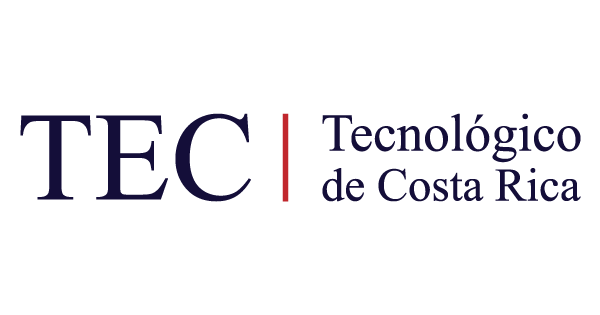
**

***Instituto Tecnológico de Costa Rica (ITCR)***

***Escuela de Computación***

***Curso:******Principios de Sistemas Operativos***

***Código del curso: IC-6600***

***Grupo 2***

***Profesor(a): Erika Marín Schumann***

***Estudiantes:***

***Marco Antonio Reveiz Rojas (2019053583)***

***David Solano Fuentes (2018167435)***

***David Castro Holguin (2018105813)***

***PROYECTO 3 File System***

# **Índice**

[**Índice**](#_8nh9iban96xm) **2**

[**Introducción**](#_prllxeqlp66g) **3**

[**Estrategias de Solución**](#_11z3v0raii3t) **3**

[Disco Virtual](#_lqlsn5qr2pzs) 3

[Jerarquía de Directorios y Archivos](#_216ag57mgcuv) 6

[Funciones del File System](#_g6qv8mlhdxm5) 7

[**Lecciones aprendidas**](#_xlsfpgn368r0) **9**

[**Casos de pruebas**](#_u5j1la20izfv) **10**

[Prueba crear Disco Virtual](#_up55ftv4exgu) 10

[Prueba crear Archivo](#_t50sxbq3nj4c) 13

[Prueba crear directorio](#_pk8oy54zkdie) 14

[Prueba Cambiar directorio](#_ldrgf6n8cqrz) 15

[Modificar archivo](#_5lorg8g7buo0) 16

[Prueba Ver Propiedades](#_1dfgp3x1you5) 17

[Prueba Ver Archivo](#_svgc0u7v64e7) 18

[Prueba Mover](#_9l22wfj53h0n) 18

[Prueba Imprimir Disco](#_hyh0kxd215ke) 20

[Prueba Remove](#_fdtigxd2f1zm) 20

[Prueba FIND](#_1t5yj84mi767) 22

[Prueba de COPY](#_lj1u3jb39o13) 23

[**Manual de usuario**](#_za6un4v3qt8a) **23**

[**Bitácora de trabajo**](#_aiznu8nn5ppt) **34**

[**Bibliografía**](#_roqer5cd7reh) **35**

# 

# 

# 

# 

# 

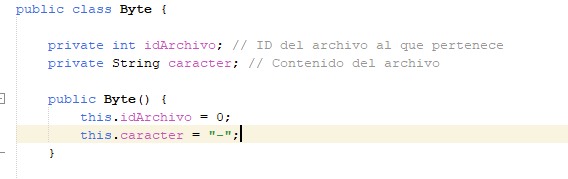
# **Introducción**

La información de un ordenador está almacenada en lo que se llaman archivos. Normalmente los archivos están formados por un nombre, un punto y una extensión. Según IBM (2022), un sistema de archivos se define como un tipo de estructura jerárquica de archivos y directorios. Todos los sistemas de archivos se componen de las estructuras necesarias para almacenar y manejar datos. Los diferentes sistemas operativos pueden utilizar distintos sistemas de archivos. Este proyecto se basa en crear un sistema de archivos que almacene sus datos en un archivo simulando un disco virtual con la forma de asignación de sectores a un archivo debe ser FIRST FIT y además debe de ser enlazada.

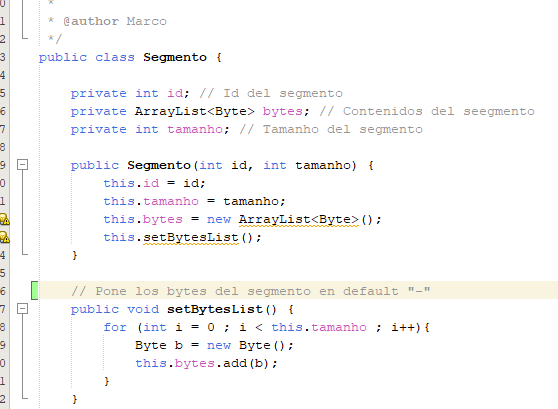
# **Estrategias de Solución**

## **Disco Virtual**

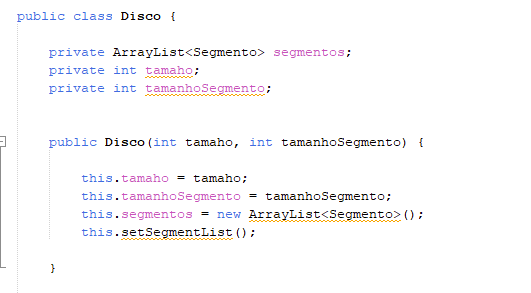
A la hora de implementar el disco virtual, se crearon una serie de estructuras que representan las distintas maneras en que se resguarda la información. Se diseñaron tres clases llamadas Disco, Segmento y Byte. Cada clase se complementa entre sí para conformar la estructura final, un Disco se compone de una cantidad de Segmentos, y un Segmento está compuesto por Bytes. Estos Bytes tienen como atributos un idArchivo y el carácter que guardan que pertenezca a ese archivo, como se ve en la siguiente imagen:



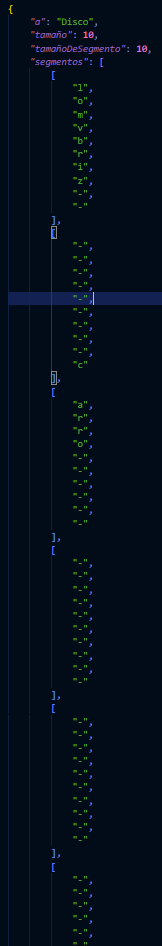
La estructura de Segmento se desarrolló de la siguiente manera:



Por último, se tiene la estructura disco, que se crea a base de la cantidad de sectores y cuantos espacios de memoria tendría cada segmento.

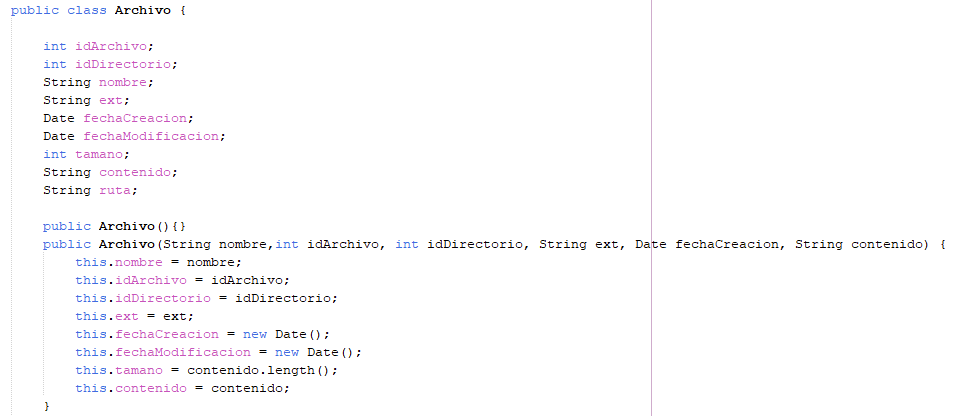


Para comprobar el funcionamiento del disco, la especificación del proyecto dictaba que se debía de mantener un archivo que guardará los datos del disco. Para esto se implementó un archivo en formato de JSON, en el cual se indicaba la cantidad de segmentos y su respectivo byte. De la siguiente manera:

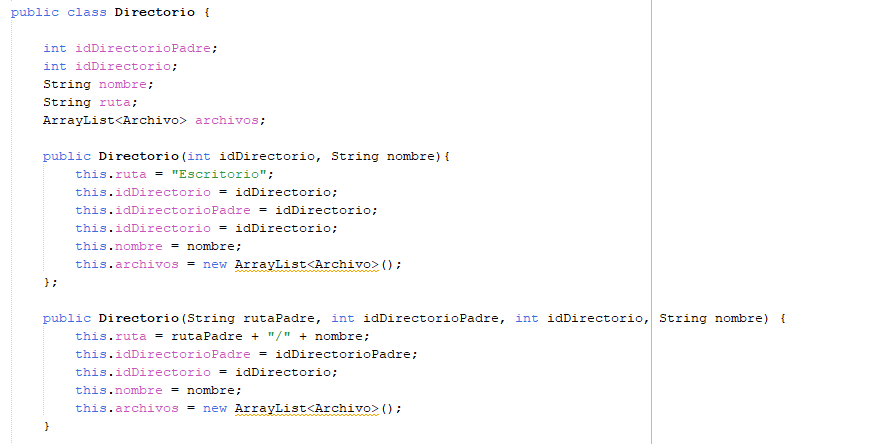


## **Jerarquía de Directorios y Archivos**

Para el desarrollo de una jerarquía de directorios, se implementaron enteros como un estilo de identificación de los directorios. Es decir, cada directorio tenía dos identificaciones, la de sí mismo y la de su directorio padre. Esto es similar para los archivos, sin embargo, estos al no poder tener “hijos”, solo tienen una identificación para el directorio padre al que pertenecen. La estructura de la clase archivo se desarrolló de la siguiente manera:

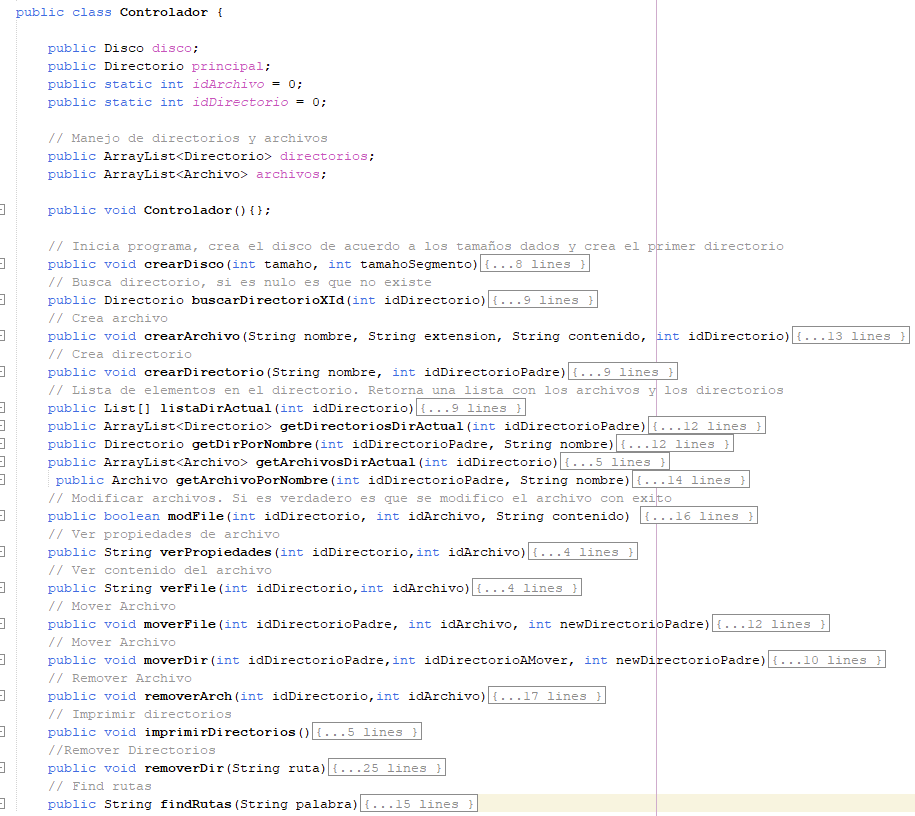


La estructura Directorio, posee una lista de los archivos que pertenecen al mismo, por lo que su desarrollo fue el siguiente:



## **Funciones del File System**

Las funciones se implementan en cada estructura, pero la principal clase que se encarga de manipular las funciones del sistema de archivos es la clase Controlador. Esta clase posee un disco y una lista de directorios y por medio de estos atributos se controlan los datos del programa. La estructura sería la siguiente:



**Análisis de Resultados**

| **Funcionalidad** | **Porcentaje de realización** | **Justificación** |
| --- | --- | --- |
| CREATE | 100 | Se desarrolló con éxito. |
| FILE | 100 | Se desarrolló con éxito. |
| MKDIR | 100 | Se desarrolló con éxito. |
| CambiarDIR | 100 | Se desarrolló con éxito. |
| ListarDIR | 100 | Se desarrolló con éxito. |
| ModFILE | 100 | Se desarrolló con éxito. |
| VerPropiedades | 100 | Se desarrolló con éxito. |
| VerFile | 100 | Se desarrolló con éxito. |
| CoPY | 75 | **No se incluyó, el copiar de directorios.** |
| MoVer | 100 | Se desarrolló con éxito. |
| ReMove | 100 | Se desarrolló con éxito. |
| FIND | 100 | Se desarrolló con éxito. |
| TREE | 0 | **No se implementó el imprimir el árbol.** |

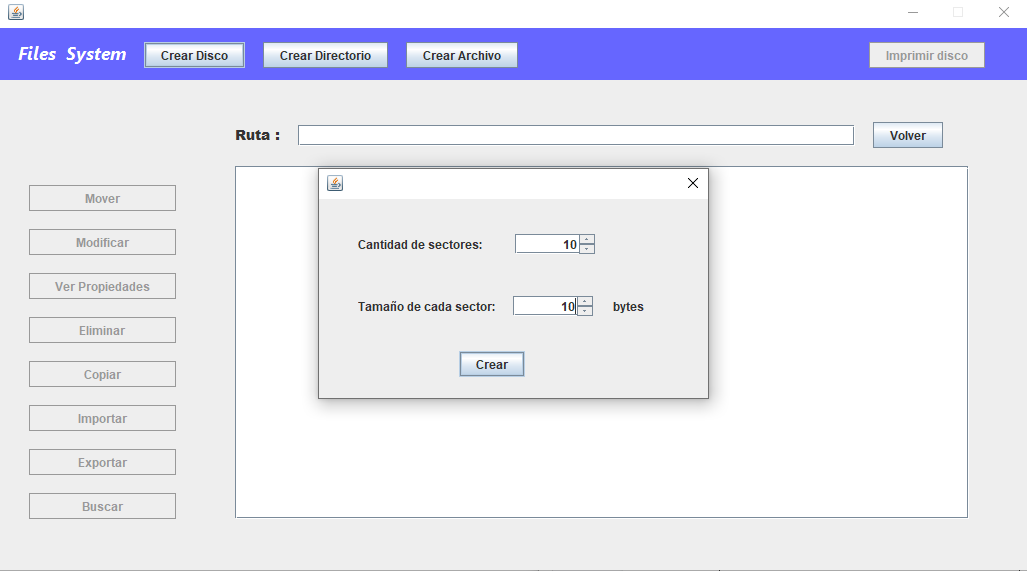
# **Lecciones aprendidas**

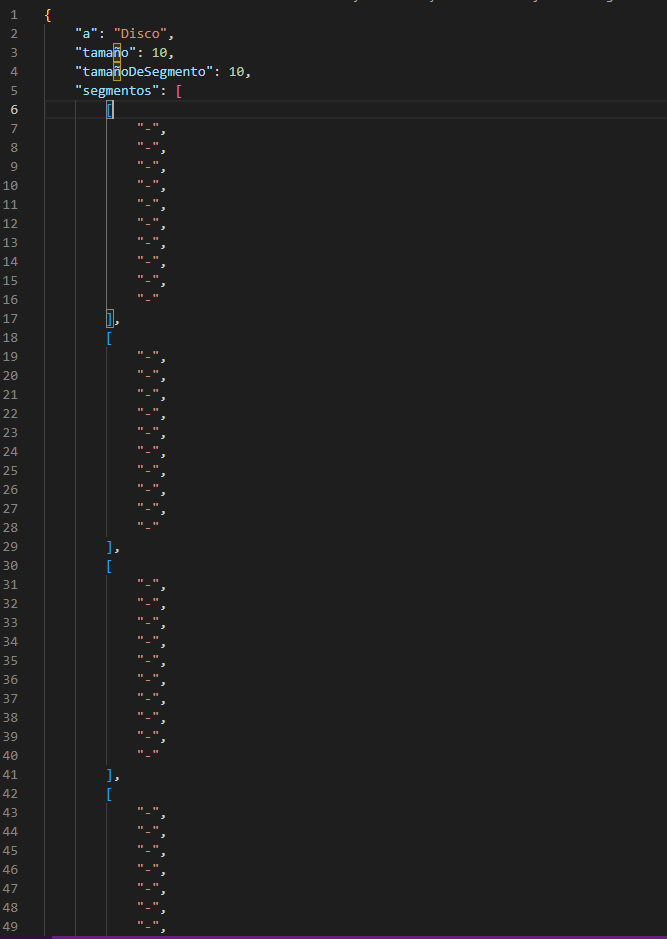
Se complementaron los conocimientos previamente aprendidos de grupos anteriores como la programación en el lenguaje Java, y el desarrollo de la interfaz gráfica por medio de NetBeans 13. También se comprendió el funcionamiento del sistema de archivos. Dando como referencia a que un sistema de archivos es el sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta.

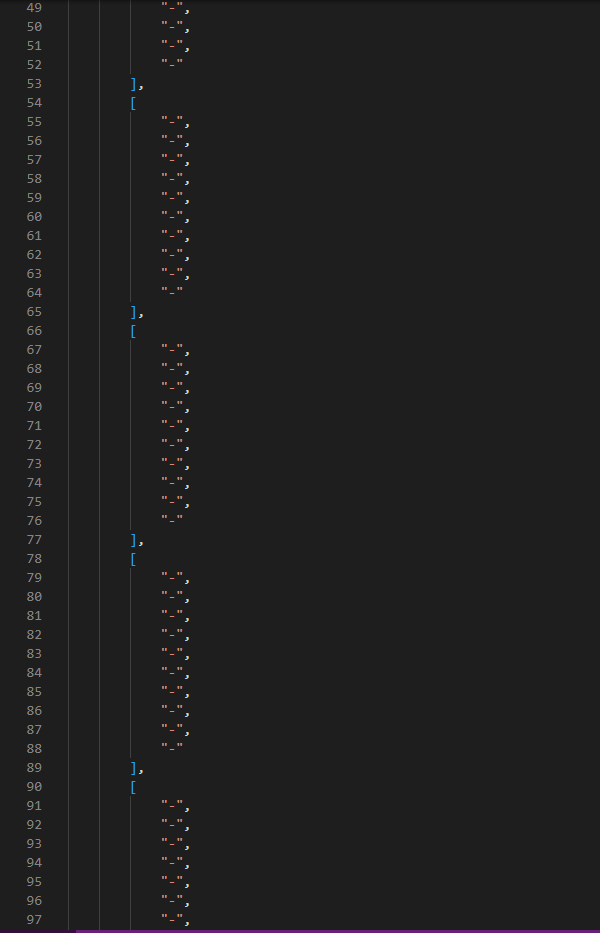
# **Casos de pruebas**

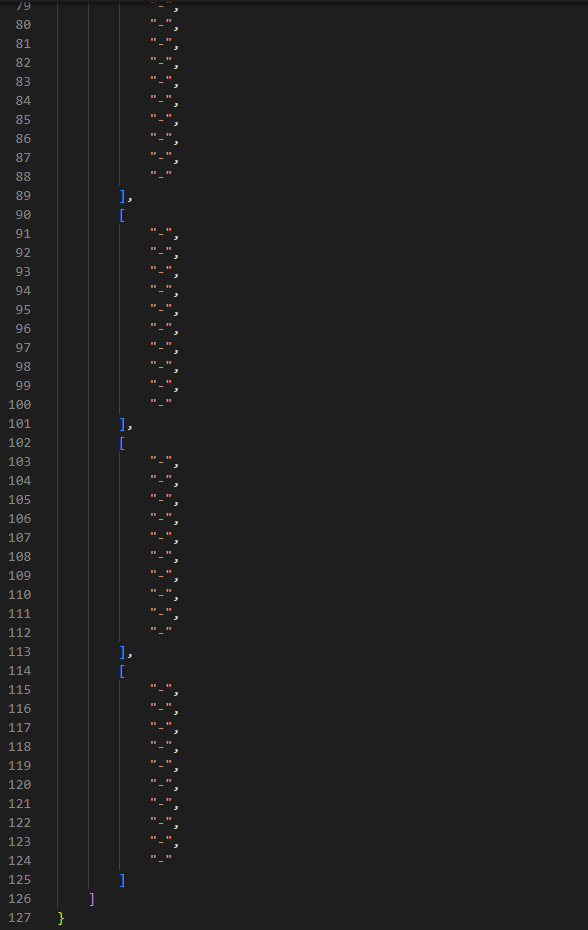
## **Prueba crear Disco Virtual**

Se ingresa 10 cantidad de sectores y 10 de tamaño por sector.



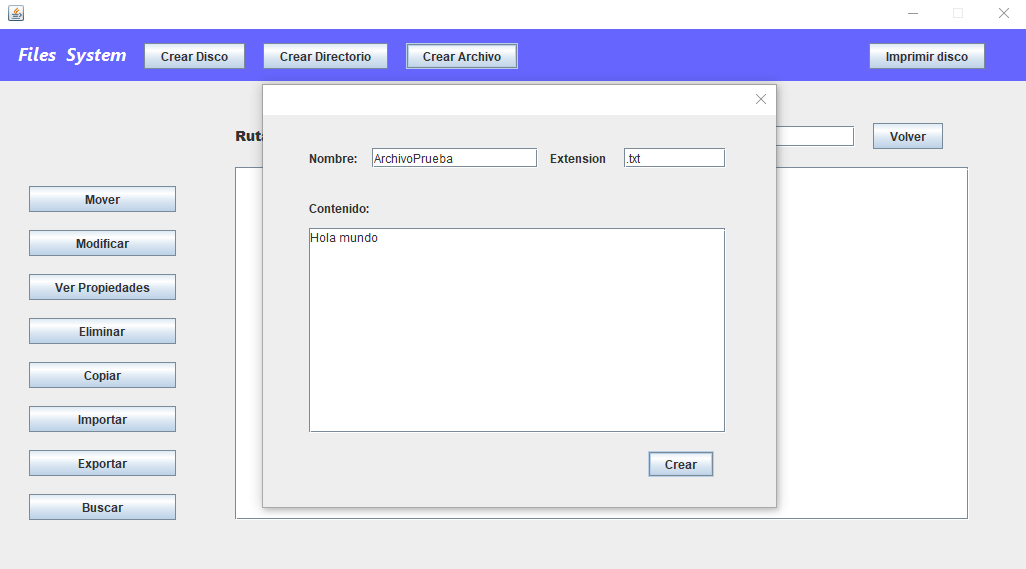


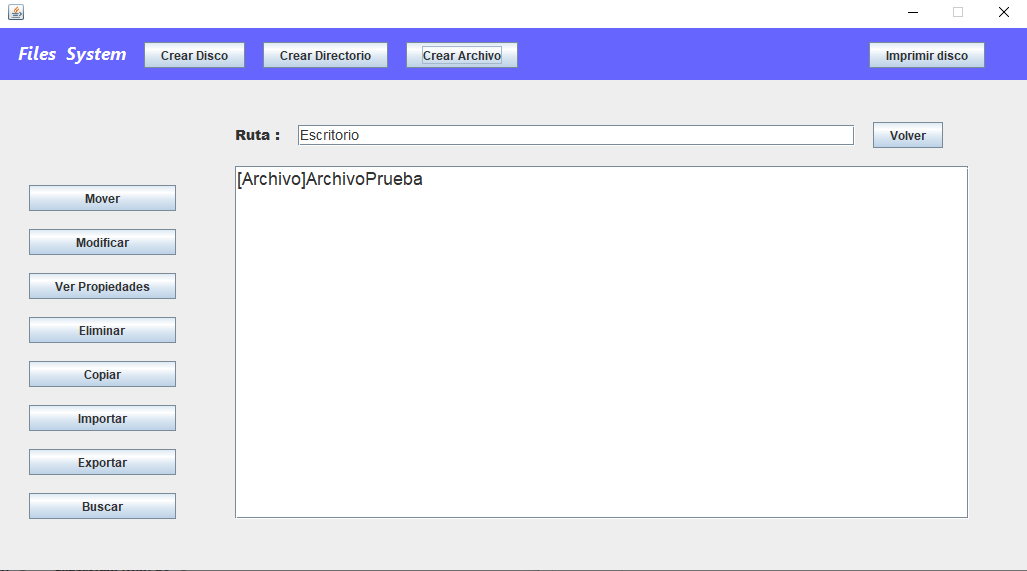




## **Prueba crear Archivo**

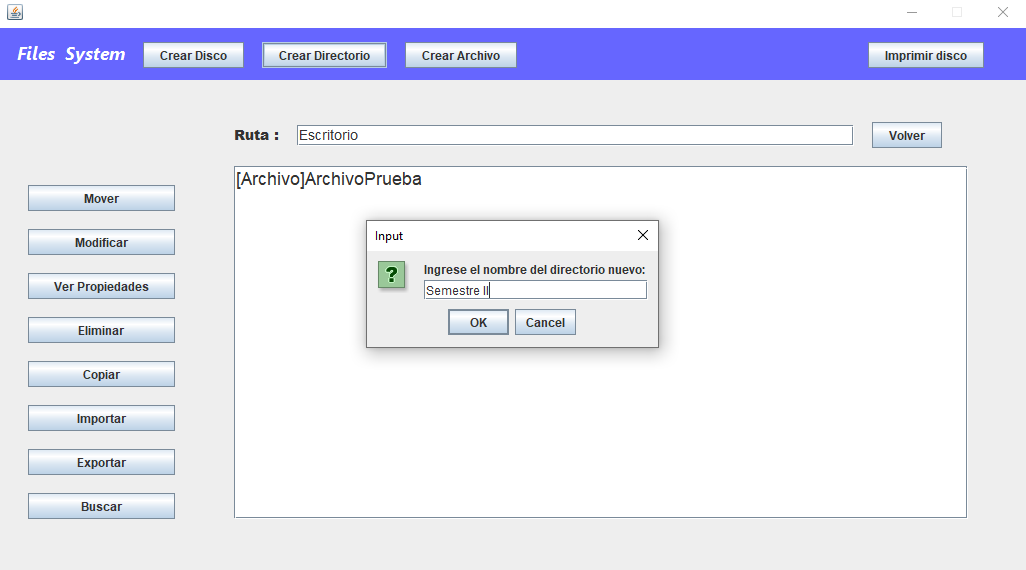
Para crear un archivo, se crea un archivo en la ruta Escritorio con los siguientes datos: Nombre: Hola mundo, Extension: txt, Contenido: “Hola mundo”.

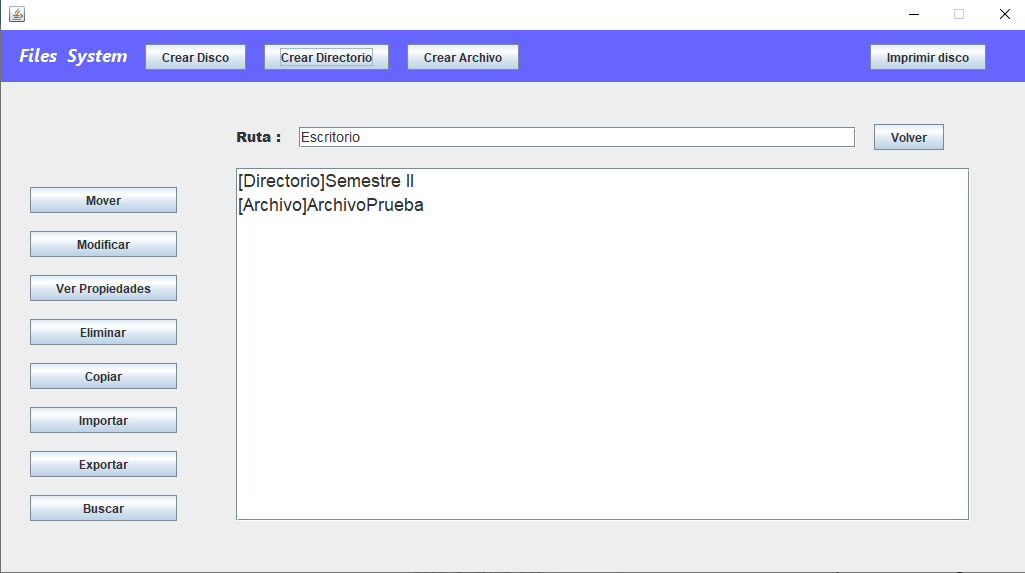




## **Prueba crear directorio**

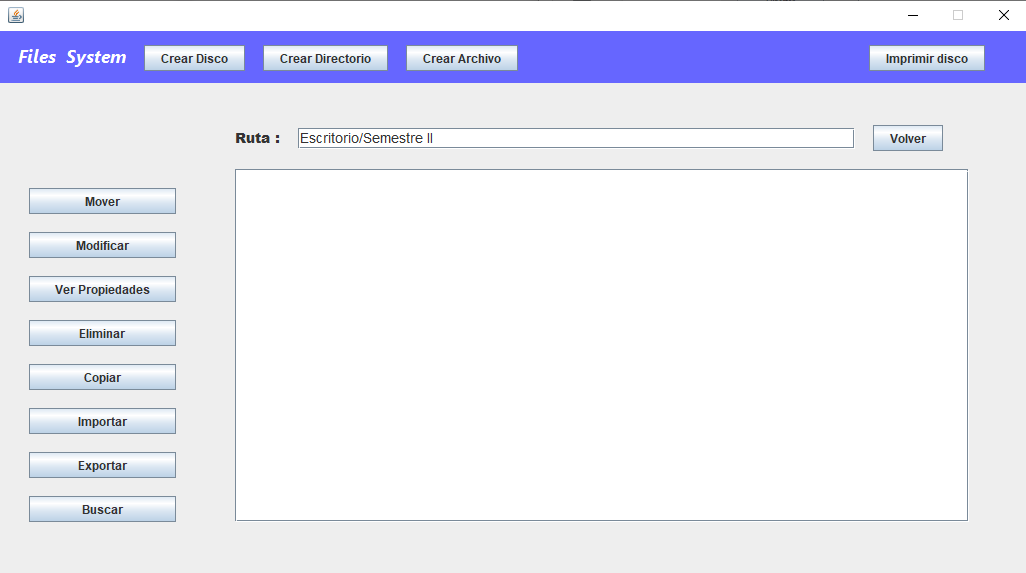
Se crea un directorio de nombre “Semestre II” en el directorio actual “Escritorio”:

****

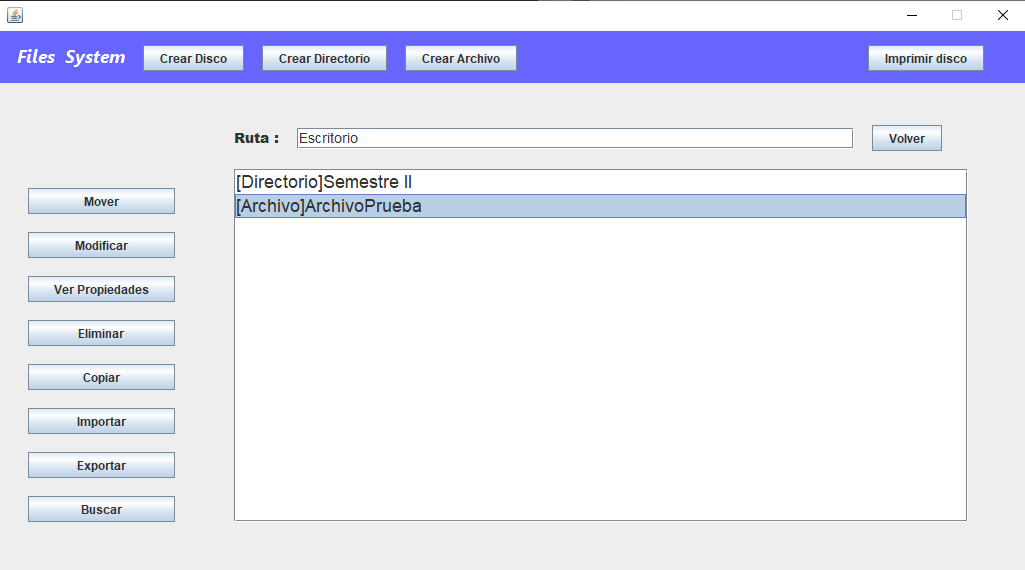


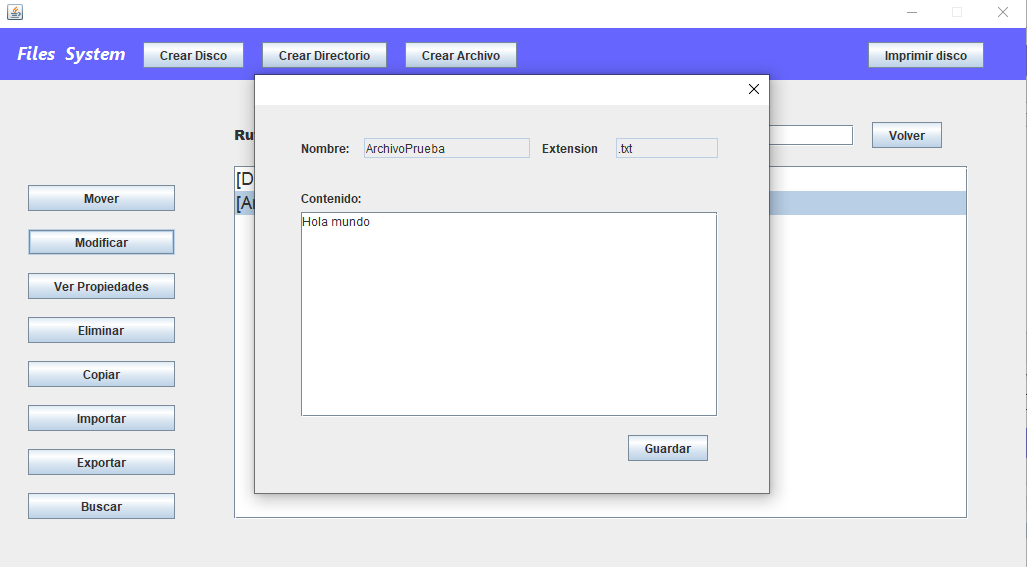
## **Prueba Cambiar directorio**

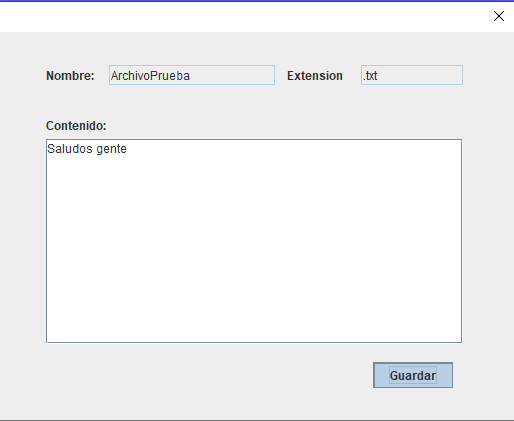
En esta prueba se presiona doble click sobre el directorio creado anteriormente de nombre “Semestre II” y esto abre dicho directorio, que de momento se encuentra vacío:



## **Modificar archivo**

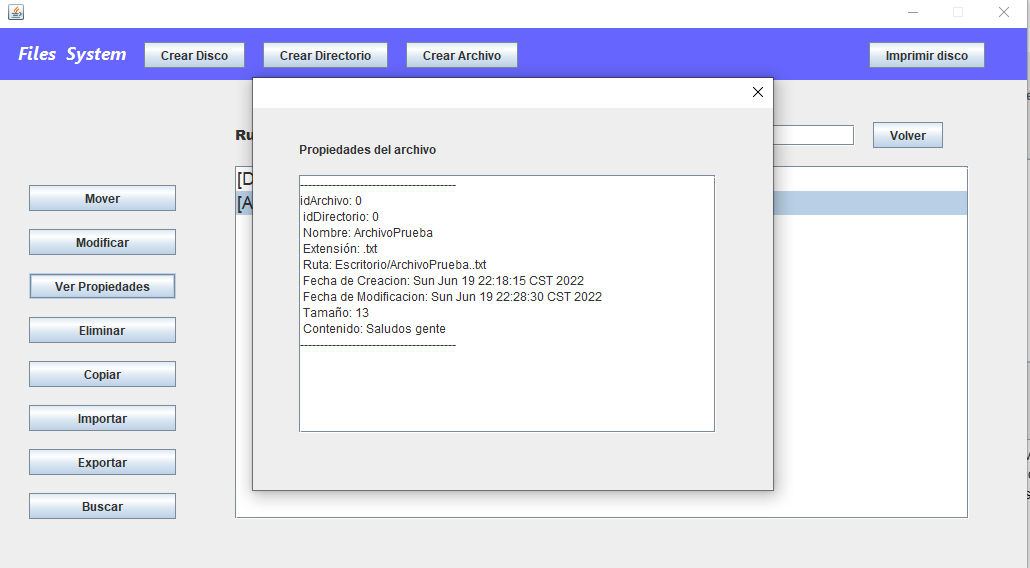
Se selecciona el ArchivoPrueba y se le modifica su contenido “Hola mundo” por “Saludos gente”.





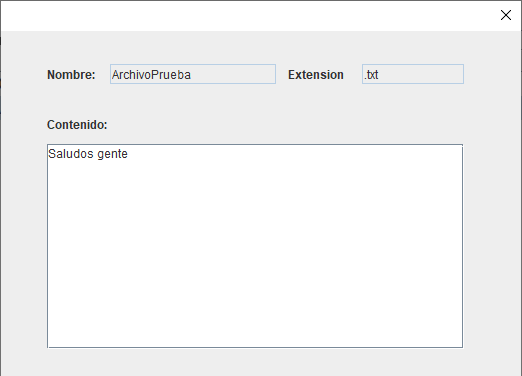
## **Prueba Ver Propiedades**

Se selecciona el ArchivoPrueba y se selecciona el botón “Ver Propiedades”, esto refleja todas las propiedades del archivo.



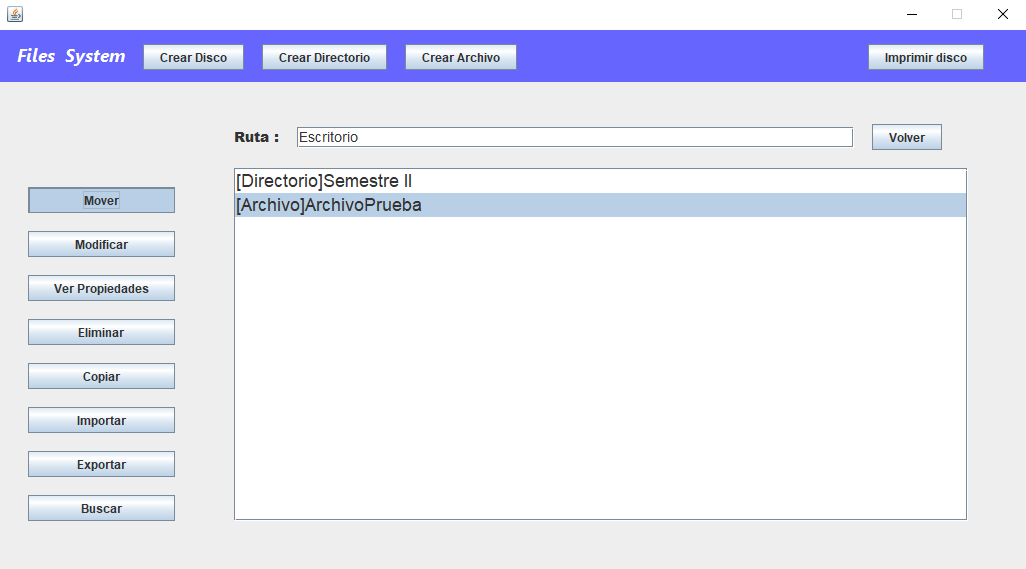
## **Prueba Ver Archivo**

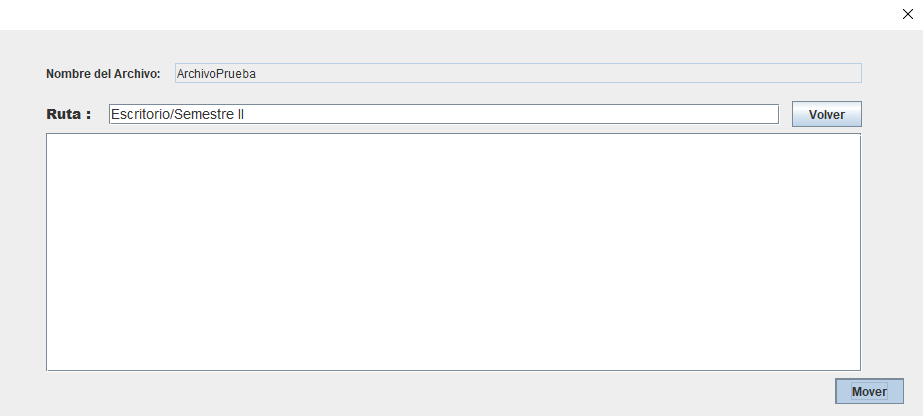
Se abre el ArchivoPrueba y se observa el contenido.

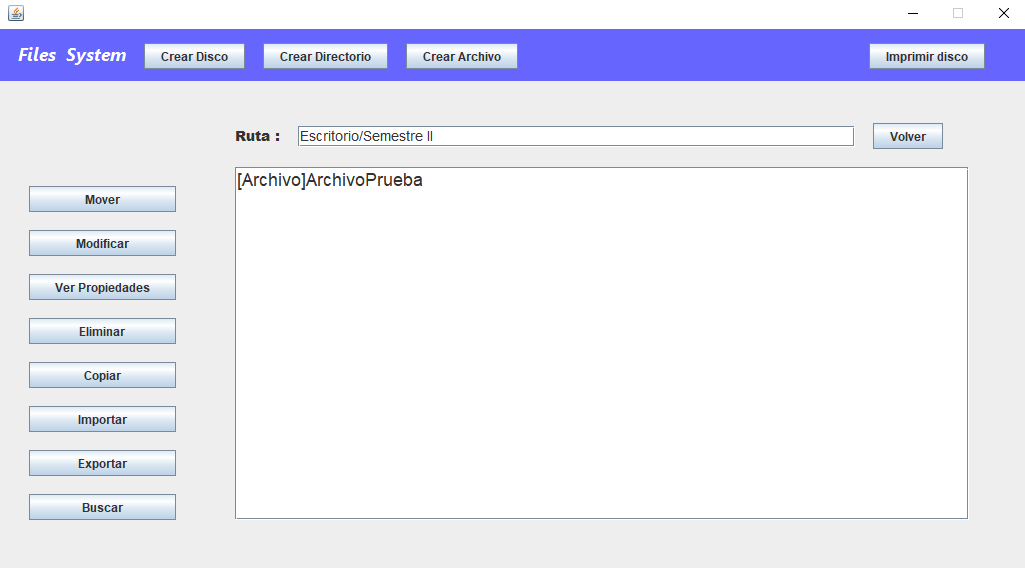


## **Prueba Mover**

Se ubica en el directorio “Escritorio” y se mueve de este directorio el “Archivo Prueba” a el directorio “Semestre II”:

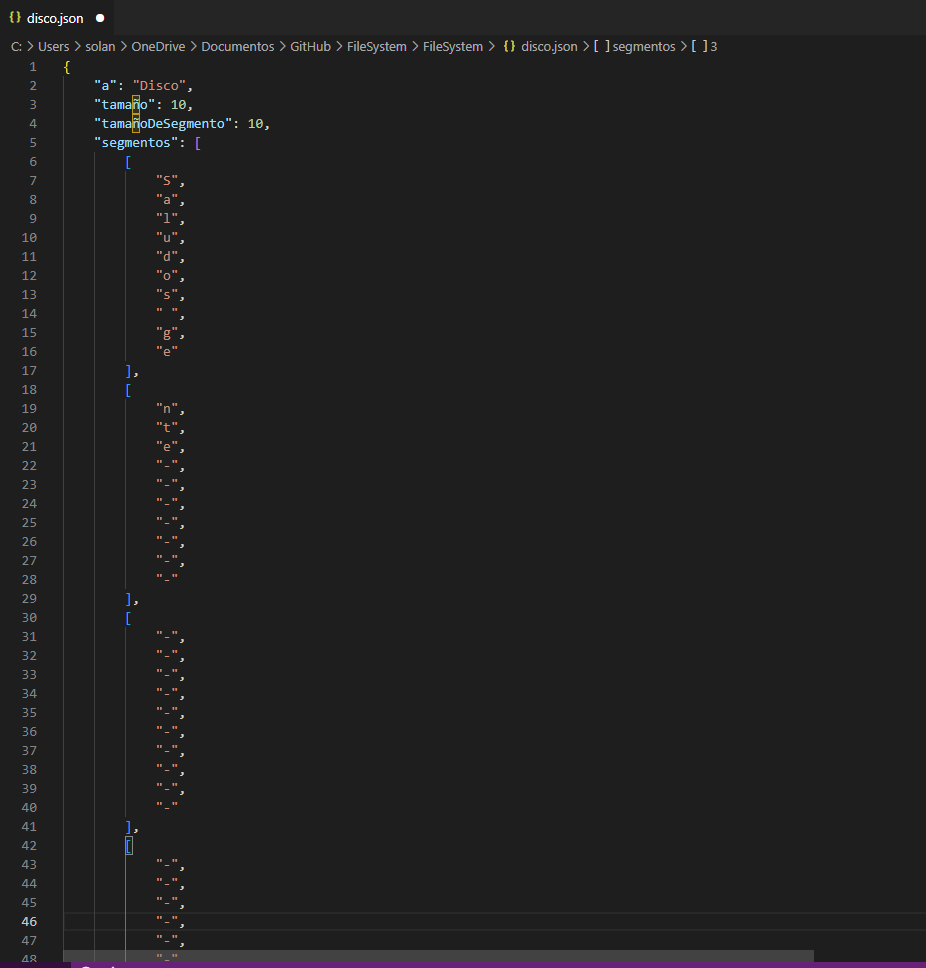






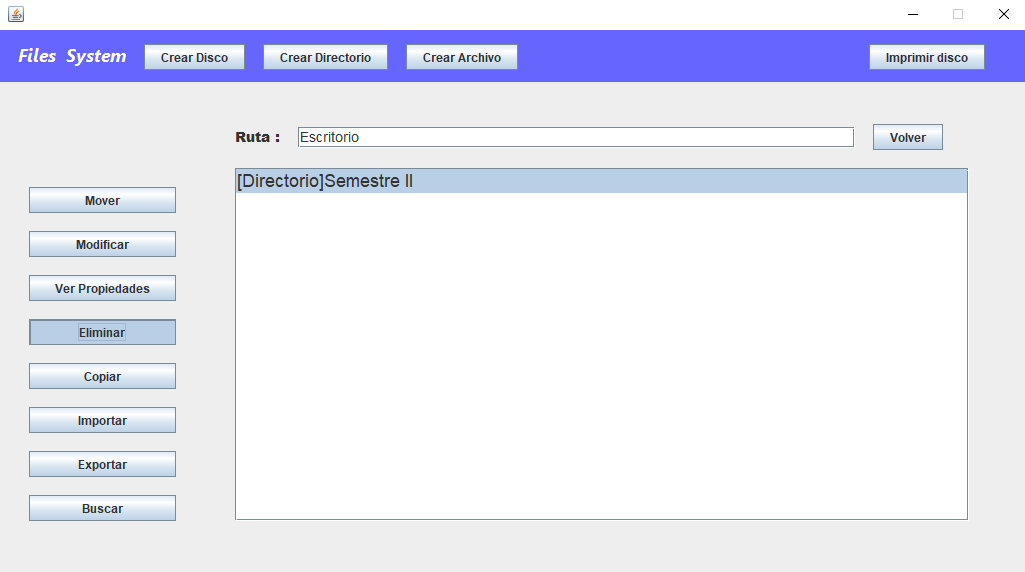
## **Prueba Imprimir Disco**

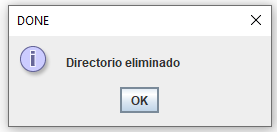
Se presiona el botón Imprimir Disco y se observa el disco actual.



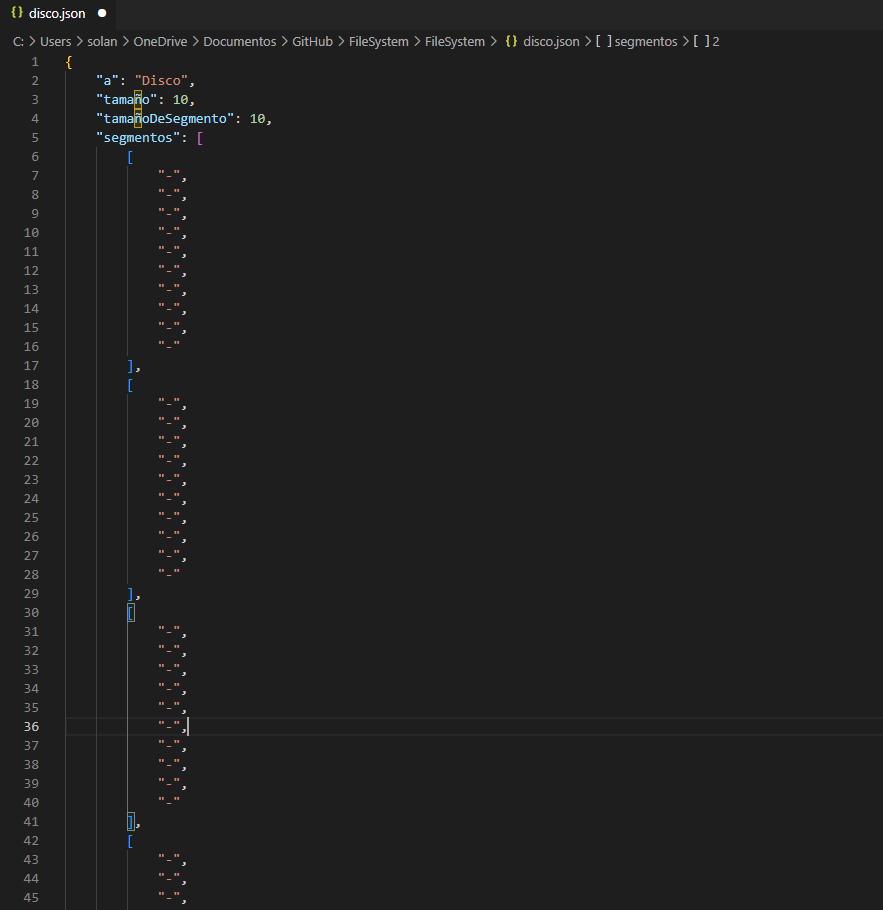
## **Prueba Remove**

Se elimina el directorio “Semestre II” que dentro de sí contiene al “ArchivoPrueba”:





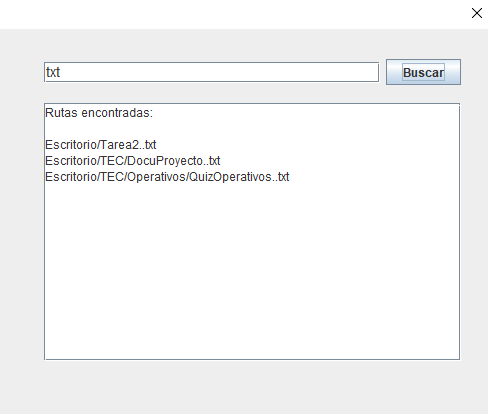
Esa eliminación del archivo se refleja en el disco:



## **Prueba FIND**

Para este caso de prueba se creó primeramente en el directorio “Escritorio” un archivo de extensión .txt llamado “Tarea2”, luego un nuevo directorio llamado “TEC” y dentro de este directorio se creó otro nuevo llamado “Operativos”. En el directorio “TEC” se creó también un archvio .txt de nombre “DocuProyecto” y en el directorio “Operativos” se creó un archivo .txt de nombre “QuizOperativos”.

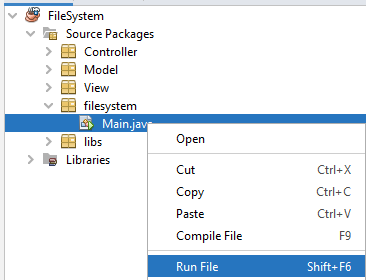
Ahora bien, se prueba buscando la palabra “.txt”:



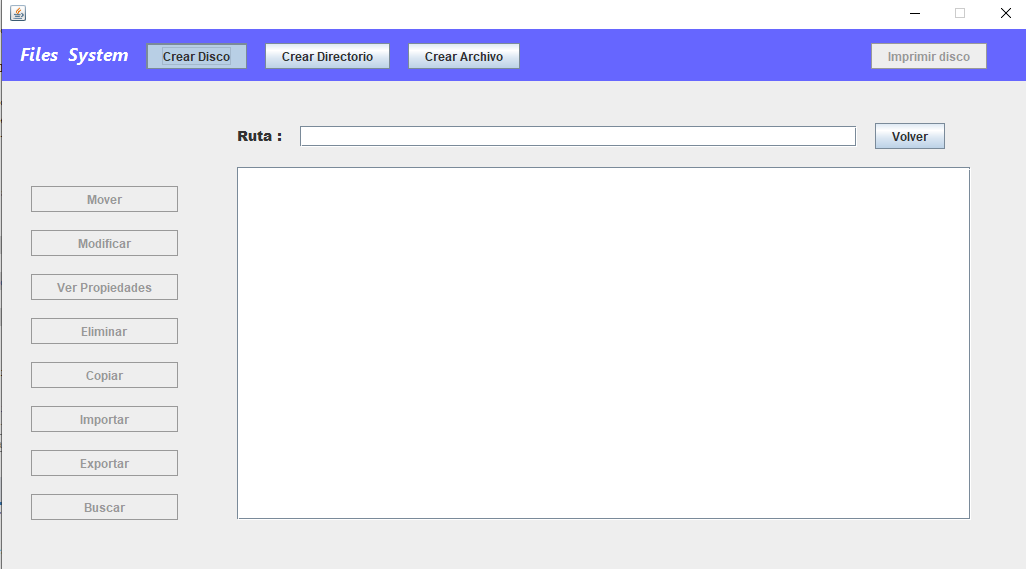
## **Prueba de COPY**

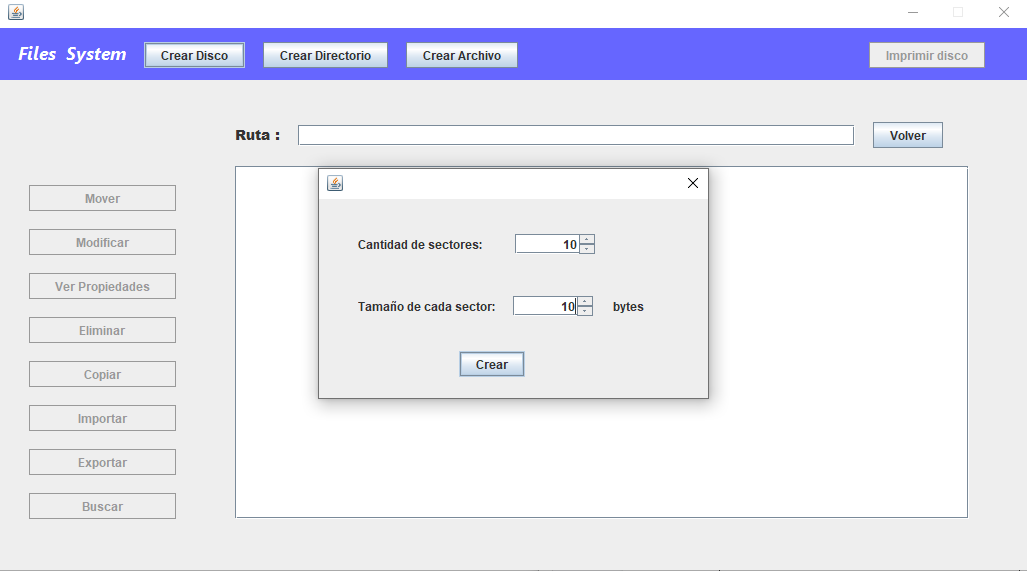
# **Manual de usuario**

Para ejecutar el proyecto primero se debe abrir el proyecto Java de File System y ejecutar el archivo **Main.Java** de la siguiente manera:



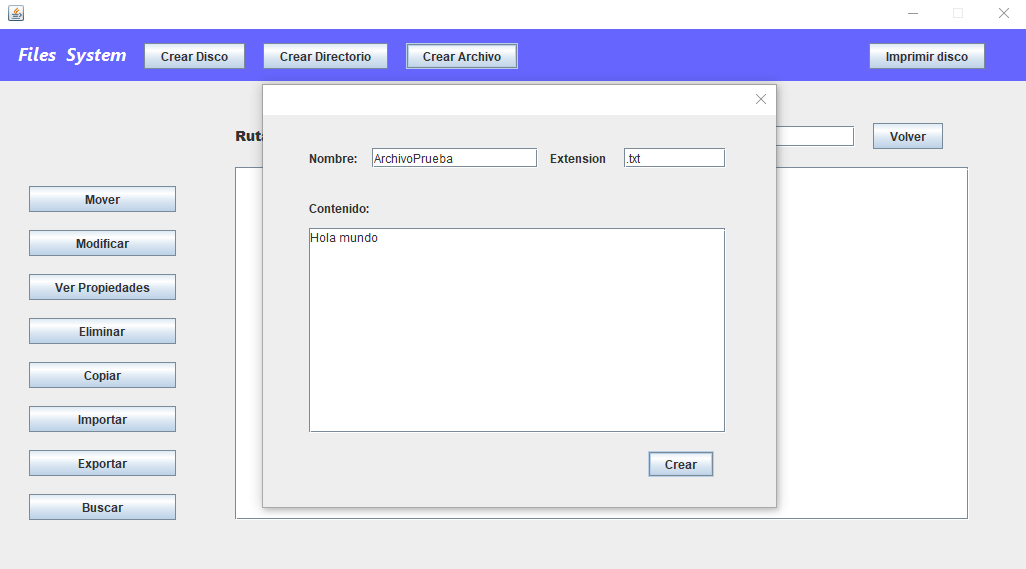
El siguiente paso es crear el disco virtual, se deben ingresar dos parámetros, uno es la cantidad de sectores y el otro es el tamaño de cada sector, a continuación se muestra un ejemplo:



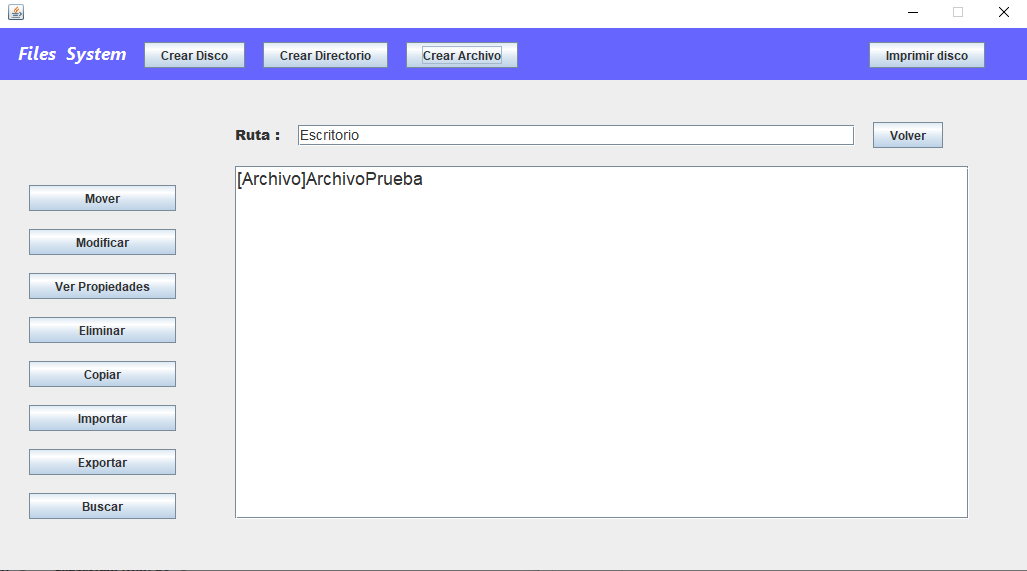


Una vez realizados los pasos anteriores ya se tendrá acceso a las funcionalidades de MI File System.

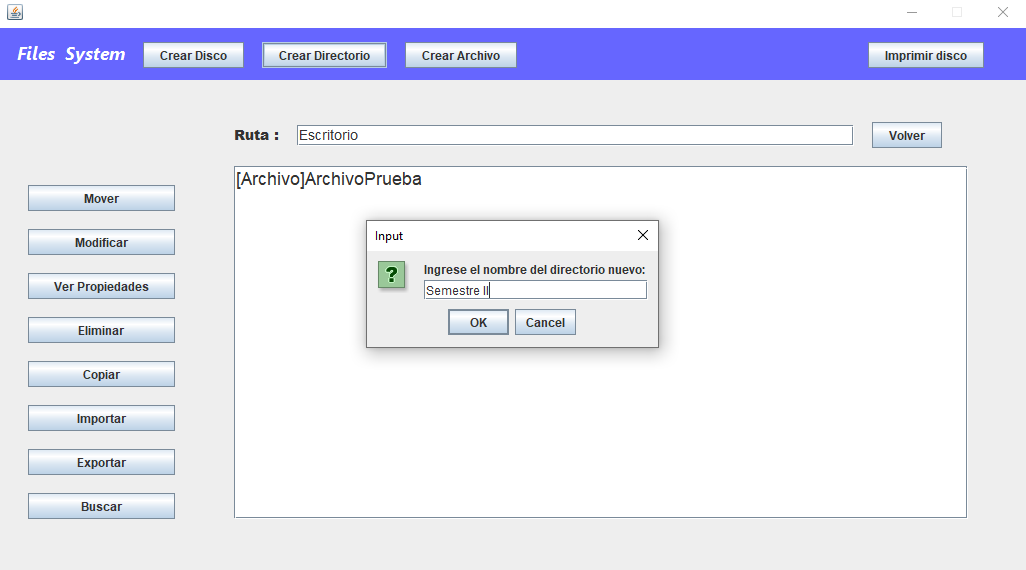
**Crear Archivo:** Para crear un archivo, se accede a través del botón “Crear Archivo”, esto abrirá una pequeña ventana donde se podrán ingresar los datos del archivo(Nombre, Extensión y Contenido), a continuación se brinda un ejemplo:



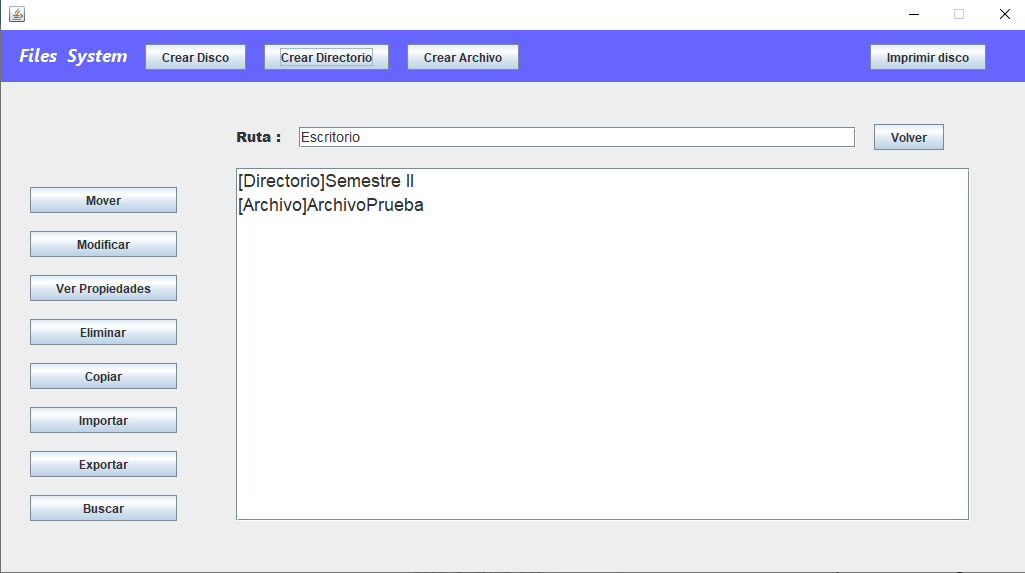
Una vez creado el archivo se verá en la ruta donde fue creado:



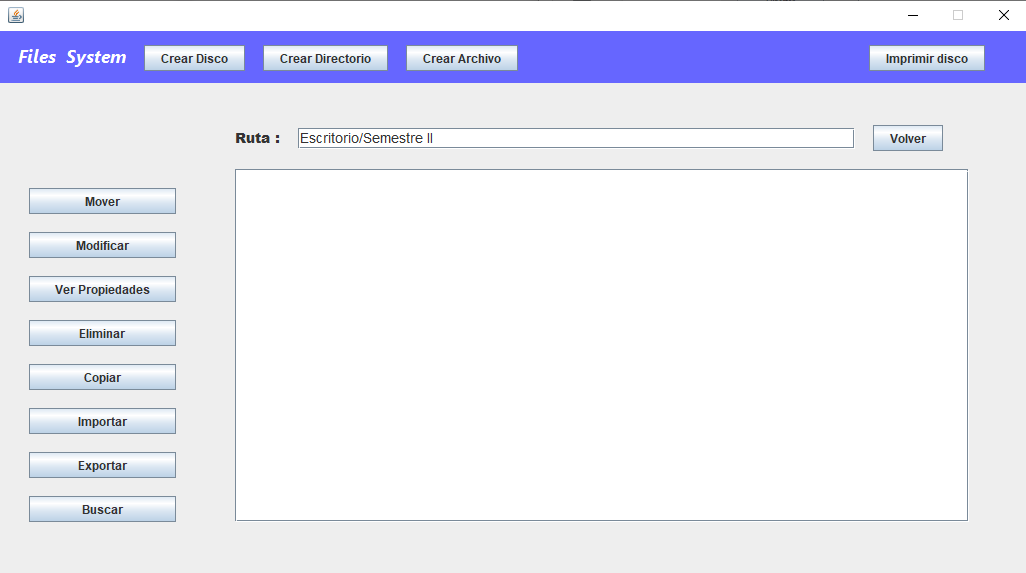
**Crear directorio:** Esta funcionalidad crea un directorio en el directorio Actual, se accede a través del botón “Crear Directorio”, una vez presionado se abre una ventana donde se solicita el nombre del directorio y se le da “ok” para crear el directorio:

****

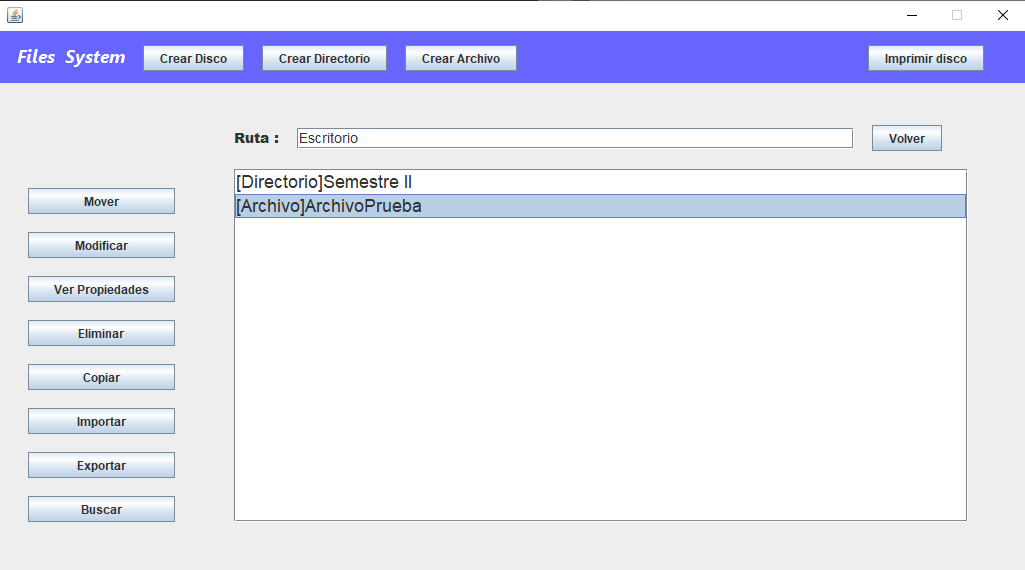
Una vez creado el directorio se refleja en la ruta donde fue creado, en este caso el directorio se encuentra dentro del directorio actual “Escritorio”:



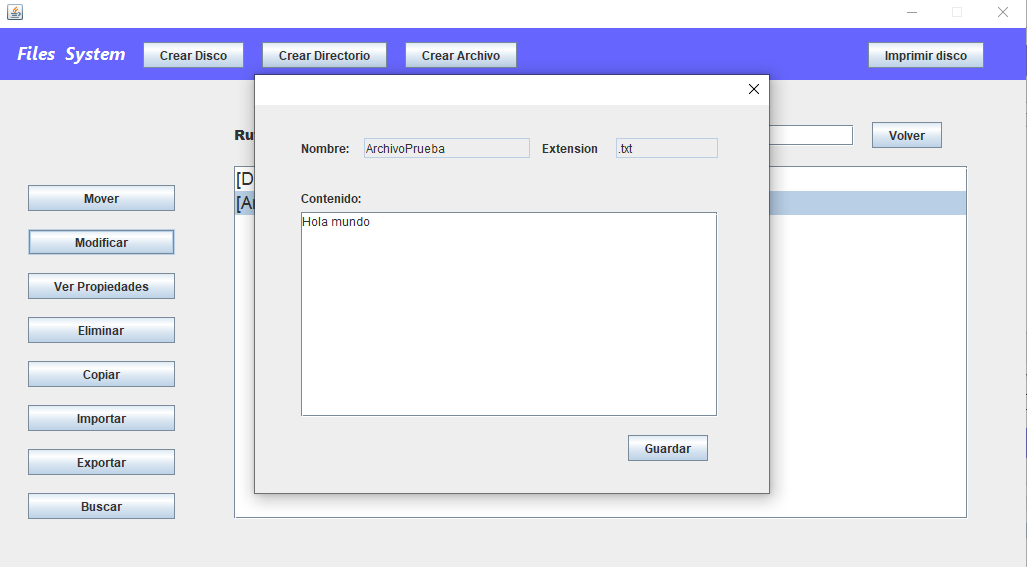
**Cambiar directorio:** Para cambiar de directorio simplemente se debe dar click dos veces sobre el directorio al cual se quiere ir, en el siguiente ejemplo se presiona doble click sobre el directorio creado anteriormente de nombre “Semestre II” y eso nos lleva dentro de este directorio, que de momento se encuentra vacío:



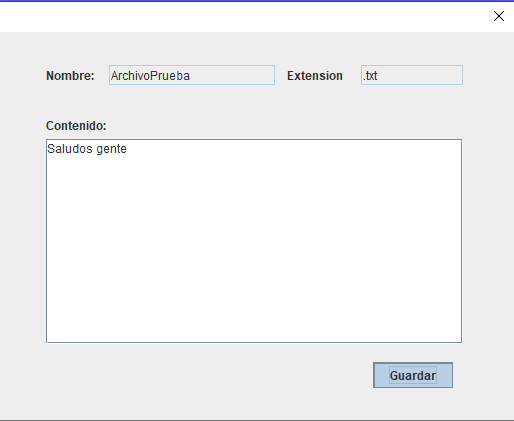
**Modificar archivo.** Permite seleccionar un archivo y cambiarle el contenido, para esto se debe seleccionar el archivo y presionar el botón “Modificar”:



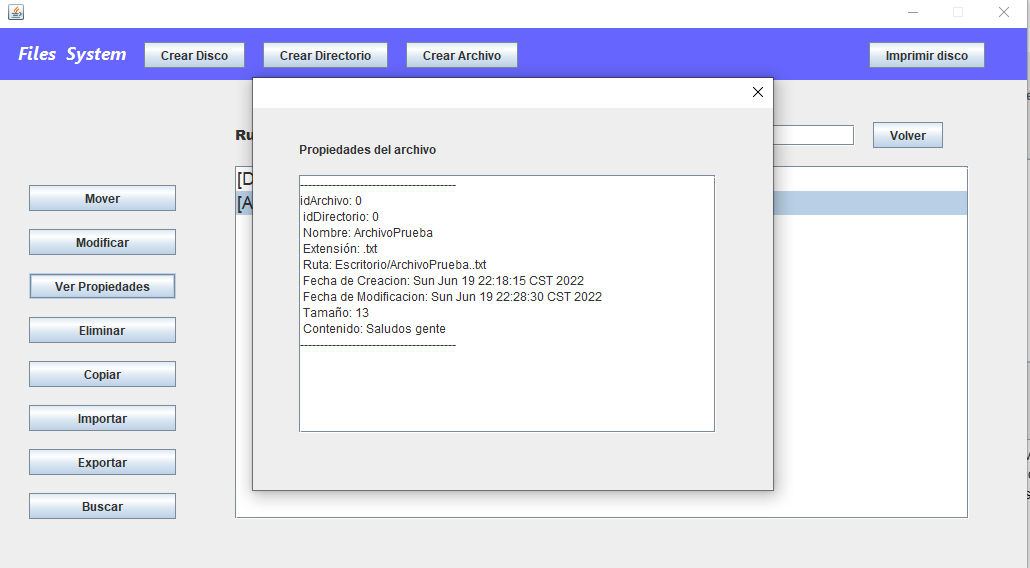
Esto abre una ventana donde se observa el nombre del archivo a modificar, su extensión y su contenido:



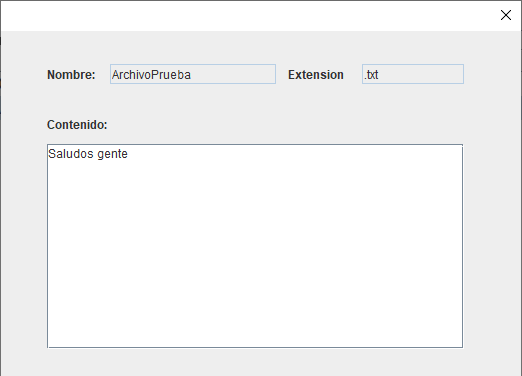
Para modificar el contenido simplemente se modifica en el campo de texto y se le da click al botón “Guardar”:



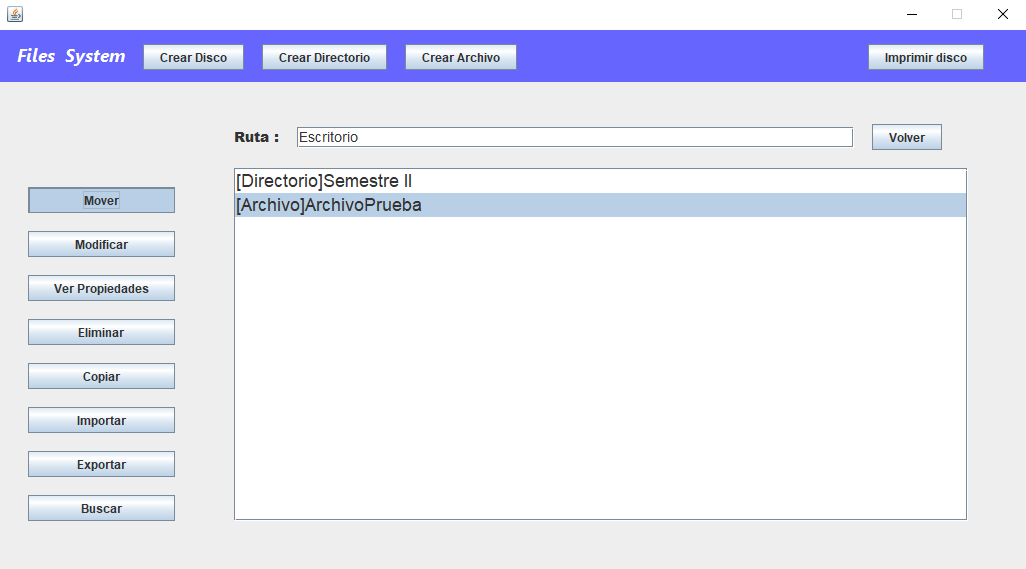
**Ver Propiedades.** Esta opción permite ver las propiedades de un archivo(Nombre, Extensión, Fecha de Creación, Fecha de Modificación y tamaño), simplemente se debe seleccionar el archivo al que se le quiere ver las propiedades y seleccionar el botón “Ver Propiedades”, esta acción nos abre una ventana con las propiedades del archivo, como se muestra en la siguiente imagen:



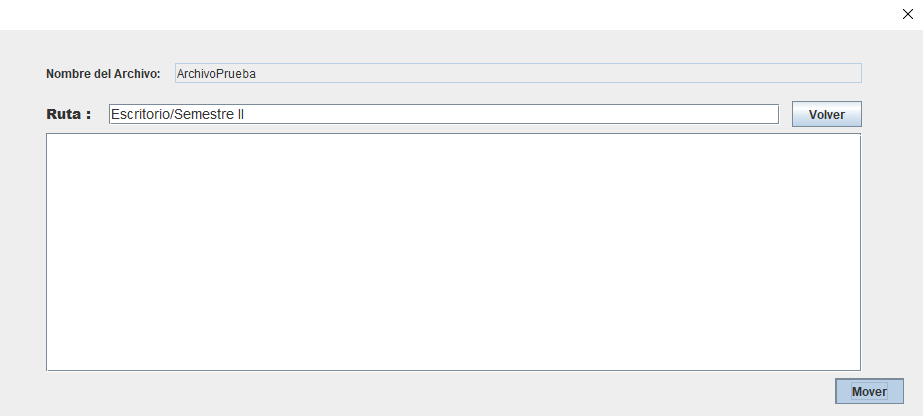
**Ver archivo:** Esta funcionalidad permite ver el contenido de un archivo, para llevar a cabo esto se debe dar click dos veces sobre el archivo:



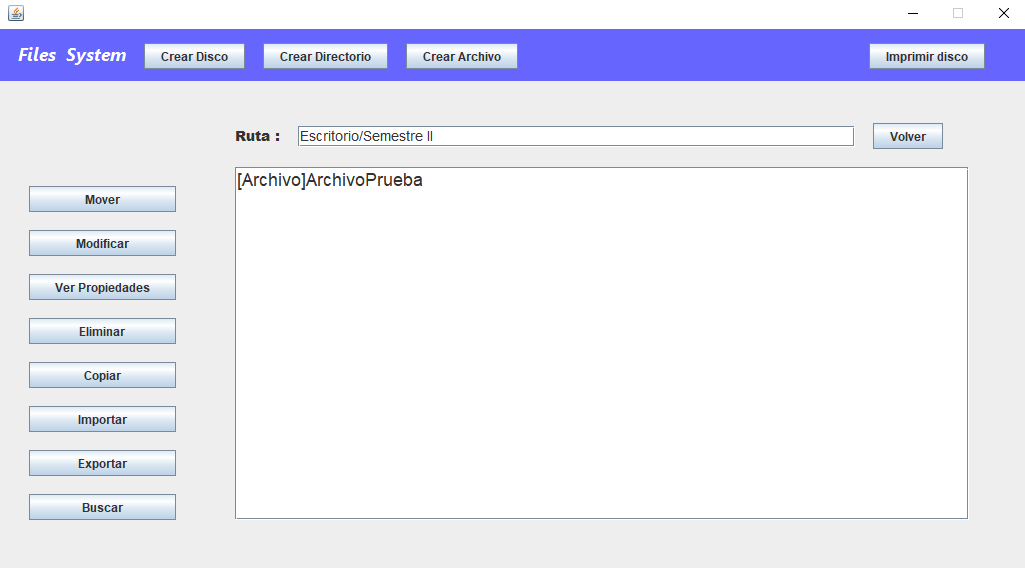
**Mover:** Esta funcionalidad permite mover un archivo o directorio, a otro directorio. Para cualquier caso simplemente se debe seleccionar el archivo o directorio y dar click al botón “Mover”, esto nos abre una ventana donde podemos seleccionar el directorio a donde se quiere mover. A continuación se muestra un ejemplo donde se mueve el “Archivo Prueba” a el directorio “Semestre II”:



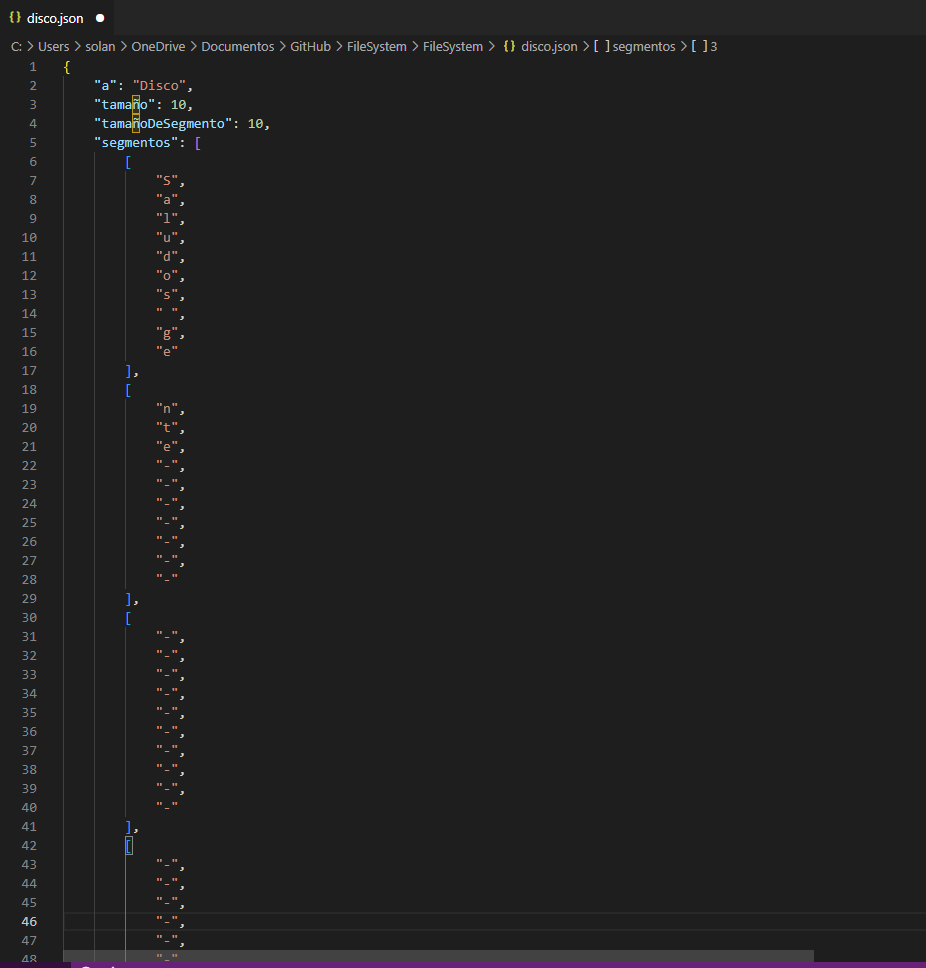
Una vez que se está en el directorio a mover, se da click en el botón “Mover”:



Ahora si ingresamos al directorio “Semestre II” se observa el archivo que movimos recientemente:

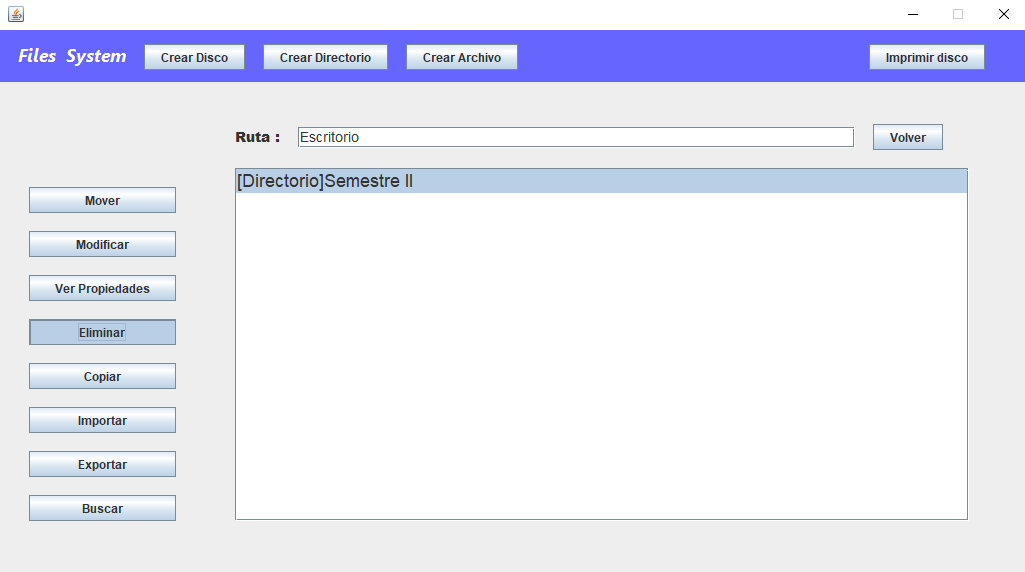


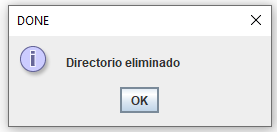
**Imprimir Disco:** Esta funcionalidad genera un archivo Json que simula el disco, este archivo puede ser visualizado en cualquier momento, en el ejemplo actual si vemos el Json(El Disco) se observa el contenido del disco en este momento:

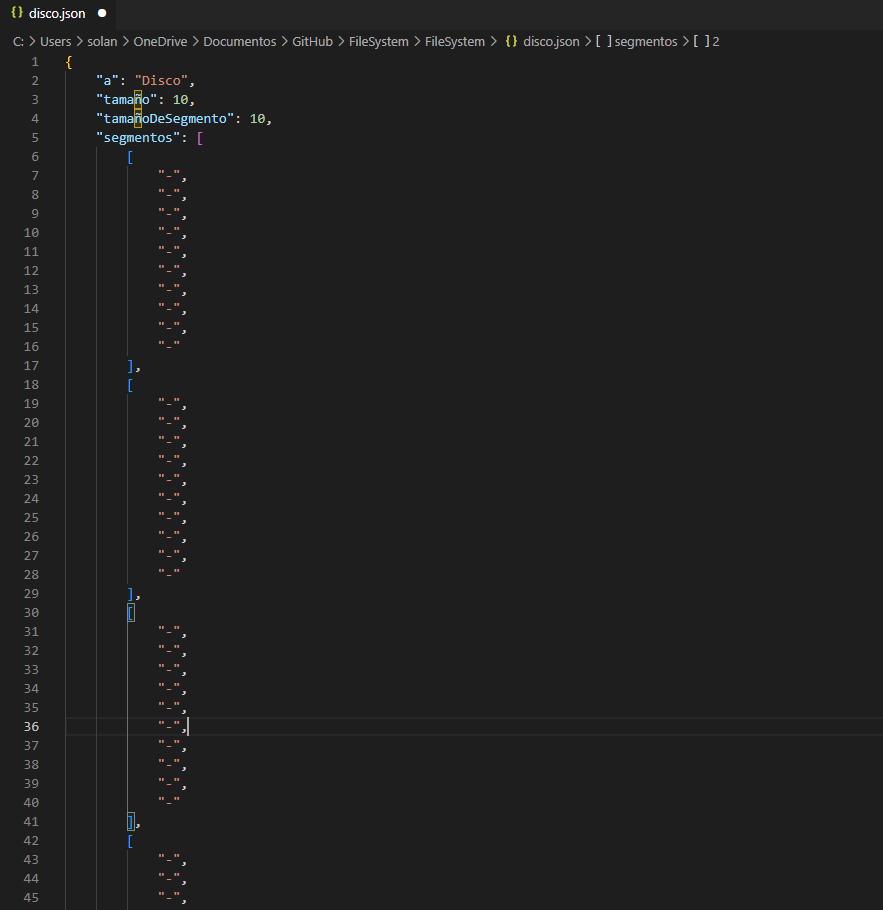


**Remove:** Esta funcionalidad permite eliminar un archivo o directorio, se debe seleccionar el archivo o directorio a eliminar y posteriormente presionar el botón “Eliminar”. En el caso de la eliminación de directorios borra todos los archivos y directorios que estén dentro de él, estos cambios se ven reflejados en el disco.

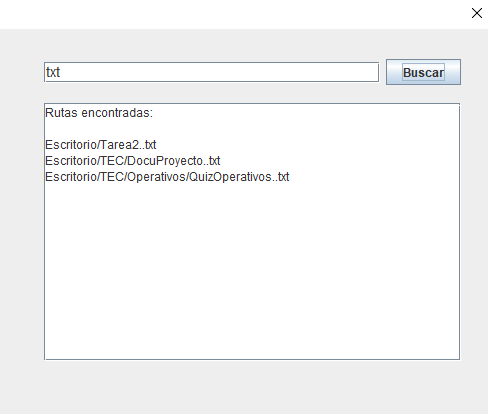
El siguiente ejemplo muestra cómo se elimina el directorio “Semestre II” que dentro de sí contiene al “ArchivoPrueba”:







**FIND:**  Esta funcionalidad recibe un nombre de archivo o directorio y lista todas las rutas de MI File System con archivos de ese nombre, para acceder a esta funcionalidad se debe presionar el botón “Buscar”, esto abrirá una pequeña ventana donde se debe ingresar el nombre a buscar:



# **Bitácora de trabajo**

Se debe recalcar que para esta sección, no se realizaron consultas ni al asistente ni a la profesora a lo largo del desarrollo del proyecto.

| **Actividad** | **Fecha** |
| --- | --- |
| Se realizó una reunión con el fin de evaluar las herramientas en las que se va a desarrollar el proyecto. | 12 de junio 2022 |
| Se repartieron los temas para investigar del desarrollo del proyecto. | 15 de junio 2022 |
| Se realizó una reunión para determinar la repartición de tareas. | 16 de junio 2022 |
| Se realizó la última reunión para terminar el proyecto y el desarrollo de la documentación. | 19 de junio 2022 |

# **Bibliografía**

*Crear archivos de texto en Java*. (2021, 1 octubre). decodigo.com. <https://decodigo.com/java-crear-archivos-de-texto>

GeeksforGeeks. (2021, 28 enero). *How to Fix «class, interface, or enum expected» Error in Java with Examples?* <https://www.geeksforgeeks.org/how-to-fix-class-interface-or-enum-expected-error-in-java-with-examples/>

*How Comment & Uncomment a block of Code in NetBeans. (2018, 13 septiembre). [Vídeo]. YouTube.*

[*https://www.youtube.com/watch?v=0tjggTwSZls*](https://www.youtube.com/watch?v=0tjggTwSZls)

*How do I read separate parts from a txt File to show in Java GUI?* (2020, 7 octubre). Stack Overflow. <https://stackoverflow.com/questions/64248097/how-do-i-read-separate-parts-from-a-txt-file-to-show-in-java-gui>

*How to Use File Choosers (The JavaTM Tutorials > Creating a GUI With Swing> Using Swing Components)*. (2022). Java Documentation Oracle.

<https://docs.oracle.com/javase/tutorial/uiswing/components/filechooser.html>