



**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**

**FACULTAD DE INGENIERÍA**

**Sistemas Operativos**

**Prof. Gunnar Wolf**

**Proyecto 2:  
(Micro) sistema de archivos multihilos.**

**Alumno:  
Maya Torres Bruno**

**Número de cuenta:  
320313338**

**Semestre:  
2026-1**

**Fecha de entrega:  
20/11/2025**



## 1. Descripción General

El objetivo de este proyecto fue desarrollar un programa capaz de interpretar y manipular un sistema de archivos propietario denominado FiUnamFS. El software permite realizar operaciones de lectura, escritura, extracción y eliminación de archivos sobre una imagen de disco (.img) simulada, respetando estrictamente las especificaciones de estructura binaria, Little Endian y geometría del disco.

## 2. Herramientas y Lenguaje de Implementación

Lenguaje: Python 3.x.

Justificación: Se eligió Python por su robustez en el manejo de estructuras binarias a través del módulo struct y su soporte nativo para programación concurrente.

### Librerías Estándar Utilizadas:

**os:** Para manejo de rutas y verificación de existencia de archivos.

**struct:** Para empaquetar y desempaquetar bytes.

**threading:** Para la implementación de hilos de ejecución.

**queue:** Para la sincronización y comunicación segura entre hilos.

## 3. Estrategia de Solución: Arquitectura Concurrente

Para cumplir con el requerimiento de sincronización de procesos, se implementó el programa de tal forma que este no opera de forma secuencial simple, sino que divide la carga en dos hilos de ejecución simultáneos:

### Hilo Principal (Interfaz de Usuario):

Se encarga de mostrar el menú al usuario y capturar los datos de entrada.

No accede al disco directamente. En su lugar, empaqueta la solicitud (comando y parámetros) y la coloca en una Cola de Órdenes (cola\_ordenes).

Queda a la espera de una respuesta en la Cola de Respuestas (cola\_respuestas), bloqueando la interfaz hasta que la operación termine.

### Hilo Secundario:

Se ejecuta en segundo plano mediante un bucle infinito.

Monitorea constantemente la cola\_ordenes. Cuando recibe una tarea, abre el archivo .img, realiza las operaciones y cierra el archivo.



---

Al finalizar, deposita el resultado (éxito o error) en la cola\_respuestas.

**Mecanismo de Sincronización:** El uso de objetos Queue garantiza la exclusión mutua implícita, evitando condiciones de carrera (race conditions) al acceder a los recursos compartidos, asegurando que las operaciones de lectura/escritura en el disco sean atómicas y seguras.

#### 4. Instrucciones de Ejecución

Para ejecutar el programa correctamente, hay que seguir estos pasos:

**Requisitos Previos:** Tener instalado Python 3.

**Archivos necesarios:** Asegúrese de que el script .py y el archivo de imagen fiunamfs.img se encuentren en la misma carpeta.

Comando: Abra una terminal en la carpeta del proyecto y ejecute:

```
python proyecto2BMT.py
```

#### Uso:

Seleccione una opción numérica del menú.

Para copiar archivos hacia el sistema (PUT), asegúrese de escribir la ruta correcta del archivo en su computadora.

Para copiar archivos desde el sistema (GET), el archivo se guardará en la carpeta actual.

#### 5. Evidencia de Funcionamiento

##### Opción 1: Información del superbloque

Esta opción realiza una lectura directa del Cluster 0 (los primeros 1024 bytes del disco). Su función principal es la validación de la integridad del sistema de archivos. El programa extrae los primeros bytes para verificar que la etiqueta del sistema corresponda a la cadena FiUnamFS y que la versión sea exactamente la 26-1



```
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 1
[Espera] Procesando en hilo secundario...
✅ ÉXITO: Sistema detectado: FiUnamFS Versión 26-1
```

### Opción 2: Listado de archivos.

Realiza un barrido secuencial y completo del área reservada para el directorio. El programa itera leyendo bloques de 64 bytes y valida el primer byte de control para filtrar únicamente los archivos activos. Para cada entrada válida, desempaqueta la estructura binaria interpretando los enteros en Little Endian, presentando al usuario una tabla detallada con el nombre del archivo, su peso exacto en bytes y el número de cluster donde inician sus datos.

```
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 2
[Espera] Procesando en hilo secundario...
✅ ÉXITO: ['README.org      | 31104 bytes | Cluster 5', 'logo.png          | 120857 bytes | Cluster 36', 'saludo.jpg       | 95307 bytes | Cluster 155']
=====
README.org      | 31104 bytes | Cluster 5
logo.png        | 120857 bytes | Cluster 36
saludo.jpg      | 95307 bytes | Cluster 155
=====
```

### Opción 3: Copiar HACIA tu PC (Get)

Permite la extracción de archivos desde el sistema FiUnamFS hacia el sistema de archivos del sistema operativo anfitrión. El usuario debe ingresar el nombre exacto del archivo alojado en la imagen .img y el nombre con el que se desea guardarlo en el sistema. El programa localiza el cluster inicial y el tamaño en el directorio, lee la cadena de bytes correspondiente desde el área de datos y reconstruye el archivo en la computadora local.






Ejemplo del funcionamiento:



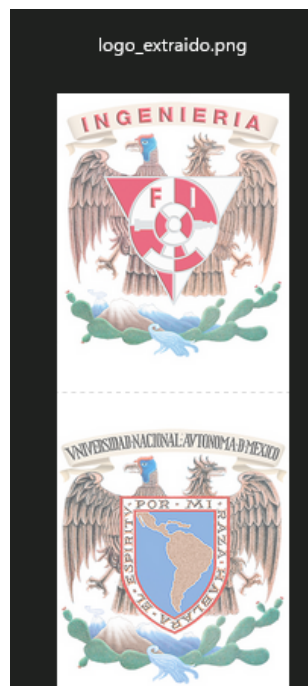
```
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 3
Nombre archivo en FiUnamFS: saludo.jpg
Nombre destino en tu PC: saludo_extraido.jpg
[Espera] Procesando en hilo secundario...
✅ ÉXITO: Archivo saludo.jpg extraído exitosamente.
```

Selecciono la opción tres e ingreso el nombre exacto del archivo que deseo extraer. Posteriormente ingreso el nombre con el que quiero guardarlo en mi sistema.

Si realizó ese procedimiento con los tres archivos que se encuentran en FiUnamFS obtengo lo siguiente:

 fiunamfs.img	21/11/2025 12:47 a. m.	Archivo de image...	1,440 KB
 logo_extraido.png	21/11/2025 01:21 a. m.	Archivo PNG	119 KB
 proyecto2BMT.py	21/11/2025 12:46 a. m.	Python File	12 KB
 readme.org	21/11/2025 01:14 a. m.	Archivo ORG	31 KB
 saludo_extraido.jpg	21/11/2025 01:17 a. m.	Archivo JPG	94 KB

Los archivos se extraen y se guardan en la misma carpeta en la que se encuentra la imagen y mi programa.





#### Opción 4: Copiar HACIA FiUnamFS (Put)

Realiza la operación de escritura e importación de archivos externos hacia la imagen. Esta función implementa la lógica de asignación de espacio:

Verifica la existencia del archivo origen en la PC.

Calcula el espacio libre necesario en el área de datos (buscando el último cluster ocupado y añadiendo al final, o append).

Busca una entrada libre en el directorio (marcada con - o /).

Escribe los datos binarios en los clusters correspondientes y actualiza los metadatos en el directorio.

Ejemplo: Seleccione la opción 4 e ingreso la ruta del archivo que quiero copiar. Posteriormente seleccione la opción 2 para ver la lista de archivos y de esta forma puedo ver que se copió correctamente dentro de FiunamFS

```
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 4
Ruta archivo en tu PC: C:\Users\bmaya\OneDrive\Escritorio\Pyramidsss\proy2SISTOP\foto.png
Nombre para guardar en FiUnamFS: foto.png
[Espera] Procesando en hilo secundario...
✅ ÉXITO: Guardado en Cluster 249

=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 2
[Espera] Procesando en hilo secundario...
✅ ÉXITO: ['README.org      | 31104 bytes | Cluster 5', 'foto.png          | 1880 bytes | Cluster 249', 'logo.png          | 120857 bytes | Cluster 36', 'saludo.jpg        | 95307 bytes | Cluster 155']

=====
README.org      | 31104 bytes | Cluster 5
foto.png        | 1880 bytes | Cluster 249
logo.png        | 120857 bytes | Cluster 36
saludo.jpg      | 95307 bytes | Cluster 155
=====
```

#### Opción 5: Eliminar Archivo

Ejecuta el borrado de un archivo dentro del sistema FiUnamFS. El programa busca el nombre del archivo en el directorio y, si lo encuentra, no elimina los datos físicos, sino que marca la entrada como "disponible" relleno el nombre con un patrón de puntos (.....), liberando así esa entrada para futuras escrituras.

Ejemplo: Se realiza la eliminación del mismo archivo (foto.png) que había copiado anteriormente. Para ello. selecciono la opción 5 e ingreso el nombre exacto del archivo que deseo eliminar. Puedo ver que la ejecución fué exitosa ya que al seleccionar la opción de listado de archivos este no aparece.



```
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 5
Nombre archivo a borrar: foto.png
[Espera] Procesando en hilo secundario...
✓ ÉXITO: Archivo foto.png eliminado.
=====
1. Info Superbloque
2. Listar Archivos
3. Copiar HACIA tu PC (Get)
4. Copiar HACIA FiUnamFS (Put)
5. Eliminar Archivo
6. Salir
Selecciona una opcion: 2
[Espera] Procesando en hilo secundario...
✓ ÉXITO: ['README.org' | 31104 bytes | Cluster 5', 'logo.png' | 120857 bytes | Cluster 36', 'saludo.jpg' | 95307 bytes | Cluster 155']
]
-----
README.org | 31104 bytes | Cluster 5
logo.png | 120857 bytes | Cluster 36
saludo.jpg | 95307 bytes | Cluster 155
-----
```

### Opción 6: Salir.

Al seleccionar esta opción concluye la ejecución del programa.