

Prueba de Caja Blanca

“Generación de proformas MarcarllTex:”

Integrantes:

Cañola Kevin

Marcalla Cristian

Lugamaña

Mateo

Tasiguano

Eduardo

Fecha: 2025/12/01

CONTROL DE VERSIONAMIENTO DE PRUEBAS CB

Versión	Fecha	Responsable	Aprobado por
PCB_V1.0.0.docx			

Prueba caja blanca

RF N1^a Inicio de sesión

1. CÓDIGO FUENTE

1.1 Manejo de intentos fallidos y validación

```
' --- Botón Ingresar ---
Private Sub btnIngresar_Click(sender As Object, e As EventArgs) Handles btnIngresar.Click
    Dim usuario, password As String
    usuario = txtUsuario.Text
    password = txtContrasena.Text
    txtUsuario.Text = ""
    txtContrasena.Text = ""

    If (usuario = "Kevin") And (password = "1234") Then
        intentosFallidos = 0
        MsgBox("Bienvenido Administrador")

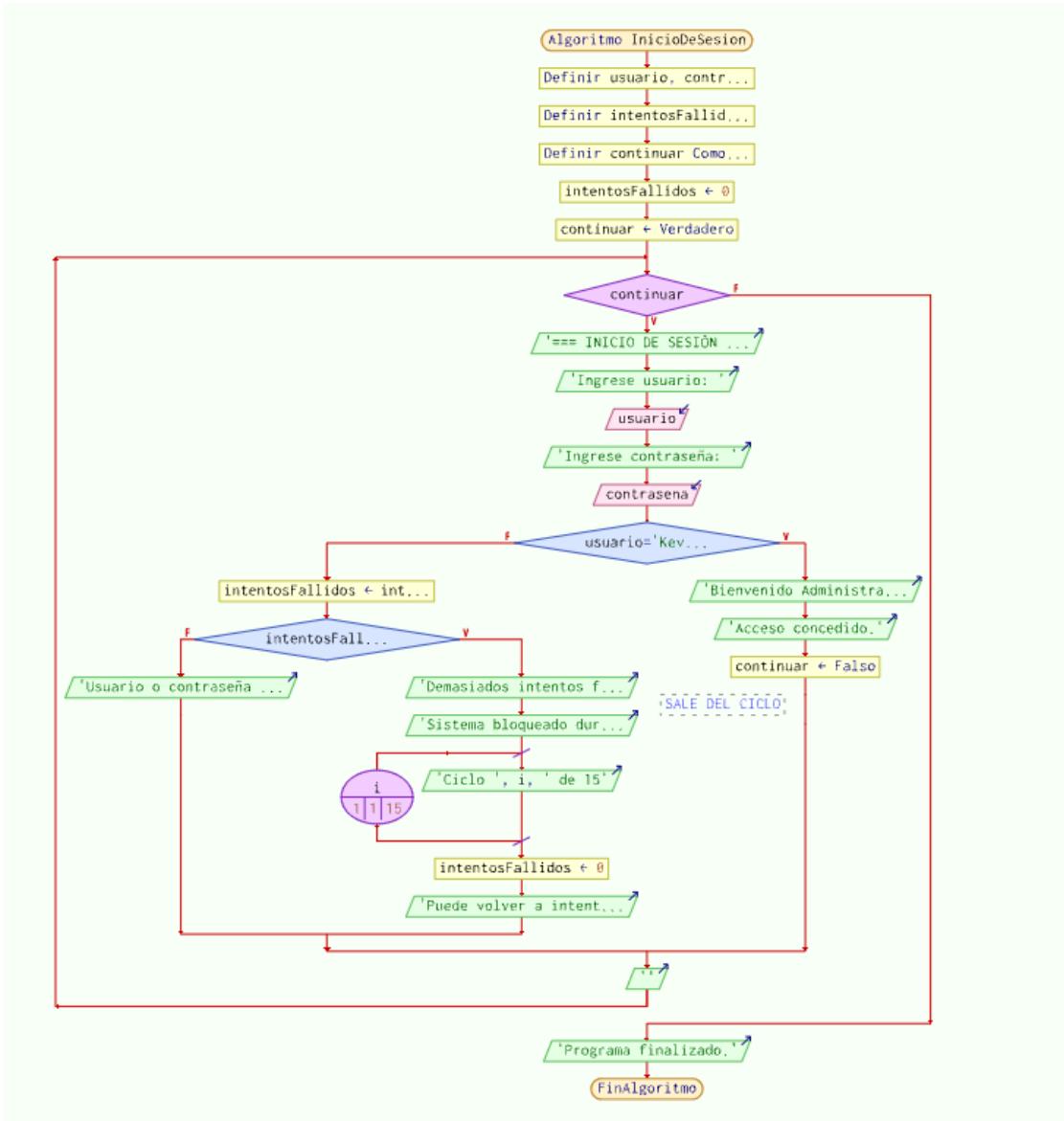
        ' Mostrar el formulario de administrador sin cerrar el login
        Dim admin As New ApartadoAdministrador()
        admin.Show()
        Me.Hide() ' Oculta el login mientras está en el panel

    Else
        intentosFallidos += 1
        If intentosFallidos >= 3 Then
            MsgBox("Demasiados intentos fallidos, espere 15 segundos para volver a intentar.")
            btnIngresar.Enabled = False
            bloqueoTimer.Start()
        Else
            MsgBox("Usuario o Contraseña Incorrecta. Inténtelo nuevamente")
        End If
    End If
End Sub
```

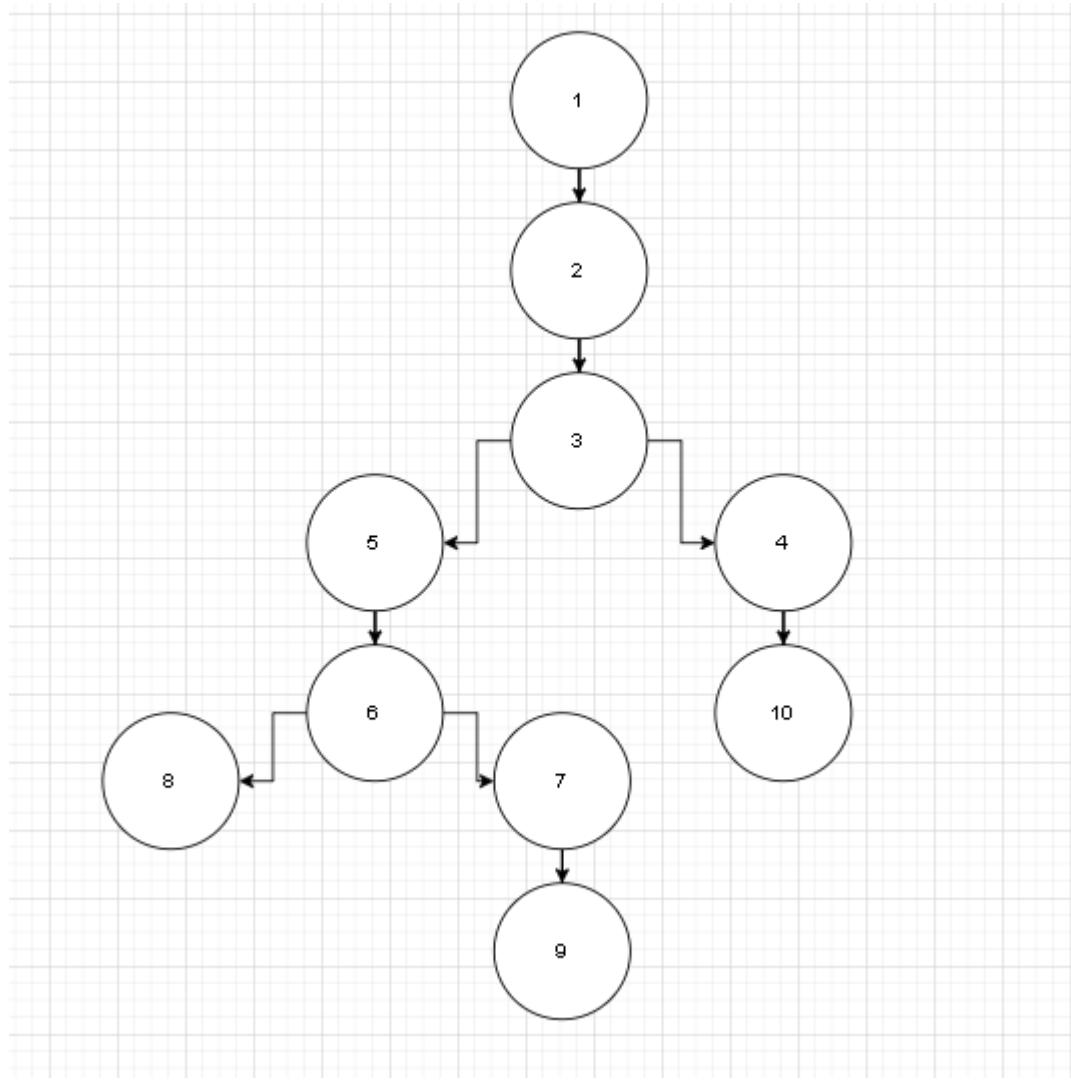
1.2 Temporizador de desbloqueo

```
' --- Desbloquea el botón Ingresar después del bloqueo ---
Private Sub DesbloquearBoton(sender As Object, e As EventArgs)
    btnIngresar.Enabled = True
    bloqueoTimer.Stop()
    intentosFallidos = 0
    MsgBox("Ahora puede volver a intentar ingresar.")
End Sub
```

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 → 10	Usuario correcto, acceso concedido
R2	1 → 2 → 3 → 5 → 6 → NO → 8 → 9 → 2	Credenciales incorrectas, menos de 3 intentos
R3	1 → 2 → 3 → 5 → 6 → SI → 7 → 9 → 2	Credenciales incorrectas, tercer intento → bloqueo
R4	Repetición de ciclo hasta que usuario correcto	Combinación de rutas R2/R3 hasta R1

5. COMPLEJIDAD CICLOMÁTICA

- Número de nodos (N): 10
- Número de nodos predicados/decisión (P): 2 (Nodo 3 y Nodo 6)
- Número de aristas (A): 11

Cálculo:

1. Por nodos predicados:

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

2. Por aristas y nodos:

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

RF N2^a Registro de clientes

1. CÓDIGO FUENTE

1.1 Validación de campos vacíos

```

12     '1. Validación de campos vacíos
13     If txtNombre.Text = "" Or txtApellido.Text = "" Or txtTelefono.Text = "" Or
14         txtCorreo.Text = "" Or txtDireccion.Text = "" Then
15         MessageBox.Show("Debe completar todos los campos.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
16         Exit Sub
17     End If
18

```

1.2 Validación del teléfono

```

20     '2. Validar teléfono
21     If Not IsNumeric(txtTelefono.Text) Or txtTelefono.Text.Length < 8 Then
22         MessageBox.Show("Ingrese un número de teléfono válido.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
23         Exit Sub
24     End If

```

1.3 Validación del correo

```

26     '3. Validar correo
27     If Not txtCorreo.Text.Contains("@") Or Not txtCorreo.Text.Contains(".") Then
28         MessageBox.Show("Ingrese un correo válido.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
29         Exit Sub
30     End If
31

```

1.4 Guardar cliente

```

32     '4. Registro exitoso (aquí iría la inserción a base de datos si tuvieras)
33     MessageBox.Show("Cliente registrado correctamente.", "Registro", MessageBoxButtons.OK, MessageBoxIcon.Information)
34
35 End Sub

```

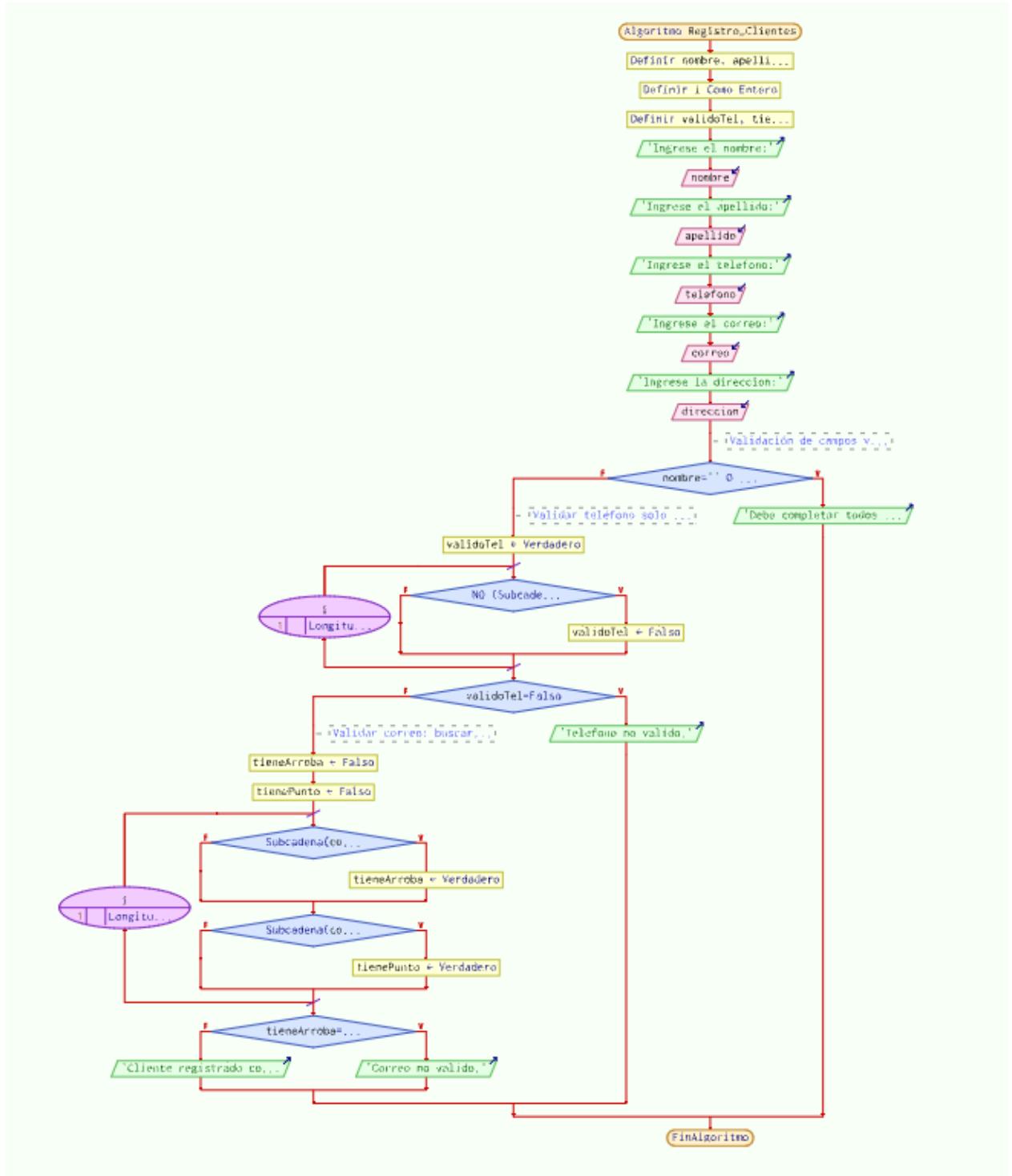
1.5 Limpiar campos

```

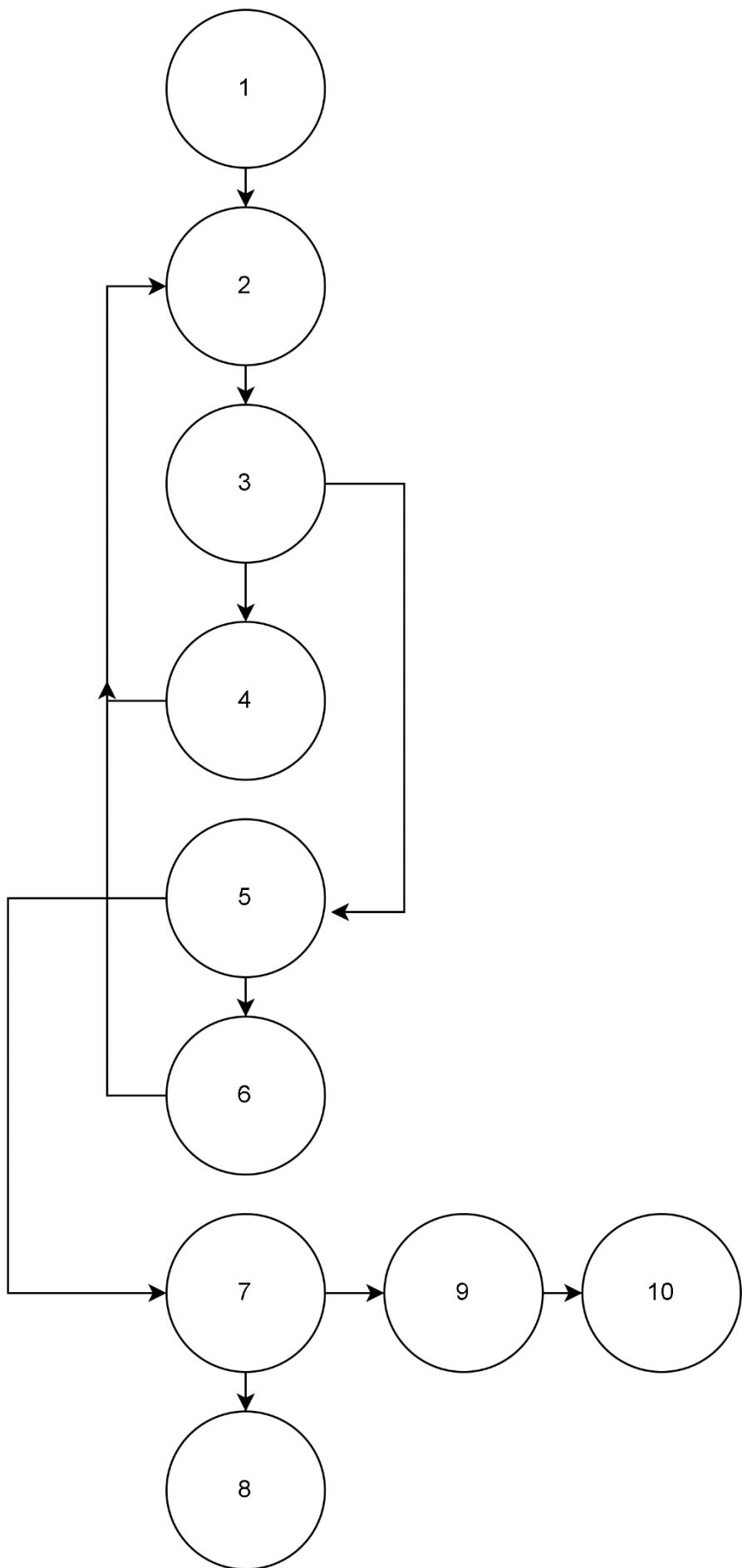
37     '-----'
38     ' BOTÓN LIMPIAR - Limpia todos los campos
39     '-----'
40     0 referencias
41     Private Sub btnLimpiar_Click(sender As Object, e As EventArgs) Handles btnLimpiar.Click
42         txtNombre.Clear()
43         txtApellido.Clear()
44         txtTelefono.Clear()
45         txtCorreo.Clear()
46         txtDireccion.Clear()
47
48         MessageBox.Show("Campos limpiados.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information)
49     End Sub

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Ruta	Secuencia	Descripción
R1	1 → 2 → 3 → 5 → 7 → 9 → 10	Todo correcto → cliente registrado
R2	1 → 2 → 3 → 4 → 10	Campos vacíos
R3	1 → 2 → 3 → 5 → 6 → 10	Teléfono incorrecto
R4	1 → 2 → 3 → 5 → 7 → 8 → 10	Correo incorrecto

5. COMPLEJIDAD CICLOMÁTICA

- **Número de nodos (N): 10**

(1: Inicio, 2: Leer datos, 3: Campos vacíos, 4: Error campos, 5: Validar teléfono, 6: Error teléfono, 7: Validar correo, 8: Error correo, 9: Registrar, 10: Fin)

- **Número de nodos predicados/decisión (P): 3**

- Nodo 3 → ¿Campos vacíos?
- Nodo 5 → ¿Teléfono válido?
- Nodo 7 → ¿Correo válido?

- **Número de aristas (A): 12**

(Cuenta de todas las conexiones entre nodos del grafo)

Cálculo:

1. Por nodos predicados:

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

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

2. Por aristas y nodos:

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

$$V(G) = 12 - 10 + 2 = 4$$