



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Sistemas Operativos

PROFESOR(A):

Gunnar Eyal Wolf Iszaevich

NOMBRES(S) ALUMNOS(s):

Hernández Ortiz Jonathan Emmanuel

Pérez Avin Paola Celina de Jesús

PROYECTO 03

(Micro) Sistema de Archivos

SEMESTRE 2024-1

ÍNDICE

ÍNDICE.....	2
OBJETIVOS.....	3
INTRODUCCIÓN.....	3
AUTORES.....	4
ESTRATEGIA.....	4
REQUISITOS DE USO.....	5
EXPLICACIÓN SOBRE EL USO.....	6
EJEMPLOS DE USO.....	7
ENTORNO DE DESARROLLO.....	15
LENGUAJE DE PROGRAMACIÓN Y VERSIÓN.....	15
PRINCIPALES MÓDULOS Y SUS VERSIONES.....	16
OTRAS INSTRUCCIONES PERTINENTES.....	16
BIBLIOGRAFÍA.....	17

OBJETIVOS

- Objetivos a considerar para el proyecto.

Desarrollar es un programa que pueda obtener, crear y modificar información en el micro-sistema-de-archivos de la Facultad de Ingeniería, *FiUnamFS*.

- Listar los contenidos del directorio dentro de *FiUnamFS*.
- Copiar un archivo desde *FiUnamFS* hacia el sistema local.
- Copiar un archivo desde el sistema local hacia *FiUnamFS*.
- Eliminar archivos específicos almacenados en *FiUnamFS*.

Implementar un programa para desfragmentar *FiUnamFS*, realizando la reorganización de datos dentro del sistema de archivos existente, con el fin de mitigar la fragmentación externa. Asegurarse de realizar la desfragmentación sin crear un nuevo sistema de archivos ni copiar hacia uno nuevo.

INTRODUCCIÓN

Un sistema de archivos es una estructura lógica y organizada que permite almacenar, administrar y descargar datos a un dispositivo de almacenamiento como un disco duro, memoria flash, tarjeta de memoria u otro medio de almacenamiento. Básicamente, es la forma en que se organizan y almacenan los archivos en un dispositivo de almacenamiento para que el sistema operativo pueda acceder a ellos.

Un sistema de archivos define cómo se nombran, almacenan y organizan los archivos y directorios dentro de un dispositivo de almacenamiento. Aquí hay algunos componentes clave de un sistema de archivos:

Directorios: Estos directorios actúan como estructuras jerárquicas que permiten organizar los archivos de manera lógica. Además, proporcionan una forma tanto intuitiva como eficiente de acceder y gestionar múltiples archivos, facilitando así la navegación y la búsqueda dentro del sistema de archivos.

Archivos: Contienen diversos tipos de datos y están identificados por un nombre único en el sistema de archivos. Representan unidades individuales de información y pueden ser creados, modificados o eliminados según sea necesario.

Metadata: Proporciona detalles cruciales adicionales sobre los archivos en un sistema de archivos. Además de la fecha de creación, tamaño, permisos de acceso y ubicación física, los metadatos pueden incluir información como la fecha de modificación, el propietario del archivo, el tipo de archivo, los atributos específicos del archivo y, en sistemas más avanzados, información de codificación o compresión, entre otros detalles relevantes.

Tabla de asignación de archivos (File Allocation Table - FAT), sistema de asignación de archivos (File Allocation System - FAS) u otras estructuras similares: Estos componentes

son responsables de registrar la ubicación física de los archivos en el dispositivo de almacenamiento, lo que facilita su acceso y manipulación por parte del sistema operativo.

Estructuras de organización de datos: Existen diversas formas de organizar físicamente los datos en un dispositivo de almacenamiento, como los sistemas de archivos FAT, NTFS, exFAT (utilizados en sistemas Windows), ext4 (empleados en sistemas Linux), HFS+ (utilizados en sistemas macOS) y otros más. Cada uno de estos sistemas tiene su propio método para organizar y administrar los archivos almacenados en el dispositivo.

AUTORES

Responsabilidades y roles de los autores:

Desarrollo del código: Es responsabilidad de todos los autores desarrollar y redactar el código requerido para implementar las funciones necesarias dentro del proyecto.

Documentación adecuada: Los autores tienen la responsabilidad de documentar de manera minuciosa el código que desarrollan, así como de proveer comentarios claros y explicativos. Esto garantiza su comprensión y mantenimiento a lo largo del tiempo.

Revisión del código: Se establece un proceso de revisión entre autores para garantizar la calidad del código. Cada autor es responsable de revisar y analizar el código producido, proporcionando retroalimentación y haciendo correcciones si es necesario.

Pruebas y capturas de pantalla: Es imperativo que los autores realicen pruebas exhaustivamente del código implementado, incluyendo capturas de pantalla que documenten visualmente el comportamiento del programa en varios escenarios.

Contribuciones individuales de cada autor: Cada autor participa activamente en el desarrollo del código, la documentación, la revisión por pares y las pruebas. Las contribuciones específicas se definen en función de la experiencia y las fortalezas individuales de cada autor.

En el marco de las contribuciones individuales, Paola se encargará específicamente de los puntos que implican la documentación detallada del código y la corrección de errores, respectivamente. Mientras tanto, Jonathan asumirá la responsabilidad de los puntos centrados en las pruebas exhaustivas del código para garantizar su calidad y en la optimización del rendimiento del proyecto, respectivamente.

Por último, se considerará la experiencia y conocimientos adquiridos durante las clases en relación al tema por parte de los autores.

ESTRATEGIA

Comprensión de la especificación:

- Analizar detalladamente la especificación proporcionada para comprender los requisitos funcionales y no funcionales del sistema de archivos *FiUnamFS*.
- Identificar los casos de uso y las operaciones requeridas para el sistema de archivos.

Planificación y diseño:

- Establecer un plan de trabajo detallado que incluya las etapas de desarrollo, pruebas y documentación.
- Diseñar una arquitectura de software que cumpla con los requisitos de *FiUnamFS*.
- Definir la estructura de datos y algoritmos necesarios para implementar las operaciones requeridas (listar, copiar, eliminar archivos, desfragmentación).

Implementación:

- Desarrollar el código del sistema de archivos *FiUnamFS* siguiendo las especificaciones.
- Implementar las funciones para listar contenidos, copiar archivos dentro y fuera del sistema, eliminar archivos y realizar la desfragmentación interna del sistema.

Pruebas:

- Realizar pruebas exhaustivas para cada función y operación implementada.
- Verificar la integridad y el correcto funcionamiento del sistema mediante pruebas de unidad, integración y sistema.

Optimización y control de calidad:

- Optimizar el sistema para mejorar su rendimiento y eficiencia.
- Realizar revisiones de código y asegurarse de cumplir con los estándares de calidad.

Metodologías utilizadas en el desarrollo del proyecto:

- Metodología de desarrollo ágil: Dividir el proyecto en iteraciones pequeñas y manejables para adaptarse a posibles cambios en los requisitos y prioridades.
- Enfoque incremental: Implementar funcionalidades de manera progresiva y continua, integrando mejoras paso a paso.
- Control de versiones: Utilizar herramientas como Git para el control de versiones y colaboración en el desarrollo del código.
- Métodos de desarrollo colaborativo: Fomentar la colaboración entre los miembros del equipo para maximizar la eficiencia y calidad del trabajo.

REQUISITOS DE USO

Funcionalidades del programa:

- Listar archivos presentes en el sistema de archivos *FiUnamFS*.
- Copiar archivos desde *FiUnamFS* hacia el sistema actual.
- Copiar archivos desde el sistema actual hacia *FiUnamFS*.
- Eliminar archivos del sistema de archivos *FiUnamFS*.

Requisitos del entorno:

- El programa necesita acceder al archivo de imagen del sistema de archivos *FiUnamFS*, el cual debe estar presente y accesible en el directorio de trabajo bajo el nombre *"fiunamfs.img"*.

Normativas y estándares:

- Se requiere que las operaciones de lectura, escritura y manipulación de archivos se realicen de manera correcta y segura dentro del sistema de archivos *FiUnamFS*.

Recursos necesarios:

- Acceso con permisos de lectura y escritura al archivo de imagen *"fiunamfs.img"*.
- Precaución al manipular el archivo del sistema de archivos, ya que cambios incorrectos pueden corromper la estructura del sistema.

Documentación:

- Se recomienda proporcionar una guía de usuario o documentación que explique el uso del programa, sus funcionalidades, limitaciones y posibles riesgos al trabajar con *FiUnamFS*.

Mejoras potenciales:

- Mejorar la manipulación de archivos para garantizar la integridad de los datos.
- Implementar un manejo de errores más robusto para evitar fallos inesperados.
- Agregar comentarios al código para facilitar su comprensión y mantenimiento.

EXPLICACIÓN SOBRE EL USO

Instrucciones para utilizar el sistema *FiUnamFS*:

El propósito central del proyecto es la manipulación del sistema de archivos *FiUnamFS*. Para ponerlo en práctica, sigue estos pasos:

1. Clone el repositorio o descarga los archivos del proyecto.
2. Asegúrese de tener instalado Python y las librerías necesarias.
3. Ejecute el programa desde la terminal utilizando el comando ``python nombre_del_programa.py``.

Ventajas y beneficios:

El sistema ofrece varias ventajas significativas:

1. Gestión simplificada de archivos: Facilita la administración al permitir listar, copiar y eliminar archivos dentro de *FiUnamFS*.
2. Manipulación de archivos binarios: Proporciona funciones para trabajar con archivos binarios, permitiendo la copia desde y hacia *FiUnamFS*.

3. Optimización del sistema: Herramientas disponibles, como la desfragmentación, mejoran la organización y el rendimiento del sistema.

Consideraciones importantes:

1. Respaldo de datos: Se recomienda realizar copias de seguridad antes de modificar o eliminar archivos.
2. Operaciones con cuidado: La eliminación de archivos es irreversible, por lo que debes asegurarte de seleccionar el archivo correcto.
3. Gestión del espacio: Es importante monitorear el espacio disponible y considerar la desfragmentación para optimizar el almacenamiento.
4. Compatibilidad y versiones: Verifica la versión del programa y las librerías para garantizar un funcionamiento adecuado del sistema.

EJEMPLOS DE USO

Primero, realiza un análisis del *FIUNAMFS* para obtener información clave, incluyendo:

- Nombre del sistema de archivos.
- Versión del sistema.
- Etiqueta del volumen.
- Peso en bytes de la información.
- Distribución de clusters.

```
****Proyecto 3 Sistema de Archivos****  
  
Nombre del Archivo Principal: FiUnamFS  
Versión: 24.1  
Volumen de Etiqueta: No quiero etiqueta.  
Peso del cluster en bytes: 2048  
Número de clusters del directorio: 4  
Número de clusters totales: 2
```

A continuación se presentan las opciones disponibles que el usuario puede ingresar:

1. Listar archivos: Muestra el contenido del sistema de archivos.
2. Copiar archivo: Permite copiar un archivo hacia o desde el sistema *FIUNAMFS*.
3. Eliminar archivo: Elimina un archivo específico del sistema.
4. Desfragmentar: Optimiza la disposición de los archivos para mejorar el rendimiento.
5. Mostrar información del sistema: Despliega detalles como nombre, versión, etiqueta del volumen, peso en bytes y distribución de clusters del *FIUNAMFS*.
6. Salir: Cierra el programa o regresa al menú principal, dependiendo del contexto.

```
Opciones:

---1. Listar de contenido de FIUNAMFS---
---2. Copiar archivo de FIUNAMFS a PC---
---3. Copiar archivo de PC a FIUNAMFS---
---4. Eliminar archivo de FiUnamFS---
---5. Salir del Programa---
```

- Opción 1: Listar Contenidos del Archivo *FIUNAMFS*.



Al seleccionar esta opción, se desplegará una lista que contiene el contenido presente en el archivo *FIUNAMFS*.

```
***Ingresa una opción: 1

Archivo: README.org      Peso: 6 Cluster inicial 858927154
Archivo: logo.png        Peso: 21 Cluster inicial 858927154
Archivo: mensaje.jpg     Peso: 154 Cluster inicial 858927154
```

- Opción 2: Copiar Archivo desde *FIUNAMFS* a la PC.

Al seleccionar esta opción, se podrá realizar una copia de un archivo específico desde el sistema de archivos *FIUNAMFS* a la computadora personal. Antes de proceder con la copia, se mostrará una lista de archivos dentro del directorio actual en el *FIUNAMFS*.

	__pycache__	24/11/2023 14:10	Carpeta de archivos	
	bueno	24/11/2023 16:09	Archivo PY	21 KB
	hola	24/11/2023 16:08	Documento de tex...	0 KB
	holamundo	24/11/2023 16:15	Documento de tex...	0 KB
	fiunamfs	21/11/2023 13:49	Archivo IMG	1.440 KB

Se solicita al usuario ingresar el nombre del archivo que desea transferir al directorio especificado. Si el archivo se encuentra dentro del sistema, se procede a realizar su copia.


```
****Ingresa una opción: 2
Nombre del Archivo a copiar: logo.png

Archivo encontrado: logo.png
Espacio que ocupa: 170192 Cluster inicial: 21







Archivo copiado
```

En caso de no encontrar el archivo especificado, se imprimirá un mensaje indicando que no se ha hallado ningún archivo con ese nombre.

```
****Ingresa una opción: 2
Nombre del Archivo a copiar: sistop.txt

No se encontro ningun archivo o esta vacio
```






Y se mostrará en tiempo real el proceso de copiado hacia el directorio especificado, permitiendo observar el progreso y la transferencia del archivo seleccionado.

	__pycache__	24/11/2023 14:10	Carpeta de archivos	
	bueno	24/11/2023 16:09	Archivo PY	21 KB
	hola	24/11/2023 16:08	Documento de tex...	0 KB
	holamundo	24/11/2023 16:15	Documento de tex...	0 KB
	logo	24/11/2023 14:15	Archivo PNG	167 KB
	fiunamfs	21/11/2023 13:49	Archivo IMG	1.440 KB

- Opción 3: Copiar desde la PC al Archivo *FIUNAMFS*.

Esta opción permite realizar una copia del archivo "*holamundo.txt*" desde la computadora personal al sistema de archivos *FIUNAMFS*.

Al seleccionar esta opción, el programa procederá a copiar el archivo "*holamundo.txt*" desde la computadora personal hacia el sistema *FIUNAMFS*, asegurándose de que el archivo se agregue correctamente al sistema de archivos especificado.

	__pycache__	24/11/2023 14:10	Carpeta de archivos	
	bueno	24/11/2023 16:09	Archivo PY	21 KB
	hola	24/11/2023 16:08	Documento de tex...	0 KB
	holamundo	24/11/2023 16:15	Documento de tex...	0 KB
	fiunamfs	21/11/2023 13:49	Archivo IMG	1.440 KB

Esta función es útil para transferir archivos desde la computadora personal al sistema *FIUNAMFS*, facilitando la incorporación de nuevos elementos al sistema de archivos para su posterior gestión y uso.

Y este archivo contiene:

```

Archivo  Editar  Ver
|hoal mundo!!!

```

Entonces, al seleccionar la opción 3 en el programa, procederemos a copiar el archivo "*holamundo.txt*" desde nuestra computadora al sistema de archivos *FIUNAMFS*.

```

****Ingresa una opción: 3

Ingresa nombre del archivo: holamundo.txt

Tamaño: 15  caracteres

Archivo Copiado

```

La copia del archivo desde nuestra computadora al sistema de archivos *FIUNAMFS* se ha completado con éxito.

```
****Ingresa una opción: 1
```

```
Archivo: README.org      Peso: 6 Cluster inicial 858927154
```

```
Archivo: holamundo.txt   Peso: 0 Cluster inicial 858927154
```

```
Archivo: logo.png        Peso: 21 Cluster inicial 858927154
```

```
Archivo: mensaje.jpg     Peso: 154 Cluster inicial 858927154
```

Y observamos en el archivo *FIUNAMFS*:

```
    /-holamundo.txt
0      /.....
0      -mensaje.jpg  0'0 š
0      /.....
0      /.....
0      /.....
0      /.....
0      /.....
0      /.....
```

Y procedemos a verificar si el archivo se ha copiado exitosamente al sistema.

Por otro lado, si se solicita copiar un archivo que no existe:

```
****Ingresa una opción: 3

Ingresa nombre del archivo: unam.txt

Archivo no encontrado :(
```

El programa indicará que el archivo denominado "*unam.txt*" no existe.

- Opción 4: Eliminar Archivo en *FIUNAMFS* (si existe).

Esta opción permite llevar a cabo la eliminación de un archivo en el sistema de archivos *FIUNAMFS*, en caso de que dicho archivo exista. Si se encuentra en el sistema, el programa procederá a eliminarlo de manera definitiva.

```
README.org
holamundo.txt
logo.png
.....
.....
mensaje.jpg

Archivo eliminado... :(
```

Muestra los archivos y el archivo a eliminar “*mensaje.jpg*”.

```
***Ingresa una opción: 1

Archivo: README.org      Peso: 6 Cluster inicial 858927154

Archivo: holamundo.txt  Peso: 0 Cluster inicial 858927154

Archivo: logo.png       Peso: 21 Cluster inicial 858927154

Archivo: /.....      Peso: 154 Cluster inicial 858927154
```

Cuando el usuario solicita listar el contenido y un archivo no existe, dicho archivo no aparecerá en el listado proporcionado por el programa. En su lugar, el programa desplegará la lista de archivos existentes en el sistema *FIUNAMFS* para que el usuario pueda visualizar los archivos presentes en el sistema de archivos.

Archivo a Eliminar no encontrado :(

- Opción 5: Salir del Programa.

Esta opción está disponible únicamente para finalizar la ejecución del programa y salir de él. Al seleccionar esta opción, el programa se cerrará, permitiendo al usuario salir del programa de manera controlada y finalizar su ejecución.

```
****Ingresa una opción: 5

Hasta luego!!! Este es el Proyecto 3 de Sistemas Operativos
Realizado por:
+Jonathan Emmanuel Hernandez Ortiz+
+Paola Celina de Jesus Perez Avin+

█
```

El programa espera durante tres segundos para que el usuario pueda revisar la información más reciente del archivo y luego finaliza su ejecución automáticamente.

```
****Ingresa una opción: 5

Hasta luego!!! Este es el Proyecto 3 de Sistemas Operativos
Realizado por:
+Jonathan Emmanuel Hernandez Ortiz+
+Paola Celina de Jesus Perez Avin+

PS C:\Users\ingjo\OneDrive\Documentos\Semestre 2024-1\Sistemas Op
erativos\Proyecto3_Archivos\buenop3so> █
```

- Ejemplo 1: Listar Archivos en *FiUnamFS*.

Este ejemplo demuestra el proceso para mostrar todos los archivos presentes en el sistema de archivos *FiUnamFS*.

- Ejemplo 2: Copiar Archivo desde *FiUnamFS* al Sistema Actual.

Permite al usuario realizar la copia de un archivo desde el sistema de archivos *FiUnamFS* hacia el directorio actual del sistema en uso.

- Ejemplo 3: Copiar Archivo del Sistema Actual a *FiUnamFS*.

En este caso, se lleva a cabo la copia de un archivo desde el sistema actual hacia el sistema de archivos *FiUnamFS*.

- Ejemplo 4: Eliminar un Archivo de *FiUnamFS*.

Esta opción habilita la eliminación de un archivo del sistema de archivos *FiUnamFS*.

Notas Adicionales:

- Estos ejemplos asumen la presencia del archivo *"fiunamfs.img"* en el directorio de trabajo y la disponibilidad de permisos de lectura y escritura.
- Se requerirá ingresar los nombres de archivo y seguir las instrucciones provistas por el programa para completar las acciones mencionadas.

ENTORNO DE DESARROLLO

El proyecto está diseñado para funcionar en entornos de desarrollo de sistemas operativos basados en Unix, como Linux o macOS. Además, es compatible con sistemas Windows si se ejecuta utilizando herramientas como Git Bash o WSL (Windows Subsystem for Linux).

- Sistemas Operativos Compatibles: Linux, macOS, Windows (mediante Git Bash, WSL u otros emuladores de terminal Unix).
- Requisitos de Hardware: No se requieren especificaciones de hardware especiales. Se recomienda un sistema con al menos 2 GB de RAM para un rendimiento óptimo.
- Software Requerido: Python 3.x y las librerías estándar de Python son necesarias para la ejecución del programa.

LENGUAJE DE PROGRAMACIÓN Y VERSIÓN

El código proporcionado está elaborado en el lenguaje de programación Python. ¡Para ejecutar correctamente el programa, se requiere tener instalada una versión específica de Python, preferiblemente la versión 3.x (como Python 3.7, 3.8, 3.9, etc.)! El código no depende de características específicas de una versión de Python en particular, por lo que debería ser compatible con la mayoría de las versiones 3.x.

No se incluyen herramientas o frameworks específicos asociados al proyecto en el código. Sin embargo, se hace uso de algunas librerías estándar de Python:

- *os*: Para funcionalidades relacionadas con el sistema operativo.
- *time*: Para operaciones relacionadas con el tiempo y fechas.
- *struct*: Para realizar operaciones de packing y unpacking de datos binarios.
- *shutil*: *Desfragmentar archivos en un sistema de archivos puede ser un proceso complejo y depende en gran medida de la estructura interna del sistema de archivos específico.*

El código hace uso de estas librerías estándar para manipular archivos y efectuar operaciones dentro del sistema de archivos FiUnamFS.

Resumiendo para mayor claridad:

- Lenguaje de programación: Python.
- Versión requerida: Se recomienda Python 3.x (por ejemplo, 3.7, 3.8, 3.9, etc.).

- Herramientas o frameworks asociados: No se aplica la utilización de frameworks específicos en el código, solo la utilización de bibliotecas estándar de Python como *os*, *time*, *struct*, entre otras.

PRINCIPALES MÓDULOS Y SUS VERSIONES

- Lista de los módulos o bibliotecas adicionales necesarias para el funcionamiento del programa.
- Versiones específicas de cada módulo requerido para evitar conflictos o incompatibilidades.
- Detalles sobre cómo instalar o acceder a estos módulos, ya sea a través de gestores de paquetes o de otros métodos.

El código proporcionado hace uso de las siguientes librerías o módulos estándar:

- *os*: Para funcionalidades relacionadas con el sistema operativo.
- *time*: Para operaciones relacionadas con el tiempo y fechas.
- *struct*: Para realizar operaciones de packing y unpacking de datos binarios.

Estos módulos son parte de la instalación estándar de Python y no requieren instalación adicional, ya que vienen incluidos con la instalación básica de Python.

Si el entorno donde se ejecuta el código no tiene alguno de estos módulos estándar disponibles, puede ser indicativo de un problema en la instalación de Python.

Sin embargo, si el código necesitara de módulos externos adicionales (no evidenciados en el fragmento proporcionado), se requeriría mencionar específicamente estos módulos y sus versiones para asegurar la compatibilidad. Para instalar módulos adicionales en Python, se suelen utilizar gestores de paquetes como *'pip'* o *'conda'*. Por ejemplo:

- Para instalar un paquete con *'pip'*, se usaría el siguiente comando en la terminal:
'pip install nombre_del_paquete'.
- Para *'conda'*, se usaría: *'conda install nombre_del_paquete'*.

Dichos comandos deben ejecutarse en el terminal o la línea de comandos del sistema, y requerirán acceso a internet para descargar e instalar los módulos externos.

Si se necesita información sobre módulos adicionales específicos para este código, se debería revisar el código en su totalidad para identificar las dependencias adicionales y sus versiones correspondientes.

OTRAS INSTRUCCIONES PERTINENTES

Integración con Otros Sistemas o Aplicaciones: El programa no está diseñado específicamente para integrarse con otros sistemas o aplicaciones externas. Sin embargo,

puede ser adaptado para trabajar como una herramienta dentro de un sistema más grande o como parte de un flujo de trabajo más complejo.

Consideraciones de Seguridad y Mejores Prácticas:

- **Respaldo de Datos:** Es recomendable realizar copias de seguridad periódicas de los archivos antes de manipularlos con este programa, ya que cualquier operación de modificación o eliminación podría ser irreversible.
- **Permisos de Archivos:** Asegúrese de tener los permisos necesarios para acceder y modificar los archivos en el sistema de archivos objetivo. Ejecute el programa con los permisos adecuados según sea necesario para evitar errores de acceso.
- **Validación de Entrada:** Si este programa se integra con otros sistemas o se utiliza para manejar datos de usuarios, se recomienda implementar validaciones de entrada para prevenir posibles vulnerabilidades de seguridad, como la inyección de código malicioso.
- **Seguridad de la Información:** Evite compartir archivos o información sensible, especialmente al trabajar con sistemas de archivos en un entorno compartido.

BIBLIOGRAFÍA

Recopilación de las fuentes empleadas para el desarrollo de este proyecto.

- Laboratories, L. S. (2023, 25 abril). *Los fundamentos de los sistemas de archivos*. UFS Explorer. <https://www.ufsexplorer.com/es/articles/file-systems-basics/>.
- Ripoll, J. A. (2022, 1 noviembre). *¿Qué es un sistema de archivos y qué tipos existen?* Jaracoder. <https://jaracoder.com/que-es-un-sistema-de-archivos-y-que-tipos-existen/>.
- Team, K. (2023, 6 marzo). *¿Qué es un sistema de ficheros? | KeepCoding Bootcamps*. KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-un-sistema-de-archivos/>.