

# 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

```
        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();

            if (intentosFallidos >= MAX_INTENTOS) {
                bloquearLogin();
            }
        } else {
            vista_validacion.dispose();
            iniciarPrograma();
            System.out.println("Acceso Exitoso");
        }

    }).start();

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

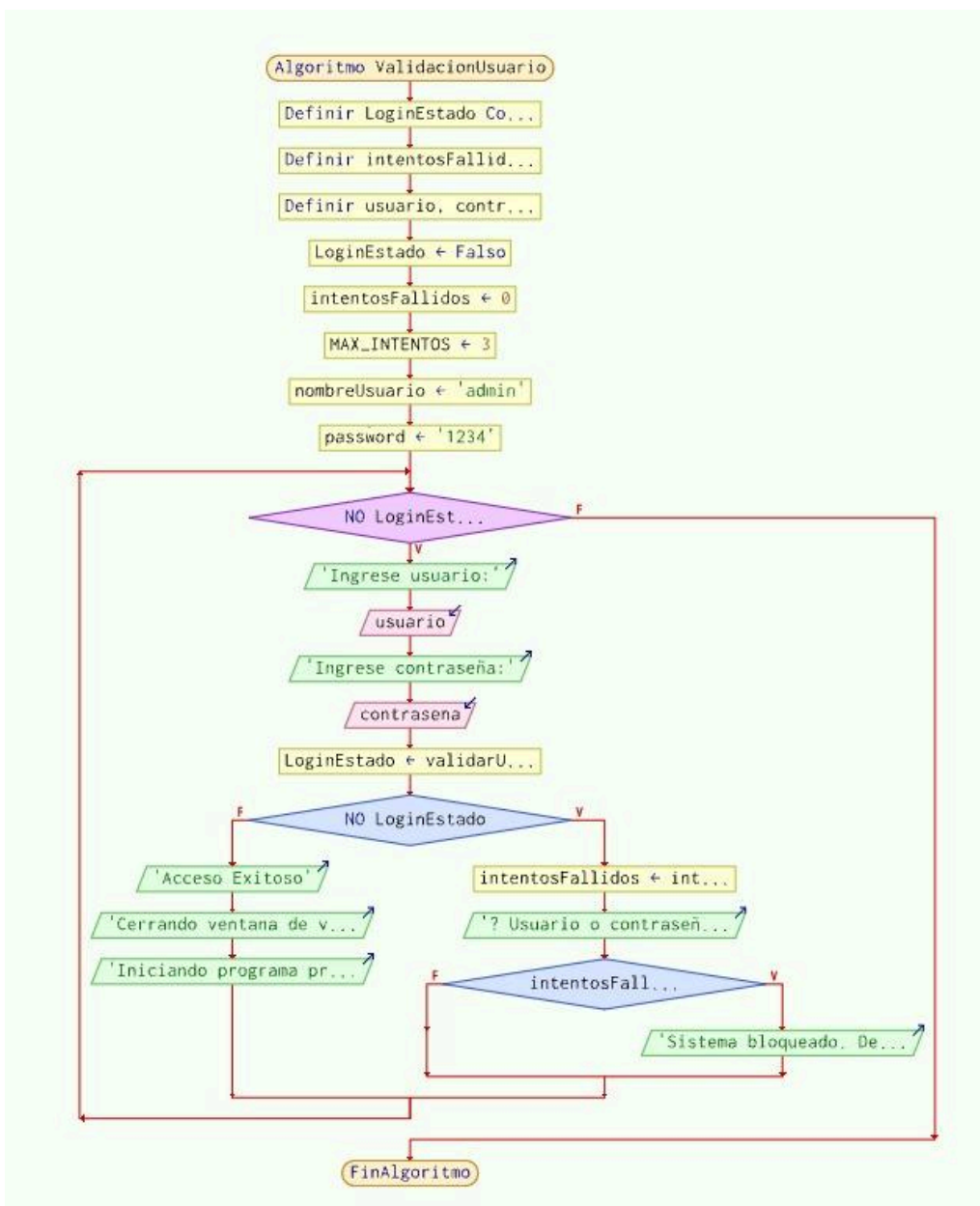
    public boolean compararPass(String Pass1, String Pass2) {
        return Pass1.equals(Pass2);
    }
}
```

## CÓDIGO DE AUTENTICACIÓN CON MONGODB

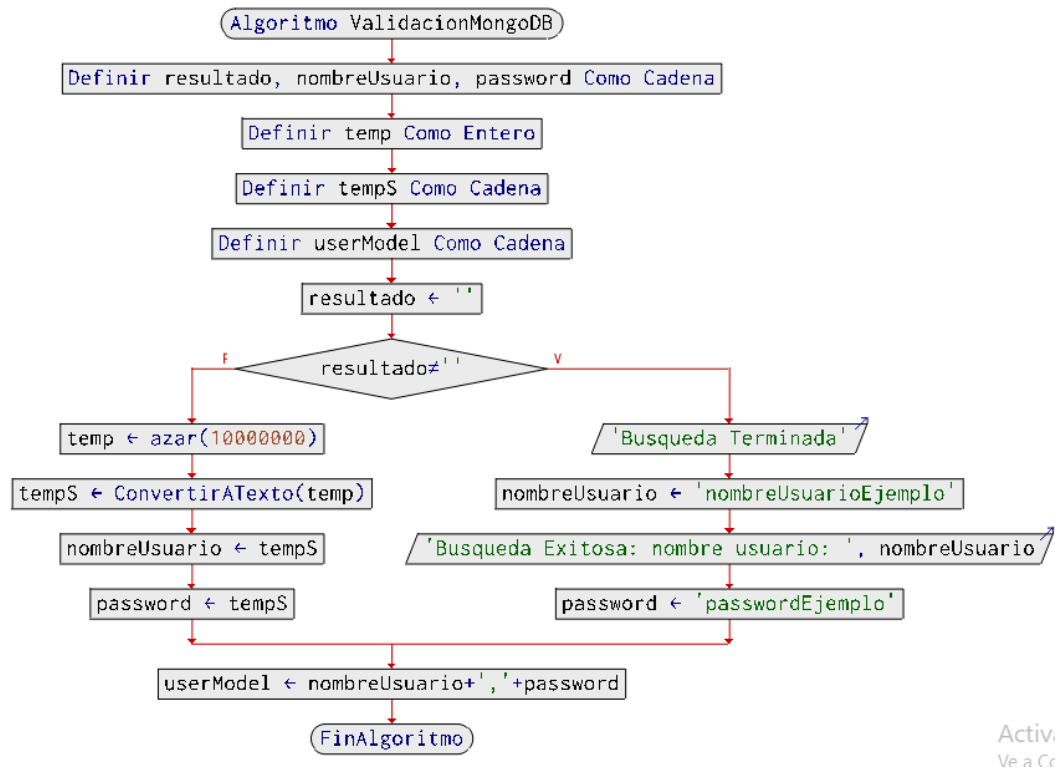
```
45 if (resultado != null) {
46     System.out.println("Busqueda Terminada ");
47     userModel.setNombreUsuario(resultado.getString("nombreUsuario"));
48     System.out.println("Busqueda Exitosa: nombre usuario: "+userModel.getNombreUsuario());
49     userModel.setPassword(resultado.getString("password"));
50 }
51 else{
52
53     int temp = random.nextInt(10000000);
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

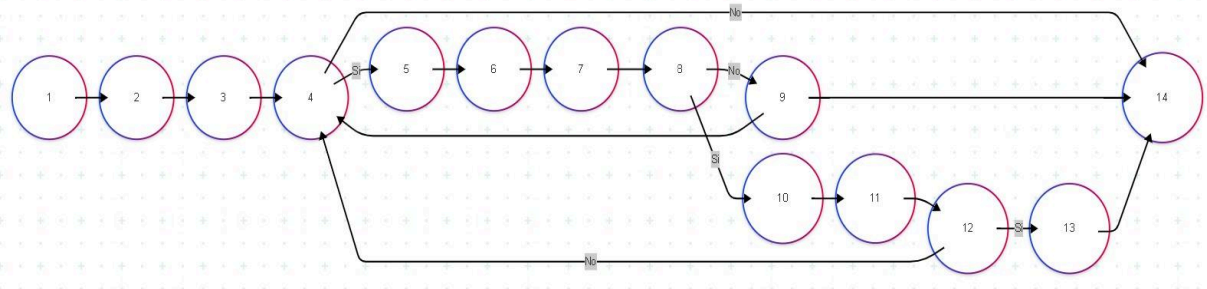


## DF DE CONEXIÓN CON MONGODB

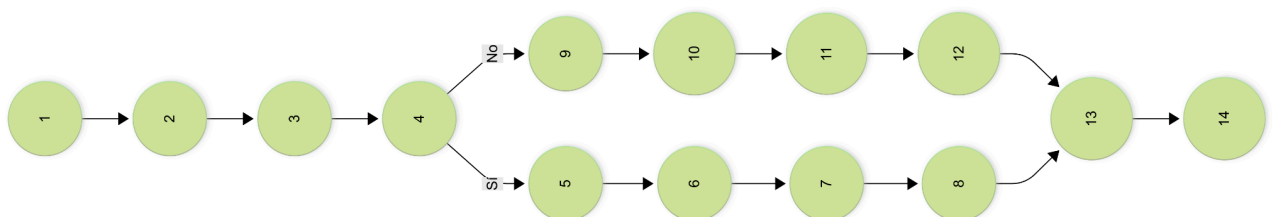


### 3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral



## GF DE CONEXIÓN CON MONGODB



### 4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4  
RUTAS

R1: 1-2-3-4-14  
R2: 1-2-3-4-5-6-7-8-9-14  
R3: 1-2-3-4-5-6-7-8-10-11-12-13-14  
R4: 1-2-3-4-9-14  
R5: 1-2-3-4-12-13-141

RUTAS

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:

$$V(G) = 4+1$$
$$V(G)= 5$$

$$V(G) = A - N + 2$$
$$V(G)= 17-14+2$$
$$V(G)=5$$

DONDE:

P: Número de nodos prediado

A: Número de aristas

N: Número de nodos

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

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