

# Prueba de Caja Blanca

---

***“GESTIÓN DE FIDEICOMISOS:”***

**Integrantes:**

**Kevin Andino**

**Luis Calle**

**Denisse Quishpe**

**Madellyn**

**Tasipanta**

**Fecha: 2025/12/07**

**CONTROL DE VERSIONAMIENTO DE PRUEBAS CB**

| <b>Versiòn</b>  | <b>Fecha</b> | <b>Responsable</b>                   | <b>Aprobado por</b> |
|-----------------|--------------|--------------------------------------|---------------------|
| PCB_V1.1.1.docx | 08/12/2025   | Denisse Quishpe<br>Madelyn Tasipanta |                     |

## Prueba caja blanca SPRINT 1

### 1. CÓDIGO FUENTE

#### REQ001: Validación de credenciales

Crear un formulario para que el administrador pueda validar sus credenciales (usuario y contraseña). El sistema contará con un máximo de 3 intentos, en caso de agotar el máximo de intentos. Se bloqueará el acceso.

#### REQ002: Revisión de perfil de usuario

El administrador accede al módulo “Mi Perfil”, donde puede cambiar su nombre de usuario y también puede cambiar su contraseña ingresando la contraseña actual, la nueva contraseña y la confirmación. Se aplican validaciones obligatorias como contraseña fuerte, coincidencia de confirmación y verificación de la contraseña actual, y al finalizar, si todo está correcto, se realiza un guardado exitoso con un mensaje de confirmación.

```
31  @Override
32  public void actionPerformed(ActionEvent e) {
33      if (estaBloqueado()) {
34          manejarBloqueo();
35          return;
36      }
37      String usuario = vista.getUsuario();
38      String contrasena = vista.getContrasena();
39      if (validarCredenciales(usuario, contrasena)) {
40          intentosFallidos = 0;
41          tiempoBloqueoHasta = 0;
42          vista.mostrarMensaje("¡Bienvenido!", "Inicio Exitoso", JOptionPane.INFORMATION_MESSAGE);
43          vista.dispose();
44          DashboardView dashboardView = new DashboardView();
45          new DashboardController(dashboardView); // Inicia el controlador del dashboard
46          dashboardView.setVisible(true);
47      } else {
48          intentosFallidos++;
49          vista.mostrarMensaje("Usuario o Contraseña incorrectos. Intento: " + intentosFallidos,
50                              "Error de Login", JOptionPane.ERROR_MESSAGE);
51          if (intentosFallidos >= MAX_INTENTOS) {
52              activarBloqueo();
53              manejarBloqueo();
54          }
55      }
56  }
```

*De LoginController.java*

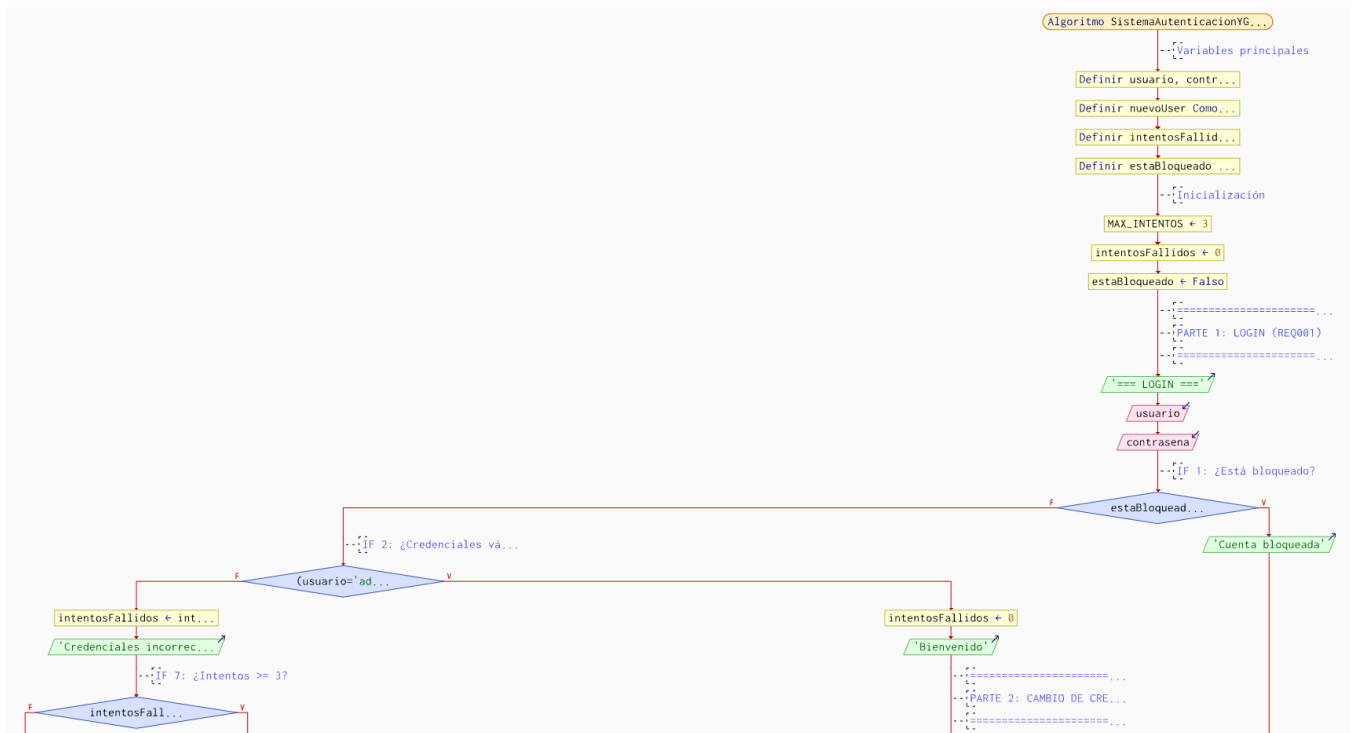
```

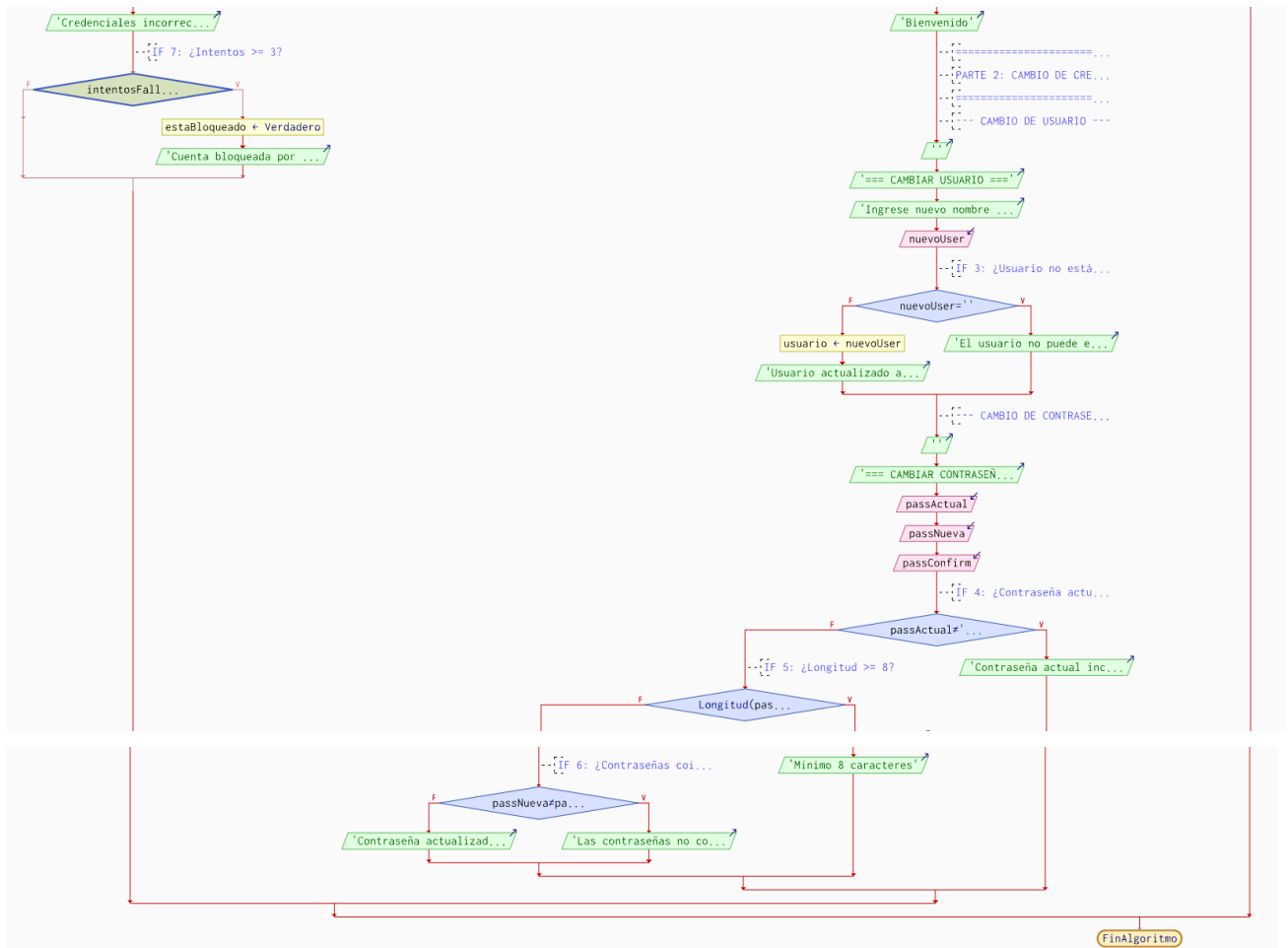
179 |         if (!nuevoUser.isEmpty()) {
180 |             gestor.setUsuario(nuevoUser);
181 |             JOptionPane.showMessageDialog(dialog, "Usuario actualizado a: " + nuevoUser);
182 |         } else {
183 |             JOptionPane.showMessageDialog(dialog, "El usuario no puede estar vacío.", "Error", JOptionPane.ERROR_MESSAGE);
184 |         }
    |
192 | if (!gestor.verificarContraseñaActual(passActual)) {
193 |     JOptionPane.showMessageDialog(dialog, "Contraseña actual incorrecta.", "Error", JOptionPane.ERROR_MESSAGE);
194 |     return;
195 | }
196 | if (passNueva.length() < 8) {
197 |     JOptionPane.showMessageDialog(dialog, "La contraseña debe tener mínimo 8 caracteres.", "Seguridad", JOptionPane.WARNING_M
198 |     return;
199 | }
200 | if (!passNueva.equals(passConfirm)) {
201 |     JOptionPane.showMessageDialog(dialog, "Las contraseñas no coinciden.", "Error", JOptionPane.ERROR_MESSAGE);
202 |     return;
203 | }

```

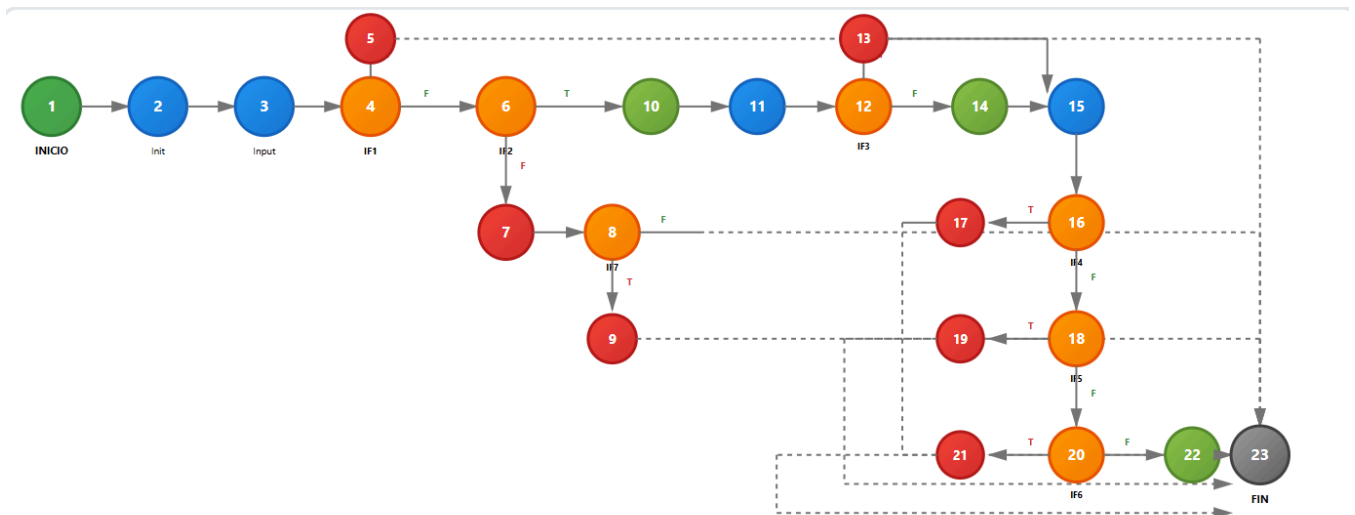
*De DashController.java*

## 2. DIAGRAMA DE FLUJO (DF)





### 3. GRAFO DE FLUJO (GF)



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

Determinar en base al GF del numeral 4

R1:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 23$

R2:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 23$

R3:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 23$

R4:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 23$

R5:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 23$

R6:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 18 \rightarrow 19 \rightarrow 23$

R7:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 18 \rightarrow 20 \rightarrow 21 \rightarrow 23$

R8:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 18 \rightarrow 20 \rightarrow 22 \rightarrow 23$

#### 5. COMPLEJIDAD CICLOMÁTICA

DONDE

- P: Número de nodos predicado
- A: Número de aristas
- N: Número de nodos

Se puede calcular de las siguientes formas:

Nodos (N): 23

Nodos Predicados (P): 7

Aristas (A): 30

- $V(G) = P + 1$
- $V(G) = 7 + 1 = 8$
- $V(G) = A - N + 2$
- $V(G) = 29 - 23 + 2 = 8$