



**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**  
**FACULTAD DE INGENIERÍA**

**División de Ingeniería Eléctrica**

**Sistemas Operativos**

**Proyecto 2**

**Asignatura: Sistemas Operativos**

**Grupo: 6**

**Docente: Ing. Gunnar Eyal Wolf Iszaevich**

- Jiménez Elizálde Josué
- Medina Guzmán Santiago

**Fecha de entrega: 07/11/2024**

**Semestre: 2025-1**

## Requisitos iniciales

1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. El programa que desarrollen debe contar, por lo menos, dos hilos de ejecución, operando *concurrentemente*, y que se *comuniquen su estado* mediante mecanismos de sincronización.

## Descripción general

Es un programa que puede obtener, crear y modificar información en el micro-sistema-de-archivos que desarrollo el Ing. **Gunnar Eyal Wolf Iszaevich** para la Facultad de Ingeniería, FiUnamFS. Esto lo hace con uso de 4 módulos para repartir el código para no hacerse pesado en leerlo.

## Módulos

Filesystem.py

Aquí está la lógica de clústers que involucra los tamaños del sistema, su escritura, su lectura, etc.

```
class FiUnamFS:
    def __init__(self):
        self.image = open(IMG_FILE, 'r+b')
        self.allocation_table = self.read_allocation_table()
```

```
def read_cluster(self, cluster_num):
```

```
def write_cluster(self, cluster_num, data):
```

```
def read_superblock(self):
```

```
def read_allocation_table(self):
```

```
def get_next_cluster(self, current_cluster):
```

```
def close(self):
```

Directory.py

Aquí esta la lógica de procesos de archivos que es lista de archivos, añadir archivos, eliminación y todos los que se encuentran en requisitos iniciales.

```
class Directory:
    def __init__(self, fs):
        self.fs = fs

    def list_files(self):

    def find_empty_entry(self):

    def add_file(self, filename, filesize, cluster_num):

    def delete_file(self, filename):
```

File\_operations.py

Aquí está la lógica de recuperación de archivos desde el micro sistemas de archivos.

```
def __init__(self, fs, directory):
    self.fs = fs
    self.directory = directory

def copy_from_fs(self, filename):

def copy_to_fs(self, source_file, dest_name):
```

Main.py

Aquí ya se encuentra toda la lógica principal del sistema, cómo el menú, imports de los otros módulos y lectura del directorio.

```
def menu():
    print("Opciones:")
    print("1. Listar archivos")
    print("2. Copiar archivo desde FiUnamFS")
    print("3. Copiar archivo hacia FiUnamFS")
    print("4. Eliminar archivo en FiUnamFS")
    print("5. Salir")
    print("6. Diagnóstico del Directorio")
```

## Conclusión

A la hora de hacer el proyecto se presentaron complicaciones al momento de recuperar archivos del micro sistema, ya que estos se corrompían, también no implementamos utilización de hilos.

Fuera de esos problemas, el programa puede listar los elementos del micro sistema, los recupera (pero corruptos), los copia hacia FiUnamFs y los elimina de forma correcta.