

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-06-12

Prueba caja blanca

Requisito funcional 1: Validar credenciales de administrador

El programa deberá tener una interfaz gráfica donde el administrador ingrese y valide sus credenciales. El usuario ingresará al sistema mediante un usuario y contraseña previamente creados. Si el usuario ingresa erróneamente su nombre o contraseña, debe volver a intentarlo y a su vez se visualizará un mensaje que dirá "usuario o contraseña no válido". Después de 3 intentos de inicio de sesión consecutivos fallidos, se bloqueará el ingreso de credenciales por 5 minutos.

1. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

Validación de credenciales

```
if (intentosFallidos >= MAX_INTENTOS) {
    bloquearLogin(); //se termina el algoritmo
}

if (!this.LoginEstado) {
    intentosFallidos++;

    JOptionPane.showMessageDialog(null,
        "X Usuario o contraseña no válidos. Intento "
        + intentosFallidos + " de " + MAX_INTENTOS);

    vista_validacion.txtContrasena.setText("");
    vista_validacion.txtUsuario.setText("");
    vista_validacion.txtUsuario.requestFocusInWindow();

} else {
    vista_validacion.dispose();
    iniciarPrograma();
    System.out.println("Acceso Exitoso");
}

public boolean validarUsuarios(){
    return this.nombreUsuario.equals("Veronica")&&this.password.equals("1234");
}

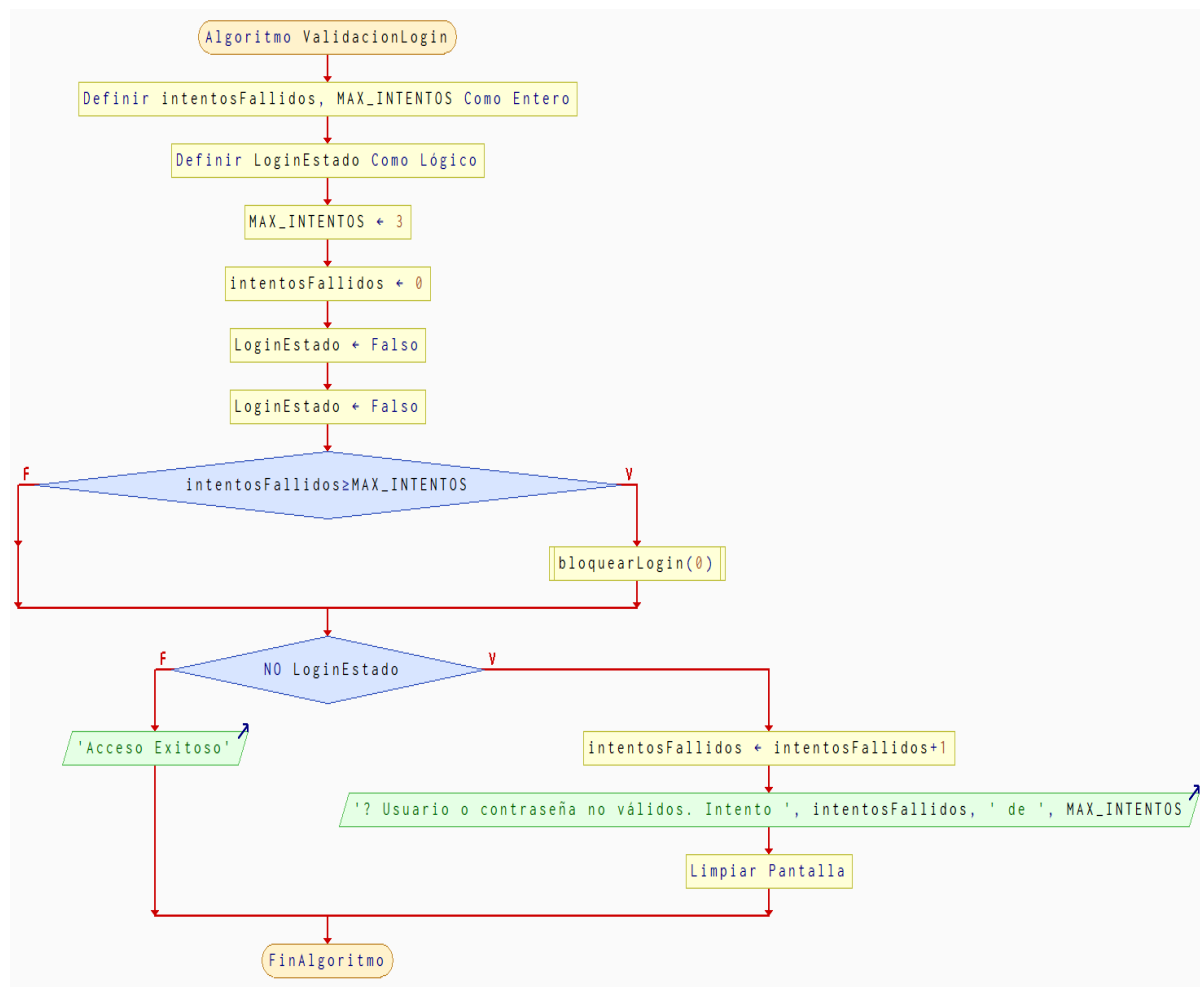
public boolean compararPass(String Pass1, String Pass2){
    return Pass1.equals(Pass2);
}
```

CÓDIGO DE AUTENTIFICACIÓN CON MONGODB

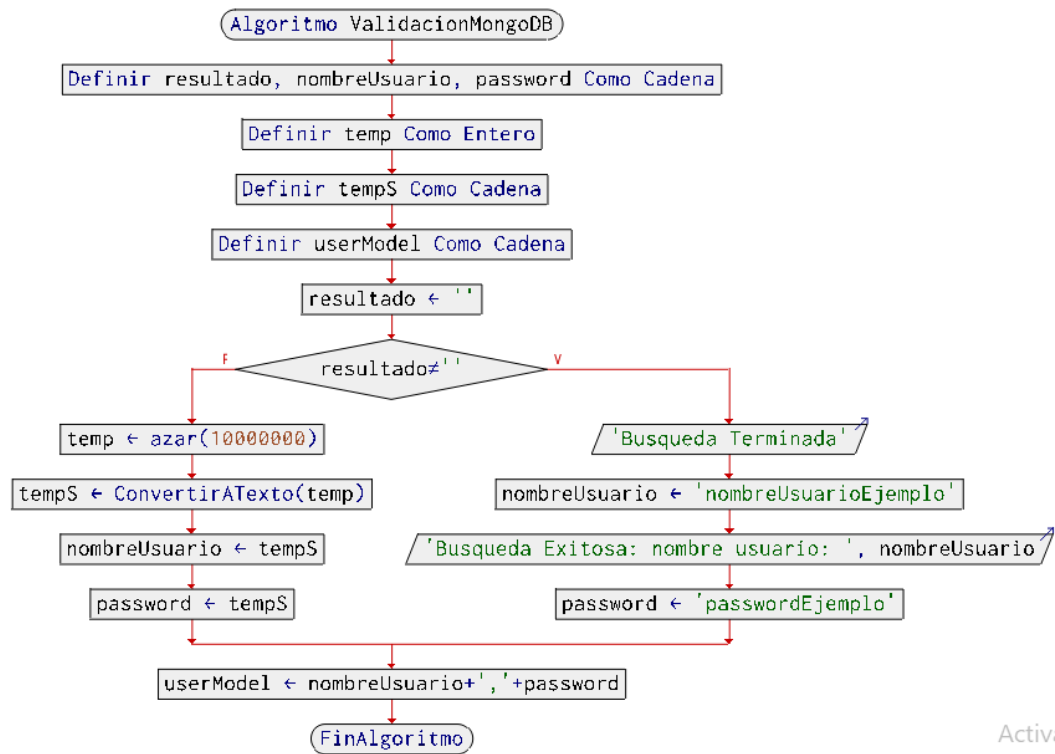
```
45 if (resultado != null) {
46     System.out.println("Busqueda Terminada ");
47     userModel.setNombreUsuario(resultado.getString("nombreUsuario"));
48     System.out.println("Busqueda Exitosa: nombre usuario: "+userModel.getNombreUsuario());
49     userModel.setPassword(resultado.getString("password"));
50 }
51 else{
52
53     int temp = random.nextInt(100000000);
54     String tempS = String.valueOf(temp);
55     userModel.setNombreUsuario(tempS);
56     userModel.setPassword(tempS);
57 }
58 }
```

2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



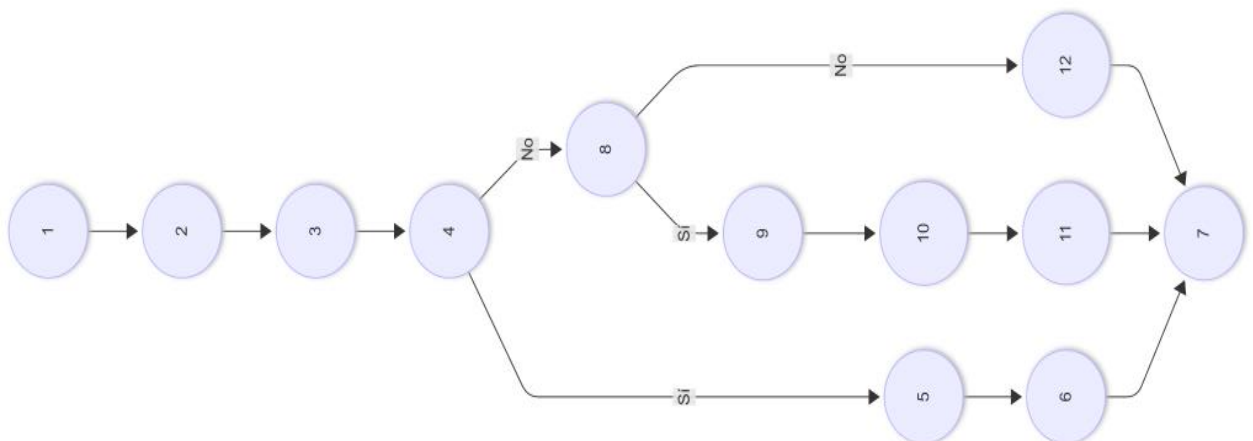
DF DE CONEXIÓN CON MONGODB



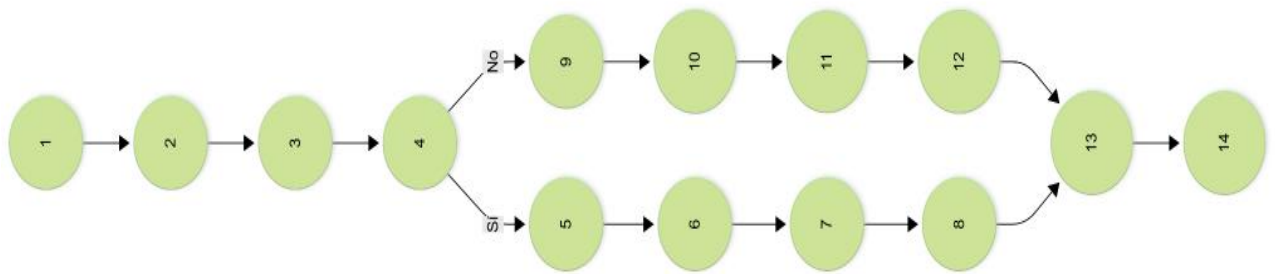
3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral

Validación de credenciales



GF DE CONEXIÓN CON MONGODB



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4
RUTAS

Validación de credenciales

R1: 1-2-3-4-5-6-7

R1: 1-2-3-4-8-9-10-11-7

R3: 1-2-3-4-8-12-7

RUTAS DE CONEXIÓN CON MONGODB

R1: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 13 - 14

R2: 1 - 2 - 3 - 4 - 9 - 10 - 11 - 12 - 13 - 14

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

Validación de credenciales

$$V(G) = 2 + 1$$

$$V(G) = 3$$

$$V(G) = A - N + 2$$

$$V(G) = 13 - 12 + 2$$

$$V(G) = 3$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

CONEXIÓN CON MONGODB

$$V(G) = 1 + 1$$

$$V(G) = 2$$

$$V(G) = A - N + 2$$

$$V(G) = 14 - 14 + 2$$

$$V(G) = 2$$

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-06-12

Prueba caja blanca

Requisito funcional 2: Validar credenciales de administrador

Permitir que el usuario registrado pueda visualizar su información de perfil (nombre de usuario) y modificar su nombre de usuario y contraseña.

Verificación de credenciales:

1. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

```
else if (e.getSource() == vistadatos.BtnCambioUser) {
    accion = AccionConfirmacion.CAMBIO_USER;
    iniciarConfirmacion();
}

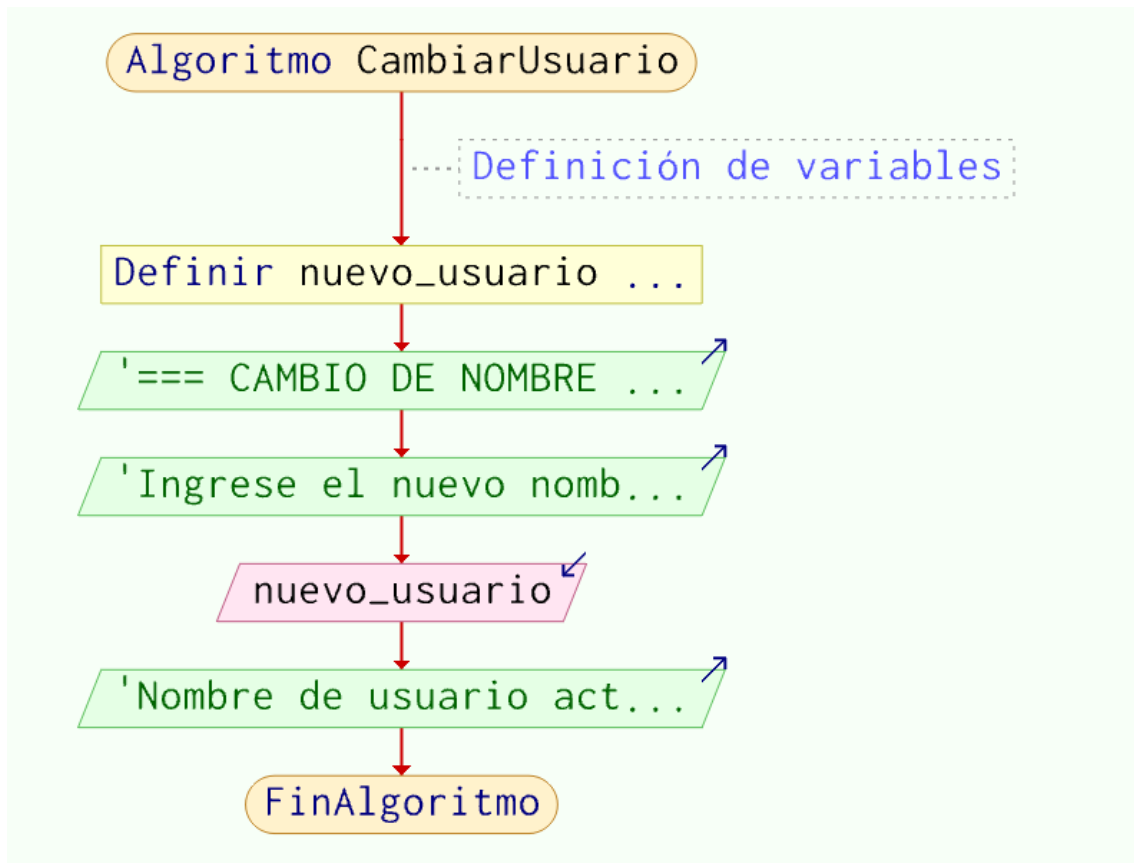
else if (e.getSource() == vistadatos.BtnCambioPassword) {
    String passNueva = vistadatos.jPasswordnew.getText();
    String passConfir = vistadatos.jPasswordConfir.getText();
    String passActual = vistadatos.jPasswordactual.getText();

    boolean passConf = passNueva.equals(passConfir);
    boolean passVer = modelo.getPassword().equals(passActual);

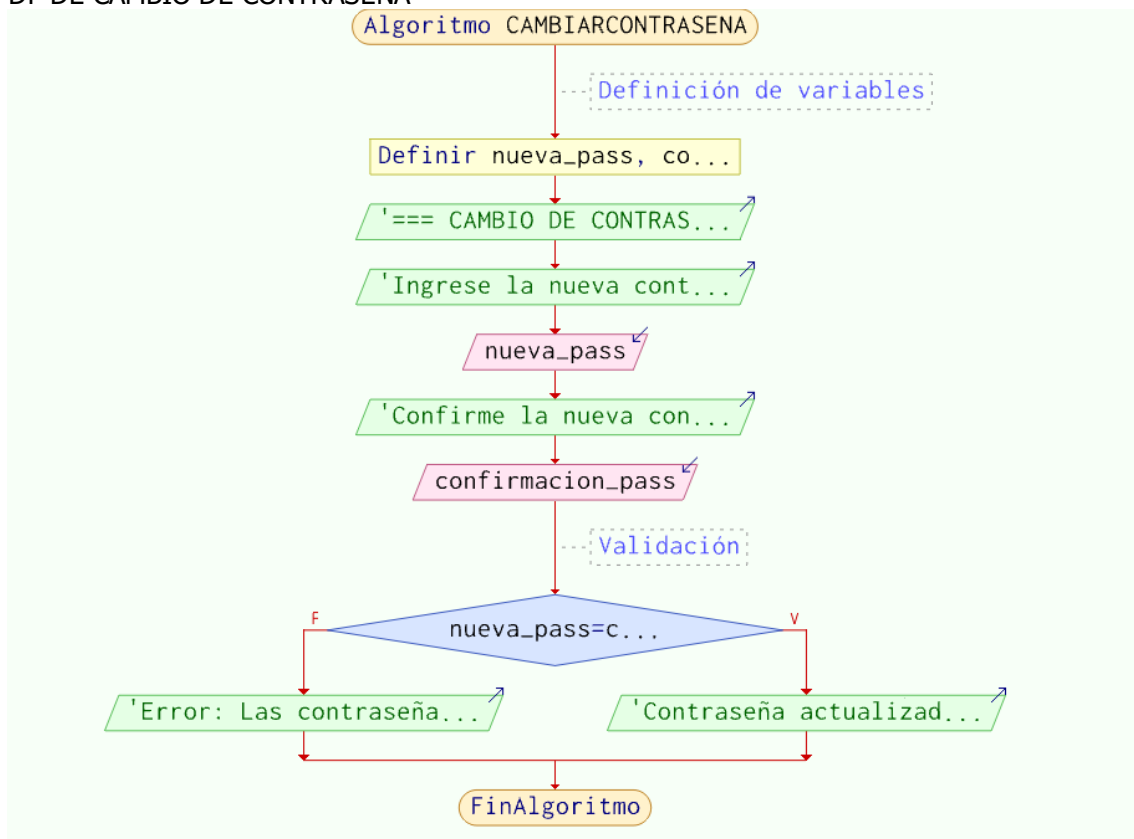
    if (!passVer) {
        JOptionPane.showMessageDialog(null, "La contraseña actual es incorrecta.");
    } else if (!passConf) {
        JOptionPane.showMessageDialog(null, "Las nuevas contraseñas no coinciden.");
    } else {
        // Ambas condiciones se cumplieron
        nuevaPasswordPendiente = passConfir; // almacenar temporalmente
        accion = AccionConfirmacion.CAMBIO_PASSWORD;
        iniciarConfirmacion(); // abrir ventana de confirmación
    }
}
```

2. DIAGRAMA DE FLUJO (DF)

DF DE CAMBIO DE USUARIO

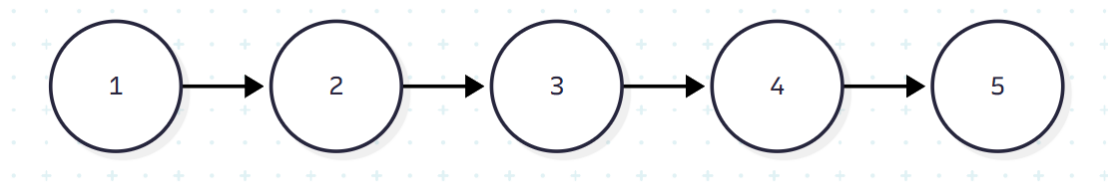


DF DE CAMBIO DE CONTRASEÑA

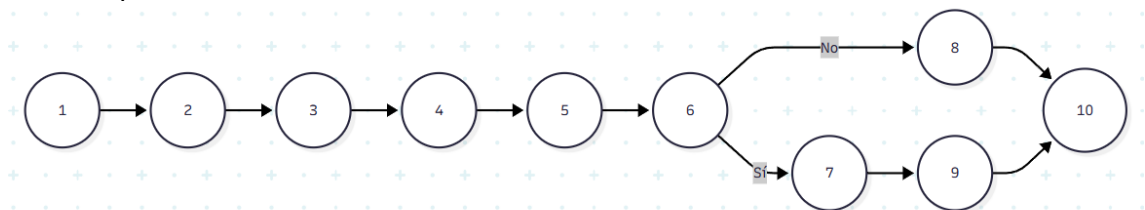


3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral



Grafo de proceso de cambio de usuario



Grafo de proceso de cambio de contraseña

4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS GRAFO CAMBIO DE USUARIO

R1: 1-2-3-4-5

RUTAS GRAFO CAMBIO DE CONTRASEÑA

R1: 1-2-3-4-5-6-8-10

R2: 1-2-3-4-5-6-7-9-10

5. COMPLEJIDAD CICLOMÁTICA

CAMBIO DE USUARIO

- $V(G) = 0 + 1$
 $V(G) = 1$
- $V(G) = A - N + 2$
 $V(G) = 4 - 5 + 2$
 $V(G) = 1$

CAMBIO DE CONTRASEÑA

- $V(G) = 1 + 1$
 $V(G) = 2$
- $V(G) = A - N + 2$
 $V(G) = 10 - 10 + 2$
 $V(G) = 2$

DONDE:

P: Número de nodos predicho

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-06-12

Prueba caja blanca

Requisito funcional 3: Plantillas de contratos con texto editable: Civil y Laboral.

El programa deberá tener una opción de "agregar contrato" donde se pueda seleccionar una plantilla de contrato con cuadros de texto editable. Al utilizar la opción de "nuevo contrato" se podrá escoger una plantilla e ingresar datos (que dependen del tipo de contrato) en cuadros de texto editables que podrán ser visualizados en una interfaz gráfica.

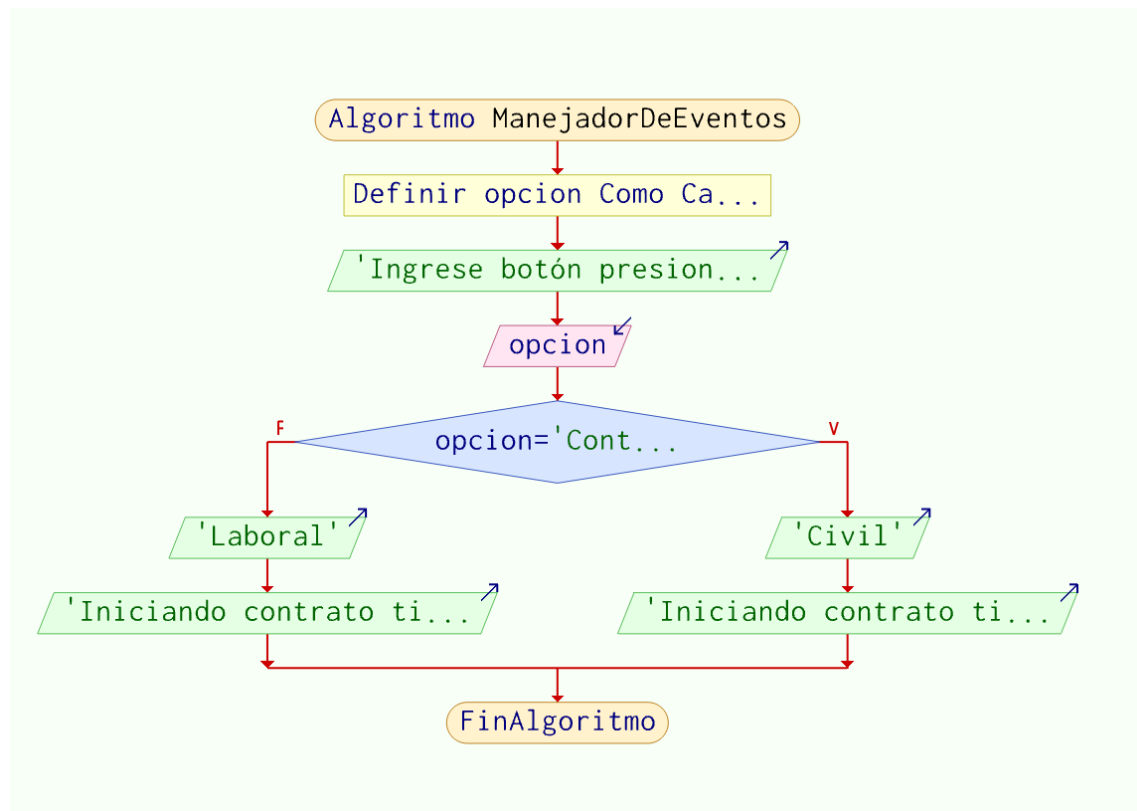
1. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

```
}  
else if(e.getSource()==programa.BtnContratoCivil){  
    System.out.println("Civil");  
    iniciarContrato(0);  
}  
else if(e.getSource()==programa.BtnContratoLaboral){  
    System.out.println("Laboral");  
    iniciarContrato(1);  
}
```

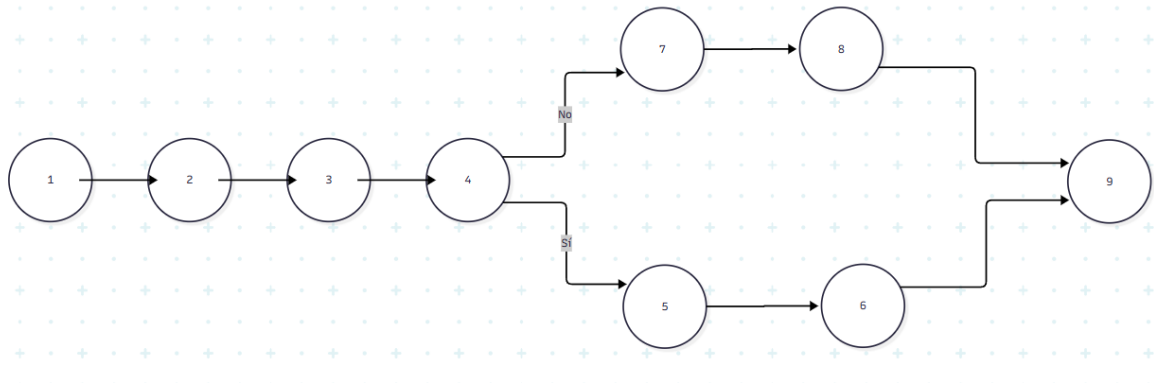
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4
RUTAS

R1: 1-2-3-4-7-8-9

R2: 1-2-3-4-5-6-9

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

$$V(G) = 1 + 1$$

$$V(G) = 2$$

$$V(G) = A - N + 2$$

$$V(G) = 9 - 9 + 2$$

$$V(G) = 2$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-07-10

Requisito funcional 4: Guardar los datos ingresados en las plantillas.

El programa deberá tener una opción de "agregar contrato" donde se pueda seleccionar una plantilla de contrato con cuadros de texto editable. Al utilizar la opción de "nuevo contrato" se podrá escoger una plantilla e ingresar datos (que dependen del tipo de contrato) en cuadros de texto editables que podrán ser visualizados en una interfaz gráfica.

1. CÓDIGO FUENTE

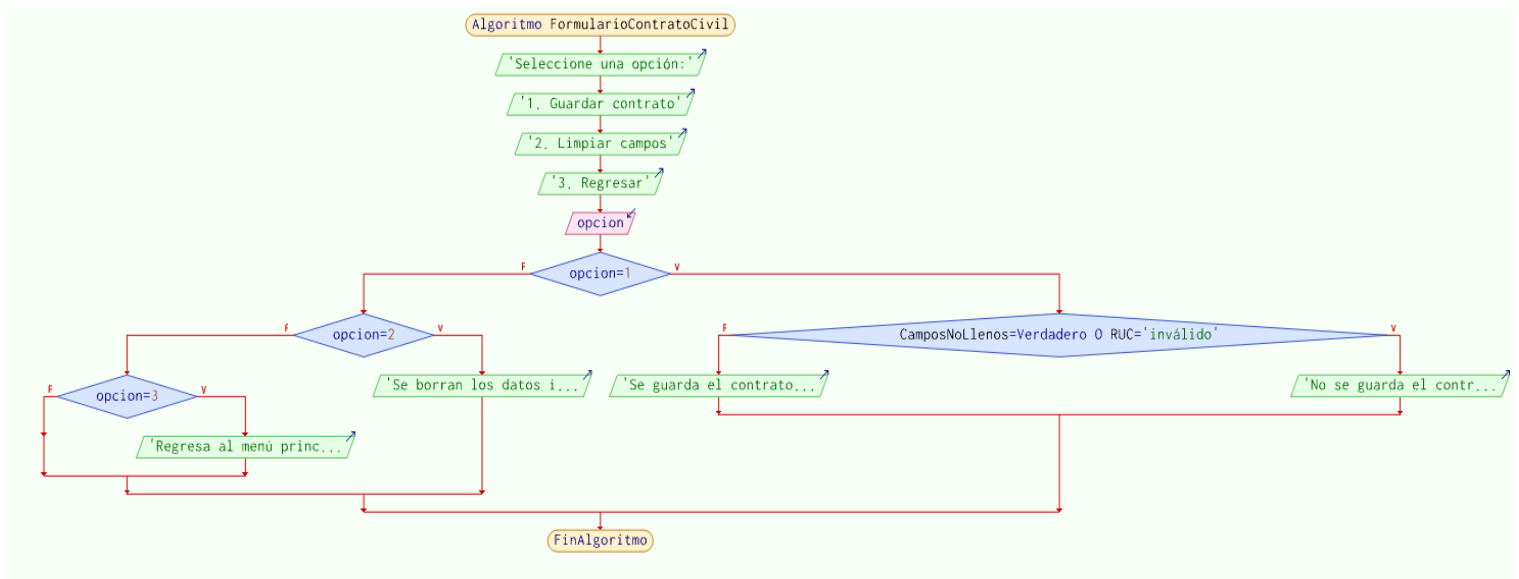
Pegar el trozo de código fuente que se requiere para el caso de prueba

```
public ContratoCivilControlador(FormularioContratoCivil formCivil, LoginControlador lgnCtrl, Usuario userModel) {  
    this.formCivil = formCivil;  
    this.lgnCtrl = lgnCtrl;  
    this.mongoCiv = new MongoDBCCivil(userModel);  
    this.formCivil.BtnGuardar.addActionListener(this);  
    agregarValidacionAutomatica();  
    Validacion();  
}
```

```
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == formCivil.BtnGuardar) {  
        mongoCiv.conectarMongo(obtenerTextosCampos(), 0);  
        System.out.println("Contrato Civil guardado en MongoDB");  
        ObjectId idNuevo = mongoCiv.insertarYObtenerId(obtenerTextosCampos());  
        System.out.println("Documento insertado con _id=" + idNuevo);  
  
        try {  
            ContratoPdfGenerator pdfGen = new ContratoPdfGenerator();  
            pdfGen.generarContratoPDF(mongoCiv, idNuevo);  
            System.out.println("Contrato PDF generado correctamente.");  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
    }  
}
```

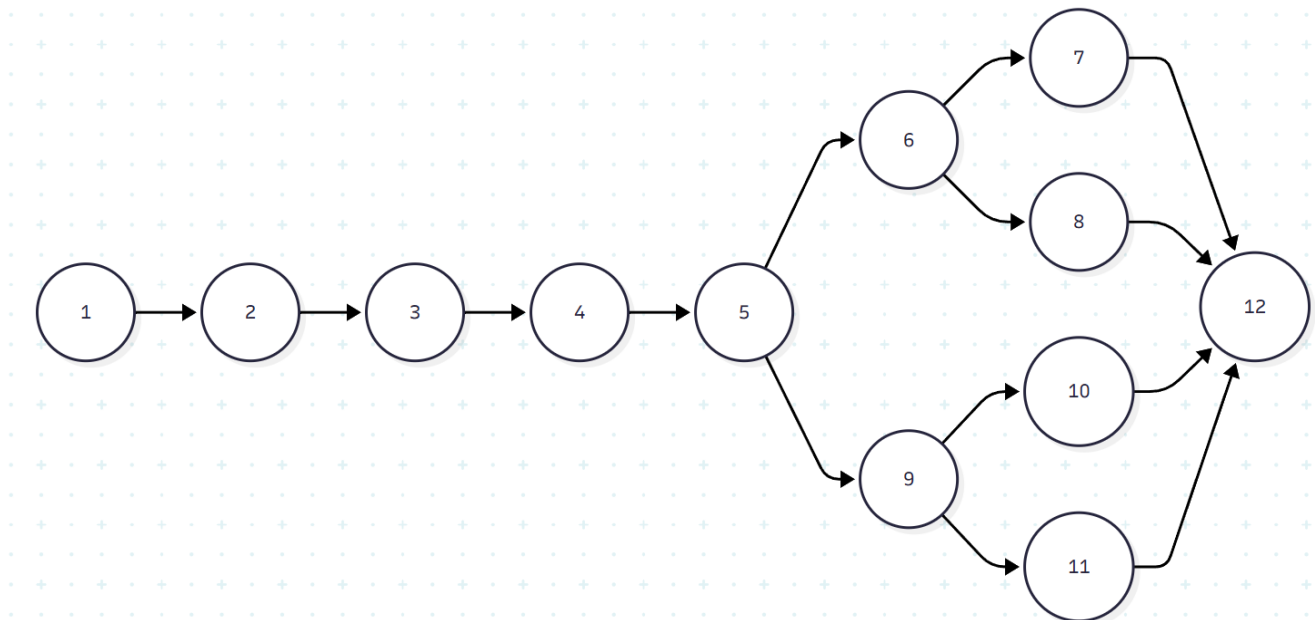
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 3
RUTAS

R1: 1-2-3-4-5-6-7-12

R2: 1-2-3-4-5-6-8-12

R3: 1-2-3-4-5-9-10-12

R4: 1-2-3-4-5-9-11-12

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

$$V(G) = 3 + 1$$

$$V(G) = 4$$

$$V(G) = A - 9 + 2$$

$$V(G) = 14 - 12 + 2$$

$$V(G) = 4$$

DONDE:

P: Número de nodos prediado

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-07-20

Requisito funcional 5: Búsqueda de contratos registrados

El programa deberá tener una opción para buscar contratos con una barra de búsqueda y filtros

1. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == this.menuBusqueda.BtnBusqueda && this.menuBusqueda.BtnRadioRUC.isSelected()) {
        String[] nombresCampos = this.mongoDBbusqueda
            .buscarContrato(this.menuBusqueda.txtBusqueda.getText(), 0);

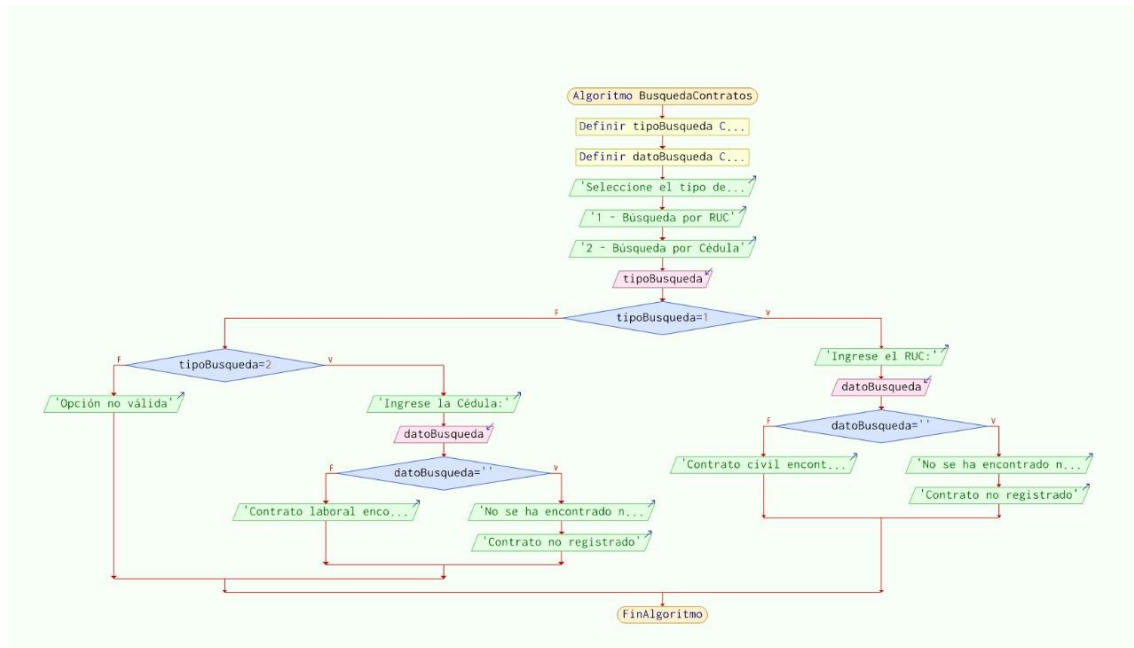
        if (nombresCampos == null) {
            JOptionPane.showMessageDialog(null,
                "No se ha encontrado ningún contrato civil con ese RUC.",
                "Contrato no registrado",
                JOptionPane.WARNING_MESSAGE);
        } else {
            this.cambiarCampos(nombresCampos, 0);
            this.formCivBus.setVisible(true);
            this.formCivBus.setResizable(false);
        }
    }

    if (e.getSource() == this.menuBusqueda.BtnBusqueda && this.menuBusqueda.BtnRadioCedula.isSelected()) {
        String[] nombresCamposLab = this.mongoDBbusqueda
            .buscarContrato(this.menuBusqueda.txtBusqueda.getText(), 1);

        if (nombresCamposLab == null) {
            JOptionPane.showMessageDialog(null,
                "No se ha encontrado ningún contrato laboral con esa cédula.",
                "Contrato no registrado",
                JOptionPane.WARNING_MESSAGE);
        } else {
            this.cambiarCampos(nombresCamposLab, 1);
            this.formLabBus.setVisible(true);
            this.formLabBus.setResizable(false);
        }
    }
}
```

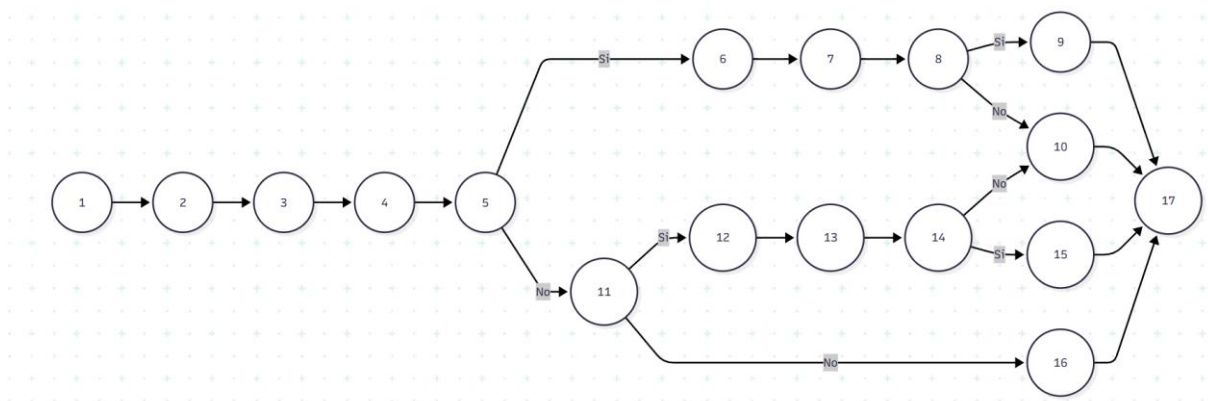
2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4
RUTAS

R1: 1-2-3-4-5-6-7-8-9-17

R2: 1-2-3-4-5-6-7-8-10-17

R3: 1-2-3-4-5-11-12-13-14-10-17

R4: 1-2-3-4-5-11-12-13-14-15-17

R5: 1-2-3-4-5-11-16-17

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

$$V(G) = 3+1$$

$$V(G) = 4$$

$$V(G) = A-N + 2$$

$$V(G) = 19-17+2$$

$$V(G)=4$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Prueba de Caja Blanca

“Diseño de sistema de gestión de contratos para una plaza comercial”

Integrantes:

Gallardo Vega Santiago Jose,
Pérez Díaz David Ismael,
Zambrano Cajas Isabela Valentina.

Fecha: 2025-07-22

Requisito funcional 6: Editar y eliminar contratos registrados

El administrador deberá poder modificar el contenido contratos y eliminar contratos mediante una opcion que aparecerá en cada contrato

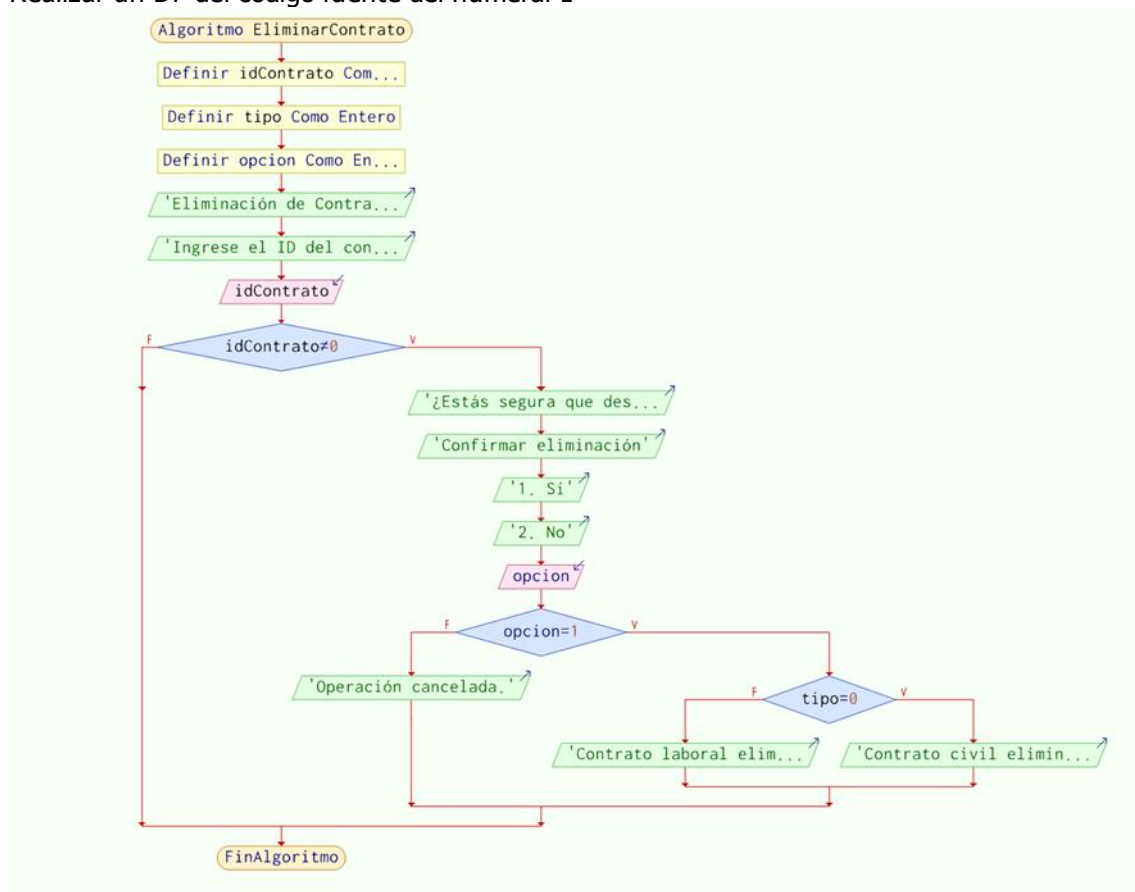
1. CÓDIGO FUENTE para eliminación de contratos

Pegar el trozo de código fuente que se requiere para el caso de prueba

```
public void eliminarContrato(ObjectId idContrato, int tipo) {  
    if (idContrato == null) {  
        JOptionPane.showMessageDialog(null,  
            "No se encontró el contrato para eliminar.",  
            "Error", JOptionPane.ERROR_MESSAGE);  
        return;  
    }  
  
    // 📌 Ventana de confirmación personalizada  
    int opcion = JOptionPane.showConfirmDialog(  
        null,  
        "¿Estás segura que deseas eliminar este contrato?\nEsta acción no se puede deshacer.",  
        "Confirmar eliminación",  
        JOptionPane.YES_NO_OPTION,  
        JOptionPane.WARNING_MESSAGE  
    );  
  
    if (opcion != JOptionPane.YES_OPTION) {  
        JOptionPane.showMessageDialog(null, "Operación cancelada.");  
        return;  
    }  
  
    try {  
        if (tipo == 0) {  
            MongoDBCivil servicio = new MongoDBCivil(userModel);  
            servicio.getCollection("ContratosCivil")  
                .deleteOne(new Document("_id", idContrato));  
            JOptionPane.showMessageDialog(null,  
                "✅ Contrato civil eliminado correctamente.");  
            formCivBus.dispose();  
        } else {  
            MongoDBCLaboral servicio = new MongoDBCLaboral(userModel);  
            servicio.getCollection("ContratosLaboral")  
                .deleteOne(new Document("_id", idContrato));  
            JOptionPane.showMessageDialog(null,  
                "✅ Contrato laboral eliminado correctamente.");  
            formLabBus.dispose();  
        }  
    } catch (Exception ex) {  
        JOptionPane.showMessageDialog(null,  
            "❌ Error al eliminar el contrato: " + ex.getMessage(),  
            "Error", JOptionPane.ERROR_MESSAGE);  
        ex.printStackTrace();  
    }  
}
```

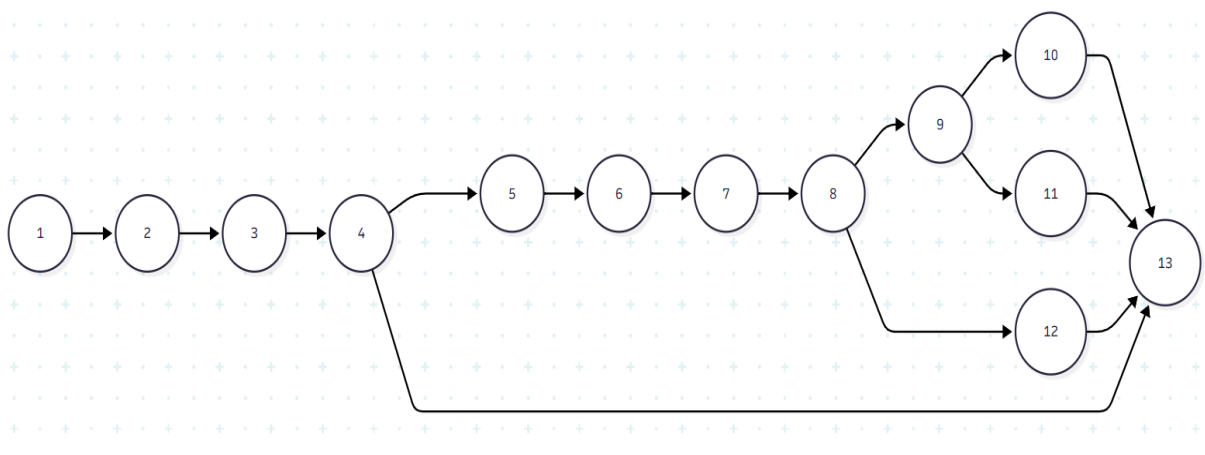
2. DIAGRAMA DE FLUJO (DF) eliminación de contratos

Realizar un DF del código fuente del numeral 1



3. GRAFO DE FLUJO (GF) eliminación de contratos

Realizar un GF en base al DF del numeral



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico) eliminación de contratos

Determinar en base al GF del numeral 4
RUTAS

R1: 1-2-3-4-5-6-7-8-9-10-13

R2: 1-2-3-4-5-6-7-8-9-11-13

R3: 1-2-3-4-5-6-7-8-12-13

R4: 1-2-3-4-13

5. COMPLEJIDAD CICLOMÁTICA eliminación de contratos

Se puede calcular de las siguientes formas:

$$V(G) = 3+1$$

$$V(G) = 4$$

$$V(G) = A-N + 2$$

$$V(G) = 15-13+2$$

$$V(G)=4$$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

6. CÓDIGO FUENTE para edición de contratos

Pegar el trozo de código fuente que se requiere para el caso de prueba


```

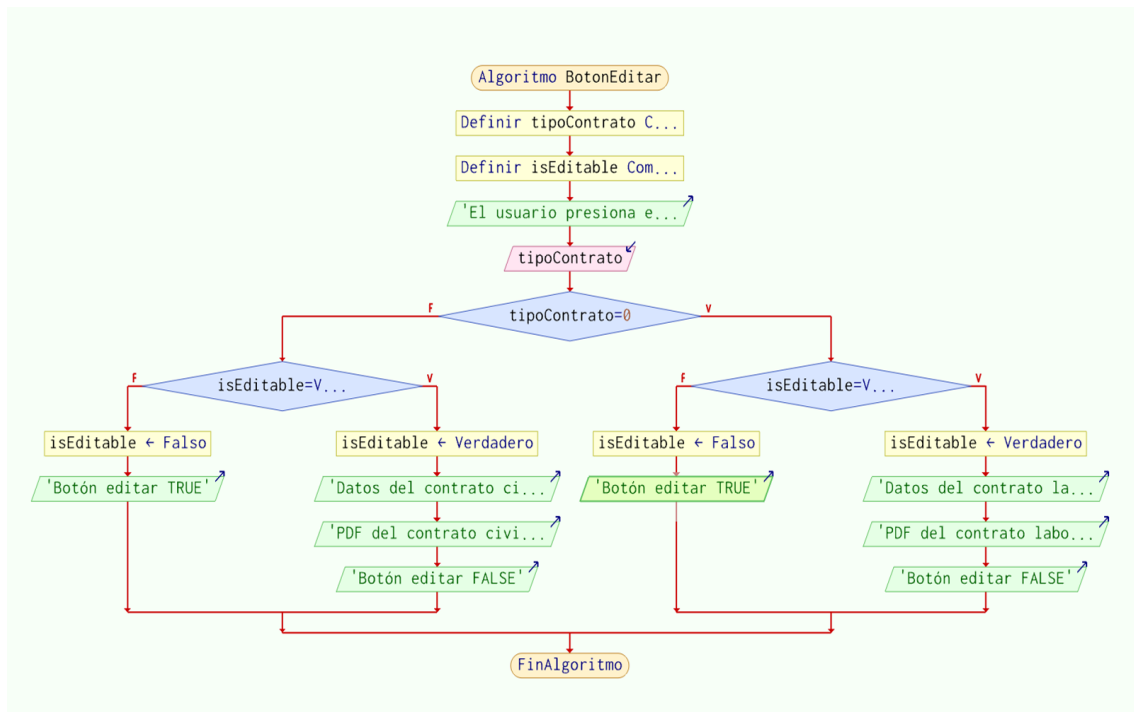
if (e.getSource() == formCivBus.BtnEditarCivil) {
    if(!isEditable){
        isEditable = setEditable(0,isEditable);
        System.out.println("Boton Editar TRUE");
    }
    else{
        this.mongoDBbusqueda.updateMongoDB(0, idCivilBuscado, obtenerTextosCampos(0));
        isEditable = setEditable(0,isEditable);
        try {
            MongoDBCcivil servicio = new MongoDBCcivil(userModel);
            new ContratoPdfGeneratoCiv().generarContratoPDF(servicio, idCivilBuscado);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Error al generar el PDF civil.",
                "Error", JOptionPane.ERROR_MESSAGE);
            ex.printStackTrace();
        }
        System.out.println("Boton editar FALSE");
    }
}

if (e.getSource() == formLabBus.BtnEditarLab) {
    if(!isEditable){
        isEditable = setEditable(1,isEditable);
    }
    else{
        this.mongoDBbusqueda.updateMongoDB(1, idLaboralBuscado, obtenerTextosCampos(1));
        isEditable = setEditable(1,isEditable);
        try {
            MongoDBCLaboral servicio = new MongoDBCLaboral(userModel);
            new ContratoPdfGeneratoLab().generarContratoPDF(servicio, idLaboralBuscado);
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, "Error al generar el PDF laboral.",
                "Error", JOptionPane.ERROR_MESSAGE);
            ex.printStackTrace();
        }
    }
}
}

```

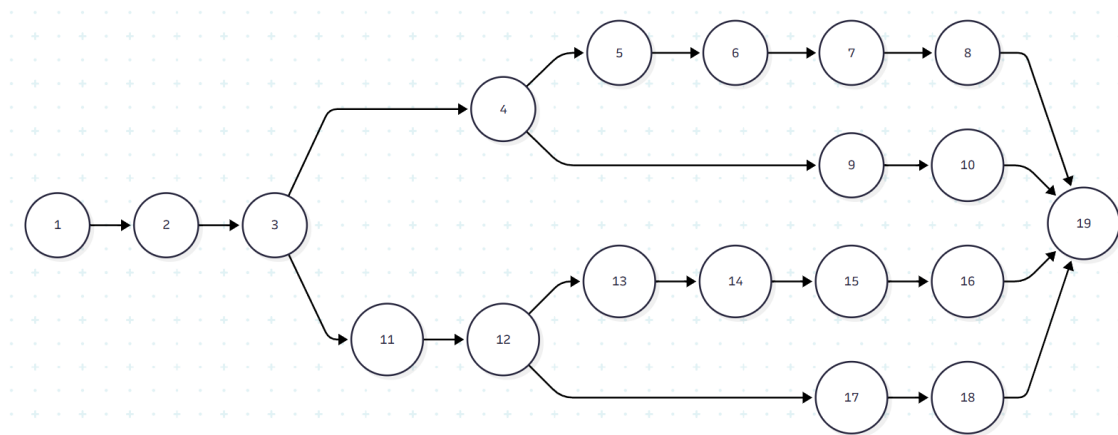
7. DIAGRAMA DE FLUJO (DF) edición de contratos

Realizar un DF del código fuente del numeral 1



8. GRAFO DE FLUJO (GF) edición de contratos

Realizar un GF en base al DF del numeral 4



9. IDENTIFICACIÓN DE LAS RUTAS (Camino básico) edición de contratos

Determinar en base al GF del numeral 4
RUTAS

R1: 1-2-3-4-5-6-7-8-9-19
R2: 1-2-3-4-9-10-19
R3: 1-2-3-11-12-13-14-15-16-19
R4: 1-2-3-11-12-17-18-19

10. COMPLEJIDAD CICLOMÁTICA edición de contratos

Se puede calcular de las siguientes formas:

$$V(G) = 3+1$$

$$V(G) = 4$$

$$V(G) = A-N + 2$$

$$V(G) = 21-19+2$$

$$V(G)=4$$

DONDE:

P: Número de nodos prediado

A: Número de aristas

N: Número de nodos