

Prueba de Caja Blanca

“Urban Style Shop”

Integrantes:

Kevin Cañola
Alexandro Molina
Christian Marcalla

Fecha: 2025-06-16

REQUISITO 1:

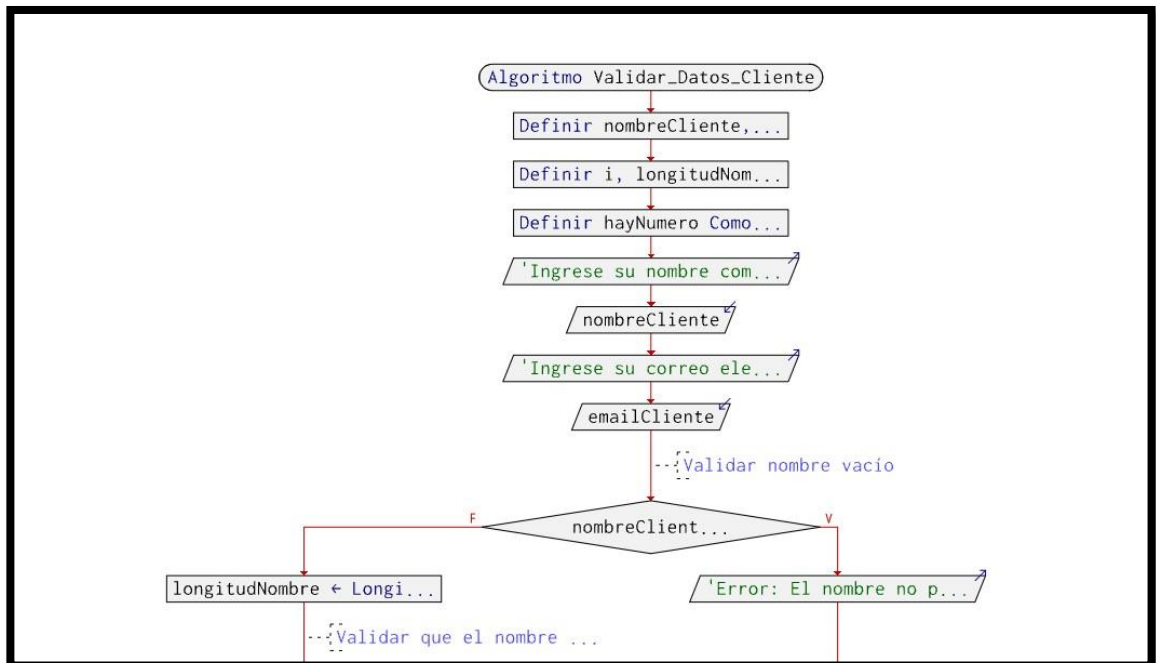
Prueba caja blanca de describa el requisito funcional

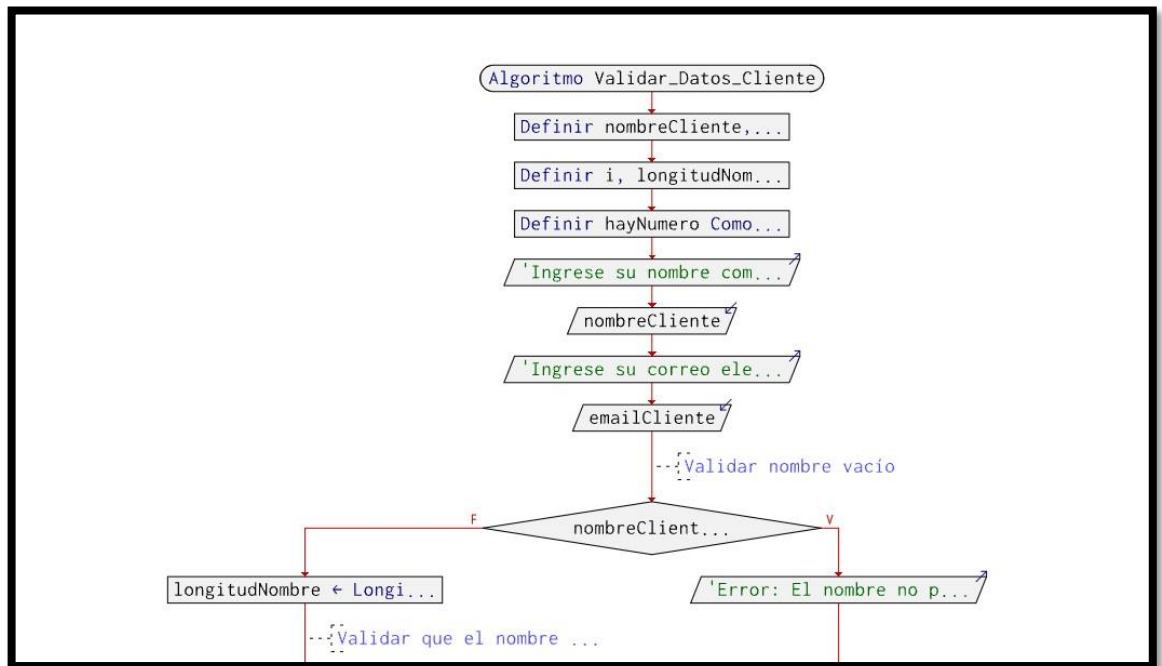
REQUISITO 1. - Registro de Clientes

1. CÓDIGO FUENTE

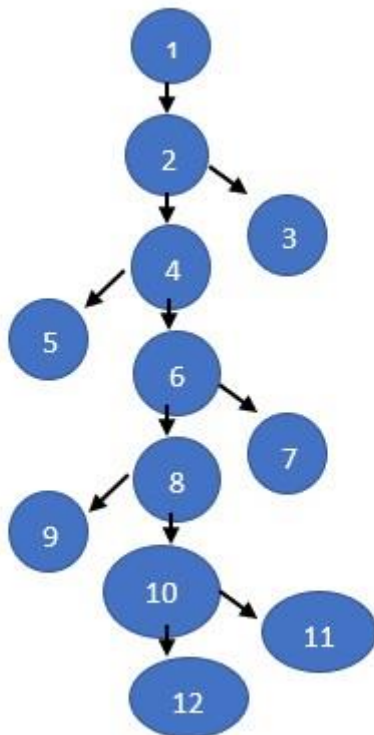
```
C:\> Users > Ordenador > OneDrive > Escritorio > KEVIN > UNIVERSIDAD ESPE > 5 SEMESTRE > MET. DESARROLLO > ESCRITORIO > UrbanStyleShop_V1.5 > carrito.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Urban Style Shop - Carrito</title>
6      <link rel="stylesheet" href="style.css">
7      <script src="https://www.paypal.com/sdk/js?client-id=AVJDAP3ZsmhvzuCmE-FuLfaWkjW-wvFKjUy8Tc5nZFvU9K8zVAa2xpcI0ctqL_7UIBaQLaGhdzsbwa1&currency=USD"></script>
8  </head>
9  <body>
10     <header>
11         <h1>Tu Carrito</h1>
12         <nav>
13             <a href="index.html"> Seguir Comprando </a>
14         </nav>
15     </header>
16
17     <div id="mensaje"></div>
18
19     <section id="carrito"></section>
20
21     <h2>Datos del Cliente</h2>
22     <input type="text" id="nombreCliente" placeholder="Nombre completo"><br>
23     <input type="email" id="emailCliente" placeholder="Correo electrónico"><br>
24     <button onclick="guardarCompra()">Finalizar compra</button>
25     <button onclick="limpiarCarrito()">Vaciar carrito</button>
26
27     <h2>Pago seguro con PayPal</h2>
28     <div id="paypal-button-container"></div>
29
30     <!-- SweetAlert2 debe ir aquí -->
31     <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
32
33     <script type="module" src="script.js"></script>
34 </body>
35 </html>
36
```

2. DIAGRAMA DE FLUJO





3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

RUTAS

R1: 1 → 2 → 3 → 4 → 5 → 6 → 12

R2: 1 → 2 → 7

R3: 1 → 2 → 3 → 8

R4: 1 → 2 → 3 → 4 → 9

R5: 1 → 2 → 3 → 4 → 5 → 10

R6: 1 → 2 → 3 → 4 → 5 → 6 → 11

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = P + 1$

$$P = 5$$

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

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

$$A = 17$$

$$N = 12$$

$$V(G) = 17 - 12 + 2 = 7$$

DONDE:

P: Número de nodos prediado

A: Número de aristas

N: Número de nodos

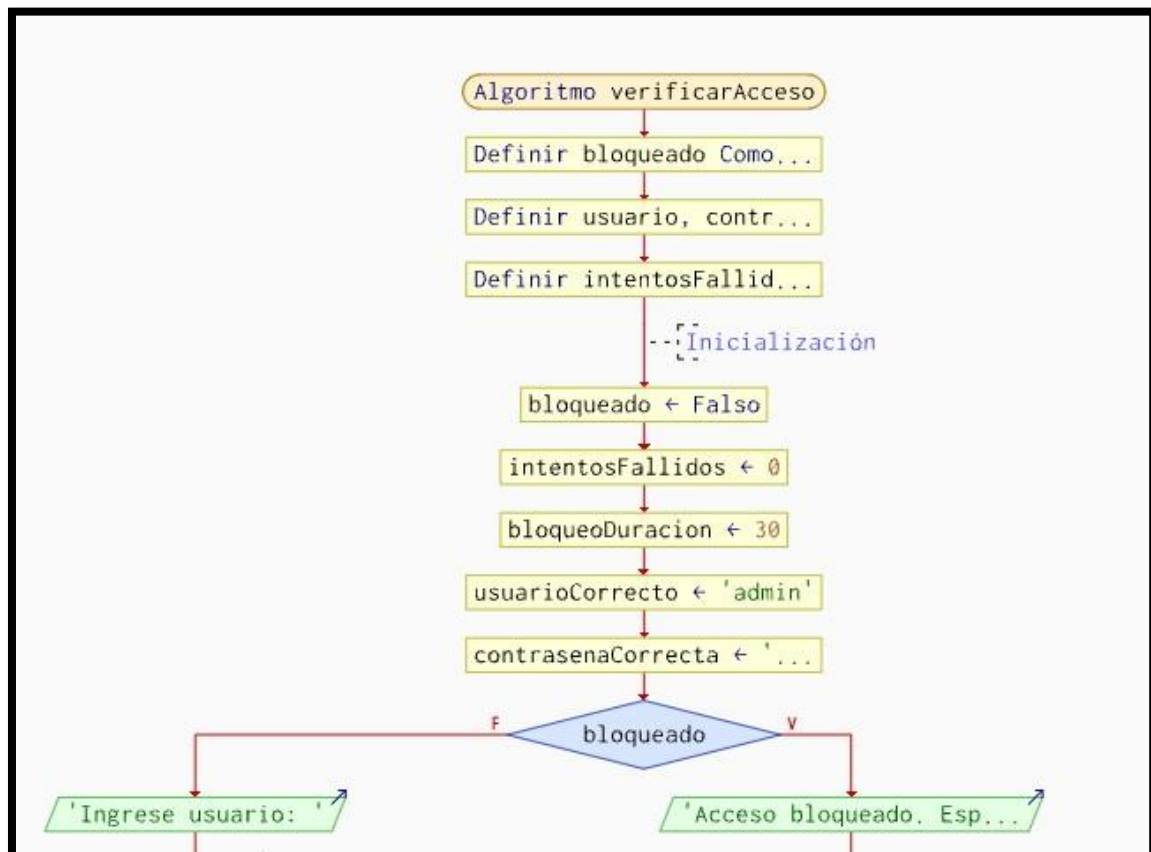
REQUISITO 2. - Registro del Administrador

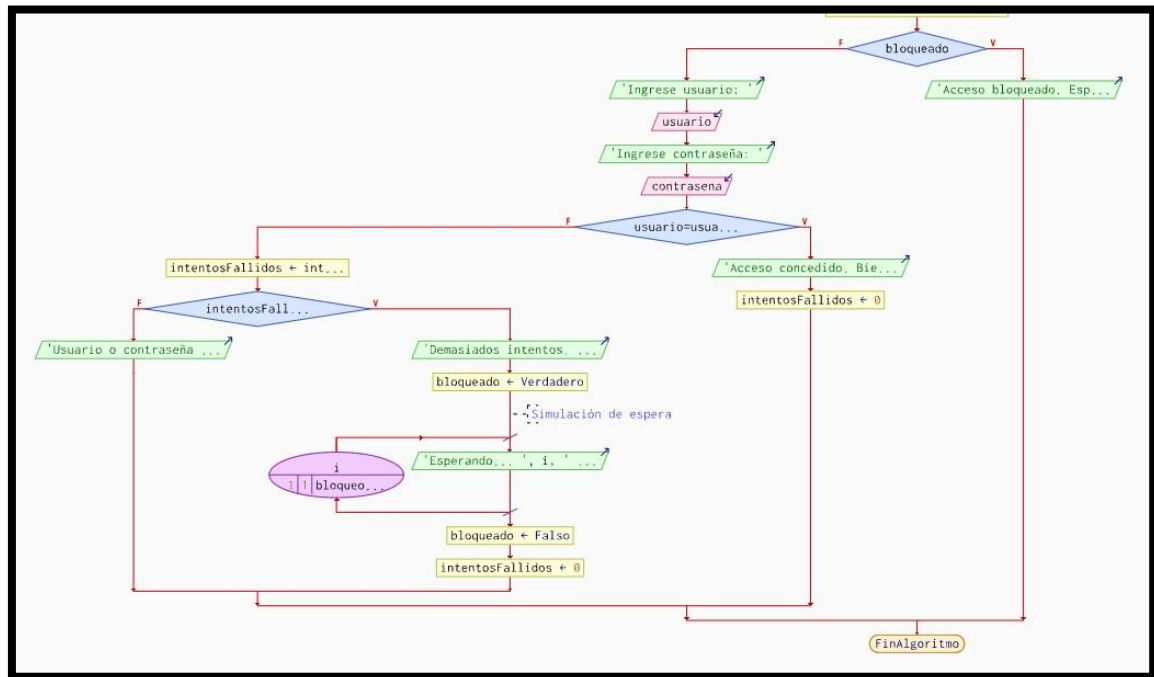
Prueba caja blanca de describe el requisito funcional

1. CÓDIGO FUENTE

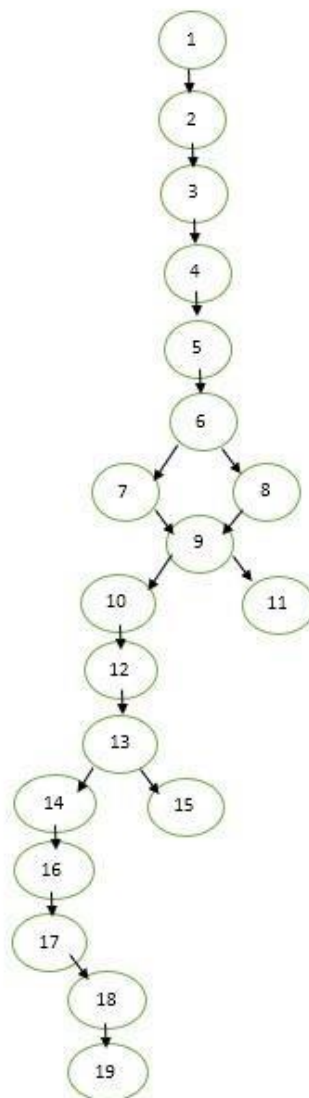
```
352 function verificarAcceso() {
353   if (bloqueado) {
354     mensaje.textContent = "Acceso bloqueado. Espera 30 segundos.";
355     return;
356   }
357   const usuario = document.getElementById("usuario").value.trim();
358   const contrasena = document.getElementById("contrasena").value.trim();
359
360   if (usuario === usuarioCorrecto && contrasena === contrasenaCorrecta) {
361     sessionStorage.setItem("logueado", "true");
362     mensaje.textContent = "";
363     loginForm.style.display = "none";
364     adminPanel.style.display = "block";
365   } else {
366     intentosFallidos++;
367     if (intentosFallidos >= 3) {
368       mensaje.textContent = "Demasiados intentos. Espera 30 segundos.";
369       bloqueado = true;
370       setTimeout(() => {
371         bloqueado = false;
372         intentosFallidos = 0;
373         mensaje.textContent = "";
374       }, bloqueoDuracion);
375     } else {
376       mensaje.textContent = `Usuario o contraseña incorrectos. Intento ${intentosFallidos} de 3.`;
377     }
378   }
379 }
```

2. DIAGRAMA DE FLUJO (DF)





3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

RUTAS

R1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19$

R2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19$ **R3:**

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 17 \rightarrow 18 \rightarrow 19$

R4: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19$

R5: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 17 \rightarrow 18 \rightarrow 19$

5. . COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

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

- $V(G) = A - N + 2$
 $A = 9$
 $N = 10$
 $V(G) = 1$ (Inconsistente)
Se adopta el método 1: $V(G) = 4$

DONDE:

P: Número de nodos prediado

A: Número de aristas

N: Número de nodos

REQUISITO 3. - Sistema deficiente del control de stock

Prueba caja blanca de describa el requisito funcional

1. CÓDIGO FUENTE

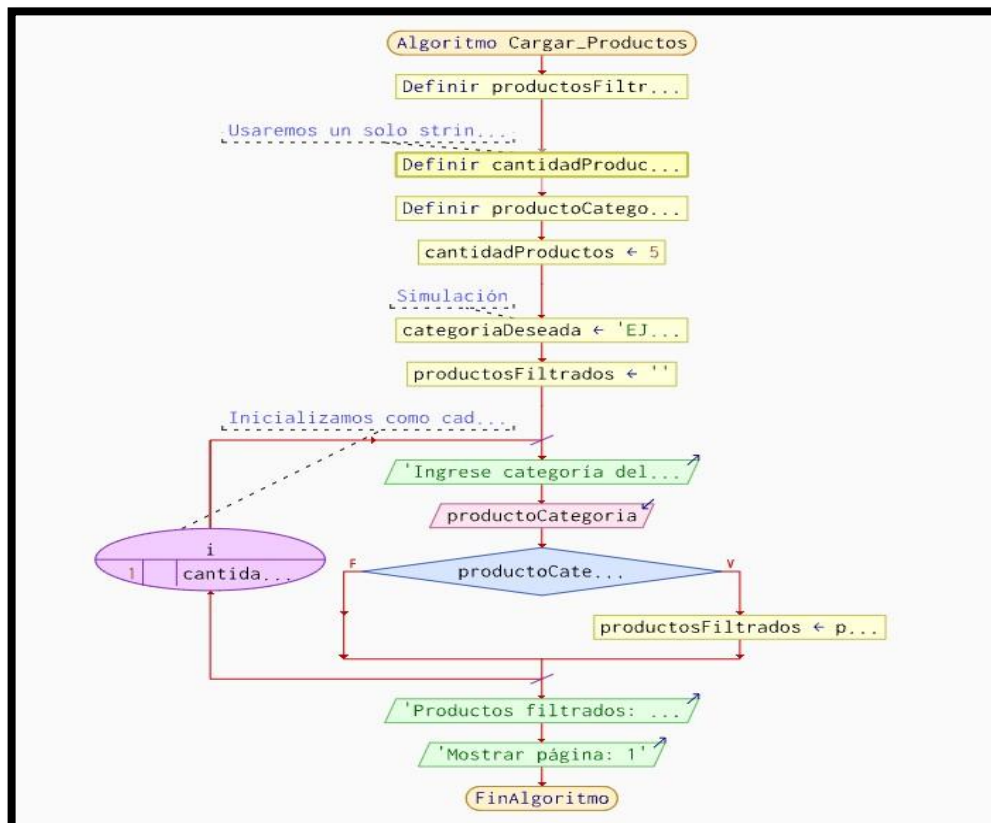
```
// [X] Agregar nuevo producto
const form = document.getElementById("formProducto");
form.addEventListener("submit", async (e) => {
  e.preventDefault();

  const nombre = document.getElementById("nombre").value;
  const precio = parseFloat(document.getElementById("precio").value);
  const stock = parseInt(document.getElementById("stock").value);

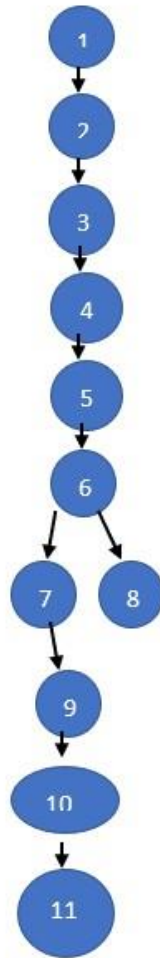
  if (!nombre || isNaN(precio) || isNaN(stock)) {
    Swal.fire("Campos incompletos", "Completa todos los campos", "warning");
    return;
  }

  try {
    await push(dbRef(db, "inventario"), {
      nombre,
      precio,
      stock
    });
    form.reset();
    Swal.fire("Producto agregado", "El producto se guardó correctamente", "success");
  } catch (error) {
    console.error("Error al agregar producto:", error);
    Swal.fire("Error", "No se pudo guardar el producto", "error");
  }
});
```

2. DIAGRAMA DE FLUJO



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

RUTAS

R1: N1 → N2 → N3 → N4 → N5 → N6(no) → N4 → ... → N9 → N10

R2: N1 → N2 → N3 → N4 → N5 → N6(sí) → N8 → N4 → ... → N9 → N10

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

$P = 1$ (único if → nodo predicado)

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

Opción 2:

Contemos:

N (Nodos): 10

A (Aristas): 11

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

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

REQUISITO 5. - Modificación de Stock

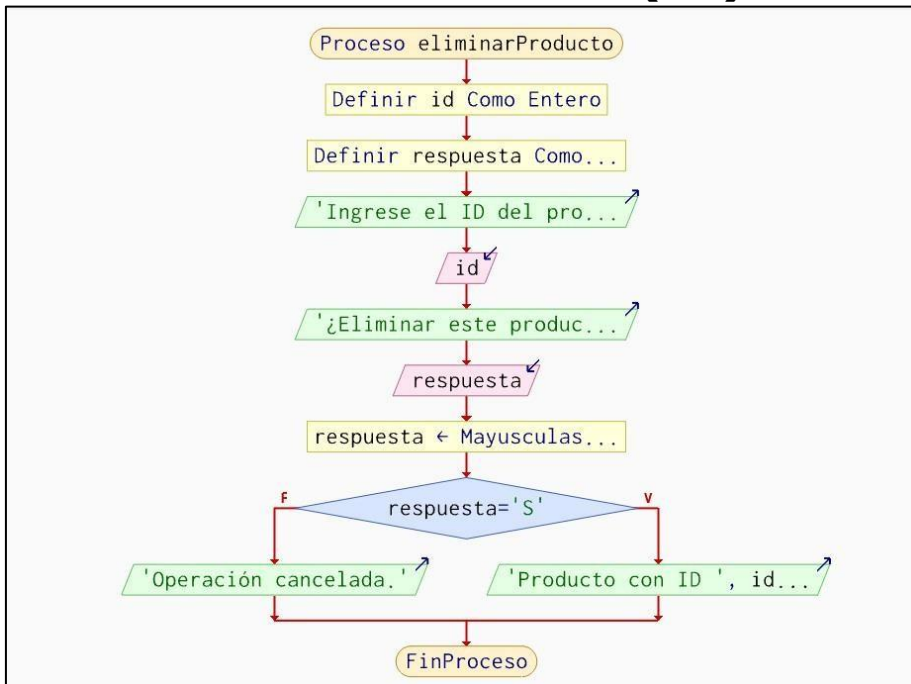
Prueba caja blanca de describa el requisito funcional

1. CÓDIGO FUENTE

```
window.sumarStock = function(id, stockActual) {
  Swal.fire({
    title: '¿Cuántos quieres agregar?',
    input: 'number',
    inputAttributes: {
      min: 1
    },
    showCancelButton: true,
    confirmButtonText: 'Actualizar',
    cancelButtonText: 'Cancelar'
  }).then((result) => {
    if (result.isConfirmed) {
      const nuevoStock = stockActual + parseInt(result.value);
      update(dbRef(db, `inventario/${id}`), { stock: nuevoStock })
        .then(() => {
          Swal.fire('Éxito', 'Stock actualizado', 'success');
        })
        .catch((err) => {
          Swal.fire('Error', err.message, 'error');
        });
    }
  });
};
```

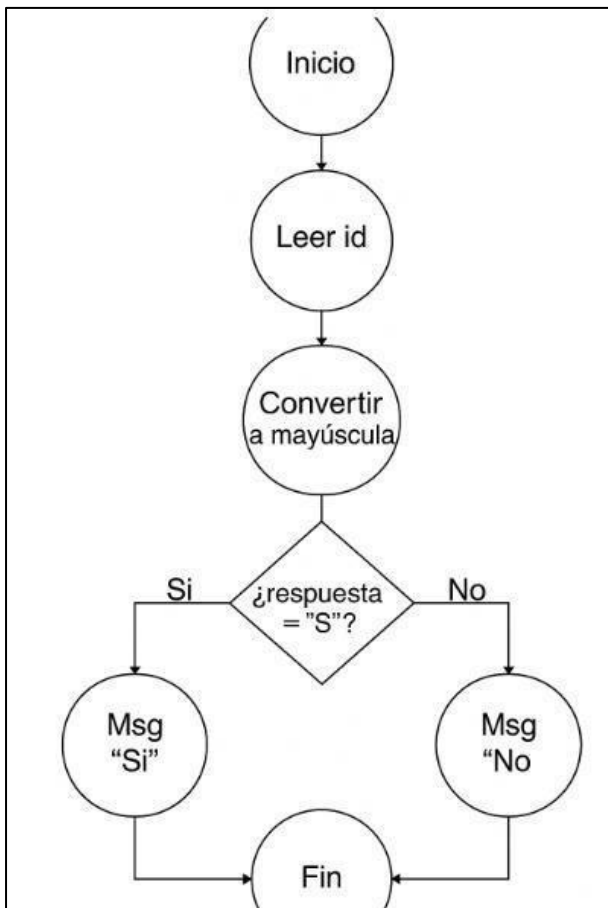
```
window.eliminarProducto = function(id) {
  Swal.fire({
    title: '¿Eliminar este producto?',
    icon: 'warning',
    showCancelButton: true,
    confirmButtonText: 'Sí, eliminar',
    cancelButtonText: 'Cancelar'
  }).then((result) => {
    if (result.isConfirmed) {
      remove(dbRef(db, `inventario/${id}`))
        .then(() => Swal.fire('Eliminado', 'Producto eliminado', 'success'))
        .catch((err) => Swal.fire('Error', err.message, 'error'));
    }
  });
};
```

2. DIAGRAMA DE FLUJO (DF)



a. (GF)

Realizar un GF en base al DF del numeral 2



3. IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Determinar en base al GF del numeral 4

RUTAS

R1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \text{ (Sí)} \rightarrow 6 \rightarrow 8$

R2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \text{ (No)} \rightarrow 7 \rightarrow 8$

4. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichos(decisiones)} + 1$
 $V(G) = 1 + 1 = 2$
- $V(G) = A - N + 2$
 $V(G) = 8 - 8 + 2 = 2$

DONDE:

P: Número de nodos predichos

A: Número de aristas

N: Número de nodos