

Facultad de matemáticas

Reporte de casos de pruebas para el Módulo de Gestor de Archivos

Versión 1.0

Autores:

Aaron Isaac Graniel Arzat

Fernando Joachín Prieto

David Peña Muñoz

Jose Luis Pooc Moo

Andrea Isabel Torres Perez

**CONTROL DE DOCUMENTACIÓN**

**Control de Configuración**

| **Título:** | Reporte de casos de pruebas para el Módulo de gestor de archivos |
| --- | --- |
| **Referencia:** |  |
| **Autor(es):** | David Peña Muñoz, Andrea Isabel Torres Perez |
| **Fecha:** | 05/03/2025 |

**Histórico de versiones**

| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| --- | --- | --- | --- | --- |
| 1.0 | 05/03/2025 | A | Andrea Isabel Torres Perez | REP\_CasosPruebasGestorArchivos\_v1.0\_2025-03-05.docx |

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

**Histórico de cambios**

| **Versión** | **Fecha** | **Cambios** |
| --- | --- | --- |
| 1.0 | 05/03/2025 | Primera versión. |

**Índice**

[**1. Introducción 4**](#_heading=h.1pbytlxhdrny)

[**2. Casos de pruebas unitarias del módulo de gestor de archivos 4**](#_heading=h.gjstr9le7sdw)

[2.1. Leer Archivo Existente 4](#_heading=h.4u8hd5vtnasv)

[2.2. Leer Archivo Inexistente 6](#_heading=h.o7v5mu9u6acb)

# **Introducción**

Este documento detalla los casos de prueba unitarios para la verificación y validación del módulo de gestor de archivos referente al sistema de conteo de líneas lógicas y físicas.

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

## 2.1. Leer Archivo Existente

*Tabla 1 - Caso de prueba 2.1*

| **Caso de prueba** | Leer Archivo Existente | | |
| --- | --- | --- | --- |
| **Identificador caso de prueba** | CP01\_LeerArchivoExistente | | |
| **Función probar** | FileManager.readLines | | |
| **Objetivo** | Verificar que el método lee correctamente un archivo existente y devuelve la lista de líneas esperadas. | | |
| **Descripción** | Se probará la lectura de un archivo que existe en el directorio y contiene varias líneas de texto. El sistema debe devolver una lista con las líneas del archivo sin modificaciones. | | |
| **Criterios de éxito** | El método retorna una lista con el contenido exacto del archivo.  No se alteran ni modifican las líneas leídas. | | |
| **Criterios de falla** | El método retorna una lista vacía o incompleta.  Se genera una excepción o error inesperado. | | |
| **Precondiciones** | Debe existir un archivo válido en el directorio especificado. | | |
| **Necesidades para el caso de prueba** | Archivo a leer:  public class TestFile{  public static void main(String[] args) {  System.out.println("Hello, world!");  }  }  Nombre del archivo para extraer las líneas  TestFile.java | | |
| **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 especificado con varias líneas de texto.** |  |
| **2** | **Ejecutar el método readLines() de la clase FileManager pasando el nombre del archivo creado.** |  |
| **3** |  | **Verifica que el archivo existe y procede a leer su contenido.** |
| **4** |  | **Retorna una lista con las líneas exactas del archivo.** |
| **5** | |  | | --- |   **Compara la lista retornada con el contenido esperado.** |  |
| **Post condiciones** | Se obtiene la lista esperada sin errores. | | |

*Tabla 2 - Resultado del caso de prueba 2.1*

| **Número de ejecución** | 1 | |
| --- | --- | --- |
| **Estatus (Pasó / No Pasó)** | Pasó | |
| **Responsable de la ejecución** | Fernando Joachín Prieto | |
| **Fecha de ejecución** | 03/04/2025 | |
| **Salida** | No se generó ninguna salida visible en la terminal durante la ejecución de la prueba. | |
| **Faltas detectadas** | | |
| **Descripción falta** | | **Severidad (Alta, Media, Baja)** |
| N/A | | |

### 

## 2.2. Leer Archivo Inexistente

*Tabla 3 - Caso de prueba 2.2*

| **Caso de prueba** | Leer Archivo Inexistente | | |
| --- | --- | --- | --- |
| **Identificador caso de prueba** | CP02\_LeerArchivoInexistente | | |
| **Función probar** | FileManager.readLines | | |
| **Criterios de éxito** | Se lanza una excepción FileException. | | |
| **Criterios de falla** | * El método no lanza excepción y devuelve un contenido incorrecto. * Se genera una excepción inesperada diferente a FileException. | | |
| **Precondiciones** | El archivo especificado no debe existir en el directorio. | | |
| **Necesidades para el caso de prueba** | Nombre del archivo inexistente para extraer las líneas  NonExistenFile.java | | |
| **Autor** | Samuel David Rodriguez Coral | | |
| **Fecha de creación** | 01/04/2025 | | |
| **Flujo del caso de prueba** | **No paso** | **Usuario del sistema** | **Sistema** |
| **1** | **Verifica que el archivo "archivoInexistente.txt" no existe.** |  |
| **2** | **Ejecutar el método readLines() de la clase FileManager pasando el nombre de "archivoInexistente.txt".** |  |
| **3** |  | **Detecta que el archivo no existe.** |
| **6** | **Confirma que la lista retornada está vacía.** |  |
| **Post condiciones** | Se devuelve una lista vacía y el sistema no lanza errores inesperados. | | |

*Tabla 4 - Resultado del caso de prueba 2.2*

| **Número de ejecución** | 1 | |
| --- | --- | --- |
| **Estatus (Pasó / No Pasó)** | Pasó | |
| **Responsable de la ejecución** | Fernando Joachín Prieto | |
| **Fecha de ejecución** | 05/03/2025 | |
| **Salida** | The file does not exist: NonExistentFile.java | |
| **Faltas detectadas** | | |
| **Descripción falta** | | **Severidad (Alta, Media, Baja)** |
| N/A | | |

## 