

Facultad de matemáticas

Reporte de casos de pruebas para el Módulo Analizador de Clases

Versión 1.0

Autores:

Josué Israel Canul Ordoñez

Mónica Garcilazo Cuevas

José Carlos Leo Fernández

Endrick Alfredo Pool Flores

Samuel David Rodríguez Coral

**CONTROL DE DOCUMENTACIÓN**

**Control de Configuración**

| **Título:** | Reporte de casos de pruebas para el Módulo de gestor de archivos |
| --- | --- |
| **Referencia:** |  |
| **Autor(es):** | Samuel David Rodriguez Coral |
| **Fecha:** | 03/04/2025 |

**Histórico de versiones**

| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| --- | --- | --- | --- | --- |
| 1.0 | 03/04/25 | A | Samuel David Rodriguez Coral | REP\_CasosPruebasAnalizadorDeClases\_v2.0\_2025-04-03 |

**Estado:** (B)orrador, (R)evisión, (A)probado

**Histórico de cambios**

| **Versión** | **Fecha** | **Cambios** |
| --- | --- | --- |
| 1.0 | 03/04/2025 | Primera versión. |

**Índice**

[**1. Introducción 4**](#_1pbytlxhdrny)

[**2. Casos de pruebas unitarias del módulo de gestor de archivos 4**](#_gjstr9le7sdw)

[2.1. Leer Archivo Existente 4](#_4u8hd5vtnasv)

[2.2. Leer Archivo Inexistente 6](#_o7v5mu9u6acb)

# **Introducción**

Este documento detalla los casos de prueba unitarios para la verificación y validación del módulo ClassAnalyzer, el cual recopila la información de una clase dentro de un archivo .java

# **Casos de pruebas unitarias del módulo de gestor de archivos**

## **2.1. Leer Archivo Inexistente**

*Tabla 3 - Caso de prueba 2.2*

| **Caso de prueba** | Leer Archivo Inexistente | | |
| --- | --- | --- | --- |
| **Identificador caso de prueba** | CP01\_Analizar datos de clase | | |
| **Función probar** | ClassAnalyzer.analyze | | |
| **Ubicación de la prueba** | src\test\java\com\example\ClassAnalyzerTest.java | | |
| **Método donde se realiza la prueba** | testAnalyze\_SingleClass | | |
| **Objetivo** | Verificar que el método analiza correctamente una clase con métodos y obtiene información precisa sobre el nombre de la clase, el número de métodos y el total de líneas físicas. | | |
| **Descripción** | Se creará un archivo de prueba en Java que contenga una clase con dos métodos y una variable. El sistema debe analizar el archivo y devolver la información correcta de la clase, incluyendo el nombre, número de métodos y cantidad de líneas físicas. | | |
| **Criterios de éxito** | * Se detecta correctamente el nombre de la clase. * Se identifican exactamente 2 métodos en la clase. * Se cuenta el número correcto de líneas físicas (5). | | |
| **Criterios de falla** | * El análisis no detecta la clase en el archivo. * El número de métodos identificado es incorrecto. * El número de líneas físicas no coincide con la cantidad esperada. | | |
| **Precondiciones** | Debe existir un archivo válido en el directorio de prueba que contenga una clase en Java con métodos definidos. | | |
| **Necesidades para el caso de prueba** | public class TestClass {  private int x;  public void myMethod() { }  public int anotherMethod() { return 0; }  } | | |
| **Autor** | Samuel David Rodriguez Coral | | |
| **Fecha de creación** | 01/04/2025 | | |
| **Flujo del caso de prueba** | **No paso** | **Usuario del sistema** | **Sistema** |
| **1** | **Crea un archivo de prueba en el directorio temporal con una clase en Java.** |  |
| **2** | **Ejecuta el método analyze() de la clase ClassAnalyzer, pasando el archivo creado.** |  |
| **3** |  | **Verifica que el archivo existe y procede a analizar su contenido.** |
| **6** |  | **Extrae información de la clase, incluyendo su nombre, cantidad de métodos y líneas físicas.** |
|  | **7** |  | **Retorna una lista con la información procesada de la clase.** |
|  | **8** |  | **Compara los datos obtenidos con los valores esperados.** |
| **Post condiciones** | Se obtiene la información correcta de la clase sin errores. | | |

*Tabla 4 - Resultado del caso de prueba 2.1*

| **Número de ejecución** | 1 | |
| --- | --- | --- |
| **Estatus (Pasó / No Pasó)** | Pasó | |
| **Responsable de la ejecución** | Samuel David Rodriguez Coral | |
| **Fecha de ejecución** | 03/04/2025 | |
| **Salida** | La prueba no arroja información con la salida fuera del resultado de la prueba | |
| **Faltas detectadas** | | |
| **Descripción falta** | | **Severidad (Alta, Media, Baja)** |
| N/A | | |

## 

## **2.1. Leer Archivo Inexistente**

*Tabla 3 - Caso de prueba 2.2*

| **Caso de prueba** | Leer Archivo Inexistente | | |
| --- | --- | --- | --- |
| **Identificador caso de prueba** | CP01\_Analizar datos de clase | | |
| **Función probar** | ClassAnalyzer.analyze | | |
| **Ubicación de la prueba** | src\test\java\com\example\ClassAnalyzerTest.java | | |
| **Método donde se realiza la prueba** | testAnalyze\_SingleClass | | |
| **Objetivo** | Verificar que el método analiza correctamente una clase con métodos y obtiene información precisa sobre el nombre de la clase, el número de métodos y el total de líneas físicas. | | |
| **Descripción** | Se creará un archivo de prueba en Java que contenga una clase con dos métodos y una variable. El sistema debe analizar el archivo y devolver la información correcta de la clase, incluyendo el nombre, número de métodos y cantidad de líneas físicas. | | |
| **Criterios de éxito** | * Se detecta correctamente el nombre de la clase. * Se identifican exactamente correctamente los métodos en la clase. * Se cuenta el número correcto de líneas físicas (5). | | |
| **Criterios de falla** | * El análisis no detecta la clase en el archivo. * El número de métodos identificado es incorrecto. * El número de líneas físicas no coincide con la cantidad esperada. | | |
| **Precondiciones** | Debe existir un archivo válido en el directorio de prueba que contenga una clase en Java con métodos definidos. | | |
| **Necesidades para el caso de prueba** | public class TestClass {  private int x;  public void myMethod() { }  public int anotherMethod() { return 0; }  } | | |
| **Autor** | Samuel David Rodriguez Coral | | |
| **Fecha de creación** | 01/04/2025 | | |
| **Flujo del caso de prueba** | **No paso** | **Usuario del sistema** | **Sistema** |
| **1** | **Crea un archivo de prueba en el directorio temporal con una clase en Java.** |  |
| **2** | **Ejecuta el método analyze() de la clase ClassAnalyzer, pasando el archivo creado.** |  |
| **3** |  | **Verifica que el archivo existe y procede a analizar su contenido.** |
| **6** |  | **Extrae información de la clase, incluyendo su nombre, cantidad de métodos y líneas físicas.** |
|  | **7** |  | **Retorna una lista con la información procesada de la clase.** |
|  | **8** |  | **Compara los datos obtenidos con los valores esperados.** |
| **Post condiciones** | Se obtiene la información correcta de la clase sin errores. | | |

*Tabla 4 - Resultado del caso de prueba 2.2*

| **Número de ejecución** | 1 | |
| --- | --- | --- |
| **Estatus (Pasó / No Pasó)** | Pasó | |
| **Responsable de la ejecución** | Samuel David Rodriguez Coral | |
| **Fecha de ejecución** | 03/04/2025 | |
| **Salida** | La prueba no arroja información con la salida fuera del resultado de la prueba | |
| **Faltas detectadas** | | |
| **Descripción falta** | | **Severidad (Alta, Media, Baja)** |
| N/A | | |

## 