|  |
| --- |
| Prueba de Caja Blanca |
| ***“Proyecto tienda de ropa”*** |
| Sprint 2  V1.0 |
| **GRUPO N#2**  **Integrantes:**  Gabriel Del Salto Kevin Lechón Paúl Jaramillo |

# Fecha: 29/06/2023

**Requisito 3: Catálogo**

**Código fuente:**

<!DOCTYPE html>

<html>

<head>

<title>Formulario de Catálogo</title>

<script>

function validarFormulario() {

// Obtener los valores ingresados ​​en el formulario

var nombreProducto = document.getElementById("nombreProducto").value;

var precio = document.getElementById("precio").value;

var cantidad = document.getElementById("cantidad").value;

var tallas = document.getElementById("tallas").value;

var color = document.getElementById("color").value;

// Validar que los campos no estén vacíos

if (nombreProducto === "") {

alert("Por favor, ingresa el nombre del producto.");

return false;

}

if (precio === "") {

alert("Por favor, ingresa el precio del producto.");

return false;

}

// Validar que el precio no sea negativo

if (parseFloat(precio) < 0) {

alert("El precio no puede ser un número negativo.");

return false;

}

// Si todos los campos están completos y la validación es exitosa

// puedes realizar alguna acción adicional aquí, como enviar el formulario a un servidor

// Por ahora, simplemente mostramos una alerta para demostrar que el formulario es válido

alert("¡El formulario es válido!");

// Limpiar los campos del formulario

document.getElementById("nombreProducto").value = "";

document.getElementById("precio").value = "";

document.getElementById("cantidad").value = "";

document.getElementById("tallas").value = "";

document.getElementById("color").value = "";

return true;

}

</script>

</head>

<body>

<h1>Formulario de Catálogo</h1>

<form onsubmit="return validarFormulario();">

<label for="nombreProducto">Nombre del producto:</label>

<input type="text" id="nombreProducto" name="nombreProducto" required><br>

<label for="precio">Precio:</label>

<input type="number" id="precio" name="precio" min="0" required><br>

<label for="cantidad">Cantidad:</label>

<input type="number" id="cantidad" name="cantidad" min="0"><br>

<label for="tallas">Tallas:</label>

<input type="text" id="tallas" name="tallas"><br>

<label for="color">Color:</label>

<input type="text" id="color" name="color"><br>

<input type="submit" value="Enviar">

</form>

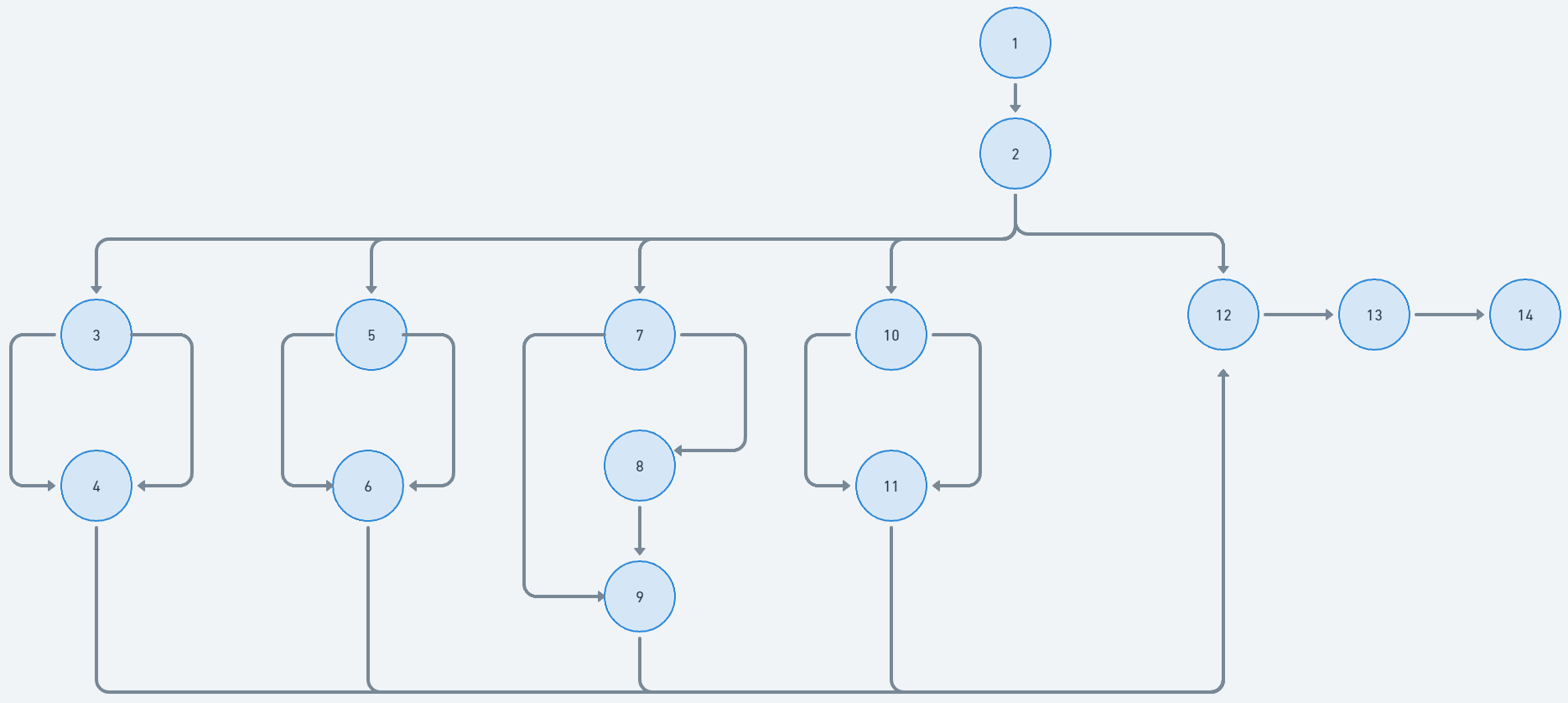
</body>

</html>

# Diagrama UML:

# 

**Grafo:**



**Rutas: (6 rutas)**

R1: 1,2,3,4,12,13,14

R2: 1,2,5,6,12,13,14

R3: 1,2,7,9,12,13,14

R4: 1,2,7,8,9,12,13,14

R5: 1,2,10,11,12,13,14

R6: 1,2,12,13,14

**Definición:**

Cuantas Aristas: 21

Cuantos Nodos: 14

Cuantos nodos Predicados: 5

# Complejidad Ciclo matica:

**V(G)=A-n+2 V(G)=**18-14+2=6

Conclusión: En conclusión, los resultados indican que la complejidad ciclomática de este programa es de 6. Esto significa que existen 6 caminos independientes en la estructura del programa, lo cual puede ser útil para evaluar su complejidad y realizar análisis de cobertura de código.

**Requisito 4: Carrito de compras**

<td class="cart-item\_\_details">

{%- if settings.show\_vendor -%}

<p class="caption-with-letter-spacing">{{ item.product.vendor }}</p>

{%- endif -%}

<a href="{{ item.url }}" class="cart-item\_\_name h4 break">{{ item.product.title | escape }}</a>

{%- if item.original\_price != item.final\_price -%}

<div class="cart-item\_\_discounted-prices">

<span class="visually-hidden">

{{ 'products.product.price.regular\_price' | t }}

</span>

<s class="cart-item\_\_old-price product-option">

{{- item.original\_price | money -}}

</s>

<span class="visually-hidden">

{{ 'products.product.price.sale\_price' | t }}

</span>

<strong class="cart-item\_\_final-price product-option">

{{ item.final\_price | money }}

</strong>

</div>

{%- else -%}

<div class="product-option">

{{ item.original\_price | money }}

</div>

{%- endif -%}

{%- if item.product.has\_only\_default\_variant == false

or item.properties.size != 0

or item.selling\_plan\_allocation != null

-%}

<dl>

{%- if item.product.has\_only\_default\_variant == false -%}

{%- for option in item.options\_with\_values -%}

<div class="product-option">

<dt>{{ option.name }}:</dt>

<dd>{{ option.value }}</dd>

</div>

{%- endfor -%}

{%- endif -%}

{%- for property in item.properties -%}

{%- assign property\_first\_char = property.first | slice: 0 -%}

{%- if property.last != blank and property\_first\_char != '\_' -%}

<div class="product-option">

<dt>{{ property.first }}:</dt>

<dd>

{%- if property.last contains '/uploads/' -%}

<a href="{{ property.last }}" class="link" target="\_blank">

{{ property.last | split: '/' | last }}

</a>

{%- else -%}

{{ property.last }}

{%- endif -%}

</dd>

</div>

{%- endif -%}

{%- endfor -%}

</dl>

<p class="product-option">{{ item.selling\_plan\_allocation.selling\_plan.name }}</p>

{%- endif -%}

<ul class="discounts list-unstyled" role="list" aria-label="{{ 'customer.order.discount' | t }}">

{%- for discount in item.discounts -%}

<li class="discounts\_\_discount">

{%- render 'icon-discount' -%}

{{ discount.title }}

</li>

{%- endfor -%}

</ul>

</td>

<td class="cart-item\_\_totals right medium-hide large-up-hide">

<div class="loading-overlay hidden">

<div class="loading-overlay\_\_spinner">

<svg

aria-hidden="true"

focusable="false"

class="spinner"

viewBox="0 0 66 66"

xmlns="http://www.w3.org/2000/svg"

>

<circle class="path" fill="none" stroke-width="6" cx="33" cy="33" r="30"></circle>

</svg>

</div>

</div>

<div class="cart-item\_\_price-wrapper">

{%- if item.original\_line\_price != item.final\_line\_price -%}

<dl class="cart-item\_\_discounted-prices">

<dt class="visually-hidden">

{{ 'products.product.price.regular\_price' | t }}

</dt>

<dd>

<s class="cart-item\_\_old-price price price--end">

{{ item.original\_line\_price | money }}

</s>

</dd>

<dt class="visually-hidden">

{{ 'products.product.price.sale\_price' | t }}

</dt>

<dd class="price price--end">

{{ item.final\_line\_price | money }}

</dd>

</dl>

{%- else -%}

<span class="price price--end">

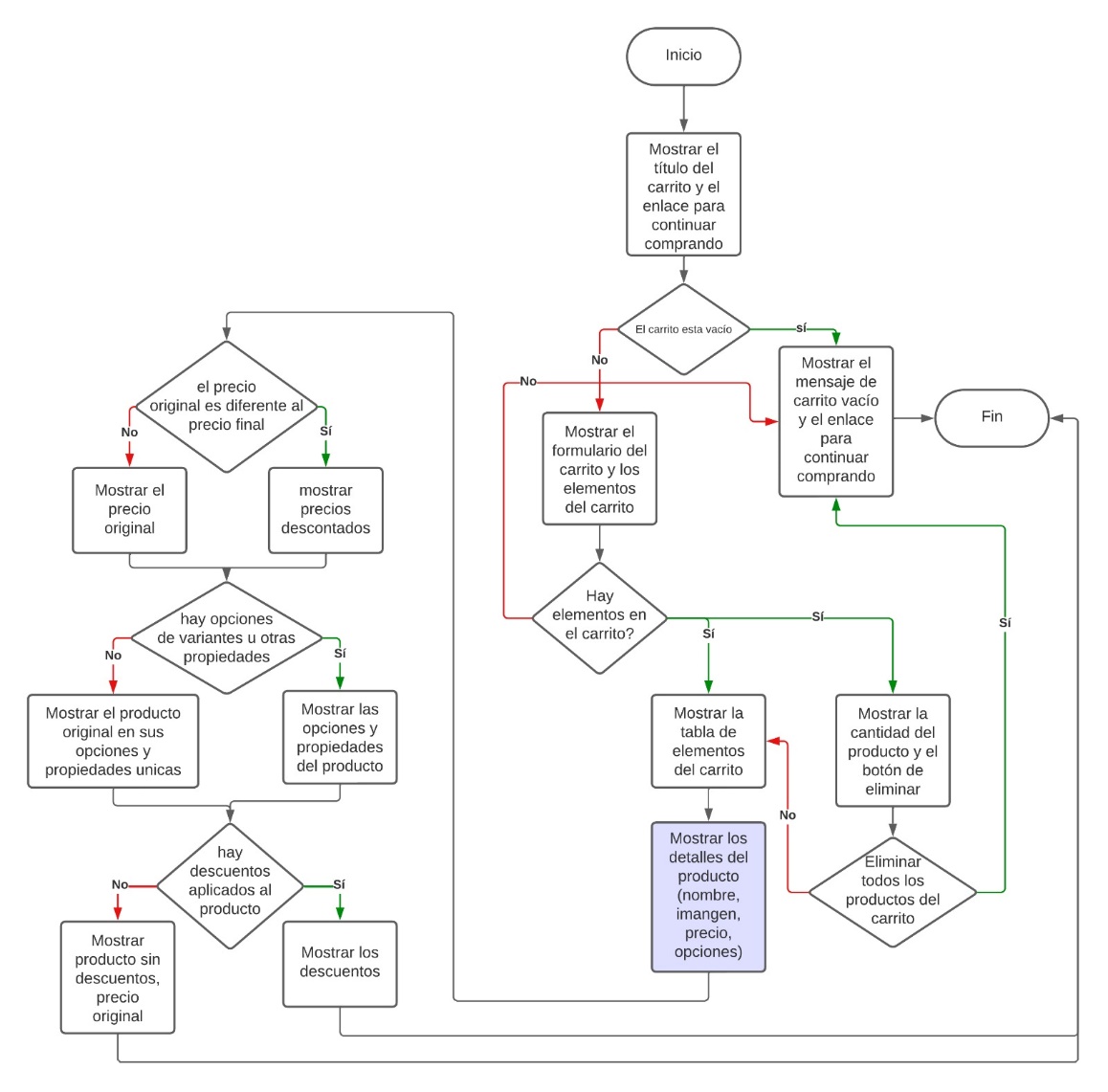
{{ item.original\_line\_price | money }}

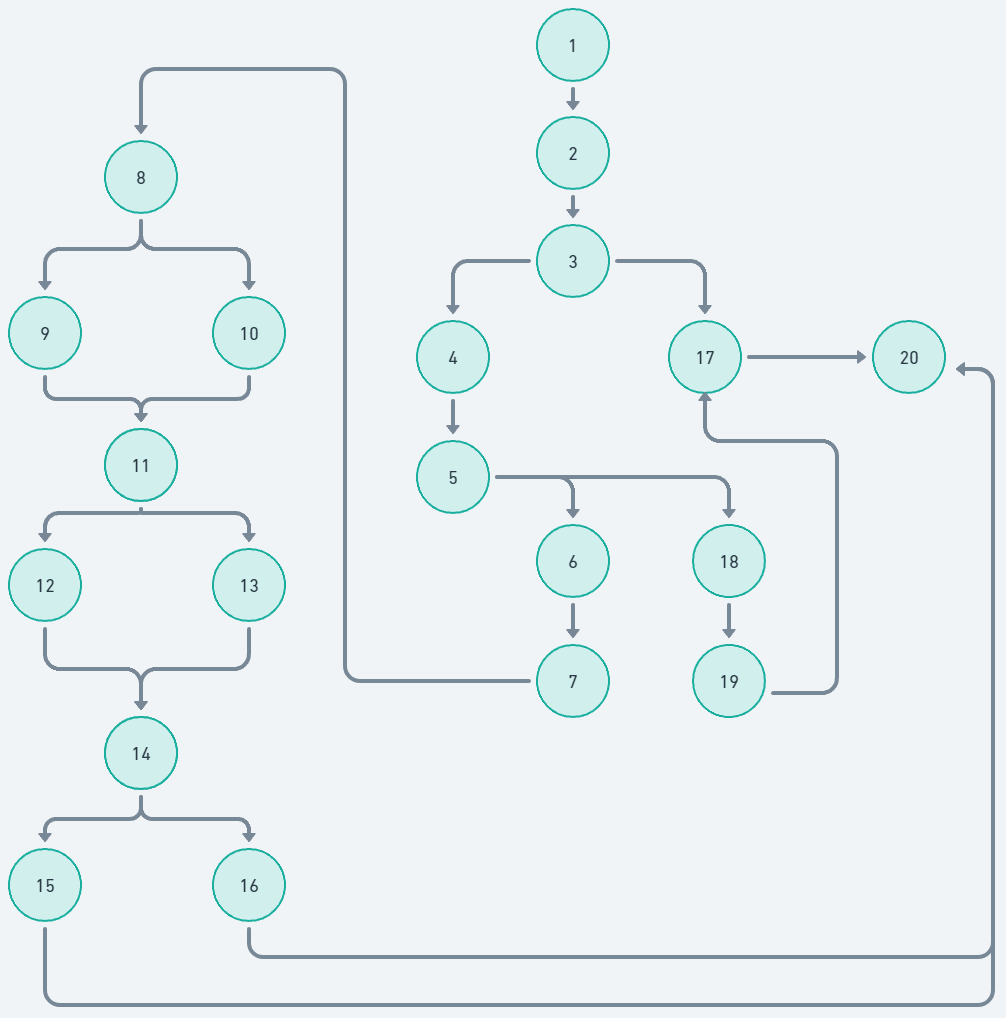
</span>

{%- endif -%}

{% endschema %}

**Diagrama UML**





**Rutas: (6 rutas)**

R1: 1,2,3,17,20

R2: 1,2,3,4,5,18,19,20

R3: 1,2,3,4,5,6,7,8,9,11,12,14,15,20

R4: 1,2,3,4,5,6,7,8,10,11,13,14,16,20

R5: 1,2,3,4,5,6,7,8,9,11,13,14,15,20

R6: 1,2,3,4,5,6,7,8,10,11,12,14,16,20

**Definición:**

Cuantas Aristas: 24

Cuantos Nodos: 20

Cuantos nodos Predicados: 5

# Complejidad Ciclo matica:

**V(G)=A-n+2 V(G)=**24-20+2=6