

Prueba de Caja Blanca

“Gestor de inventario”

Integrantes:

- Mirley Ñacato,
- Melany Torres
- Gael Gonzalez

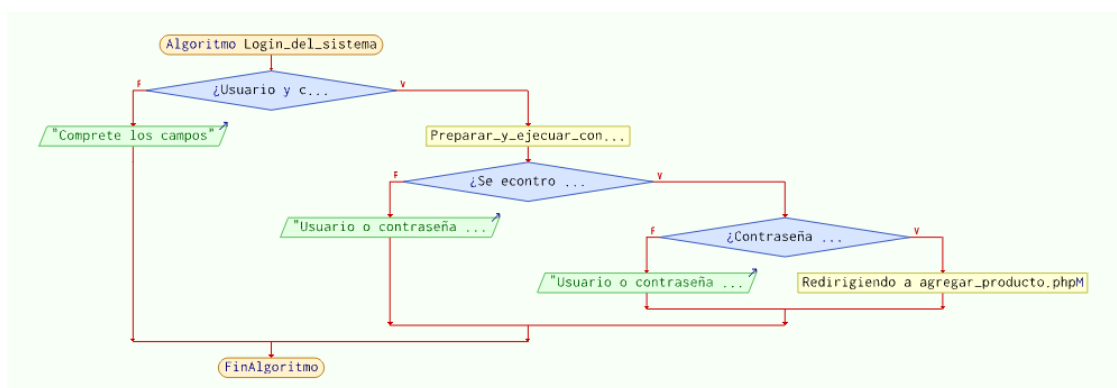
Fecha 2025-06-23

Prueba caja blanca de Login al Sistema

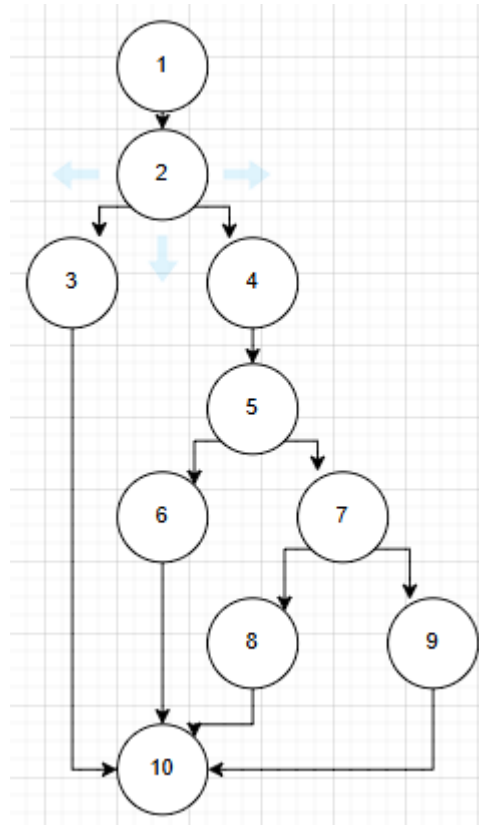
1. CÓDIGO FUENTE

```
13
14  if (!empty($usuario) && !empty($clave)) {
15      $sql = "SELECT password_hash FROM usuarios WHERE usuario = ?";
16      $stmt = $conexion->prepare($sql);
17      $stmt->bind_param("s", $usuario);
18      $stmt->execute();
19      $usuarioEncontrado = $stmt->get_result();
20
21      if ($usuarioEncontrado->num_rows === 1) {
22          $fila = $usuarioEncontrado->fetch_assoc();
23          if (password_verify($clave, $fila['password_hash'])) {
24              $_SESSION['usuario'] = $usuario;
25              header("Location: menu_admin.php");
26              exit;
27          } else {
28              $mensaje = "Usuario o contraseña incorrectos.";
29          }
30      } else {
31          $mensaje = "Usuario o contraseña incorrectos.";
32      }
33
34      $stmt->close();
35  } else {
36      $mensaje = "Por favor complete todos los campos.";
37  }
38
39  $conexion->close();
40  }
41  }
42  ?>
43
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

Rutas posibles:

- **R1:** 1, 2, 3, 10
- **R2:** 1, 2, 4, 5, 6, 10
- **R3:** 1, 2, 4, 5, 7, 8, 10
- **R4:** 1, 2, 4, 5, 7, 9, 10

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=10

Aristas(E)=12

$V(G) = \text{Número de nodos de decisión} + 1$

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

Forma 2:

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

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

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Prueba caja blanca de Agregar un producto

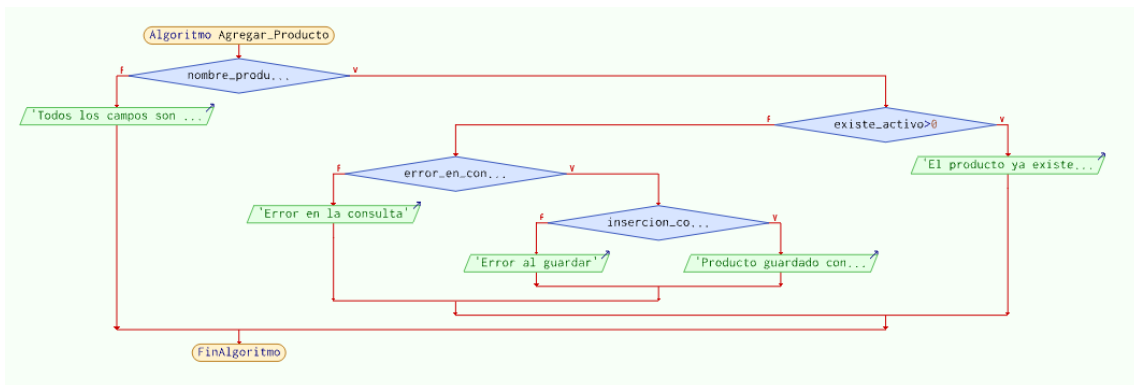
1. CODIGO FUENTE

```

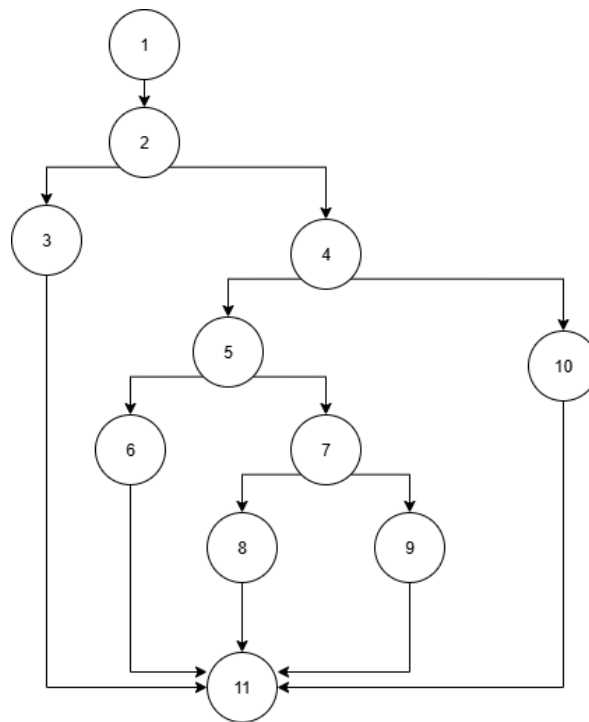
24 if (empty($nombre_producto) && $cantidad > 0 && $precio_compra >= 0) {
25     // Verificar si ya existe un producto activo con ese nombre
26     $verificar = $conexion->prepare("SELECT COUNT(*) FROM agregar WHERE nombre_producto = ? AND eliminado = 0");
27     $verificar->bind_param("s", $nombre_producto);
28     $verificar->execute();
29     $verificar->bind_result($existe_activo);
30     $verificar->fetch();
31     $verificar->close();
32
33     if ($existe_activo > 0) {
34         $mensaje = "⚠ El producto '$nombre_producto' ya existe en el inventario.";
35     } else {
36         // Insertar nuevo producto sin afectar a los que están en papelera
37         $stmt = $conexion->prepare("
38             INSERT INTO agregar (nombre_producto, cantidad, cantidad_anterior, fecha_ingreso, precio_compra, eliminado)
39             VALUES (?, ?, ?, ?, ?, 0)
40         ");
41
42         if ($stmt) {
43             $stmt->bind_param("siisd", $nombre_producto, $cantidad, $cantidad_anterior, $fecha_ingreso, $precio_compra);
44
45             if ($stmt->execute()) {
46                 $id_producto = $conexion->insert_id;
47
48                 // Historial
49                 $accion = 'Nuevo producto';
50                 $detalle = "Se agregó el producto '$nombre_producto'";
51                 $usuario = $_SESSION['usuario'];
52
53                 $historial = $conexion->prepare("
54                     INSERT INTO historial_acciones (producto_id, usuario, accion, detalle, fecha)
55                     VALUES (?, ?, ?, ?, NOW())
56                 ");
57                 $historial->bind_param("iiss", $id_producto, $usuario, $accion, $detalle);
58                 $historial->execute();
59                 $historial->close();
60
61                 $mensaje = "✅ ¡Producto guardado con éxito!";
62             } else {
63                 $mensaje = "❌ Error al guardar: " . $stmt->error;
64             }
65             $stmt->close();
66         } else {
67             $mensaje = "❌ Error en la consulta: " . $conexion->error;
68         }
69     }
70 } else {
71     $mensaje = "❌ Todos los campos son obligatorios y deben ser válidos.";
72 }

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

Rutas posibles:

- **R1:** 1, 2, 3, 11
- **R2:** 1, 2, 4, 5, 6, 11
- **R3:** 1, 2, 4, 5, 7, 8, 11
- **R4:** 1, 2, 4, 5, 7, 9, 11
- **R5:** 1, 2, 4, 10, 11

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=11

Aristas(E)=14

$V(G) = \text{Número de nodos de decisión} + 1$

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

Forma 2:

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

$V(G) = 14 - 11 + 2 = 5$

DONDE:

P: Número de nodos prediado

A: Número de aristas

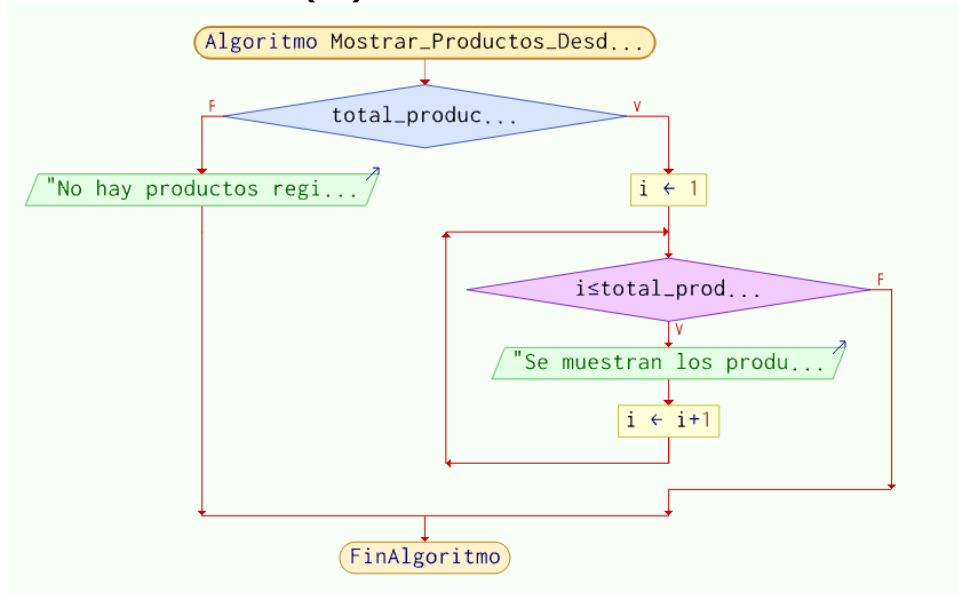
N: Número de nodos

Prueba caja blanca de Visualizar inventario

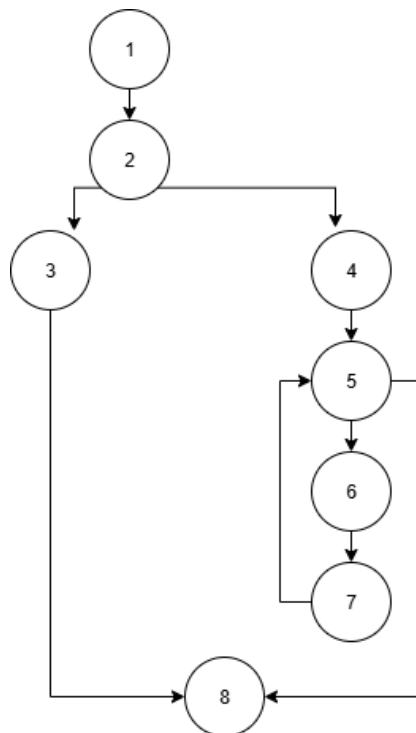
1. CODIGO FUENTE

```
113 <?php if ($productosEncontrados->num_rows > 0): ?>
114 <?php while ($fila = $productosEncontrados->fetch_assoc()): ?>
115 <tr>
116 <td><?=$contador++ ?></td>
117 <td><?=$htmlspecialchars($fila["nombre_producto"]) ?></td>
118 <td><?=$fila["cantidad"] ?></td>
119 <td><?=$fila["fecha_ingreso"] ?></td>
120 <td><?=$number_format($fila["precio_compra"], 2) ?></td>
121 <td>
122 <a href="editar_producto.php?btneditar=<?=$fila["id"] ?>">Editar</a> |
123 <a href="eliminar_producto.php?btneliminar=<?=$fila["id"] ?>" onclick="return confirm('¿Seguro que quieres eliminar este producto?');">Eliminar</a>
124 </td>
125 </tr>
126 <?php endwhile; ?>
127 <?php else: ?>
128 <tr><td colspan="6">No hay productos registrados.</td></tr>
129 <?php endif; ?>
130 </table>
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

- **R1:** 1, 2, 3, 8
- **R2:** 1, 2, 4, 5, 6, 7, 8
- **R3:** 1, 2, 4, 5, 6, 7, 5, 6, 7, 5, ..., 8

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=8

Aristas(E)=9

$V(G) = \text{Número de nodos de decisión} + 1$

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

Forma 2:

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

$V(G) = 9 - 8 + 2 = 3$

DONDE:

P: Número de nodos predicho

A: Número de aristas

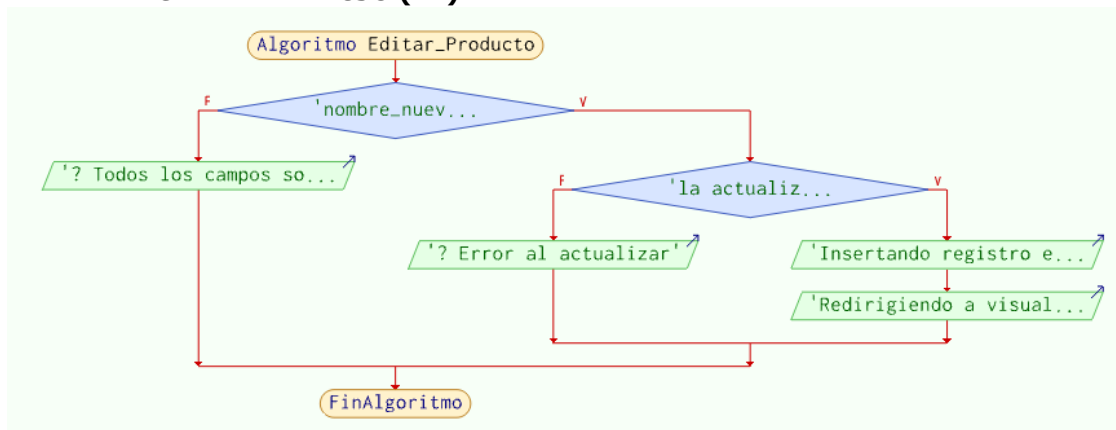
N: Número de nodos

Prueba caja blanca de Edición de productos

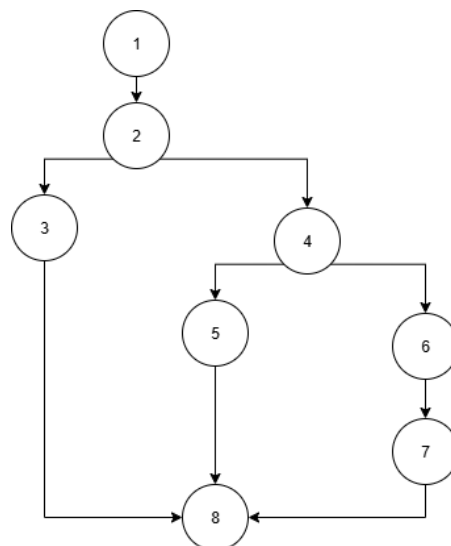
1. CODIGO FUENTE

```
45 if (empty($nombre_nuevo) && $precio_nuevo >= 0) {  
46     $sql = "UPDATE agregar  
47         SET nombre_producto = ?,  
48           precio_compra = ?,  
49           fecha_modificacion = NOW()  
50         WHERE id = ?";  
51     $stmt = $conexion->prepare($sql);  
52     $stmt->bind_param("sdi", $nombre_nuevo, $precio_nuevo, $id);  
53  
54     if ($stmt->execute()) {  
55         // Registro en historial  
56         $nombre_anterior = $producto['nombre_producto'];  
57         $precio_anterior = $producto['precio_compra'];  
58         $detalle = "Nombre: '$nombre_anterior' -> '$nombre_nuevo', Precio: $precio_anterior -> $precio_nuevo";  
59         $usuario = $_SESSION['usuario'];  
60  
61         $historial_sql = "  
62             INSERT INTO historial_acciones (producto_id, usuario, accion, detalle, fecha)  
63             VALUES (?, ?, 'Edición de producto', ?, NOW())  
64         ";  
65         $historial_stmt = $conexion->prepare($historial_sql);  
66         $historial_stmt->bind_param("iss", $id, $usuario, $detalle);  
67         $historial_stmt->execute();  
68         $historial_stmt->close();  
69  
70         header("Location: visualizar_inventario.php");  
71         exit;  
72     } else {  
73         $mensaje = "❌ Error al actualizar: " . $stmt->error;  
74     }  
75     $stmt->close();  
76 } else {  
77     $mensaje = "❌ Todos los campos son obligatorios y deben ser válidos.";  
78 }
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

- **R1:** 1, 2, 3, 8
- **R2:** 1, 2, 4, 5, 8
- **R3:** 1, 2, 4, 6, 7, 8

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=8

Aristas(E)=9

$V(G)$ = Número de nodos de decisión + 1

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

Forma 2:

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

$V(G) = 9 - 8 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Prueba caja blanca de Eliminar producto

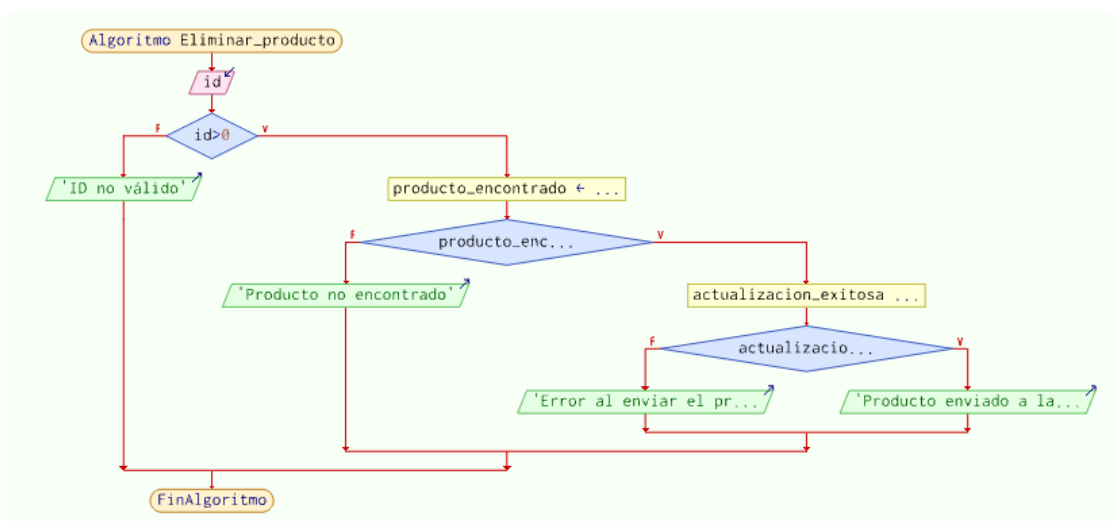
1. CODIGO FUENTE

```

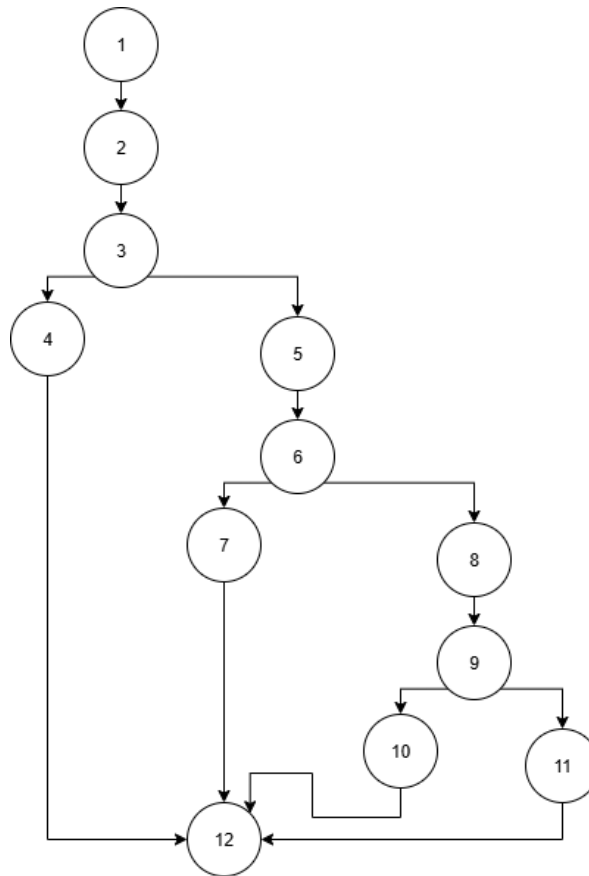
20 ~ if ($id > 0) {
21     // Obtener datos del producto antes de marcarlo como eliminado
22     $consulta = $conexion->prepare("SELECT nombre_producto, precio_compra FROM agregar WHERE id = ?");
23     $consulta->bind_param("i", $id);
24     $consulta->execute();
25     $resultado = $consulta->get_result();
26
27     if ($resultado->num_rows > 0) {
28         $producto = $resultado->fetch_assoc();
29         $consulta->close();
30
31         $nombre = trim($producto['nombre_producto']);
32         $precio = floatval($producto['precio_compra']);
33         $usuario = $_SESSION['usuario'];
34
35         // Marcar como eliminado (soft delete)
36         $stmt = $conexion->prepare("UPDATE agregar SET eliminado = 1 WHERE id = ?");
37         $stmt->bind_param("i", $id);
38
39         if ($stmt->execute()) {
40             // Registrar acción en historial
41             $detalle = "Producto enviado a la papeleria. Nombre: '$nombre', Precio: '$precio'";
42             $historial = $conexion->prepare("
43                 INSERT INTO historial_acciones (producto_id, usuario, accion, detalle, fecha)
44                 VALUES (?, ?, 'Eliminación temporal', ?, NOW())
45             ");
46             $historial->bind_param("iss", $id, $usuario, $detalle);
47             $historial->execute();
48             $historial->close();
49
50             $stmt->close();
51             $conexion->close();
52             header("Location: visualizar_inventario.php?mensaje=Producto+enviado+a+la+papeleria");
53             exit;
54         } else {
55             $stmt->close();
56             $conexion->close();
57             header("Location: visualizar_inventario.php?mensaje=Error+al+enviar+el+producto+a+la+papeleria");
58             exit;
59         }
60     } else {
61         $consulta->close();
62         $conexion->close();
63         header("Location: visualizar_inventario.php?mensaje=Producto+no+encontrado");
64         exit;
65     }
66 } else {
67     $conexion->close();
68     header("Location: visualizar_inventario.php?mensaje=ID+no+válido");
69     exit;
70 }

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

Rutas posibles:

- **R1:** 1, 2, 3, 4, 12
- **R2:** 1, 2, 3, 5, 6, 7, 12
- **R3:** 1, 2, 3, 5, 6, 8, 9, 10, 12
- **R4:** 1, 2, 3, 5, 6, 8, 9, 11, 12

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=12

Aristas(E)=14

V(G) = Número de nodos de decisión + 1

V(G) = 3+1 = 4

Forma 2:

V(G) = A - N + 2

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

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

Prueba caja blanca de Búsqueda de productos

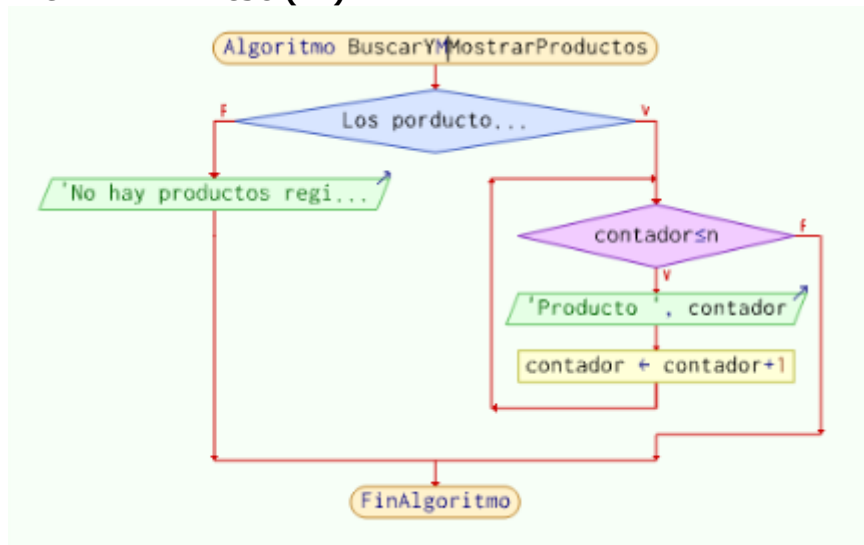
1. CODIGO FUENTE

```

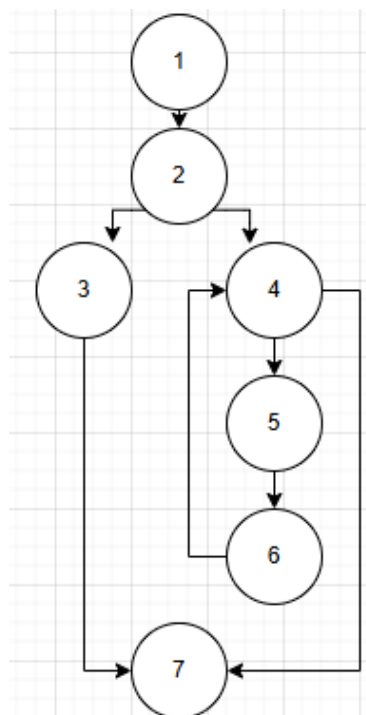
147 <?php if ($productosEncontrados->num_rows > 0): ?>
148 <?php while ($fila = $productosEncontrados->Fetch_assoc()): ?>
149     <tr>
150         <td><?=$contador++ ?></td>
151         <td><?=$htmlspecialchars($fila["nombre_producto"]) ?></td>
152         <td><?=$fila["cantidad_inicial"] ?></td>
153         <td><?=$obtenerStock($conexion, $fila["id"], $fila["cantidad_inicial"]) ?></td>
154         <td><?=$fila["fecha_ingreso"] ?></td>
155         <td><?=$number_format($fila["precio_venta"], 2) ?></td>
156         <td class="acciones">
157             <a href="editar_producto.php?ideditar=<?=$fila["id"] ?>" class="btn-editar">✎ Editar</a>
158             <a href="eliminar_producto.php?ideliminar=<?=$fila["id"] ?>" class="btn-eliminar" onclick="return confirm('¿Seguro que quieres el
159             </td>
160         </tr>
161     <?php endwhile; ?>
162 <?php else: ?>
163     <tr><td colspan="7">No hay productos registrados.</td></tr>
164 <?php endif; ?>

```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

- a. **R1:** 1, 2, 3, 8
- b. **R2:** 1, 2, 4, 5, 6, 7, 8
- c. **R3:** 1, 2, 4, 5, 6, 7, 5, 6, 7, 5, ..., 8

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=8

Aristas(E)=9

$V(G) = \text{Número de nodos de decisión} + 1$

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

Forma 2:

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

$V(G) = 9 - 8 + 2 = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

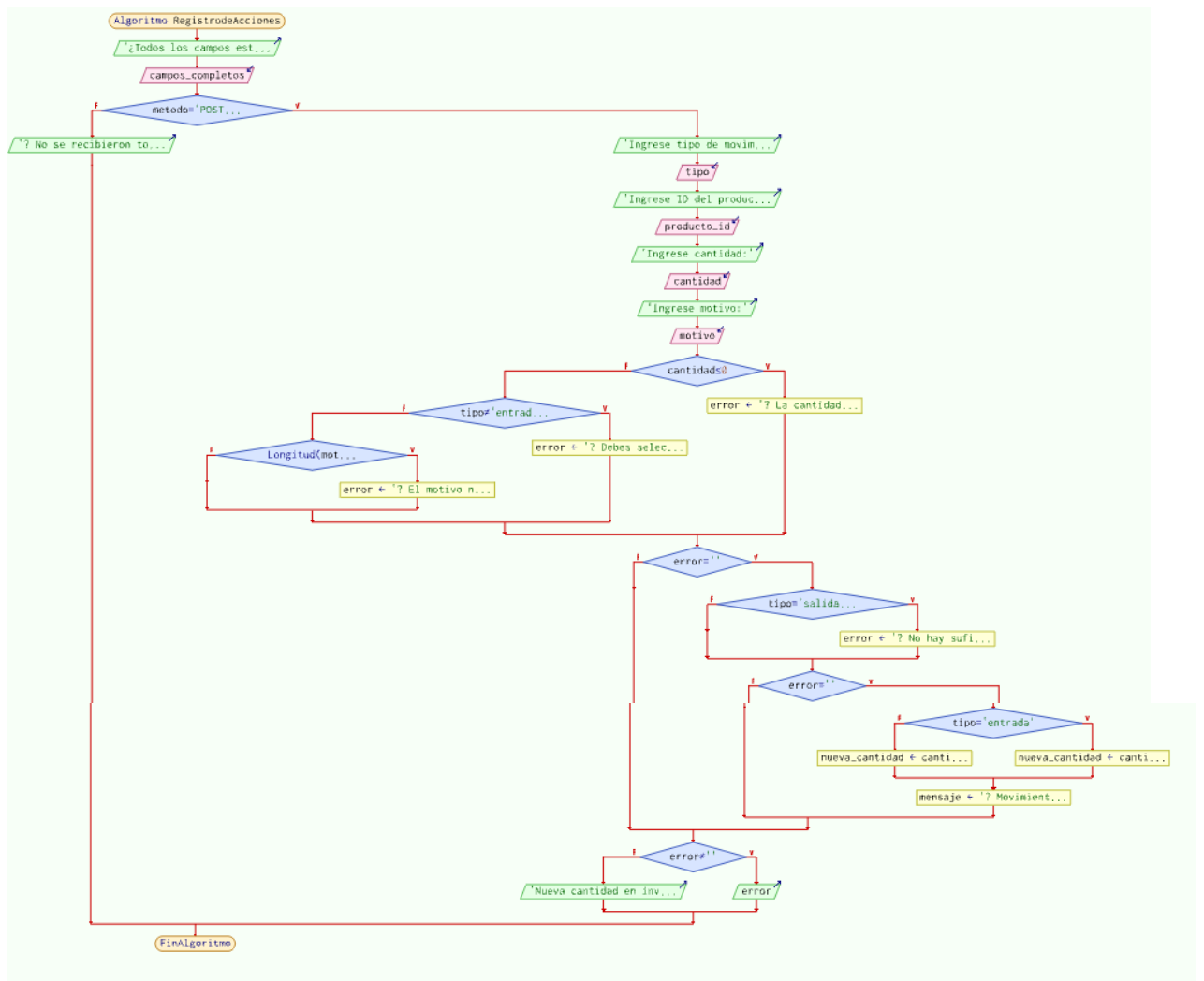
N: Número de nodos

Prueba caja blanca de Gestión de movimientos

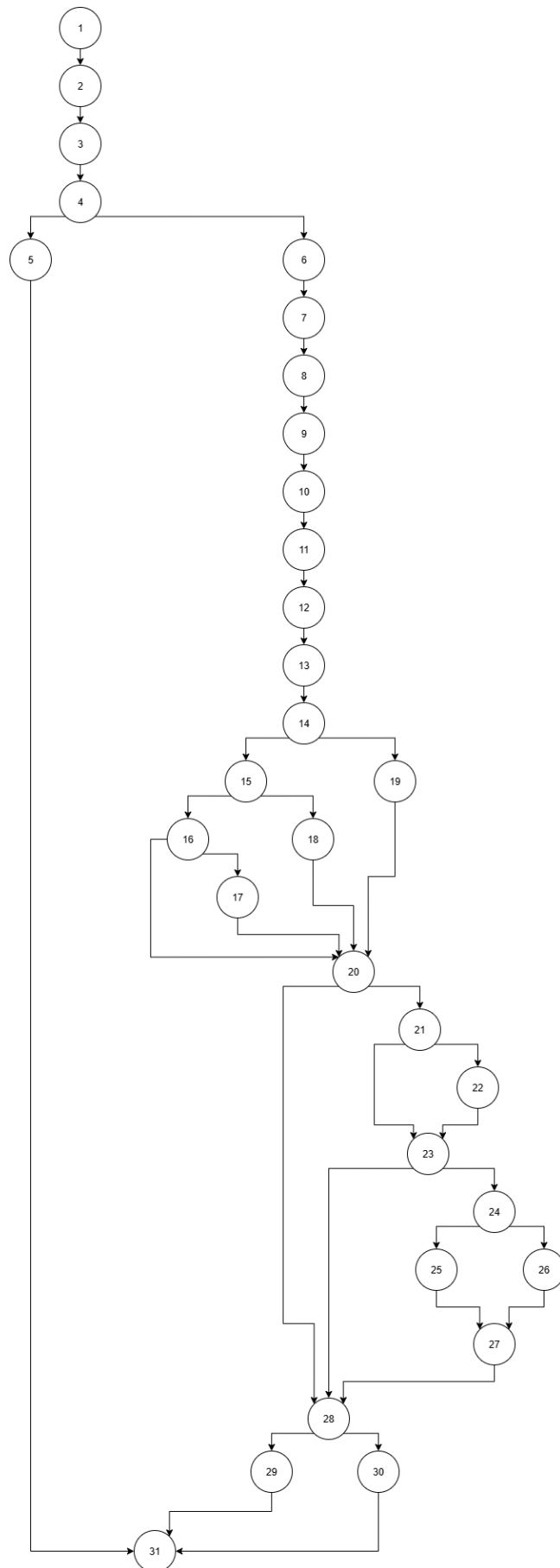
1. CODIGO FUENTE

```
18 if ($SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['tipo'], $_POST['producto_id'], $_POST['cantidad'], $_POST['motivo'])) {
19     $tipo = $_POST['tipo'];
20     $producto_id = (int)$_POST['producto_id'];
21     $cantidad = (int)$_POST['cantidad'];
22     $motivo = trim($conexion->real_escape_string($_POST['motivo']));
23
24     // Validaciones
25     if ($cantidad <= 0) {
26         $error = "❌ La cantidad debe ser mayor que cero.";
27     } elseif ($tipo !== 'entrada' && $tipo !== 'salida') {
28         $error = "❌ Debes seleccionar un tipo de movimiento (ingreso o salida).";
29     } elseif (strlen($motivo) === 0) {
30         $error = "❌ El motivo no puede estar vacío.";
31     }
32
33     if (empty($error)) {
34         // Obtener información del producto
35         $producto_result = $conexion->query("SELECT nombre_producto, cantidad FROM agregar WHERE id = $producto_id AND eliminado = 0");
36         $producto = $producto_result->fetch_assoc();
37
38         if (!$producto) {
39             $error = "❌ Producto no encontrado o fue eliminado.";
40         } else {
41             $nombre_producto = $producto['nombre_producto'];
42             $cantidad_actual = (int)$producto['cantidad'];
43
44             // Verificar stock si es salida
45             if ($tipo === 'salida' && $cantidad > $cantidad_actual) {
46                 $error = "❌ No hay suficiente stock disponible (Stock actual: $cantidad_actual).";
47             }
48
49             if (empty($error)) {
50                 // Registrar el movimiento
51                 $stmt = $conexion->prepare("INSERT INTO movimientos_inventario (producto_id, tipo, cantidad, motivo) VALUES (?, ?, ?, ?)");
52                 $stmt->bind_param("isis", $producto_id, $tipo, $cantidad, $motivo);
53                 if ($stmt->execute()) {
54                     // Guardar en historial
55                     $accion = $tipo === 'entrada' ? 'Entrada de producto' : 'Salida de producto';
56                     $detalle = "Cantidad: $cantidad, Motivo: $motivo";
57                     $usuario = $_SESSION['usuario'];
58
59                     $hist = $conexion->prepare("INSERT INTO historial_acciones (producto_id, usuario, accion, detalle, fecha) VALUES (?, ?, ?, ?, ?)");
60                     $hist->bind_param("iiss", $producto_id, $usuario, $accion, $detalle);
61                     $hist->execute();
62                     $hist->close();
63
64                     // Calcular nueva cantidad
65                     $nueva_cantidad = ($tipo === 'entrada') ? ($cantidad_actual + $cantidad) : ($cantidad_actual - $cantidad);
66
67                     // Actualizar cantidad anterior y nueva cantidad
68                     $upd = $conexion->prepare("UPDATE agregar SET cantidad_anterior = ?, cantidad = ?, fecha_modificacion = NOW() WHERE ID = ?");
69                     $upd->bind_param("iii", $cantidad_actual, $nueva_cantidad, $producto_id);
70                     $upd->execute();
71                     $upd->close();
72
73                     $mensaje = "✅ Movimiento registrado correctamente.";
74                 } else {
75                     $error = "❌ Error al registrar movimiento.";
76                 }
77             }
78             $stmt->close();
79         }
80     }
81 }
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

Rutas posibles:

- **R1:** 1, 2, 3, 4, 5, 31
- **R2:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 28, 29, 31
- **R3:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 20, 21, 22, 23, 28, 29, 31
- **R4:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 19, 20, 21, 22, 23, 28, 29, 31
- **R5:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 23, 28, 29, 31
- **R6:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 20, 21, 23, 28, 29, 31
- **R7:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 19, 20, 21, 23, 28, 29, 31
- **R8:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 28, 30, 31
- **R9:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 18, 20, 21, 22, 23, 24, 26, 27, 28, 30, 31
- **R10:** 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 19, 20, 21, 22, 23, 24, 26, 27, 28, 30, 31

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=31

Aristas(E)=39

$V(G) = \text{Número de nodos de decisión} + 1$

$V(G) = 9+1 = 10$

Forma 2:

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

$V(G) = 39-31 + 2 = 10$

DONDE:

P: Número de nodos prediado

A: Número de aristas

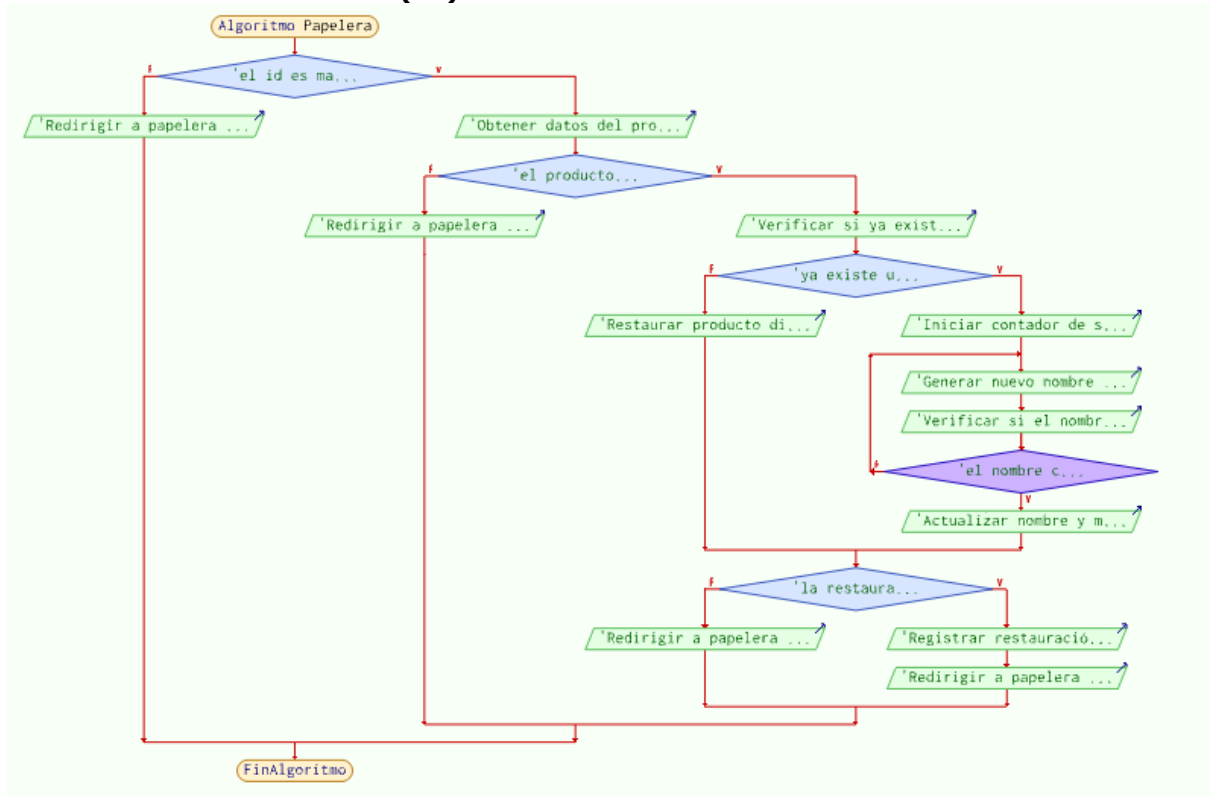
N: Número de nodos

Prueba caja blanca de Papelera de productos

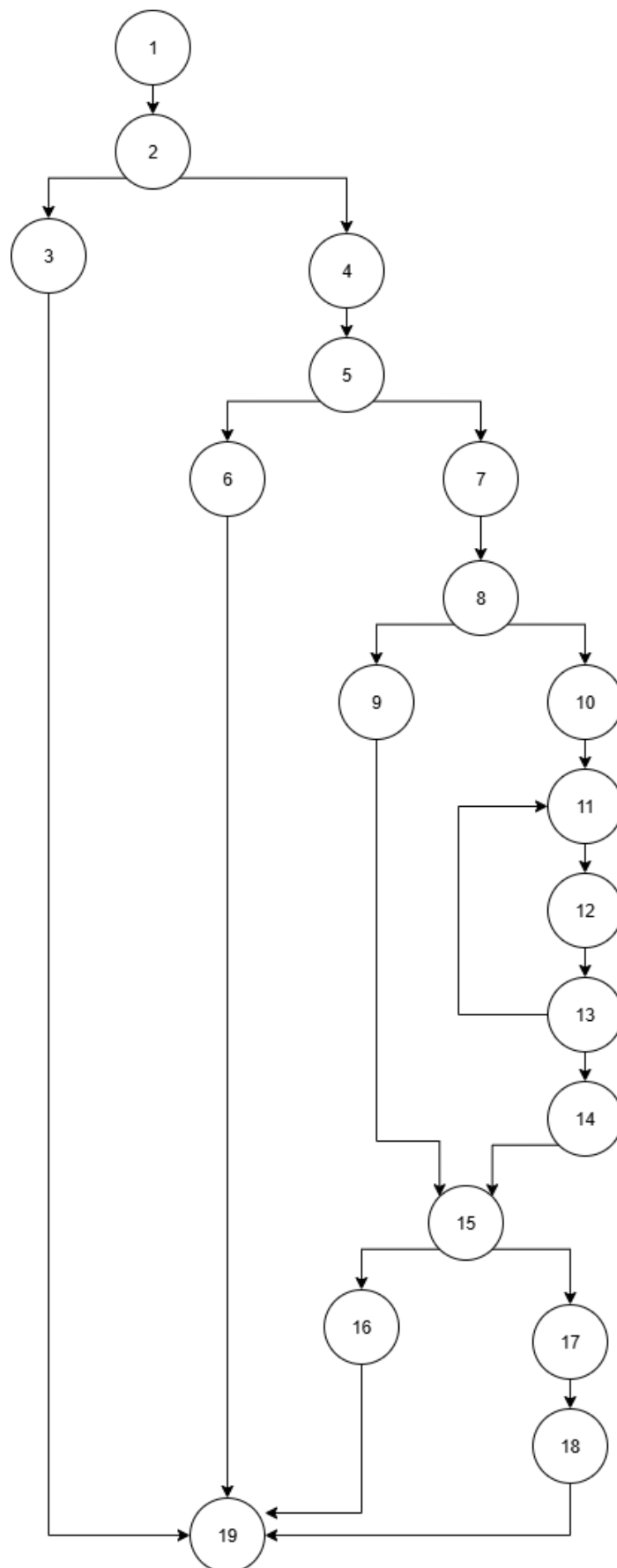
1. CODIGO FUENTE

```
20 if ($id > 0) {
21     // Obtener datos del producto en papelera
22     $consulta = $conexion->prepare("SELECT nombre_producto, precio_compra FROM agregar WHERE id = ?");
23     $consulta->bind_param("i", $id);
24     $consulta->execute();
25     $resultado = $consulta->get_result();
26
27     if ($resultado->num_rows > 0) {
28         $producto = $resultado->fetch_assoc();
29         $consulta->close();
30
31         $nombre_original = $producto['nombre_producto'];
32         $precio = floatval($producto['precio_compra']);
33
34         // Verificar si ya hay un producto activo con el mismo nombre
35         $verificar = $conexion->prepare("SELECT COUNT(*) FROM agregar WHERE nombre_producto = ? AND eliminado = 0");
36         $verificar->bind_param("s", $nombre_original);
37         $verificar->execute();
38         $verificar->bind_result($existe_activo);
39         $verificar->fetch();
40         $verificar->close();
41
42         $nuevo_nombre = $nombre_original;
43
44         if ($existe_activo > 0) {
45             // Generar nombre con sufijo (1), (2), ...
46             $contador = 1;
47             do {
48                 $nuevo_nombre = $nombre_original . " ($contador)";
49                 $consulta_nombre = $conexion->prepare("SELECT COUNT(*) FROM agregar WHERE nombre_producto = ?");
50                 $consulta_nombre->bind_param("s", $nuevo_nombre);
51                 $consulta_nombre->execute();
52                 $consulta_nombre->bind_result($ya_existe);
53                 $consulta_nombre->fetch();
54                 $consulta_nombre->close();
55                 $contador++;
56             } while ($ya_existe > 0);
57
58             // Actualizar nombre + restaurar
59             $stmt = $conexion->prepare("UPDATE agregar SET nombre_producto = ?, eliminado = 0 WHERE id = ?");
60             $stmt->bind_param("si", $nuevo_nombre, $id);
61         } else {
62             // Restaurar directamente sin renombrar
63             $stmt = $conexion->prepare("UPDATE agregar SET eliminado = 0 WHERE id = ?");
64             $stmt->bind_param("i", $id);
65         }
66
67         if ($stmt->execute()) {
68             $stmt->close();
69
70             // Registrar en historial
71             $detalle = "Producto restaurado desde la papelera. Nombre: '$nuevo_nombre', Precio: $precio";
72             $historial = $conexion->prepare("
73             INSERT INTO historial_acciones (producto_id, usuario, accion, detalle, fecha)
74             VALUES (?, ?, 'Restauración desde papelera', ?, NOW())
75             ");
76             $historial->bind_param("iss", $id, $usuario, $detalle);
77             $historial->execute();
78             $historial->close();
79
80             header("Location: papelera.php?mensaje=Producto+restaurado+como:+$nuevo_nombre");
81             exit;
82         } else {
83             header("Location: papelera.php?error=No+se+pudo+restaurar+el+producto");
84             exit;
85         }
86     } else {
87         header("Location: papelera.php?error=Producto+no+encontrado");
88         exit;
89     }
90 } else {
91     header("Location: papelera.php?error=ID+no+válido");
92     exit();
93 }
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS

Rutas posibles:

- **R1:** 1, 2, 3, 19
- **R2:** 1, 2, 4, 5, 6, 19
- **R3:** 1, 2, 4, 5, 7, 8, 9, 15, 16, 19
- **R4:** 1, 2, 4, 5, 7, 8, 9, 15, 17, 18, 19
- **R5:** 1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 16, 19
- **R6:** 1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15, 17, 18, 19

5. COMPLEJIDAD CICLOMÁTICA

Nodos(N)=19

Aristas(E)=23

$V(G) = \text{Número de nodos de decisión} + 1$

$V(G) = 5 + 1 = 6$

Forma 2:

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

$V(G) = 23 - 19 + 2 = 6$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

