

Prueba de Caja Blanca

“Sistema de Gestión de Inventario”

Integrantes:

Noe Cahuatijo

Bryan Cevallos

Leopoldo Bolaños

Jelen Lucio

Fecha: 2025/12/03

CONTROL DE VERSIONAMIENTO DE PRUEBAS CB

Version	Fecha	Responsable	Aprobado por
PCB_V1.0.0.docx			

Prueba caja blanca

RF N1: Inicio de sesión seguro

CÓDIGO FUENTE

```
def validar_login(username, password):

    conn = conectar_db()

    if not conn: return False, "Sin conexión a BD" # Nodo 1

    try:

        cursor = conn.cursor()

        sql = "SELECT ... WHERE u.username = ? AND u.activo = 1"

        cursor.execute(sql, (username,))

        user = cursor.fetchone() # Nodo 2

        if not user: # Nodo 3 (Decisión)

            return False, "Usuario no encontrado." # Nodo 4

        # Nodo 5 (Decisión)

        if hashlib.sha256(password.encode()).hexdigest() == user['password_hash']:

            user_data = {...}

            return True, user_data # Nodo 6

        else:

            return False, "Contraseña incorrecta." # Nodo 7

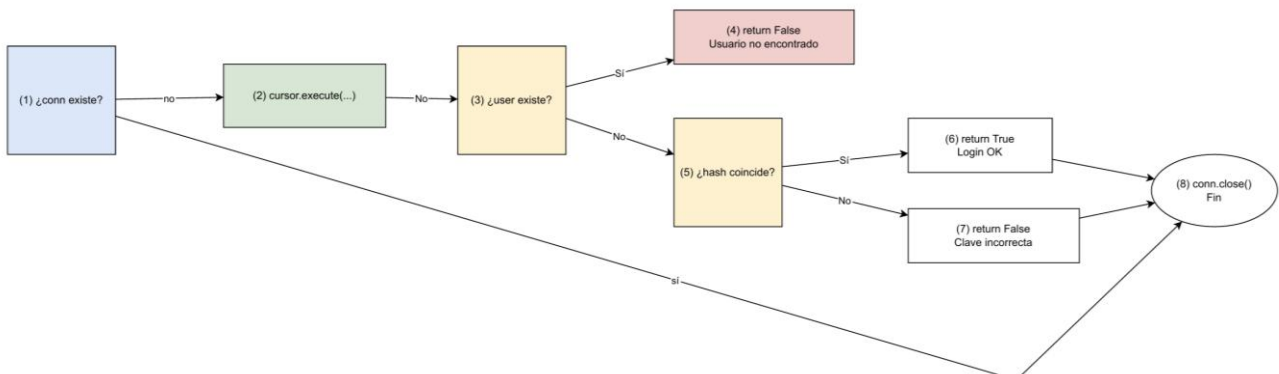
    finally:

        conn.close() # Nodo 8
```

DIAGRAMA DE FLUJO (DF)



GRAFO DE FLUJO (GF)



IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Rutas del Grafo de Flujo:

Ruta 1: 1-8
Ruta 2: 1-2-3-4-8
Ruta 3: 1-2-3-5-6-8
Ruta 4: 1-2-3-5-7-8

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

Nodos (N): Son todos los círculos numerados.

Nodos predicados (P): Son los nodos de decisión, que tienen más de una salida: -

A (aristas): Contando todas las flechas entre nodos.

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = P + 1$
 $= 3 + 1 = 4$
- $V(G) = A - N + 2$
 $= 10 - 8 + 2$
 $= 4$

DONDE

P: 3

A: 10

N: 8

RF N2: Ingreso y validación de productos

CÓDIGO FUENTE

```
def crear_producto(sku, nombre, precio, categoria_id):  
  
    # Nodo 1: Validar campos de texto vacíos  
  
    if not sku.strip() or not nombre.strip():  
  
        return "Datos incompletos." # Nodo 2  
  
    # Nodo 3: Validar tipo de dato numérico  
  
    try:  
  
        precio_float = float(precio)  
  
    except:
```

```
        return "Precio inválido." # Nodo 4

# Nodo 5: Validar lógica de negocio (precio positivo)
if precio_float < 0:
    return "Precio negativo." # Nodo 6

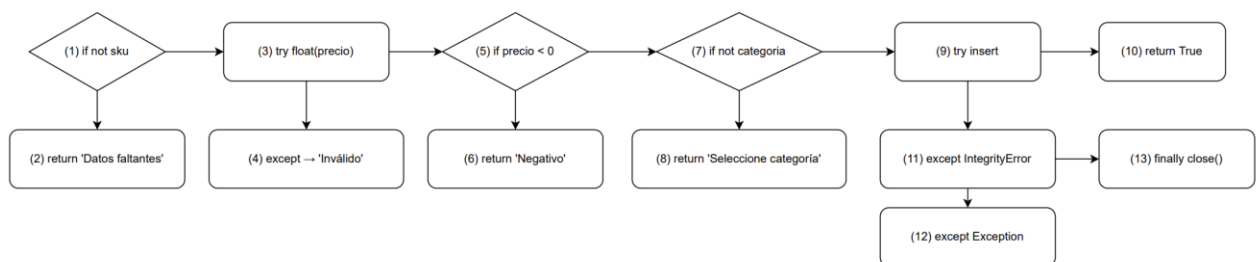
# Nodo 7: Validar selección de categoría
if not categoria_id:
    return "Selecione categoría." # Nodo 8

# Nodo 9: Intento de inserción en BD
try:
    conn.execute("INSERT INTO Producto ... VALUES (?, ?, ?, ?)", (sku, ...))
    conn.commit()
    return True # Nodo 10 (Éxito)
except sqlite3.IntegrityError:
    return "SKU duplicado." # Nodo 11 (Error unicidad)
except Exception as e:
    return f"Error: {e}" # Nodo 12 (Error genérico)
finally:
    conn.close() # Nodo 13 (Fin común)
```

DIAGRAMA DE FLUJO (DF)



GRAFO DE FLUJO (GF)



IDENTIFICACIÓN DE LAS RUTAS (Camino básico)

Rutas del Grafo de Flujo:

Ruta 1: 1 - 2 - 13

Ruta 2: 1 - 3 - 4 - 13

Ruta 3: 1 - 3 - 5 - 6 - 13

Ruta 4: 1 - 3 - 5 - 7 - 8 - 13

Ruta 5: 1 - 3 - 5 - 7 - 9 - 10 - 13

Ruta 6: 1 - 3 - 5 - 7 - 9 - 11 - 13

COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

Nodos (N): Son todos los círculos numerados.

Nodos predicados (P): Son los nodos de decisión, que tienen más de una salida: -

A (aristas): Contando todas las flechas entre nodos.

- $V(G) = \text{número de nodos predicados(decisiones)} + 1$
 $V(G) = P + 1$
 $= 5 + 1 = 6$
- $V(G) = A - N + 2$
 $= 17 - 13 + 2$
 $= 6$

DONDE

P: 5

A: 17

N: 13