



LasⁱESpeñetas



UNIÓN EUROPEA
Fondo Social Europeo
El FSE invierte en tu futuro

I.E.S. Las Espeñetas - Orihuela
1º Grado Superior
Desarrollo de Aplicaciones Multiplataforma

Agenda

Departamento de Informática



Alumno: Luis Roberto Cordero Martínez

Alumno: Sergio Medina Cerezo

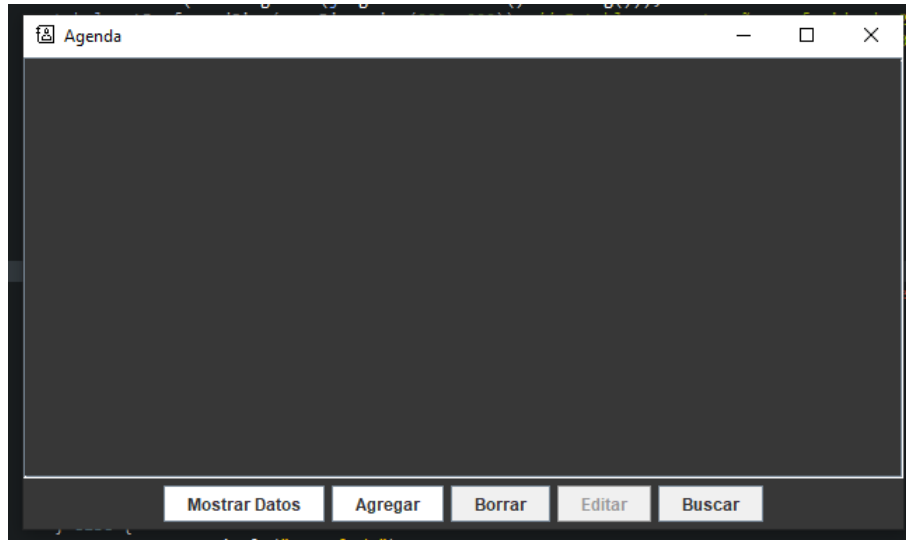
Profesor: Jaime Riquelme García

26 de febrero de 2023

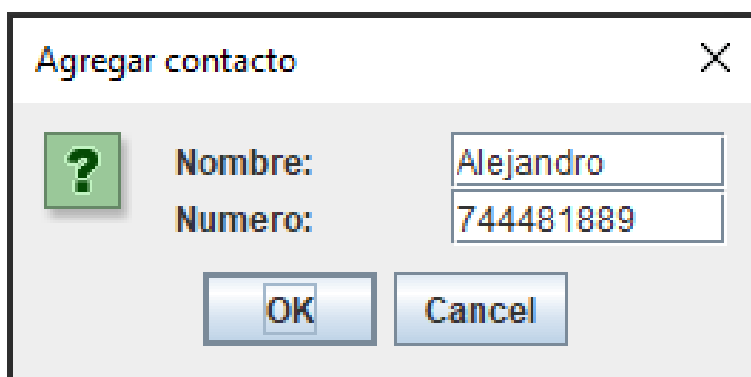
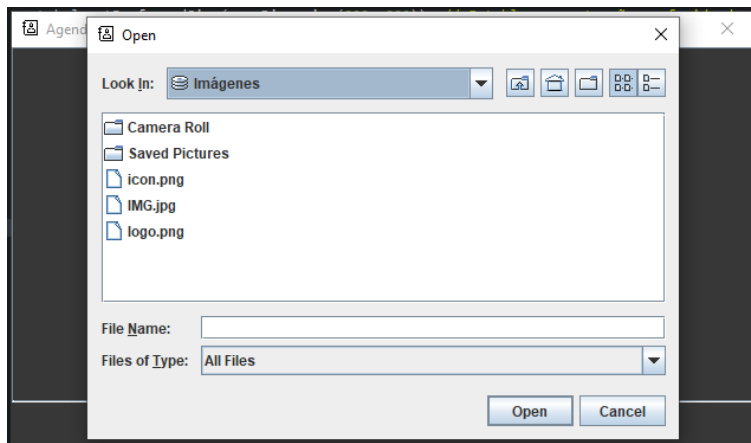


Ejecucion de los botones

En este apartado voy a explicar el funcionamiento de los botones de la agenda.



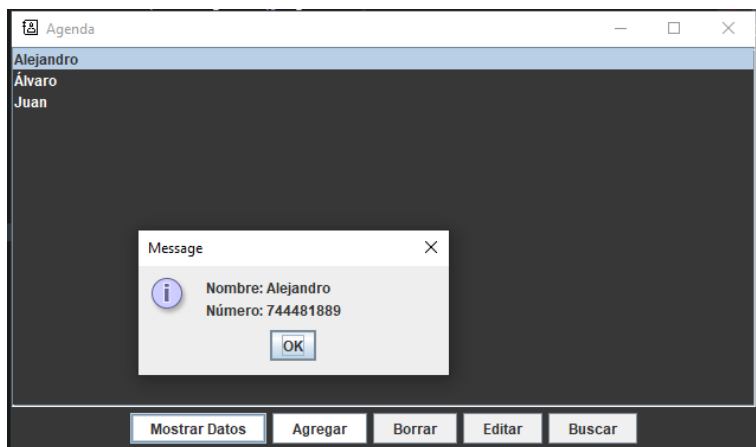
Esta seria la interfaz de la agenda



Para agregar un contacto presionamos el boton de agregar, seleccionamos la foto del contacto y posteriormente introducimos su nombre y número de teléfono.



Seleccionamos el contacto, presionamos en Mostrar Datos.



Primero mostrara el nombre y numero del contacto seleccionado



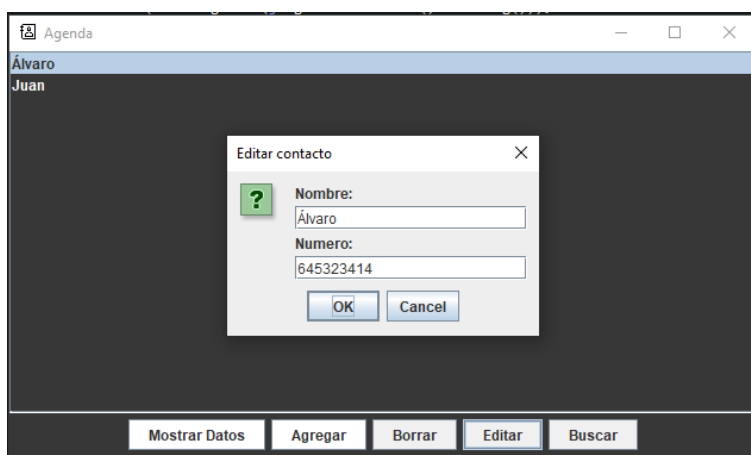
Despues la foto del contacto



Para borrar un contacto seleccionamos el contacto y presionamos el boton borrar.

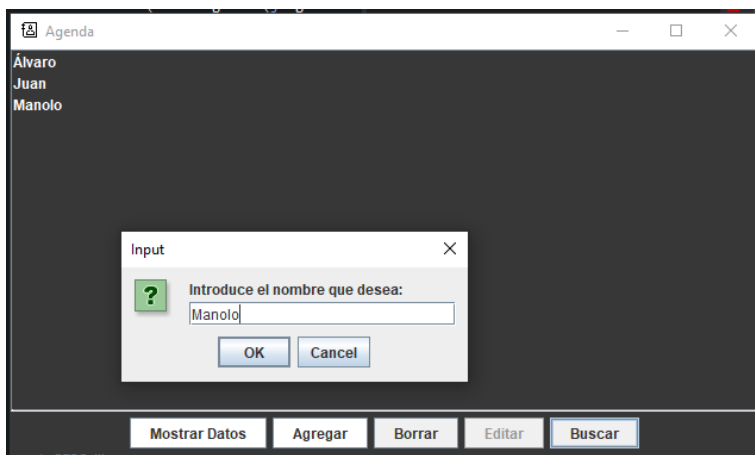


Para editar un contacto, seleccionamos el contacto y presionamos el boton editar.





Para buscar un contacto presionamos el boton buscar y introducimos el nombre del contacto.





Código fuente de los botones

En este apartado explicaremos el código fuente.

Código para añadir un contacto.

```
public void actionPerformed(ActionEvent e) {  
    //Crea los campos  
    String nombre1 = null ;  
    String numero1 = null ;  
    if (e.getSource() == botonAgregar) {  
        //Añadir imagen  
        JFileChooser jf = new JFileChooser();  
        jf.setMultiSelectionEnabled(false);  
        if(jf.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {  
            rsdragdropfiles.RSDragDropFiles.setCopiar(jf.getSelectedFile().toString(),"src/img/logo.png");  
            Label.setIcon(new ImageIcon(jf.getSelectedFile().toString()));  
            Label.setPreferredSize(new Dimension(200, 200)); // Establecer un tamaño preferido de 200x200 píxeles  
            Label.setMaximumSize(new Dimension(300, 300)); // Establecer un tamaño máximo de 200x200 píxeles  
        }  
        // Crear el panel con dos campos de entrada de texto  
        JPanel panel = new JPanel(new GridLayout(2, 2));  
        JTextField nombre = new JTextField();  
        JTextField numero = new JTextField();  
        panel.add(new JLabel("Nombre:"));  
        panel.add(nombre);  
        panel.add(new JLabel("Numero:"));  
        panel.add(numero);  
    }  
}
```

```
// Mostrar la ventana de diálogo personalizada  
int result = JOptionPane.showConfirmDialog(null, panel, "Agregar contacto", JOptionPane.OK_CANCEL_OPTION);  
  
// Obtener los datos ingresados por el usuario  
if (result == JOptionPane.OK_OPTION) {  
    String dato1 = nombre.getText();  
    String dato2 = numero.getText();  
    nombre1 = dato1;  
    numero1 = dato2;  
    System.out.println("Nombre: " + dato1);  
    System.out.println("Numero: " + dato2);  
} else {  
    System.out.println("Cancelado");  
}
```



Código para buscar un contacto.

```
case 1: (e.getSource() == botonBuscar) {  
    // Obtener el nombre del contacto a buscar  
    String nombreBuscar = JOptionPane.showInputDialog(null, "Introduce el nombre que desea: ");  
  
    // Verificar que se haya ingresado un nombre  
    if (nombreBuscar.isEmpty()) {  
        JOptionPane.showMessageDialog(this, "Debe ingresar un nombre.");  
        return;  
    }  
  
    // Buscar el contacto en la lista  
    int indiceEncontrado = -1;  
  
    for (int i = 0; i < contactos.size(); i++) {  
        if (contactos.get(i).getNombre().equals(nombreBuscar)) {  
            indiceEncontrado = i;  
            break;  
        }  
    }  
  
    // Mostrar el resultado de la búsqueda  
    if (indiceEncontrado != -1) {  
        listaContactos.setSelectedIndex(indiceEncontrado);  
        listaContactos.ensureIndexIsVisible(indiceEncontrado);  
    } else {  
        JOptionPane.showMessageDialog(this, "No se encontró ningún contacto con el nombre especificado.");  
    }  
}
```

Código para borrar un contacto.

```
// Borrar el contacto deseado  
} else if (e.getSource() == botonBorrar) {  
    // Obtener el índice del contacto seleccionado  
    int indiceSeleccionado = listaContactos.getSelectedIndex();  
  
    // Comprobar que se haya seleccionado un contacto  
    if (indiceSeleccionado == -1) {  
        JOptionPane.showMessageDialog(this, "Debe seleccionar un contacto.");  
        return;  
    }  
  
    // Borrar el contacto de la lista  
    contactos.remove(indiceSeleccionado);  
  
    // Actualizar la lista de contactos  
    actualizarLista();  
}
```




Código para editar un contacto.

```
} else if (e.getSource() == botonEditar) {  
    // Obtener el índice del contacto seleccionado  
    int indiceSeleccionado = listaContactos.getSelectedIndex();  
  
    // Obtener el contacto seleccionado  
    Contacto contactoSeleccionado = contactos.get(indiceSeleccionado);  
  
    // Crear una ventana emergente para editar el contacto seleccionado  
    JTextField campoNombreEditar = new JTextField(contactoSeleccionado.getNombre());  
    JTextField campoNotasEditar = new JTextField(contactoSeleccionado.getNotas());  
    Object[] camposEditar = {  
        "Nombre:", campoNombreEditar,  
        "Numero:", campoNotasEditar  
    };  
};
```