

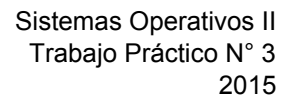
## Actividad a desarrollar

En base a las estructura **Nodol**, **NodoApuntador** y **NodoDatos**, desarrollar una clase **SistemaArchivos** que permita ejecutar acciones, haciendo uso de las estructuras, simulando el uso de un sistemas de archivos. Dichas acciones se leerán de un archivo de texto.

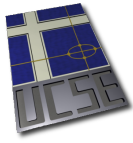
Cada archivo del Sistema de Archivos debe estar asociado a esta estructura **Nodol**, y los datos se deben colocar en cada uno de los **NodosDatos**, utilizando los **NodoApuntador** de apuntadores cuando es necesario escribir mayor cantidad de información.

## Estructuras

Nodol	NodoApuntador	NodoDatos
<div><div>Metadatos</div><div>Apuntadores Directos</div><div>Apuntadores Indirectos</div><div><div>name</div><div>size</div><div>user</div><div>group</div><div>mode</div><div>ctime</div><div>mtime</div><div>atime</div><div>b1</div><div>b2</div><div>b3</div><div>b4</div><div>b5</div><div>b6</div><div>b7</div><div>b8</div><div>b9</div><div>b10</div><div>L1</div><div>L2</div><div>L3</div></div></div>	<div><div>Apuntadores Directos</div><div><div>b1</div><div>b2</div><div>b3</div><div>b4</div><div>b5</div><div>b6</div><div>b7</div><div>b8</div><div>b9</div><div>b10</div></div></div>	<div><div>Datos</div></div>

[illegible]

- **MD** Crear Directorio
- **RD** Eliminar Directorio
- **CD** Cambiar Directorio de trabajo
- **CF** Crear Archivo
- **DF** Eliminar Archivo
- **WF** Escribir Archivo
- **DU** Definir Usuario (se aplica en modo global)
- **CG** Definir Grupo (se aplica en modo global)
- **AG** Asignar Grupo a un archivo/directorio



## Uso de las acciones (estructura del archivo de acciones)

```
MD|NombreDirectorio
RD|NombreDirectorio
CD|NombreDirectorio
CF|NombreArchivo
DF|NombreArchivo
WF|NombreArchivo|Datos (cadena de caracteres)
DU|NombreUsuario
CG|NombreGrupo
AG|NombreArchivo|NombreGrupo
```

## Permisos

Para los archivos y directorios que se crean, por defecto se le deben asignar los siguientes permisos: 755 (-rwx r-x r-x). Para los directorios, los permisos inician con una "d".

Cuando se va a realizar alguna acción, es necesario que se validen los permisos del archivo/directorio y se correspondan con lo requerido para el acceso. Por ejemplo, si un archivo quiere escribirse en una carpeta, la carpeta debe permitirle al usuario la escritura, si un usuario quiere leer un archivo, este tiene que tener, mínimo, permiso de lectura sobre el archivo.

En el control de los permisos, se debe contemplar el grupo activo para corroborar que el usuario pueda acceder mediante ese nivel de permiso.

## Acciones de la clase que no se ejecutan desde el archivo

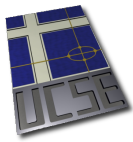
### Leer Archivo

Tras la carga de los datos desde el archivo, la clase debe permitir leer el contenido de un archivo mediante una función **LeerArchivo**, la cual recibe como parámetro la ruta al archivo a leer, la posición inicial de lectura y la cantidad de caracteres a leer.

Esta función debe retornar una cadena con los datos solicitados.

### Obtener metadatos

Mediante la función **ObtenerInfo**, indicando la ruta del archivo, se obtienen los metadatos del archivo, almacenados en el nodo-i cabecera.



### Mostrar Árbol de Directorios

Se debe desarrollar una función **get\_tree**, la cual recibe la ruta a un directorio y debe devolver un diccionario, con los encadenamientos de los directorios en el sistema de archivos. Por ejemplo:

```
{
  'dir1': {},
  'dir2': {
    'dir21': {},
    'dir22': {}
  },
  'dir3': {
    'dir31': {}
  }
}
```

## Forma de trabajo y evaluación

- El trabajo debe realizarse en grupos de no más de 3 personas.
- Se puede realizar en cualquier lenguaje de programación, entregando la documentación pertinente para la instalación y configuración del ambiente de pruebas a la cátedra, el cual debe permitir el seguimiento del código de forma simple y completa.

## Fecha y Forma de entrega

- El trabajo se debe enviar a la cátedra vía correo electrónico.
- **El Trabajo Práctico se puede entregar hasta el 02/07/2015 a las 18:00hs.**