

Universidad San Carlos de Guatemala

Facultad de ingeniería

Escuela Ciencias y sistemas



Galicia Urrutia, Miguel Ricardo

Introducción a la programación y Computación 1

202201117

24 marzo de 2023

Verificación de datos

Registro de datos

Cargar imagen

```

@Override
    public void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        int resultado;
        Foto f = new Foto();
        FileNameExtensionFilter filter = new FileNameExtensionFilter("Imagenes (JPG u PNG)", "jpg", "png");
        v.jfchCargarFoto.setFileFilter(filter);
        resultado = v.jfchCargarFoto.showOpenDialog(null);
        if (JFileChooser.APPROVE_OPTION == resultado) {
            fichero = v.jfchCargarFoto.getSelectedFile();
            try {
                ImageIcon icon = new ImageIcon(fichero.toString());
                Icon icono = new ImageIcon(icon.getImage().getScaledInstance(lblImagen.getWidth(), lblImagen.getHeight(), Image.SCALE_DEFAULT));
                lblImagen.setText(null);
                lblImagen.setIcon(icono);
            } catch (Exception e) {
                JOptionPane.showMessageDialog(parentComponent, "Error al abrir la imagen", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

```

Buscar Usuarios

```
    }  
    public void llenarCbox(){  
        datausers dta = new datausers();  
        ArrayList<String> cK = dta.getCorreos();  
        for(String correo: cK){  
            emails.addItem(index: correo);  
        }  
    }  
    public void busqueda(){  
        //esto es lo que se va a buscar en el combobox  
        datausers dta = new datausers();  
        if(existencia(email.getEmails())){  
            declararUsuarios(nombre: dta.getNm().get(index: indice), apellido: dta.getLstNm().get(index: indice), cui: dta.getDPIs().get(index: indice), celu: dta.getcel().get(index: indice), nacionalidad:  
        )else JOptionPane.showMessageDialog(parentComponent: null, message: "Correo no encontrado", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);  
    }  
    public boolean existencia(String email){  
        datausers dta = new datausers();  
        if(dta.getCorreos().contains(email)){  
            indice=dta.getCorreos().indexOf(email);  
        }else return false;  
        return true;  
    }  
    public void declararUsuarios(String nombre,String apellido,String cui, String celu, String nacionalidad, String rol, String genero,int dia,String mes,int año,String usuario){  
        Name.setText(text: nombre);  
        LastName.setText(text: apellido);  
        DPI.setText(text: cui);  
        Celular.setText(text: celu);  
        Roll.setText(text: rol);  
        Genero.setText(text: genero);  
        Nacionalidad.setText(text: nacionalidad);  
        Usuario.setText(text: usuario);  
        Nacimiento.setText(text: dia+"/"+mes+"/"+año);  
    }  
}
```

Cambiar datos de usuarios (menú administración)

```
    public void cambiarUsuario(){  
        datausers dta = new datausers();  
        int cambiardatos = JOptionPane.showConfirmDialog(parentComponent: null, message: "¿Está seguro de que desea cambiar los datos?", title: "Confirmar acción", optionType: JOptionPane.YES_NO_OPTION);  
        if (cambiardatos == JOptionPane.YES_OPTION){  
            Object[] opciones = {"Nombre", "Apellido", "DPI", "Celular"};  
            int selectedOption = JOptionPane.showOptionDialog(parentComponent: null, message: "Seleccione una opción:", title: "Seleccionar región", optionType: JOptionPane.DEFAULT_OPTION, messageOptions:  
            switch(selectedOption){  
                case 0:  
                    String newName= JOptionPane.showInputDialog(message: "Ingrese el nuevo nombre");  
                    dta.getNm().set(index: indice, element: newName);  
                    break;  
                case 1:  
                    String newLastName= JOptionPane.showInputDialog(message: "Ingrese el nuevo apellido");  
                    dta.getLstNm().set(index: indice, element: newLastName);  
                    break;  
                case 2:  
                    String newDPI= JOptionPane.showInputDialog(message: "Ingrese el nuevo nombre");  
                    dta.getDPIs().set(index: indice, element: newDPI);  
                    break;  
                case 3:  
                    String newCelular= JOptionPane.showInputDialog(message: "Ingrese el nuevo celular");  
                    dta.getNm().set(index: indice, element: newCelular);  
                    break;  
            }  
        }  
    }  
}
```

Eliminar datos de usuarios (menú administración)

```
    public void eliminar(){  
        int eliminar = JOptionPane.showConfirmDialog(parentComponent: null, message: "¿Está seguro que desea Eliminar al usuario?", title: "Confirmar acción", optionType: JOptionPane.YES_NO_OPTION);  
        if (eliminar == JOptionPane.YES_OPTION){  
            if (eliminar == JOptionPane.YES_OPTION){  
                int dobleC = JOptionPane.showConfirmDialog(parentComponent: null, message: "¿Confirma que esta accion es irreversible?", title: "Confirmar acción", optionType: JOptionPane.YES_NO_OPTION);  
                if (dobleC == JOptionPane.YES_OPTION){  
                    datausers dta = new datausers();  
                    dta.getCorreos().remove(index: indice);  
                    dta.getContras().remove(index: indice);  
                    dta.getNm().remove(index: indice);  
                    dta.getLstNm().remove(index: indice);  
                    dta.getDPIs().remove(index: indice);  
                    dta.getgenero().remove(index: indice);  
                    dta.getcel().remove(index: indice);  
                    dta.getrol().remove(index: indice);  
                    dta.getnacionalidad().remove(index: indice);  
                    dta.getusr().remove(index: indice);  
                    Name.setText(text: "");  
                    LastName.setText(text: "");  
                    DPI.setText(text: "");  
                    Celular.setText(text: "");  
                    Roll.setText(text: "");  
                    Genero.setText(text: "");  
                    Nacionalidad.setText(text: "");  
                    Usuario.setText(text: "");  
                    Nacimiento.setText(text: "");  
                    emails.setSelectedItem(anObject: "");  
                    emails.remove(index: indice);  
                }  
            }  
        }  
    }  
}
```

Manejo de Departamentos y municipios

```
public void BuscarDepartamento() {  
    if(contieneValor( array: getAreaNorte(), departamento: getjComboBox1())) {  
        region.setText( text: "Norte");  
        codeRegion.setText( text: "NT");  
    }  
    else if(contieneValor( array: getAreaMetropolitana(), departamento: getjComboBox1())) {  
        region.setText( text: "Metropolitana");  
        codeRegion.setText( text: "M");  
    }  
    else if(contieneValor( array: getAreaNOR(), departamento: getjComboBox1())) {  
        region.setText( text: "Nororientado");  
        codeRegion.setText( text: "NO");  
    }  
    else if(contieneValor( array: getAreaNOC(), departamento: getjComboBox1())) {  
        region.setText( text: "Noroccidente");  
        codeRegion.setText( text: "NOC");  
    }  
    else if(contieneValor( array: getAreaSURO(), departamento: getjComboBox1())) {  
        region.setText( text: "Surorientado");  
        codeRegion.setText( text: "SO");  
    }  
    else if(contieneValor( array: getAreaSUC(), departamento: getjComboBox1())) {  
        region.setText( text: "Suroccidente");  
        codeRegion.setText( text: "SOC");  
    }  
}
```

```
int indiceDepartamento=buscarMuni( municipio:municipioBuscar, listado:lista);  
switch(indiceDepartamento){  
    case 0:  
        buscarDtpto.setText( text: "Alta Verapaz");  
        codeMunicipio.setText( text: "NT");  
        break;  
    case 1:  
        buscarDtpto.setText( text: "Baja Verapaz");  
        codeMunicipio.setText( text: "NT");  
        break;  
    case 2:  
        buscarDtpto.setText( text: "Chimaltenango");  
        codeMunicipio.setText( text: "M");  
        break;  
}
```

cotización según departamentos y región

Precio según región

```
public void precios() {
    String regionSelecta = getjComboBox1();

    switch (regionSelecta) {
        case "Metropolitana":
            // código a ejecutar si la región es Metropolitana
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltAreaMetro());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclAreaMetro());
            break;
        case "Norte":
            // código a ejecutar si la región es Norte
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltNorte());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclNorte());
            break;
        case "Nororiente":
            // código a ejecutar si la región es Nororiente
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltAreaNOR());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclAreaNOR());
            break;
        case "Noroccidente":
            // código a ejecutar si la región es Noroccidente
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltAreaNOC());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclAreaNOC());
            break;
        case "Suroccidente":
            // código a ejecutar si la región es Suroccidente
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltSuro());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclSuro());
            break;
        case "Suroriente":
            // código a ejecutar si la región es Suroriente
            PrecioEstandar.setText("Precio estandar (entrega de 3-5 dias habiles): Q"+getPrecioDfltSUC());
            PrecioEspecial.setText("Precio especial (entrega de 1-2 dias habiles): Q"+getPrecioSpclSUC());
            break;
        default:
            // código a ejecutar si la región no coincide con ninguna de las anteriores
            JOptionPane.showMessageDialog(parentComponent, null, message: "Ingrese una Region", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
            break;
    }
}
```

Buscar región según departamento

```
public void ElegirRegion() {
    String regionSelecta = getjComboBox1();

    switch (regionSelecta) {
        case "Metropolitana":
            // código a ejecutar si la región es Metropolitana
            String metro = Arrays.toString(areas.AreaMetropolitana);
            Departamentos.setText(text: metro);
            break;
        case "Norte":
            // código a ejecutar si la región es Norte
            String Norte = Arrays.toString(areas.AreaNorte);
            Departamentos.setText(text: Norte);
            break;
        case "Nororiente":
            // código a ejecutar si la región es Nororiente
            String NOR = Arrays.toString(areas.AreaNOR);
            Departamentos.setText(text: NOR);
            break;
        case "Noroccidente":
            // código a ejecutar si la región es Noroccidente
            String NOC = Arrays.toString(areas.AreaNOC);
            Departamentos.setText(text: NOC);
            break;
        case "Suroccidente":
            // código a ejecutar si la región es Suroccidente
            String SURO = Arrays.toString(areas.AreaSURO);
            Departamentos.setText(text: SURO);
            break;
        case "Suroriente":
            // código a ejecutar si la región es Suroriente
            String SUC = Arrays.toString(areas.AreaSUC);
            Departamentos.setText(text: SUC);
            break;
        default:
            // código a ejecutar si la región no coincide con ninguna de las anteriores
            JOptionPane.showMessageDialog(parentComponent, null, message: "Ingrese una Region", title: "Error", messageType: JOptionPane.ERROR_MESSAGE);
            break;
    }
}
```