

# Prueba de Caja Blanca

---

*“Sistema de inventario del Universo del fomix”*

**Integrantes:**  
**Alfonso Arroyo**  
**David Pilaguano**  
**Stalin Uvidia**

**Fecha 2024/08/15**

## Prueba caja blanca del requisito "añadir productos"

### 1. CÓDIGO FUENTE

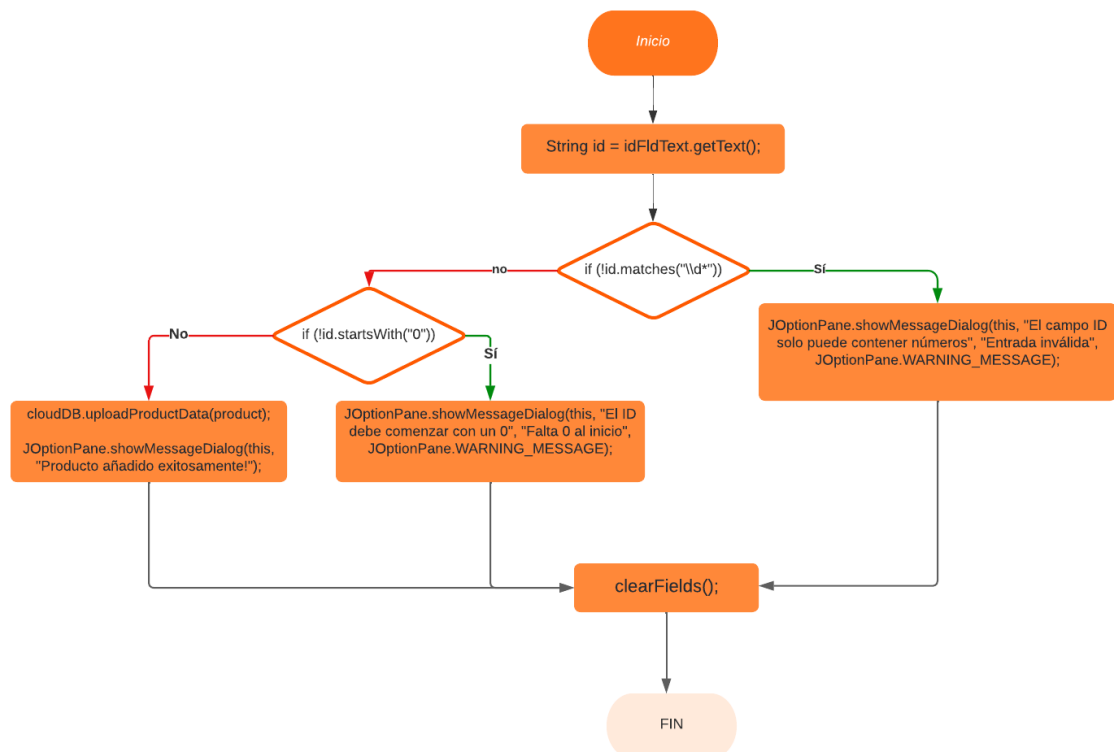
Pegar el trozo de código fuente que se requiere para el caso de prueba

```
String id = idFldText.getText();
if (!id.matches("\\d*")) {
    JOptionPane.showMessageDialog(this, "El campo ID solo puede contener números", "Entrada inválida", JOptionPane.WARNING_MESSAGE);
    clearFields();
} else if (!id.startsWith("0")) {
    JOptionPane.showMessageDialog(this, "El ID debe comenzar con un 0", "Falta 0 al inicio", JOptionPane.WARNING_MESSAGE);
    clearFields();
}
```

```
private void idFldTextKeyReleased(java.awt.event.KeyEvent evt) {
    String id = idFldText.getText();
    if (!id.matches("\\d*")) {
        JOptionPane.showMessageDialog(this, "El campo ID solo puede contener números", "Entrada inválida", JOptionPane.WARNING_MESSAGE);
        clearFields();
    } else if (!id.startsWith("0")) {
        JOptionPane.showMessageDialog(this, "El ID debe comenzar con un 0", "Falta 0 al inicio", JOptionPane.WARNING_MESSAGE);
        clearFields();
    }
}
```

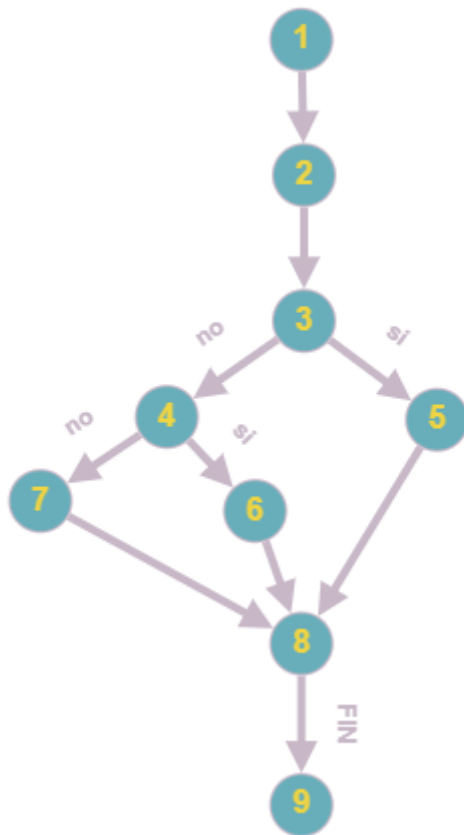
### 2. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



### 3. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



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

Determinar en base al GF del numeral 4

#### RUTAS

**R1: 1 → 2 → 3 → 4 → 5 → 9 (El ID contiene caracteres no numéricos)**

**R2: 1 → 2 → 3 → 6 → 7 → 8 → 9 (El ID solo contiene números, pero no empieza en 0)**

**R3: 1 → 2 → 3 → 6 → 9 (El ID es válido: contiene solo números y empieza con 0)**

### 5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

**V(G) = Número de regiones en el grafo de flujo**

- Número de regiones (incluyendo la región exterior): **3**.

**V(G) = Número de nodos predicados (decisiones) + 1**

- Nodos predicados: 2 (Nodos 3 y 6)
- $V(G) = 2 + 1 = 3$

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

- A (Número de aristas): 10
- N (Número de nodos): 9
- $V(G) = 10 - 9 + 2 = 3$

DONDE:

**P:** Número de nodos predicado 2

**A:** Número de aristas 10

**N:** Número de nodos 9

## Prueba caja blanca del requisito "Editar Productos"

### 6. CÓDIGO FUENTE

Pegar el trozo de código fuente que se requiere para el caso de prueba

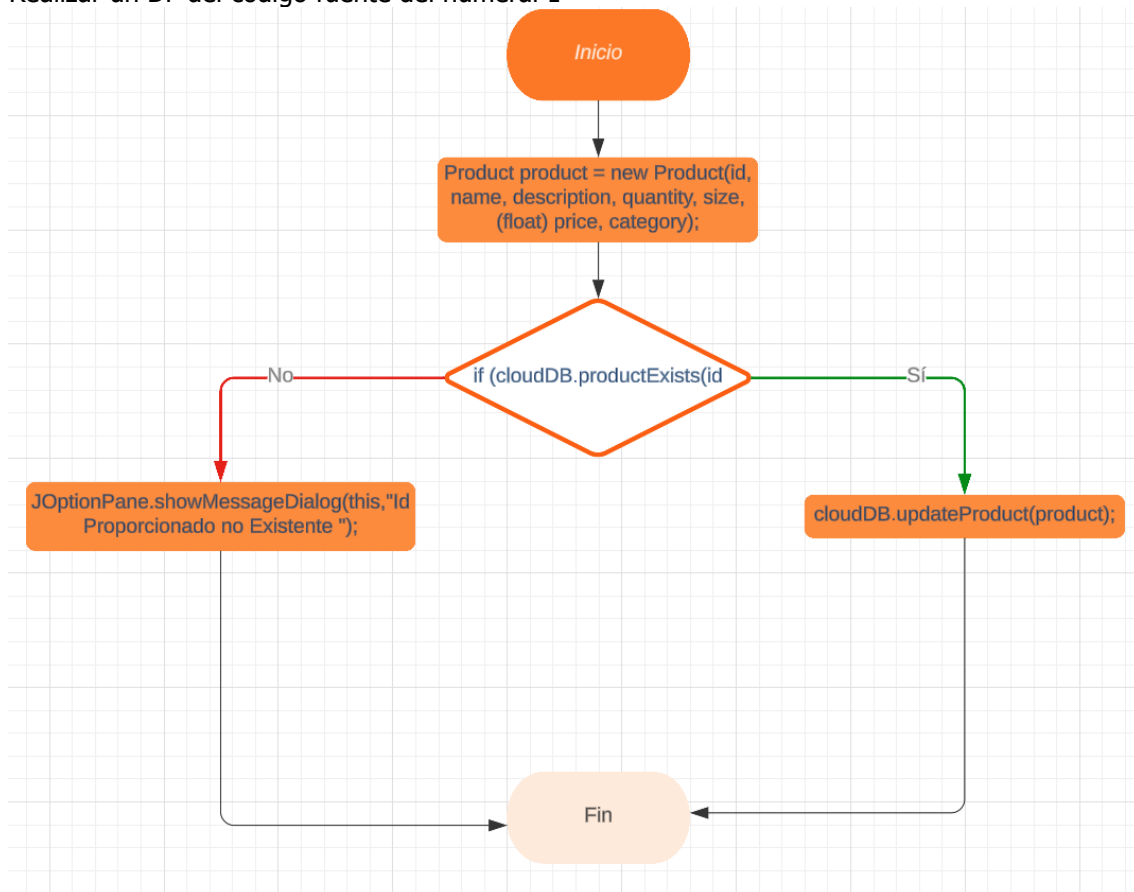
```
Product product = new Product(id, name, description, quantity, size, (float) price, category);
```

```
if (cloudDB.productExists(id)) {  
    cloudDB.updateProduct(product);  
} else {  
    JOptionPane.showMessageDialog(this,"Id Proporcionado no Existente ");  
}
```

```
Product product = new Product(id, name, description, quantity, size, (float) price, category);  
  
if (cloudDB.productExists(id)) {  
    cloudDB.updateProduct(product);  
} else {  
    JOptionPane.showMessageDialog(this,"Id Proporcionado no Existente ");  
}
```

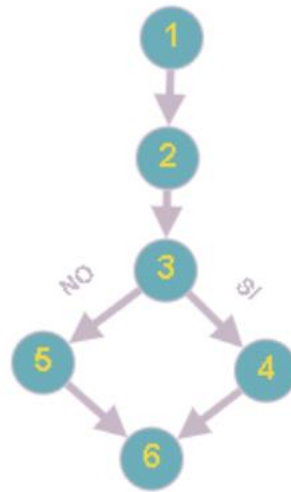
### 7. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



## 8. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



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

Determinar en base al GF del numeral 4

### RUTAS

**R1:**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$  (Si el id del producto existe, modificar en la base de datos)

**R2:**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$  (Si el id no existe, mostrar mensaje)

## 10. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

### **$V(G)$ = Número de regiones en el grafo de flujo**

- La cantidad de regiones (incluyendo la región exterior) en el grafo de flujo es 3.

### **$V(G)$ = Número de nodos predicados (decisiones) + 1**

- Nodos predicados: 1 (Nodo 3)
- $V(G) = 1 + 1 = 2$

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

- A (Número de aristas) = 6
- N (Número de nodos) = 6
- $V(G) = 8 - 6 + 2 = 2$

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

## Prueba caja blanca del requisito "Ingresar al sistema | Login"

### 11. CÓDIGO FUENTE

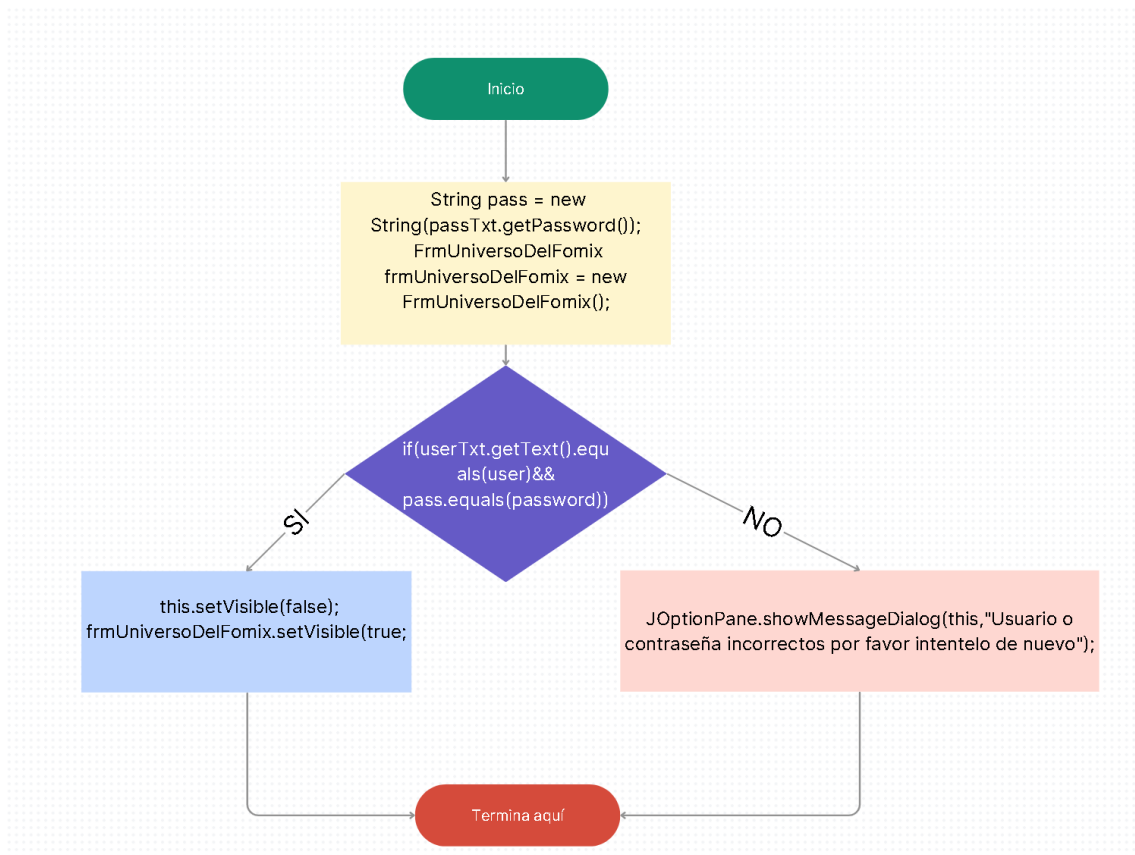
Pegar el trozo de código fuente que se requiere para el caso de prueba

```
String user = "admin";
String password = "admin";
String pass = new String(passTxt.getPassword());
FrmUniversoDelFomix frmUniversoDelFomix = new FrmUniversoDelFomix();
if(userTxt.getText().equals(user)&& pass.equals(password)){
    this.setVisible(false);
    frmUniversoDelFomix.setVisible(true);
}else{
    JOptionPane.showMessageDialog(this,"Usuario o contraseña incorrectos por favor intentelo de nuevo");
}
```

```
String user = "admin";
String password = "admin";
String pass = new String(passTxt.getPassword());
FrmUniversoDelFomix frmUniversoDelFomix = new FrmUniversoDelFomix();
if(userTxt.getText().equals(user)&& pass.equals(password)) {
    this.setVisible(false);
    frmUniversoDelFomix.setVisible(true);
}else{
    JOptionPane.showMessageDialog(this,"Usuario o contraseña incorrectos por favor intentelo de nuevo");
}
}
```

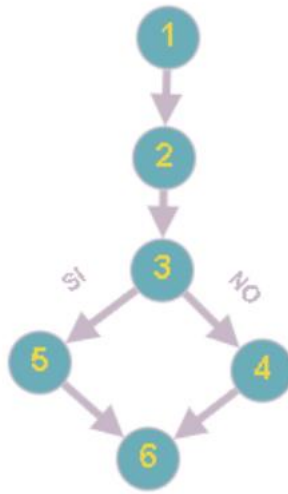
### 12. DIAGRAMA DE FLUJO (DF)

Realizar un DF del código fuente del numeral 1



### 13. GRAFO DE FLUJO (GF)

Realizar un GF en base al DF del numeral 2



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

Determinar en base al GF del numeral 4

#### RUTAS

**R1:**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$  (Contraseña y Usuario concuerdan con los registrados)

**R2:**  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6$  (Contraseña y Usuario NO concuerdan con los registrados)

### 15. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

**$V(G)$  = Número de regiones en el grafo de flujo**

- La cantidad de regiones (incluyendo la región exterior) en el grafo de flujo es 3.

**$V(G)$  = Número de nodos predicados (decisiones) + 1**

- Nodos predicados: 1 (Nodo 3)
- $V(G) = 1 + 1 = 2$

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

- A (Número de aristas) = 6
- N (Número de nodos) = 6
- $V(G) = 8 - 6 + 2 = 2$

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos