

# SISTEMA DE INVENTARIO



Presentación a cargo del grupo n°1

# AGENDA



**01** Introducción

**02** Objetivo

**03** Aplicacion

**04** Metodologia

**05** Matriz de historia de usuario

**06** Caja blanca

**07** Caja negra

**08** Conlusión

# INTRODUCCIÓN

Este proyecto surge ante la necesidad de un pequeño negocio de víveres que aún gestiona su inventario y proveedores de forma manual, lo que provoca errores y pérdida de información. Se propone una herramienta simple, en C y ejecutada por consola, para organizar y monitorear productos y proveedores, mejorando la toma de decisiones y la reposición de stock.





# OBJETIVO

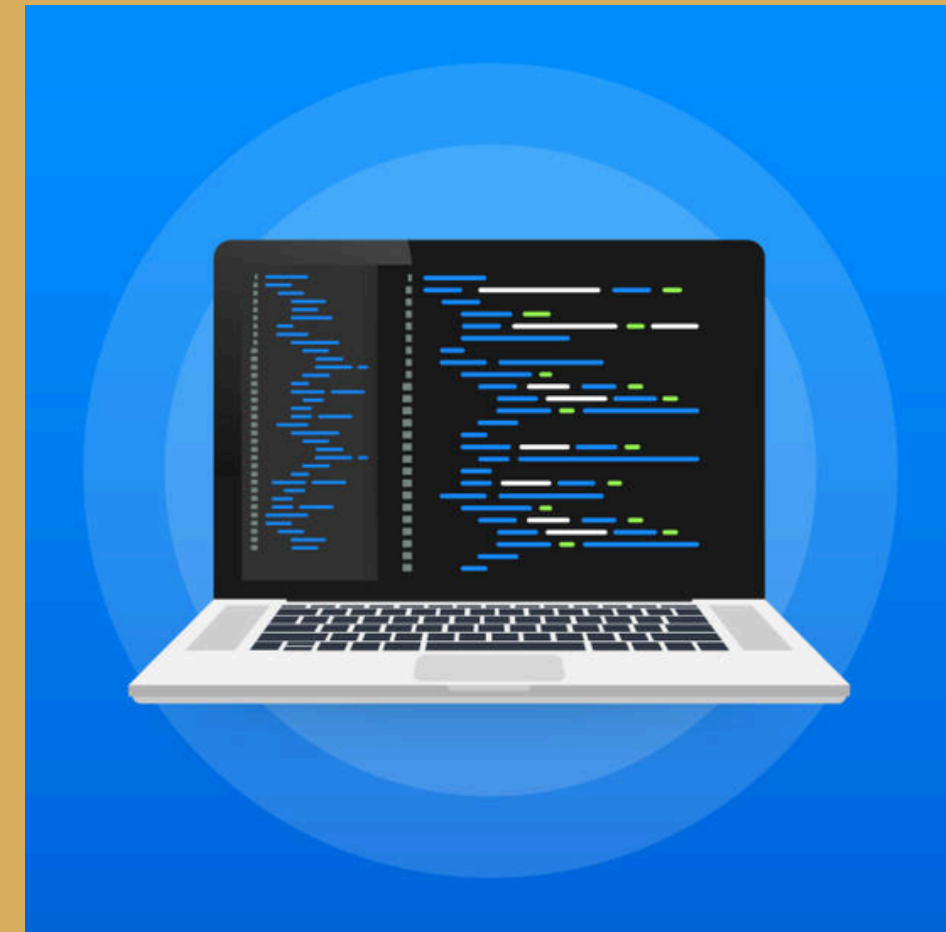
Desarrollar una aplicación de consola en lenguaje C que se ejecute en computadora, la cual permita gestionar de manera eficiente productos y proveedores para un negocio local de víveres y frutos secos, mediante el uso de estructuras de datos y la matriz de historia de usuario con el fin de optimizar los procesos de registro, consulta, modificación y seguimiento del stock disponible.



# APLICACIÓN



Este código en C es un sistema de inventario por consola que permite gestionar productos y proveedores, controlando stock, precios y datos básicos. Guarda la información en archivos de texto y permite realizar operaciones como agregar, editar, buscar y eliminar registros de forma sencilla.



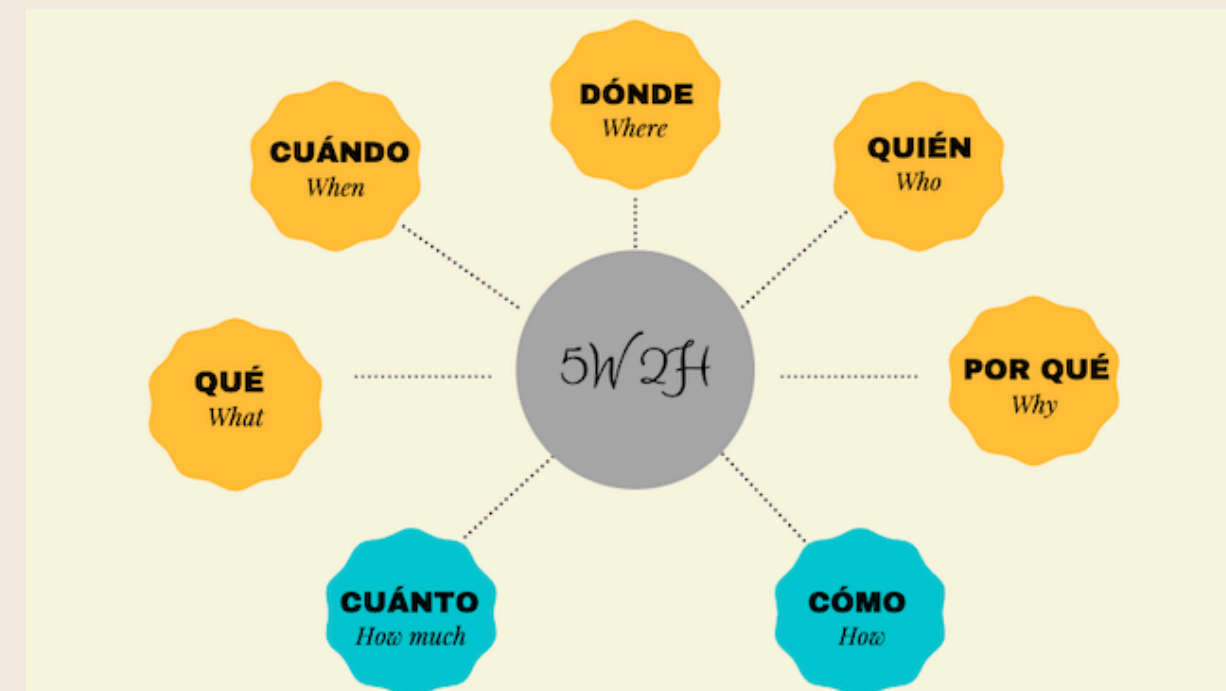
## Ejemplo

```
===== SISTEMA DE INVENTARIO Y PROVEEDORES =====  
---- PRODUCTOS ----  
1. Agregar producto  
2. Mostrar productos  
3. Buscar producto  
4. Editar producto  
5. Eliminar producto  
-----  
---- PROVEEDORES ----  
6. Agregar proveedor  
7. Mostrar proveedores  
8. Buscar proveedor  
9. Editar proveedor  
10. Eliminar proveedor  
-----  
11. Salir  
Seleccione una opcion:
```

# M E T O D O L O G I A



Se trata de una herramienta basada en las preguntas clave del método 5W2H: what, why, who, how y how much, esta herramienta ayuda a estructurar el análisis y la comprensión de una situación o proyecto, identificando los elementos esenciales para la toma de decisiones



# Matriz de historia de usuario

La matriz de historia de usuario (User Story Map) es una herramienta visual de metodologías ágiles que organiza y prioriza historias de usuario, permitiendo entender sus necesidades, planificar el desarrollo y detectar mejoras.



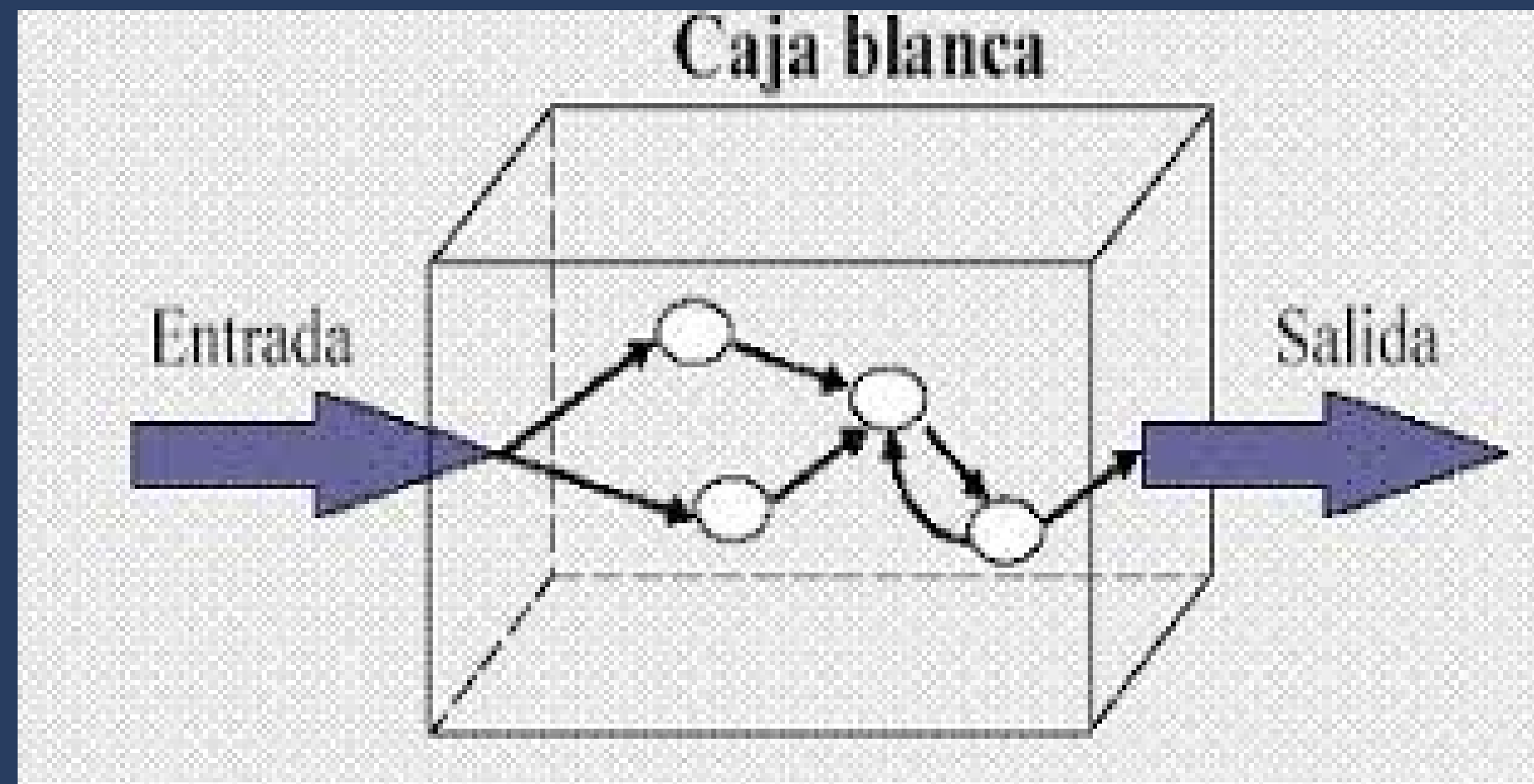


Ejemplo

ITEM	PROBLEMA	QUE (NECESIDAD)	PARA QUE (SOLUCIÓN)	PARA QUIEN (USUARIO)	COMO (DESCRIPCIÓN DE TAREAS)	HECHO POR (PROG. RESP.)	TIEMPO (ESTIMADO EN HRS)	FECHA DE ENTREGA	PRIORIDAD	STATUS	PRUEBA (COMO SE VERIFICA)	COMENTARIOS	NOMBRE DE HISTORIA
REQ001	¿Cómo elijo entre las opciones del programa?	El sistema debera iniciar con un menú principal con opciones separadas y visualmente claras para productos y proveedores.	Facilitar navegación.	Sra. Hilda Guajala	Crear menú con secciones separadas para productos y proveedores.	Mateo Velecela	2h	2025-07-22	Alta	En proceso	Caja negra: Probar si el usuario puede ver y elegir entre las opciones del programa. Caja blanca: Verificar que se muestren correctamente las opciones.		Inicio del programa
REQ002	¿Cómo ingreso los productos del inventario al programa?	: El sistema debera agregar productos con ID de 10 dígitos (autogenerado o relleno con ceros).	Registrar productos en el sistema	Sra. Hilda Guajala	Solicitar datos, validar y guardar con ID autogenerado o corregido.	Mateo Velecela	3h	2025-07-22	Alta	No iniciado	Caja negra: Producto aparece en listado Caja blanca: Verificar que el sistema guarde los datos del producto.		Ingrsesos del producto
REQ003	¿Cómo eliminar los productos del inventario del programa?	El sistema debera eliminar productos lógicamente (marcar como inactivos) por ID con confirmación.	Mantener control sin pérdida de datos	Sra. Hilda Guajala	Eliminar productos lógicamente	Mateo Velecela	2h	2025-07-22	Alta	No iniciado	Caja negra: Estado cambia a inactivo Caja blanca: Comprobar que el sistema lo borre del documento txt		Eliminacion de productos
REQ004	¿Cómo veo la lista de los productos activos?	El sistema debera mostrar productos activos en tabla legible, con alertas si el stock está en cero o bajo mínimo.	Visualizar estado de inventario	Sra. Hilda Guajala	Crear tabla con columnas e incluir alertas por stock bajo o cero	Mateo Velecela	2h	2025-07-21	Alta	No iniciado	Caja negra: Visualización con alertas Caja blanca: Verificar que el programa recorra bien todos los productos guardados.		Listado de inventario.
REQ005	¿Cómo busco productos?	El sistema debe permitir una búsqueda de productos por ID (rellenado a 10 dígitos) o por nombre.	Búsqueda flexible por ID o nombre	Sra. Hilda Guajala	Permitir búsquedas con ID corregido o por nombre.	Mateo Velecela	2h	2025-07-21	Alta	No iniciado	Caja negra: Resultados correctos Caja blanca: Revisar que el sistema compare bien los nombres al buscar.		Buscador de productos.
REQ006	¿Cómo alertar sobre stock?	El programa debera mostrar alertas: "SIN EXISTENCIA" (stock = 0), "STOCK BAJO" (stock < mínimo).	Evitar quiebres de inventario	Sra. Hilda Guajala	Marcar "SIN EXISTENCIA" y "STOCK BAJO" según stock actual.	Alvear Alexnader	2h	2025-07-21	Media	No iniciado	Caja negra: Alertas visibles. Caja blanca: Confirmar que se compruebe el mínimo permitido antes de mostrar la alerta.		Reporte de inventario.
REQ007	¿Cómo validar entrada de datos?	El sistema debera validar y guardar productos: nombre en mayúsculas, precio > 0, stock válido, ID de proveedor corregido a 10 dígitos..	Garantizar integridad del sistema.	Sra. Hilda Guajala	Verificar que los valores ingresados sean correctos.	Alvear Alexander	2h	2025-07-21	Media	No iniciado	Caja negra: No se aceptan datos inválidos Caja blanca: Ver que el sistema verifique esos datos.		Validacion de datos
REQ008	¿Cómo editar productos?	El sistema debera editar datos de productos con validación y corrección del ID de proveedor.	Mantener información actualizada	Sra. Hilda Guajala	Validar entrada, corregir formato y actualizar datos.	alvear alexander	2h	2025-07-21	Media	No iniciado	Caja negra: Cambios reflejados correctamente Caja blanca: Revisar que el sistema actualize correctamente el producto		Edicion de productos
		El sistema debera agregar proveedores con ID de 10 dígitos.	Registrar información de		Solicitar datos y validar ID, nombre						Caja negra: Proveedor agregado correctamente Caja blanca: Revisar que		

# CAJA BLANCA

La caja blanca es una técnica de prueba de software que analiza la estructura interna del código para verificar su funcionamiento y detectar errores, a diferencia de la caja negra, que evalúa solo la funcionalidad sin acceder al código.



## Ejemplo

**Prueba caja blanca :** El sistema deberá iniciar con un menú principal

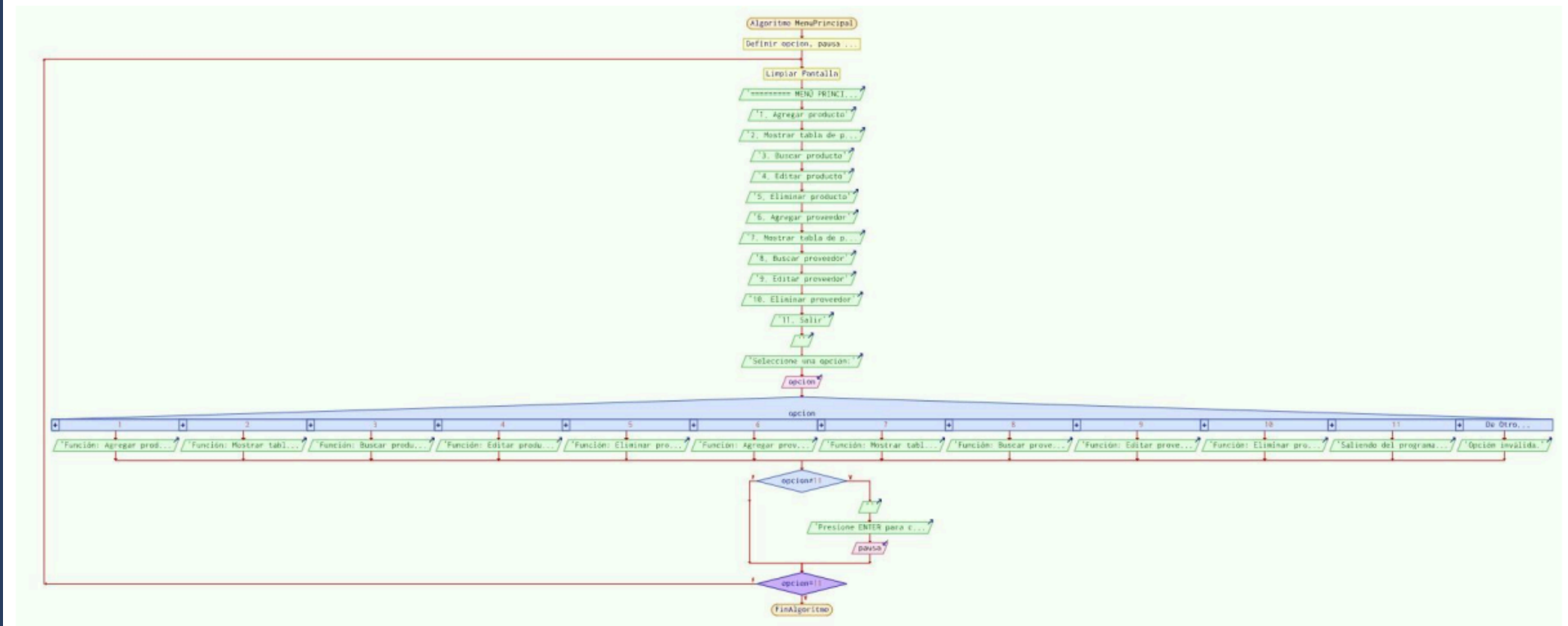
## 1. CÓDIGO FUENTE

```

669     do {
670         system(CLEAR);
671         imprimirMenu();
672         if (scanf("%d", &opcion) != 1) {
673             printf("Entrada invalida.\n");
674             limpiarBuffer();
675             opcion = 0;
676             continue;
677         }
678         limpiarBuffer();
679
680         switch (opcion) {
681             case 1: agregarProducto(); break;
682             case 2: mostrarTablaProductos(); break;
683             case 3: buscarProducto(); break;
684             case 4: editarProducto(); break;
685             case 5: eliminarProducto(); break;
686             case 6: agregarProveedor(); break;
687             case 7: mostrarTablaProveedores(); break;
688             case 8: buscarProveedor(); break;
689             case 9: editarProveedor(); break;
690             case 10: eliminarProveedor(); break;
691             case 11: printf("Saliendo...\n"); break;
692             default: printf("Opcion invalida.\n"); pausa(); break;
693         }
694     } while (opcion != 11);
695
696     return 0;
697 }

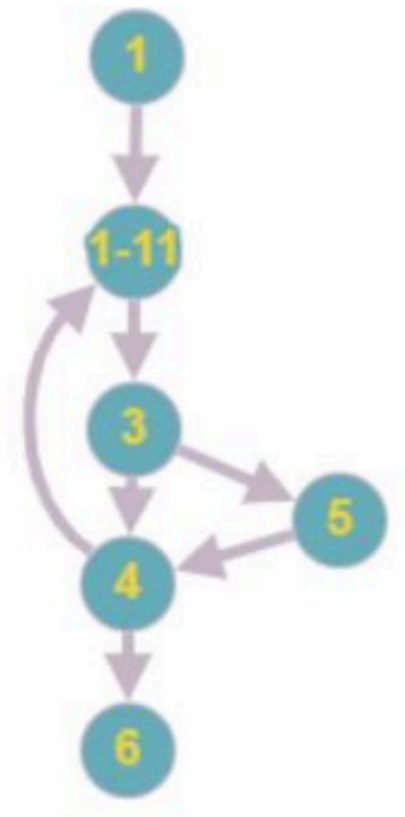
```

## 2. DIAGRAMA DE FLUJO (DF) PSEINT



# Ejemplo

## 3. GRAFO DE FLUJO (GF)



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

### RUTAS

R1: 1-2-3-5-6  
R2: 1-2-3-4-5-2  
R3: 1-2-3-4-5-6

## 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$   
 $V(G) = 2 + 1 = 3$
- $V(G) = A - N + 2$   
 $V(G) = 7 - 6 + 2 = 3$

DONDE:

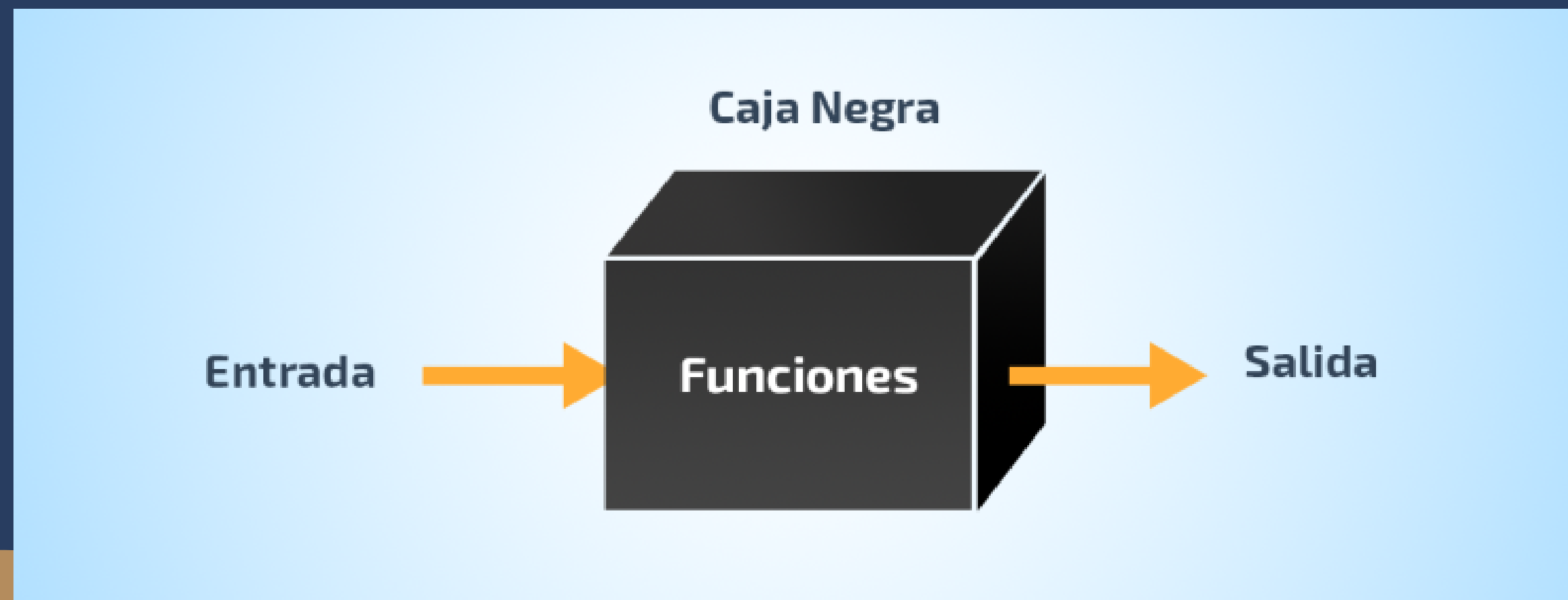
**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

# CAJA NEGRA

La caja negra es un enfoque donde solo se analizan las entradas y salidas de un sistema, sin conocer su funcionamiento interno. En pruebas de software, se usa para verificar que el programa cumpla con sus especificaciones sin revisar el código fuente.





# Ejemplo

TABLA 2: Agregar un producto

**Requisito Funcional RF2:** El sistema deberá agregar productos con ID de 10 dígitos (autogenerado o relleno con ceros).

VARIABLE	CLASE DE EQUIVALENCIA	ESTADO	REPRESENTANTE
Id	EC1: Id=00000000123	Válido	Id=00000000123
	EC2: Id=123	Válido	Id=00000000123
	EC3: Id=1234567890123	No válido	Id=/1234567890123
	EC4: Id=abc123	No válido	Id=/abc123

```
===== SISTEMA DE INVENTARIO Y PROVEEDORES =====
---- PRODUCTOS ----
1. Agregar producto
2. Mostrar productos
3. Buscar producto
4. Editar producto
5. Eliminar producto
-----
---- PROVEEDORES ----
6. Agregar proveedor
7. Mostrar proveedores
8. Buscar proveedor
9. Editar proveedor
10. Eliminar proveedor
-----
11. Salir
Seleccione una opcion: 1
ID generado para producto: 0000000001
Ingrese nombre del producto: Pepinillo
Ingrese precio (> 0): 0.30
Ingrese stock (0-1000): 100
Ingrese stock minimo (0-100): 10
Ingrese ID proveedor (solo numeros, se rellenara a 10 digitos): 1
Producto agregado exitosamente.

Presione ENTER para continuar...
```

```
===== SISTEMA DE INVENTARIO Y PROVEEDORES =====
---- PRODUCTOS ----
1. Agregar producto
2. Mostrar productos
3. Buscar producto
4. Editar producto
5. Eliminar producto
-----
---- PROVEEDORES ----
6. Agregar proveedor
7. Mostrar proveedores
8. Buscar proveedor
9. Editar proveedor
10. Eliminar proveedor
-----
11. Salir
Seleccione una opcion: 3
Ingrese ID del producto a buscar (numeros, se rellenara a 10 digitos): abc
Producto no encontrado.

Presione ENTER para continuar...
```

# CONCLUSION

La implementación de la aplicación en C cumple con el objetivo de facilitar la gestión de productos y proveedores en un negocio local, optimizando los procesos de registro, consulta y control de stock. Gracias al uso de estructuras de datos y la matriz de historia de usuario, se logró una herramienta funcional, clara y adaptada a las necesidades reales del negocio



MUCHAS  
GRACIAS

