClientes);  barra.add(facturasProveedores);  articulos.add(altaArticulo);  articulos.add(bajaArticulo);  articulos.add(modificacionArticulo);  articulos.add(consultaArticulo);  articulos.add(imprimirArticulos);  clientes.add(altaCliente);  clientes.add(bajaCliente);  clientes.add(modificacionCliente);  clientes.add(consultaCliente);  clientes.add(imprimirClientes);  proveedores.add(altaProveedor);  proveedores.add(bajaProveedor);  proveedores.add(modificacionProveedor);  proveedores.add(consultaProveedor);  proveedores.add(imprimirProveedores);  facturasClientes.add(altaFacturaCliente);  facturasClientes.add(bajaFacturaCliente);  facturasClientes.add(modificacionFacturaCliente);  facturasClientes.add(consultaFacturaCliente);  facturasClientes.add(imprimirFacturasCliente);  facturasProveedores.add(altaFacturaProveedor);  facturasProveedores.add(bajaFacturaProveedor);  facturasProveedores.add(modificacionFacturaProveedor);  facturasProveedores.add(consultaFacturaProveedor);  facturasProveedores.add(imprimirFacturasProveedor);  dialogInformacion.setBackground(new Color (255, 97, 3));  dialogInformacion.setForeground(Color.white);  dialogInformacion.setFont(fuenteLabel);  //escuchadores  addWindowListener(this);  dialogInformacion.addWindowListener(this);  altaArticulo.addActionListener(this);  bajaArticulo.addActionListener(this);  modificacionArticulo.addActionListener(this);  consultaArticulo.addActionListener(this);  imprimirArticulos.addActionListener(this);  altaCliente.addActionListener(this);  bajaCliente.addActionListener(this);  modificacionCliente.addActionListener(this);  consultaCliente.addActionListener(this);  imprimirClientes.addActionListener(this);  altaProveedor.addActionListener(this);  bajaProveedor.addActionListener(this);  modificacionProveedor.addActionListener(this);  consultaProveedor.addActionListener(this);  imprimirProveedores.addActionListener(this);  altaFacturaCliente.addActionListener(this);  bajaFacturaCliente.addActionListener(this);  modificacionFacturaCliente.addActionListener(this);  consultaFacturaCliente.addActionListener(this);  imprimirFacturasCliente.addActionListener(this);  altaFacturaProveedor.addActionListener(this);  bajaFacturaProveedor.addActionListener(this);  modificacionFacturaProveedor.addActionListener(this);  consultaFacturaProveedor.addActionListener(this);  imprimirFacturasProveedor.addActionListener(this);  //gestion cuadro de dialogo  dialogInformacion.setLayout(new FlowLayout());  dialogInformacion.setSize(200,100);  dialogInformacion.setResizable(false);  dialogInformacion.setLocation(120,120);  dialogInformacion.add(informacion);  }  public void paint(Graphics g)  {  // Dibujar la imagen  g.drawImage(imagen,200,150,this);  }  public void actionPerformed(ActionEvent ap)  {  //articulos  if(altaArticulo.equals(ap.getSource()))  {  altaArticulo alArt = new altaArticulo(null);  }  if(bajaArticulo.equals(ap.getSource()))  {  bajaArticulo baArt = new bajaArticulo(null);  }  if(modificacionArticulo.equals(ap.getSource()))  {  modificacionArticulo moArt = new modificacionArticulo(null);  }  if(consultaArticulo.equals(ap.getSource()))  {  consultaArticulo coArt = new consultaArticulo(null);  }  if(imprimirArticulos.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE ARTÍCULOS", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código", 50, 100);  pg1.drawString("Nombre", 130, 100);  pg1.drawString("Descripción", 260, 100);  pg1.drawString("Precio C.", 365, 100);  pg1.drawString("Precio V.", 435, 100);  pg1.drawString("Iva", 500, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM articulos");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idArticulo"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreArticulo"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("descripcionArticulo"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("precioCompraArticulo"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("precioVentaArticulo"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("idIvaFK");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //clientes  if(altaCliente.equals(ap.getSource()))  {  altaCliente alCli = new altaCliente(null);  }  if(bajaCliente.equals(ap.getSource()))  {  bajaCliente baCli = new bajaCliente(null);  }  if(modificacionCliente.equals(ap.getSource()))  {  modificacionCliente moCli = new modificacionCliente(null);  }  if(consultaCliente.equals(ap.getSource()))  {  consultaCliente coCli = new consultaCliente(null);  }  if(imprimirClientes.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE CLIENTES", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("CIF", 70, 100);  pg1.drawString("Nombre", 140, 100);  pg1.drawString("Dirección", 240, 100);  pg1.drawString("Ciudad", 320, 100);  pg1.drawString("C.P.", 390, 100);  pg1.drawString("Teléfono", 445, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM clientes");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("cifCliente"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreCliente"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("direccionCliente"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ciudadCliente"))+" ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("cpCliente"))+" ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("telefonoCliente");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //proveedores  if(altaProveedor.equals(ap.getSource()))  {  altaProveedor alProv = new altaProveedor(null);  }  if(bajaProveedor.equals(ap.getSource()))  {  bajaProveedor baProv = new bajaProveedor(null);  }  if(modificacionProveedor.equals(ap.getSource()))  {  modificacionProveedor moProv = new modificacionProveedor(null);  }  if(consultaProveedor.equals(ap.getSource()))  {  consultaProveedor coProv = new consultaProveedor(null);  }  if(imprimirProveedores.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE PROVEEDORES", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("CIF", 70, 100);  pg1.drawString("Nombre", 140, 100);  pg1.drawString("Dirección", 240, 100);  pg1.drawString("Ciudad", 320, 100);  pg1.drawString("C.P.", 390, 100);  pg1.drawString("Teléfono", 445, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM proveedores");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("cifProveedor"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreProveedor"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("direccionProveedor"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ciudadProveedor"))+" ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("cpProveedor"))+" ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("telefonoProveedor");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //facturas clientes  if(altaFacturaCliente.equals(ap.getSource()))  {  altaFacturaCliente alFactCli = new altaFacturaCliente(null);  }  if(bajaFacturaCliente.equals(ap.getSource()))  {  bajaFacturaCliente baFactCli = new bajaFacturaCliente(null);  }  if(modificacionFacturaCliente.equals(ap.getSource()))  {  modificacionFacturaCliente moFactCli = new modificacionFacturaCliente(null);  }  if(consultaFacturaCliente.equals(ap.getSource()))  {  consultaFacturaCliente coFactCli = new consultaFacturaCliente();  }  if(imprimirFacturasCliente.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg = pjob.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg.drawString("RESUMEN FACTURAS DE CLIENTES", 150, 50);  pg.setFont(new Font("Arial",Font.BOLD,12));  pg.drawString("Código", 80, 100);  pg.drawString("Fecha", 155, 100);  pg.drawString("Base", 230, 100);  pg.drawString("IVA", 290, 100);  pg.drawString("Total", 340, 100);  pg.drawString("CIF", 420, 100);  pg.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_clientes");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaCliente"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaCliente"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaCliente"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaCliente"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaCliente"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifClienteFK");  pg.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //facturas proveedores  if(altaFacturaProveedor.equals(ap.getSource()))  {  altaFacturaProveedor alFactProv = new altaFacturaProveedor(null);  }  if(bajaFacturaProveedor.equals(ap.getSource()))  {  bajaFacturaProveedor baFactProv = new bajaFacturaProveedor(null);  }  if(modificacionFacturaProveedor.equals(ap.getSource()))  {  modificacionFacturaProveedor moFactProv = new modificacionFacturaProveedor(null);  }  if(consultaFacturaProveedor.equals(ap.getSource()))  {  consultaFacturaProveedor coFactProv = new consultaFacturaProveedor();  }  if(imprimirFacturasProveedor.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN FACTURAS DE PROVEEDORES", 130, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código", 80, 100);  pg1.drawString("Fecha", 160, 100);  pg1.drawString("Base", 240, 100);  pg1.drawString("IVA", 310, 100);  pg1.drawString("Total", 360, 100);  pg1.drawString("CIF", 435, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_proveedores");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaProveedor"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaProveedor"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaProveedor"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaProveedor"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaProveedor"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifProveedorFK");  pg1.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  panelAcceso acceso = new panelAcceso();  this.dispose();  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **panelAdministrador** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.Frame;  import java.awt.Graphics;  import java.awt.Image;  import java.awt.Label;  import java.awt.Menu;  import java.awt.MenuBar;  import java.awt.MenuItem;  import java.awt.PrintJob;  import java.awt.Toolkit;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.File;  import java.io.IOException;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class panelAdministrador extends Frame implements WindowListener,ActionListener  {  // Declarar el objeto Image  Image imagen;  // Declarar el objeto Toolkit para manejo de imágenes  Toolkit herramienta;  //label  Label informacion = new Label("Seleccione alguna opción !!");  //fuentes  Font fuenteGeneral = new Font("Arial", Font.PLAIN, 12);  Font fuenteLabel = new Font("Arial", Font.BOLD, 12);  //menu  MenuBar barra = new MenuBar();  Menu articulos = new Menu("Artículos");  Menu clientes = new Menu("Clientes");  Menu proveedores = new Menu("Proveedores");  Menu facturasClientes = new Menu("Facturas de Clientes");  Menu facturasProveedores = new Menu("Facturas de Proveedores");  Menu iva = new Menu("Tipos IVA");  MenuItem altaArticulo = new MenuItem ("Alta de Artículo");  MenuItem bajaArticulo = new MenuItem ("Baja de Artículo");  MenuItem modificacionArticulo = new MenuItem ("Modificación de Artículo");  MenuItem consultaArticulo = new MenuItem ("Consulta de Artículo");  MenuItem imprimirArticulos = new MenuItem ("Imprimir Artículos");  MenuItem altaCliente = new MenuItem ("Alta de Cliente");  MenuItem bajaCliente = new MenuItem ("Baja de Cliente");  MenuItem modificacionCliente = new MenuItem ("Modificación de Cliente");  MenuItem consultaCliente = new MenuItem ("Consulta de Cliente");  MenuItem imprimirClientes = new MenuItem ("Imprimir Clientes");  MenuItem altaProveedor = new MenuItem ("Alta de Proveedor");  MenuItem bajaProveedor = new MenuItem ("Baja de Proveedor");  MenuItem modificacionProveedor = new MenuItem ("Modificación de Proveedor");  MenuItem consultaProveedor = new MenuItem ("Consulta de Proveedor");  MenuItem imprimirProveedores = new MenuItem ("Imprimir Proveedores");  MenuItem altaFacturaCliente = new MenuItem ("Alta de Factura Cliente");  MenuItem bajaFacturaCliente = new MenuItem ("Baja de Factura Cliente");  MenuItem modificacionFacturaCliente = new MenuItem ("Modificación de Factura Cliente");  MenuItem consultaFacturaCliente = new MenuItem ("Consulta de Factura Cliente");  MenuItem imprimirFacturasCliente = new MenuItem ("Imprimir Facturas de Clientes");  MenuItem altaFacturaProveedor = new MenuItem ("Alta de Factura Proveedor");  MenuItem bajaFacturaProveedor = new MenuItem ("Baja de Factura Proveedor");  MenuItem modificacionFacturaProveedor = new MenuItem ("Modificación de Factura Proveedor");  MenuItem consultaFacturaProveedor = new MenuItem ("Consulta de Factura Proveedor");  MenuItem imprimirFacturasProveedor = new MenuItem ("Imprimir Facturas de Proveedores");  MenuItem altaIva = new MenuItem ("Alta de Iva");  MenuItem bajaIva = new MenuItem ("Baja de Iva");  MenuItem modificacionIva = new MenuItem ("Modificación de Iva");  MenuItem consultaIva = new MenuItem ("Consulta de Iva");  MenuItem imprimirIVA = new MenuItem ("Imprimir Iva");  //ventana de dialogo  Dialog dialogInformacion = new Dialog (this,"Información",true);  public panelAdministrador()//constructor  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  // Establecer el método de trabajo con imágenes  herramienta = getToolkit();  // Especificar la ruta de la imagen  imagen = herramienta.getImage("img/hcAdministrador.jpg");  setTitle("PANEL DE ADMINISTRADOR");  setVisible(true);  setSize(900,800);//horizontal,vertical  setResizable(false);  setLocation(120,120);  //añadir menu  setMenuBar(barra);  barra.add(articulos);  barra.add(clientes);  barra.add(proveedores);  barra.add(facturasClientes);  barra.add(facturasProveedores);  barra.add(iva);  articulos.add(altaArticulo);  articulos.add(bajaArticulo);  articulos.add(modificacionArticulo);  articulos.add(consultaArticulo);  articulos.add(imprimirArticulos);  clientes.add(altaCliente);  clientes.add(bajaCliente);  clientes.add(modificacionCliente);  clientes.add(consultaCliente);  clientes.add(imprimirClientes);  proveedores.add(altaProveedor);  proveedores.add(bajaProveedor);  proveedores.add(modificacionProveedor);  proveedores.add(consultaProveedor);  proveedores.add(imprimirProveedores);  facturasClientes.add(altaFacturaCliente);  facturasClientes.add(bajaFacturaCliente);  facturasClientes.add(modificacionFacturaCliente);  facturasClientes.add(consultaFacturaCliente);  facturasClientes.add(imprimirFacturasCliente);  facturasProveedores.add(altaFacturaProveedor);  facturasProveedores.add(bajaFacturaProveedor);  facturasProveedores.add(modificacionFacturaProveedor);  facturasProveedores.add(consultaFacturaProveedor);  facturasProveedores.add(imprimirFacturasProveedor);  iva.add(altaIva);  iva.add(bajaIva);  iva.add(modificacionIva);  iva.add(consultaIva);  iva.add(imprimirIVA);  dialogInformacion.setBackground(new Color (255, 97, 3));  dialogInformacion.setForeground(Color.white);  dialogInformacion.setFont(fuenteLabel);  //escuchadores  addWindowListener(this);  dialogInformacion.addWindowListener(this);  altaArticulo.addActionListener(this);  bajaArticulo.addActionListener(this);  modificacionArticulo.addActionListener(this);  consultaArticulo.addActionListener(this);  imprimirArticulos.addActionListener(this);  altaCliente.addActionListener(this);  bajaCliente.addActionListener(this);  modificacionCliente.addActionListener(this);  consultaCliente.addActionListener(this);  imprimirClientes.addActionListener(this);  altaProveedor.addActionListener(this);  bajaProveedor.addActionListener(this);  modificacionProveedor.addActionListener(this);  consultaProveedor.addActionListener(this);  imprimirProveedores.addActionListener(this);  altaIva.addActionListener(this);  bajaIva.addActionListener(this);  modificacionIva.addActionListener(this);  consultaIva.addActionListener(this);  imprimirIVA.addActionListener(this);  altaFacturaCliente.addActionListener(this);  bajaFacturaCliente.addActionListener(this);  modificacionFacturaCliente.addActionListener(this);  consultaFacturaCliente.addActionListener(this);  imprimirFacturasCliente.addActionListener(this);  altaFacturaProveedor.addActionListener(this);  bajaFacturaProveedor.addActionListener(this);  modificacionFacturaProveedor.addActionListener(this);  consultaFacturaProveedor.addActionListener(this);  imprimirFacturasProveedor.addActionListener(this);  //gestion cuadro de dialogo  dialogInformacion.setLayout(new FlowLayout());  dialogInformacion.setSize(200,100);  dialogInformacion.setResizable(false);  dialogInformacion.setLocation(120,120);  dialogInformacion.add(informacion);  }  public void paint(Graphics g)  {  // Dibujar la imagen  g.drawImage(imagen,200,150,this);  }  public void actionPerformed(ActionEvent ap)  {  //articulos  if(altaArticulo.equals(ap.getSource()))  {  altaArticulo alArt = new altaArticulo(null);  }  if(bajaArticulo.equals(ap.getSource()))  {  bajaArticulo baArt = new bajaArticulo(null);  }  if(modificacionArticulo.equals(ap.getSource()))  {  modificacionArticulo moArt = new modificacionArticulo(null);  }  if(consultaArticulo.equals(ap.getSource()))  {  consultaArticulo coArt = new consultaArticulo(null);  }  if(imprimirArticulos.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE ARTÍCULOS", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código", 50, 100);  pg1.drawString("Nombre", 130, 100);  pg1.drawString("Descripción", 260, 100);  pg1.drawString("Precio C.", 365, 100);  pg1.drawString("Precio V.", 435, 100);  pg1.drawString("Iva", 500, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM articulos");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idArticulo"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreArticulo"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("descripcionArticulo"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("precioCompraArticulo"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("precioVentaArticulo"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("idIvaFK");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //clientes  if(altaCliente.equals(ap.getSource()))  {  altaCliente alCli = new altaCliente(null);  }  if(bajaCliente.equals(ap.getSource()))  {  bajaCliente baCli = new bajaCliente(null);  }  if(modificacionCliente.equals(ap.getSource()))  {  modificacionCliente moCli = new modificacionCliente(null);  }  if(consultaCliente.equals(ap.getSource()))  {  consultaCliente coCli = new consultaCliente(null);  }  if(imprimirClientes.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE CLIENTES", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("CIF", 70, 100);  pg1.drawString("Nombre", 140, 100);  pg1.drawString("Dirección", 240, 100);  pg1.drawString("Ciudad", 320, 100);  pg1.drawString("C.P.", 390, 100);  pg1.drawString("Teléfono", 445, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM clientes");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("cifCliente"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreCliente"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("direccionCliente"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ciudadCliente"))+" ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("cpCliente"))+" ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("telefonoCliente");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //proveedores  if(altaProveedor.equals(ap.getSource()))  {  altaProveedor alProv = new altaProveedor(null);  }  if(bajaProveedor.equals(ap.getSource()))  {  bajaProveedor baProv = new bajaProveedor(null);  }  if(modificacionProveedor.equals(ap.getSource()))  {  modificacionProveedor moProv = new modificacionProveedor(null);  }  if(consultaProveedor.equals(ap.getSource()))  {  consultaProveedor coProv = new consultaProveedor(null);  }  if(imprimirProveedores.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE PROVEEDORES", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("CIF", 70, 100);  pg1.drawString("Nombre", 140, 100);  pg1.drawString("Dirección", 240, 100);  pg1.drawString("Ciudad", 320, 100);  pg1.drawString("C.P.", 390, 100);  pg1.drawString("Teléfono", 445, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM proveedores");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("cifProveedor"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("nombreProveedor"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("direccionProveedor"))+" ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ciudadProveedor"))+" ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("cpProveedor"))+" ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("telefonoProveedor");  pg1.drawString(cadena, 50, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }      //iva  if(altaIva.equals(ap.getSource()))  {  altaIva alIva = new altaIva(null);  }  if(bajaIva.equals(ap.getSource()))  {  bajaIva baIva = new bajaIva(null);  }  if(modificacionIva.equals(ap.getSource()))  {  modificacionIva moIva = new modificacionIva(null);  }  if(consultaIva.equals(ap.getSource()))  {  consultaIva coIva = new consultaIva(null);  }  if(imprimirIVA.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN DE IVA", 200, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código", 70, 100);  pg1.drawString("Descripcion", 140, 100);  pg1.drawString("Cantidad", 240, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM tipos\_iva");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idIva"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("descripcionIva"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("cantidadIva"))+" ";  pg1.drawString(cadena, 80, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  //facturas clientes  if(altaFacturaCliente.equals(ap.getSource()))  {  altaFacturaCliente alFactCli = new altaFacturaCliente(null);  }  if(bajaFacturaCliente.equals(ap.getSource()))  {  bajaFacturaCliente baFactCli = new bajaFacturaCliente(null);  }  if(modificacionFacturaCliente.equals(ap.getSource()))  {  modificacionFacturaCliente moFactCli = new modificacionFacturaCliente(null);  }  if(consultaFacturaCliente.equals(ap.getSource()))  {  consultaFacturaCliente coFactCli = new consultaFacturaCliente();  }  if(imprimirFacturasCliente.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg = pjob.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg.drawString("RESUMEN FACTURAS DE CLIENTES", 150, 50);  pg.setFont(new Font("Arial",Font.BOLD,12));  pg.drawString("Código", 80, 100);  pg.drawString("Fecha", 155, 100);  pg.drawString("Base", 230, 100);  pg.drawString("IVA", 290, 100);  pg.drawString("Total", 340, 100);  pg.drawString("CIF", 420, 100);  pg.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_clientes");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaCliente"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaCliente"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaCliente"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaCliente"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaCliente"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifClienteFK");  pg.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }    //facturas proveedores  if(altaFacturaProveedor.equals(ap.getSource()))  {  altaFacturaProveedor alFactProv = new altaFacturaProveedor(null);  }  if(bajaFacturaProveedor.equals(ap.getSource()))  {  bajaFacturaProveedor baFactProv = new bajaFacturaProveedor(null);  }  if(modificacionFacturaProveedor.equals(ap.getSource()))  {  modificacionFacturaProveedor moFactProv = new modificacionFacturaProveedor(null);  }  if(consultaFacturaProveedor.equals(ap.getSource()))  {  consultaFacturaProveedor coFactProv = new consultaFacturaProveedor();  }  if(imprimirFacturasProveedor.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  pg1.setFont(new Font("Arial",Font.BOLD,16));  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.drawString("RESUMEN FACTURAS DE PROVEEDORES", 130, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código", 80, 100);  pg1.drawString("Fecha", 160, 100);  pg1.drawString("Base", 240, 100);  pg1.drawString("IVA", 310, 100);  pg1.drawString("Total", 360, 100);  pg1.drawString("CIF", 435, 100);  pg1.setFont(new Font("Arial",Font.PLAIN,12));  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_proveedores");  //Recorrer los registros obtenidos  int contador=125;  while(rs.next())  {  String cadena="";  // Obtener el campo cif del registro activo  cadena+=(rs.getString("idFacturaProveedor"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("fechaFacturaProveedor"))+" ";  // Obtener el campo direccion del registro activo  cadena+=(rs.getString("baseImponibleFacturaProveedor"))+"€ ";  // Obtener el campo ciudad del registro activo  cadena+=(rs.getString("ivaFacturaProveedor"))+"€ ";  // Obtener el campo cp del registro activo  cadena+=(rs.getString("totalFacturaProveedor"))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("cifProveedorFK");  pg1.drawString(cadena, 75, contador);  contador+=20;  }  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  panelAcceso acceso = new panelAcceso();  this.dispose();  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaArticulo** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaArticulo extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE ARTÍCULO");  Label LbIdArticulo = new Label("Código del Artículo");  Label LbNombreArticulo= new Label("Nombre del Artículo");  Label LbDescripArticulo = new Label("Descripción del Artículo");  Label LbPrecCompArticulo = new Label("Precio de Compra");  Label LbPrecVentArticulo = new Label("Precio de Venta");  Label LbIvaArticulo = new Label("IVA del Artículo");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtIdArticulo = new TextField(20);  TextField txtNombreArticulo = new TextField(20);  TextField txtDescripArticulo = new TextField(30);  TextField txtPrecCompArticulo = new TextField(20);  TextField txtPrecVentArticulo = new TextField(20);  //choice  Choice chIva = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdArticulo = new String ();  String NombreArticulo= new String ();  String DescripcionArticulo = new String ();  float PrecioCompraArticulo;  float PrecioVentaArticulo;  String ivaNumero = new String();  String IdIVAFK = new String ();  public altaArticulo(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdArticulo.setFont(fuenteLabelNegrita);  LbNombreArticulo.setFont(fuenteLabelNegrita);  LbDescripArticulo.setFont(fuenteLabelNegrita);  LbPrecCompArticulo.setFont(fuenteLabelNegrita);  LbPrecVentArticulo.setFont(fuenteLabelNegrita);  LbIvaArticulo.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE ARTÍCULO");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIdArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(txtIdArticulo,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbNombreArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtNombreArticulo,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbDescripArticulo,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=3;  add(txtDescripArticulo,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbPrecCompArticulo,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=4;  add(txtPrecCompArticulo,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbPrecVentArticulo,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=5;  add(txtPrecVentArticulo,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=6;  add(LbIvaArticulo,gbc);  //situacion chiice iva  gbc.gridx=1;  gbc.gridy=6;  add(chIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=7;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=1;  gbc.gridy=7;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el iva  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CantidadIVA FROM tipos\_iva");  //Recorrer los registros obtenidos  while(rs2.next())  {  float dato=rs2.getFloat("CantidadIVA");  String dato1 = new String (Float.toString(dato));  chIva.addItem(dato1);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  IdArticulo = (txtIdArticulo.getText());  NombreArticulo= (txtNombreArticulo.getText());  DescripcionArticulo = (txtDescripArticulo.getText());  if((txtPrecCompArticulo.getText().isEmpty()))  {  }  else  {  PrecioCompraArticulo = Float.parseFloat(txtPrecCompArticulo.getText());  }  if((txtPrecVentArticulo.getText().isEmpty()))  {  }  else  {  PrecioVentaArticulo = Float.parseFloat(txtPrecVentArticulo.getText());  }  ivaNumero = (chIva.getSelectedItem());  if (ivaNumero.equals("8.0"))  {  IdIVAFK="iva8";  }  if (ivaNumero.equals("10.0"))  {  IdIVAFK="iva10";  }  if (ivaNumero.equals("21.0"))  {  IdIVAFK="iva21";  }  if (ivaNumero.equals("25.0"))  {  IdIVAFK="iva25";  }  if (ivaNumero.equals("30.0"))  {  IdIVAFK="iva30";  }  if ((IdArticulo.length()>10)||(txtIdArticulo.getText().isEmpty())||(NombreArticulo.length()>45)||(txtNombreArticulo.getText().isEmpty())||(DescripcionArticulo.length()>100)||(txtDescripArticulo.getText().isEmpty())||(PrecioCompraArticulo>99999999.99)||(PrecioCompraArticulo<0)||(txtPrecCompArticulo.getText().isEmpty())||(PrecioVentaArticulo>99999999.99)||(PrecioVentaArticulo<0)||(txtPrecVentArticulo.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO articulos (IdArticulo,NombreArticulo,DescripcionArticulo,PrecioCompraArticulo,PrecioVentaArticulo,IdIVAFK) VALUES ('"+IdArticulo+"','"+NombreArticulo+"','"+DescripcionArticulo+"','"+PrecioCompraArticulo+"','"+PrecioVentaArticulo+"','"+IdIVAFK+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE CLIENTE");  Label LbCifCliente = new Label("CIF del Cliente");  Label LbNombreCliente = new Label("Nombre del Cliente");  Label LbDireccionCliente = new Label("Dirección del Cliente");  Label LbCiudadCliente = new Label("Ciudad del Cliente");  Label LbCpCliente = new Label("CP del Cliente");  Label LbTelefonoCliente = new Label("Teléfono del Cliente");  Label hClinicNom = new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf = new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtCifCliente = new TextField(20);  TextField txtNombreCliente = new TextField(20);  TextField txtDireccionCliente = new TextField(30);  TextField txtCiudadCliente = new TextField(20);  TextField txtCpCliente = new TextField(20);  TextField txtTelefonoCliente = new TextField(20);  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String CifCliente = new String ();  String NombreCliente = new String ();  String DireccionCliente = new String ();  String CiudadCliente = new String ();  int CpCliente;  int TelefonoCliente;  public altaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCifCliente.setFont(fuenteLabelNegrita);  LbNombreCliente.setFont(fuenteLabelNegrita);  LbDireccionCliente.setFont(fuenteLabelNegrita);  LbCiudadCliente.setFont(fuenteLabelNegrita);  LbCpCliente.setFont(fuenteLabelNegrita);  LbTelefonoCliente.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label cif  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbCifCliente,gbc);  //situacion txt cif  gbc.gridx=1;  gbc.gridy=1;  add(txtCifCliente,gbc);  //situacion label nombre  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbNombreCliente,gbc);  //situacion txt nombre  gbc.gridx=1;  gbc.gridy=2;  add(txtNombreCliente,gbc);  //situacion label direccion  gbc.gridx=0;  gbc.gridy=3;  add(LbDireccionCliente,gbc);  //situacion txt direccion  gbc.gridx=1;  gbc.gridy=3;  add(txtDireccionCliente,gbc);  //situacion label ciudad  gbc.gridx=0;  gbc.gridy=4;  add(LbCiudadCliente,gbc);  //situacion txt ciudad  gbc.gridx=1;  gbc.gridy=4;  add(txtCiudadCliente,gbc);  //situacion label cp  gbc.gridx=0;  gbc.gridy=5;  add(LbCpCliente,gbc);  //situacion txt cp  gbc.gridx=1;  gbc.gridy=5;  add(txtCpCliente,gbc);  //situacion labeltf  gbc.gridx=0;  gbc.gridy=6;  add(LbTelefonoCliente,gbc);  //situacion txt tf  gbc.gridx=1;  gbc.gridy=6;  add(txtTelefonoCliente,gbc);  //botoneS  gbc.gridx=0;  gbc.gridy=7;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=1;  gbc.gridy=7;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  CifCliente = (txtCifCliente.getText());  NombreCliente = (txtNombreCliente.getText());  DireccionCliente = (txtDireccionCliente.getText());  CiudadCliente = (txtCiudadCliente.getText());  if((txtCpCliente.getText().isEmpty()))  {  }  else  {  CpCliente = Integer.parseInt(txtCpCliente.getText());  }  if((txtTelefonoCliente.getText().isEmpty()))  {  }  else  {  TelefonoCliente = Integer.parseInt(txtTelefonoCliente.getText());  }    if ((CifCliente.length()>9)||(txtCifCliente.getText().isEmpty())||(NombreCliente.length()>45)||(txtNombreCliente.getText().isEmpty())||(DireccionCliente.length()>45)||(txtDireccionCliente.getText().isEmpty())||(CiudadCliente.length()>45)||(txtCiudadCliente.getText().isEmpty())||(CpCliente>99999)||(CpCliente<0)||(txtCpCliente.getText().isEmpty())||(TelefonoCliente>999999999)||(TelefonoCliente<0)||(txtTelefonoCliente.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO clientes (CifCliente,NombreCliente,DireccionCliente,CiudadCliente,CpCliente,TelefonoCliente) VALUES ('"+CifCliente+"','"+NombreCliente+"','"+DireccionCliente+"','"+CiudadCliente+"','"+CpCliente+"','"+TelefonoCliente+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaFacturaCliente extends Dialog implements WindowListener,ActionListener{  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE FACTURA DE CLIENTE");  //Label LbIdFacturaCliente = new Label("Código Factura de Cliente");  Label LbFechaFacturaCliente = new Label("Fecha Factura de Cliente (DD-MM-AAAA)");  //Label LbBaseImponibleFacturaCliente = new Label("Base Imponible Factura de Cliente");  //Label LbIvaFactutaCliente = new Label("IVA Factura de Cliente");  //Label LbTotalFacturaCliente = new Label("Total Factura de Cliente");  Label LbCifClienteFK = new Label("CIF del Cliente");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button nuevaLinea = new Button ("Insertar Línea");  Button cerrar = new Button ("Cerrar");  //texfield  //TextField txtIdFacturaCliente = new TextField(20);  TextField txtFechaFacturaCliente = new TextField(20);  //TextField txtBaseImponibleFacturaCliente = new TextField(20);  //TextField txtIvaFactutaCliente = new TextField(20);  //TextField txtTotalFacturaCliente = new TextField(20);  //choice  Choice chCifClienteFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  int IdFacturaCliente=0;  String FechaFacturaCliente = new String ();  //float BaseImponibleFacturaCliente;  //float IvaFactutaCliente;  //float TotalFacturaCliente;  String CifClienteFK = new String();  public altaFacturaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  //LbIdFacturaCliente.setFont(fuenteLabelNegrita);  LbFechaFacturaCliente.setFont(fuenteLabelNegrita);  //LbBaseImponibleFacturaCliente.setFont(fuenteLabelNegrita);  //LbIvaFactutaCliente.setFont(fuenteLabelNegrita);  //LbTotalFacturaCliente.setFont(fuenteLabelNegrita);  LbCifClienteFK.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE DE FACTURA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(600, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label id factura cliente  //gbc.gridx=0;  //gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  //add(LbIdFacturaCliente,gbc);  //situacion txt id factura cliente  //gbc.gridx=1;  //gridy=1;  // add(txtIdFacturaCliente,gbc);  //situacion label fecha factura  gbc.gridx=0;  gbc.gridy=1;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbFechaFacturaCliente,gbc);  //situacion txt fecha factura  gbc.gridx=1;  gbc.gridy=1;  add(txtFechaFacturaCliente,gbc);  //situacion label base imponible  //gbc.gridx=0;  //gbc.gridy=3;  //add(LbBaseImponibleFacturaCliente,gbc);  //situacion txt base imponible  //gbc.gridx=1;  //gbc.gridy=3;  //add(txtBaseImponibleFacturaCliente,gbc);  //situacion label iva factura  //gbc.gridx=0;  //gbc.gridy=4;  //add(LbIvaFactutaCliente,gbc);  //situacion txt iva factura  //gbc.gridx=1;  //gbc.gridy=4;  //add(txtIvaFactutaCliente,gbc);  //situacion label total factura  //gbc.gridx=0;  //gbc.gridy=5;  //add(LbTotalFacturaCliente,gbc);  //situacion txt total factura  //gbc.gridx=1;  //gbc.gridy=5;  //add(txtTotalFacturaCliente,gbc);  //situacion lable cif cliente  gbc.gridx=0;  gbc.gridy=3;  add(LbCifClienteFK,gbc);  //situacion txt cif cliente  gbc.gridx=1;  gbc.gridy=3;  add(chCifClienteFK,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=0;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.FIRST\_LINE\_END;  add(cerrar,gbc);  gbc.gridx=1;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.CENTER;  add(nuevaLinea,gbc);  nuevaLinea.setEnabled(false);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  nuevaLinea.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el cif  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CifCliente FROM clientes");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("CifCliente");  chCifClienteFK.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    //select ultimo codigo de factura  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT idFacturaCliente FROM facturas\_clientes ORDER BY idfacturaCliente DESC LIMIT 1;");  //Recorrer los registros obtenidos  while(rs2.next())  {  IdFacturaCliente=Integer.parseInt(rs2.getString("IdFacturaCliente"));  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  IdFacturaCliente++;  FechaFacturaCliente = (txtFechaFacturaCliente.getText()).substring(6,10)+"-"+(txtFechaFacturaCliente.getText()).substring(3,5)+"-"+(txtFechaFacturaCliente.getText()).substring(0,2);  /\*if((txtBaseImponibleFacturaCliente.getText().isEmpty()))  {  }  else  {  BaseImponibleFacturaCliente = Float.parseFloat(txtBaseImponibleFacturaCliente.getText());\*/  }  /\*if((txtIvaFactutaCliente.getText().isEmpty()))  {  }  else  {  IvaFactutaCliente = Float.parseFloat(txtIvaFactutaCliente.getText());  }  if((txtTotalFacturaCliente.getText().isEmpty()))  {  }  else  {  TotalFacturaCliente = Float.parseFloat(txtTotalFacturaCliente.getText());  }\*/  CifClienteFK=chCifClienteFK.getSelectedItem();  if ((FechaFacturaCliente.length()>10)||(FechaFacturaCliente.isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotel\_clinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO facturas\_clientes (IdFacturaCliente,FechaFacturaCliente,CifClienteFK) VALUES ('"+Integer.toString(IdFacturaCliente)+"','"+FechaFacturaCliente+"','"+CifClienteFK+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  nuevaLinea.setEnabled(true);    if(nuevaLinea.equals(ap.getSource()))  {  lineaFacturaCliente lFCli = new lineaFacturaCliente(null);  }  }    public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaFacturaProveedor extends Dialog implements WindowListener,ActionListener{  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE FACTURA DE PROVEEDOR");  //Label LbIdFacturaProveedor = new Label("Código Factura de Proveedor");  Label LbFechaFacturaProveedor = new Label("Fecha Factura de Proveedor (DD-MM-AAAA)");  //Label LbBaseImponibleFacturaProveedor = new Label("Base Imponible Factura de Proveedor");  //Label LbIvaFactutaProveedor = new Label("IVA Factura de Proveedor");  //Label LbTotalFacturaProveedor = new Label("Total Factura de Proveedor");  Label LbCifProveedorFK = new Label("CIF del Proveedor");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button nuevaLinea = new Button ("Insertar Línea");  Button cerrar = new Button ("Cerrar");  //texfield  //TextField txtIdFacturaProveedor = new TextField(20);  TextField txtFechaFacturaProveedor = new TextField(20);  //TextField txtBaseImponibleFacturaProveedor = new TextField(20);  //TextField txtIvaFactutaProveedor = new TextField(20);  //TextField txtTotalFacturaProveedor = new TextField(20);  //choice  Choice chCifProveedorFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  int IdFacturaProveedor=0;  String FechaFacturaProveedor = new String ();  //float BaseImponibleFacturaProveedor;  //float IvaFactutaProveedor;  //float TotalFacturaProveedor;  String CifProveedorFK = new String();  public altaFacturaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  //LbIdFacturaProveedor.setFont(fuenteLabelNegrita);  LbFechaFacturaProveedor.setFont(fuenteLabelNegrita);  //LbBaseImponibleFacturaProveedor.setFont(fuenteLabelNegrita);  //LbIvaFactutaProveedor.setFont(fuenteLabelNegrita);  //LbTotalFacturaProveedor.setFont(fuenteLabelNegrita);  LbCifProveedorFK.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE DE FACTURA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(675, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label id factura Proveedor  //gbc.gridx=0;  //gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  //add(LbIdFacturaProveedor,gbc);  //situacion txt id factura Proveedor  //gbc.gridx=1;  //gridy=1;  // add(txtIdFacturaProveedor,gbc);  //situacion label fecha factura  gbc.gridx=0;  gbc.gridy=1;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbFechaFacturaProveedor,gbc);  //situacion txt fecha factura  gbc.gridx=1;  gbc.gridy=1;  add(txtFechaFacturaProveedor,gbc);  //situacion label base imponible  //gbc.gridx=0;  //gbc.gridy=3;  //add(LbBaseImponibleFacturaProveedor,gbc);  //situacion txt base imponible  //gbc.gridx=1;  //gbc.gridy=3;  //add(txtBaseImponibleFacturaProveedor,gbc);  //situacion label iva factura  //gbc.gridx=0;  //gbc.gridy=4;  //add(LbIvaFactutaProveedor,gbc);  //situacion txt iva factura  //gbc.gridx=1;  //gbc.gridy=4;  //add(txtIvaFactutaProveedor,gbc);  //situacion label total factura  //gbc.gridx=0;  //gbc.gridy=5;  //add(LbTotalFacturaProveedor,gbc);  //situacion txt total factura  //gbc.gridx=1;  //gbc.gridy=5;  //add(txtTotalFacturaProveedor,gbc);  //situacion lable cif Proveedor  gbc.gridx=0;  gbc.gridy=3;  add(LbCifProveedorFK,gbc);  //situacion txt cif Proveedor  gbc.gridx=1;  gbc.gridy=3;  add(chCifProveedorFK,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=0;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.FIRST\_LINE\_END;  add(cerrar,gbc);  gbc.gridx=1;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.CENTER;  add(nuevaLinea,gbc);  nuevaLinea.setEnabled(false);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  nuevaLinea.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el cif  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CifProveedor FROM Proveedores");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("CifProveedor");  chCifProveedorFK.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    //select ultimo codigo de factura  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT idFacturaProveedor FROM facturas\_Proveedores ORDER BY idfacturaProveedor DESC LIMIT 1;");  //Recorrer los registros obtenidos  while(rs2.next())  {  IdFacturaProveedor=Integer.parseInt(rs2.getString("IdFacturaProveedor"));  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  IdFacturaProveedor++;  FechaFacturaProveedor = (txtFechaFacturaProveedor.getText()).substring(6,10)+"-"+(txtFechaFacturaProveedor.getText()).substring(3,5)+"-"+(txtFechaFacturaProveedor.getText()).substring(0,2);  /\*if((txtBaseImponibleFacturaProveedor.getText().isEmpty()))  {  }  else  {  BaseImponibleFacturaProveedor = Float.parseFloat(txtBaseImponibleFacturaProveedor.getText());\*/  }  /\*if((txtIvaFactutaProveedor.getText().isEmpty()))  {  }  else  {  IvaFactutaProveedor = Float.parseFloat(txtIvaFactutaProveedor.getText());  }  if((txtTotalFacturaProveedor.getText().isEmpty()))  {  }  else  {  TotalFacturaProveedor = Float.parseFloat(txtTotalFacturaProveedor.getText());  }\*/  CifProveedorFK=chCifProveedorFK.getSelectedItem();  if ((FechaFacturaProveedor.length()>10)||(FechaFacturaProveedor.isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotel\_clinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO facturas\_Proveedores (IdFacturaProveedor,FechaFacturaProveedor,CifProveedorFK) VALUES ('"+Integer.toString(IdFacturaProveedor)+"','"+FechaFacturaProveedor+"','"+CifProveedorFK+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  nuevaLinea.setEnabled(true);    if(nuevaLinea.equals(ap.getSource()))  {  lineaFacturaProveedor lFCli = new lineaFacturaProveedor(null);  }  }    public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaIva** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaIva extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE IVA");  Label LbIdIva = new Label("Código del IVA");  Label LbDescripcionIva = new Label("Descripción del IVA");  Label LbCantidadIva = new Label("Cantidad del IVA");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtIdIva = new TextField(20);  TextField txtDescripcionIva = new TextField(20);  TextField txtCantidadIva = new TextField(20);    //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdIva = new String ();  String DescripcionIva = new String ();  float CantidadIva;    public altaIva(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdIva.setFont(fuenteLabelNegrita);  LbDescripcionIva.setFont(fuenteLabelNegrita);  LbCantidadIva.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE IVA");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label idIva  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIdIva,gbc);  //situacion txt idIva  gbc.gridx=1;  gbc.gridy=1;  add(txtIdIva,gbc);  //situacion label descripcion iva  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbDescripcionIva,gbc);  //situacion txt descripcion iva  gbc.gridx=1;  gbc.gridy=2;  add(txtDescripcionIva,gbc);  //situacion label cantidad iva  gbc.gridx=0;  gbc.gridy=3;  add(LbCantidadIva,gbc);  //situacion txt cantidad iva  gbc.gridx=1;  gbc.gridy=3;  add(txtCantidadIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=1;  gbc.gridy=4;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  IdIva = (txtIdIva.getText());  DescripcionIva = (txtDescripcionIva.getText());  if((txtCantidadIva.getText().isEmpty()))  {  }  else  {  CantidadIva = Float.parseFloat(txtCantidadIva.getText());  }    if ((IdIva.length()>9)||(txtIdIva.getText().isEmpty())||(DescripcionIva.length()>45)||(txtDescripcionIva.getText().isEmpty())||(txtCantidadIva.getText().isEmpty())||(CantidadIva>99.99)||(CantidadIva<0))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO tipos\_iva (IdIva,DescripcionIva,cantidadIva) VALUES ('"+IdIva+"','"+DescripcionIva+"','"+CantidadIva+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **altaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class altaProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ALTA DE PROVEEDOR");  Label LbCifProveedor = new Label("CIF del Proveedor");  Label LbNombreProveedor= new Label("Nombre del Proveedor");  Label LbDireccionProveedor = new Label("Dirección del Proveedor");  Label LbCiudadProveedor = new Label("Ciudad del Proveedor");  Label LbCpProveedor = new Label("CP del Proveedor");  Label LbTelefonoProveedor = new Label("Teléfono del Proveedor");  Label hClinicNom = new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf = new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  //Button  Button grabar = new Button ("Grabar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtCifProveedor = new TextField(20);  TextField txtNombreProveedor = new TextField(20);  TextField txtDireccionProveedor = new TextField(30);  TextField txtCiudadProveedor = new TextField(20);  TextField txtCpProveedor = new TextField(20);  TextField txtTelefonoProveedor = new TextField(20);  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String CifProveedor = new String ();  String NombreProveedor = new String ();  String DireccionProveedor = new String ();  String CiudadProveedor = new String ();  int CpProveedor;  int TelefonoProveedor;  public altaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCifProveedor.setFont(fuenteLabelNegrita);  LbNombreProveedor.setFont(fuenteLabelNegrita);  LbDireccionProveedor.setFont(fuenteLabelNegrita);  LbCiudadProveedor.setFont(fuenteLabelNegrita);  LbCpProveedor.setFont(fuenteLabelNegrita);  LbTelefonoProveedor.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ALTA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbCifProveedor,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(txtCifProveedor,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbNombreProveedor,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtNombreProveedor,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbDireccionProveedor,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=3;  add(txtDireccionProveedor,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbCiudadProveedor,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=4;  add(txtCiudadProveedor,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbCpProveedor,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=5;  add(txtCpProveedor,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=6;  add(LbTelefonoProveedor,gbc);  //situacion chiice iva  gbc.gridx=1;  gbc.gridy=6;  add(txtTelefonoProveedor,gbc);  //botoneS  gbc.gridx=0;  gbc.gridy=7;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(grabar,gbc);  gbc.gridx=1;  gbc.gridy=7;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  grabar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(grabar.equals(ap.getSource()))  {  CifProveedor = (txtCifProveedor.getText());  NombreProveedor = (txtNombreProveedor.getText());  DireccionProveedor = (txtDireccionProveedor.getText());  CiudadProveedor = (txtCiudadProveedor.getText());  if((txtCpProveedor.getText().isEmpty()))  {  }  else  {  CpProveedor = Integer.parseInt(txtCpProveedor.getText());  }  if((txtTelefonoProveedor.getText().isEmpty()))  {  }  else  {  TelefonoProveedor = Integer.parseInt(txtTelefonoProveedor.getText());  }    if ((CifProveedor.length()>9)||(txtCifProveedor.getText().isEmpty())||(NombreProveedor.length()>45)||(txtNombreProveedor.getText().isEmpty())||(DireccionProveedor.length()>45)||(txtDireccionProveedor.getText().isEmpty())||(CiudadProveedor.length()>45)||(txtCiudadProveedor.getText().isEmpty())||(CpProveedor>99999)||(CpProveedor<0)||(txtCpProveedor.getText().isEmpty())||(TelefonoProveedor>999999999)||(TelefonoProveedor<0)||(txtTelefonoProveedor.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("INSERT INTO Proveedores (CifProveedor,NombreProveedor,DireccionProveedor,CiudadProveedor,CpProveedor,TelefonoProveedor) VALUES ('"+CifProveedor+"','"+NombreProveedor+"','"+DireccionProveedor+"','"+CiudadProveedor+"','"+CpProveedor+"','"+TelefonoProveedor+"')");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaArticulo** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaArticulo extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE ARTÍCULO");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Artículo dado de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaArticulo(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE ARTÍCULO");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("Código");  chCampo.addItem("Nombre");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("Código".equals(chCampo.getSelectedItem()))  {  campo = "IdArticulo";  }  if ("Nombre".equals(chCampo.getSelectedItem()))  {  campo = "NombreArticulo";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM articulos WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE CLIENTE");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Cliente dado de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("CIF");  chCampo.addItem("Nombre");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("CIF".equals(chCampo.getSelectedItem()))  {  campo = "CifCliente";  }  if ("Nombre".equals(chCampo.getSelectedItem()))  {  campo = "NombreCliente";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM clientes WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaFacturaCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE FACTURA DE CLIENTE");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Factura Cliente dada de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaFacturaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE FACTURA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(300,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("Código");  chCampo.addItem("Base Imponible");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("Código".equals(chCampo.getSelectedItem()))  {  campo = "IdFacturaCliente";  }  if ("Base Imponible".equals(chCampo.getSelectedItem()))  {  campo = "BaseImponibleFacturaCliente";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM facturas\_clientes WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaFacturaProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE FACTURA DE PROVEEDOR");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Factura Proveedor dada de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaFacturaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE FACTURA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(300,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("Código");  chCampo.addItem("Base Imponible");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("Código".equals(chCampo.getSelectedItem()))  {  campo = "IdFacturaProveedor";  }  if ("Base Imponible".equals(chCampo.getSelectedItem()))  {  campo = "BaseImponibleFacturaProveedor";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM facturas\_Proveedores WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaIva** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaIva extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE IVA");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("IVA dado de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaIva(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE IVA");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("Código");  chCampo.addItem("Descripción");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("Código".equals(chCampo.getSelectedItem()))  {  campo = "IdIva";  }  if ("Descripción".equals(chCampo.getSelectedItem()))  {  campo = "DescripcionIva";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM tipos\_iva WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **bajaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class bajaProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("BAJA DE PROVEEDOR");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbCampo = new Label("Campo");  Label LbIgual = new Label(" sea igual a ");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Proveedor dado de baja correctamente !!");  //Button  Button btBaja= new Button ("Dar baja");  Button btCerrar = new Button ("Cerrar");  //texfield  TextField txtCondicion = new TextField(20);  //choice  Choice chCampo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String campo = new String ();  String valor = new String ();  public bajaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("BAJA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label campo  gbc.gridx=0;  gbc.gridy=2;  add(LbCampo,gbc);  //situacion choice campo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.CENTER;  add(chCampo,gbc);  //situacion label igual  gbc.gridx=2;  gbc.gridy=2;  add(LbIgual,gbc);  //situacion txt condicion  gbc.gridx=3;  gbc.gridy=2;  add(txtCondicion,gbc);  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btBaja,gbc);  gbc.gridx=3;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=4;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btBaja.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los campos  chCampo.addItem("CIF");  chCampo.addItem("Nombre");  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btBaja.equals(ap.getSource()))  {  if ("CIF".equals(chCampo.getSelectedItem()))  {  campo = "CifProveedor";  }  if ("Nombre".equals(chCampo.getSelectedItem()))  {  campo = "NombreProveedor";  }  valor=(txtCondicion.getText());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("DELETE FROM Proveedores WHERE "+campo+"='"+valor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaArticulo** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaArticulo extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE ARTÍCULO");  Label LbCodConsulta = new Label ("Código del Artículo");  Label LbIdArticulo = new Label("Código del Artículo");  Label LbNombreArticulo= new Label("Nombre del Artículo");  Label LbDescripArticulo = new Label("Descripción del Artículo");  Label LbPrecCompArticulo = new Label("Precio de Compra");  Label LbPrecVentArticulo = new Label("Precio de Venta");  Label LbIvaArticulo = new Label("IVA del Articulo");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");    Label txtIdArticulo = new Label(" ");  Label txtNombreArticulo = new Label(" ");  Label txtDescripArticulo = new Label(" ");  Label txtPrecCompArticulo = new Label(" ");  Label txtPrecVentArticulo = new Label(" ");  Label chIva = new Label(" ");  //Button  Button cerrar = new Button ("Cerrar");  Button mostrar = new Button ("Mostrar");    //choice  Choice chIdArticulo = new Choice();    //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String IdArticulo = new String ();  String NombreArticulo= new String ();  String DescripcionArticulo = new String ();  float PrecioCompraArticulo;  float PrecioVentaArticulo;  String ivaNumero = new String();  String IdIVAFK = new String ();  public consultaArticulo(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());    //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdArticulo.setFont(fuenteLabelNegrita);  LbNombreArticulo.setFont(fuenteLabelNegrita);  LbDescripArticulo.setFont(fuenteLabelNegrita);  LbPrecCompArticulo.setFont(fuenteLabelNegrita);  LbPrecVentArticulo.setFont(fuenteLabelNegrita);  LbIvaArticulo.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE ARTÍCULO");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion choice idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chIdArticulo,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtIdArticulo,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreArticulo,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbDescripArticulo,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=4;  add(txtDescripArticulo,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbPrecCompArticulo,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=5;  add(txtPrecCompArticulo,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=6;  add(LbPrecVentArticulo,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=6;  add(txtPrecVentArticulo,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=7;  add(LbIvaArticulo,gbc);  //situacion choice iva  gbc.gridx=1;  gbc.gridy=7;  add(chIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 30, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdArticulo FROM articulos");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("idArticulo");  chIdArticulo.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=chIdArticulo.getSelectedItem();  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM articulos WHERE IdArticulo ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  IdArticulo = rs1.getString("IdArticulo");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  NombreArticulo = rs1.getString("NombreArticulo");  // Obtener el campo direccion del registro activo  DescripcionArticulo = rs1.getString("DescripcionArticulo");  // Obtener el campo PrecioCompraArticulo del registro activo  PrecioCompraArticulo = rs1.getFloat("PrecioCompraArticulo");  // Obtener el campo PrecioVentaArticulo del registro activo  PrecioVentaArticulo= rs1.getFloat("PrecioVentaArticulo");  // Obtener el campo iva del registro activo  IdIVAFK = rs1.getString("IdIVAFK");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdArticulo.setText(IdArticulo);  txtNombreArticulo.setText(NombreArticulo);  txtDescripArticulo.setText(DescripcionArticulo);  txtPrecCompArticulo.setText(Float.toString(PrecioCompraArticulo)+" €");  txtPrecVentArticulo.setText(Float.toString(PrecioVentaArticulo)+" €");  if (IdIVAFK.equals("iva10"))  {  chIva.setText("10.00 %");  }  if (IdIVAFK.equals("iva21"))  {  chIva.setText("21.00 %");  }  if (IdIVAFK.equals("iva25"))  {  chIva.setText("25.00 %");  }  if (IdIVAFK.equals("iva30"))  {  chIva.setText("30.00 %");  }  if (IdIVAFK.equals("iva8"))  {  chIva.setText("8.00 %");  }      }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE CLIENTE");  Label LbCodConsulta = new Label ("CIF del Cliente");  Label LbCifCliente = new Label("CIF del Cliente");  Label LbNombreCliente = new Label("Nombre del Cliente");  Label LbDireccionCliente = new Label("Dirección del Cliente");  Label LbCiudadCliente = new Label("Ciudad del Cliente");  Label LbCpCliente = new Label("CP del Cliente");  Label LbTelefonoCliente = new Label("Teléfono del Cliente");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  Label txtCifCliente = new Label(" ");  Label txtNombreCliente = new Label(" ");  Label txtDireccionCliente = new Label(" ");  Label txtCiudadCliente = new Label(" ");  Label txtCpCliente = new Label(" ");  Label txtTelefonoCliente = new Label(" ");  //Button  Button cerrar = new Button ("Cerrar");  Button mostrar = new Button ("Mostrar");  //choice  Choice chCifCliente = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String CifCliente = new String ();  String NombreCliente= new String ();  String DireccionCliente = new String ();  String CiudadCliente = new String ();  int CpCliente;  int TelefonoCliente ;  public consultaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbCifCliente.setFont(fuenteLabelNegrita);  LbNombreCliente.setFont(fuenteLabelNegrita);  LbDireccionCliente.setFont(fuenteLabelNegrita);  LbCiudadCliente.setFont(fuenteLabelNegrita);  LbCpCliente.setFont(fuenteLabelNegrita);  LbTelefonoCliente.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE ARTÍCULO");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chCifCliente,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbCifCliente,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtCifCliente,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreCliente,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreCliente,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbDireccionCliente,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=4;  add(txtDireccionCliente,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbCiudadCliente,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=5;  add(txtCiudadCliente,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=6;  add(LbCpCliente,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=6;  add(txtCpCliente,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=7;  add(LbTelefonoCliente,gbc);  //situacion choice iva  gbc.gridx=1;  gbc.gridy=7;  add(txtTelefonoCliente,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 30, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT cifCliente FROM clientes");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("cifCliente");  chCifCliente.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=chCifCliente.getSelectedItem();  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM clientes WHERE CifCliente ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  CifCliente = rs1.getString("CifCliente");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  NombreCliente = rs1.getString("NombreCliente");  // Obtener el campo direccion del registro activo  DireccionCliente = rs1.getString("DireccionCliente");  // Obtener el campo direccion del registro activo  CiudadCliente = rs1.getString("CiudadCliente");  // Obtener el campo PrecioCompraArticulo del registro activo  CpCliente = rs1.getInt("CpCliente");  // Obtener el campo PrecioVentaArticulo del registro activo  TelefonoCliente= rs1.getInt("TelefonoCliente");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtCifCliente.setText(CifCliente);  txtNombreCliente.setText(NombreCliente);  txtDireccionCliente.setText(DireccionCliente);  txtCiudadCliente.setText(CiudadCliente);  txtCpCliente.setText(Integer.toString(CpCliente));  txtTelefonoCliente.setText(Integer.toString(TelefonoCliente));  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.Font;  import java.awt.Frame;  import java.awt.Graphics;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.PrintJob;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaFacturaCliente extends Frame implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE FACTURA DE CLIENTE");  Label LbCodConsulta = new Label ("Código de Factura");  Label LbIdFacturaCliente = new Label("Código Factura de Cliente");  Label LbFechaFacturaCliente = new Label("Fecha Factura de Cliente");  Label LbBaseImponibleFacturaCliente = new Label("Base Imponible Factura de Cliente");  Label LbIvaFactutaCliente = new Label("IVA Factura de Cliente");  Label LbTotalFacturaCliente = new Label("Total Factura de Cliente");  Label LbCifClienteFK = new Label("CIF del Cliente");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  Label txtIdFacturaCliente = new Label(" ");  Label txtFechaFacturaCliente = new Label(" ");  Label txtBaseImponibleFacturaCliente = new Label(" ");  Label txtIvaFactutaCliente = new Label(" ");  Label txtTotalFacturaCliente = new Label(" ");  Label chCifClienteFK = new Label(" ");  //Button  Button cerrar = new Button ("Cerrar");  Button mostrar = new Button ("Mostrar");  Button imprimir= new Button ("Imprimir");  //choice  Choice chIdFacturaCliente = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  static String valor = new String ();  static String IdFacturaCliente = new String ();  static String FechaFacturaCliente= new String ();  float BaseImponibleFacturaCliente;  float IvaFactutaCliente;  float TotalFacturaCliente;  static String CifClienteFK = new String();    String nombreCliente = new String();  String direccionCliente = new String();  String ciudadCliente = new String();  String cpCliente = new String();  String telefonoCliente = new String();    float ivaProducto=0;  float ivaLinea=0;  float base=0;  float subtotal=0;      public consultaFacturaCliente()  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdFacturaCliente.setFont(fuenteLabelNegrita);  LbFechaFacturaCliente.setFont(fuenteLabelNegrita);  LbBaseImponibleFacturaCliente.setFont(fuenteLabelNegrita);  LbIvaFactutaCliente.setFont(fuenteLabelNegrita);  LbTotalFacturaCliente.setFont(fuenteLabelNegrita);  LbCifClienteFK.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE FACTURA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(625, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt codigoconsulta  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=1;  add(chIdFacturaCliente,gbc);  //situacion label id factura  gbc.gridx=0;  gbc.gridy=2;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdFacturaCliente,gbc);  //situacion txt id factura  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=2;  add(txtIdFacturaCliente,gbc);  //situacion label fecha factura  gbc.gridx=0;  gbc.gridy=3;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbFechaFacturaCliente,gbc);  //situacion txt fecha factura  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=3;  add(txtFechaFacturaCliente,gbc);  //situacion label base imponible  gbc.gridx=0;  gbc.gridy=4;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbBaseImponibleFacturaCliente,gbc);  //situacion txt base imponible  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=4;  add(txtBaseImponibleFacturaCliente,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbIvaFactutaCliente,gbc);  //situacion txt iva  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=5;  add(txtIvaFactutaCliente,gbc);  //situacion label total  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbTotalFacturaCliente,gbc);  //situacion txt total  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=6;  add(txtTotalFacturaCliente,gbc);  //situacion label ciffk  gbc.gridx=0;  gbc.gridy=7;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbCifClienteFK,gbc);  //situacion txt ciffk  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=7;  add(chCifClienteFK,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.anchor=GridBagConstraints.LINE\_START;  add(imprimir,gbc);  gbc.gridx=2;  gbc.gridy=8;  gbc.anchor=GridBagConstraints.CENTER;  add(cerrar,gbc);  imprimir.setEnabled(false);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  mostrar.addActionListener(this);  imprimir.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdFacturaCliente FROM facturas\_Clientes");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdFacturaCliente");  chIdFacturaCliente.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=chIdFacturaCliente.getSelectedItem();    //rescata datos de facturas clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_clientes WHERE IdFacturaCliente ='"+valor+"'");  while (rs.next())  {  // Obtener el campo IdArticulo del registro activo  IdFacturaCliente = rs.getString("IdFacturaCliente");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  FechaFacturaCliente = rs.getString("FechaFacturaCliente");  // Obtener el campo direccion del registro activo  BaseImponibleFacturaCliente = rs.getFloat("BaseImponibleFacturaCliente");  // Obtener el campo PrecioCompraArticulo del registro activo  IvaFactutaCliente = rs.getFloat("IvaFacturaCliente");  // Obtener el campo PrecioVentaArticulo del registro activo  TotalFacturaCliente= rs.getFloat("TotalFacturaCliente");  // Obtener el campo iva del registro activo  CifClienteFK = rs.getString("CifClienteFK");  }  //Cerrar los objetos creados  rs.close();//fue el ultimo en abrirse  sen.close();  con.close();  //detalle.setEnabled(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdFacturaCliente.setText(IdFacturaCliente);  txtFechaFacturaCliente.setText(FechaFacturaCliente);  txtBaseImponibleFacturaCliente.setText(Float.toString(BaseImponibleFacturaCliente)+" €");  txtIvaFactutaCliente.setText(Float.toString(IvaFactutaCliente)+" €");  txtTotalFacturaCliente.setText(Float.toString(TotalFacturaCliente)+" €");  chCifClienteFK.setText(CifClienteFK);    //rescata datos de clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT nombreCliente,direccionCliente,ciudadCliente,cpCliente,telefonoCliente FROM clientes WHERE cifCliente ='"+CifClienteFK+"'");  while (rs.next())  {  // Obtener el campo nombreCliente del registro activo  nombreCliente = rs.getString("nombreCliente");//getString porque es un varchar en la BD  // Obtener el campo direccionCliente del registro activo  direccionCliente = rs.getString("direccionCliente");  // Obtener el campo ciudadCliente del registro activo  ciudadCliente = rs.getString("ciudadCliente");  // Obtener el campo cpClienteo del registro activo  cpCliente = rs.getString("cpCliente");  // Obtener el campo telefonoCliente del registro activo  telefonoCliente = rs.getString("telefonoCliente");  }  //Cerrar los objetos creados  rs.close();//fue el ultimo en abrirse  sen.close();  con.close();  //detalle.setEnabled(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdFacturaCliente.setText(IdFacturaCliente);  txtFechaFacturaCliente.setText(FechaFacturaCliente);  txtBaseImponibleFacturaCliente.setText(Float.toString(BaseImponibleFacturaCliente)+" €");  txtIvaFactutaCliente.setText(Float.toString(IvaFactutaCliente)+" €");  txtTotalFacturaCliente.setText(Float.toString(TotalFacturaCliente)+" €");  chCifClienteFK.setText(CifClienteFK);  imprimir.setEnabled(true);  }  if(imprimir.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.setFont(new Font("Arial",Font.BOLD,60));  pg1.drawString("HC", 50, 100);  pg1.setFont(new Font("Arial",Font.BOLD,16));  pg1.drawString("FACTURA DE CLIENTE", 400, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código Factura: "+txtIdFacturaCliente.getText(), 400, 100);  String fechaSpain = new String();  String fechaAmerica = new String(txtFechaFacturaCliente.getText());  fechaSpain = fechaAmerica.substring(8,10)+"-"+fechaAmerica.substring(5,7) +"-"+fechaAmerica.substring(0,4);  pg1.drawString("Fecha Factura: "+fechaSpain, 400, 120);  pg1.drawString("Nombre Cliente: "+nombreCliente, 50, 140);  pg1.drawString("Dirección Cliente: "+direccionCliente, 50, 160);  pg1.drawString("Ciudad Cliente: "+ciudadCliente, 50, 180);  pg1.drawString("CP Cliente: "+cpCliente, 50, 200);  pg1.drawString("Telefono Cliente: "+telefonoCliente, 50, 220);  pg1.drawString("Nif Cliente: "+chCifClienteFK.getText(), 50, 240);  pg1.drawString("Código", 50, 280);  pg1.drawString("Artículo", 115, 280);  pg1.drawString("Cantidad", 270, 280);  pg1.drawString("Precio", 330, 280);  pg1.drawString("Base", 390, 280);  pg1.drawString("Iva", 450, 280);  pg1.drawString("Subtotal", 515, 280);    pg1.setFont(new Font("Arial",Font.PLAIN,12));    try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT idArticuloFK,UdsArticulos,NombreArticulo,PrecioVentaArticulo,idIvaFK FROM articulos,lineas\_facturas\_clientes WHERE IdArticulo=IdArticuloFK AND IdFacturaClienteFK='"+valor+"'");  //Recorrer los registros obtenidos    int contador=300;  while(rs.next())  {  int ud = Integer.parseInt(rs.getString("UdsArticulos"));  float precio = Float.parseFloat(rs.getString("PrecioVentaArticulo"));  pg1.drawString(rs.getString("idArticuloFK"), 50, contador);  pg1.drawString(rs.getString("NombreArticulo"), 115, contador);  pg1.drawString(rs.getString("UdsArticulos"), 270, contador);  pg1.drawString(rs.getString("PrecioVentaArticulo"), 330, contador);  base=ud\*precio;  pg1.drawString(Float.toString(base), 390, contador);  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  ivaProducto = (float) 0.30;  }  ivaLinea=Math.round(ivaProducto\*(Float.parseFloat(rs.getString("PrecioVentaArticulo")))\*(Float.parseFloat(rs.getString("UdsArticulos")))); ;  pg1.drawString(Float.toString(ivaLinea), 450, contador);  subtotal=ivaLinea+base;  pg1.drawString(Float.toString(subtotal), 520, contador);      /\*String cadena="";  // Obtener el campo idarticulo del registro activo  cadena+=(rs.getString("idArticuloFK"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("NombreArticulo"))+" ";    // Obtener el campo udarticulo del registro activo  cadena+=(rs.getString("UdsArticulos"))+" ";  // Obtener el campo precio venta del registro activo  cadena+=(rs.getString("PrecioVentaArticulo"))+"€ ";  // Obtener el campo iva del registro activo  cadena+=(rs.getString(""))+"€ ";  // Obtener el campo subtotal del registro activo  cadena+=(rs.getString(""))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("idIvaFK");  pg1.drawString(cadena, 50, contador)\*/  contador+=20;  }  pg1.setFont(new Font("Arial",Font.BOLD,14));  pg1.drawString("Subtotal: "+txtBaseImponibleFacturaCliente.getText(), 400, 700);  pg1.drawString("Iva: "+txtIvaFactutaCliente.getText(),400, 720);  pg1.drawString("Total: "+txtTotalFacturaCliente.getText(),400, 740);  pg1.setFont(new Font("Arial",Font.PLAIN,8));  pg1.drawString("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla",150, 800);  pg1.drawString("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es",200, 810);  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.Font;  import java.awt.Frame;  import java.awt.Graphics;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.PrintJob;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaFacturaProveedor extends Frame implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE FACTURA DE PROVEEDOR");  Label LbCodConsulta = new Label ("Código de Factura");  Label LbIdFacturaProveedor = new Label("Código Factura de Proveedor");  Label LbFechaFacturaProveedor = new Label("Fecha Factura de Proveedor");  Label LbBaseImponibleFacturaProveedor = new Label("Base Imponible Factura de Proveedor");  Label LbIvaFactutaProveedor = new Label("IVA Factura de Proveedor");  Label LbTotalFacturaProveedor = new Label("Total Factura de Proveedor");  Label LbCifProveedorFK = new Label("CIF del Proveedor");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  Label txtIdFacturaProveedor = new Label(" ");  Label txtFechaFacturaProveedor = new Label(" ");  Label txtBaseImponibleFacturaProveedor = new Label(" ");  Label txtIvaFactutaProveedor = new Label(" ");  Label txtTotalFacturaProveedor = new Label(" ");  Label chCifProveedorFK = new Label(" ");  //Button  Button cerrar = new Button ("Cerrar");  Button mostrar = new Button ("Mostrar");  Button imprimir= new Button ("Imprimir");  //choice  Choice chIdFacturaProveedor = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  static String valor = new String ();  static String IdFacturaProveedor = new String ();  static String FechaFacturaProveedor= new String ();  float BaseImponibleFacturaProveedor;  float IvaFactutaProveedor;  float TotalFacturaProveedor;  static String CifProveedorFK = new String();    String nombreProveedor = new String();  String direccionProveedor = new String();  String ciudadProveedor = new String();  String cpProveedor = new String();  String telefonoProveedor = new String();    float ivaProducto=0;  float ivaLinea=0;  float base=0;  float subtotal=0;      public consultaFacturaProveedor()  {  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdFacturaProveedor.setFont(fuenteLabelNegrita);  LbFechaFacturaProveedor.setFont(fuenteLabelNegrita);  LbBaseImponibleFacturaProveedor.setFont(fuenteLabelNegrita);  LbIvaFactutaProveedor.setFont(fuenteLabelNegrita);  LbTotalFacturaProveedor.setFont(fuenteLabelNegrita);  LbCifProveedorFK.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE FACTURA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(700, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt codigoconsulta  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=1;  add(chIdFacturaProveedor,gbc);  //situacion label id factura  gbc.gridx=0;  gbc.gridy=2;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdFacturaProveedor,gbc);  //situacion txt id factura  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=2;  add(txtIdFacturaProveedor,gbc);  //situacion label fecha factura  gbc.gridx=0;  gbc.gridy=3;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbFechaFacturaProveedor,gbc);  //situacion txt fecha factura  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=3;  add(txtFechaFacturaProveedor,gbc);  //situacion label base imponible  gbc.gridx=0;  gbc.gridy=4;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbBaseImponibleFacturaProveedor,gbc);  //situacion txt base imponible  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=4;  add(txtBaseImponibleFacturaProveedor,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbIvaFactutaProveedor,gbc);  //situacion txt iva  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=5;  add(txtIvaFactutaProveedor,gbc);  //situacion label total  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbTotalFacturaProveedor,gbc);  //situacion txt total  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=6;  add(txtTotalFacturaProveedor,gbc);  //situacion label ciffk  gbc.gridx=0;  gbc.gridy=7;  gbc.gridwidth=1;  gbc.gridheight=1;  add(LbCifProveedorFK,gbc);  //situacion txt ciffk  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.gridx=1;  gbc.gridy=7;  add(chCifProveedorFK,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.anchor=GridBagConstraints.LINE\_START;  add(imprimir,gbc);  gbc.gridx=2;  gbc.gridy=8;  gbc.anchor=GridBagConstraints.CENTER;  add(cerrar,gbc);  imprimir.setEnabled(false);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=3;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  mostrar.addActionListener(this);  imprimir.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdFacturaProveedor FROM facturas\_Proveedores");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdFacturaProveedor");  chIdFacturaProveedor.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=chIdFacturaProveedor.getSelectedItem();    //rescata datos de facturas Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT \* FROM facturas\_Proveedores WHERE IdFacturaProveedor ='"+valor+"'");  while (rs.next())  {  // Obtener el campo IdArticulo del registro activo  IdFacturaProveedor = rs.getString("IdFacturaProveedor");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  FechaFacturaProveedor = rs.getString("FechaFacturaProveedor");  // Obtener el campo direccion del registro activo  BaseImponibleFacturaProveedor = rs.getFloat("BaseImponibleFacturaProveedor");  // Obtener el campo PrecioCompraArticulo del registro activo  IvaFactutaProveedor = rs.getFloat("IvaFacturaProveedor");  // Obtener el campo PrecioVentaArticulo del registro activo  TotalFacturaProveedor= rs.getFloat("TotalFacturaProveedor");  // Obtener el campo iva del registro activo  CifProveedorFK = rs.getString("CifProveedorFK");  }  //Cerrar los objetos creados  rs.close();//fue el ultimo en abrirse  sen.close();  con.close();  //detalle.setEnabled(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdFacturaProveedor.setText(IdFacturaProveedor);  txtFechaFacturaProveedor.setText(FechaFacturaProveedor);  txtBaseImponibleFacturaProveedor.setText(Float.toString(BaseImponibleFacturaProveedor)+" €");  txtIvaFactutaProveedor.setText(Float.toString(IvaFactutaProveedor)+" €");  txtTotalFacturaProveedor.setText(Float.toString(TotalFacturaProveedor)+" €");  chCifProveedorFK.setText(CifProveedorFK);    //rescata datos de Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT nombreProveedor,direccionProveedor,ciudadProveedor,cpProveedor,telefonoProveedor FROM Proveedores WHERE cifProveedor ='"+CifProveedorFK+"'");  while (rs.next())  {  // Obtener el campo nombreProveedor del registro activo  nombreProveedor = rs.getString("nombreProveedor");//getString porque es un varchar en la BD  // Obtener el campo direccionProveedor del registro activo  direccionProveedor = rs.getString("direccionProveedor");  // Obtener el campo ciudadProveedor del registro activo  ciudadProveedor = rs.getString("ciudadProveedor");  // Obtener el campo cpProveedoro del registro activo  cpProveedor = rs.getString("cpProveedor");  // Obtener el campo telefonoProveedor del registro activo  telefonoProveedor = rs.getString("telefonoProveedor");  }  //Cerrar los objetos creados  rs.close();//fue el ultimo en abrirse  sen.close();  con.close();  //detalle.setEnabled(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdFacturaProveedor.setText(IdFacturaProveedor);  txtFechaFacturaProveedor.setText(FechaFacturaProveedor);  txtBaseImponibleFacturaProveedor.setText(Float.toString(BaseImponibleFacturaProveedor)+" €");  txtIvaFactutaProveedor.setText(Float.toString(IvaFactutaProveedor)+" €");  txtTotalFacturaProveedor.setText(Float.toString(TotalFacturaProveedor)+" €");  chCifProveedorFK.setText(CifProveedorFK);  imprimir.setEnabled(true);  }  if(imprimir.equals(ap.getSource()))  {  //imprimir a pdf  pack();  //Se obtiene el objeto PrintJob  PrintJob pjob1 = this.getToolkit().getPrintJob(this,"", null);  //Se obtiene el objeto graphics sobre el que pintar  Graphics pg1 = pjob1.getGraphics();  //Se fija la fuente de caracteres con la que escribir  //Se escribe el mensaje del Cuadro de Texto indicando  //posición con respecto a la parte superior izquierda  //TITULO  pg1.setFont(new Font("Arial",Font.BOLD,60));  pg1.drawString("HC", 50, 100);  pg1.setFont(new Font("Arial",Font.BOLD,16));  pg1.drawString("FACTURA DE PROVEEDOR", 350, 50);  pg1.setFont(new Font("Arial",Font.BOLD,12));  pg1.drawString("Código Factura: "+txtIdFacturaProveedor.getText(), 350, 100);  String fechaSpain = new String();  String fechaAmerica = new String(txtFechaFacturaProveedor.getText());  fechaSpain = fechaAmerica.substring(8,10)+"-"+fechaAmerica.substring(5,7) +"-"+fechaAmerica.substring(0,4);  pg1.drawString("Fecha Factura: "+fechaSpain, 350, 120);  pg1.drawString("Nombre Proveedor: "+nombreProveedor, 50, 140);  pg1.drawString("Dirección Proveedor: "+direccionProveedor, 50, 160);  pg1.drawString("Ciudad Proveedor: "+ciudadProveedor, 50, 180);  pg1.drawString("CP Proveedor: "+cpProveedor, 50, 200);  pg1.drawString("Telefono Proveedor: "+telefonoProveedor, 50, 220);  pg1.drawString("Nif Proveedor: "+chCifProveedorFK.getText(), 50, 240);  pg1.drawString("Código", 50, 280);  pg1.drawString("Artículo", 115, 280);  pg1.drawString("Cantidad", 270, 280);  pg1.drawString("Precio", 330, 280);  pg1.drawString("Base", 390, 280);  pg1.drawString("Iva", 450, 280);  pg1.drawString("Subtotal", 515, 280);    pg1.setFont(new Font("Arial",Font.PLAIN,12));    try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT idArticuloFK,UdsArticulos,NombreArticulo,PrecioVentaArticulo,idIvaFK FROM articulos,lineas\_facturas\_Proveedores WHERE IdArticulo=IdArticuloFK AND IdFacturaProveedorFK='"+valor+"'");  //Recorrer los registros obtenidos    int contador=300;  while(rs.next())  {  int ud = Integer.parseInt(rs.getString("UdsArticulos"));  float precio = Float.parseFloat(rs.getString("PrecioVentaArticulo"));  pg1.drawString(rs.getString("idArticuloFK"), 50, contador);  pg1.drawString(rs.getString("NombreArticulo"), 115, contador);  pg1.drawString(rs.getString("UdsArticulos"), 270, contador);  pg1.drawString(rs.getString("PrecioVentaArticulo"), 330, contador);  base=ud\*precio;  pg1.drawString(Float.toString(base), 390, contador);  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  ivaProducto = (float) 0.30;  }  ivaLinea=Math.round(ivaProducto\*(Float.parseFloat(rs.getString("PrecioVentaArticulo")))\*(Float.parseFloat(rs.getString("UdsArticulos")))); ;  pg1.drawString(Float.toString(ivaLinea), 450, contador);  subtotal=ivaLinea+base;  pg1.drawString(Float.toString(subtotal), 520, contador);      /\*String cadena="";  // Obtener el campo idarticulo del registro activo  cadena+=(rs.getString("idArticuloFK"))+" ";//getString porque es un varchar en la BD  // Obtener el campo nombre del registro activo  cadena+=(rs.getString("NombreArticulo"))+" ";    // Obtener el campo udarticulo del registro activo  cadena+=(rs.getString("UdsArticulos"))+" ";  // Obtener el campo precio venta del registro activo  cadena+=(rs.getString("PrecioVentaArticulo"))+"€ ";  // Obtener el campo iva del registro activo  cadena+=(rs.getString(""))+"€ ";  // Obtener el campo subtotal del registro activo  cadena+=(rs.getString(""))+"€ ";  // Obtener el campo telefono del registro activo  cadena+=rs.getString("idIvaFK");  pg1.drawString(cadena, 50, contador)\*/  contador+=20;  }      pg1.setFont(new Font("Arial",Font.BOLD,14));  pg1.drawString("Subtotal: "+txtBaseImponibleFacturaProveedor.getText(), 400, 700);  pg1.drawString("Iva: "+txtIvaFactutaProveedor.getText(),400, 720);  pg1.drawString("Total: "+txtTotalFacturaProveedor.getText(),400, 740);  pg1.setFont(new Font("Arial",Font.PLAIN,8));  pg1.drawString("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla",150, 800);  pg1.drawString("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es",200, 810);  //Se finaliza la página  pg1.dispose();  //Se hace que la impresora termine el trabajo y suelte la página  pjob1.end();  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaIva** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaIva extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE IVA");  Label LbCodConsulta = new Label ("Código del Iva");  Label LbIdIva = new Label("Código del IVA");  Label LbDescripcionIva= new Label("Descripción del IVA");  Label LbCantidadIva = new Label("Cantidad del IVA");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");    Label txtIdIva = new Label(" ");  Label txtDescripcionIva = new Label(" ");  Label txtCantidadIva = new Label(" ");    //Button  Button cerrar = new Button ("Cerrar");  Button Mostrar = new Button ("Mostrar");  //choice  Choice chIva = new Choice();    //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String IdIva = new String ();  String DescripcionIva= new String ();  float CantidadIva;    public consultaIva(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());    //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdIva.setFont(fuenteLabelNegrita);  LbDescripcionIva.setFont(fuenteLabelNegrita);  LbCantidadIva.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE IVA");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chIva,gbc);  //situacion label idiva  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdIva,gbc);  //situacion txt idiva  gbc.gridx=1;  gbc.gridy=2;  add(txtIdIva,gbc);  //situacion label descripcion iva  gbc.gridx=0;  gbc.gridy=3;  add(LbDescripcionIva,gbc);  //situacion txt descripcion iva  gbc.gridx=1;  gbc.gridy=3;  add(txtDescripcionIva,gbc);  //situacion label cantidad iva  gbc.gridx=0;  gbc.gridy=4;  add(LbCantidadIva,gbc);  //situacion txt cantidad iva  gbc.gridx=1;  gbc.gridy=4;  add(txtCantidadIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=5;  gbc.insets= new Insets(30, 30, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(Mostrar,gbc);  gbc.gridx=1;  gbc.gridy=5;  gbc.insets= new Insets(30, 0, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=7;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  Mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdIVA FROM tipos\_Iva");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("idIVA");  chIva.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(Mostrar.equals(ap.getSource()))  {  valor=(chIva.getSelectedItem());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM tipos\_iva WHERE IdIva ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo Idiva del registro activo  IdIva = rs1.getString("IdIva");//getString porque es un varchar en la BD  // Obtener el campo Descripcion iva del registro activo  DescripcionIva = rs1.getString("DescripcionIva");  // Obtener el campo cantidad iv del registro activo  CantidadIva = rs1.getFloat("CantidadIva");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdIva.setText(IdIva);  txtDescripcionIva.setText(DescripcionIva);  txtCantidadIva.setText(Float.toString(CantidadIva)+" %");  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **consultaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class consultaProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("CONSULTA DE PROVEEDORES");  Label LbCodConsulta = new Label ("CIF del Proveedor");  Label LbCifProveedor = new Label("CIF del Proveedor");  Label LbNombreProveedor = new Label("Nombre del Proveedor");  Label LbDireccionProveedor = new Label("Dirección del Proveedor");  Label LbCiudadProveedor = new Label("Ciudad del Proveedor");  Label LbCpProveedor = new Label("CP del Proveedor");  Label LbTelefonoProveedor = new Label("Teléfono del Proveedor");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  Label txtCifProveedor = new Label(" ");  Label txtNombreProveedor = new Label(" ");  Label txtDireccionProveedor = new Label(" ");  Label txtCiudadProveedor = new Label(" ");  Label txtCpProveedor = new Label(" ");  Label txtTelefonoProveedor = new Label(" ");  //Button  Button cerrar = new Button ("Cerrar");  Button Mostrar = new Button ("Mostrar");  //choice  Choice chCifProveedor = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String CifProveedor = new String ();  String NombreProveedor= new String ();  String DireccionProveedor = new String ();  String CiudadProveedor = new String ();  int CpProveedor;  int TelefonoProveedor ;  public consultaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbCifProveedor.setFont(fuenteLabelNegrita);  LbNombreProveedor.setFont(fuenteLabelNegrita);  LbDireccionProveedor.setFont(fuenteLabelNegrita);  LbCiudadProveedor.setFont(fuenteLabelNegrita);  LbCpProveedor.setFont(fuenteLabelNegrita);  LbTelefonoProveedor.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("CONSULTA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chCifProveedor,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbCifProveedor,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtCifProveedor,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreProveedor,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreProveedor,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbDireccionProveedor,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=4;  add(txtDireccionProveedor,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbCiudadProveedor,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=5;  add(txtCiudadProveedor,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=6;  add(LbCpProveedor,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=6;  add(txtCpProveedor,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=7;  add(LbTelefonoProveedor,gbc);  //situacion choice iva  gbc.gridx=1;  gbc.gridy=7;  add(txtTelefonoProveedor,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 30, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(Mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  Mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT cifProveedor FROM Proveedores");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("cifProveedor");  chCifProveedor.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(Mostrar.equals(ap.getSource()))  {  valor=chCifProveedor.getSelectedItem();  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM Proveedores WHERE CifProveedor ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  CifProveedor = rs1.getString("CifProveedor");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  NombreProveedor = rs1.getString("NombreProveedor");  // Obtener el campo direccion del registro activo  DireccionProveedor = rs1.getString("DireccionProveedor");  // Obtener el campo direccion del registro activo  CiudadProveedor = rs1.getString("CiudadProveedor");  // Obtener el campo PrecioCompraArticulo del registro activo  CpProveedor = rs1.getInt("CpProveedor");  // Obtener el campo PrecioVentaArticulo del registro activo  TelefonoProveedor= rs1.getInt("TelefonoProveedor");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtCifProveedor.setText(CifProveedor);  txtNombreProveedor.setText(NombreProveedor);  txtDireccionProveedor.setText(DireccionProveedor);  txtCiudadProveedor.setText(CiudadProveedor);  txtCpProveedor.setText(Integer.toString(CpProveedor));  txtTelefonoProveedor.setText(Integer.toString(TelefonoProveedor));  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **eliminarLineaFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.ItemEvent;  import java.awt.event.ItemListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class eliminarLineaFacturaCliente extends Dialog implements WindowListener,ActionListener,ItemListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ELIMINAR LINEA DE FACTURA CLIENTE");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbIdFacturaClienteFK = new Label("Código Factura");  Label LbIdArticuloFK = new Label("Código Artículo");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Linea eliminada correctamente !!");  //Button  Button btEliminar= new Button ("Eliminar Linea");  Button btCerrar = new Button ("Cerrar");  //choice id facturas  Choice chIdFacturaClienteFK = new Choice();  //choice id articulos  Choice chIdArticuloFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdArticuloFK = new String ();  String IdFacturaClienteFK = new String ();  int UdsArticulos=0;  float precioProducto=0;  float ivaProducto=0;  float BI=0;  float IV=0;  float TF=0;  public eliminarLineaFacturaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ELIMINAR LINEA DE FACTURA CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(620, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label idfactura  gbc.gridx=0;  gbc.gridy=2;  add(LbIdFacturaClienteFK,gbc);  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdFacturaClienteFK,gbc);  chIdFacturaClienteFK.addItem(" ");  //situacion label idarticulo  gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIdArticuloFK,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdArticuloFK,gbc);  chIdArticuloFK.addItem(" ");  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btEliminar,gbc);  gbc.gridx=1;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btEliminar.addActionListener(this);  chIdFacturaClienteFK.addItemListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice id factura  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  chIdFacturaClienteFK.removeAll();  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT DISTINCT IdFacturaClienteFK FROM lineas\_facturas\_clientes");  //Recorrer los registros obtenidos  while(rs.next())  {  String dato=rs.getString("IdFacturaClienteFK");  chIdFacturaClienteFK.addItem(dato);  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btEliminar.equals(ap.getSource()))  {  //recuperar datos precio ud e iva articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT IdIvaFK,precioVentaArticulo FROM articulos WHERE idArticulo ='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  precioProducto = rs.getFloat("precioVentaArticulo");  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  ivaProducto = (float) 0.30;  }  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  //System.out.println(precioProducto);  //System.out.println(ivaProducto);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  //calculos en tabla facturas  //recuperar datos de la tabla facturas de clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT baseImponibleFacturaCliente,IvaFacturaCliente,TotalFacturaCliente FROM facturas\_clientes ORDER BY idfacturaCliente DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  BI=rs.getFloat("baseImponibleFacturaCliente");  IV=rs.getFloat("IvaFacturaCliente");  TF=rs.getFloat("TotalFacturaCliente");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  //System.out.println(BI);  //System.out.println(IV);  //System.out.println(TF);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  //recuperar uds de la tabla lineas de facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT udsArticulos FROM Lineas\_facturas\_clientes WHERE IdArticuloFK='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  UdsArticulos=rs.getInt("udsArticulos");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  //System.out.println(UdsArticulos);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  BI-=precioProducto\*UdsArticulos;  IV-=precioProducto\*UdsArticulos\*ivaProducto;  TF=BI+IV;  //restar datos en facturas\_clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("UPDATE facturas\_clientes SET BaseImponibleFacturaCliente='"+BI+"',IvaFacturaCliente='"+IV+"',TotalFacturaCliente='"+TF+"' WHERE IdFacturaCliente='"+chIdFacturaClienteFK.getSelectedItem()+"'");  //Cerrar los objetos creados  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  //eliminar lineas de tabla lineas\_facturas\_clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practic\_hotel\_clinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("DELETE FROM lineas\_facturas\_clientes WHERE IdArticuloFK='"+chIdArticuloFK.getSelectedItem()+"'AND IdfacturaClienteFK='"+chIdFacturaClienteFK.getSelectedItem()+"'");  //Cerrar los objetos creados  sen.close();  con.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  public void itemStateChanged(ItemEvent e)  {  chIdArticuloFK.removeAll();  //llenar el choice id articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdArticuloFK FROM lineas\_facturas\_clientes WHERE IdFacturaClienteFK ='"+chIdFacturaClienteFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdArticuloFK");  chIdArticuloFK.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  } |

|  |
| --- |
| **eliminarLineaFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.ItemEvent;  import java.awt.event.ItemListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class eliminarLineaFacturaProveedor extends Dialog implements WindowListener,ActionListener,ItemListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("ELIMINAR LINEA DE FACTURA PROVEEDOR");  Label LbIndicacion = new Label("CONDICIÓN DE BAJA");  Label LbIdFacturaProveedorFK = new Label("Código Factura");  Label LbIdArticuloFK = new Label("Código Artículo");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Linea eliminada correctamente !!");  //Button  Button btEliminar= new Button ("Eliminar Linea");  Button btCerrar = new Button ("Cerrar");  //choice id facturas  Choice chIdFacturaProveedorFK = new Choice();  //choice id articulos  Choice chIdArticuloFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdArticuloFK = new String ();  String IdFacturaProveedorFK = new String ();  int UdsArticulos=0;  float precioProducto=0;  float ivaProducto=0;  float BI=0;  float IV=0;  float TF=0;  public eliminarLineaFacturaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIndicacion.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("ELIMINAR LINEA DE FACTURA PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(620, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion indicacion  gbc.gridx=0;  gbc.gridy=1;  gbc.insets= new Insets(20, 0, 0, 0);  add(LbIndicacion,gbc);  //situacion label idfactura  gbc.gridx=0;  gbc.gridy=2;  add(LbIdFacturaProveedorFK,gbc);  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdFacturaProveedorFK,gbc);  chIdFacturaProveedorFK.addItem(" ");  //situacion label idarticulo  gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIdArticuloFK,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdArticuloFK,gbc);  chIdArticuloFK.addItem(" ");  //botones  gbc.gridx=0;  gbc.gridy=4;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(btEliminar,gbc);  gbc.gridx=1;  gbc.gridy=4;  add(btCerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=5;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  btCerrar.addActionListener(this);  btEliminar.addActionListener(this);  chIdFacturaProveedorFK.addItemListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice id factura  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  chIdFacturaProveedorFK.removeAll();  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT DISTINCT IdFacturaProveedorFK FROM lineas\_facturas\_Proveedores");  //Recorrer los registros obtenidos  while(rs.next())  {  String dato=rs.getString("IdFacturaProveedorFK");  chIdFacturaProveedorFK.addItem(dato);  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(btCerrar.equals(ap.getSource()))  {  this.dispose();  }  if(btEliminar.equals(ap.getSource()))  {  //recuperar datos precio ud e iva articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT IdIvaFK,precioVentaArticulo FROM articulos WHERE idArticulo ='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  precioProducto = rs.getFloat("precioVentaArticulo");  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  ivaProducto = (float) 0.30;  }  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();    //System.out.println(precioProducto);  //System.out.println(ivaProducto);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }      //calculos en tabla facturas  //recuperar datos de la tabla facturas de Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT baseImponibleFacturaProveedor,IvaFacturaProveedor,TotalFacturaProveedor FROM facturas\_Proveedores ORDER BY idfacturaProveedor DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  BI=rs.getFloat("baseImponibleFacturaProveedor");  IV=rs.getFloat("IvaFacturaProveedor");  TF=rs.getFloat("TotalFacturaProveedor");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  //System.out.println(BI);  //System.out.println(IV);  //System.out.println(TF);  }    catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    //recuperar uds de la tabla lineas de facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT udsArticulos FROM Lineas\_facturas\_Proveedores WHERE IdArticuloFK='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  UdsArticulos=rs.getInt("udsArticulos");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  //System.out.println(UdsArticulos);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    BI-=precioProducto\*UdsArticulos;  IV-=precioProducto\*UdsArticulos\*ivaProducto;  TF=BI+IV;    //System.out.println(BI);  //System.out.println(IV);  //System.out.println(TF);  //restar datos en facturas\_Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("UPDATE facturas\_Proveedores SET BaseImponibleFacturaProveedor='"+BI+"',IvaFacturaProveedor='"+IV+"',TotalFacturaProveedor='"+TF+"' WHERE IdFacturaProveedor='"+chIdFacturaProveedorFK.getSelectedItem()+"'");  //Cerrar los objetos creados  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    //eliminar lineas de tabla lineas\_facturas\_Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practic\_hotel\_clinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("DELETE FROM lineas\_facturas\_Proveedores WHERE IdArticuloFK='"+chIdArticuloFK.getSelectedItem()+"'AND IdfacturaProveedorFK='"+chIdFacturaProveedorFK.getSelectedItem()+"'");  //Cerrar los objetos creados  sen.close();  con.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  else  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  public void itemStateChanged(ItemEvent e)  {  chIdArticuloFK.removeAll();  //llenar el choice id articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdArticuloFK FROM lineas\_facturas\_Proveedores WHERE IdFacturaProveedorFK ='"+chIdFacturaProveedorFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdArticuloFK");  chIdArticuloFK.addItem(dato+" ");  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  } |

|  |
| --- |
| **lineaFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.ItemEvent;  import java.awt.event.ItemListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class lineaFacturaCliente extends Dialog implements WindowListener,ActionListener,ItemListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("NUEVA LÍNEA FACTURA CLIENTE");  Label LbIdFacturaClienteFK= new Label("Código Factura de Cliente");  Label LbIdArticuloFK = new Label("Código del Artículo");  Label LbDescripcionArticulo = new Label("Descripción del Artículo");  Label LbPrecioVentaArticulo = new Label("Precio del Artículo");  Label LbIvaArticulo = new Label("IVA del Articulo");  Label LbUdsArticulo = new Label("Unidades de Artículos");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  Label txtDescripcionArticulo = new Label(" ");  Label txtPrecioVentaArticulo = new Label(" ");  Label txtIvaArticulo = new Label(" ");  //Button  Button insertar = new Button ("Insertar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtUdsArticulo = new TextField(20);  //choice  Choice chIdArticuloFK = new Choice();  Choice chIdFacturaClienteFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdArticuloFK = new String ();  String IdFacturaClienteFK = new String ();  int UdsArticulo;  float precioProducto=0;  float ivaProducto=0;  float BaseImponibleFacturaCliente=0;  float IvaFacturaCliente=0;  float TotalFacturaCliente=0;  public lineaFacturaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdFacturaClienteFK.setFont(fuenteLabelNegrita);  LbIdArticuloFK.setFont(fuenteLabelNegrita);  LbDescripcionArticulo.setFont(fuenteLabelNegrita);  LbPrecioVentaArticulo.setFont(fuenteLabelNegrita);  LbIvaArticulo.setFont(fuenteLabelNegrita);  LbUdsArticulo.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("NUEVA LÍNEA FACTURA CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label id factura  gbc.gridx=0;  gbc.gridy=1;  add(LbIdFacturaClienteFK,gbc);  //situacion choice factura  gbc.gridx=1;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdFacturaClienteFK,gbc);  //chIdFacturaClienteFK.select(altaFacturaCliente.IdFacturaCliente);  //situacion label id articulo  gbc.gridx=0;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIdArticuloFK,gbc);  //situacion choice articulo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdArticuloFK,gbc);  //situacion label desc Articulo  gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbDescripcionArticulo,gbc);  //situacion txt desc Articulo  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtDescripcionArticulo,gbc);    //situacion label precio Articulo  gbc.gridx=0;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbPrecioVentaArticulo,gbc);  //situacion txt precio Articulo  gbc.gridx=1;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtPrecioVentaArticulo,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=5;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIvaArticulo,gbc);  //situacion choice iva  gbc.gridx=1;  gbc.gridy=5;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtIvaArticulo,gbc);  //situacion label ud Articulo  gbc.gridx=0;  gbc.gridy=6;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbUdsArticulo,gbc);  //situacion txt ud Articulo  gbc.gridx=1;  gbc.gridy=6;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtUdsArticulo,gbc);  //botones  gbc.gridx=0;  gbc.gridy=7;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(insertar,gbc);  gbc.gridx=1;  gbc.gridy=7;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  insertar.addActionListener(this);  chIdArticuloFK.addItemListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los codigos de articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT IdArticulo FROM articulos");  //Recorrer los registros obtenidos  while(rs1.next())  {  chIdArticuloFK.addItem(rs1.getString("IdArticulo"));  }  //Cerrar los objetos creados  rs1.close();  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  //llenar el choice con los codigos de facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT IdFacturaCliente FROM facturas\_clientes ORDER BY idfacturaCliente DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  chIdFacturaClienteFK.addItem(rs.getString("IdFacturaCliente"));  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }    public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(insertar.equals(ap.getSource()))  {  IdArticuloFK = (chIdArticuloFK.getSelectedItem());  IdFacturaClienteFK = (chIdFacturaClienteFK.getSelectedItem());  if((txtUdsArticulo.getText().isEmpty()))  {  }  else  {  UdsArticulo = Integer.parseInt(txtUdsArticulo.getText());  }  if ((IdArticuloFK.length()>10)||(txtUdsArticulo.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("INSERT INTO lineas\_facturas\_clientes (IdArticuloFK,IdFacturaClienteFK,UdsArticulos) VALUES ('"+IdArticuloFK+"','"+IdFacturaClienteFK+"','"+UdsArticulo+"')");  //Cerrar los objetos creados  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }    //calculos en tabla facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT baseImponibleFacturaCliente,IvaFacturaCliente,TotalFacturaCliente FROM facturas\_clientes ORDER BY idfacturaCliente DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  BaseImponibleFacturaCliente=rs.getFloat("baseImponibleFacturaCliente");  IvaFacturaCliente=rs.getFloat("IvaFacturaCliente");  TotalFacturaCliente=rs.getFloat("TotalFacturaCliente");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  BaseImponibleFacturaCliente+=precioProducto\*Integer.parseInt(txtUdsArticulo.getText());  IvaFacturaCliente+=(precioProducto\*Integer.parseInt(txtUdsArticulo.getText())\*ivaProducto);  TotalFacturaCliente=BaseImponibleFacturaCliente+IvaFacturaCliente;    //insertar datos en facturas\_clientes  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("UPDATE facturas\_clientes SET BaseImponibleFacturaCliente='"+BaseImponibleFacturaCliente+"',IvaFacturaCliente='"+IvaFacturaCliente+"',TotalFacturaCliente='"+TotalFacturaCliente+"' WHERE IdFacturaCliente='"+IdFacturaClienteFK+"'");  //Cerrar los objetos creados  sen.close();  con.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  public void itemStateChanged(ItemEvent isc)  {  if(chIdArticuloFK .equals(isc.getSource()))  {  //consulta descripcion e iva articulo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT descripcionArticulo,idIvaFK,precioVentaArticulo FROM articulos WHERE idArticulo ='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  txtDescripcionArticulo.setText(rs.getString("descripcionArticulo"));  txtPrecioVentaArticulo.setText(Float.toString(rs.getFloat("precioVentaArticulo")));  precioProducto = rs.getFloat("precioVentaArticulo");  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  txtIvaArticulo.setText("8.00 %");  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  txtIvaArticulo.setText("10.00 %");  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  txtIvaArticulo.setText("21.00 %");  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  txtIvaArticulo.setText("25.00 %");  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  txtIvaArticulo.setText("30.00 %");  ivaProducto = (float) 0.30;  }    }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  } |

|  |
| --- |
| **lineaFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.ItemEvent;  import java.awt.event.ItemListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class lineaFacturaProveedor extends Dialog implements WindowListener,ActionListener,ItemListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("NUEVA LÍNEA FACTURA PROVEEDOR");  Label LbIdFacturaProveedorFK= new Label("Código Factura de Proveedor");  Label LbIdArticuloFK = new Label("Código del Artículo");  Label LbDescripcionArticulo = new Label("Descripción del Artículo");  Label LbPrecioVentaArticulo = new Label("Precio del Artículo");  Label LbIvaArticulo = new Label("IVA del Articulo");  Label LbUdsArticulo = new Label("Unidades de Artículos");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos insertados correctamente !!");  Label txtDescripcionArticulo = new Label(" ");  Label txtPrecioVentaArticulo = new Label(" ");  Label txtIvaArticulo = new Label(" ");  //Button  Button insertar = new Button ("Insertar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtUdsArticulo = new TextField(20);  //choice  Choice chIdArticuloFK = new Choice();  Choice chIdFacturaProveedorFK = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String IdArticuloFK = new String ();  String IdFacturaProveedorFK = new String ();  int UdsArticulo;  float precioProducto=0;  float ivaProducto=0;  float BaseImponibleFacturaProveedor=0;  float IvaFacturaProveedor=0;  float TotalFacturaProveedor=0;  public lineaFacturaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdFacturaProveedorFK.setFont(fuenteLabelNegrita);  LbIdArticuloFK.setFont(fuenteLabelNegrita);  LbDescripcionArticulo.setFont(fuenteLabelNegrita);  LbPrecioVentaArticulo.setFont(fuenteLabelNegrita);  LbIvaArticulo.setFont(fuenteLabelNegrita);  LbUdsArticulo.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("NUEVA LÍNEA FACTURA PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 450);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label id factura  gbc.gridx=0;  gbc.gridy=1;  add(LbIdFacturaProveedorFK,gbc);  //situacion choice factura  gbc.gridx=1;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdFacturaProveedorFK,gbc);  //chIdFacturaProveedorFK.select(altaFacturaProveedor.IdFacturaProveedor);  //situacion label id articulo  gbc.gridx=0;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIdArticuloFK,gbc);  //situacion choice articulo  gbc.gridx=1;  gbc.gridy=2;  gbc.anchor=GridBagConstraints.LINE\_START;  add(chIdArticuloFK,gbc);  //situacion label desc Articulo  gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbDescripcionArticulo,gbc);  //situacion txt desc Articulo  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtDescripcionArticulo,gbc);    //situacion label precio Articulo  gbc.gridx=0;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbPrecioVentaArticulo,gbc);  //situacion txt precio Articulo  gbc.gridx=1;  gbc.gridy=4;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtPrecioVentaArticulo,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=5;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbIvaArticulo,gbc);  //situacion choice iva  gbc.gridx=1;  gbc.gridy=5;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtIvaArticulo,gbc);  //situacion label ud Articulo  gbc.gridx=0;  gbc.gridy=6;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbUdsArticulo,gbc);  //situacion txt ud Articulo  gbc.gridx=1;  gbc.gridy=6;  gbc.anchor=GridBagConstraints.LINE\_START;  add(txtUdsArticulo,gbc);  //botones  gbc.gridx=0;  gbc.gridy=7;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(insertar,gbc);  gbc.gridx=1;  gbc.gridy=7;  gbc.insets= new Insets(30, 30, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=8;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  insertar.addActionListener(this);  chIdArticuloFK.addItemListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con los codigos de articulos  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT IdArticulo FROM articulos");  //Recorrer los registros obtenidos  while(rs1.next())  {  chIdArticuloFK.addItem(rs1.getString("IdArticulo"));  }  //Cerrar los objetos creados  rs1.close();  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  //llenar el choice con los codigos de facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT IdFacturaProveedor FROM facturas\_Proveedores ORDER BY idfacturaProveedor DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  chIdFacturaProveedorFK.addItem(rs.getString("IdFacturaProveedor"));  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }    public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(insertar.equals(ap.getSource()))  {  IdArticuloFK = (chIdArticuloFK.getSelectedItem());  IdFacturaProveedorFK = (chIdFacturaProveedorFK.getSelectedItem());  if((txtUdsArticulo.getText().isEmpty()))  {  }  else  {  UdsArticulo = Integer.parseInt(txtUdsArticulo.getText());  }  if ((IdArticuloFK.length()>10)||(txtUdsArticulo.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("INSERT INTO lineas\_facturas\_Proveedores (IdArticuloFK,IdFacturaProveedorFK,UdsArticulos) VALUES ('"+IdArticuloFK+"','"+IdFacturaProveedorFK+"','"+UdsArticulo+"')");  //Cerrar los objetos creados  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }    //calculos en tabla facturas  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT baseImponibleFacturaProveedor,IvaFacturaProveedor,TotalFacturaProveedor FROM facturas\_Proveedores ORDER BY idfacturaProveedor DESC LIMIT 1");  //Recorrer los registros obtenidos  while(rs.next())  {  BaseImponibleFacturaProveedor=rs.getFloat("baseImponibleFacturaProveedor");  IvaFacturaProveedor=rs.getFloat("IvaFacturaProveedor");  TotalFacturaProveedor=rs.getFloat("TotalFacturaProveedor");  }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  BaseImponibleFacturaProveedor+=precioProducto\*Integer.parseInt(txtUdsArticulo.getText());  IvaFacturaProveedor+=(precioProducto\*Integer.parseInt(txtUdsArticulo.getText())\*ivaProducto);  TotalFacturaProveedor=BaseImponibleFacturaProveedor+IvaFacturaProveedor;    //insertar datos en facturas\_Proveedores  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen.executeUpdate("UPDATE facturas\_Proveedores SET BaseImponibleFacturaProveedor='"+BaseImponibleFacturaProveedor+"',IvaFacturaProveedor='"+IvaFacturaProveedor+"',TotalFacturaProveedor='"+TotalFacturaProveedor+"' WHERE IdFacturaProveedor='"+IdFacturaProveedorFK+"'");  //Cerrar los objetos creados  sen.close();  con.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  public void itemStateChanged(ItemEvent isc)  {  if(chIdArticuloFK .equals(isc.getSource()))  {  //consulta descripcion e iva articulo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen = con.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs = sen.executeQuery("SELECT descripcionArticulo,idIvaFK,precioVentaArticulo FROM articulos WHERE idArticulo ='"+chIdArticuloFK.getSelectedItem()+"'");  //Recorrer los registros obtenidos  while(rs.next())  {  txtDescripcionArticulo.setText(rs.getString("descripcionArticulo"));  txtPrecioVentaArticulo.setText(Float.toString(rs.getFloat("precioVentaArticulo")));  precioProducto = rs.getFloat("precioVentaArticulo");  String datoIva =rs.getString("idIvaFK");  if (datoIva.equals("iva8"))  {  txtIvaArticulo.setText("8.00 %");  ivaProducto = (float) 0.08;  }  if (datoIva.equals("iva10"))  {  txtIvaArticulo.setText("10.00 %");  ivaProducto = (float) 0.10;  }  if (datoIva.equals("iva21"))  {  txtIvaArticulo.setText("21.00 %");  ivaProducto = (float) 0.21;  }  if (datoIva.equals("iva25"))  {  txtIvaArticulo.setText("25.00 %");  ivaProducto = (float) 0.25;  }  if (datoIva.equals("iva30"))  {  txtIvaArticulo.setText("30.00 %");  ivaProducto = (float) 0.30;  }    }  //Cerrar los objetos creados  rs.close();  sen.close();  con.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  } |

|  |
| --- |
| **modificacionArticulo** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionArticulo extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN DE ARTÍCULO");  Label LbCodConsulta = new Label ("Código del Artículo");  Label LbIdArticulo = new Label("Código del Artículo");  Label LbNombreArticulo= new Label("Nombre del Artículo");  Label LbDescripArticulo = new Label("Descripción del Artículo");  Label LbPrecCompArticulo = new Label("Precio de Compra");  Label LbPrecVentArticulo = new Label("Precio de Venta");  Label LbIvaArticulo = new Label("IVA del Articulo");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button mostrar = new Button ("Mostrar");  Button modificar = new Button ("Modificar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtIdArticulo = new TextField(20);  TextField txtNombreArticulo = new TextField(20);  TextField txtDescripArticulo = new TextField(30);  TextField txtPrecCompArticulo = new TextField(20);  TextField txtPrecVentArticulo = new TextField(20);  //choice  Choice chIva = new Choice();  Choice chIdArticulo = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String IdArticulo = new String ();  String NombreArticulo= new String ();  String DescripcionArticulo = new String ();  float PrecioCompraArticulo;  float PrecioVentaArticulo;  String ivaNumero = new String();  String IdIVAFK = new String ();  public modificacionArticulo(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdArticulo.setFont(fuenteLabelNegrita);  LbNombreArticulo.setFont(fuenteLabelNegrita);  LbDescripArticulo.setFont(fuenteLabelNegrita);  LbPrecCompArticulo.setFont(fuenteLabelNegrita);  LbPrecVentArticulo.setFont(fuenteLabelNegrita);  LbIvaArticulo.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN DE ARTÍCULO");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chIdArticulo,gbc);  //situacion label idArticulo  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=2;  add(txtIdArticulo,gbc);  //situacion label nombre Articulo  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreArticulo,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreArticulo,gbc);  //situacion label descripcion Articulo  gbc.gridx=0;  gbc.gridy=4;  add(LbDescripArticulo,gbc);  //situacion txt descripcion  gbc.gridx=1;  gbc.gridy=4;  add(txtDescripArticulo,gbc);  //situacion label precio compra Articulo  gbc.gridx=0;  gbc.gridy=5;  add(LbPrecCompArticulo,gbc);  //situacion txt precio compra  gbc.gridx=1;  gbc.gridy=5;  add(txtPrecCompArticulo,gbc);  //situacion label precio venta Articulo  gbc.gridx=0;  gbc.gridy=6;  add(LbPrecVentArticulo,gbc);  //situacion txt precio venta  gbc.gridx=1;  gbc.gridy=6;  add(txtPrecVentArticulo,gbc);  //situacion label iva  gbc.gridx=0;  gbc.gridy=7;  add(LbIvaArticulo,gbc);  //situacion chiice iva  gbc.gridx=1;  gbc.gridy=7;  add(chIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 60);  add(modificar,gbc);  modificar.setEnabled(false);  gbc.gridx=2;  gbc.gridy=8;  gbc.insets= new Insets(30, -10, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  modificar.addActionListener(this);  mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el iva  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CantidadIVA FROM tipos\_iva");  //Recorrer los registros obtenidos  while(rs2.next())  {  float dato=rs2.getFloat("CantidadIVA");  String dato1 = new String (Float.toString(dato));  chIva.addItem(dato1);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdArticulo FROM articulos");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("idArticulo");  chIdArticulo.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=chIdArticulo.getSelectedItem();  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM articulos WHERE IdArticulo ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  IdArticulo = rs1.getString("IdArticulo");//getString porque es un varchar en la BD  // Obtener el campo nombreArticulo del registro activo  NombreArticulo = rs1.getString("NombreArticulo");  // Obtener el campo descripcion del registro activo  DescripcionArticulo = rs1.getString("DescripcionArticulo");  // Obtener el campo PrecioCompraArticulo del registro activo  PrecioCompraArticulo = rs1.getFloat("PrecioCompraArticulo");  // Obtener el campo PrecioVentaArticulo del registro activo  PrecioVentaArticulo= rs1.getFloat("PrecioVentaArticulo");  // Obtener el campo iva del registro activo  IdIVAFK = rs1.getString("IdIVAFK");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdArticulo.setText(IdArticulo);  txtIdArticulo.setEnabled(false);  txtNombreArticulo.setText(NombreArticulo);  txtDescripArticulo.setText(DescripcionArticulo);  txtPrecCompArticulo.setText(Float.toString(PrecioCompraArticulo));  txtPrecVentArticulo.setText(Float.toString(PrecioVentaArticulo));  if (IdIVAFK.equals("iva10"))  {  chIva.select(0);  }  if (IdIVAFK.equals("iva21"))  {  chIva.select(1);  }  if (IdIVAFK.equals("iva25"))  {  chIva.select(2);  }  if (IdIVAFK.equals("iva30"))  {  chIva.select(3);  }  if (IdIVAFK.equals("iva8"))  {  chIva.select(4);  }  modificar.setEnabled(true);  }  if(modificar.equals(ap.getSource()))  {  IdArticulo = (txtIdArticulo.getText());  NombreArticulo= (txtNombreArticulo.getText());  DescripcionArticulo = (txtDescripArticulo.getText());  if((txtPrecCompArticulo.getText().isEmpty()))  {  }  else  {  PrecioCompraArticulo = Float.parseFloat(txtPrecCompArticulo.getText());  }  if((txtPrecVentArticulo.getText().isEmpty()))  {  }  else  {  PrecioVentaArticulo = Float.parseFloat(txtPrecVentArticulo.getText());  }  ivaNumero = (chIva.getSelectedItem());  if (ivaNumero.equals("8.0"))  {  IdIVAFK="iva8";  }  if (ivaNumero.equals("10.0"))  {  IdIVAFK="iva10";  }  if (ivaNumero.equals("21.0"))  {  IdIVAFK="iva21";  }  if (ivaNumero.equals("25.0"))  {  IdIVAFK="iva25";  }  if (ivaNumero.equals("30.0"))  {  IdIVAFK="iva30";  }  if ((IdArticulo.length()>10)||(txtIdArticulo.getText().isEmpty())||(NombreArticulo.length()>45)||(txtNombreArticulo.getText().isEmpty())||(DescripcionArticulo.length()>100)||(txtDescripArticulo.getText().isEmpty())||(PrecioCompraArticulo>99999999.99)||(PrecioCompraArticulo<0)||(txtPrecCompArticulo.getText().isEmpty())||(PrecioVentaArticulo>99999999.99)||(PrecioVentaArticulo<0)||(txtPrecVentArticulo.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("UPDATE articulos SET NombreArticulo='"+NombreArticulo+"',DescripcionArticulo='"+DescripcionArticulo+"',PrecioCompraArticulo='"+PrecioCompraArticulo+"',PrecioVentaArticulo='"+PrecioVentaArticulo+"',IdIVAFK='"+IdIVAFK+"' WHERE IdArticulo='"+IdArticulo+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }    }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **modificacionCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN DE CLIENTE");  Label LbCodConsulta = new Label ("CIF del Cliente");  Label LbCifCliente = new Label("CIF del Cliente");  Label LbNombreCliente = new Label("Nombre del Cliente");  Label LbDireccionCliente = new Label("Dirección del Cliente");  Label LbCiudadCliente = new Label("Ciudad del Cliente");  Label LbCpCliente = new Label("CP del Cliente");  Label LbTelefonoCliente = new Label("Teléfono del Cliente");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button mostrar = new Button ("Mostrar");  Button modificar = new Button ("Modificar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtCifCliente = new TextField(20);  TextField txtNombreCliente = new TextField(20);  TextField txtDireccionCliente = new TextField(30);  TextField txtCiudadCliente = new TextField(20);  TextField txtCpCliente = new TextField(20);  TextField txtTelefonoCliente = new TextField(20);  //choice  Choice chCifCliente = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String CifCliente = new String ();  String NombreCliente = new String ();  String DireccionCliente = new String ();  String CiudadCliente = new String ();  int CpCliente;  int TelefonoCliente;  public modificacionCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbCifCliente.setFont(fuenteLabelNegrita);  LbNombreCliente.setFont(fuenteLabelNegrita);  LbDireccionCliente.setFont(fuenteLabelNegrita);  LbCiudadCliente.setFont(fuenteLabelNegrita);  LbCpCliente.setFont(fuenteLabelNegrita);  LbTelefonoCliente.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chCifCliente,gbc);  //situacion label cif  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbCifCliente,gbc);  //situacion txt cif  gbc.gridx=1;  gbc.gridy=2;  add(txtCifCliente,gbc);  //situacion label nombre  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreCliente,gbc);  //situacion txt nombre  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreCliente,gbc);  //situacion label direccion  gbc.gridx=0;  gbc.gridy=4;  add(LbDireccionCliente,gbc);  //situacion txt direccion  gbc.gridx=1;  gbc.gridy=4;  add(txtDireccionCliente,gbc);  //situacion label ciudad  gbc.gridx=0;  gbc.gridy=5;  add(LbCiudadCliente,gbc);  //situacion txt ciudad  gbc.gridx=1;  gbc.gridy=5;  add(txtCiudadCliente,gbc);  //situacion label cp  gbc.gridx=0;  gbc.gridy=6;  add(LbCpCliente,gbc);  //situacion txt cp  gbc.gridx=1;  gbc.gridy=6;  add(txtCpCliente,gbc);  //situacion label tf  gbc.gridx=0;  gbc.gridy=7;  add(LbTelefonoCliente,gbc);  //situacion tf  gbc.gridx=1;  gbc.gridy=7;  add(txtTelefonoCliente,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 60);  add(modificar,gbc);  modificar.setEnabled(false);  gbc.gridx=2;  gbc.gridy=8;  gbc.insets= new Insets(30, -10, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  modificar.addActionListener(this);  mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CifCliente FROM clientes");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("CifCliente");  chCifCliente .addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=(chCifCliente.getSelectedItem());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM clientes WHERE CifCliente ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  CifCliente = rs1.getString("CifCliente");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  NombreCliente = rs1.getString("NombreCliente");  // Obtener el campo direccion del registro activo  DireccionCliente = rs1.getString("DireccionCliente");  // Obtener el campo ciudad del registro activo  CiudadCliente = rs1.getString("CiudadCliente");  // Obtener el campo PrecioCompraArticulo del registro activo  CpCliente = rs1.getInt("CpCliente");  // Obtener el campo PrecioVentaArticulo del registro activo  TelefonoCliente = rs1.getInt("TelefonoCliente");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtCifCliente.setText(CifCliente);  txtCifCliente.setEnabled(false);  txtNombreCliente.setText(NombreCliente);  txtDireccionCliente.setText(DireccionCliente);  txtCiudadCliente.setText(CiudadCliente);  txtCpCliente.setText(Integer.toString(CpCliente));  txtTelefonoCliente.setText(Integer.toString(TelefonoCliente));  modificar.setEnabled(true);  }  if(modificar.equals(ap.getSource()))  {  CifCliente = (txtCifCliente.getText());  NombreCliente= (txtNombreCliente.getText());  DireccionCliente = (txtDireccionCliente.getText());  CiudadCliente = (txtCiudadCliente.getText());  if((txtCpCliente.getText().isEmpty()))  {  }  else  {  CpCliente = Integer.parseInt(txtCpCliente.getText());  }  if((txtTelefonoCliente.getText().isEmpty()))  {  }  else  {  TelefonoCliente = Integer.parseInt(txtTelefonoCliente.getText());  }  if ((CifCliente.length()>9)||(txtCifCliente.getText().isEmpty())||(NombreCliente.length()>45)||(txtNombreCliente.getText().isEmpty())||(DireccionCliente.length()>45)||(txtDireccionCliente.getText().isEmpty())||(CiudadCliente.length()>45)||(txtCiudadCliente.getText().isEmpty())||(CpCliente>99999)||(CpCliente<0)||(txtCpCliente.getText().isEmpty())||(TelefonoCliente>999999999)||(TelefonoCliente<0)||(txtTelefonoCliente.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("UPDATE clientes SET NombreCliente='"+NombreCliente+"',DireccionCliente='"+DireccionCliente+"',CiudadCliente='"+CiudadCliente+"',CpCliente='"+CpCliente+"',TelefonoCliente='"+TelefonoCliente+"' WHERE CifCliente='"+CifCliente+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **modificacionFacturaCliente** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionFacturaCliente extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN FACTURA DE CLIENTE");  Label LbIdFacturaCliente = new Label ("Código de Factura");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button cerrar = new Button ("Cerrar");  Button nuevaLinea = new Button ("Nueva Linea");  Button eliminarLinea = new Button ("Eliminar Linea");  //choice  Choice chIdFacturaCliente = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String IdFacturaCliente = new String ();  float BaseImponibleFacturaCliente;  float IvaFacturaCliente;  float TotalFacturaCliente;  String CifClienteFK = new String();  public modificacionFacturaCliente(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdFacturaCliente.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN FACTURA DE CLIENTE");  setResizable(false);  setLocation(50,50);  setSize(700, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbIdFacturaCliente,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chIdFacturaCliente,gbc);  //botones    gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(nuevaLinea,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.CENTER;  add(eliminarLinea,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  nuevaLinea.addActionListener(this);  eliminarLinea.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdFacturaCliente FROM facturas\_Clientes");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdFacturaCliente");  chIdFacturaCliente.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(nuevaLinea.equals(ap.getSource()))  {  lineaFacturaCliente lFCli = new lineaFacturaCliente(null);  }  if(eliminarLinea.equals(ap.getSource()))  {  eliminarLineaFacturaCliente elLFCli = new eliminarLineaFacturaCliente(null);  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **modificacionFacturaProveedor** |

|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionFacturaProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN FACTURA DE PROVEEDOR");  Label LbIdFacturaProveedor = new Label ("Código de Factura");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button cerrar = new Button ("Cerrar");  Button nuevaLinea = new Button ("Nueva Linea");  Button eliminarLinea = new Button ("Eliminar Linea");  //choice  Choice chIdFacturaProveedor = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String IdFacturaProveedor = new String ();  float BaseImponibleFacturaProveedor;  float IvaFacturaProveedor;  float TotalFacturaProveedor;  String CifProveedorFK = new String();  public modificacionFacturaProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbIdFacturaProveedor.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN FACTURA DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(700, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbIdFacturaProveedor,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chIdFacturaProveedor,gbc);  //botones    gbc.gridx=0;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_START;  add(nuevaLinea,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.CENTER;  add(eliminarLinea,gbc);  gbc.gridx=1;  gbc.gridy=3;  gbc.anchor=GridBagConstraints.LINE\_END;  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  nuevaLinea.addActionListener(this);  eliminarLinea.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdFacturaProveedor FROM facturas\_Proveedores");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("IdFacturaProveedor");  chIdFacturaProveedor.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(nuevaLinea.equals(ap.getSource()))  {  lineaFacturaProveedor lFCli = new lineaFacturaProveedor(null);  }  if(eliminarLinea.equals(ap.getSource()))  {  eliminarLineaFacturaProveedor elLFCli = new eliminarLineaFacturaProveedor(null);  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **modificacionIva** |

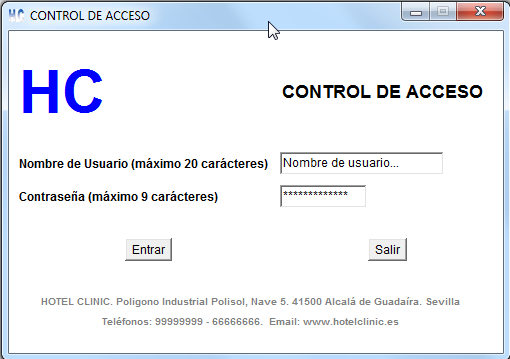
|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionIva extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN DE IVA");  Label LbCodConsulta = new Label ("Código del IVA");  Label LbIdIva = new Label("Código del IVA");  Label LbDescripcionIva = new Label("Descripción del IVA");  Label LbCantidadIva = new Label("Cantidad del IVA");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button Mostrar = new Button ("Mostrar");  Button modificar = new Button ("Modificar");  Button salir = new Button ("Salir");  //texfield  TextField txtIdIva = new TextField(20);  TextField txtDescripcionIva = new TextField(20);  TextField txtCantidadIva = new TextField(20);  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //choice  Choice chIva = new Choice();  //variables gestion BD  String valor = new String ();  String IdIva = new String ();  String DescripcionIva = new String ();  float CantidadIva;  public modificacionIva(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbIdIva.setFont(fuenteLabelNegrita);  LbDescripcionIva.setFont(fuenteLabelNegrita);  LbCantidadIva.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN DE IVA");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt CodConsulta  gbc.gridx=1;  gbc.gridy=1;  add(chIva,gbc);  //situacion label idiva  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbIdIva,gbc);  //situacion txt idiva  gbc.gridx=1;  gbc.gridy=2;  add(txtIdIva,gbc);  //situacion label descripcion iva  gbc.gridx=0;  gbc.gridy=3;  add(LbDescripcionIva,gbc);  //situacion txt descripcion iva  gbc.gridx=1;  gbc.gridy=3;  add(txtDescripcionIva,gbc);  //situacion label cantidad iva  gbc.gridx=0;  gbc.gridy=4;  add(LbCantidadIva,gbc);  //situacion txt cantidad iva  gbc.gridx=1;  gbc.gridy=4;  add(txtCantidadIva,gbc);  //botones  gbc.gridx=0;  gbc.gridy=5;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(Mostrar,gbc);  gbc.gridx=1;  gbc.gridy=5;  gbc.insets= new Insets(30, 0, 0, 60);  modificar.setEnabled(false);  add(modificar,gbc);  gbc.gridx=2;  gbc.gridy=5;  gbc.insets= new Insets(30, -10, 0, 0);  add(salir,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=6;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=7;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  salir.addActionListener(this);  modificar.addActionListener(this);  Mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT IdIVA FROM tipos\_Iva");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("idIVA");  chIva.addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(salir.equals(ap.getSource()))  {  this.dispose();  }  if(Mostrar.equals(ap.getSource()))  {  valor=(chIva.getSelectedItem());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM tipos\_iva WHERE IdIva ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdIva del registro activo  IdIva = rs1.getString("IdIva");//getString porque es un varchar en la BD  // Obtener el campo DescripcionIvao del registro activo  DescripcionIva = rs1.getString("DescripcionIva");  // Obtener el campo CantidadIva del registro activo  CantidadIva = rs1.getFloat("CantidadIva");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtIdIva.setText(IdIva);  txtIdIva.setEnabled(false);  txtDescripcionIva.setText(DescripcionIva);  txtCantidadIva.setText(Float.toString(CantidadIva));  modificar.setEnabled(true);  }  if(modificar.equals(ap.getSource()))  {  IdIva = (txtIdIva.getText());  DescripcionIva = (txtDescripcionIva.getText());  if((txtCantidadIva.getText().isEmpty()))  {  }  else  {  CantidadIva = Float.parseFloat(txtCantidadIva.getText());  }  if ((IdIva.length()>9)||(txtIdIva.getText().isEmpty())||(DescripcionIva.length()>45)||(txtDescripcionIva.getText().isEmpty())||(txtCantidadIva.getText().isEmpty())||(CantidadIva>99.99)||(CantidadIva<0))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("UPDATE tipos\_iva SET DescripcionIva='"+DescripcionIva+"',CantidadIva='"+CantidadIva+"' WHERE IdIva='"+IdIva+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **modificacionProveedor** |

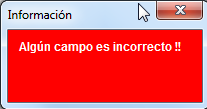
|  |
| --- |
| package es.studium.practica\_primera\_parte;  import java.awt.Button;  import java.awt.Choice;  import java.awt.Color;  import java.awt.Dialog;  import java.awt.FlowLayout;  import java.awt.Font;  import java.awt.GridBagConstraints;  import java.awt.GridBagLayout;  import java.awt.Insets;  import java.awt.Label;  import java.awt.TextField;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import java.awt.event.WindowEvent;  import java.awt.event.WindowListener;  import java.io.BufferedReader;  import java.io.InputStreamReader;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.sql.Statement;  import javax.swing.ImageIcon;  public class modificacionProveedor extends Dialog implements WindowListener,ActionListener  {  //Label  Label logo = new Label("HC");  Label LbTitulo = new Label("MODIFICACIÓN DE PROVEEDOR");  Label LbCodConsulta = new Label ("CIF del Proveedor");  Label LbCifProveedor = new Label("CIF del Proveedor");  Label LbNombreProveedor = new Label("Nombre del Proveedor");  Label LbDireccionProveedor = new Label("Dirección del Proveedor");  Label LbCiudadProveedor = new Label("Ciudad del Proveedor");  Label LbCpProveedor = new Label("CP del Proveedor");  Label LbTelefonoProveedor = new Label("Teléfono del Proveedor");  Label hClinicNom= new Label("HOTEL CLINIC. Poligono Industrial Polisol, Nave 5. 41500 Alcalá de Guadaíra. Sevilla");  Label hClinicTf= new Label("Teléfonos: 99999999 - 66666666. Email: www.hotelclinic.es");  Label informacion1 = new Label("Algún campo es incorrecto !!");  Label informacion2 = new Label("Datos modificados correctamente !!");  //Button  Button mostrar = new Button ("Mostrar");  Button modificar = new Button ("Modificar");  Button cerrar = new Button ("Cerrar");  //texfield  TextField txtCifProveedor = new TextField(20);  TextField txtNombreProveedor = new TextField(20);  TextField txtDireccionProveedor = new TextField(30);  TextField txtCiudadProveedor = new TextField(20);  TextField txtCpProveedor = new TextField(20);  TextField txtTelefonoProveedor = new TextField(20);  //choice  Choice chCifProveedor = new Choice();  //fuentes  Font fuenteLabelNegrita = new Font("Arial", Font.BOLD, 12);  Font fuenteLabelTitulo = new Font("Arial", Font.BOLD, 18);  Font fuenteLabelLogo= new Font("Arial", Font.BOLD,60);  Font fuenteLabelHC= new Font("Arial", Font.CENTER\_BASELINE,10);  //ventana de dialogo  Dialog dialogInformacion1 = new Dialog (this,"Información",true);  Dialog dialogInformacion2 = new Dialog (this,"Información",true);  //variables gestion BD  String valor = new String ();  String CifProveedor = new String ();  String NombreProveedor = new String ();  String DireccionProveedor = new String ();  String CiudadProveedor = new String ();  int CpProveedor;  int TelefonoProveedor;  public modificacionProveedor(Dialog owner)  {  super(owner);  //cambio de icono  ImageIcon icono = new ImageIcon("img\\icono1.jpg");  this.setIconImage(icono.getImage());  //aplicar fuentes  logo.setFont(fuenteLabelLogo);  LbTitulo.setFont(fuenteLabelTitulo);  LbCodConsulta.setFont(fuenteLabelNegrita);  LbCifProveedor.setFont(fuenteLabelNegrita);  LbNombreProveedor.setFont(fuenteLabelNegrita);  LbDireccionProveedor.setFont(fuenteLabelNegrita);  LbCiudadProveedor.setFont(fuenteLabelNegrita);  LbCpProveedor.setFont(fuenteLabelNegrita);  LbTelefonoProveedor.setFont(fuenteLabelNegrita);  hClinicNom.setFont(fuenteLabelHC);  hClinicTf.setFont(fuenteLabelHC);  logo.setForeground(Color.blue);  hClinicNom.setForeground(Color.GRAY);  hClinicTf.setForeground(Color.GRAY);  dialogInformacion1.setBackground(Color.red);  dialogInformacion1.setForeground(Color.white);  dialogInformacion1.setFont(fuenteLabelNegrita);  dialogInformacion2.setBackground(Color.blue);  dialogInformacion2.setForeground(Color.white);  dialogInformacion2.setFont(fuenteLabelNegrita);  setTitle("MODIFICACIÓN DE PROVEEDOR");  setResizable(false);  setLocation(50,50);  setSize(550, 500);  setVisible(true);  setLayout(new GridBagLayout());  GridBagConstraints gbc = new GridBagConstraints();  //situacion logo  gbc.gridx=0;  gbc.gridy=0;  gbc.gridwidth=1;  gbc.gridheight=1;  gbc.anchor=GridBagConstraints.LINE\_START;  add(logo,gbc);  //situacion titulo  gbc.gridx=1;  gbc.gridy=0;  gbc.anchor=GridBagConstraints.LINE\_END;  add(LbTitulo,gbc);  //situacion label CodConsulta  gbc.gridx=0;  gbc.gridy=1;  gbc.anchor=GridBagConstraints.LINE\_START;  gbc.insets= new Insets(20, 0, 20, 0);  add(LbCodConsulta,gbc);  //situacion txt idArticulo  gbc.gridx=1;  gbc.gridy=1;  add(chCifProveedor,gbc);  //situacion label cif  gbc.gridx=0;  gbc.gridy=2;  gbc.insets= new Insets(10, 0, 0, 10);  add(LbCifProveedor,gbc);  //situacion txt cif  gbc.gridx=1;  gbc.gridy=2;  add(txtCifProveedor,gbc);  //situacion label nombre  gbc.gridx=0;  gbc.gridy=3;  add(LbNombreProveedor,gbc);  //situacion txt nombre  gbc.gridx=1;  gbc.gridy=3;  add(txtNombreProveedor,gbc);  //situacion label direccion  gbc.gridx=0;  gbc.gridy=4;  add(LbDireccionProveedor,gbc);  //situacion txt direccion  gbc.gridx=1;  gbc.gridy=4;  add(txtDireccionProveedor,gbc);  //situacion label ciudad  gbc.gridx=0;  gbc.gridy=5;  add(LbCiudadProveedor,gbc);  //situacion txt ciudad  gbc.gridx=1;  gbc.gridy=5;  add(txtCiudadProveedor,gbc);  //situacion label cp  gbc.gridx=0;  gbc.gridy=6;  add(LbCpProveedor,gbc);  //situacion txt cp  gbc.gridx=1;  gbc.gridy=6;  add(txtCpProveedor,gbc);  //situacion label tf  gbc.gridx=0;  gbc.gridy=7;  add(LbTelefonoProveedor,gbc);  //situacion tf  gbc.gridx=1;  gbc.gridy=7;  add(txtTelefonoProveedor,gbc);  //botones  gbc.gridx=0;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 0);  gbc.anchor=GridBagConstraints.CENTER;  add(mostrar,gbc);  gbc.gridx=1;  gbc.gridy=8;  gbc.insets= new Insets(30, 0, 0, 60);  add(modificar,gbc);  modificar.setEnabled(false);  gbc.gridx=2;  gbc.gridy=8;  gbc.insets= new Insets(30, -10, 0, 0);  add(cerrar,gbc);  //datos hc  //nombre  gbc.gridx=0;  gbc.gridy=9;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(30, 0, 0, 0);  add(hClinicNom,gbc);  //telefono  gbc.gridx=0;  gbc.gridy=10;  gbc.gridwidth=2;  gbc.gridheight=1;  gbc.insets= new Insets(0, 0, 0, 0);  add(hClinicTf,gbc);  //escuchadores  addWindowListener(this);  cerrar.addActionListener(this);  modificar.addActionListener(this);  mostrar.addActionListener(this);  dialogInformacion1.addWindowListener(this);  dialogInformacion2.addWindowListener(this);  //gestion cuadro de dialogo1  dialogInformacion1.setLayout(new FlowLayout());  dialogInformacion1.setSize(200,100);  dialogInformacion1.setResizable(false);  dialogInformacion1.setLocation(250,250);  dialogInformacion1.add(informacion1);  //gestion cuadro de dialogo2  dialogInformacion2.setLayout(new FlowLayout());  dialogInformacion2.setSize(250,100);  dialogInformacion2.setResizable(false);  dialogInformacion2.setLocation(250,250);  dialogInformacion2.add(informacion2);  //llenar el choice con el codigo  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con2 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen2 = con2.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs2 = sen2.executeQuery("SELECT CifProveedor FROM Proveedores");  //Recorrer los registros obtenidos  while(rs2.next())  {  String dato=rs2.getString("CifProveedor");  chCifProveedor .addItem(dato);  }  //Cerrar los objetos creados  rs2.close();  sen2.close();  con2.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  public void actionPerformed(ActionEvent ap)  {  if(cerrar.equals(ap.getSource()))  {  this.dispose();  }  if(mostrar.equals(ap.getSource()))  {  valor=(chCifProveedor.getSelectedItem());  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  ResultSet rs1 = sen1.executeQuery("SELECT \* FROM Proveedores WHERE CifProveedor ='"+valor+"'");  while (rs1.next())  {  // Obtener el campo IdArticulo del registro activo  CifProveedor = rs1.getString("CifProveedor");//getString porque es un varchar en la BD  // Obtener el campo DescripcionArticulo del registro activo  NombreProveedor = rs1.getString("NombreProveedor");  // Obtener el campo direccion del registro activo  DireccionProveedor = rs1.getString("DireccionProveedor");  // Obtener el campo ciudad del registro activo  CiudadProveedor = rs1.getString("CiudadProveedor");  // Obtener el campo PrecioCompraArticulo del registro activo  CpProveedor = rs1.getInt("CpProveedor");  // Obtener el campo PrecioVentaArticulo del registro activo  TelefonoProveedor = rs1.getInt("TelefonoProveedor");  }  //Cerrar los objetos creados  rs1.close();//fue el ultimo en abrirse  sen1.close();  con1.close();  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  txtCifProveedor.setText(CifProveedor);  txtCifProveedor.setEnabled(false);  txtNombreProveedor.setText(NombreProveedor);  txtDireccionProveedor.setText(DireccionProveedor);  txtCiudadProveedor.setText(CiudadProveedor);  txtCpProveedor.setText(Integer.toString(CpProveedor));  txtTelefonoProveedor.setText(Integer.toString(TelefonoProveedor));  modificar.setEnabled(true);  }  if(modificar.equals(ap.getSource()))  {  CifProveedor = (txtCifProveedor.getText());  NombreProveedor= (txtNombreProveedor.getText());  DireccionProveedor = (txtDireccionProveedor.getText());  CiudadProveedor = (txtCiudadProveedor.getText());  if((txtCpProveedor.getText().isEmpty()))  {  }  else  {  CpProveedor = Integer.parseInt(txtCpProveedor.getText());  }  if((txtTelefonoProveedor.getText().isEmpty()))  {  }  else  {  TelefonoProveedor = Integer.parseInt(txtTelefonoProveedor.getText());  }  if ((CifProveedor.length()>9)||(txtCifProveedor.getText().isEmpty())||(NombreProveedor.length()>45)||(txtNombreProveedor.getText().isEmpty())||(DireccionProveedor.length()>45)||(txtDireccionProveedor.getText().isEmpty())||(CiudadProveedor.length()>45)||(txtCiudadProveedor.getText().isEmpty())||(CpProveedor>99999)||(CpProveedor<0)||(txtCpProveedor.getText().isEmpty())||(TelefonoProveedor>999999999)||(TelefonoProveedor<0)||(txtTelefonoProveedor.getText().isEmpty()))  {  dialogInformacion1.setVisible(true);  }  else  {  try  {  Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  }  catch(ClassNotFoundException cnfe)  {  System.out.println("Error:"+cnfe.toString());  }  try  {  //Establecer la conexión con la BD practica\_hotelClinic  Connection con1 = DriverManager.getConnection("jdbc:odbc:hotelClinic","","");  //Crear una sentencia  Statement sen1 = con1.createStatement();  //Crear un objeto ResultSet para guardar lo obtenido  //y realizar la sentencia SQL  sen1.executeUpdate("UPDATE Proveedores SET NombreProveedor='"+NombreProveedor+"',DireccionProveedor='"+DireccionProveedor+"',CiudadProveedor='"+CiudadProveedor+"',CpProveedor='"+CpProveedor+"',TelefonoProveedor='"+TelefonoProveedor+"' WHERE CifProveedor='"+CifProveedor+"'");  //Cerrar los objetos creados  sen1.close();  con1.close();  dialogInformacion2.setVisible(true);  }  catch(SQLException sqle)  {  System.out.println("Error:"+sqle.toString());  }  }  }  }  public void windowActivated(WindowEvent wa)  {  }  public void windowClosed(WindowEvent wcd)  {  }  public void windowClosing(WindowEvent wcg)  {  if (dialogInformacion1.equals(wcg.getSource()))  {  dialogInformacion1.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (dialogInformacion2.equals(wcg.getSource()))  {  dialogInformacion2.dispose();//oculta de dialog pulsando el simbolo de cierre del frame  }  if (this.equals(wcg.getSource()))  {  this.dispose();  }  }  public void windowDeactivated(WindowEvent wd)  {  }  public void windowDeiconified(WindowEvent wdf)  {  }  public void windowIconified(WindowEvent wi)  {  }  public void windowOpened(WindowEvent wo)  {  }  } |

|  |
| --- |
| **4. MANUEL DE USO E INSTALACIÓN. PRUEBAS DE FUNCIONAMIENTO** |

Al ejecutar el programa ,obtendremos la pantala principal llamada **PANEL DE ACCESO**, aqui indicaremos el nombre y contraseña de cada uno de los usuarios configurados en la base de datos.



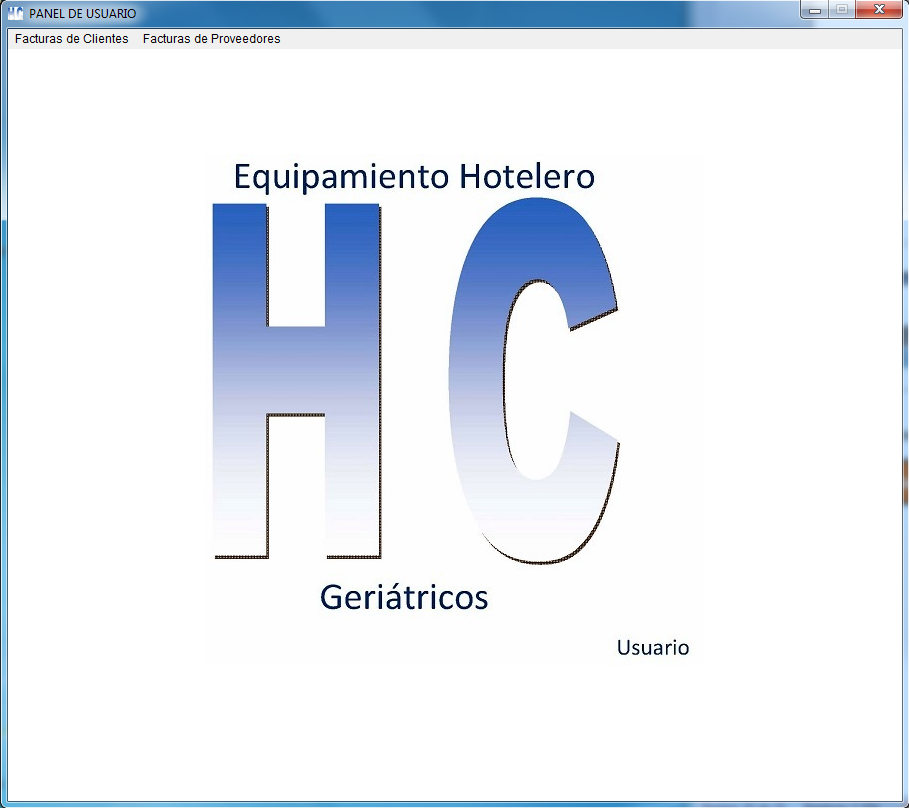
Si algunos de los datos solicitado no coinciden con los configurados en la base de datos, obtendremos el suguiente mensaje.



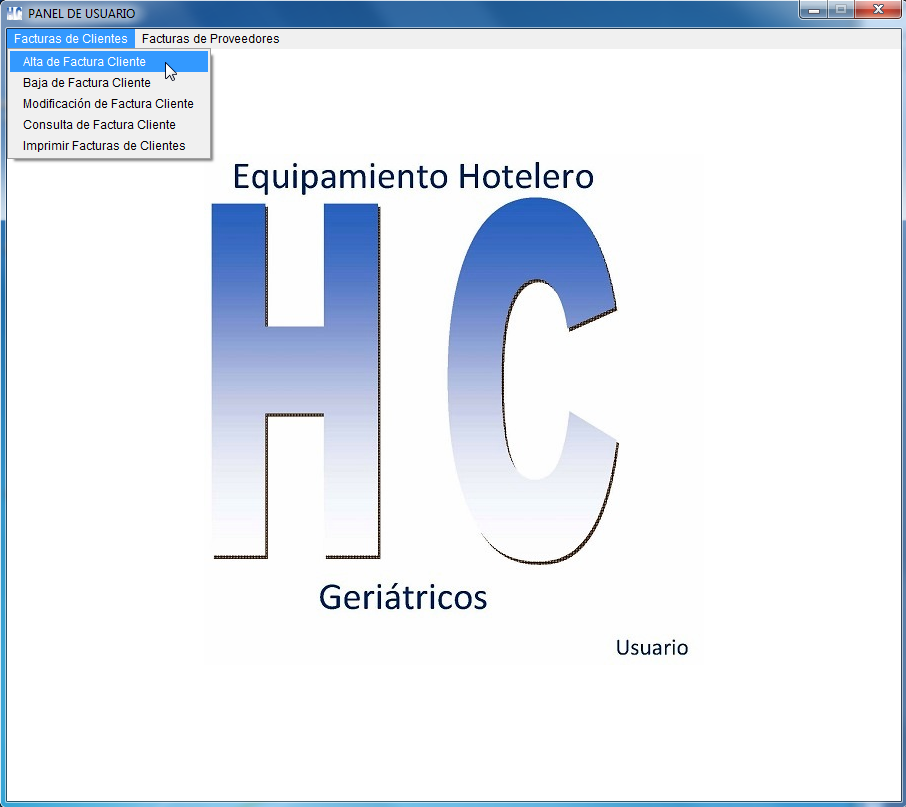
Siguiendo las indicaciones de la practica, para comprobar el buen funcionameinto del programa realizaremos las siguientes operaciones:

|  |
| --- |
| **Gestión del usuario** |

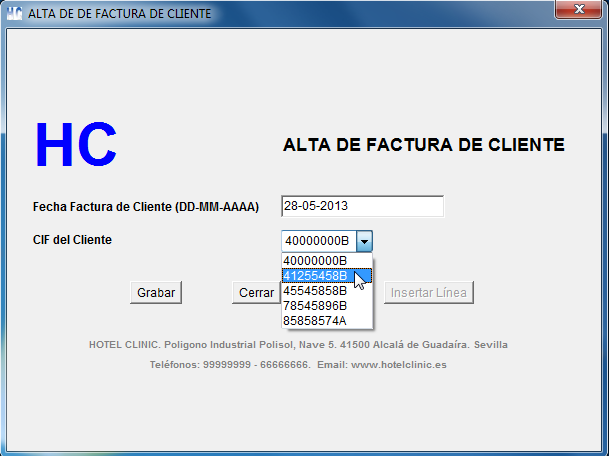
Una vez logeado el usuario, obtendremos la siguiente pantalla con acceso a **Facturas de Clientes** y a **Facturas de Proveedores**

****

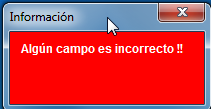
A continuación el usuario creará un registro en cada una de las tablas donde tenga acceso. Comenzaremos por un registro en las **facturas de Clientes** accediendo a través de la opción de **Alta de Facturas Clientes** contenidas en el menu.



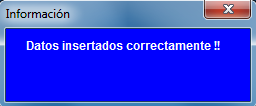
En el panel de alta de cliente, rellenaremos los datos correspondientes



Si algún campo no corresponde con las limitaciones configuradas en la base de datos, obtendremos el siguiente mensaje



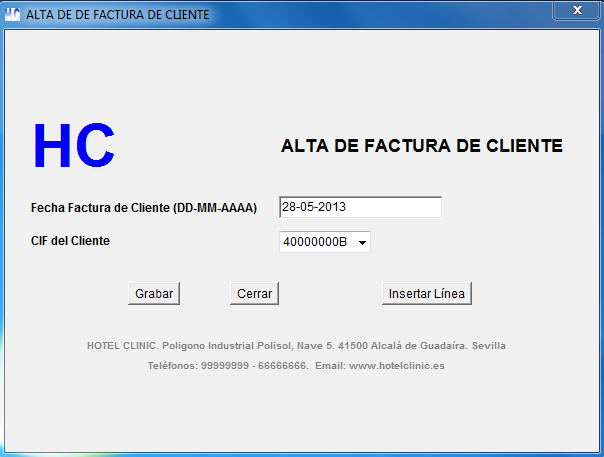
Si todos los datos son sorrectos obtendremos el siguiente mensaje.



Una vez registrada la factura, observamos que esta ha sido grabada correctamente en la base de datos. Observar que aún no se ha ingresado ninguna línea de factura.



Ahora vamos a insertar la primera línea de factura.



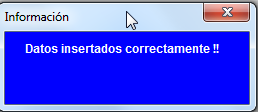
Seleccionamos algún artículo de la lista desplegable.



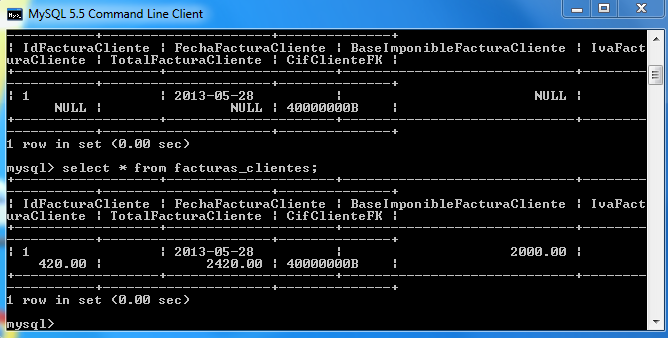
Automáticamente se cargan los datos del artículo. Indicamos las unidades del producto y pulsamos en Insertar.



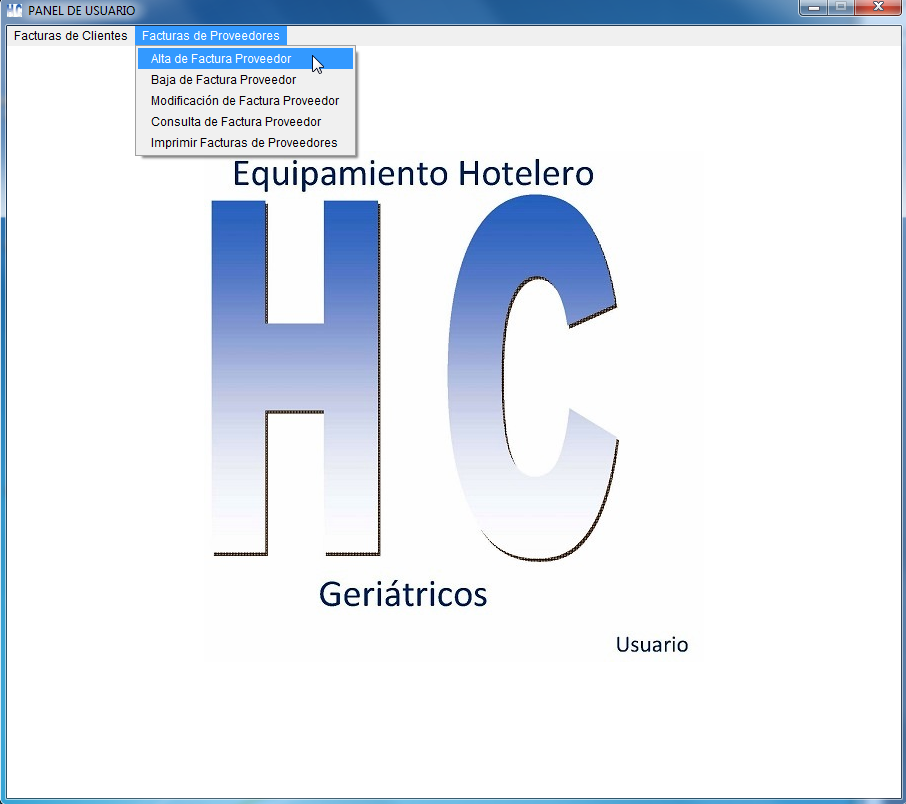
El sistema nos muestra la confirmación de uqe los datos han sido insertados.



Se observa que ha se ha cargado la línea en la factura.



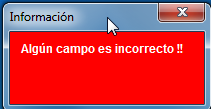
Igualmente procederemos con las **Facturas de Proveedores**, entrando en el panel de gestión através de la opción **Alta de Factura Proveedor.**



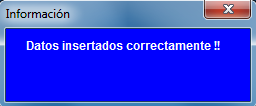
En el panel de alta de cliente, rellenaremos los datos correspondientes.

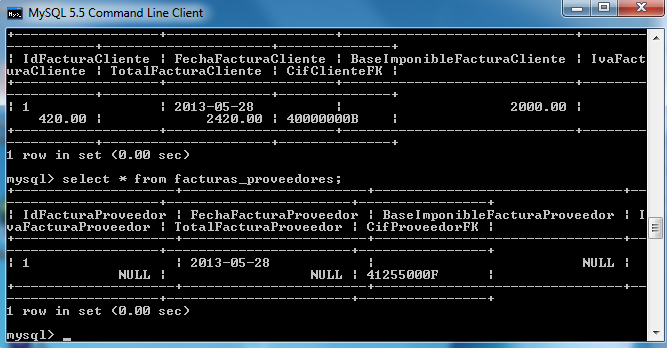


Si algún campo no corresponde con las limitaciones configuradas en la base de datos, obtendremos el siguiente mensaje

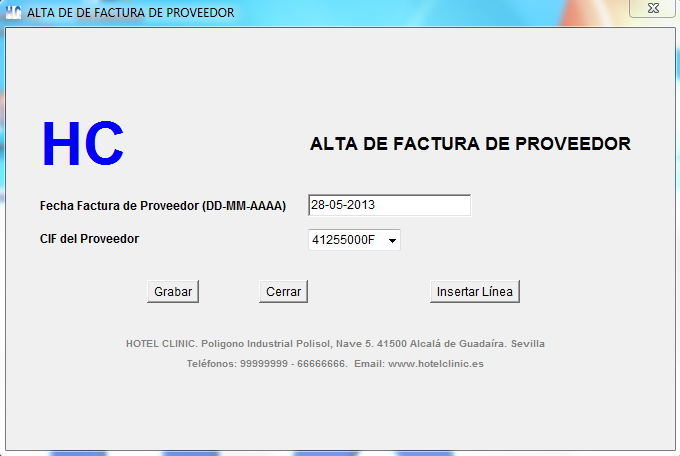


Si todos los datos son sorrectos obtendremos el siguiente mensaje.



Una vez registrada la factura, observamos que esta ha sido grabada correctamente en la base de datos. Observar que aún no se ha ingresado ninguna línea de factura. 

Ahora vamos a insertar la primera línea de factura.



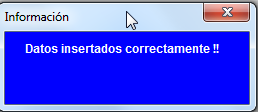
Seleccionamos algún artículo de la lista desplegable.



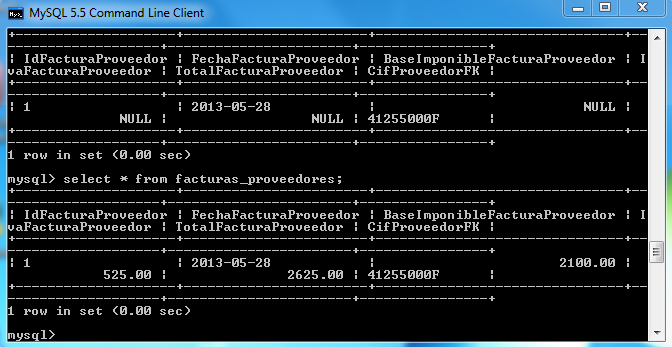
Automáticamente se cargan los datos del artículo. Indicamos las unidades del producto y pulsamos en Insertar.



El sistema nos muestra la confirmación de uqe los datos han sido insertados.

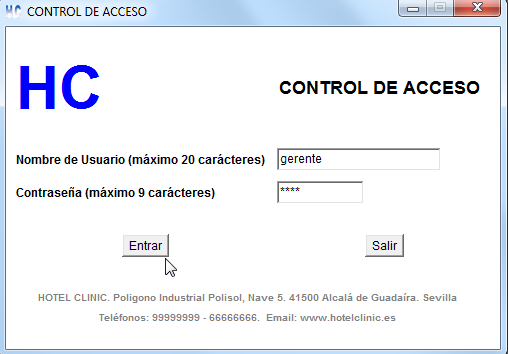


Se observa que ha se ha cargado la línea en la factura.

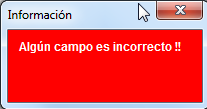


|  |
| --- |
| **Gestión del gerente** |

A continuación entraremos en el sistema como gerente.



De igual forma al usuario, si no introducimos alguno de los campos correctamente, obtendremos el siguiente mensaje, y no podremos acceder al sistema.



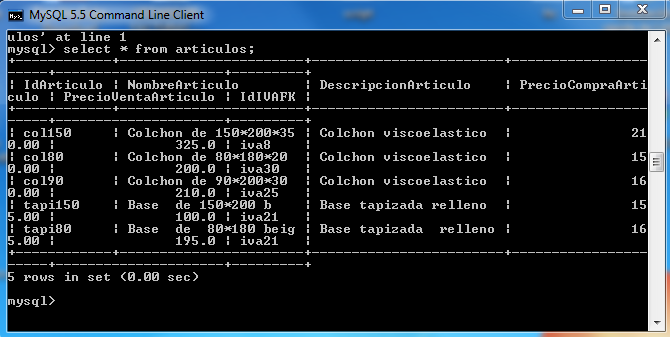
Para comprobar el perfecto funcionamiento de la aplicación, realizaremos algunas gestiones en las tablas a las que el gerente tenga acceso. Además realizaremos alguna modificación en los datos, e imprimiremos la información en un archivo pdf.

Comenzamos realizadno las consultas por la tabla de clientes accediendo a través de la opción correspondiente enel menú.

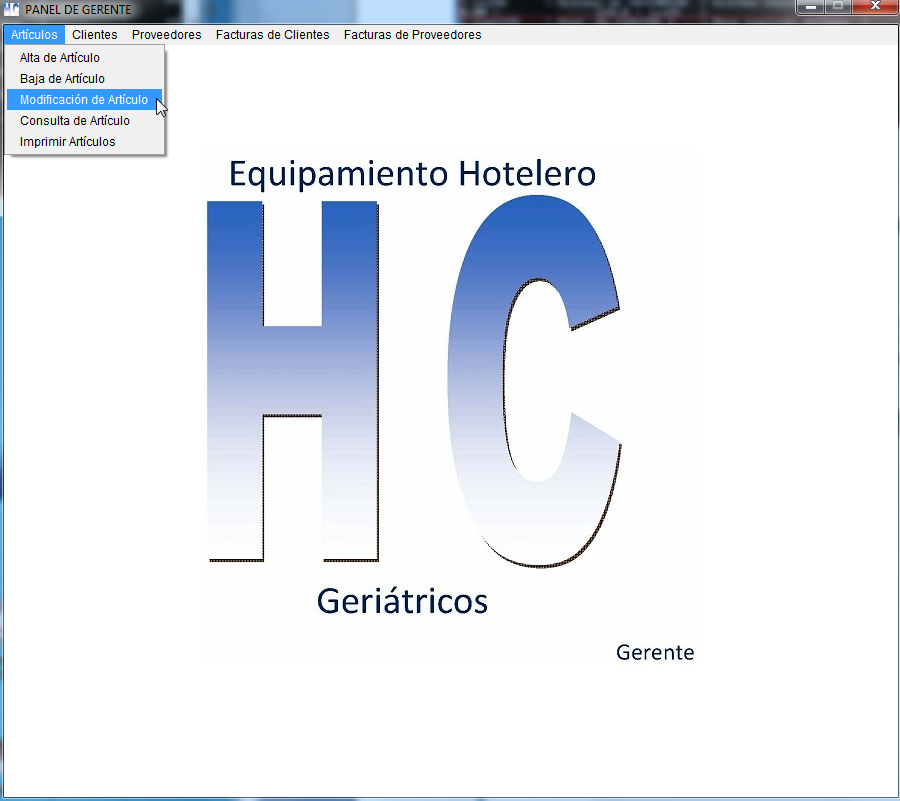


En la lista desplegable de los códigos de artículos disponibles, realizamos una consulta de los datos del artículo con el código **col90**¸ y pùlsamos **Mostrar.**

****

Observamos que los datos de la consulta se corresponden con la base de datos.****

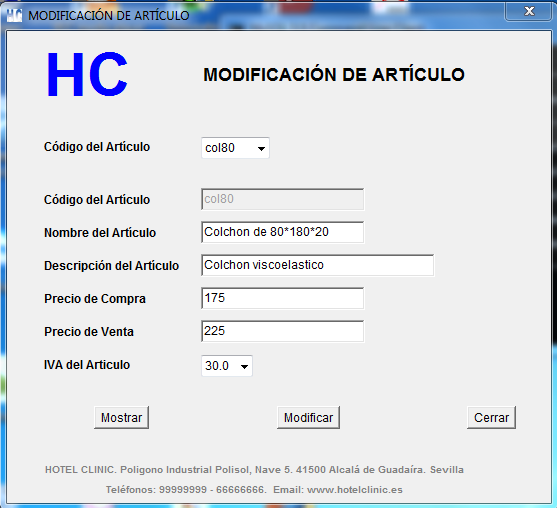
Ahora realizamos una modificación en alguno de los registros, por ejemplo en el precio de compra y el de venta del artículo col80.



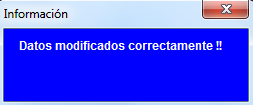
Seleccionamos el código de todos los disponibles en el desplegable y pulsamos **Mostrar**. Obtendremos la sigueinte pantalla.



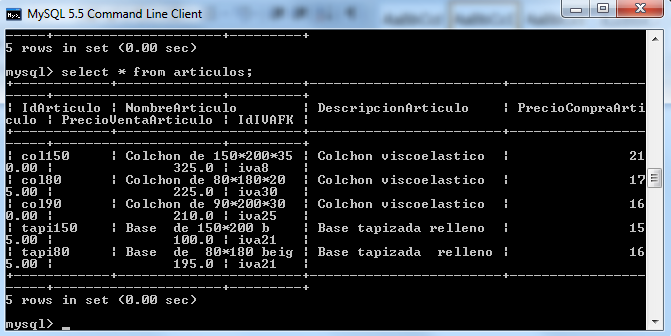
Pulsamos en **Modificar**



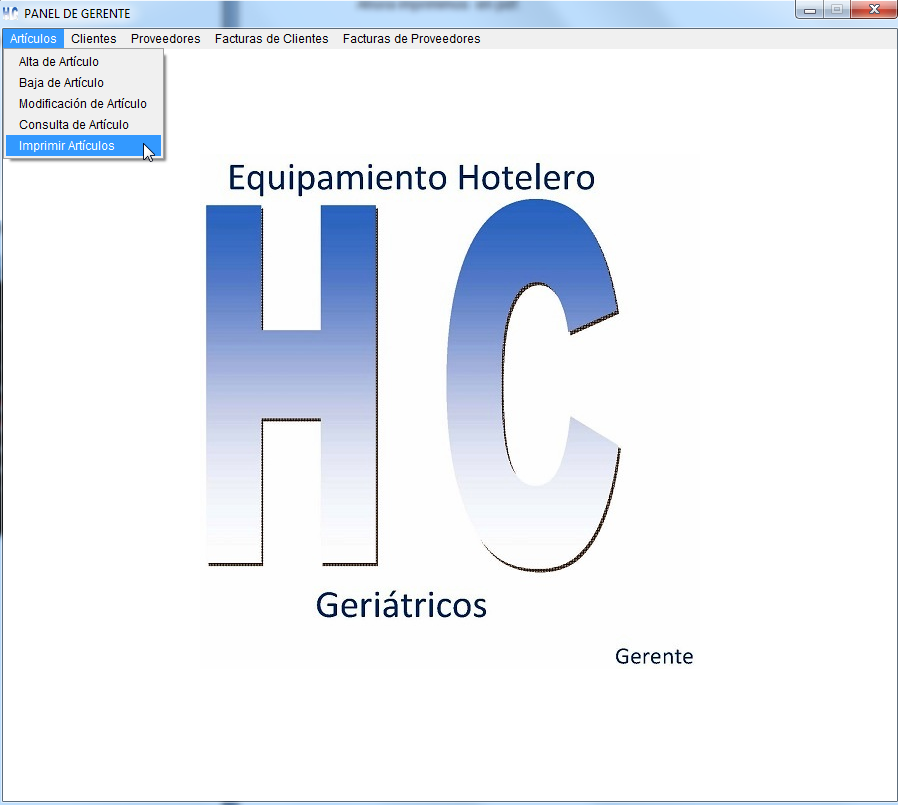
Confirmación de que la modificación se realizado correctamente.



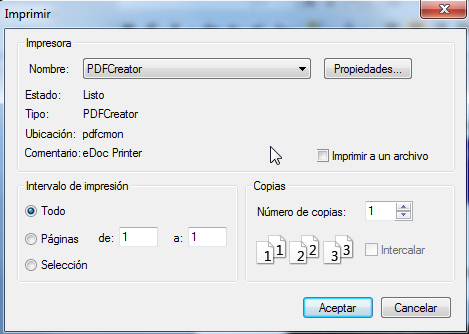
Observamos que los datos de han modificado en la base de datos



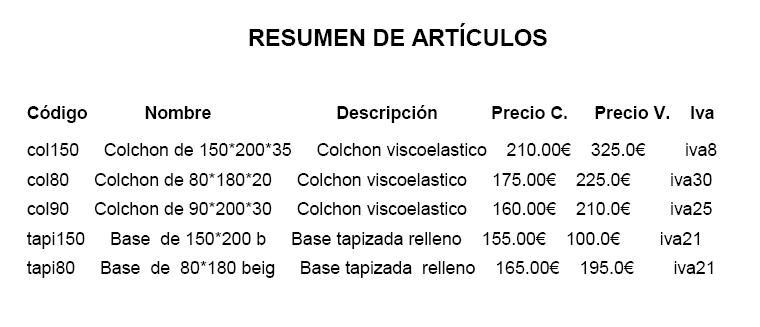
Ahora imprimimos en pdf.



Indicamos la impresora pdf.



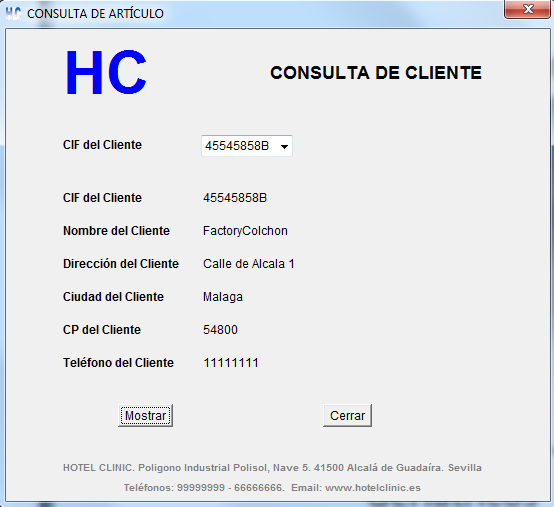
Obtenemos la siguiente impresión, donde observamos las modificaciones realizadas anteriormente.



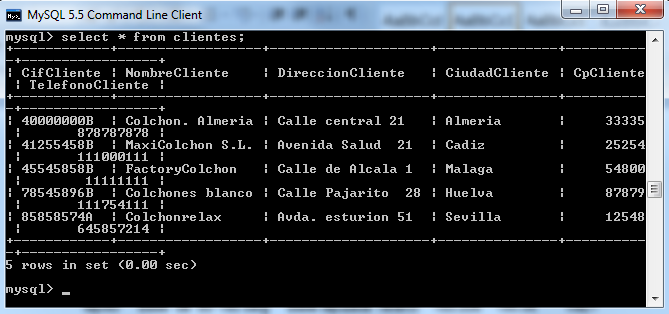
Seguidamente procederemos de igual forma con la tabla **Clientes**



Seleccionamos el CIF del cliente a consultar y pulsamos **Mostrar**



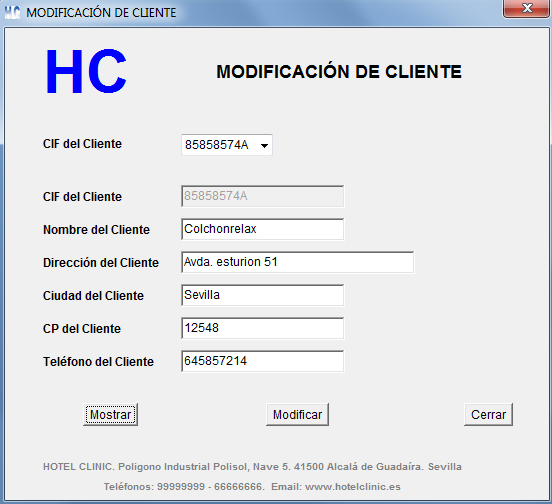
Observamos la correspondencia con la Base de datos.



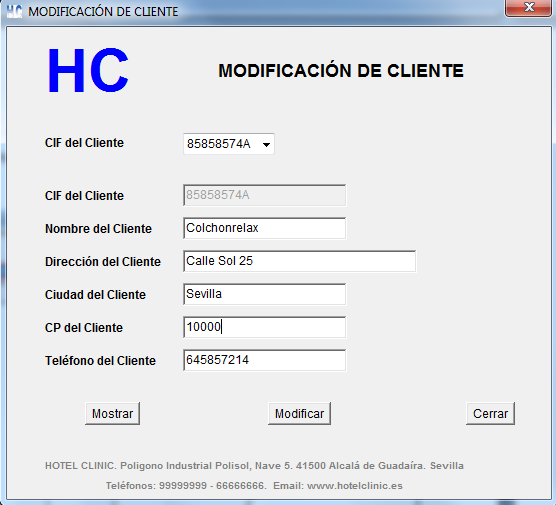
Ahora realizamos una modificación de algunos de los datos de alguno de los clientes.



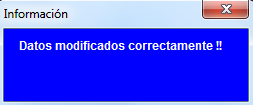
Una vez seleccionado el cif del cliente sobre el que deseamos modificar los datos, pulsamos mostrar.



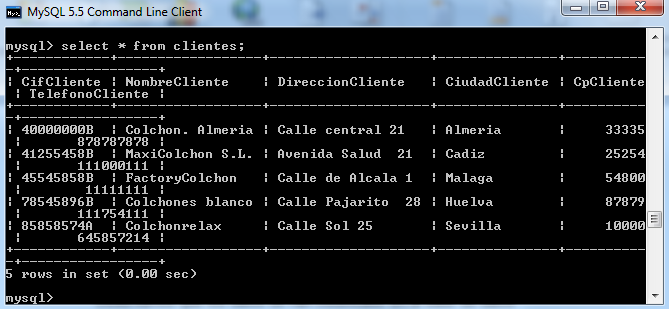
Modificaremos la direccion del cliente y el CP y pulsamos **Modificar.**



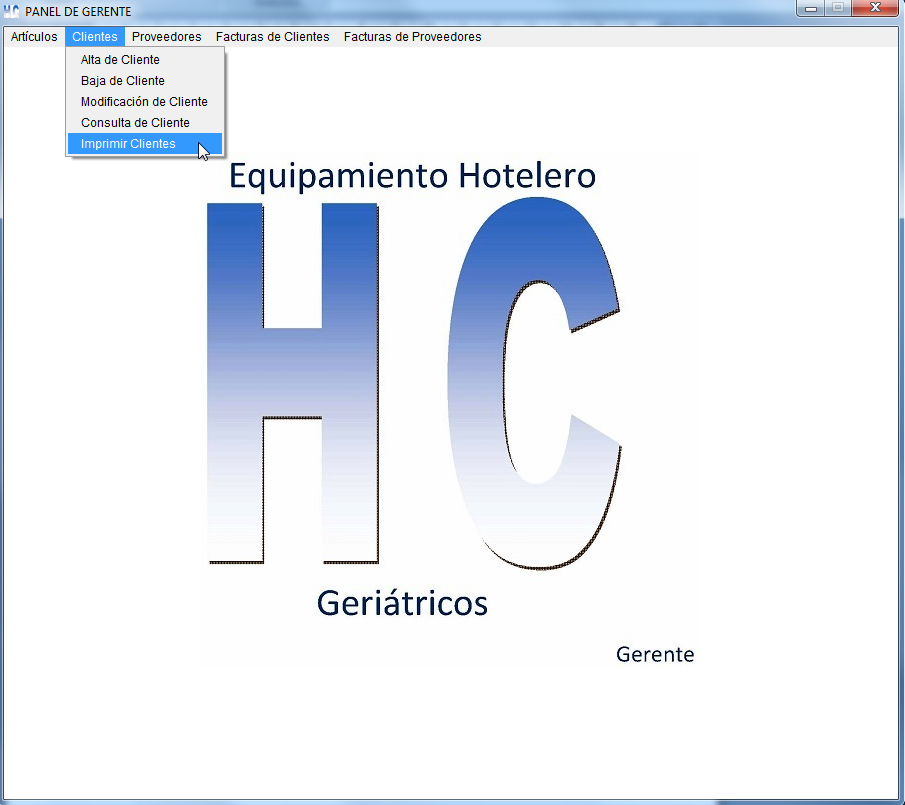
Confirmación de que la modificación se ha realizado correctamente.



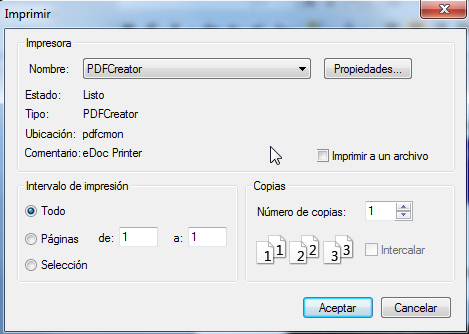
Observamos que los datos de han modificado en la base de datos.



Ahora imprimimos en pdf.



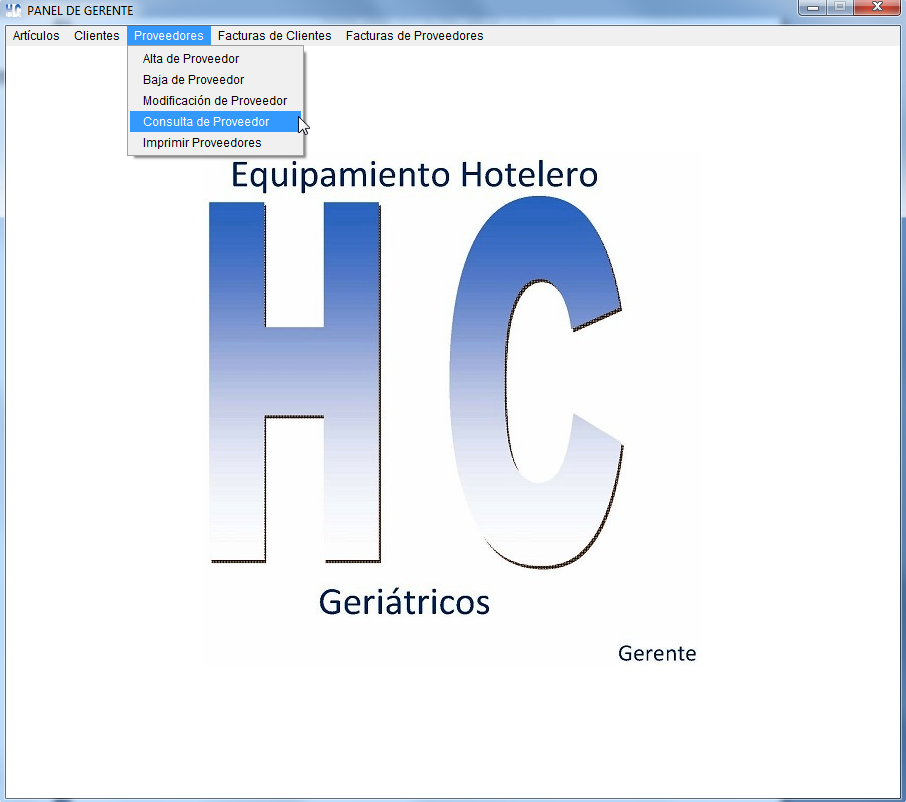
Indicamos la impresora pdf.



Obtenemos la siguiente impresión, donde observamos las modificaciones realizadas anteriormente.



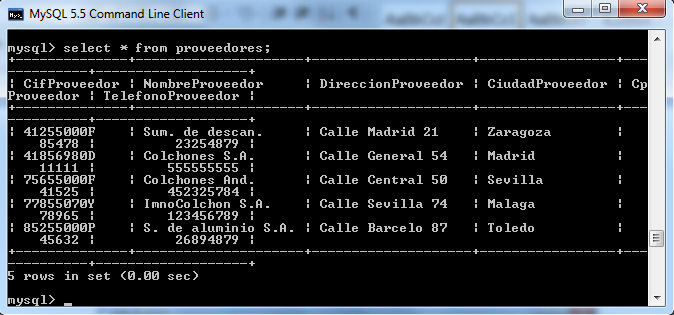
Seguidamente procederemos de igual forma con la tabla **Proveedores**

****

Seleccionamos el CIF del proveedor a consultar y pulsamos **Mostrar**

****

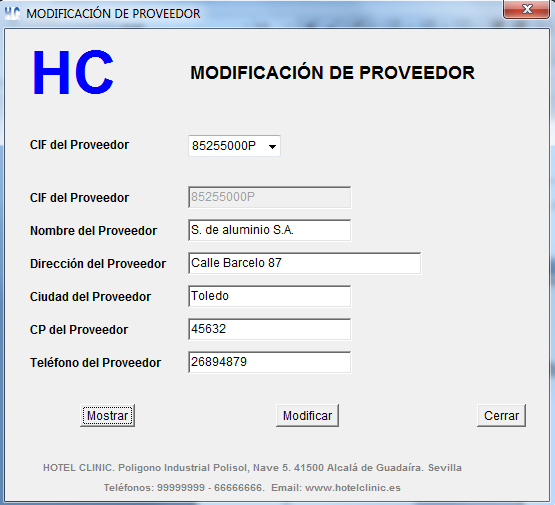
Observamos la correspondencia con la Base de datos.



Ahora realizamos una modificación de algunos de los datos de alguno de los Provedores.



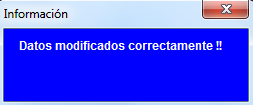
Una vez seleccionado el cif del cliente sobre el que deseamos modificar los datos, pulsamos **Mostrar.**



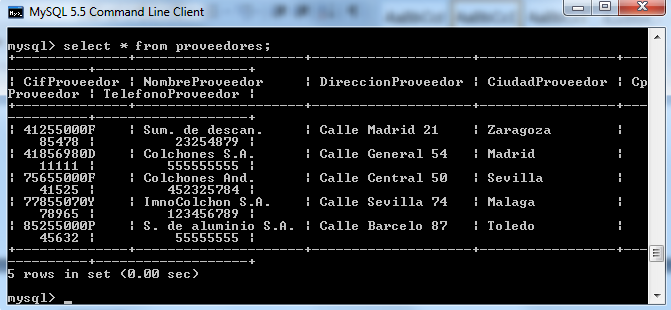
Modificaremos el teléfono del Proveedor y pulsamos **Modificar.**



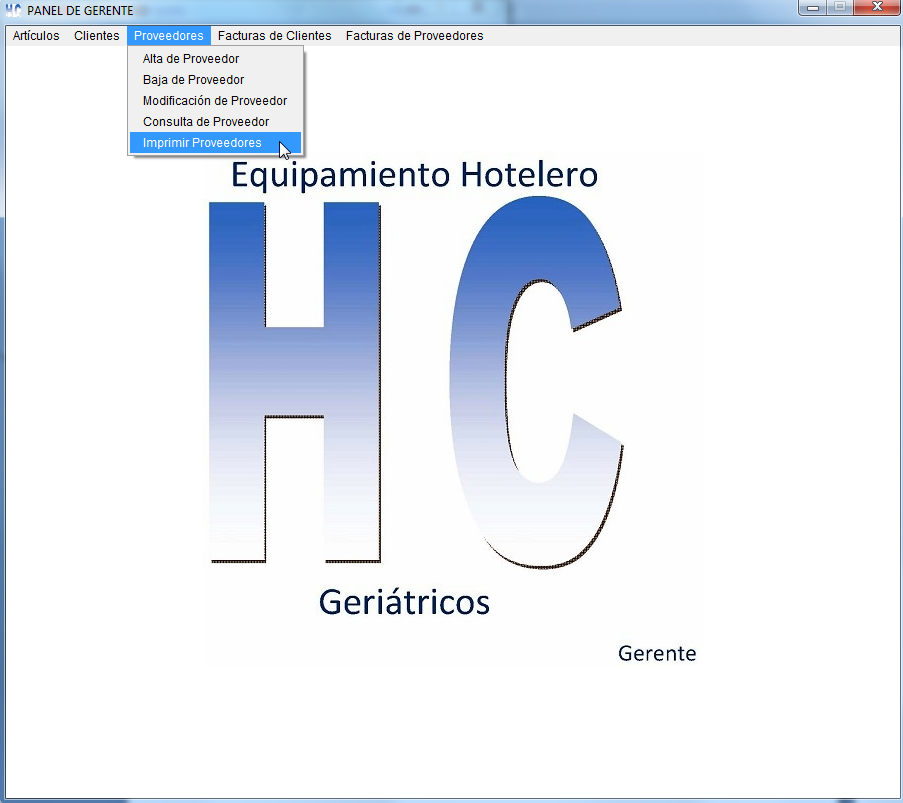
Confirmación de que la modificación se ha realizado correctamente.



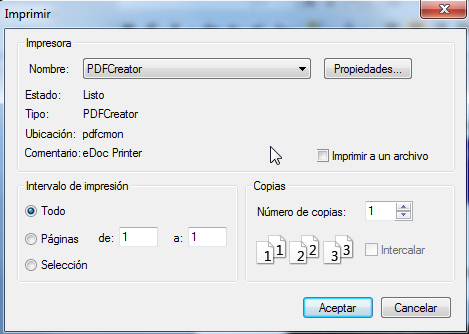
Observamos que los datos de han modificado en la base de datos.



Ahora imprimimos en pdf.



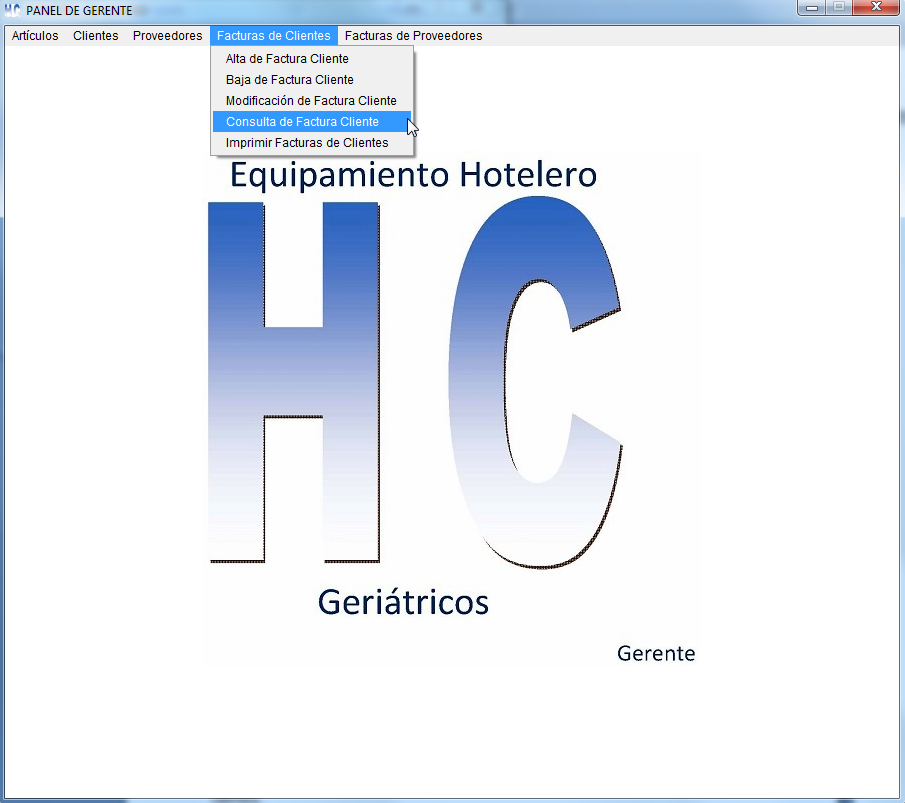
Indicamos la impresora pdf.



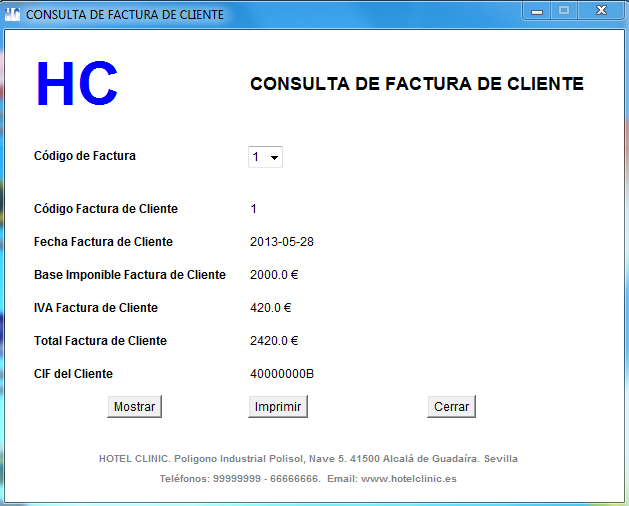
Obtenemos la siguiente impresión, donde observamos las modificaciones realizadas anteriormente.



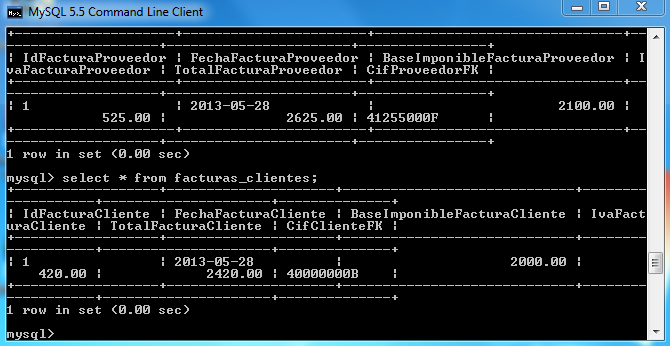
Seguidamente procederemos de igual forma con la tabla **Facturas Clientes**

****

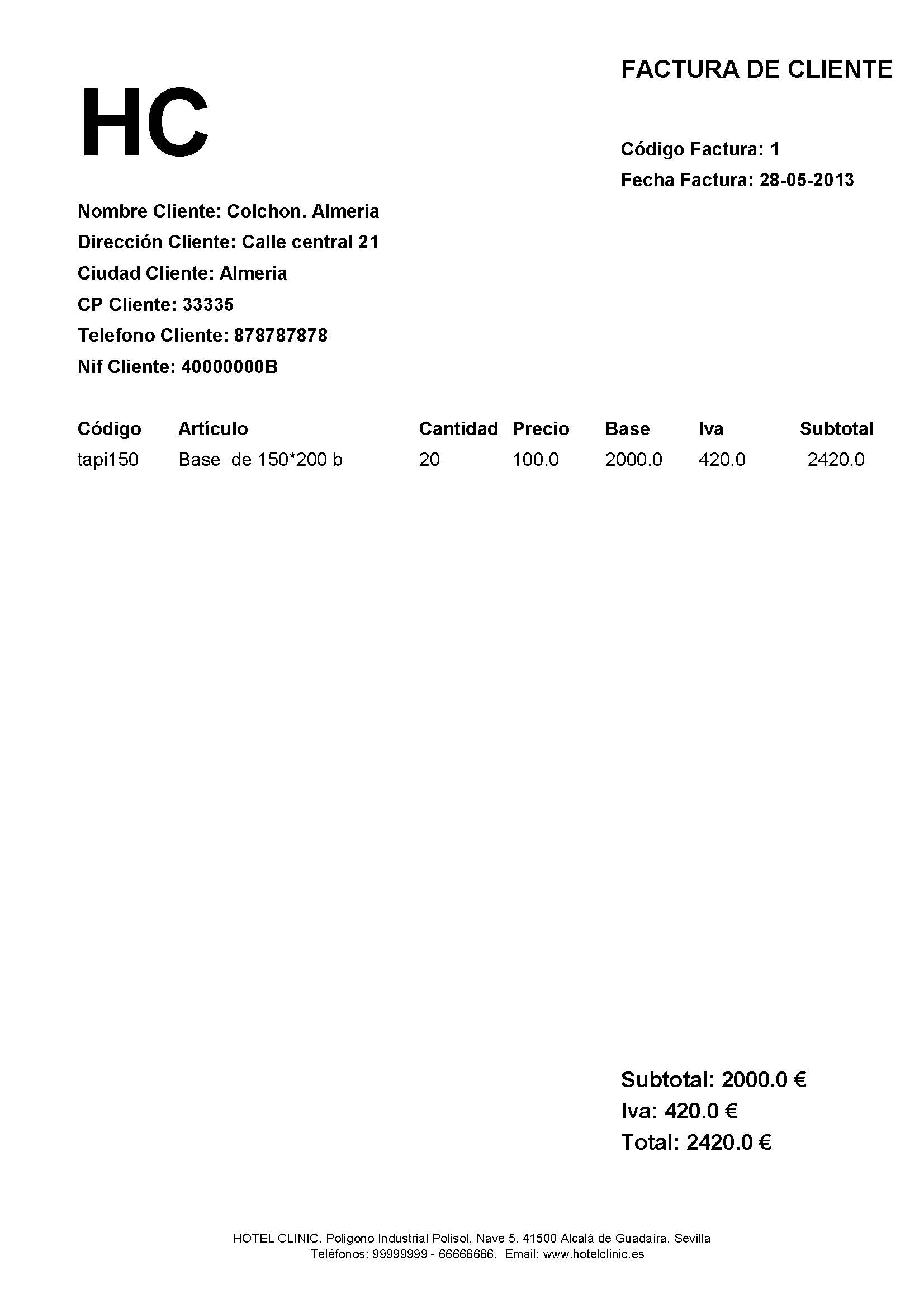
Seleccionamos el codigo de la factura a consultar y pulsamos **Mostrar**

****

Observamos la correspondencia con la Base de datos.

****

Ahora imprimimos una factura de cliente, pulsando en el botón imprimiar de la página anterior. Obtenemos lo siguiente:



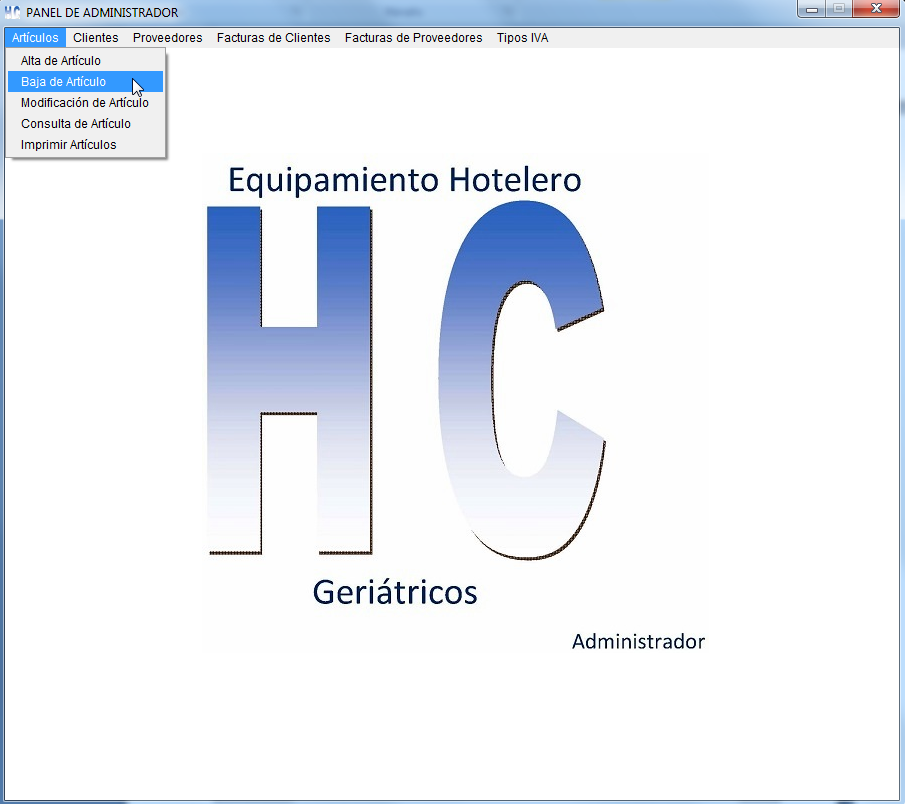
|  |
| --- |
| **Gestión del administrador** |

Por último realizaremos algunas gestiones como administrador. Vamos a borrar un registro de algunas tablas

Para acceder al panel del usuario, indicaremos el nombre y su contraseña.



Accedemos al panel del administrador, y comenzamos eliminando un registro de la tabla Artículos, pulsando en Baja de Artículos del Menu Artículos.



Para realizar la baja de un artículo, indicaremos el campo de busqueda del artículo y el valor del campo del artículo que deseamos eliminar. Pulsaremos **Dar Baja**.



Confirmación de la baja del artículo.

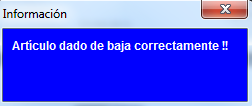


Imagen de la base de datos antes de dar de baja el artículo

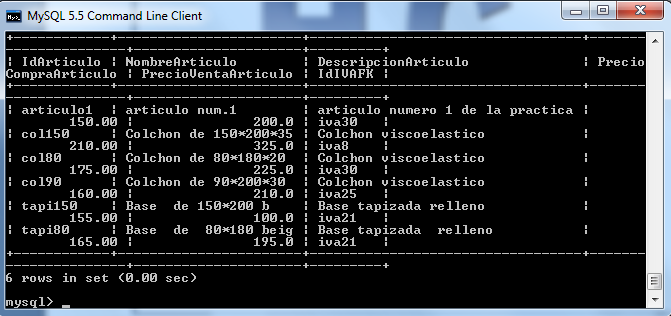
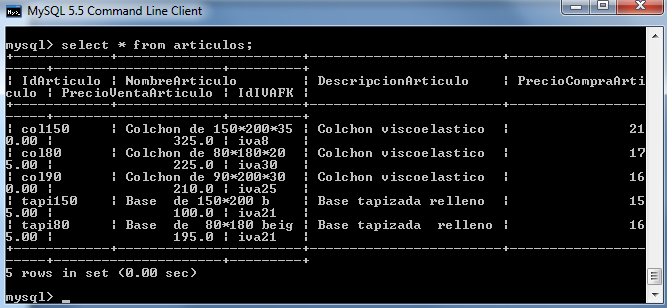


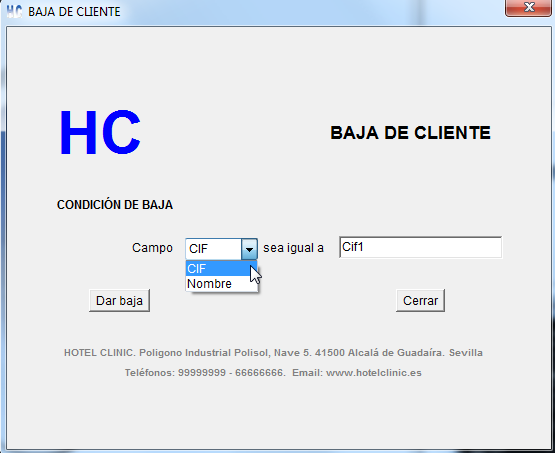
Imagen de la base de datos trás la eliminación del artículo, donde se observa que el artículo se ha eliminado.



Ahora eliminaremos un registro de la tabla Clientes pulsando en Baja de Cliente del Menu Clientes.



Para realizar la baja de un cliente, indicaremos el campo de busqueda del cliente y el valor del campo del cliente que deseamos eliminar. Pulsaremos **Dar Baja**.



Confirmación de la baja del artículo.

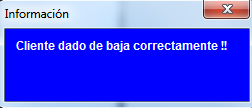


Imagen de la base de datos antes de dar de baja el cliente.

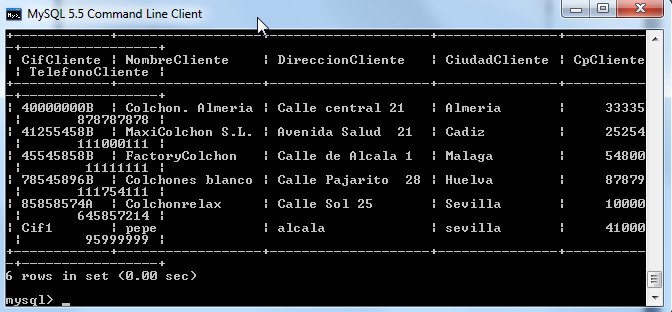
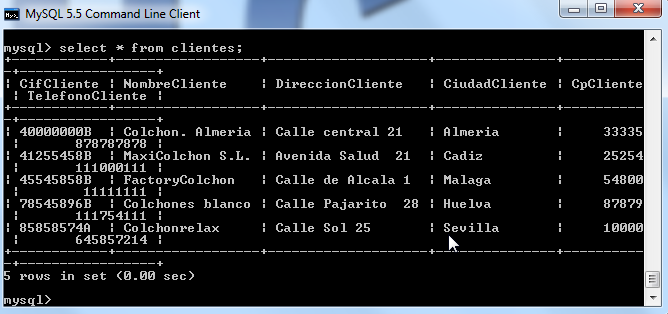


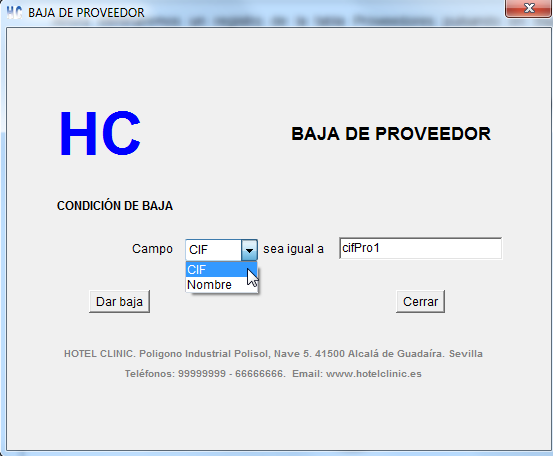
Imagen de la base de datos trás la eliminación del cliente, donde se observa que el cliente se ha eliminado.



Ahora eliminaremos un registro de la tabla Proveedores pulsando en Baja de Proveedor del Menu Proveedores.



Para realizar la baja de un proveedor, indicaremos el campo de busqueda del proveedor y el valor del campo del proveedor que deseamos eliminar. Pulsaremos **Dar Baja**.



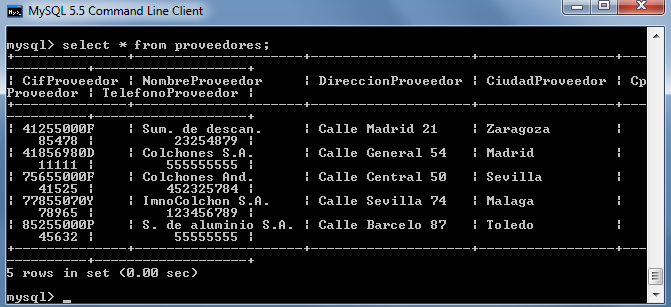
Confirmación de la baja del artículo.



Imagen de la base de datos antes de dar de baja el proveedor.



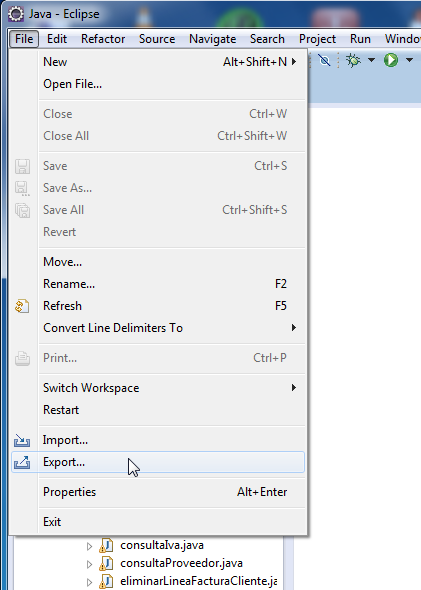
Imagen de la base de datos trás la eliminación del proveedor, donde se observa que el cliente se ha eliminado.



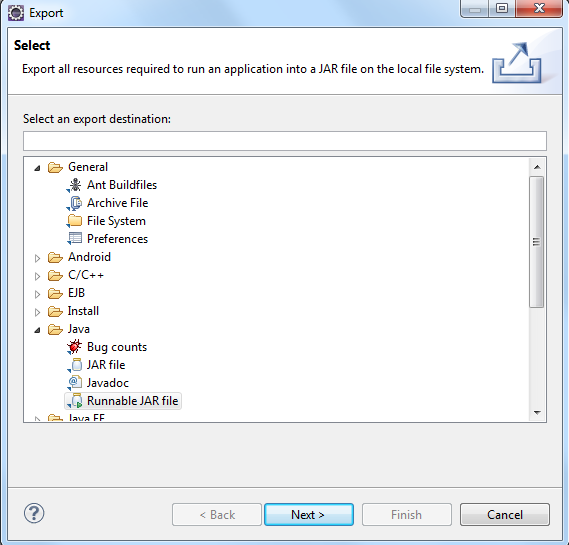
|  |
| --- |
| **5. CREACIÓN DE UN ARCHIVO EJECUTABLE Y SU INSTALACIÓN** |

|  |
| --- |
| **5.1. ARCHIVO EJECUTABLE** |

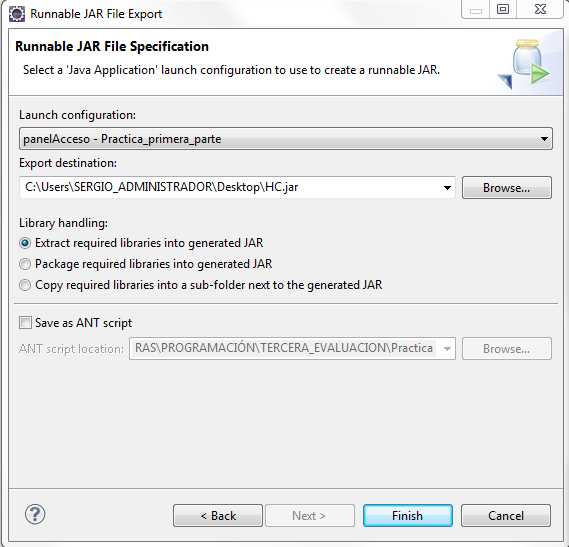
La primera actuación será generar un archivo .jar de nuestro proyecto desde Eclipse. Para ello seleccionamos en la barra de menu la pestaña **File** y en el desplegable seleccionamos **Export.**



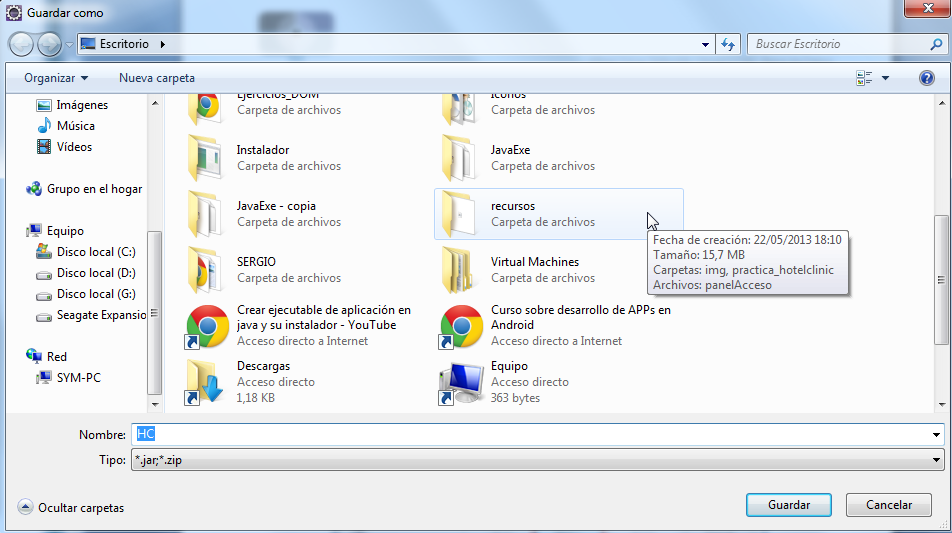
En la siguiente ventana seleccionamos **Runnable JAR file**. Pulsamos **Next**



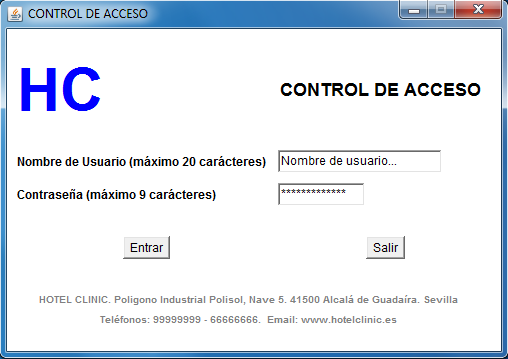
Indicamos que archivo arrancará el sistema, en nuestro caso es **panelAcceso.**



Seleccionamos la ruta donde se guardará el archivo generado. En nuestro caso lo guardamos en una carpeta llamada recursos. Aquí también guardaremos los recursos que necesite el sistema como bases de datos, imágenes, iconos, etc., que después serán instalados.



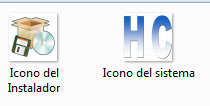
Mediante este archivo ya podemos ejecutar nuestra aplicación sin necesidad de tener instalado eclipse en nuestro equipo. Pero iremos un poco mas lejos, y generaremos un instalador para instalar la aplicación en cualquier equipo que disponga de máquina virtual.





El siguiente paso consiste en generar un par de icono, uno para el sistema y otro para el instalador. Para ello podemos utilizar algún programa de los existentes. En mi caso suelo utilizar alguno de los siguientes:

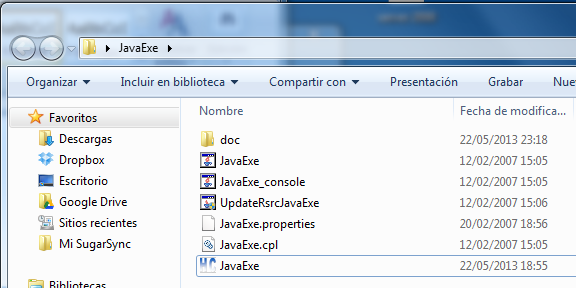
* Axialis IconWorkshop 6.8.0.0 <http://www.axialis.com/download/>
* Format Factory <http://www.pcfreetime.com/es/>



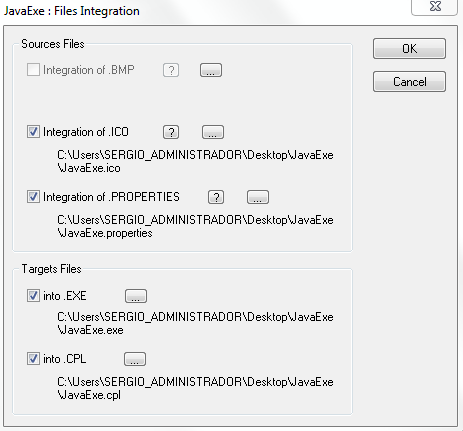
A continuación crearemos el ejecutable mediante la aplicación JavaExe. Podemos descargarla en <http://devwizard.free.fr/html/en/JavaExe.html>

Una vez descargada la carpeta realizaremos una copia de esta dado que solo tiene un uso.

Seguidamente copiamos el icono que creamos anteriormente para el sistema dentro de esta carpeta y le cambiamos el nombre por JavaExe.



Ejecutamos el archivo contenido dentro de esta carpeta llamado **UpdateRsrcJavaExe** y pulsamos **OK**

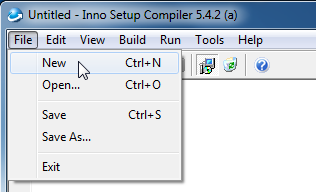


Ahora copiamos todos los archivos **.java** y **.class** de nuestro proyecto en esta carpeta, y cambiamos el nombre del archivo JavaExe.exe por el nombre de la clase que arrancará el sistema (main), en nuestro caso es panelAcceso.

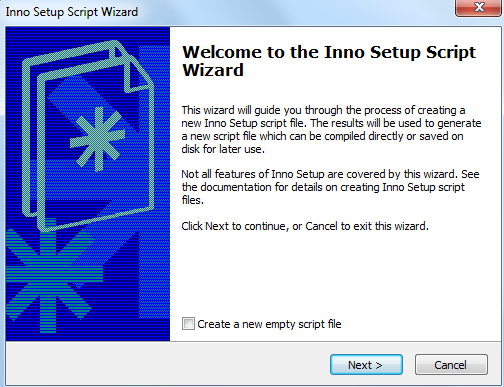


A continuación crearemos el instalador del sistema. Para ello utilizaremos una aplicación llamada **Inno setup Compiler.** Podemos descargarla en <http://www.jrsoftware.org/isinfo.php>

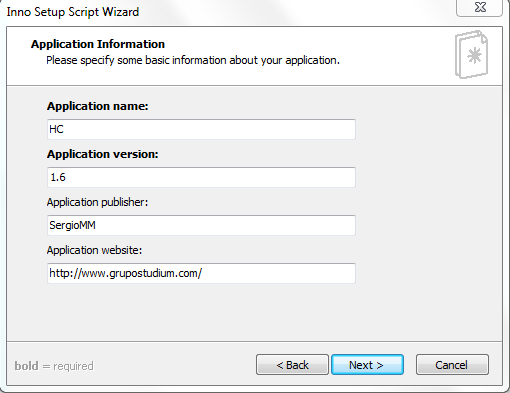
Una vez instalada la aplicación anterior, accedemos a ella y seleccionamos **new** en la etiqueta **FIle** del menú



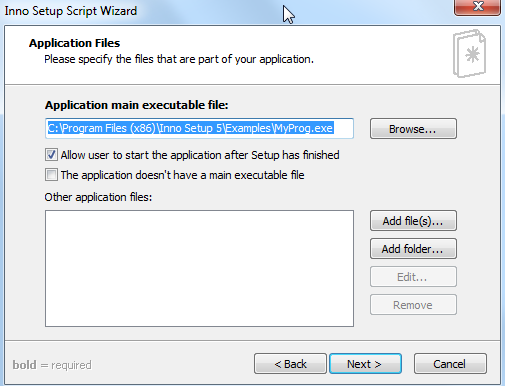
Pulsamos **Next.**

****

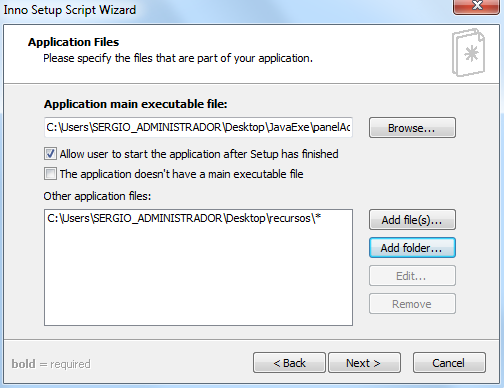
Establecemos el nombre de nuestro programa, la versión de la aplicación, el nombre del autor de la aplicación y una página web del autor. Pulsamos Next.



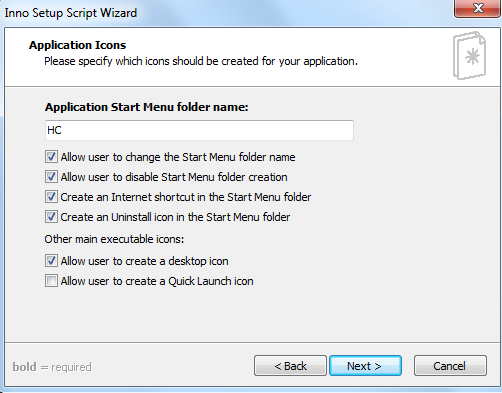
Indicamos donde se instala el porgrama por defecto en Windows. Pulsar Next.



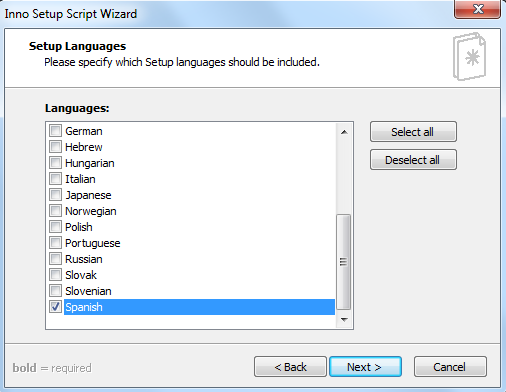
Buscamos el programa compilado que se encuentra en la carpeta JavaExe con el nombre panelAcceso y la carpeta donde se encuentran todos los recursos.



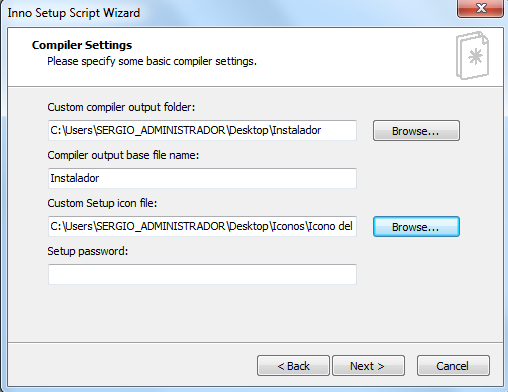
Marcamos las casillas siguientes y pulsamos Next



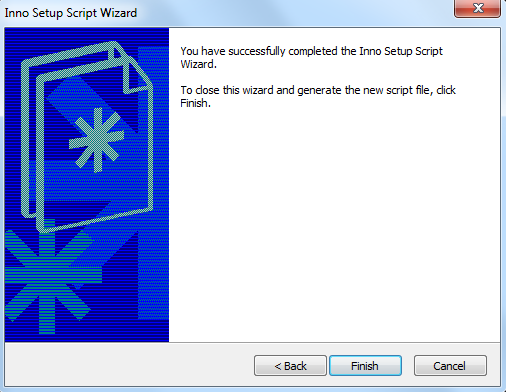
Seleccionamos los idiomas para la instalación y pulsamos Next.



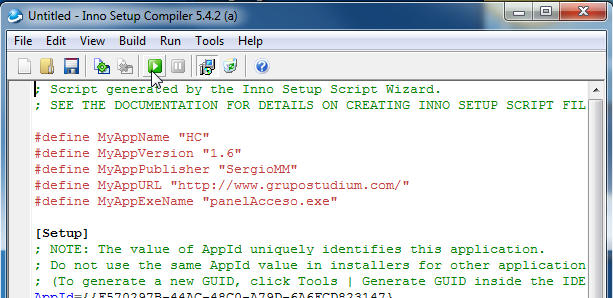
Indicaremos donde guardaremos el instalador, en nuestro caso será en una carpeta llamada Instalador que se ha creado en el escritorio. También elegimos un nombre para el instalador y seleccionaremos un icono para el mismo. Pulsamos Next.

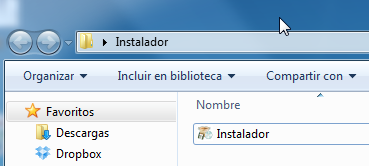


Pulsamos en Finish.



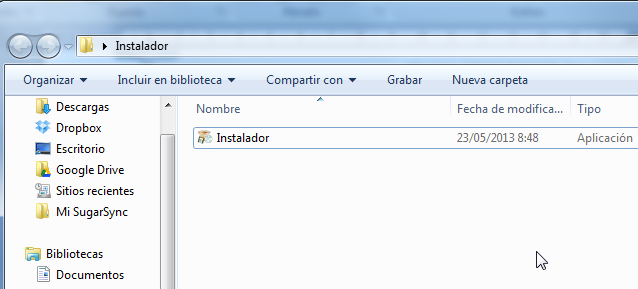
Pulsamos en Run y comenzará a crear el instalador.



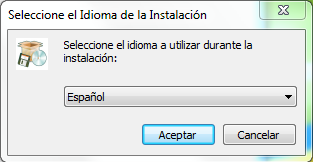


|  |
| --- |
| **5.2. INSTALACIÓN** |

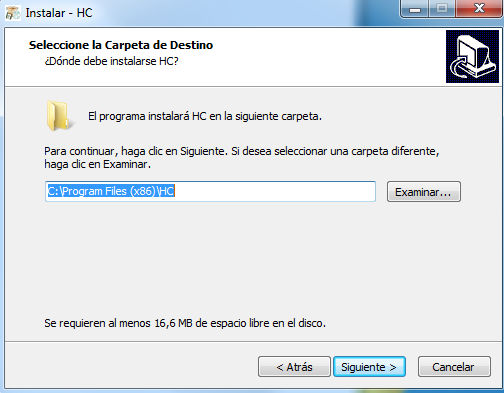
Una vez creado el instalador, vamos a instalar la aplicación. Vamos a la carpeta creada en el escritorio llamada Instalador y allí se debe haber creado el mismo.



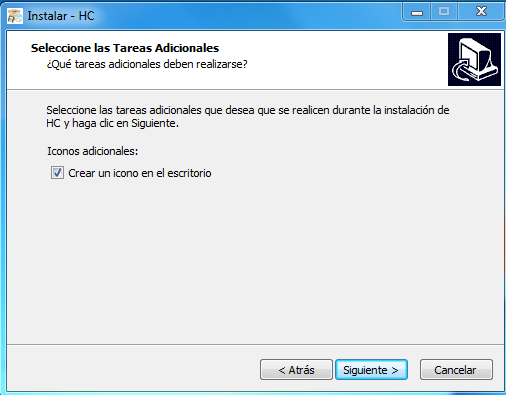
Procedemos a instalar la aplicación y seleccionamos el idioma de instalación.



Indicamos la ruta de instalación



Creamos un icono de acceso directo en el escritorio.



Una vez instalada la aplicación podemos apreciar como se ha creado un icono de acceso directo en el escritorio y una carpeta en el menú inicio. En el menu inicio también se ha creado un enlace para la deinstalación de la aplicación y un enlace a la página web que configuramos en la instalación (www.grupostudium.com).

