

Prueba de Caja Blanca

“GESTIÓN DE FIDEICOMISOS:”

Integrantes:

Kevin Andino

Luis Calle

Denisse Quishpe

Madellyn

Tasipanta

Fecha: 2025/12/08

CONTROL DE VERSIONAMIENTO DE PRUEBAS CB

Versión	Fecha	Responsable	Aprobado por
PCB_V1.1.1.docx	08/12/2025	Denisse Quishpe Madellyn Tasipanta	

Prueba caja blanca REQ002

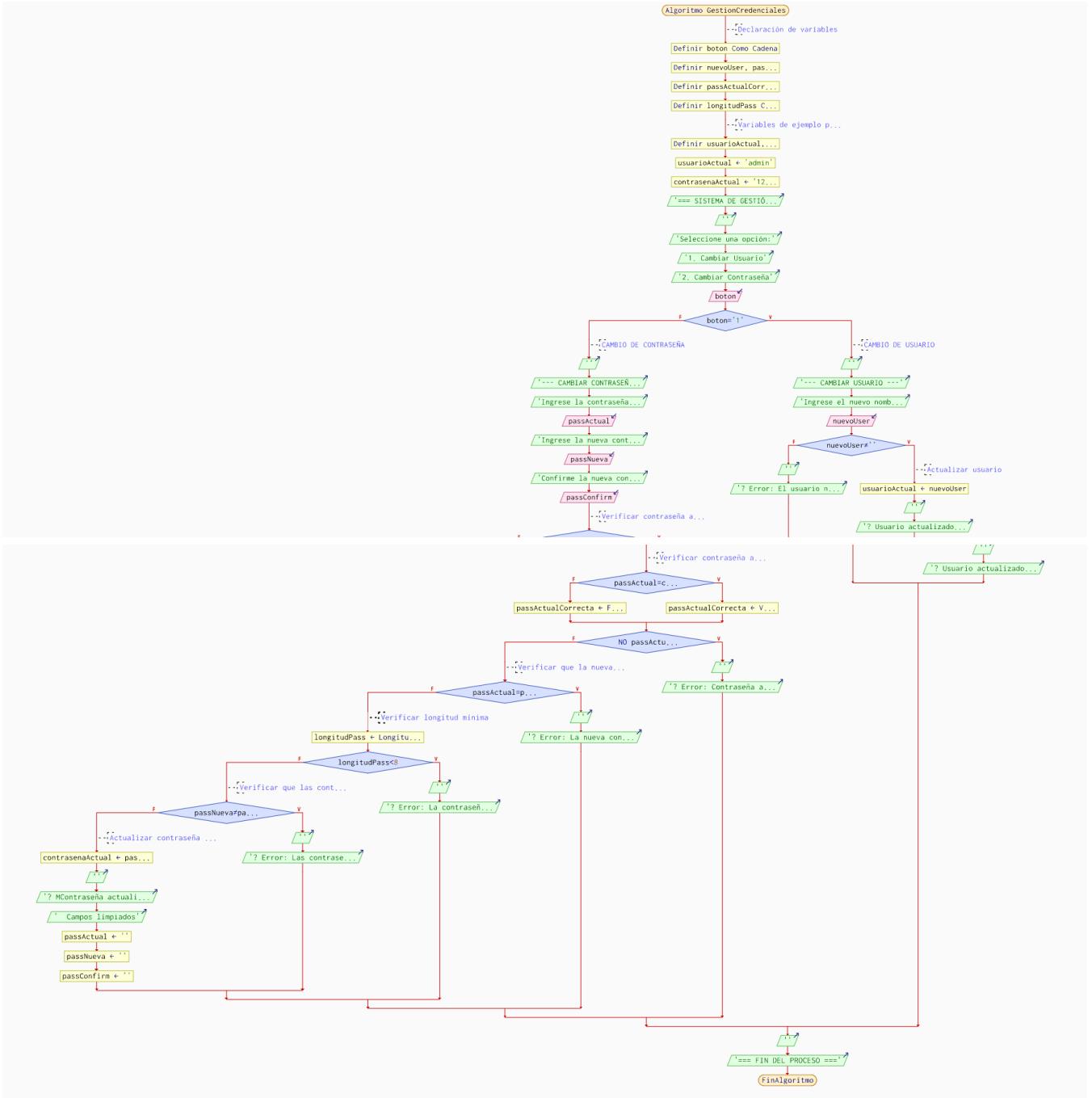
1. CÓDIGO FUENTE

REQ002: 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

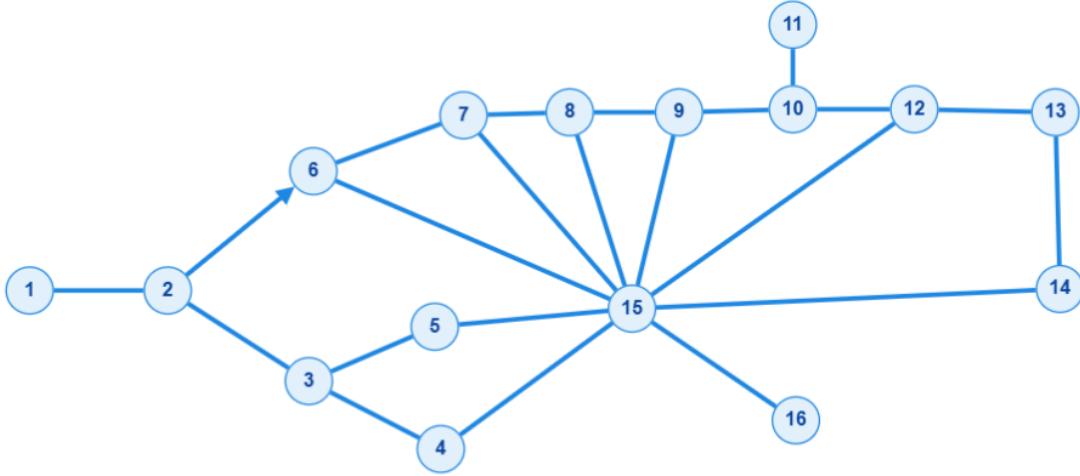
```
171     private void abrirCambioCredenciales() {
172         CambiarCredencialesView dialog = new CambiarCredencialesView(view);
173         GestorAutenticacion gestor = GestorAutenticacion.getInstancia();
174
175         dialog.setUsuarioInput(gestor.getUsuarioActual());
176
177         dialog.getBtnCambiarUsuario().addActionListener(evt -> {
178             String nuevoUser = dialog.getUsuarioInput().trim();
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             }
185         });
186
187         dialog.getBtnCambiarPass().addActionListener(evt -> {
188             String passActual = new String(dialog.getPassActual());
189             String passNueva = new String(dialog.getPassNueva());
190             String passConfirm = new String(dialog.getPassConfirmacion());
191
192             dialog.getBtnCambiarPass().addActionListener(evt -> {
193                 String passActual = new String(dialog.getPassActual());
194                 String passNueva = new String(dialog.getPassNueva());
195                 String passConfirm = new String(dialog.getPassConfirmacion());
196
197                 if (!gestor.verificarContrasenaActual(passActual)) {
198                     JOptionPane.showMessageDialog(dialog, "Contraseña actual incorrecta.", "Error", JOptionPane.ERROR_MESSAGE);
199                     return;
200                 }
201                 if (passNueva.length() < 8) {
202                     JOptionPane.showMessageDialog(dialog, "La contraseña debe tener mínimo 8 caracteres.", "Seguridad", JOptionPane.WARNING_MESSAGE);
203                     return;
204                 }
205                 if (!passNueva.equals(passConfirm)) {
206                     JOptionPane.showMessageDialog(dialog, "Las contraseñas no coinciden.", "Error", JOptionPane.ERROR_MESSAGE);
207                     return;
208                 }
209
210                 gestor.setContrasena(passNueva);
211                 JOptionPane.showMessageDialog(dialog, "¡Contraseña actualizada con éxito!");
212                 dialog.limpiarCamposPass();
213             });
214
215             dialog.getBtnRegresar().addActionListener(evt -> dialog.dispose());
216             dialog.setVisible(true);
217         });
218     }
```

De DashboardController.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

RUTAS

R1: 1-2-3-4-15-16

R2: 1-2-3-5-15-16

R3: 1-2-6-7-15-16

R4: 1-2-6-8-9-15-16

R5: 1-2-6-8-10-11-15-16

R6: 1-2-6-8-10-12-13-15-16

R7: 1-2-6-8-10-12-14-15-16

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): 16

Nodos Predicados (P): 7

Aristas (A): 22

- $V(G) = P + 1$
- $V(G) = 6 + 1$
- $V(G) = 8$
- $V(G) = A - N + 2$
- $V(G) = 22 - 16 + 2$
- $V(G) = 8$