Tabla de contenido

[Memoria práctica Sistemas Distribuidos 2](#_Toc532208707)

[El proyecto: 2](#_Toc532208708)

[Realización de los elementos: 2](#_Toc532208709)

[El nodo: 2](#_Toc532208710)

[El árbol: 2](#_Toc532208711)

[La terminal: 3](#_Toc532208712)

[Diskmanager: 3](#_Toc532208713)

[Disk: 3](#_Toc532208714)

[Planteamiento de la práctica: 3](#_Toc532208715)

[Fase de construcción: 3](#_Toc532208716)

[Realización de la práctica: 3](#_Toc532208717)

[Uso de Git: 3](#_Toc532208718)

[QtCreator: 4](#_Toc532208719)

[Fase de pruebas: 4](#_Toc532208720)

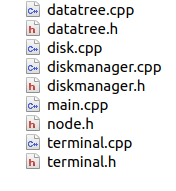
[Documentación final: 4](#_Toc532208721)

# Memoria práctica Sistemas Distribuidos

A la hora de realizar esta práctica, tuvimos en consideración varios factores que fuimos cambiando a lo largo de la práctica. Sobre todo con métodos que nos funcionaban al principio y que más tarde habría que darles una vuelta al ir añadiendo nuevas funcionalidades.

# El proyecto:

Nuestro proyecto de manera actual cuenta con los siguientes archivos:



Así como distintos archivos que se pueden ir creando y demás, como por ejemplo, archivos que subimos al disco o que borramos de él.

Nuestro proyecto es lo que podría decirse un árbol de directorios con ficheros. Un disco duro. Un disco de cualquier ordenador, que tiene ficheros, directorios, se pueden borrar y añadir ficheros, y se puede navegar por el mismo árbol (como cd, ls/dir (según el SO).

# Realización de los elementos:

A pesar de estar todo debidamente documentado, vamos a proceder a hacer una explicación breve de cada elemento que contiene nuestra práctica.

## El nodo:

El nodo de nuestro proyecto va a ser una estructura que va a contener toda la información necesaria de CADA nodo del árbol. Estos datos son, por ejemplo, su id, su nombre, el puntero a su padre… Cada vez que se cree un nodo, se inicializarán todos los parámetros necesarios.

## El árbol:

Va a ser el árbol de nuestro programa. Es el que va a almacenar todos los nodos, direcciones de hijos y demás, y del que pende la principal usabilidad de la práctica. Tendrá parámetros como el puntero al nodo root, la dirección total y demás, y luego las funciones del árbol.

Estas funciones son, por ejemplo, cargar el árbol o guardarlo, añadir un nodo y demás.

## La terminal:

La clase terminal va a ser la que gestione todos los inputs y outputs del programa. En este caso, cada vez que queramos hacer algo con el árbol, va a llamar a nodo, o árbol.

Será la que tenga todas las funciones de encontrar rutas, directorios, nodos, y así mismo, creación de directorios, de ficheros y un gran etc (como cd, mkdir, las funcionalidades del árbol).

## Diskmanager:

El manejador de disco va a ser el que controle todo lo relacionado con el espacio de discos y la repartición del mismo a la hora de subir o leer o eliminar información.

Como parámetros tendrá por ejemplo, el número de bloques y el espacio que hay en disco, y funciones, leer fichero o settear el número de bloques.

## Disk:

Va a ser un cpp que sea cada disco. En él vamos a poder escribir o leer. Tendremos N instancias de disk, siendo N el número de discos que tendremos en total.

# Planteamiento de la práctica:

A la hora de plantearnos esta práctica desde el principio, ya sabíamos que más adelante se tendría que implementar con MPI. Es por eso que pensamos en hacer los métodos de manera clara por si había que cambiar ciertos aspectos en el futuro y así tener más facilidad a la hora de entender y cambiar el código.

## Fase de construcción:

Antes de hacer cualquier cosa, nos planteamos ciertos diseños en papel (también llamado “construcción”), diagramas con flechas, para ver qué conectaba con qué, qué clases tenía sentido que tuvieran qué métodos y no otros, atributos y demás. Esto nos sería útil también más tarde con MPI para ver mejor y saber quien se comunica con quien, y cuándo se deben ir intercambiando los mensajes.

Siempre con los diagramas a mano, comentarnos las cosas para programarlas era más sencillo al tenerlo de manera más visual y más rápidamente accesible (en vez de “el método Y de X clase”).

# Realización de la práctica:

A la hora de realizar la práctica, íbamos cogiendo el código de la anterior, para poder ir haciendo un proyecto iterativo e incremental, mejorando el resultado de cada nueva implementación.

En cuanto a métodos que sí nos han comentado que eran “curiosos”, son los métodos recursivos que hemos realizado para poder obtener, por ejemplo, un directorio y demás. Tenemos una función principal, por ejemplo, Encontrar Directorio, que esta llama a una función Encontrar Directorio Recursivamente. Esto favorece a la mejora y optimización del código.

## Uso de Git:

Hemos usado git para ir subiendo cada actualización al repositorio y poder ir trabajando de manera remota sin problemas. Si por algún motivo teníamos que cambiar de máquina virtual o lo que fuera, era tan simple como descargar el proyecto desde git. Con cada commit, comentábamos qué habíamos realizado para que nos fuera más fácil saber la próxima vez, dónde nos habíamos quedado.

## QtCreator:

El ide utilizado ha sido QtCreator, que es el que se nos ha enseñado en clase, y hemos salido bastante contentos con el mismo. Esta práctica se ha tenido que realizar en Linux ya que queríamos utilizar las opciones del propio sistema operativo para la parte del árbol local como “ls” y demás. De esta manera, se podían implementar de manera directa. Más tarde, Linux ha sido la opción adecuada al utilizar mpi.

# Fase de pruebas:

Para la fase de pruebas de la práctica y ver que todo funcionaba, o ver qué cosas no deberían funcionar, nos hemos basado en las correcciones de prácticas anteriores del profesor, así como dejarle la práctica a algún compañero para que intentara “romperla” y poder añadir gestión de errores sobre esas cosas.

Una cosa que sí hemos realizado es no dejar todas las gestiones de errores para el final, si no a medida que implementábamos nuevas funcionalidades, si veíamos que podía generar algún tipo de error, gestionarlo de manera sencilla y rápida en el momento, para ahorrarnos quebraderos de cabeza más tarde.

# Documentación final:

Para finalizar, una vez teníamos la práctica hecha, hemos procedido a documentar toda la práctica con Doxygen en el formato de Qt. Esto lo hemos hecho para poder tener una documentación en forma de página web sencilla y muy intuitiva, tanto para nosotros como para futuras personas que pudieran estar interesadas en ver la documentación de la práctica.

Encontramos Doxygen algo útil y estructurado que es sobre todo útil en proyectos grandes, para poder moverte por el código del programa, sin meterte literalmente en el código del programa.