

Prueba de Caja Blanca

“Título de proyecto”

Integrantes: Diego Casignia

Javier Ramos

Anthony Villarreal

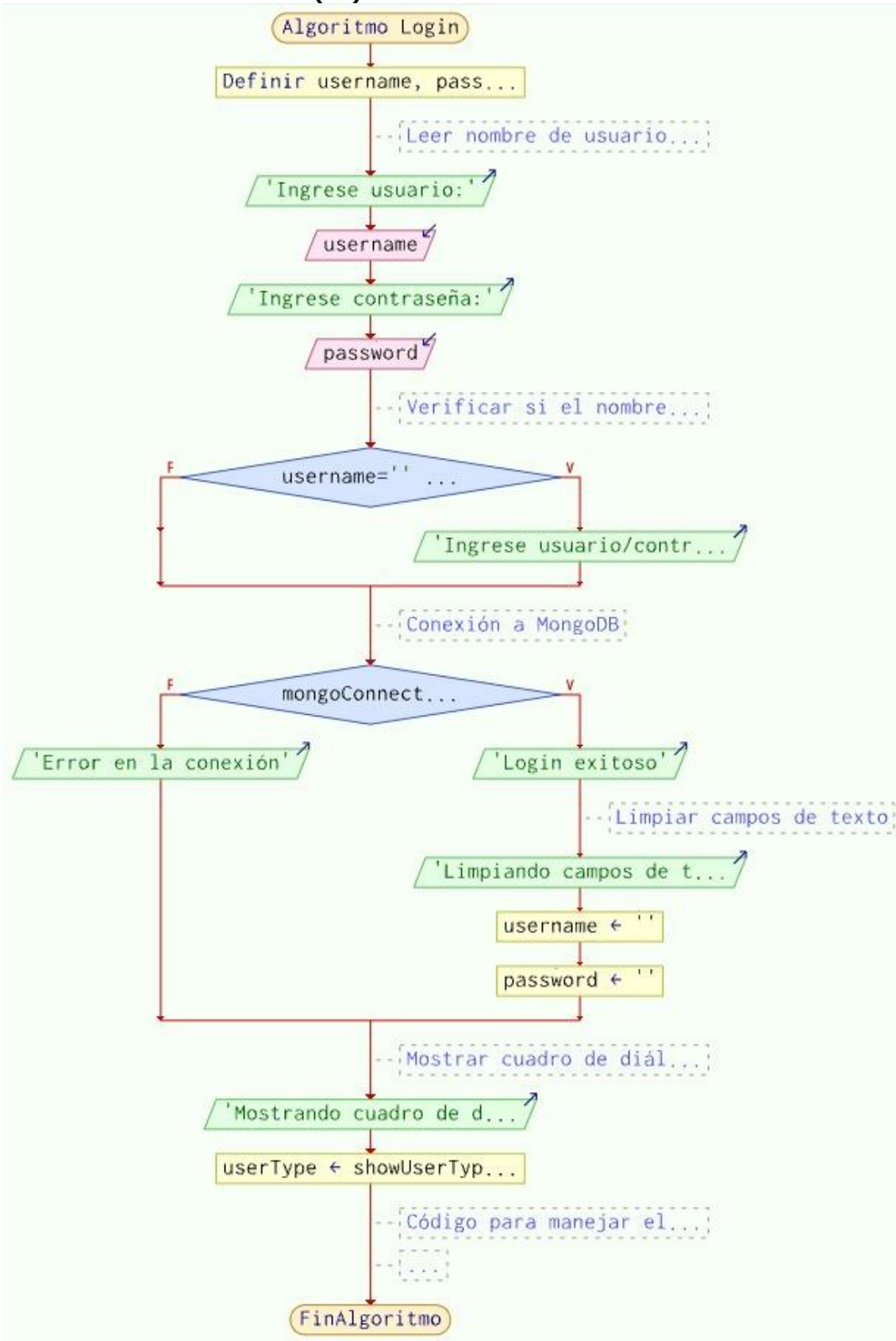
Fecha 2025-07-16

Prueba caja blanca de Login

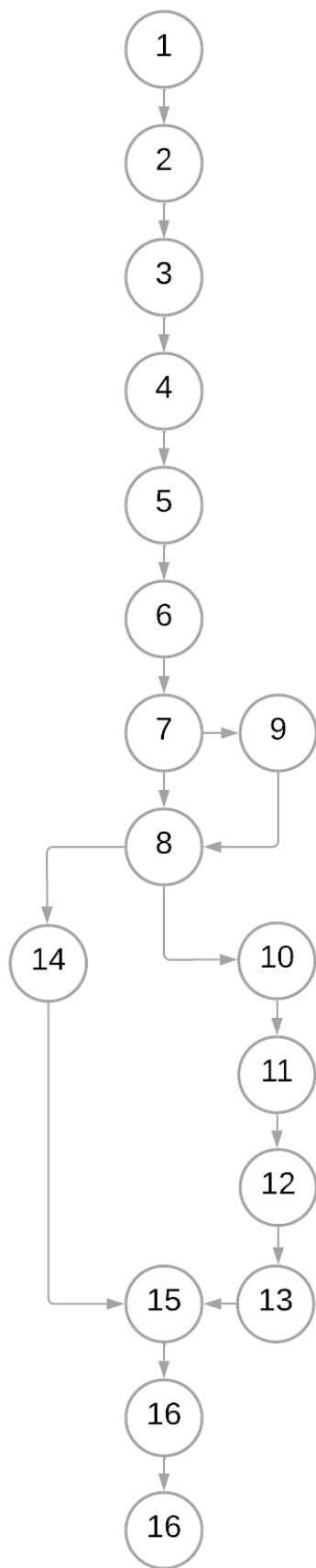
1. CÓDIGO FUENTE

```
1 package com.panaderia.ruminahui.config;
2
3 > import ...
4
13
14 public class MongoDBConfig { 5 usages
15     private static final Logger LOGGER = Logger.getLogger(MongoDBConfig.class.getName()); 6 usages
16     private static MongoClient mongoClient; 4 usages
17     private static MongoDB database; 5 usages
18
19     public static MongoDB getDatabase() { 2 usages
20         if (database == null) {
21             try {
22                 // Configure MongoDB connection
23                 MongoClientSettings settings = MongoClientSettings.builder()
24                     .applyToConnectionPoolSettings( Builder builder -> builder.maxSize(20))
25                     .build();
26                 mongoClient = MongoClient.create(settings);
27                 database = mongoClient.getDatabase("panaderia_ruminahui");
28
29                 // Test connection
30                 database.listCollectionNames().first(); // Forces a connection test
31                 LOGGER.info(msg: "Successfully connected to MongoDB database: panaderia_ruminahui");
32
33                 // Initialize admin user
34                 initializeAdminUser();
35             } catch (MongoException e) {
36                 LOGGER.severe(msg: "Failed to connect to MongoDB: " + e.getMessage());
37                 throw new RuntimeException("Failed to connect to MongoDB: " + e.getMessage(), e);
38             }
39         }
40         return database;
41     }
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



4. IDENTIFICACIÓN DE LAS RUTAS (Camino basico)

Nodos (N): 17

Aristas (E): 18

Nodo Predicado/Condición /Decisión(P): 2

RUTAS

R1: 1,2,3,4,5,6,7,9,17

R2: 1,2,3,4,5,6,7,8,10,11,12,16,17

R2: 1,2,3,4,5,6,7,8,10,13,14,15,17

5. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{DECISIONES} + 1$
 $V(G) = 2 + 1$
 $V(G) = 3$

- $V(G) = A - N + 2$
 $V(G) = 18 - 17 + 2$
 $V(G) = 3$

DONDE:

P: Número de nodos predicado

A: Número de aristas

N: Número de nodos

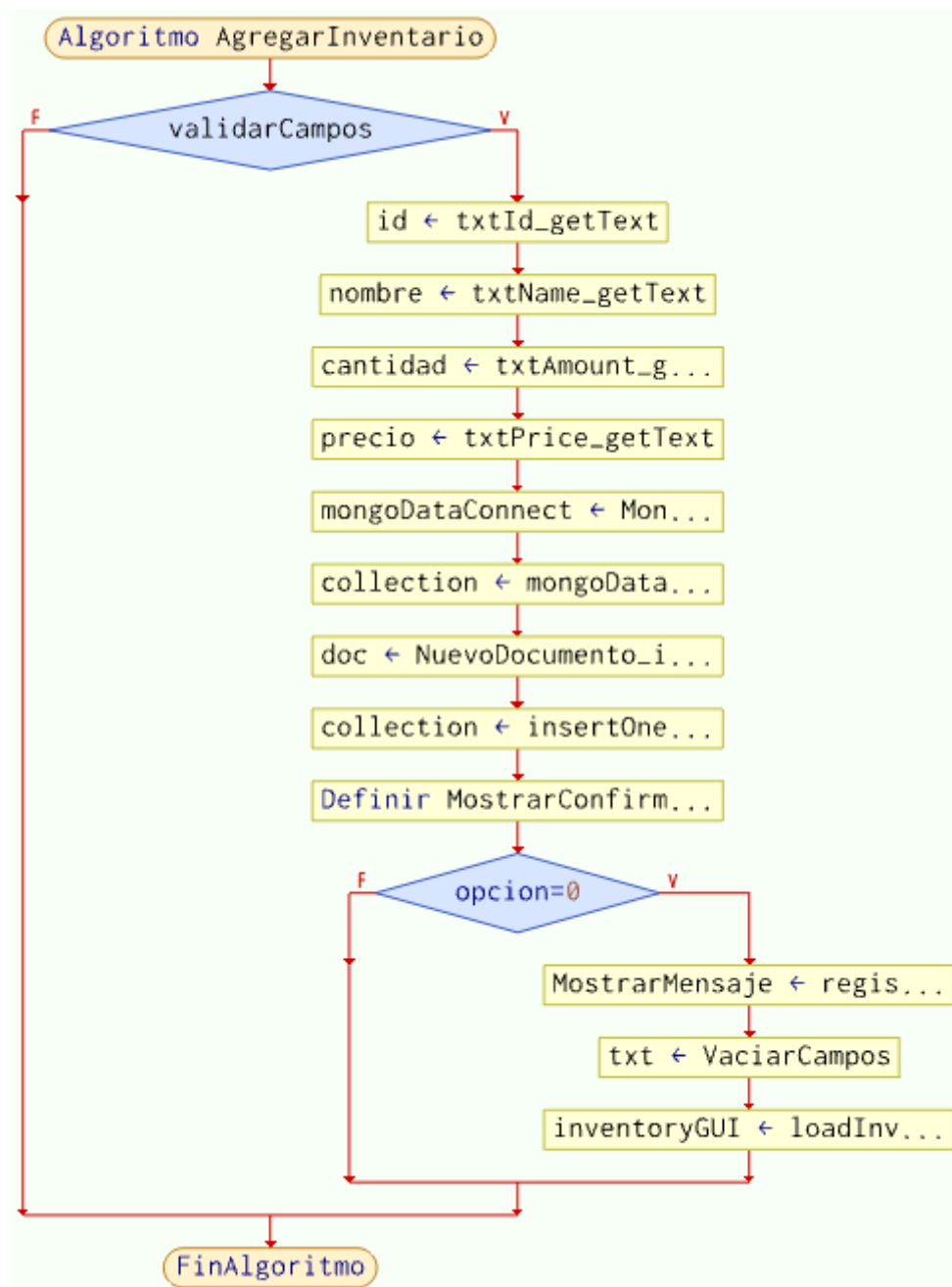
Prueba caja blanca de Gestión de secciones

1. CÓDIGO FUENTE

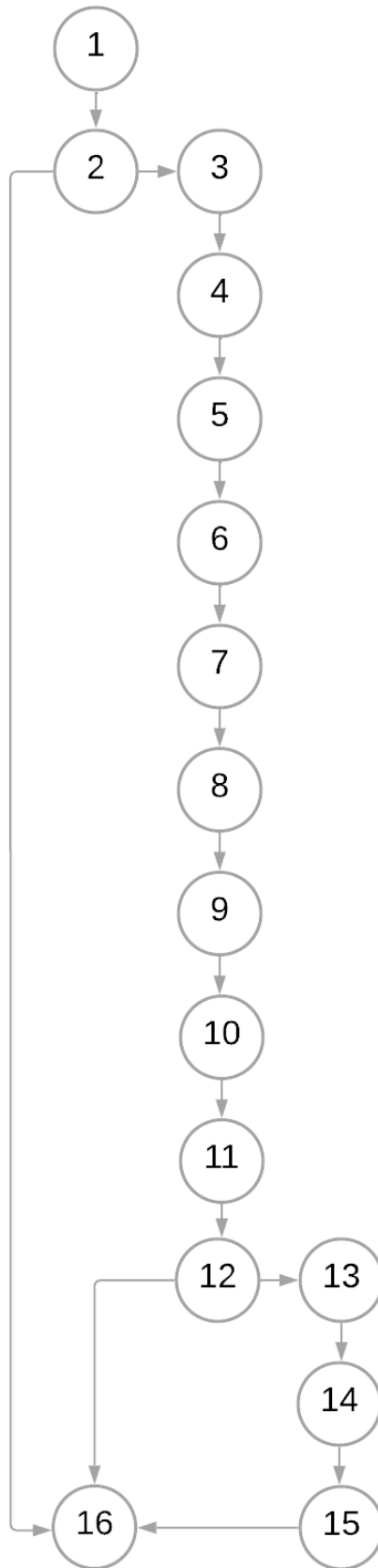
```
private static void initializeAdminUser() { 1 usage
    MongoClient database = MongoClientFactory.getClient();
    MongoCollection<Document> collection = database.getCollection("users");
    // Check if admin user exists
    Document admin = collection.find(Filters.eq("username", "admin")).first();
    if (admin == null) {
        // Create admin user
        Document adminDoc = new Document()
            .append("_id", "admin-" + java.util.UUID.randomUUID().toString())
            .append("username", "admin")
            .append("password", "admin123"); // Plain text for simplicity; consider hashing in production
        collection.insertOne(adminDoc);
        LOGGER.info(msg: "Admin user created: username=admin, password=admin123");
    } else {
        LOGGER.info(msg: "Admin user already exists: username=admin");
    }
}

public static void close() { no usages
    if (mongoClient != null) {
        try {
            mongoClient.close();
            LOGGER.info(msg: "MongoDB connection closed successfully");
        } catch (Exception e) {
            LOGGER.severe(msg: "Error closing MongoDB connection: " + e.getMessage());
        }
    }
}
```

2. DIAGRAMA DE FLUJO (DF)



3. GRAFO DE FLUJO (GF)



6. IDENTIFIACCIÓN DE LAS RUTAS (Camino basico)

Nodos (N): 16

Aristas (E): 17

Nodo Predicado/Condición /Decisión(P): 2

RUTAS

R1: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

R2: 1,2,3,4,5,6,7,8,9,10,11,12,16

R2: 1,2,16

7. COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = \text{DECISIONES} + 1$
 $V(G) = 2 + 1$
 $V(G) = 3$

- $V(G) = A - N + 2$
 $V(G) = 17 - 16 + 2$
 $V(G) = 3$

DONDE:

P: Número de nodos predado

A: Número de aristas

N: Número de nodos