

Memoria

Práctica 1

PC Parts

GIITIN01-2-006 Comunicación Persona-Máquina

Alumno:

UO282978

Juan José Vázquez Prieto

Bloque	Incluido
1	Sí
2	Sí
3	Sí
4	Sí
5	Sí

Contenido

Introducción	3
Temática del proyecto	3
Guía de usuario básica	4
Subtítulo 1.....	¡Error! Marcador no definido.
Subtítulo 2.....	¡Error! Marcador no definido.
Bloques	5
Bloque 1. Ventana y eventos	5
1.1 Barra de menús con algunos menús y submenús	5
1.2 Algunos componentes en un <i>layout</i> diferente del default.....	5
1.3 Componentes modificados por eventos	6
1.4 Tratamiento de eventos de teclado	6
1.5 Modificación de componentes con un número variable de elementos.....	7
1.6 Elementos no vistos	9
Bloque 2. Varias ventanas	10
2.1 La ventana principal modifica componentes de otra ventana	10
2.2 Una ventana no principal modifica componentes de la principal	11
2.3 Una ventana no principal modifica componentes de otra ventana no principal	12
2.4 Una ventana inicial que no sea la principal	14
Bloque 3. Diálogos	15
3.1 Un diálogo usando JOptionPane	15
3.2 Un diálogo predefinido	15
3.3 Un diálogo creado por el usuario que pida información al usuario	15
Bloque 4. Interfaz en primer plano	22
4.1 Métodos set para dar información a la tarea.....	22
4.2 Métodos que envíen información de la tarea a la interfaz	22
4.3 Posibilidad de hacer un stop	22
Bloque 5. Gráficos.....	25
5.1 Clase hija de un componente del que se refine su método Paint	25
5.2 Métodos set para modificar lo que se pinta en la clase anterior	25
5.3 Utilización del método repaint.....	25
5.4 Elemento no visto en clase ni en apuntes.....	26

Introducción

En este informe se detallará tanto el código fuente, como el funcionamiento del programa “PC Parts”, el cual se presenta como proyecto de la primera parte de la asignatura Comunicación Persona-Máquina.

Temática del proyecto

Este llamado PC Parts que se presenta como una aplicación gráfica desarrollado en Java, tiene como objetivo simular una tienda de ordenadores y tecnología, ofreciendo a los usuarios registrados la posibilidad de navegar y “comprar” una variedad de productos y componentes de ordenadores y crear su propia configuración personalizada. Además, se ha implementado una herramienta de diseño propio para que los usuarios puedan personalizar su ordenador añadiendo su propio diseño en una herramienta de paint.

Este proyecto también cuenta con una funcionalidad para los usuarios administradores, quienes tienen acceso a un menú que les permite gestionar la base de datos simulada de usuarios y productos, permitiéndoles llevar a cabo procesos de Creación, Lectura, Actualización y Eliminación de registros. En definitiva, este proyecto busca ofrecer una experiencia completa de compra y personalización de ordenadores y componentes, así como una gestión eficiente de la base de datos de la tienda.

Guía de usuario básica

¿Qué debe saber el usuario?

Antes de comenzar a utilizar la tienda de ordenadores y tecnología, es importante que el usuario conozca las funcionalidades que ofrece el sitio web y cómo utilizarlas de manera efectiva. En esta guía se describen las acciones básicas que se pueden realizar en la plataforma.

Navegación por la tienda

Para poder encontrar los productos que se necesitan, es importante que el usuario se familiarice con la navegación de la tienda. La página principal ofrece una lista con todo el catálogo de productos y en caso de un equipo personalizado cuenta con una ventana de configuración.

Configuración personalizada

Una de las características principales de la tienda es la posibilidad de crear una configuración personalizada de ordenador. El usuario puede elegir los componentes que desea incluir en su ordenador y diseñar su propio modelo utilizando la herramienta de paint personalizada.

Proceso de compra

El proceso de compra es sencillo y guiado, el usuario debe elegir los componentes que desea incluir en su ordenador y añadirlos al carrito de compra. Una vez que todos los componentes están en el carrito, el usuario puede pasar por una ventana de pago donde puede elegir entre dos modalidades de pago (Tarjeta bancaria o Bitcoin). Una vez introducidos los datos de pago se procederá a calcular el importe total (calculando el coste del envío en caso de estar seleccionado al igual que si se introduce un código de descuento).

Proceso de administración

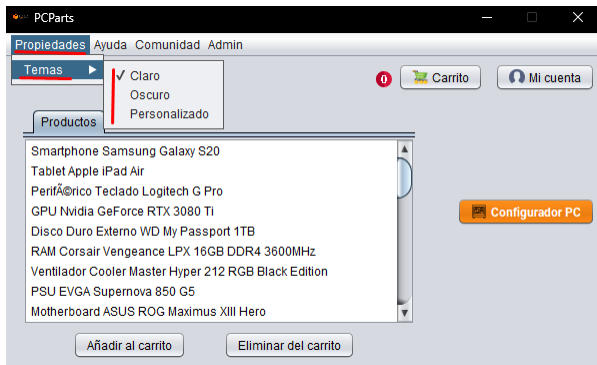
Para los usuarios administradores, la tienda ofrece un menú especial que les permite gestionar la base de datos simulada de usuarios y productos. Esta sección describe las principales acciones que se pueden realizar en el proceso de administración.

Bloques

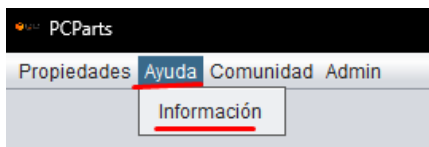
Bloque 1. Ventana y eventos

1.1 Barra de menús con algunos menús y submenús

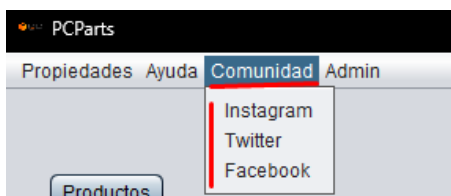
Incluir captura de pantalla. (Opcional: otros detalles).



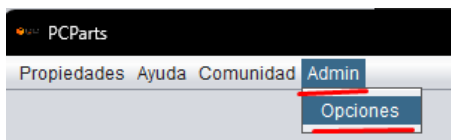
Menú Propiedades con submenú Temas dando opción a cambiar el tema de la ventana principal.



Menú ayuda con opción a abrir un panel de información.



Menú comunidad con las redes sociales de la tienda de la que esta inspirado el trabajo (PC Componentes).



Menú Admin con Item Opciones que es solo visible para usuarios administradores y abre una ventana con múltiples opciones de administración.

1.2 Algunos componentes en un *layout* diferente del default

Indicar nombre de ventana o ventana/componente y layout que tiene.

Ventana	Componente	Layout
Admin		Grid Bag Layout
Paint	PPaint (Panel)	Grid Layout

1.3 Componentes modificados por eventos

Indicar nombre de ventana, nombre de componente, clase de componente y nombre del evento (nombre del método) que modifica este componente

Ventana	Componente	Clase	Evento
Principal	Modificado	LElementosCarrito	Principal
	Modificado	LElementosCarrito	Principal
Código			
<pre>private void BAñadirCarritoActionPerformed(java.awt.event.ActionEvent evt) { if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.add(p); mostrarCantidadCarrito(); } } } } private void BEliminarCarritoActionPerformed(java.awt.event.ActionEvent evt) { if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.remove(p); mostrarCantidadCarrito(); } } } }</pre>			

1.4 Tratamiento de eventos de teclado

Indicar el nombre de la ventana y el nombre del evento (nombre del método) que trata eventos de teclado.

Ventana	Componente	Evento
GoogleAuth	TFMail	KeyReleased
Código		
<pre>private void TFMailKeyReleased(java.awt.event.KeyEvent evt) { if(BSiguiente.isVisible()){ String CC = TFMail.getText(); if(!CC.contains("@GMAIL.COM")){ TFMail.setBorder(BorderFactory.createLineBorder(Color.red, 2, true)); } else{ TFMail.setBorder(BorderFactory.createLineBorder(Color.green, 2, true)); BSiguiente.setEnabled(true); } } }</pre>		

Ventana	Componente	Evento
GoogleAuth	TFMail	KeyTyped
Código		
<pre>private void TFMailKeyTyped(java.awt.event.KeyEvent evt) { if(BSiguiente.isVisible()){ char c=evt.getKeyChar(); String caux = ""+c; caux = caux.toUpperCase(); c = caux.charAt(0); evt.setKeyChar(c); } else{ if(!TFMail.getText().isEmpty()){ BNext.setEnabled(true); } } }</pre>		

1.5 Modificación de componentes con un número variable de elementos

Indicar nombre de ventana, nombre de componente, clase de componente y nombre del evento (nombre del método) que modifica este componente.

Ventana	Componente		Evento
Principal y Carrito	Modificado	LCarrito	BAñadirCarritoActionPerformed
	Modificado	LCarrito	BEliminarCarritoActionPerformed
Código			
<pre>Clase Principal: private void BAAñadirCarritoActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.add(p); mostrarCantidadCarrito(); } } } } private void BEliminarCarritoActionPerformed(java.awt.event.ActionEvent evt) { if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.remove(p); mostrarCantidadCarrito(); } } } }</pre>			

Clase Carrito:

```

public void cargaCarrito() {
    javax.swing.DefaultListModel M = (javax.swing.DefaultListModel)
this.LCarrito.getModel();
    Map<Integer, Producto> mapProductos = new HashMap<Integer, Producto>();
    if (this.carrito != null) {
        if (this.carrito.size() > 0) {
            this.BRealizarPedido.setEnabled(true);
        } else {
            this.BRealizarPedido.setEnabled(false);
        }
        for (Producto producto : this.carrito) {
            int hashCode = producto.hashCode();
            if (mapProductos.containsKey(hashCode)) {
                Producto p = mapProductos.get(hashCode);
                int cantidad = cantidadProducto(producto);
                String productoStr = cantidad + " " + producto.getNombre() + " "
+ producto.getPrecio();
                // Buscar y eliminar entradas repetidas del producto a insertar
                for (int i = 0; i < M.getSize(); i++) {
                    String entry = (String) M.getElementAt(i);
                    if (entry.contains(producto.getNombre())) {
                        M.removeElementAt(i);
                        break;
                    }
                }

                M.addElement(productoStr);
            } else {
                mapProductos.put(hashCode, producto);
                String productoStr = "1 " + producto.getNombre() + " " +
String.format("%.2f", producto.getPrecio());

                // Buscar y eliminar entradas repetidas del producto a insertar
                for (int i = 0; i < M.getSize(); i++) {
                    String entry = (String) M.getElementAt(i);
                    if (entry.contains(producto.getNombre())) {
                        M.removeElementAt(i);
                        break;
                    }
                }

                M.addElement(productoStr);
            }
        }
    }
    precioCarrito();
}

```

Ventana	Componente	Evento
Carrito	Modificado	LCarrito
Código		
<pre> private void BEliminarCarritoActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: if (LCarrito.getSelectedValue() != null) { System.out.println("Selected Item: " + LCarrito.getSelectedValue()); if (this.carrito != null) { javax.swing.DefaultListModel M = (javax.swing.DefaultListModel) this.LCarrito.getModel(); Producto selected = null; for (Producto producto : this.carrito) { String[] splitSelectedItem = LCarrito.getSelectedValue().split(" "); </pre>		


```

        String nameSelectedItem = "";
        for (int i = 1; i <= splitSelectedItem.length - 2; i++) {
            if (i != 1) {
                nameSelectedItem += " " + splitSelectedItem[i];
            } else {
                nameSelectedItem += splitSelectedItem[i];
            }
        }
        if (producto.getNombre().contains(nameSelectedItem)) {
            selected = producto;
        }
    }
    if (selected != null) {
        while (cantidadProducto(selected) > 0) {
            this.carrito.remove(this.carrito.indexOf(selected));
            System.out.println("Remove Item: " + selected.getNombre());
            System.out.println(this.carrito.size());
        }
        this.VPrincipal.carrito = this.carrito;
    }
    M.remove(LCarrito.getSelectedIndex());
    this.VPrincipal.mostrarCantidadCarrito();
    precioCarrito();
    if (this.carrito.size() > 0) {
        this.BRealizarPedido.setEnabled(true);
    } else {
        this.BRealizarPedido.setEnabled(false);
    }
}
}
}

```

1.6 Elementos no vistos

Indicar información necesaria (Ventana, Componente) para localizarlo

Ventana	Componente	Evento
Login	LExit	LExitMouseClicked
Código		
<pre> private void LExitMouseClicked(java.awt.event.MouseEvent evt) { // Lo importante no es este evento, seria que la ventana se encuentra //undecorated y no cuenta con un marco del SO y este botón (que es un label pero //cumple la función de botón) cierra el programa System.exit(0); } </pre>		



Bloque 2. Varias ventanas

2.1 La ventana principal modifica componentes de otra ventana

Nombre de la ventana de principal. Nombre de la otra ventana y nombre del componente modificado

Ventana principal	Ventana no principal	Componente
Principal	Carrito	LCarrito
Código		
<p>Copiar aquí el código del evento (incluida definición de método) que modifica el componente de la otra ventana:</p> <p>Clase Principal:</p> <pre>private void BAñadirCarritoActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.add(p); mostrarCantidadCarrito(); } } } }</pre> <p>private void BEliminarCarritoActionPerformed(java.awt.event.ActionEvent evt) { if (LProductos.getSelectedValue() != null) { for (Producto p : this.productos) { if (p.getNombre().equals(LProductos.getSelectedValue())) { this.carrito.remove(p); mostrarCantidadCarrito(); } } } }</p> <p>Clase Carrito:</p> <pre>public void cargaCarrito() { javax.swing.DefaultListModel M = (javax.swing.DefaultListModel) this.LCarrito.getModel(); Map<Integer, Producto> mapProductos = new HashMap<Integer, Producto>(); if (this.carrito != null) { if (this.carrito.size() > 0) { this.BRealizarPedido.setEnabled(true); } else { this.BRealizarPedido.setEnabled(false); } for (Producto producto : this.carrito) { int hashCode = producto.hashCode(); if (mapProductos.containsKey(hashCode)) { Producto p = mapProductos.get(hashCode); int cantidad = cantidadProducto(producto); String productoStr = cantidad + " " + producto.getNombre() + " " + producto.getPrecio(); // Buscar y eliminar entradas repetidas del producto a insertar for (int i = 0; i < M.getSize(); i++) { String entry = (String) M.getElementAt(i); if (entry.contains(producto.getNombre())) { M.removeElementAt(i); break; } } } } } }</pre>		

```

        M.addElement(productoStr);
    } else {
        mapProductos.put(hashCode, producto);
        String productoStr = "1 " + producto.getNombre() + " " +
String.format("%.2f", producto.getPrecio());

        // Buscar y eliminar entradas repetidas del producto a insertar
        for (int i = 0; i < M.getSize(); i++) {
            String entry = (String) M.getElementAt(i);
            if (entry.contains(producto.getNombre())) {
                M.removeElementAt(i);
                break;
            }
        }

        M.addElement(productoStr);
    }
}
}
precioCarrito();
}

```

Copiar aquí el constructor o constructores de la ventana principal:

```

public Principal() {
    VLogin = new Login(this);
    VLogin.setVisible(true);
    initComponents();
    LProductos.setModel(new javax.swing.DefaultListModel());
    cargaProductos();
    mostrarCantidadCarrito();
    this.setIconImage(new ImageIcon("src/images/logo.png").getImage());
    this.MAdmin.setVisible(false);
}

```

2.2 Una ventana no principal modifica componentes de la principal

Nombre de la ventana de principal. Nombre de la otra ventana y nombre del componente modificado

Ventana principal	Ventana no principal	Componente
Principal	Carrito	BEliminar
Código		
<p>Copiar aquí el código del evento (incluida definición de método) que modifica el componente de la ventana principal:</p> <pre> private void BEliminarCarritoActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: if (LCarrito.getSelectedValue() != null) { System.out.println("Selected Item: " + LCarrito.getSelectedValue()); if (this.carrito != null) { javax.swing.DefaultListModel M = (javax.swing.DefaultListModel) this.LCarrito.getModel(); Producto selected = null; for (Producto producto : this.carrito) { String[] splitSelectedItem = LCarrito.getSelectedValue().split(" "); String nameSelectedItem = ""; for (int i = 1; i <= splitSelectedItem.length - 2; i++) { if (i != 1) { nameSelectedItem += " " + splitSelectedItem[i]; } else { </pre>		

```

        nameSelectedItem += splitSelectedItem[i];
    }
}
if (producto.getNombre().contains(nameSelectedItem)) {
    selected = producto;
}
}
if (selected != null) {
    while (cantidadProducto(selected) > 0) {
        this.carrito.remove(this.carrito.indexOf(selected));
        System.out.println("Remove Item: " + selected.getNombre());
        System.out.println(this.carrito.size());
    }
    this.VPrincipal.carrito = this.carrito;
}
M.remove(LCarrito.getSelectedIndex());
this.VPrincipal.mostrarCantidadCarrito();
precioCarrito();
if (this.carrito.size() > 0) {
    this.BRealizarPedido.setEnabled(true);
} else {
    this.BRealizarPedido.setEnabled(false);
}
}
}
}
}

```

Copiar aquí el constructor o constructores de la ventana no principal:

```

private Carrito() {
    initComponents();
}

public Carrito(Principal VPrincipal) {
    this();
    this.VPrincipal = VPrincipal;
    this.carrito = VPrincipal.carrito;
    this.productos = VPrincipal.productos;
    LCarrito.setModel(new javax.swing.DefaultListModel());
    this.setIconImage(new ImageIcon("src/images/logo.png").getImage());
    cargaCarrito();
}

```

2.3 Una ventana no principal modifica componentes de otra ventana no principal

Nombre de la ventana no principal que modifica. Nombre de la otra ventana y nombre del componente modificado.

Ventana modificadora	Ventana modificada	Componente
Configurador	Carrito	BConfigurador
Código		
<p>Copiar aquí el código del evento (incluida definición de método) que modifica el componente de la otra ventana:</p> <pre> private void BConfiguradorActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: this.DOC.VPrincipal.carrito.add(new Producto("PC CUSTOM", calculaPrecioTotal(), creaEspecificacion(), TipoProducto.PC)); this.DOC.VPrincipal.mostrarCantidadCarrito(); this.dispose(); } </pre> <p>Copiar aquí el constructor o constructores de la ventana no principal:</p>		

```
private Configurator() {  
    initComponents();  
    this.setIconImage(new ImageIcon("src/images/logo.png").getImage());  
}  
  
public Configurator(OpcionConfigurator DOC) {  
    this();  
    this.DOC = DOC;  
    this.LChipset.setText(this.DOC.getChipsetSelected());  
    cargarCheckBoxes();  
}
```

2.4 Una ventana inicial que no sea la principal

Nombre de la ventana inicial

Ventana
Login
Código
<p>Copiar constructor o constructores de la ventana inicial:</p> <pre>private Login() { initComponents(); this.setIconImage(new ImageIcon("src/images/logo.png").getImage()); } public Login(Principal VPrincipal) { this(); this.VPrincipal = VPrincipal; }</pre>

Bloque 3. Diálogos

3.1 Un diálogo usando JOptionPane

Nombre de la ventana que tiene la llamada a este diálogo.

Ventana
Login
Código
<pre>Copiar el método (entero) donde se utilice un diálogo de la clase JOptionPane: private void BLoginActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: char[] password = PFPassword.getPassword(); String passwordString = new String(password); if (TFUser.getText().length() > 0 && passwordString.length() > 0) { boolean succes = login(TFUser.getText(), passwordString); if (succes) { String user = TFUser.getText(); usuario = new Usuario(user, passwordString); VPrincipal.setUsuario(this.usuario); VCarga = new Carga(VPrincipal); this.setVisible(false); } else { JOptionPane.showMessageDialog(null, "Inicio de sesión fallido", "Error", JOptionPane.ERROR_MESSAGE); } } }</pre>

3.2 Un diálogo predefinido

Nombre de la ventana que tiene la llamada a este diálogo

Ventana
Principal
Código
<pre>Copiar el método (entero) donde se utilice un diálogo predefinido: private void CBPersonalizadoActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: Color selectedColor = JColorChooser.showDialog(null, "Seleccione un color", this.getBackground()); this.setTheme(selectedColor); }</pre>

3.3 Un diálogo creado por el usuario que pida información al usuario

La ventana del diálogo

Ventana diálogo

DatosProducto

Código

Copiar el código (entero) de la clase del diálogo:

```
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JDialog.java to edit this
template
 */
/**
 *
 * @author ruflas
 */
public class DatosProducto extends javax.swing.JDialog {

    /**
     * Creates new form DatosProducto
     */
    private DatosProducto(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        cargaTipoProductos();
    }

    public DatosProducto(Admin VAdmin) {
        this(null, true);
        this.setIconImage(new ImageIcon("src/images/logo.png").getImage());
        POK = false;
        this.VAdmin = VAdmin;
    }

    public boolean showDialog() {
        POK = false;
        TFNombre.setText("");
        setVisible(true);
        return POK;
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        BCancel = new javax.swing.JButton();
        BOk = new javax.swing.JButton();
        TFNombre = new javax.swing.JTextField();
        TFPrecio = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        TFOferta = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        CBTipoProducto = new javax.swing.JComboBox<>();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        TAEspecificaciones = new javax.swing.JTextArea();
        LChipset = new javax.swing.JLabel();
        TFChipset = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
```



```

setTitle("Datos Producto");

jLabel1.setText("Nombre:");

BCancel.setText("Cancelar");
BCancel.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BCancelActionPerformed(evt);
    }
});

BOk.setText("Aceptar");
BOk.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BOkActionPerformed(evt);
    }
});

TFPrecio.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        TFPrecioKeyTyped(evt);
    }
});

jLabel2.setText("Precio:");

jLabel4.setText("Oferta:");

TFOferta.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        TFOfertaKeyTyped(evt);
    }
});

jLabel5.setText("Tipo:");

CBTipoProducto.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CBTipoProductoActionPerformed(evt);
    }
});

jLabel3.setText("Especificaciones:");

TAEspecificaciones.setColumns(20);
TAEspecificaciones.setRows(5);
jScrollPane1.setViewportViewView(TAEspecificaciones);

LChipset.setText("Chipset:");
LChipset.setEnabled(false);

TFChipset.setEnabled(false);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(BOk)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(BCancel)
                .addContainerGap())
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(LChipset)
                .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(TFPrecio)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(TFOferta)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(CBTipoProducto)
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(TAEspecificaciones)
            .addContainerGap())
);

```

```

        .addComponent(LChipset)
        .addGap(18, 18, 18)
        .addComponent(TFChipset,
javax.swing.GroupLayout.PREFERRED_SIZE, 103, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel13)
        .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel1)
        .addComponent(jLabel2)
        .addComponent(jLabel4)
        .addComponent(jLabel5))
        .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(TFPrecio)
        .addComponent(TFNombre,
javax.swing.GroupLayout.DEFAULT_SIZE, 106, Short.MAX_VALUE)
        .addComponent(TFOferta)
        .addComponent(CBTipoProducto, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
196, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(103, 103, 103))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addGap(24, 24, 24)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel1)
        .addComponent(TFNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(TFPrecio, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel2))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(TFOferta, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(CBTipoProducto,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(21, 21, 21)
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
110, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(LChipset)
        .addComponent(TFChipset, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(BOK)
        .addComponent(BCancel))
        .addGap(28, 28, 28))

```

```

    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

private void BCancelActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
}

private void BOkActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (TFNombre.getText().length() > 0 && TFOferta.getText().length() > 0 &&
        TFPrecio.getText().length() > 0) {
        if (!(Double.parseDouble(TFOferta.getText()) >= 0 &&
            Double.parseDouble(TFOferta.getText()) < 1)) {
            JOptionPane.showMessageDialog(null, "Oferta debe ser mayor o igual que
            0 y menor que 1.", "Error", JOptionPane.ERROR_MESSAGE);
        } else {
            if ((CBTipoProducto.getSelectedItem().equals("CPU") ||
                CBTipoProducto.getSelectedItem().equals("Motherboard")) && TFChipset.getText().length()
                == 0) {
                JOptionPane.showMessageDialog(null, "Debes introducir el chipset.",
                "Error", JOptionPane.ERROR_MESSAGE);
            } else {
                POK = true;
                TipoProducto tipo = TipoProducto.valueOf("'" +
                CBTipoProducto.getSelectedItem());
                System.out.println(tipo);
                if (tipo.equals(TipoProducto.CPU) ||
                    tipo.equals(TipoProducto.Motherboard)) {
                    VAdmin.setDatosProducto(TFNombre.getText(),
                    Double.parseDouble(TFPrecio.getText()), Double.parseDouble(TFOferta.getText()), tipo,
                    TAEspecificaciones.getText(), TFChipset.getText());
                }
                else {
                    System.out.print("Vaya me he librado pero sigue sin ir siendo
                    tipo "+tipo);
                    VAdmin.setDatosProducto(TFNombre.getText(),
                    Double.parseDouble(TFPrecio.getText()), Double.parseDouble(TFOferta.getText()), tipo,
                    TAEspecificaciones.getText(), "");
                }
                this.dispose();
            }
        }
    } else {
        JOptionPane.showMessageDialog(null, "Algun campo esta vacio.", "Error",
        JOptionPane.ERROR_MESSAGE);
    }
}

private void TFPrecioKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (!(Character.isDigit(c) || c == '.')) {
        evt.consume();
    }
}

private void TFOfertaKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if (!(Character.isDigit(c) || c == '.')) {
        evt.consume();
    }
}

private void CBTipoProductoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (CBTipoProducto.getSelectedItem().equals("CPU") ||

```

```

CBTipoProducto.getSelectedItem().equals("Motherboard")) {
    LChipset.setEnabled(true);
    TFChipset.setEnabled(true);
} else {
    LChipset.setEnabled(false);
    TFChipset.setEnabled(false);
}
}

private void cargaTipoProductos() {
    for (TipoProducto tipo : TipoProducto.values()) {
        CBTipoProducto.addItem(tipo.toString());
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(DatosProducto.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(DatosProducto.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(DatosProducto.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(DatosProducto.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            DatosProducto dialog = new DatosProducto(new javax.swing.JFrame(),
true);

            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}
private boolean POK;
Admin VAdmin;

```

```
// Variables declaration - do not modify
private javax.swing.JButton BCancel;
private javax.swing.JButton BOk;
private javax.swing.JComboBox<String> CBTipoProducto;
private javax.swing.JLabel LChipset;
private javax.swing.JTextArea TAEspecificaciones;
private javax.swing.JTextField TFChipset;
private javax.swing.JTextField TFNombre;
private javax.swing.JTextField TFOferta;
private javax.swing.JTextField TFPrecio;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration
}
```

Nombre de la ventana que tiene la llamada a este diálogo

Ventana

Admin

Código

```
Copiar el método (entero) donde se llama al diálogo creado por el usuario:
private void BAñadirProductoActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    boolean POK = DP.showDialog();
    Producto p;
    if (POK) {
        if (this.chipset.isEmpty()) {
            p = new Producto(this.nombre_producto, this.precio_producto,
this.especificaciones_producto, this.oferta_producto, this.tipo_producto);
            this.añadirProducto(p, "");
        } else {
            if (this.tipo_producto.equals(TipoProducto.CPU)) {
                CPU c= new CPU(this.nombre_producto, this.precio_producto,
this.especificaciones_producto, this.oferta_producto, this.chipset);
                this.añadirProducto(c, this.chipset);
            } else {
                PlacaBase e = new PlacaBase(this.nombre_producto,
this.precio_producto, this.especificaciones_producto, this.oferta_producto,
this.chipset);
                this.añadirProducto(e, this.chipset);
            }
        }
        VPrincipal.cargaProductos();
    }
}
```

Bloque 4. Interfaz en primer plano

4.1 Métodos set para dar información a la tarea

Nombre de la clase donde se llaman.

Clase
Pago
Código
<pre>Método set ciclos tarea: public void setCiclos(int ciclos){ this.ciclos = ciclos; } Método donde se llama: private void BImporteActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: BCancel.setEnabled(true); BPago.setEnabled(true); this.setCiclos(10); this.tarea(this.ciclos); }</pre>

4.2 Métodos que envíen información de la tarea a la interfaz

Nombre de la clase que implementa los métodos por los que la tarea informa a la interfaz.

Clase
Pago
Código
<p>Copiar aquí el código (completo) de estos métodos No he utilizado ninguna interfaz pues he aplicado el código directamente sobre una función tarea() como se puede ver debajo en 4.3 .</p>

4.3 Posibilidad de hacer un stop

Copiar aquí el método que permite al usuario hacer un Stop

Clase
Pago
Método stop:
<pre>Copiar aquí el método que permite al usuario hacer un Stop: public boolean Stop() { return this.stop; } private void BCancelActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: this.stop = true; this.BCancel.setEnabled(false); }</pre>

Llamada al método stop

Clase
Pago
Código
<pre> public void tarea(int ciclos) { boolean error = false; if (!TFPago.getText().isEmpty()) { if ((CBTarjeta.isSelected() && !TFCVV.getText().isEmpty()) CBBitcoin.isSelected()) { if (RBEEnvio.isSelected() RBRecogida.isSelected()) { this.stop = false; TAImporte.setText(""); HiloBack = new Thread() { public void run() { double rand = (Math.random() + 1) * Math.random() * 5, total = 0; int i = 0; while (i < ciclos) { try { Thread.sleep((int) (0.3 * 1000)); } catch (java.lang.InterruptedException E) { } if (Stop()) // El interfaz informa al proceso { break; } if (i == 0) { TAImporte.setText(TAImporte.getText() + "Importe carrito: " + String.format("%.2f", VCarrito.precioCarrito())); total += VCarrito.precioCarrito(); } else if (i == (int) ciclos*0.2) { if (RBEEnvio.isSelected()) { TAImporte.setText(TAImporte.getText() + "\nEnvio: " + String.format("%.2f", rand)); total += rand; } else { TAImporte.setText(TAImporte.getText() + "\nRecogida: " + 0); } } else if (i == (int) ciclos*0.7) { if (TFCodDescuento.getText().equals("-10")) { TAImporte.setText(TAImporte.getText() + "\nDescuento: -10%"); total = total - (total * 0.1); } else { TAImporte.setText(TAImporte.getText() + "\nDescuento: NO APLICADO"); } } else if (i == (int) ciclos*0.9) { TAImporte.setText(TAImporte.getText() + "\nTotal: " + String.format("%.2f €", total)); } i++; } } }; HiloBack.start(); } else { error = true; } } else { error = true; } } } </pre>

```
        } else {  
            error = true;  
        }  
        if (error) {  
            JOptionPane.showMessageDialog(null, "Algún campo está vacío.",  
"Error", JOptionPane.ERROR_MESSAGE);  
        }  
    }  
}
```


Bloque 5. Gráficos

5.1 Clase hija de un componente del que se refine su método Paint

Nombre de la clase.

Clase
PaintPanel
Código
Código completo del método <i>Paint</i>: <pre>@Override public void paintComponent(Graphics g) { super.paintComponent(g); Graphics2D g2 = (Graphics2D) g; // Dibujar todas las formas en la lista Graphics2D g2d = (Graphics2D) g; for (Shape shape : shapes) { g2d.setColor(color); g2d.fill(shape); } }</pre>

5.2 Métodos set para modificar lo que se pinta en la clase anterior

Declaración de las estructuras de datos que se modifican

Clase
PaintPanel
Código
<pre>public void setColor(Color c) { this.color = c; }</pre>
Código
Declaración de las estructuras de datos: <pre>public ArrayList<Shape> shapes = new ArrayList<>(); private Color color = Color.black;</pre>

5.3 Utilización del método repaint

Clase
Paint
Código
<pre>private void MIColorActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: Color c = JColorChooser.showDialog(null, null, paintPanel.getColor()); if (c != null) { paintPanel.setColor(c); paintPanel.repaint(); } }</pre>

```

private void PPaintMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (this.RBPincel.isSelected()) {
        // Crear un nuevo círculo en la posición actual del ratón
        Shape circle = new Ellipse2D.Double(evt.getX() - 10, evt.getY() -
10, 20, 20);
        // Agregar el círculo a la lista de formas
        paintPanel.shapes.add(circle);
        // Repintar el panel de dibujo para mostrar las formas
        repaint();

    } else {
        paintPanel.removeComponent(evt.getX(), evt.getY());
        repaint();
    }
}

```

5.4 Elemento no visto en clase ni en apuntes

Clase

Paint y PaintPanel

Código

```

Clase Paint:
private void PPaintMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (this.RBPincel.isSelected()) {
        // Crear un nuevo círculo en la posición actual del ratón
        Shape circle = new Ellipse2D.Double(evt.getX() - 10, evt.getY() -
10, 20, 20);
        // Agregar el círculo a la lista de formas
        paintPanel.shapes.add(circle);
        // Repintar el panel de dibujo para mostrar las formas
        repaint();

    } else {
        paintPanel.removeComponent(evt.getX(), evt.getY());
        repaint();
    }
}

private void PPaintMouseDragged(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (this.RBPincel.isSelected()) {
        // Crear un nuevo círculo en la posición actual del ratón
        Shape circle = new Ellipse2D.Double(evt.getX() - 10, evt.getY() -
10, 20, 20);
        // Agregar el círculo a la lista de formas
        paintPanel.shapes.add(circle);
        // Repintar el panel de dibujo para mostrar las formas
        repaint();
    } else {
        paintPanel.removeComponent(evt.getX(), evt.getY());
        repaint();
    }
}

Clase PaintPanel:
public void removeComponent(int x, int y){
    ArrayList<Shape> aux = new ArrayList<>();
    aux.addAll(shapes);
    for(Shape s: aux){
        if(s.contains(x, y)){
            this.shapes.remove(s);
        }
    }
}

```

