

Prueba de Caja Blanca

“Generación de proformas MarcallTex”

Integrantes:

Cañola Kevin

Marcalla

Cristhian

Lugamaña

Mateo

Tasiguano

Eduardo

Fecha: 2025/11/25

CONTROL DE VERSIONAMIENTO DE PRUEBAS CB

| Version | Fecha | Responsable | Aprobado por |
|-----------------|-------|-------------|--------------|
| PCB_V1.0.0.docx | | | |

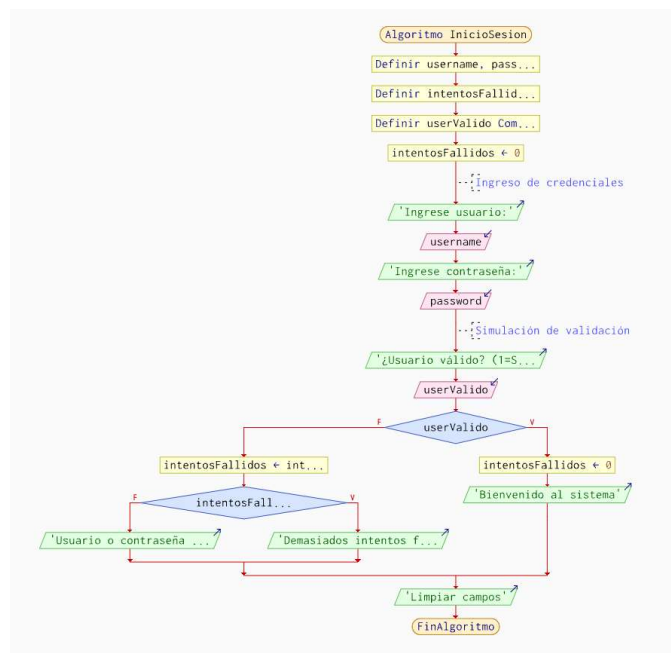
Prueba caja blanca

RF N1ª INICIO DE SESIÓN

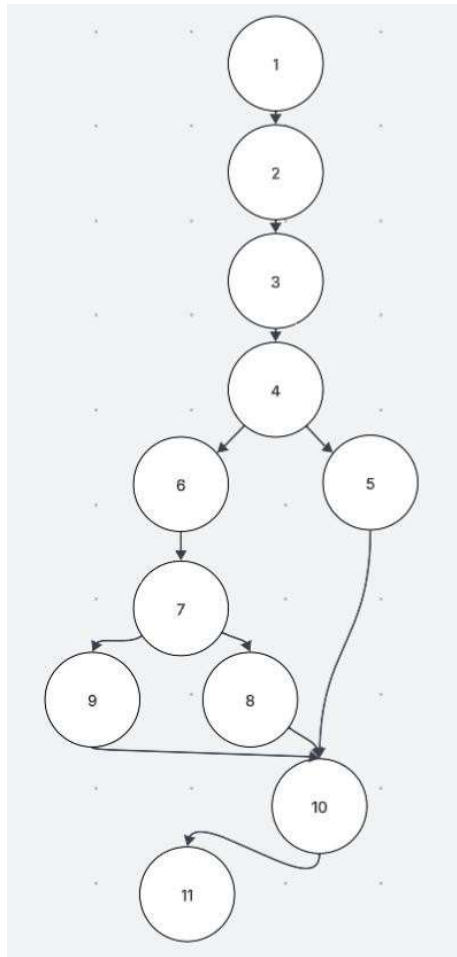
1. CÓDIGO FUENTE

```
35 Private Sub btnIngresar_Click(sender As Object, e As EventArgs) Handles btnIngresar.Click
36     Dim username As String = txtUsuario.Text.Trim()
37     Dim password As String = txtContraseña.Text.Trim()
38
39     Dim user = Usuario.ValidarLogin(username, password)
40
41     If user IsNot Nothing Then
42         intentosFallidos = 0
43
44         MsgBox("Bienvenido " & user.usuario, MsgBoxStyle.Information, "BIENVENIDO AL SISTEMA")
45
46         Dim admin As New ApartadoAdministrador()
47
48         ' ✓ flujo correcto
49         Me.Hide()
50         admin.Show()
51
52     Else
53         intentosFallidos += 1
54
55         If intentosFallidos >= 3 Then
56             MsgBox("Demasiados intentos fallidos.", MsgBoxStyle.Critical, "Error")
57             btnIngresar.Enabled = False
58             bloqueoTimer.Start()
59         Else
60             MsgBox("Usuario o contraseña incorrectos.", MsgBoxStyle.Critical, "Error")
61         End If
62     End If
63
64     LimpiarCampos()
65 End Sub
66
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

| Ruta | Secuencia de nodos | Descripción |
|------|-------------------------------------|--|
| R1 | 1 → 2 → 3 → 4 → 5 → 10 → 11 | Usuario correcto, acceso concedido |
| R2 | 1 → 2 → 3 → 4 → 6 → 7 → 9 → 10 → 11 | Credenciales incorrectas, menos de 3 intentos |
| R3 | 1 → 2 → 3 → 4 → 6 → 7 → 8 → 10 → 11 | Credenciales incorrectas, tercer intento → bloqueo |

5. COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática del Grafo de Flujo se calcula para determinar el número de caminos independientes del sistema.

Datos del grafo

- **N (Número de nodos):** 11
- **P (Número de nodos predicados):** 2
- **A (Número de aristas):** 12

Cálculo

Fórmula 1:

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

$$V(G) = 2 + 1 = 3$$

Fórmula 2:

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

$$V(G) = 12 - 11 + 2 = 3$$

Resultado

La complejidad ciclomática del requisito **Inicio de Sesión** es **3**, lo que indica que existen **tres caminos básicos independientes**.

DONDE

P: Número de nodos predicado

A: Número de aristas

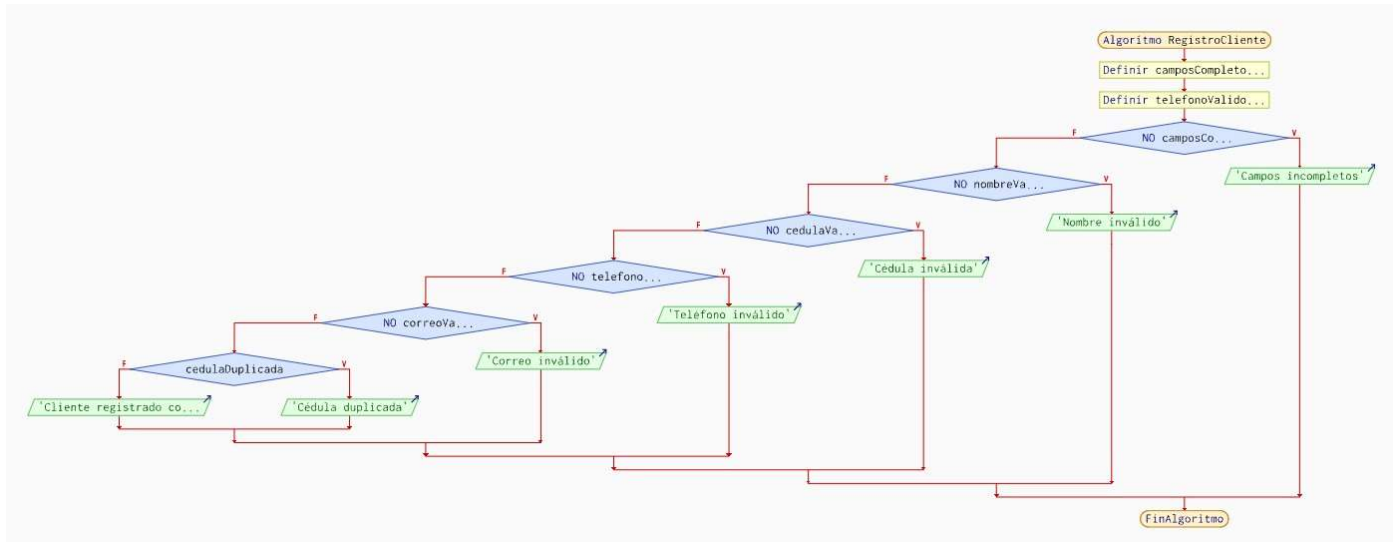
N: Número de nodos

RF N2ª REGISTRO DE CLIENTES

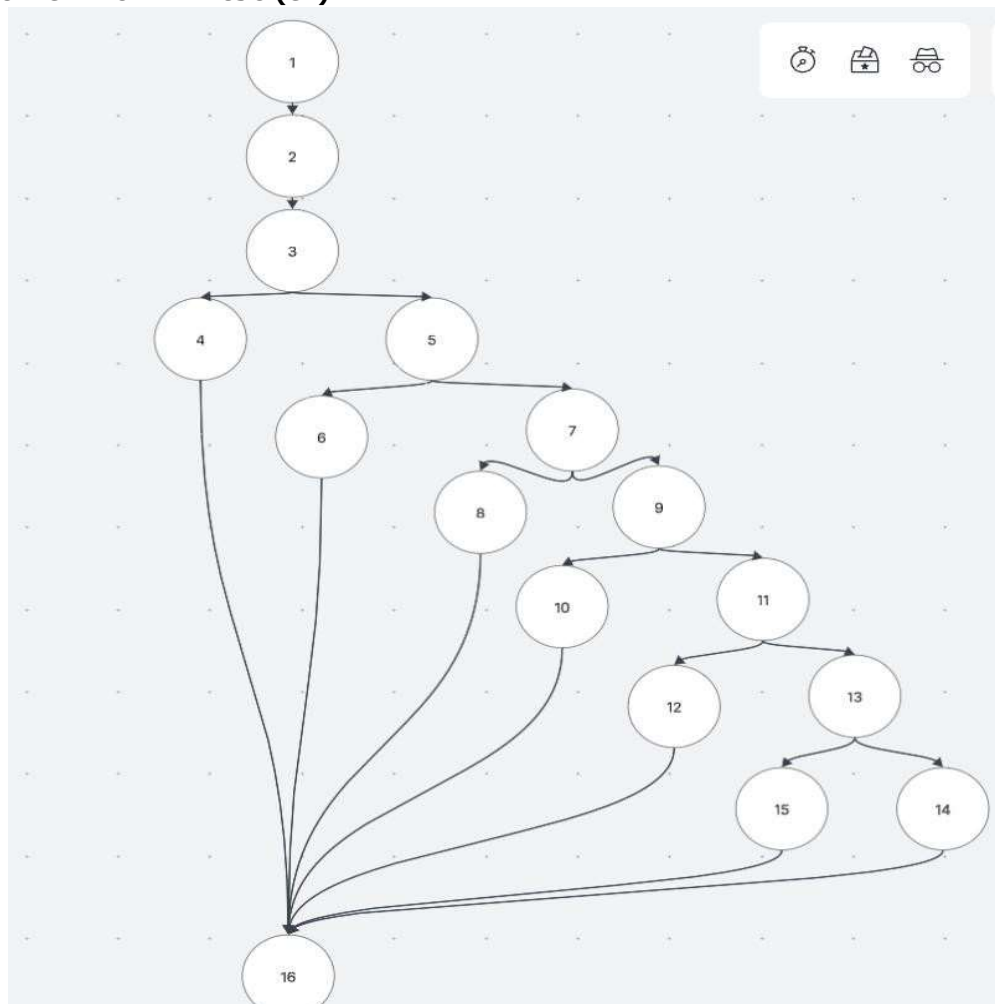
6. CÓDIGO FUENTE

```
1 Private Sub btnRegistrar_Click(sender As Object, e As EventArgs) Handles btnRegistrar.Click
2     Try
3         If txtNombre.Text.Trim() = "" OrElse
4             txtCedula.Text.Trim() = "" OrElse
5             txtTelefono.Text.Trim() = "" OrElse
6             txtCorreo.Text.Trim() = "" OrElse
7             txtDireccion.Text.Trim() = "" Then
8             MessageBox.Show("Por favor, complete todos los campos.")
9             Exit Sub
10        End If
11
12        If Not Regex.IsMatch(txtNombre.Text.Trim(), "[A-Za-zÁÉÍÓÚáéíóúÑñ ]+$") Then
13            MessageBox.Show("El nombre solo puede contener letras.")
14            Exit Sub
15        End If
16
17        If Not CedulaValida(txtCedula.Text.Trim()) Then
18            MessageBox.Show("La cédula ingresada no es válida.")
19            Exit Sub
20        End If
21
22        If Not Regex.IsMatch(txtTelefono.Text.Trim(), "\d{10}$") Then
23            MessageBox.Show("El teléfono debe contener 10 dígitos.")
24            Exit Sub
25        End If
26
27        If Not Regex.IsMatch(txtCorreo.Text.Trim(), "[a-z0-9._%+-]@(gmail|outlook|hotmail|yahoo)\.(com|es)$") Then
28            MessageBox.Show("Ingrese un correo válido.")
29            Exit Sub
30        End If
31
32        Dim existente = coleccion.Find(filtroCedula).FirstOrDefault()
33        If existente IsNot Nothing Then
34            MessageBox.Show("Ya existe un cliente registrado con esta cédula.")
35            Exit Sub
36        End If
37
38        coleccion.InsertOne(nuevoCliente)
39        MessageBox.Show("Cliente registrado correctamente.")
40        LimpiarCampos()
41
42        Catch ex As Exception
43            MessageBox.Show("Error al registrar cliente.")
44        End Try
45    End Sub
46
```

7. DIAGRAMA DE FLUJO (DF)



8. GRAFO DE FLUJO (GF)



9. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

| Ruta | Secuencia de nodos | Descripción |
|------|---|------------------------------|
| R1 | 1 → 2 → 3 → 5 → 7 → 9 → 11 → 13 → 15 → 16 | Registro exitoso del cliente |
| R2 | 1 → 2 → 3 → 4 → 16 | Campos incompletos |
| R3 | 1 → 2 → 3 → 5 → 6 → 16 | Nombre inválido |
| R4 | 1 → 2 → 3 → 5 → 7 → 8 → 16 | Cédula inválida |
| R5 | 1 → 2 → 3 → 5 → 7 → 9 → 10 → 16 | Teléfono inválido |
| R6 | 1 → 2 → 3 → 5 → 7 → 9 → 11 → 12 → 16 | Correo inválido |
| R7 | 1 → 2 → 3 → 5 → 7 → 9 → 11 → 13 → 14 → 16 | Cédula duplicada |

10. COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática del Grafo de Flujo se calcula para determinar el número de caminos independientes del sistema.

Datos del grafo

- **N (Número de nodos):** 16
- **P (Número de nodos predicados):** 6
- **A (Número de aristas):** 21

Cálculo

Fórmula 1:

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

$$V(G) = 6 + 1 = 7$$

Fórmula 2:

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

$$V(G) = 21 - 16 + 2 = 7$$

Resultado

La complejidad ciclomática del requisito **Registro de Clientes** es 7, lo que indica que existen **siete caminos básicos independientes**.

DONDE

P: Número de nodos predicado

A: Número de aristas

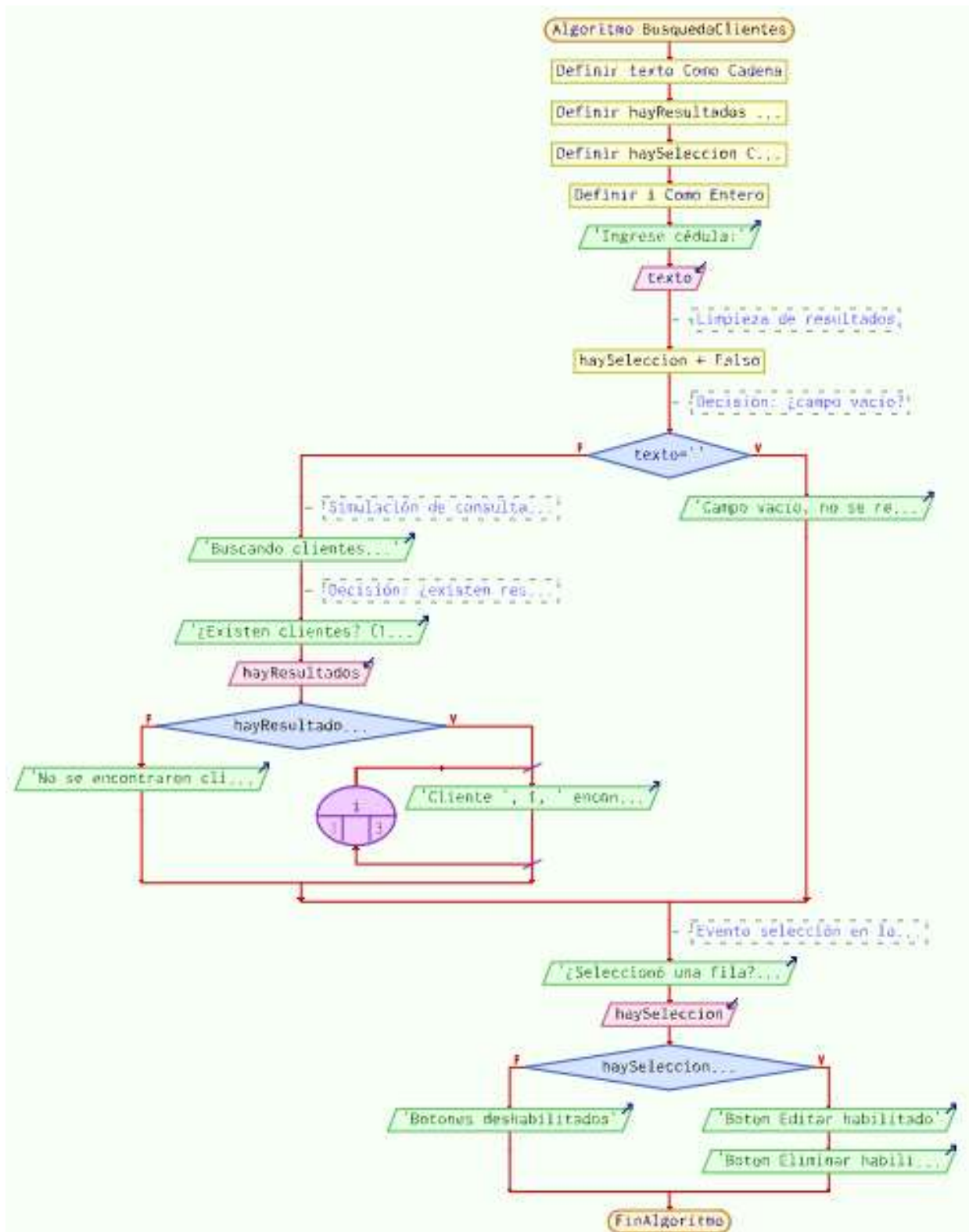
N: Número de nodos

RF N3ª BUSQUEDA CLIENTES

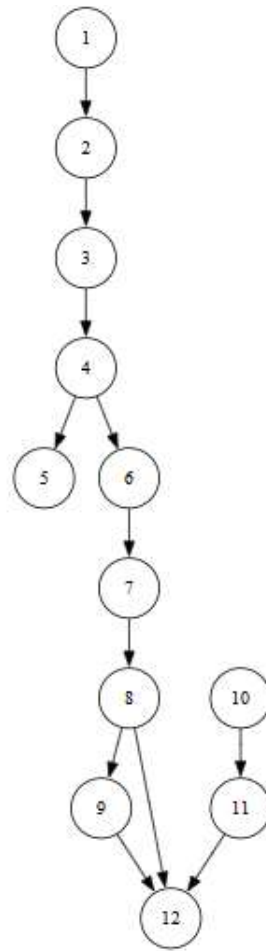
11. CÓDIGO FUENTE

```
39      ' ===== BUSCAR =====
40      3 referencias
41      Private Sub BuscarCliente()
42          Dim texto As String = txtCedula.Text.Trim()
43
44          dgvResultados.Rows.Clear()
45          btnEditar.Enabled = False
46          btnEliminar.Enabled = False
47
48          If texto = "" Then Exit Sub
49
50          Dim filtro = Builders(Of BsonDocument).Filter.Regex(
51              "cedula",
52              New BsonRegularExpression("^" & texto)
53          )
54
55          Dim clientes = clientesCollection.Find(filtro).ToList()
56
57          For Each cliente In clientes
58              dgvResultados.Rows.Add(
59                  cliente("cedula").ToString(),
60                  cliente("nombre").ToString(),
61                  cliente("telefono").ToString(),
62                  cliente("correo").ToString(),
63                  cliente("direccion").ToString()
64              )
65          Next
66      End Sub
67
68      ' ===== BUSCAR AUTOMÁTICO =====
69      0 referencias
70      Private Sub txtCedula_TextChanged(sender As Object, e As EventArgs) Handles txtCedula.TextChanged
71          BuscarCliente()
72      End Sub
73
74      ' ===== SOLO NÚMEROS =====
75      0 referencias
76      Private Sub txtCedula_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtCedula.KeyPress
77          If Not Char.IsControl(e.KeyChar) AndAlso Not Char.IsDigit(e.KeyChar) Then
78              e.Handled = True
79          End If
80      End Sub
81
82      ' ===== EVENTO CORRECTO =====
83      0 referencias
84      Private Sub dgvResultados_SelectionChanged(sender As Object, e As EventArgs) _
85          Handles dgvResultados.SelectionChanged
86
87          Dim haySeleccion As Boolean = dgvResultados.SelectedRows.Count > 0
88          btnEditar.Enabled = haySeleccion
89          btnEliminar.Enabled = haySeleccion
90      End Sub
```

12. DIAGRAMA DE FLUJO (DF)



13. GRAFO DE FLUJO (GF)



14. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

| Ruta | Secuencia de nodos | Descripción |
|------|------------------------------------|---|
| R1 | 1 → 2 → 3 → 4 → 5 | Campo vacío, no se realiza búsqueda |
| R2 | 1 → 2 → 3 → 4 → 6 → 7 → 8 → 9 → 12 | Texto ingresado con resultados encontrados |
| R3 | 1 → 2 → 3 → 4 → 6 → 7 → 8 → 12 | Texto ingresado sin resultados |
| R4 | 10 → 11 → 12 | Selección de fila y habilitación de botones |

15. COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática permite determinar el número de caminos independientes del sistema.

Datos del grafo

- **N (Número de nodos):** 12
- **P (Número de nodos predicados):** 3
- **A (Número de aristas):** 14

Cálculo

Fórmula 1:

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

$$V(G) = 3 + 1 = 4$$

Fórmula 2:

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

$$V(G) = 14 - 12 + 2 = 4$$

Resultado

La complejidad ciclomática del requisito **Búsqueda de Clientes** es **4**, lo que indica que existen **cuatro caminos básicos independientes** que deben ser cubiertos por las pruebas de caja blanca.

DONDE

P: Número de nodos predicado

A: Número de aristas

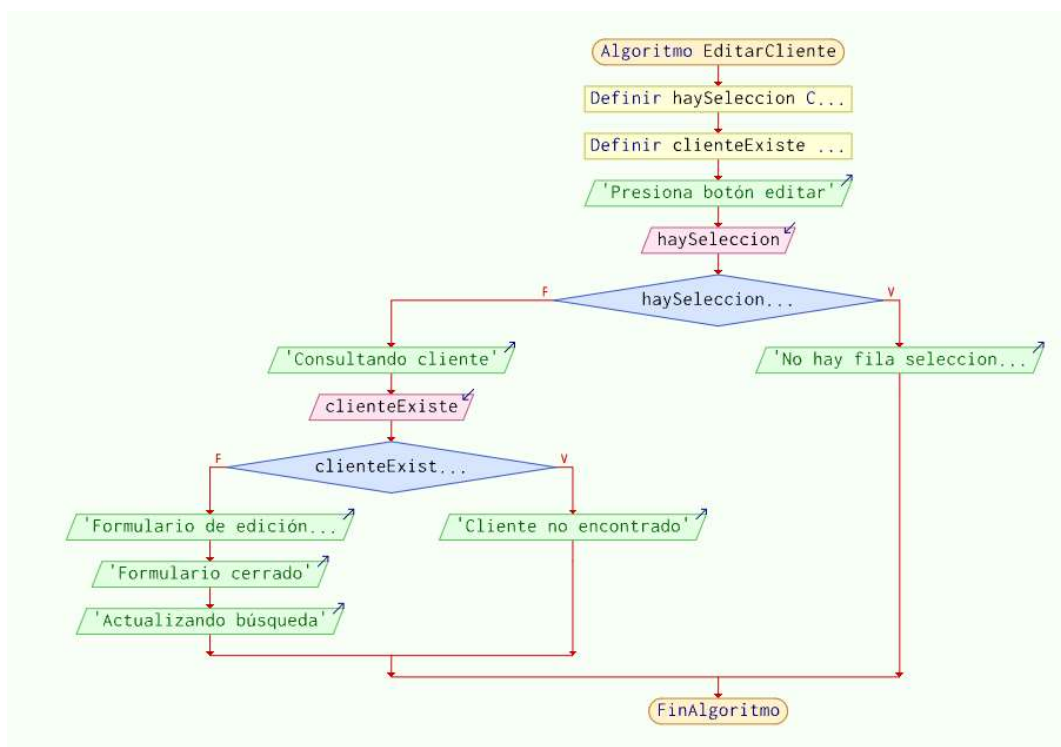
N: Número de nodos

RF N4ª MODIFICACION DE CLIENTES

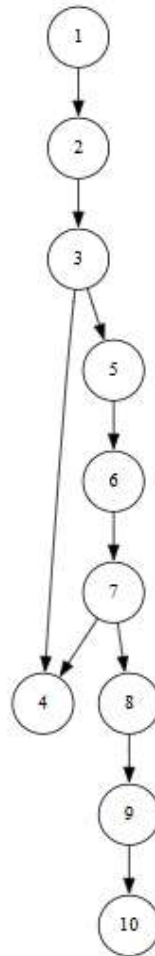
16. CÓDIGO FUENTE

```
88 ' ===== EDITAR =====
89 ' D referencias
90 Private Sub btnEditar_Click(sender As Object, e As EventArgs) Handles btnEditar.Click
91     If dgvResultados.SelectedRows.Count = 0 Then Exit Sub
92
93     Dim fila = dgvResultados.SelectedRows(0)
94     Dim cedula As String = fila.Cells("cedula").Value.ToString()
95
96     Dim cliente = clientesCollection.Find(
97         New BsonDocument("cedula", cedula)
98     ).FirstOrDefault()
99
100     If cliente Is Nothing Then Exit Sub
101
102     Dim frm As New editarCliente(
103         cliente("cedula").ToString(),
104         cliente("nombre").ToString(),
105         cliente("telefono").ToString(),
106         cliente("correo").ToString(),
107         cliente("direccion").ToString(),
108         If(cliente.Contains("infoAdicional"), cliente("infoAdicional").ToString(), "")
109     )
110
111     frm.ShowDialog()
112     BuscarCliente()
113 End Sub
```

17. DIAGRAMA DE FLUJO (DF)



18. GRAFO DE FLUJO (GF)



19. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

| Ruta | Secuencia de nodos | Descripción |
|------|------------------------------------|-----------------------------------|
| R1 | 1 → 2 → 3 → 4 | No hay fila seleccionada |
| R2 | 1 → 2 → 3 → 5 → 6 → 7 → 4 | Cliente no encontrado |
| R3 | 1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 | Modificación correcta del cliente |

20. COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática permite determinar el número de caminos independientes del sistema.

Datos del grafo

- **N (Número de nodos):** 10
- **P (Número de nodos predicados):** 2
- **A (Número de aristas):** 11

Cálculo

Fórmula 1:

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

$$V(G) = 2 + 1 = 3$$

Fórmula 2:

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

$$V(G) = 11 - 10 + 2 = 3$$

Resultado

La complejidad ciclomática del requisito **Editar Cliente** es **3**, lo que indica que existen **tres caminos independientes** que deben ser probados.

DONDE

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

RF N5ª ELIMINACION DE CLIENTES

21. CÓDIGO FUENTE

```
114 ----- ELIMINAR -----
115 0 referencias
116 Private Sub btnEliminar_Click(sender As Object, e As EventArgs) Handles btnEliminar.Click
117     If dgvResultados.SelectedRows.Count = 0 Then Exit Sub
118
119     Dim fila = dgvResultados.SelectedRows(0)
120
121     Dim frm As New EliminarCliente(
122         fila.Cells("cedula").Value.ToString(),
123         fila.Cells("nombre").Value.ToString(),
124         fila.Cells("telefono").Value.ToString(),
125         fila.Cells("correo").Value.ToString(),
126         fila.Cells("direccion").Value.ToString()
127     )
128
129     frm.ShowDialog()
130     BuscarCliente()
131 End Sub
```

```

Private Sub btnEliminar_Click(sender As Object, e As EventArgs) Handles btnEliminar.Click
    ' Primera confirmación
    Dim result = MessageBox.Show("¿Estás seguro de eliminar al cliente con cédula " & cedulaCliente & "?",
        "Confirmar eliminación",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Warning)

    If result = DialogResult.Yes Then
        ' Segunda confirmación
        Dim finalConfirm = MessageBox.Show("¡Esta acción no se puede deshacer! ¿Seguro que deseas eliminar este cliente?",
            "Confirmación final",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Warning)

        If finalConfirm = DialogResult.Yes Then
            Try
                ' Eliminar de MongoDB
                Dim db = ConexionMongo.ObtenerConexion()
                Dim collection = db.GetCollection(Of BsonDocument)("Clientes")

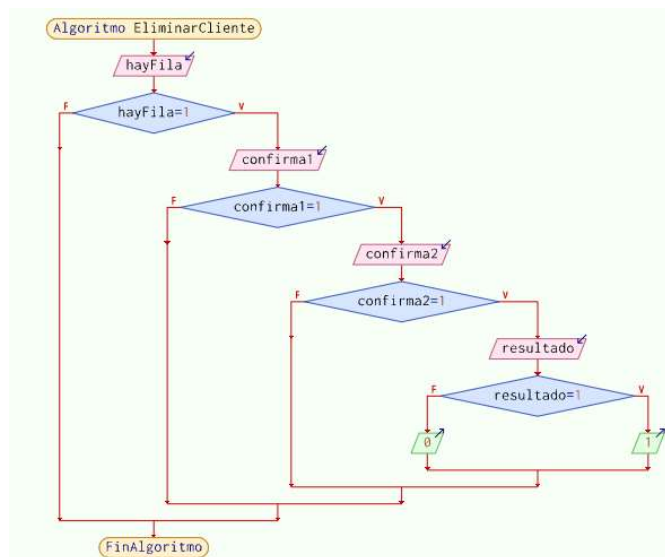
                Dim filtro = New BsonDocument("cedula", cedulaCliente)
                collection.DeleteOne(filtro)

                MessageBox.Show("Cliente eliminado correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information)
                Me.Close()

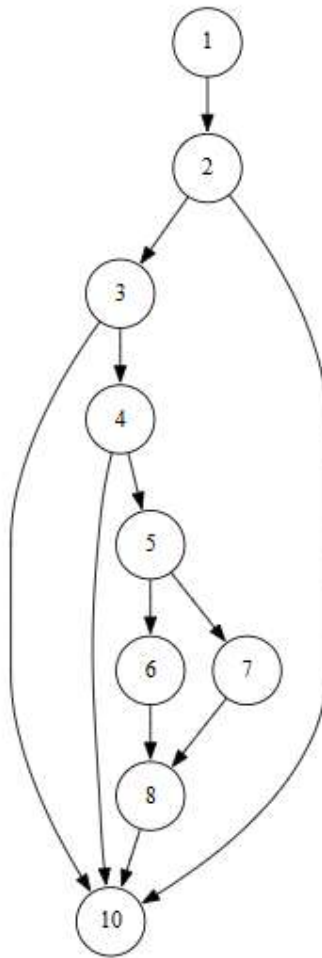
            Catch ex As Exception
                MessageBox.Show("Error al eliminar: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If
    End If
End Sub

```

22. DIAGRAMA DE FLUJO (DF)



23. GRAFO DE FLUJO (GF)



24. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

| Ruta | Secuencia de nodos | Descripción |
|------|--------------------------------|--|
| R1 | 1 → 2 → 10 | No hay fila seleccionada |
| R2 | 1 → 2 → 3 → 10 | Fila seleccionada, primera confirmación = No |
| R3 | 1 → 2 → 3 → 4 → 10 | Primera confirmación Sí, segunda confirmación = No |
| R4 | 1 → 2 → 3 → 4 → 5 → 6 → 8 → 10 | Eliminación exitosa |
| R5 | 1 → 2 → 3 → 4 → 5 → 7 → 8 → 10 | Error al eliminar cliente |

25. COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática permite determinar el número de caminos independientes del sistema.

Datos del grafo

- **N (Número de nodos):** 10
- **P (Número de nodos predicados):** 4
- **A (Número de aristas):** 13

Cálculo

Fórmula 1:

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

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

Fórmula 2:

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

$$V(G) = 13 - 10 + 2 = 5$$

Resultado

La complejidad ciclomática del requisito **Eliminar Cliente** es **5**, lo que indica que existen **cinco caminos básicos independientes** que deben ser probados para garantizar la cobertura completa del flujo de control.

DONDE

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos