

Prueba de Caja Blanca

“Generación de proformas MarcallTex”

Integrantes:

Cañola Kevin

Marcalla

Cristhian

Lugamaña

Mateo

Tasiguano

Eduardo

CONTROL DE VERSIONAMIENTO DE PRUEBAS CB

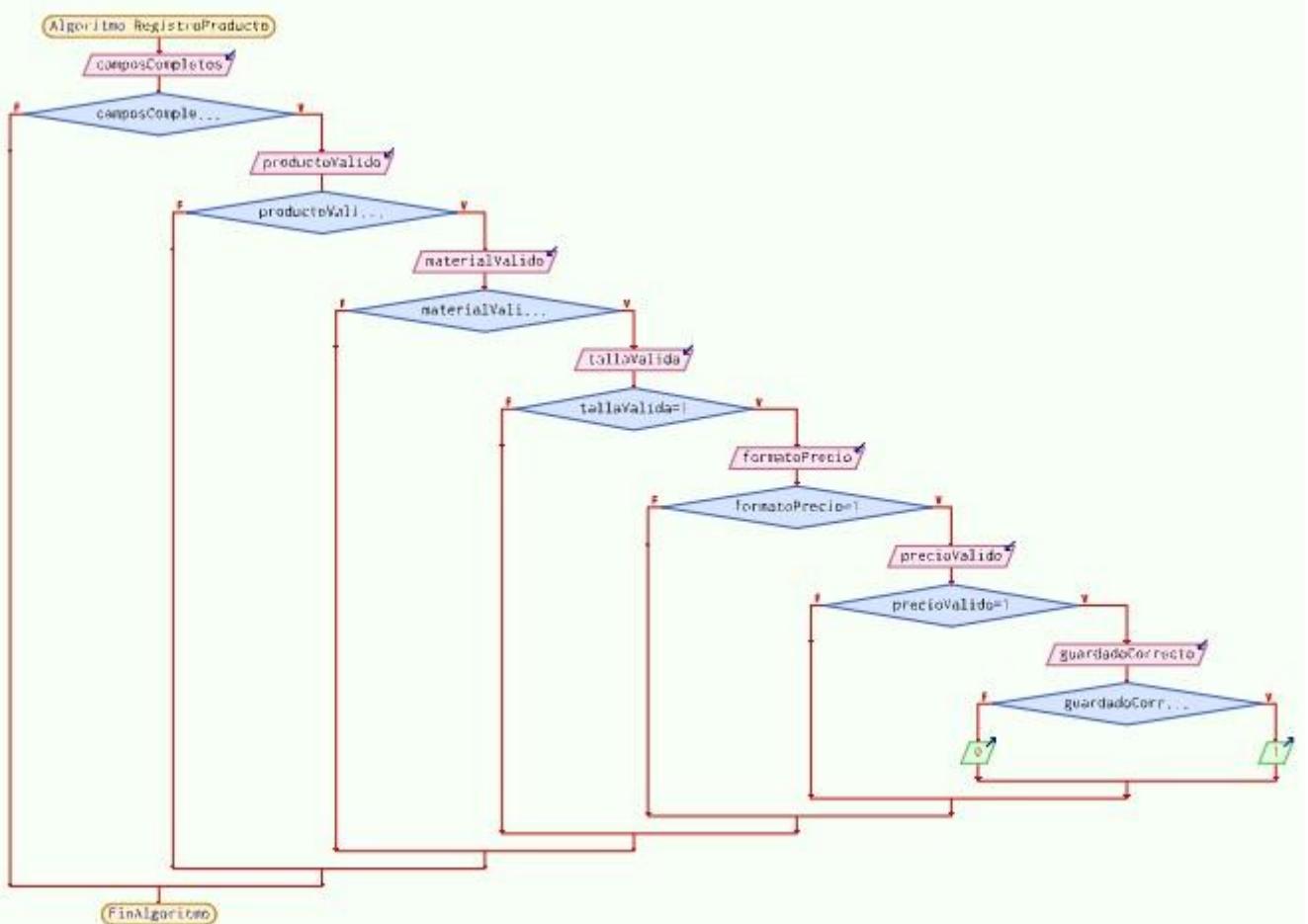
Versión	Fecha	Responsable	Aprobado por
PCB_V1.0.0.docx			

RF N6^a REGISTROS DE PRODUCTOS

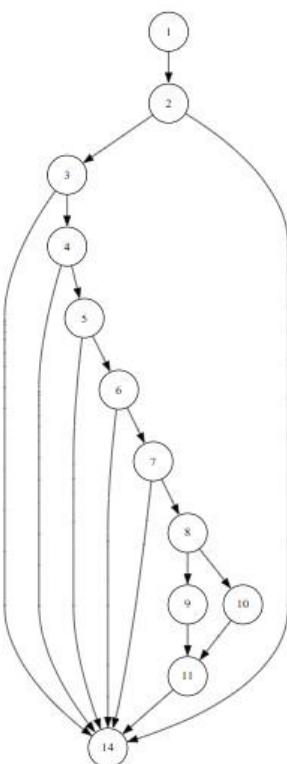
1. CÓDIGO FUENTE

```
55     ' Validación precio
56     If Not Regex.IsMatch(txtPrecioUnitario.Text.Trim, "^\d+(\.\d{1,2})?") Then
57         MessageBox.Show("El precio unitario debe ser un número válido, máximo 2 decimales")
58         txtPrecioUnitario.Focus()
59         Exit Sub
60     End If
61
62     Dim precio As Double
63     If Not Double.TryParse(txtPrecioUnitario.Text.Trim, precio) Then
64         MessageBox.Show("Ingrese un precio válido")
65         txtPrecioUnitario.Focus()
66         Exit Sub
67     End If
68
69     ' Guardar producto
70     Dim nuevoProducto As New Producto With {
71         .Producto = txtProducto.Text.Trim(),
72         .Talla = valorTalla,
73         .TipoMaterial = txtTipoMaterial.Text.Trim(),
74         .PrecioUnitario = precio
75     }
76
77     Try
78         productosCollection.InsertOne(nuevoProducto)
79         MessageBox.Show("Producto registrado correctamente")
80         LimpiarCampos()
81     Catch ex As Exception
82         MessageBox.Show("Error al guardar: " & ex.Message)
83     End Try
84 End Sub
85
86
87 ' ===== BOTÓN GUARDAR =====
88 0 referencias
89 Private Sub btnGuardar_Click(sender As Object, e As EventArgs) Handles btnGuardar.Click
90     ' Campos obligatorios
91     If txtProducto.Text.Trim = "" OrElse
92         txtTalla.Text.Trim = "" OrElse
93         txtTipoMaterial.Text.Trim = "" OrElse
94         txtPrecioUnitario.Text.Trim = "" Then
95             MessageBox.Show("Complete todos los campos")
96             Exit Sub
97         End If
98
99     ' Validación producto
100    If Not Regex.IsMatch(txtProducto.Text.Trim, "^[a-zA-Z\s]+") Then
101        MessageBox.Show("El nombre del producto solo debe contener letras y espacios")
102        txtProducto.Focus()
103        Exit Sub
104    End If
105
106    ' Validación tipo de material
107    If Not Regex.IsMatch(txtTipoMaterial.Text.Trim, "^[a-zA-Z\s]+") Then
108        MessageBox.Show("El tipo de material solo debe contener letras y espacios")
109        txtTipoMaterial.Focus()
110        Exit Sub
111    End If
112
113    ' Validación talla lógica
114    Dim valorTalla As String = txtTalla.Text.Trim().ToUpper()
115    Dim tallasValidas As String() = {"S", "M", "L", "XL", "XXL"}
116    Dim esNumeroValido As Boolean = Integer.TryParse(valorTalla, Nothing) AndAlso CInt(valorTalla) >= 30 AndAlso CInt(valorTalla) <
117        tallasValidas.Contains(valorTalla) AndAlso Not esNumeroValido Then
118            MessageBox.Show("Ingrese una talla válida (S, M, L, XL, XXL o número entre 30 y 50)")
119            txtTalla.Focus()
120            Exit Sub
121        End If
```

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 → 14	Campos incompletos
R2	1 → 2 → 3 → 14	Producto inválido
R3	1 → 2 → 3 → 4 → 14	Material inválido
R4	1 → 2 → 3 → 4 → 5 → 14	Talla inválida
R5	1 → 2 → 3 → 4 → 5 → 6 → 14	Precio formato inválido
R6	1 → 2 → 3 → 4 → 5 → 6 → 7 → 14	Precio no numérico
R7	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 11 → 14	Registro exitoso
R8	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 10 → 11 → 14	Error al guardar

5. COMPLEJIDAD CICLOMÁTICA

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

Datos del grafo

- N (Número de nodos): 14
- P (Número de nodos predicados): 7
- A (Número de aristas): 4

Cálculo

Fórmula 1:

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

$$V(G) = 7 + 1 = 8$$

Fórmula 2:

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

$$V(G) = 20 - 14 + 2 = 8$$

Resultado

La complejidad ciclomática del requisito **Registro de Productos** es **8**, por lo tanto existen **8 caminos básicos independientes** que deben probarse.

DONDE

P: Número de nodos predicado

A: Número de aristas

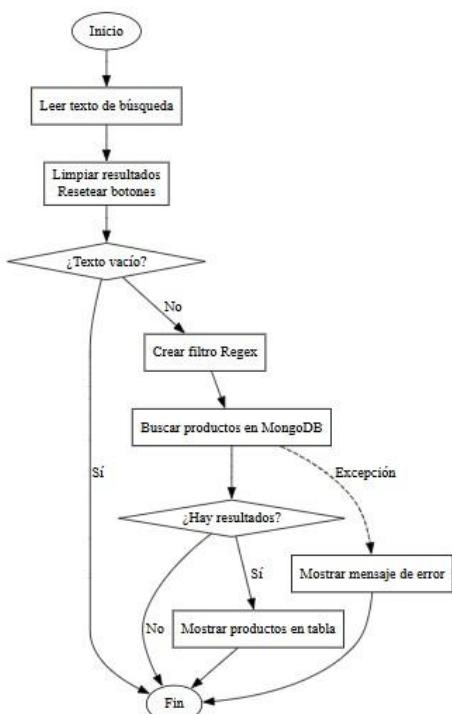
N: Número de nodos

RF N7ª BUSQUEDA DE PRODUCTOS

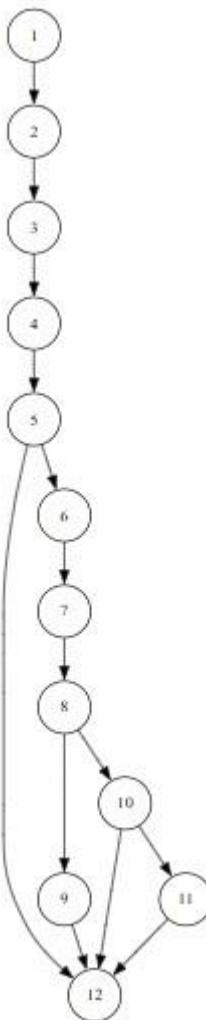
1. CÓDIGO FUENTE

```
42      | 3 referencias
43      | Private Sub BuscarProducto()
44      |     Dim texto As String = txtBuscarProducto.Text.Trim()
45      |
46      |     dgvResultados.Rows.Clear()
47      |     productoSeleccionado = Nothing
48      |     btnEditar.Enabled = False
49      |     btnEliminar.Enabled = False
50
51      |     If texto = "" Then Exit Sub
52
53      | Try
54      |     ' SIMILITUD (EMPIEZA CON / CONTIENE)
55      |     Dim filtro = Builders(Of Producto).Filter.Regex(
56      |         "producto",
57      |         New BsonRegularExpression(texto, "i")
58      |     )
59
60      |     Dim lista = productosCollection.Find(filtro).ToList()
61
62      |     For Each p In lista
63      |         dgvResultados.Rows.Add(
64      |             p.Id,
65      |             p.Producto,
66      |             p.Talla,
67      |             p.TipoMaterial,
68      |             Math.Round(p.PrecioUnitario, 2)
69      |         )
70
71      |     Next
72
73      | Catch ex As Exception
74      |     MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
75
76  End Try
77
78 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 – 5 – 12	El usuario no ingresa texto de búsqueda, el proceso termina sin consultar la base de datos.
R2	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12	El usuario ingresa texto válido, la consulta se ejecuta correctamente y se muestran los productos encontrados.
R3	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 12	El usuario ingresa texto válido, la consulta se ejecuta correctamente pero no se encuentran productos.
R4	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12	El usuario ingresa texto válido pero ocurre un error durante la consulta a la base de datos y se muestra un mensaje de error.

5. 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):** 4
- **A (Número de aristas):** 14

Cálculo

Fórmula 1:

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

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

Fórmula 2:

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

$$V(G) = 14 - 12 + 2 = 5$$

Resultado

La **complejidad ciclomática del requisito 7 (Búsqueda de productos por similitud)** es **5**, lo que indica que existen **cinco caminos básicos independientes** que deben ser probados mediante pruebas de caja blanca.

DONDE

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

RF N8ª MODIFICACION DE PRODUCTOS

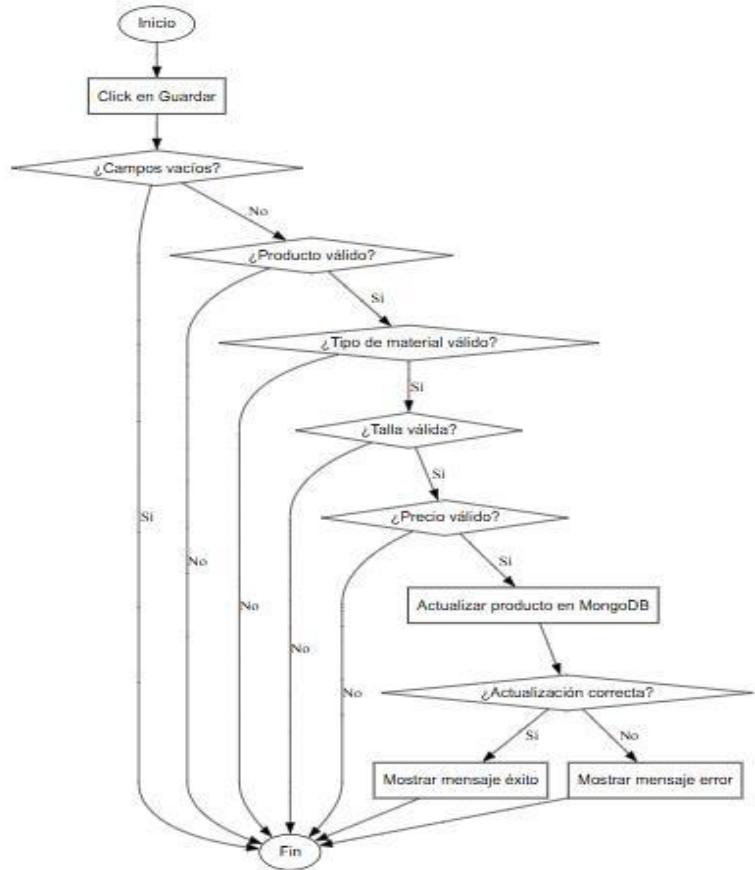
1. CÓDIGO FUENTE

```
28     Private Sub btnGuardar_Click(sender As Object, e As EventArgs) Handles btnGuardar.Click
29
30         ' ===== VALIDAR CAMPOS VACÍOS =====
31         If txtProducto.Text.Trim() = "" OrElse
32             txtTalla.Text.Trim() = "" OrElse
33                 txtTipoMaterial.Text.Trim() = "" OrElse
34                     txtPrecioUnitario.Text.Trim() = "" Then
35                         MessageBox.Show("Complete todos los campos")
36                         Exit Sub
37                     End If
38
39         ' ===== VALIDAR PRODUCTO =====
40         If Not Regex.IsMatch(txtProducto.Text.Trim(), "^[A-Za-z\s]+$") Then
41             MessageBox.Show("El producto solo debe contener letras y espacios")
42             txtProducto.Focus()
43             Exit Sub
44         End If
45
46         ' ===== VALIDAR TIPO MATERIAL =====
47         If Not Regex.IsMatch(txtTipoMaterial.Text.Trim(), "^[A-Za-z\s]+$") Then
48             MessageBox.Show("El tipo de material solo debe contener letras y espacios")
49             txtTipoMaterial.Focus()
50             Exit Sub
51         End If
52
53         ' ===== VALIDAR TALLA =====
54         Dim talla As String = txtTalla.Text.Trim().ToUpper()
55         Dim tallasValidas As String() = {"S", "M", "L", "XL", "XXL"}
56
57         Dim esTallaNumero As Boolean = Regex.IsMatch(talla, "\d+")
58         Dim esTallaLetra As Boolean = tallasValidas.Contains(talla)
59
60         If esTallaNumero Then
61             Dim tallaNum As Integer = CInt(talla)
62             If tallaNum < 30 OrElse tallaNum > 50 Then
63                 MessageBox.Show("La talla numérica debe estar entre 30 y 50")
64
65             txtTalla.Focus()
66             Exit Sub
67         End If
68         ElseIf Not esTallaLetra Then
69             MessageBox.Show("Ingrese una talla válida (S, M, L, XL, XXL o número entre 30 y 50)")
70             txtTalla.Focus()
71             Exit Sub
72         End If
73
74         ' ===== VALIDAR PRECIO =====
75         Dim precio As Decimal
76         If Not Decimal.TryParse(
77             txtPrecioUnitario.Text.Trim(),
78             NumberStyles.AllowDecimalPoint,
79             CultureInfo.InvariantCulture,
80             precio
81         ) Then
82             MessageBox.Show("Ingrese un precio válido")
83             txtPrecioUnitario.Focus()
84             Exit Sub
85         End If
86
87         precio = Math.Round(precio, 2)
88
89         ' ===== ACTUALIZAR EN MONGODB =====
90         Try
91             Dim filter = Builders(Of Producto).Filter.Eq(Function(p) p.Id, idProducto)
92
93             Dim update = Builders(Of Producto).Update _
94                 .Set(Function(p) p.Producto, txtProducto.Text.Trim()) _
95                 .Set(Function(p) p.Talla, talla) -
96                 .Set(Function(p) p.TipoMaterial, txtTipoMaterial.Text.Trim()) -
97                 .Set(Function(p) p.PrecioUnitario, Convert.ToDouble(precio))
98
99             productosCollection.UpdateOne(filter, update)
100
```

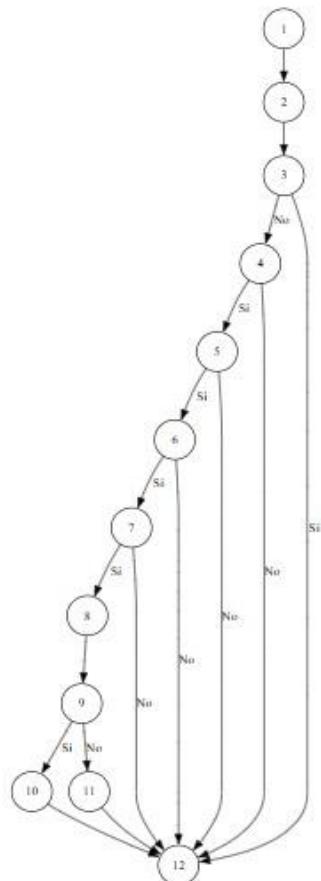
```
100     MessageBox.Show("Producto actualizado correctamente")
101     Me.Close()
102
103     Catch ex As Exception
104         MessageBox.Show("Error al actualizar: " & ex.Message)
105     End Try
106 End Sub
107
108     ' ----- CANCELAR -----
109     Private Sub btnCancelar_Click(sender As Object, e As EventArgs) Handles btnCancelar.Click
110         Me.Close()
111     End Sub
112
113     ' ----- BLOQUEO DE ENTRADAS -----
114
115     ' Producto: solo letras
116     Private Sub txtProducto_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtProducto.KeyPress
117         If Not Char.IsLetter(e.KeyChar) AndAlso
118             Not Char.IsWhiteSpace(e.KeyChar) AndAlso
119             Not Char.IsControl(e.KeyChar) Then
120                 e.Handled = True
121             End If
122     End Sub
123
124     ' Tipo material: solo letras
125     Private Sub txtTipoMaterial_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtTipoMaterial.KeyPress
126         If Not Char.IsLetter(e.KeyChar) AndAlso
127             Not Char.IsWhiteSpace(e.KeyChar) AndAlso
128             Not Char.IsControl(e.KeyChar) Then
129                 e.Handled = True
130             End If
131     End Sub
```

```
132
133     ' Talla: letras o números (no símbolos)
134     Private Sub txtTalla_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtTalla.KeyPress
135         If Not Char.IsLetterOrDigit(e.KeyChar) AndAlso
136             Not Char.IsControl(e.KeyChar) Then
137                 e.Handled = True
138             End If
139     End Sub
140
141     ' Precio: solo números y un punto
142     Private Sub txtPrecioUnitario_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtPrecioUnitario.KeyPress
143         Dim txt As TextBox = CType(sender, TextBox)
144
145         If Not Char.IsDigit(e.KeyChar) AndAlso
146             Not Char.IsControl(e.KeyChar) AndAlso
147             e.KeyChar <> "."c Then
148                 e.Handled = True
149             End If
150
151         ' Solo un punto decimal
152         If e.KeyChar = ".c" AndAlso txt.Text.Contains(".") Then
153             e.Handled = True
154         End If
155     End Sub
156
157 End Class
```

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 → 12	Campos vacíos → el proceso termina sin actualizar
R2	1 → 2 → 3 → 4 → 12	Producto inválido → termina sin actualizar
R3	1 → 2 → 3 → 4 → 5 → 12	Tipo de material inválido → termina sin actualizar
R4	1 → 2 → 3 → 4 → 5 → 6 → 12	Talla inválida → termina sin actualizar
R5	1 → 2 → 3 → 4 → 5 → 6 → 7 → 12	Precio inválido → termina sin actualizar
R6	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 12	Todo correcto → actualización exitosa
R7	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 11 → 12	Todo correcto → error en actualización

5. 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): 6
- A (Número de aristas): 14

Cálculo

Fórmula 1:

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

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

Fórmula 2:

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

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

Resultado

La complejidad ciclomática del REQ 8 es 7, lo que significa que existen **7 caminos básicos independientes** que debes probar para cobertura de caja blanca.

DONDE

P: Número de nodos predicado

A: Número de aristas

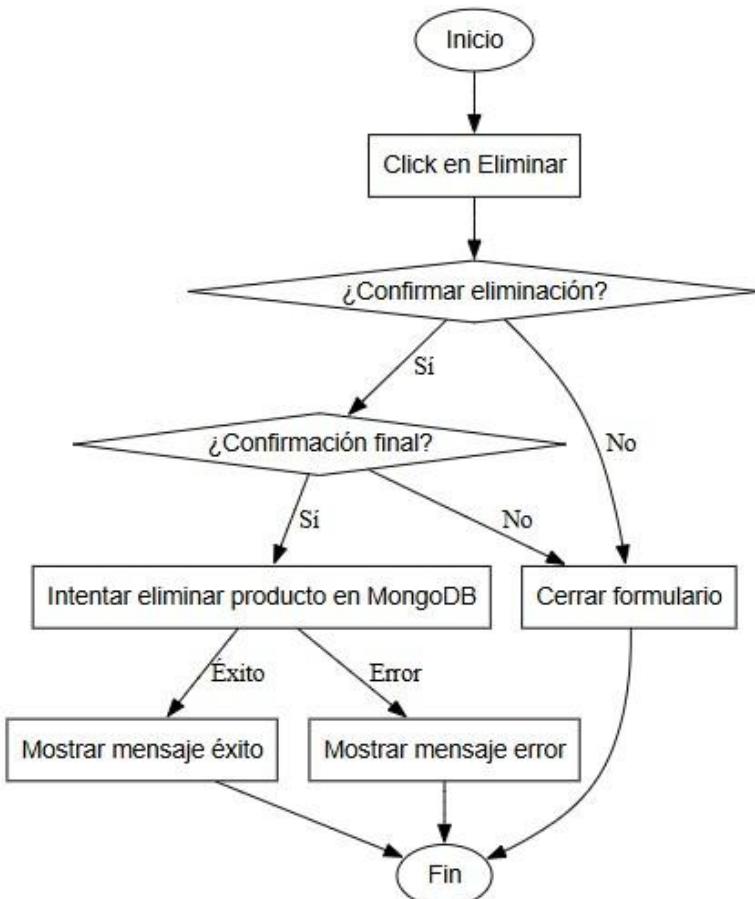
N: Número de nodos

RF N9ª ELIMINACION DE PRODUCTO

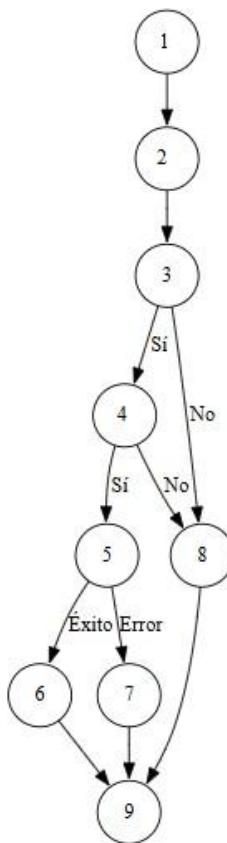
1. CÓDIGO FUENTE

```
28  Private Sub btnEliminar_Click(sender As Object, e As EventArgs) Handles btnEliminar.Click
29      ' Primera confirmación
30      Dim result = MessageBox.Show("¿Estás seguro de eliminar el producto '" & txtNombre.Text & "'?",
31          "Confirmar eliminación",
32          MessageBoxButtons.YesNo,
33          MessageBoxIcon.Warning)
34
35      If result = DialogResult.Yes Then
36          ' Segunda confirmación
37          Dim finalConfirm = MessageBox.Show("¡Esta acción no se puede deshacer! ¿Deseas eliminar este producto?",
38              "Confirmación final",
39              MessageBoxButtons.YesNo,
40              MessageBoxIcon.Warning)
41
42          If finalConfirm = DialogResult.Yes Then
43              Try
44                  Dim db = ConexionMongo.ObtenerConexion()
45                  Dim collection = db.GetCollection(Of BsonDocument)("Productos")
46
47                  Dim filtro = Builders(Of BsonDocument).Filter.Eq(Of ObjectId)("_id", New ObjectId(idProducto))
48                  collection.DeleteOne(filtro)
49
50                  MessageBox.Show("Producto eliminado correctamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information)
51                  Me.Close()
52
53                  Catch ex As Exception
54                      MessageBox.Show("Error al eliminar: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
55                  End Try
56              End If
57          End Sub
58
59          ' ===== BOTÓN CANCELAR =====
60          0 referencias
61          Private Sub btnCancelar_Click(sender As Object, e As EventArgs) Handles btnCancelar.Click
62              Me.Close()
63          End Sub
64
65          0 referencias
66          Private Sub EliminarProducto_Load(sender As Object, e As EventArgs) Handles MyBase.Load
67              ' Inicialización extra si necesitas
68          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 → 8 → 9	Usuario cancela en la primera confirmación → fin del proceso
R2	1 → 2 → 3 → 4 → 8 → 9	Usuario confirma primero, pero cancela en la segunda confirmación → fin del proceso
R3	1 → 2 → 3 → 4 → 5 → 6 → 9	Usuario confirma ambas y eliminación es exitosa → fin del proceso
R4	1 → 2 → 3 → 4 → 5 → 7 → 9	Usuario confirma ambas y ocurre error al eliminar → fin del proceso

5. COMPLEJIDAD CICLOMÁTICA

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

Datos del grafo

- N (Número de nodos): 9
- P (Número de nodos predicados): 3
- A (Número de aristas): 10

Cálculo

Fórmula 1:

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

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

Fórmula 2:

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

$$V(G) = 10 - 9 + 2 = 3$$

Resultado

La complejidad ciclomática del REQ 9 es 7, lo que significa que existen **7 caminos básicos independientes** que debes probar para cobertura de caja blanca.

DONDE

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos