Simulacion y optimizacion de un programa en un procesador escalar segmentado

David Bocanegra

Laura Barona

Alfonso Narvaez

Universidad Internacional de la Rioja

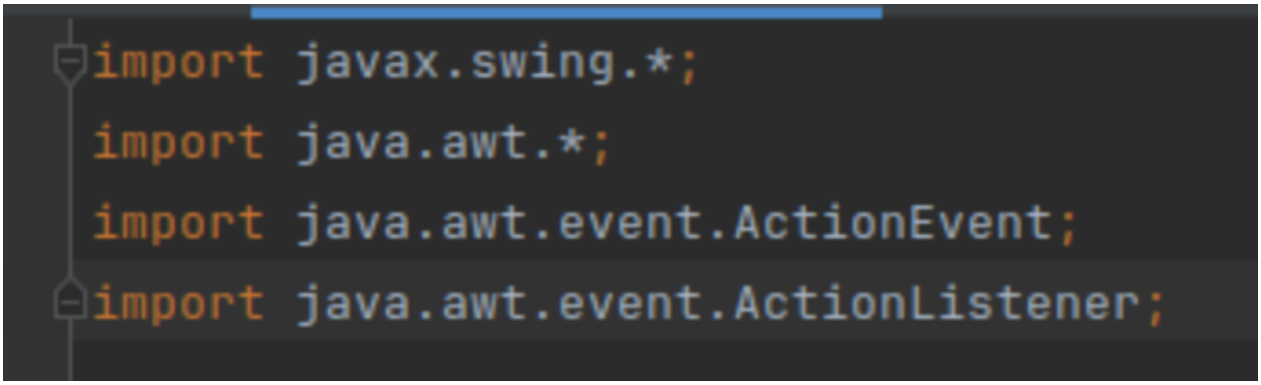
Javier Diaz Diaz

18/septiembre/2023

Documentación código interfaz de usuario.

Para poder entender todo este documento vamos a explicar parte por parte del código en donde vamos a poder encontrar la descripción detallada de cada uno de los aspectos del mismo

Así que la primera parte que vamos a explicar son las importaciones y paquetes de Java que realizamos para la creación de la una interfaz de usuario.



Así que la primera importación es "javax.swing" la cual nos funciona para proporcionar clases y componentes para poder crear las interfaces gráficas de usuario. Por lo cual está importación contiene clases para poder crear ventanas, botones, campos de texto entre otros.

La segunda importación que vamos a realizar es "java.awt" en donde esta va a poder darnos la creación de las interfaces de usuario, teniendo clases relacionadas con la gestión de diseños y componentes gráficos como *GridLayout* el cual se encarga de organizar los componentes en filas y en columnas.

La tercera importación que se realiza es "java.awt.event.ActionEvent" la cual Importa la clase ActionEvent del paquete java.awt.event. ActionEvent en donde la función de esta es representar eventos generados por componentes de la GUA, como hacer clic en un botón.

Y por último la importación que se realiza es "java.awt.event.ActionListener" Importa la interfaz *ActionListener* del paquete *java.awt.event. ActionListener* es una interfaz que define un método llamado *actionPerformed*, que debe ser implementado para manejar acciones, como hacer clic en un botón, cuando ocurren en la GUI.

Ahora la siguiente parte a explicar va a ser:

```
public class FarmaciaApp {
    6 usages
    private JFrame frame;
    4 usages
    private JTextField nombreMedicamentoField;
    4 usages
    private JComboBox<String> tipoMedicamentoComboBox;
    4 usages
    private JTextField cantidadField;
    5 usages
    private ButtonGroup distribuidorGroup;
    5 usages
    private JCheckBox principalCheckBox;
    5 usages
    private JCheckBox secundariaCheckBox;
```

En la cual podemos indicar que como se ve en la imagen estamos creando una clase con diferentes campos de métodos privados en donde encontramos que los campos se utilizan para interactuar con los componentes de la interfaz de usuario y para recopilar la información ingresada por el usuario en la aplicación y esto lo que va a realizar es indicar que cada uno de ellos tiene un tipo específico que corresponde al tipo de componente que representa (ventana, campo de texto, cuadro de lista, etc.), y se utilizan para acceder y manipular estos componentes dentro del código de la aplicación.

Ahora la sección a explicar es la siguiente:

```
1 usage   Oscar Bocanegra
public FarmaciaApp() {
    frame = new JFrame( title: "Sistema de Pedidos de Medicamentos");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize( width: 400,  height: 400);

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout( rows: 7,  cols: 2));
```

Esta parte del código es la parte del constructor de la clase FarmaciaApp donde se observa que se ha creado y configurado la interfaz gráfica de usuario de la aplicación de la farmacia.

En donde explicando paso a paso encontramos que:

frame = new JFrame("Sistema de Pedidos de Medicamentos"); es el encargado de crear una nueva instancia de JFrame llamada frame, que representa la ventana principal de la aplicación y de igual manera se establece el título de la ventana como "Sistema de Pedidos de Medicamentos" mediante el constructor del JFrame.

luego en la línea *frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);*

Lo que hacemos es configurar la acción que ocurrirá cuando el usuario cierre la ventana principal. En este caso, se ha configurado para que la aplicación se cierre por completo al cerrar la ventana.

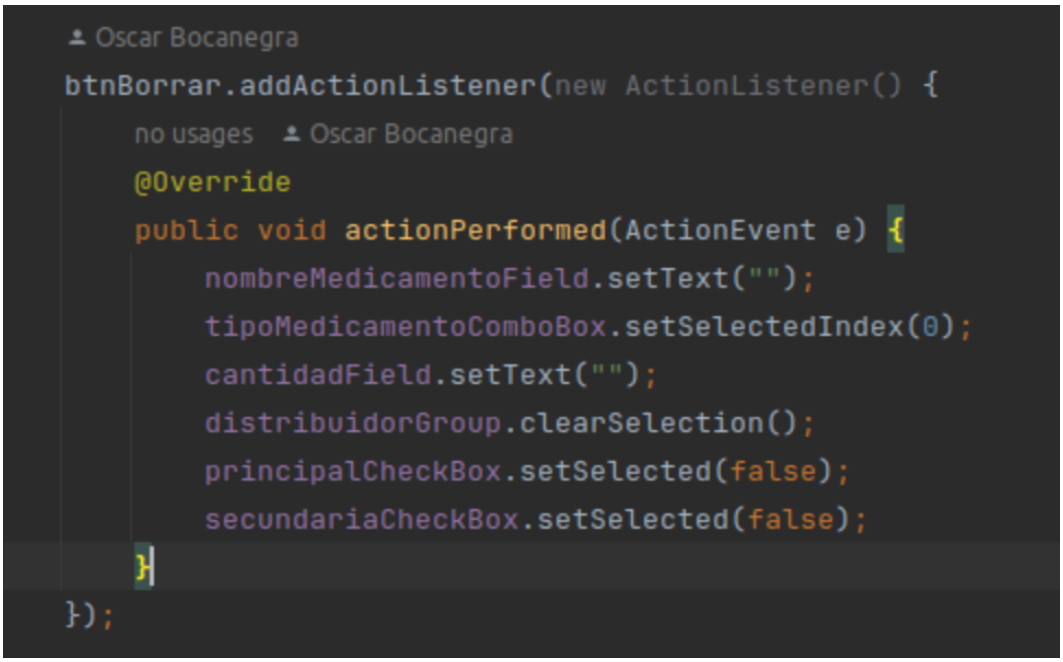
Y luego de esto en la línea frame.setSize lo que hacemos es establecer el tamaño de la ventana principal de 400 píxeles de ancho y 400 píxeles de alto.

Y por último se crea un nuevo panel (JPanel) llamado panel, que se utilizará para organizar los componentes de la interfaz de usuario y se configura el diseño del panel como una cuadrícula (GridLayout) con 7 filas y 2 columnas. Esto significa que los componentes se organizan en 7 filas y 2 columnas en el panel.

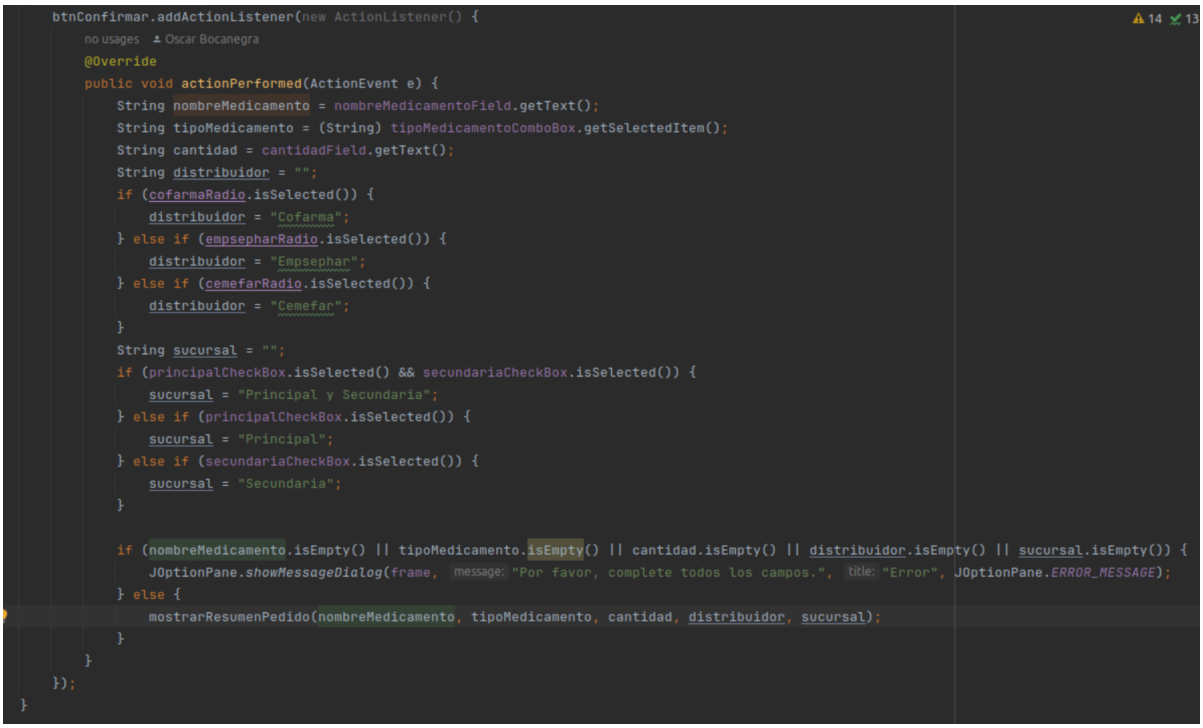
```
JLabel lblNombreMedicamento = new JLabel( text: "Nombre del Medicamento:");
nombreMedicamentoField = new JTextField();
JLabel lblTipoMedicamento = new JLabel( text: "Tipo del Medicamento:");
tipoMedicamentoComboBox = new JComboBox<>(new String[]{"Analgésico", "Analéptico",
        "Anestésico", "Antiácido", "Antidepresivo", "Antibiótico"});
JLabel lblCantidad = new JLabel( text: "Cantidad de Producto:");
cantidadField = new JTextField();
JLabel lblDistribuidor = new JLabel( text: "Distribuidor Farmacéutico:");
JPanel distribuidorPanel = new JPanel();
distribuidorGroup = new ButtonGroup();
JRadioButton cofarmaRadio = new JRadioButton( text: "Cofarma");
JRadioButton empsepharRadio = new JRadioButton( text: "Empsephar");
JRadioButton cemefarRadio = new JRadioButton( text: "Cemefar");
distribuidorGroup.add(cofarmaRadio);
distribuidorGroup.add(empsepharRadio);
distribuidorGroup.add(cemefarRadio);
distribuidorPanel.add(cofarmaRadio);
distribuidorPanel.add(empsepharRadio);
distribuidorPanel.add(cemefarRadio);

JLabel lblSucursal = new JLabel( text: "Sucursal de la Farmacia:");
JPanel sucursalPanel = new JPanel();
principalCheckBox = new JCheckBox( text: "Principal");
secundariaCheckBox = new JCheckBox( text: "Secundaria");
sucursalPanel.add(principalCheckBox);
sucursalPanel.add(secundariaCheckBox);

JButton btnBorrar = new JButton( text: "Borrar");
JButton btnConfirmar = new JButton( text: "Confirmar");
```

Ahora en la imagen anterior lo que vamos a hacer es crear y configurar la interfaz de usuario de una aplicación de farmacia, donde el usuario puede ingresar información sobre el medicamento que desea pedir, seleccionar un distribuidor y elegir la sucursal de la farmacia para la entrega. Los componentes mencionados se organizan en un panel que luego se agrega a la ventana principal de la aplicación.

Ahora en la siguiente parte del código podemos encontrar que cuando el usuario hace clic en el botón "Borrar", todas las entradas y selecciones realizadas en la interfaz de usuario (nombre del medicamento, tipo, cantidad, distribuidor y sucursal) se borran o deseleccionan, lo que permite al usuario empezar de nuevo y realizar un nuevo pedido sin la necesidad de borrar cada campo manualmente.



A la hora de explicar el código el cual contiene una mayor cantidad de condicionales podemos decir que se considera una de las partes mas importantes del codigo ya que la función de este es configura un oyente de acción (action listener) para el botón "Confirmar" (btnConfirmar) en la interfaz de usuario de la aplicación de la farmacia. El oyente de acción se activará cuando el usuario haga clic en el botón "Confirmar", y se encargará de realizar varias tareas relacionadas con la validación de los datos ingresados antes de confirmar un pedido, así que en resumen y para que quede un poco más claro la siguiente parte del código podemos decir que este código se encarga de validar los datos ingresados por el usuario antes de confirmar un pedido en la aplicación de farmacia. Si todos los campos obligatorios están completos, se muestra un resumen del pedido; de lo contrario, se muestra un mensaje de error.



```
btnConfirmar.addActionListener(new ActionListener() {
    no usages  Oscar Bocanegra
    @Override
    public void actionPerformed(ActionEvent e) {
        String nombreMedicamento = nombreMedicamentoField.getText();
        String tipoMedicamento = (String) tipoMedicamentoComboBox.getSelectedItem();
        String cantidad = cantidadField.getText();
        String distribuidor = "";
        if (cofarmaRadio.isSelected()) {
            distribuidor = "Cofarma";
        } else if (empsepharRadio.isSelected()) {
            distribuidor = "Empsephar";
        } else if (cemefarRadio.isSelected()) {
            distribuidor = "Cemefar";
        }
        String sucursal = "";
        if (principalCheckBox.isSelected() && secundariaCheckBox.isSelected()) {
            sucursal = "Principal y Secundaria";
        } else if (principalCheckBox.isSelected()) {
            sucursal = "Principal";
        } else if (secundariaCheckBox.isSelected()) {
            sucursal = "Secundaria";
        }

        if (nombreMedicamento.isEmpty() || tipoMedicamento.isEmpty() || cantidad.isEmpty() || distribuidor.isEmpty() || sucursal.isEmpty()) {
            JOptionPane.showMessageDialog(frame,  message: "Por favor, complete todos los campos.",  title: "Error", JOptionPane.ERROR_MESSAGE);
        } else {
            mostrarResumenPedido(nombreMedicamento, tipoMedicamento, cantidad, distribuidor, sucursal);
        }
    }
});
}
```

Ahora en la siguiente imagen vamos a explicar el código en el cual la función principal es definir un método llamado mostrarResumenPedido que se utiliza para crear y mostrar una nueva ventana de resumen del pedido en la aplicación de farmacia asi que podemos concluir que este código crea una ventana de resumen del pedido con etiquetas informativas y botones para realizar acciones relacionadas con el pedido, como cancelarlo o enviarlo. La información del pedido se pasa como argumentos al método mostrarResumenPedido, y la ventana se muestra al usuario cuando se llama a este método.

```
1 usage  Oscar Bocanegra
private void mostrarResumenPedido(String nombreMedicamento, String tipoMedicamento, String cantidad, String distribuidor, String sucursal) {
    JFrame resumenFrame = new JFrame( title: "Pedido al distribuidor " + distribuidor);
    resumenFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    resumenFrame.setSize( width: 400,  height: 200);

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout( rows: 3,  cols: 1));

    JLabel lblPedido = new JLabel( text: "Pedido al distribuidor " + distribuidor);
    JLabel lblMedicamento = new JLabel( text: cantidad + " unidades del " + tipoMedicamento + " " + nombreMedicamento);
    JLabel lblDireccion = new JLabel( text: "Para la farmacia situada en " + obtenerDireccionFarmacia(sucursal));

    JButton btnCancelar = new JButton( text: "Cancelar");
    JButton btnEnviar = new JButton( text: "Enviar Pedido");

    panel.add(lblPedido);
    panel.add(lblMedicamento);
    panel.add(lblDireccion);
    panel.add(btnCancelar);
    panel.add(btnEnviar);

    resumenFrame.add(panel);
    resumenFrame.setVisible(true);
```

Y por último podemos evidenciar que en la parte final del código se encarga de configurar dos acciones que ocurren cuando se hacen clic en los botones "Cancelar" y "Enviar Pedido" en la ventana de resumen del pedido en la aplicación de farmacia.

```
        btnCancelar.addActionListener(new ActionListener() {
            no usages  Oscar Bocanegra
            @Override
            public void actionPerformed(ActionEvent e) {
                resumenFrame.dispose();
            }
        });

        Oscar Bocanegra
        btnEnviar.addActionListener(new ActionListener() {
            no usages  Oscar Bocanegra
            @Override
            public void actionPerformed(ActionEvent e) {
                System.out.println("Pedido enviado");
                resumenFrame.dispose();
            }
        });
    }

    1 usage  Oscar Bocanegra
    private String obtenerDireccionFarmacia(String sucursal) {
        if (sucursal.equals("Principal")) {
            return "Calle de la Rosa n. 28";
        } else if (sucursal.equals("Secundaria")) {
            return "Calle Alcazabilla n. 3";
        } else {
            return "Calle de la Rosa n. 28 y Calle Alcazabilla n. 3";
        }
    }
```

Conclusiones

- En este proyecto, se desarrolló una aplicación de farmacia que permite a los usuarios realizar pedidos de medicamentos. La aplicación utiliza una interfaz gráfica de usuario (GUI) implementada con Java Swing para facilitar la interacción del usuario.
- La aplicación presenta una interfaz de usuario intuitiva que permite a los usuarios ingresar información sobre el medicamento que desean pedir, como el nombre, el tipo, la cantidad, el distribuidor y la sucursal de la farmacia.
- Los botones "Cancelar" y "Enviar Pedido" proporcionan opciones adicionales a los usuarios. El botón "Cancelar" cierra la ventana de resumen, mientras que el botón "Enviar Pedido" imprime un mensaje de confirmación y luego cierra la ventana.