

# 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**

<b>Versión</b>	<b>Fecha</b>	<b>Responsable</b>	<b>Aprobado por</b>
<b>PCB_V1.0.0.docx</b>			

## Prueba caja blanca

### RF N°: REQ010 Envío de Catalogo

0 referencias

```
Private Sub btnEnviarCatalogo_Click(sender As Object, e As EventArgs) _
    Handles btnEnviarCatalogo.Click

    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12

    Try
        If dgvResultados.SelectedRows.Count = 0 Then
            Throw New Exception("Seleccione un cliente.")
        End If

        Dim correoCliente As String =
            dgvResultados.SelectedRows(0).Cells("correo").Value.ToString().Trim()

        If correoCliente = "" Then
            Throw New Exception("El cliente no tiene correo.")
        End If

        ' ===== PDF SIEMPRE NUEVO =====
        Dim rutaPDF As String = Path.Combine(
            Path.GetTempPath(),
            "Catalogo_" & DateTime.Now.Ticks & ".pdf"
        )

        ModuloPDF.GenerarCatalogoPDF(rutaPDF)

        If Not File.Exists(rutaPDF) Then
            Throw New Exception("No se pudo generar el catálogo.")
        End If
```

```
        ' ===== CORREO =====
        Using mensaje As New MailMessage()
            mensaje.From = New MailAddress("kevin1mic2020@gmail.com")
            mensaje.To.Add(correoCliente)
            mensaje.Subject = "Catálogo de Productos - MarcallaTex"
            mensaje.Body = "Adjuntamos nuestro catálogo de productos actualizado."
            mensaje.Attachments.Add(New Attachment(rutaPDF))

            Using smtp As New SmtpClient("smtp.gmail.com", 587)
                smtp.EnableSsl = True
                smtp.UseDefaultCredentials = False
                smtp.Credentials = New NetworkCredential(
                    "kevin1mic2020@gmail.com",
                    "uvjlonhyapgrxidxk"
                )

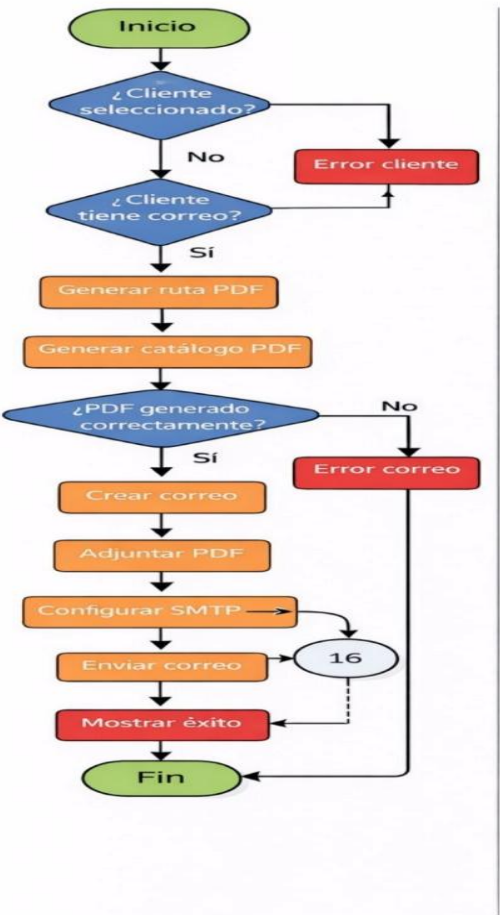
                smtp.Send(mensaje)
            End Using
        End Using

        MessageBox.Show("Catálogo enviado correctamente ☒",
            "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information)

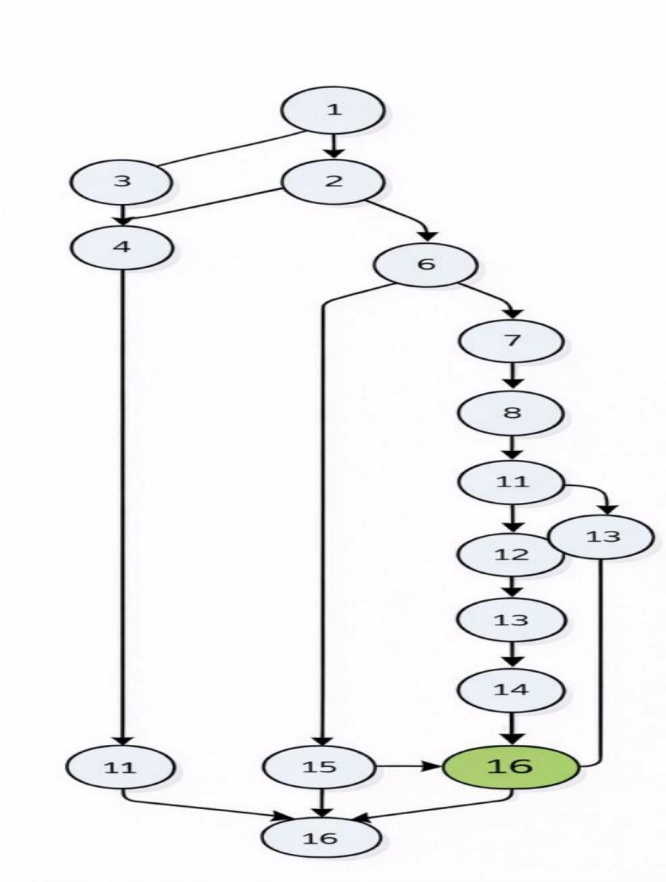
    Catch ex As Exception
        MessageBox.Show(ex.Message,
            "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
```

End Sub

2. DIAGRAMA DE FLUJO (DF)



3. Diagrama de grafo



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

Ruta	Secuencia de nodos	Descripción
R1	1 → 2 → 3 → 16	No hay cliente seleccionado
R2	1 → 2 → 4 → 5 → 16	Cliente sin correo
R3	1 → 2 → 4 → 6 → 7 → 8 → 9 → 16	Error al generar el PDF
R4	1 → 2 → 4 → 6 → 7 → 8 → 10 → 11 → 12 → 13 → 14 → 16	Envío exitoso del catálogo
R5	1 → 2 → 4 → 6 → 7 → 8 → 10 → 11 → 12 → 13 → 15 → 16	Error durante el envío (excepción)

#### 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):** 16
- **P (Número de nodos predicado):** 4
  - Nodo 2 → Cliente seleccionado
  - Nodo 4 → Correo válido
  - Nodo 8 → PDF generado
  - Nodo 13 → Resultado del envío
- **A (Número de aristas):** 19

**Cálculo**

**Fórmula 1**

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

**Fórmula 2**

$$V(G) = A - N + 2$$
$$V(G) = 19 - 16 + 2 = 5$$

**Resultado**

La **complejidad ciclomática del proceso “Enviar Catálogo por Correo” es 5**, lo que indica que existen **cinco caminos básicos independientes** que deben ser considerados en las pruebas del sistema.

## RF N°: REQ011 - Generación de Proforma PDF

### 1. CÓDIGO FUENTE

```
1 referencia
Private Sub ExportarAPDF(contentPanel As FlowLayoutPanel, productos As List(Of DetalleProducto))
    Try
        Dim saveDialog As New SaveFileDialog()
        saveDialog.Filter = "Archivos PDF (*.pdf)|*.pdf"
        saveDialog.FileName = $"Proforma_{clienteSeleccionado.Nombre.Replace(" ", "_")}_{DateTime.Now:yyyyMMdd_HHmss}.pdf"

        If saveDialog.ShowDialog() = DialogResult.OK Then
            rutaPDFGenerado = saveDialog.FileName
            Dim documento As New Document(PageSize.A4, 36, 36, 36, 36)
            Dim writer As PdfWriter = PdfWriter.GetInstance(documento, New FileStream(saveDialog.FileName, FileMode.Create))
            documento.Open()

            Dim fuenteSubtitulo As New PdfFont(PdfFont.FontFamily.HELVETICA, 12, PdfFont.BOLD)
            Dim fuenteNormal As New PdfFont(PdfFont.FontFamily.HELVETICA, 10)
            Dim fLabel As New PdfFont(PdfFont.FontFamily.HELVETICA, 10, PdfFont.BOLD)
            Dim fDato As New PdfFont(PdfFont.FontFamily.HELVETICA, 10, PdfFont.NORMAL)

            Dim rutaLogo As String = Path.Combine(Application.StartupPath, "logo.png")
            If File.Exists(rutaLogo) Then
                Try
                    Dim imagen As iTextSharp.text.Image = iTextSharp.text.Image.GetInstance(rutaLogo)
                    imagen.ScaleToFit(180.0F, 80.0F)
                    imagen.Alignment = iTextSharp.text.Image.ALIGN_CENTER
                    documento.Add(imagen)
                Catch ex As Exception
                    End Try
            End If

            Dim espacio As New Paragraph(" ")
            espacio.SpacingAfter = 5
            documento.Add(espacio)

            Dim linea As New PdfPTable(1)
            linea.WidthPercentage = 100
            Dim celdaLinea As New PdfPCell(New Phrase(""))
            celdaLinea.Border = iTextSharp.text.Rectangle.BOTTOM_BORDER
            celdaLinea.BorderWidthBottom = 2.0F
            celdaLinea.FixedHeight = 5
            celdaLinea.BorderColorBottom = BaseColor.BLACK
            celdaLinea.Padding = 0
            linea.AddCell(celdaLinea)
```

```
        celdaLinea.Padding = 0
        linea.AddCell(celdaLinea)
        documento.Add(linea)

        FontFactory.RegisterDirectory("C:\Windows\Fonts")
        Dim fuenteTitulo As iTextSharp.text.Font = FontFactory.GetFont("Arial", 20, iTextSharp.text.Font.BOLD, BaseColor.BLACK)
        If fuenteTitulo Is Nothing Then
            fuenteTitulo = New PdfFont(PdfFont.FontFamily.HELVETICA, 20, PdfFont.BOLD, BaseColor.BLACK)
        End If

        Dim titulo As New Paragraph("PROFORMA DE COTIZACIÓN", fuenteTitulo)
        titulo.Alignment = Element.ALIGN_CENTER
        titulo.SpacingBefore = 10
        titulo.SpacingAfter = 20
        documento.Add(titulo)

        Dim infoCliente As New PdfPTable(2)
        infoCliente.WidthPercentage = 100
        infoCliente.SetWidths({1.2F, 0.8F})

        Dim celdaIzq As New PdfPCell()
        celdaIzq.Border = iTextSharp.text.Rectangle.BOX
        celdaIzq.Padding = 8
        celdaIzq.BackgroundColor = New BaseColor(245, 245, 245)

        Dim pCliente As New Paragraph()
        pCliente.Add(New Chunk("CLIENTE: ", fLabel))
        pCliente.Add(New Chunk(clienteSeleccionado.Nombre, fDato))
        celdaIzq.AddElement(pCliente)
        Dim pCedula As New Paragraph()
        pCedula.Add(New Chunk("CÉDULA/RUC: ", fLabel))
        pCedula.Add(New Chunk(clienteSeleccionado.Cedula, fDato))
        celdaIzq.AddElement(pCedula)
        Dim pTelf As New Paragraph()
        pTelf.Add(New Chunk("TELÉFONO: ", fLabel))
        pTelf.Add(New Chunk(clienteSeleccionado.Telefono, fDato))
        celdaIzq.AddElement(pTelf)
        Dim pCorreo As New Paragraph()
        pCorreo.Add(New Chunk("CORREO: ", fLabel))
        pCorreo.Add(New Chunk(clienteSeleccionado.Correo, fDato))
        celdaIzq.AddElement(pCorreo)
        Dim pDir As New Paragraph()
        pDir.Add(New Chunk("DIRECCIÓN: ", fLabel))
        pDir.Add(New Chunk(clienteSeleccionado.Direccion, fDato))
```



```

pDir.Add(New Chunk("DIRECCIÓN: ", fLabel))
pDir.Add(New Chunk(clienteSeleccionado.Direccion, fDato))
celdaIzq.AddElement(pDir)

Dim celdaDer As New PdfPCell()
celdaDer.Border = iTextSharp.text.Rectangle.BOX
celdaDer.Padding = 8
celdaDer.BackgroundColor = New BaseColor(245, 245, 245)
celdaDer.VerticalAlignment = Element.ALIGN_MIDDLE

Dim pFecha As New Paragraph()
pFecha.Alignment = Element.ALIGN_LEFT
pFecha.Add(New Chunk("FECHA: ", fLabel))
pFecha.Add(New Chunk(DateTime.Now.ToString("dd/MM/yyyy"), fDato))
celdaDer.AddElement(pFecha)
Dim pHora As New Paragraph()
pHora.Alignment = Element.ALIGN_LEFT
pHora.Add(New Chunk("HORA: ", fLabel))
pHora.Add(New Chunk(DateTime.Now.ToString("HH:mm"), fDato))
celdaDer.AddElement(pHora)

InfoCliente.AddCell(celdaIzq)
InfoCliente.AddCell(celdaDer)
InfoCliente.SpacingAfter = 20
documento.Add(InfoCliente)

Dim tablaProductos As New PdfPTable(6)
tablaProductos.WidthPercentage = 100
tablaProductos.SetWidths({0.6F, 2.0F, 2.5F, 0.8F, 1.0F, 1.0F})

Dim encabezados() As String = {"#", "PRODUCTO", "DESCRIPCIÓN", "CANT.", "PRECIO UNIT.", "SUBTOTAL"}
For Each enc In encabezados
    Dim celda As New PdfPCell(New Phrase(enc, New PdfFont(PdfFont.FontFamily.HELVETICA, 9, PdfFont.BOLD, BaseColor.WHITE)))
    celda.BackgroundColor = New BaseColor(52, 73, 94)
    celda.HorizontalAlignment = Element.ALIGN_CENTER
    celda.Padding = 8
    celda.Border = iTextSharp.text.Rectangle.BOX
    tablaProductos.AddCell(celda)
Next

Dim itemNum As Integer = 1
For Each prod In productos
    Dim celdaNum As New PdfPCell(New Phrase(itemNum.ToString(), fuenteNormal))
    celdaNum.HorizontalAlignment = Element.ALIGN_CENTER

```

```

    Dim celdaNum As New PdfPCell(New Phrase(itemNum.ToString(), fuenteNormal))
    celdaNum.HorizontalAlignment = Element.ALIGN_CENTER
    celdaNum.Padding = 5
    tablaProductos.AddCell(celdaNum)
    Dim celdaProd As New PdfPCell(New Phrase(prod.Producto, fuenteNormal))
    celdaProd.HorizontalAlignment = Element.ALIGN_LEFT
    celdaProd.Padding = 5
    tablaProductos.AddCell(celdaProd)
    Dim textoDesc As String = If(prod.Descripcion Is Nothing, "", prod.Descripcion)
    Dim celdaDesc As New PdfPCell(New Phrase(textoDesc, fuenteNormal))
    celdaDesc.HorizontalAlignment = Element.ALIGN_LEFT
    celdaDesc.Padding = 5
    tablaProductos.AddCell(celdaDesc)
    Dim celdaCant As New PdfPCell(New Phrase(prod.Cantidad.ToString(), fuenteNormal))
    celdaCant.HorizontalAlignment = Element.ALIGN_CENTER
    celdaCant.Padding = 5
    tablaProductos.AddCell(celdaCant)
    Dim celdaPrecio As New PdfPCell(New Phrase("$ " & prod.Precio.ToString("0.00"), fuenteNormal))
    celdaPrecio.HorizontalAlignment = Element.ALIGN_RIGHT
    celdaPrecio.Padding = 5
    tablaProductos.AddCell(celdaPrecio)
    Dim celdaSubtotal As New PdfPCell(New Phrase("$ " & prod.Subtotal.ToString("0.00"), fuenteNormal))
    celdaSubtotal.HorizontalAlignment = Element.ALIGN_RIGHT
    celdaSubtotal.Padding = 5
    tablaProductos.AddCell(celdaSubtotal)
    itemNum += 1
Next

tablaProductos.SpacingAfter = 20
documento.Add(tablaProductos)

Dim lineaInferior As New PdfPTable(1)
lineaInferior.WidthPercentage = 100
Dim celdaLineaInf As New PdfPCell(New Phrase(""))
celdaLineaInf.Border = iTextSharp.text.Rectangle.BOTTOM_BORDER
celdaLineaInf.BorderWidthBottom = 2.0F
celdaLineaInf.FixedHeight = 10
celdaLineaInf.BorderColorBottom = BaseColor.BLACK
celdaLineaInf.Padding = 0
lineaInferior.AddCell(celdaLineaInf)
documento.Add(lineaInferior)

Dim tablaTotales As New PdfPTable(2)
tablaTotales.WidthPercentage = 100

```

```

Dim lineaInferior As New PdfPTable(1)
lineaInferior.WidthPercentage = 100
Dim celdaLineaInf As New PdfPCell(New Phrase(""))
celdaLineaInf.Border = iTextSharp.text.Rectangle.BOTTOM_BORDER
celdaLineaInf.BorderWidthBottom = 2.0F
celdaLineaInf.FixedHeight = 10
celdaLineaInf.BorderColorBottom = BaseColor.BLACK
celdaLineaInf.Padding = 0
lineaInferior.AddCell(celdaLineaInf)
documento.Add(lineaInferior)

Dim tablaTotales As New PdfPTable(2)
tablaTotales.WidthPercentage = 100
tablaTotales.SetWidths({2.0F, 1.0F})
Dim celdaObs As New PdfPCell()
celdaObs.Border = iTextSharp.text.Rectangle.NO_BORDER
celdaObs.AddElement(New Paragraph("OBSERVACIONES:", fuenteSubtitulo))
Dim observacionesTexto As String = If(String.IsNullOrEmpty(txtObservaciones.Text), "Ninguna", txtObservaciones.Text)
celdaObs.AddElement(New Paragraph(observacionesTexto, fuenteNormal))
tablaTotales.AddCell(celdaObs)

Dim subtotal As Decimal = productos.Sum(Function(p) p.Subtotal)
Dim abono As Decimal = 0
Decimal.TryParse(txtAbono.Text.Replace(",", "."), NumberStyles.Any, CultureInfo.InvariantCulture, abono)
Dim total As Decimal = subtotal - abono

Dim celdaTotales As New PdfPCell()
celdaTotales.Border = iTextSharp.text.Rectangle.NO_BORDER
celdaTotales.HorizontalAlignment = Element.ALIGN_RIGHT
celdaTotales.AddElement(New Paragraph($"Subtotal: ${subtotal:0.00}", fuenteSubtitulo))
celdaTotales.AddElement(New Paragraph($"Abono: ${abono:0.00}", fuenteSubtitulo))
celdaTotales.AddElement(New Paragraph(" "))
Dim parrafoTotal As New Paragraph($"Total a Pagar: ${total:0.00}", New PdfFont(PdfFont.FontFamily.HELVETICA, 14, PdfFont.BOLD, BaseColor.RED))
parrafoTotal.Alignment = Element.ALIGN_RIGHT
celdaTotales.AddElement(parrafoTotal)
tablaTotales.AddCell(celdaTotales)
tablaTotales.SpacingAfter = 30
documento.Add(tablaTotales)

documento.Add(New Paragraph(" "))
documento.Add(New Paragraph(" "))
documento.Add(New Paragraph(" "))
Dim firma As New Paragraph("_____ ", fuenteNormal)

```

```

documento.Add(New Paragraph(" "))
documento.Add(New Paragraph(" "))
documento.Add(New Paragraph(" "))
Dim firma As New Paragraph("_____ ", fuenteNormal)
firma.Alignment = Element.ALIGN_CENTER
documento.Add(firma)
Dim textoFirma As New Paragraph("FIRMA Y SELLO", fuenteNormal)
textoFirma.Alignment = Element.ALIGN_CENTER
documento.Add(textoFirma)

documento.Close()
writer.Close()
rutaPDFGenerado = saveDialog.FileName
MessageBox.Show($"PDF generado exitosamente en:{vbCrLf}{saveDialog.FileName}", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information)
End If
Catch ex As Exception
    MessageBox.Show($"Error al generar PDF: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try
End Sub

```



```

0 referencias
Private Sub btnVistaPrevia_Click(sender As Object, e As EventArgs) Handles btnVistaPrevia.Click
    ' 1. Validaciones
    If clienteSeleccionado Is Nothing Then
        MessageBox.Show("Seleccione un cliente válido.", "Cliente", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Exit Sub
    End If

    ' 2. Recolección de datos
    Dim productosList As New List(Of DetalleProducto)
    For Each fila As DataGridViewRow In dgvDetalle.Rows
        If Not fila.IsNewRow AndAlso fila.Cells(0).Value IsNot Nothing AndAlso Not String.IsNullOrEmpty(fila.Cells(0).Value.ToString()) Then
            Dim cant As Integer = 0
            Dim precio As Decimal = 0
            Dim subT As Decimal = 0
            Dim desc As String = fila.Cells(1).Value?.ToString()
            Integer.TryParse(fila.Cells(2).Value?.ToString(), cant)
            Decimal.TryParse(fila.Cells(3).Value?.ToString()?.Replace(",", "."), NumberStyles.Any, CultureInfo.InvariantCulture, precio)
            Decimal.TryParse(fila.Cells(4).Value?.ToString()?.Replace(",", "."), NumberStyles.Any, CultureInfo.InvariantCulture, subT)

            productosList.Add(New DetalleProducto With {
                .Producto = fila.Cells(0).Value.ToString(),
                .Descripcion = desc,
                .Cantidad = cant,
                .Precio = precio,
                .Subtotal = subT
            })
        End If
    Next

    If productosList.Count = 0 Then
        MessageBox.Show("Agregue al menos un producto.", "Productos", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Exit Sub
    End If

    ' 3. Configuración de la Ventana
    Dim visorForm As New Form() With {
        .Text = "Vista Previa de Proforma",
        .Size = New Size(950, 800),
        .StartPosition = FormStartPosition.CenterScreen,
        .BackColor = Color.White ' Fondo del formulario BLANCO
    }

```

```

    }

    ' Panel de fondo (Donde va la hoja) - AHORA BLANCO
    Dim panelPreview As New Panel() With {
        .Dock = DockStyle.Fill,
        .AutoScroll = True,
        .BackColor = Color.White ' <--- CAMBIO AQUÍ: FONDO BLANCO
    }

    ' La "Hoja" de papel
    Dim contentPanel As New FlowLayoutPanel() With {
        .FlowDirection = FlowDirection.TopDown,
        .WrapContents = False,
        .AutoSize = True,
        .AutoSizeMode = AutoSizeMode.GrowAndShrink,
        .Padding = New Padding(40),
        .BackColor = Color.White,
        .BorderStyle = BorderStyle.FixedSingle ' Borde fino para distinguir la hoja del fondo blanco
    }

    ' Generar contenido
    GenerarContenidoProforma(contentPanel, productosList)
    panelPreview.Controls.Add(contentPanel)

    ' Lógica de Centrado
    Dim CentrarHoja = Sub()
        Dim x As Integer = (panelPreview.ClientSize.Width - contentPanel.Width) \ 2
        If x < 10 Then x = 10
        contentPanel.Location = New Point(x, 10)
    End Sub
    AddHandler panelPreview.Resize, Sub(s, ev) CentrarHoja()
    AddHandler visorForm.Shown, Sub(s, ev) CentrarHoja()

    ' 4. Panel de Botones (Inferior)
    Dim panelBotones As New Panel() With {
        .Dock = DockStyle.Bottom,
        .Height = 70,
        .BackColor = Color.FromArgb(52, 73, 94) ' Azul oscuro para el pie
    }

    ' --- BOTÓN VERDE (GENERAR) ---
    Dim btnAccionPDF As New Button() With {
        .Text = "GENERAR / IMPRIMIR PDF",
        .Size = New Size(100, 45)
    }

```

```

        .BackColor = Color.FromArgb(46, 204, 113),
        .ForeColor = Color.White,
        .Font = New WinFont("Segoe UI", 11, FontStyle.Bold),
        .FlatStyle = FlatStyle.Flat,
        .Cursor = Cursors.Hand
    }
    btnAccionPDF.FlatAppearance.BorderSize = 0
    AddHandler btnAccionPDF.Click, Sub(s, ev) ExportarAPDF(contentPanel, productosList)

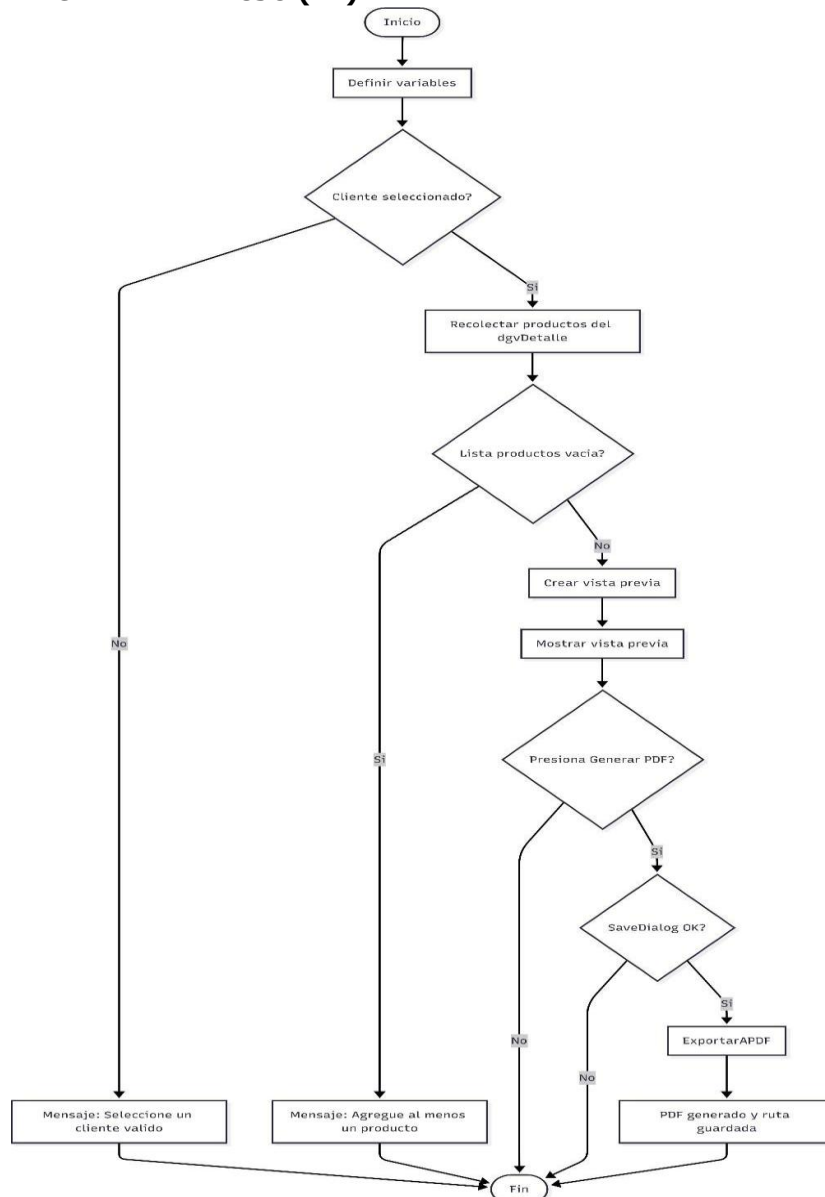
    ' --- BOTÓN ROJO (CERRAR) ---
    Dim btnCerrar As New Button() With {
        .Text = "CERRAR",
        .Size = New Size(150, 45),
        .Location = New Point(visorForm.Width - 190, 12),
        .Anchor = AnchorStyles.Top Or AnchorStyles.Right, ' Para que se mueva al maximizar
        .BackColor = Color.FromArgb(231, 76, 60),
        .ForeColor = Color.White,
        .Font = New WinFont("Segoe UI", 11, FontStyle.Bold),
        .FlatStyle = FlatStyle.Flat,
        .Cursor = Cursors.Hand
    }
    btnCerrar.FlatAppearance.BorderSize = 0
    AddHandler btnCerrar.Click, Sub(s, ev) visorForm.Close()

    ' Agregar botones al panel
    panelBotones.Controls.AddRange({btnAccionPDF, btnCerrar})

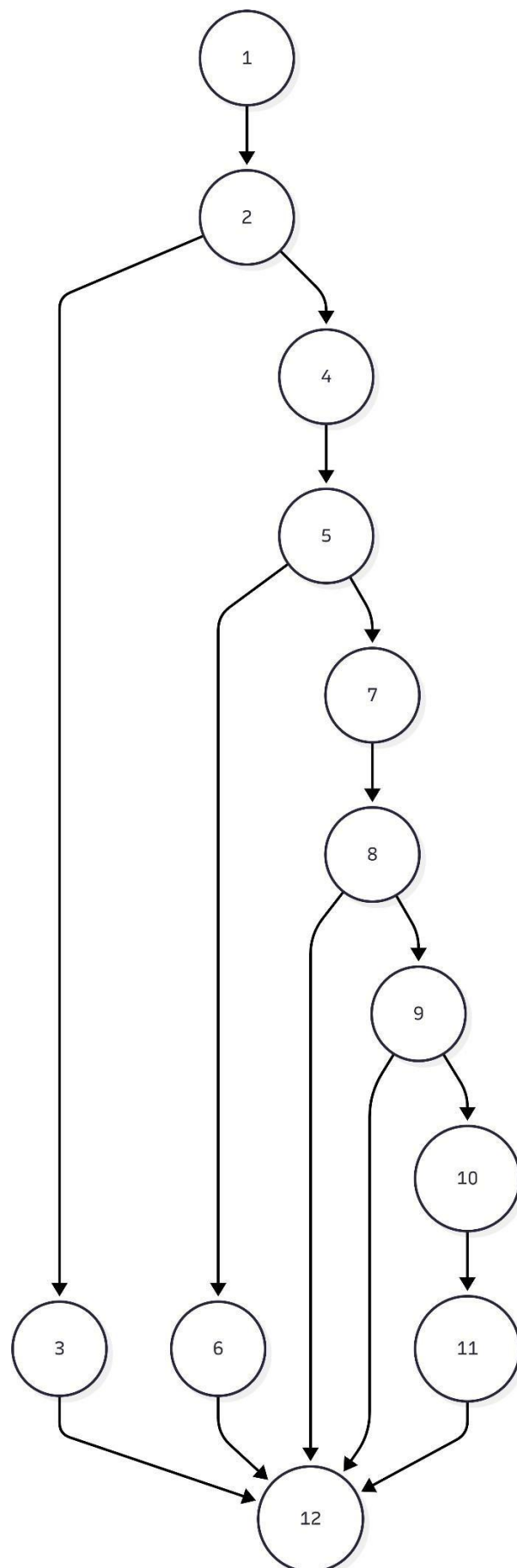
    ' Agregar paneles al formulario
    visorForm.Controls.Add(panelPreview)
    visorForm.Controls.Add(panelBotones)
    visorForm.ShowDialog()
End Sub

```

## 2. DIAGRAMA DE FLUJO (DF)



### 3. GRAFO DE FLUJO (GF)



- 1 = Inicio
- 2 = Validar cliente seleccionado

- 3 = Mensaje cliente invalido
- 4 = Recolectar productos
- 5 = Validar lista productos vacía
- 6 = Mensaje sin productos
- 7 = Crear vista previa
- 8 = Usuario decide generar PDF
- 9 = SaveDialog OK?
- 10 = Exportar PDF
- 11 = Guardar rutaPDFGenerado / mensaje éxito
- 12 = Fin

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

Ruta	Secuencia de nodos	Descripción
<b>R1</b>	1 → 2 → 4 → 5 → 7 → 8 → 9 → 10 → 11 → 12	Flujo correcto: genera PDF
<b>R2</b>	1 → 2 → 3 → 12	No hay cliente seleccionado
<b>R3</b>	1 → 2 → 4 → 5 → 6 → 12	No hay productos
<b>R4</b>	1 → 2 → 4 → 5 → 7 → 8 → 12	No presiona generar PDF
<b>R5</b>	1 → 2 → 4 → 5 → 7 → 8 → 9 → 12	Cancela el SaveDialog

#### 5. COMPLEJIDAD CICLOMÁTICA

Datos del grafo (según nuestro GF)

- N (Numero de nodos): 12
- P (Numero de nodos predicados): 4  
(son los que tienen 2 salidas: 2, 5, 8, 9)
- A (Numero de aristas): 15

Cálculo

Formula 1:

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

Formula 2:

$$V(G) = A - N + 2 = 15 - 12 + 2 = 5$$

**Resultado:** La complejidad ciclomática es 5 (existen 5 caminos básicos).