

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
MANEJO E IMPLEMENTACIÓN DE ARCHIVOS A-, A+
Ing. Oscar Paz Campos Aux. Luis Vargas
Ing. Alvaro Díaz Aux. Luis Pablo Hernández



ENUNCIADO PROYECTO 1 - SISTEMA DE ARCHIVOS LWH

Introducción

El curso de Manejo e Implementación de Archivos se busca que el estudiante aprenda los conceptos sobre la administración de archivos, tanto en hardware como software, sistemas de archivos, particiones, entre otros conceptos, para después introducirse en las bases de datos. El sistema de archivos a implementar utiliza algunos aspectos similares a los sistemas de archivos de Windows y GNU/Linux. Aplicando los conceptos adquiridos en el curso y laboratorio de Manejo e Implementación de Archivos sobre estos temas.

Objetivos

- Aprender a administrar archivos y escribir estructuras en Go
- Comprender el sistema de archivos LWH
- Aplicar el formateo rápido y completo en una partición
- Crear una aplicación de comandos
- Aplicar la teoría de particiones
- Utilizar GraphViz para mostrar reportes
- Restringir y administrar el acceso a los archivos y carpetas en LWH por medio de usuarios
- Administrar los usuarios y permisos por medio de grupos

Comandos

La aplicación será totalmente en consola, a excepción de los reportes en Graphviz. La aplicación consta únicamente de una consola en la cual el usuario podrá ingresar comando que le servirán para la realización de las diferentes acciones posibles. Algunas normas que debe cumplir se presentan a continuación, no distinguirá entre mayúsculas y minúsculas (case-insensitive), existen parámetros obligatorios y opcionales los cuales deben ser validados por la aplicación de consola, solo se puede colocar un comando por línea.

Si se utiliza un parámetro que no está especificado en este documento, debe mostrar un mensaje de error. Se utilizarán espacios en blanco para separar cada parámetro. Los parámetros pueden venir en cualquier orden.

Si un comando necesita más de una línea se utilizará `*` al final de la línea para indicar que continúa en la siguiente línea. También se podrán ejecutar archivos de scripts con estos comandos a través del comando `exec`, que se describe más adelante. Se podrá comentar cada comando Los comentarios de estos scripts empezarán con `#`, dichos comandos son únicamente de línea. Estos comandos se explicarán en detalle a continuación.

Exec

El programa podrá ejecutar scripts con el comando exec. Debe mostrar el contenido de la línea que está leyendo y su resultado. También debe mostrar los comentarios del script.

Parámetro	Categoría	Descripción
-path	Obligatorio	Especifica el nombre del script que se va a ejecutar, ejecutara archivos con extensión “.mia”

Ejemplo:

#ejecuta el script

```
exec -path->/home/Desktop/calificacion.mia
```

Pause

Este comando será solo la palabra “pause” no tiene atributos al ingresar este comando se pondrá en pausa solicitando que apache cualquier tecla para continuar. Este comando **NO** detiene la ejecución de un archivo solo queda a la espera de presionar una tecla para continuar su ejecución.

Administración de disco

Para la administración del disco se contará con una serie de comandos con los cuales se podrán modificar cualquier característica lógica del sistema del disco, con estos comandos se realizará la creación de archivos que simularán ser un disco duro, en dicho archivo se podrá dar formato con el sistema de archivos. Estos comandos están disponibles desde que se inicia el programa. Estos comandos son:

MKDISK

Este comando creará un archivo binario que simulará un disco duro, estos archivos binarios tendrán la extensión dsk y su contenido al inicio será \0. Deberá ocupar físicamente el tamaño indicado por los parámetros, (no importa que el sistema operativo no muestre el tamaño exacto). Recibirá el name del archivo que simulará el disco duro y tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-SIZE	Obligatorio	Este parámetro recibirá un número que indicará el tamaño del disco a crear. Debe ser positivo y mayor que cero, si no se mostrará un error.
-PATH	Obligatorio	Este parámetro será la ruta en el que se creará el archivo que representará el disco duro. Si las carpetas de la ruta no existen deberán crearse. Debe considerar que la aplicación

		<p>debe tener los permisos necesarios para la generación de carpetas en cualquier ruta.</p> <p>NOTA: La dirección de la carpeta únicamente llevará comillas dobles (") cuando algún name dentro de la dirección contenga espacios, de lo contrario podría venir sin comillas.</p> <p>Ejemplo1: /home/user/documents/</p> <p>Ejemplos2: "/home/user/documents/juan gonzalez/Mis Discos/"</p>
- NAME	Obligatorio	<p>Este parámetro será el name del disco con extensión dsk. Si no contiene la extensión dsk debe mostrar un mensaje de error.</p> <p>El name únicamente puede estar compuesto por letras, números y el único símbolo aceptado será "_" para unir caracteres, como por ejemplo: Disco_1.dsk</p> <p>NOTA: este parámetro NO se encerrará entre comillas de ningún tipo.</p>
-UNIT	Opcional	<p>Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:</p> <p>k que indicará que se utilizarán Kilobytes (1024 bytes)</p> <p>m en el que se utilizarán Megabytes (1024 * 1024 bytes)</p> <p>Este parámetro es opcional, si no se encuentra se creará un disco con tamaño en Megabytes. Si se utiliza otro valor debe mostrarse un mensaje de error.</p>

Ejemplos:

#Crea un disco de 3000 Kb en la carpeta home

```
Mkdisk -size->32 -path->/home/user/ \*
-name->Disco1.dsk -unit->k
```

#Se crearán carpetas si no existen

```
mkdisk -SiZe->8 -pAth->"/home/mis discos/DISCO Prueba/" \*
-namE->Disco_3.dsk
```

#Crearé un disco de 16 Mb ya que no hay parámetro unit

```
Mkdisk -size->16 -path->"/home/mis discos/" -NaMe->Disco4.dsk
```

RMDISK

Este parámetro elimina un archivo que representa a un disco duro mostrando un mensaje de confirmación para eliminar, sí el usuario acepta realizar el cambio el disco duro debe ser eliminado. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-PATH	Obligatorio	Este parámetro será la ruta en el que se eliminará el archivo que representará el disco duro. Si el archivo no existe, debe mostrar un mensaje de error.

Ejemplos:

#Eliminar Disco_4.dsk

```
rmDisk -path->"/home/mis discos/Disco_4.dsk"
```

FDISK

Este comando administra las particiones en el archivo que representa al disco duro. Se utilizará el primer ajuste para buscar espacio dentro del disco y crear la partición.

IMPORTANTE: Este primer ajuste solo es para buscar espacio para crear la partición, es diferente del ajuste que utilizará cada partición para crear los archivos que se utilizará más adelante en el proyecto.

Deberá mostrar un error si no se pudo realizar la operación solicitada sobre la partición, especificando por qué razón no pudo crearse (Por espacio, por restricciones de particiones, etc.). No se considerará el caso de que se coloquen parámetros incompatibles, por ejemplo en un mismo comando fdisk llamar a delete y add.

La estructura de cada disco se explicará más adelante. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-SIZE	Obligatorio	Este parámetro recibirá un número que indicará el tamaño de la partición a crear. Debe ser positivo y mayor a cero, si no mostrará un mensaje de error.
-UNIT	Opcional	Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro size. Podrá tener los siguientes valores:

		<p>B: que indicará que se utilizarán bytes K: que indicará que se utilizarán Kilobytes (1024 bytes) M: en el que se utilizarán Megabytes (1024 * 1024 bytes)</p> <p>Este parámetro es opcional, si no se encuentra se creará una partición en Kilobytes. Si se utiliza un valor diferente mostrará un mensaje de error.</p>
-PATH	Obligatorio	<p>Este parámetro será la ruta en la que se encuentra el disco en el que se creará la partición. Este archivo ya debe existir, si no se mostrará un error.</p> <p>NOTA: La dirección de la carpeta únicamente llevará comillas dobles (") cuando algún name dentro de l dirección contenga espacios, de lo contrario podría venir sin comillas.</p> <p>Ejemplo1: /home/user/documents/ Ejemplos2: "/home/user/documents/juan gonzalez/Mis Discos/"</p>
-TYPE	Opcional	<p>Indicará que tipo de partición se creará. Ya que es opcional, se tomará como primaria en caso de que no se indique. Podrá tener los siguientes valores:</p> <p>P: en este caso se creará una partición primaria. E: en este caso se creará una partición extendida. L: Con este valor se creará una partición lógica.</p> <p>Nota: Las particiones lógicas sólo pueden estar dentro de la extendida sin sobrepasar su tamaño. Deberá tener en cuenta las restricciones de teoría de particiones: La suma de primarias y extendidas debe ser como máximo 4. Solo puede haber una partición extendida por disco. No se puede crear una partición lógica si no hay una extendida.</p> <p>Si se utiliza otro valor diferente a los anteriores deberá mostrar un mensaje de error</p>
-FIT	Opcional	<p>Indicará el ajuste que utilizará la partición para asignar espacio. Podrá tener los siguientes valores:</p> <p>BF: Indicará el mejor ajuste Best Fit FF: Utilizará el primer ajuste First Fit WF: Utilizará el peor ajuste Worst Fit</p> <p>Ya que es opcional, se tomará el peor ajuste si no está especificado en el comando. Si se utiliza otro valor que no sea alguno de los anteriores mostrará un mensaje de error.</p>
-DELETE	Opcional	<p>Este parámetro indica que se eliminará una partición. Este</p>

		<p>parámetro se utiliza junto con -name y -path. Se deberá mostrar un mensaje que permita confirmar la eliminación de dicha partición.</p> <p>Si la partición no existe deberá mostrar error. Si se elimina la partición extendida, deben eliminarse las particiones lógicas que tenga adentro.</p> <p>Recibirá los siguientes valores:</p> <p>Fast: Esta opción marca como vacío el espacio en la tabla de particiones.</p> <p>Full: Esta opción además marcar como vacío el espacio en la tabla de particiones, rellena el espacio con el carácter \0.</p> <p>Si se utiliza otro valor diferente, mostrará un mensaje de error.</p>
-NAME	Obligatorio	Indicará el nombre de la partición. El nombre no debe repetirse dentro de las particiones de cada disco. Si se va a eliminar, la partición ya debe existir, si no existe debe mostrar un mensaje de error.
-ADD	Opcional	Este parámetro se utilizará para agregar o quitar espacio de la partición. Puede ser positivo o negativo. Tomará el parámetro -UNIT para las unidades a agregar o eliminar. En el caso de agregar espacio, deberá comprobar que exista espacio libre después de la partición. En el caso de quitar espacio se debe comprobar que quede espacio en la partición (no espacio negativo).

Ejemplos:

```
#Crear una Particion primaria llamada Particion1 de 72kb
#Con el peor ajuste y con asignacion Indexada en el Disco1.dsk
fdisk -size->72 -path->/home/Disco1.dsk -name->Particion1
#Crea una partición extendida dentro de Disco2 de 56 kb
#Tiene el peor ajuste y asignación Enlazada
fdisk -Type->E -path->/home/Disco2.dsk -Unit->K \*
-name->Particion2 -size->56
#Crea una partición lógica con el mejor ajuste, llamada Particion3,y
#de 1 Mb en el Disco3 y asignación contigua
fdisk -size->1 -tipo->L -unit->M -fit->BF \*
-path->/mis discos/Disco3.dsk -name->Particion3
#Intenta crear una partición extendida dentro de Disco2 de 200 kb
#Debería mostrar error ya que ya existe una partición extendida
#dentro de Disco2
fdisk -tipo->E -path->/home/Disco2.dsk -name->Part3 \*
-Unit->K -size->200
#Elimina de forma rápida una partición llamada Particion1
```

```
fdisk -delete->fast -name->"Particion1" -path->"/home/Disco1.dsk"
#Elimina de forma completa una partición llamada Particion1
fdisk -name->"Particion1" -delete->full -path->"/home/Disco1.dsk"
#Agrega 1 Mb a la partición Particion4 del Disco4.dsk
#Se debe validar que haya espacio libre después de la partición
fdisk -add->1 -unit->M -path->"/home/mis discos/Disco4.dsk" \*
-name->"Particion 4"
```

MOUNT

Este comando montará una partición del disco en el sistema. Cada partición se identificará por un id que tendrá la siguiente estructura: VD - LETRA - NUMERO. Por ejemplo: vda1, vda2, vdb1, vdc1... La letra será la misma para particiones en el mismo disco y el número diferente para particiones en el mismo disco.

Si únicamente viene la palabra reservada MOUNT deberá mostrar un listado de particiones montadas.

Parámetro	Categoría	Descripción
-PATH	Obligatorio	Este parámetro será la ruta en la que se encuentra el disco que se montará en el sistema. Este archivo ya debe existir.
-NAME	Obligatorio	Indica el nombre de la partición a cargar. Si no existe debe mostrar error.

Ejemplo:

```
#Monta las particiones de Disco1.dsk
mount -path->/home/Disco1.dsk -name->Part1 #id->vda1
mount -path->/home/Disco2.dsk -name->Part1 #id->vdb1
```



```
mount -path->/home/Disco3.dsk -name->Part2 #id->vdc1
mount -path->/home/Disco1.dsk -name->Part2 #id->vda2
#Si se coloca el comando mount sin parametros mostrará en la consola las particiones montadas.
id->vda1 -path->"/home/Disco1.dsk" -name->"Part1"
id->vdb1 -path->"/home/Disco2.dsk" -name->"Part1"
id->vdc1 -path->"/home/Disco3.dsk" -name->"Part2" id-
>vda2 -path->"/home/Disco1.dsk" -name->"Part2"
```

UNMOUNT

Desmonta una partición del sistema. Se utilizará el id que se le asignó a la partición al momento de cargarla.

Se debe poder desmontar por listado de particiones. Recibirá los siguientes parámetros.

Parámetro	Categoría	Descripción
-IDn	Obligatorio	Especifica una lista de id de las particiones que se desmontaran, Si no existe algún ID mostrara error.

Ejemplos:

#Desmonta la partición con id vda1 (En Disco1.dsk)

```
umount -id1->vda1
```

#Si no existe, se debe mostrar error

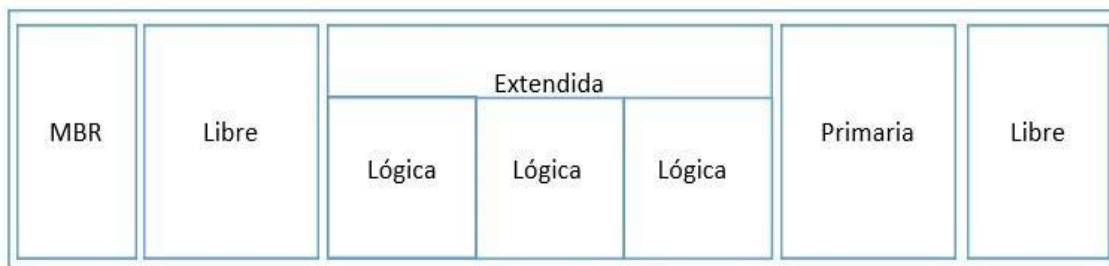
```
umount -id1->vdx1
```

#Desmonta una lista de particiones.

```
umount -id1->vda1 -id2->vdb2 -id3->vdc1
```

Estructuras Discos

Los discos serán archivos binarios que tendrán información del MBR, y espacio con particiones o bien, espacio sin utilizar. La siguiente figura es un ejemplo de los bloques en un disco con particiones en el que ya se ha eliminado una partición:



Master Boot Record

Cuando se crea una partición debe utilizarse el primer ajuste para crearla. En la figura anterior debería utilizarse el primer bloque libre para crear una nueva partición. El MBR tendrá los siguientes campos:

Nombre	Tipo	Descripción
mbr_tamaño	int	Tamaño total del disco en bytes
mbr_fecha_creacion	time	Fecha y hora de creación del disco
mbr_disk_signature	int	Número random, que identificará de forma única a cada disco
mbr_partition_1	partition	Estructura con información de la partición 1
mbr_partition_2	partition	Estructura con información de la partición 2
mbr_partition_3	partition	Estructura con información de la partición 3
mbr_partition_4	partition	Estructura con información de la partición 4

Cada partición tendrá los siguientes campos:

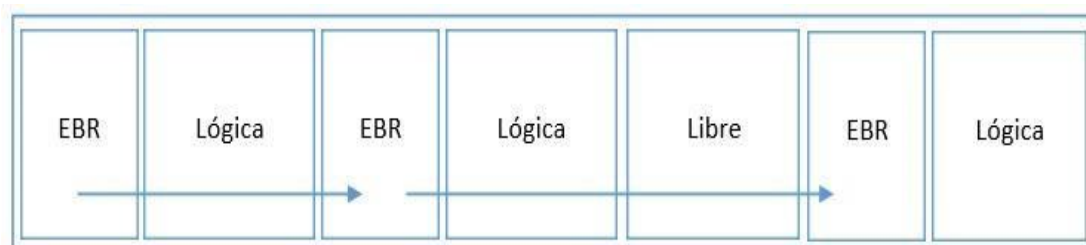
Nombre	Tipo	Descripción
part_status	char	Indica si la partición está activa o no
part_type	char	Indica el tipo de partición, primaria o extendida. Tendrá los valores P o E
part_fit	char	Tipo de ajuste de la partición. Tendrá los valores B (Best), F (First) o W (worst)
part_start	int	Indica en qué byte del disco inicia la partición
part_size	int	Contiene el tamaño total de la partición en bytes.
part_name	char[16]	Nombre de la partición

Extended Boot Record

Las particiones extendidas tendrán una estructura diferente. Se utilizará una estructura llamada EBR (Extended Boot Record) en forma de lista enlazada, que será como la siguiente:

Nombre	Tipo	Descripción
part_status	char	Indica si la partición está activa o no
part_fit	char	Tipo de ajuste de la partición. Tendrá los valores B (Best), F (First) o W (worst)
part_start	int	Indica en que byte del disco inicia la partición
part_size	int	Contiene el tamaño total de la partición en bytes.
part_next	int	Byte en el que está el próximo EBR. -1 si no hay siguiente
part_name	char[16]	Nombre de la partición

La Estructura Lógica de La Partición Extendida será como la Siguiente:



EBR inicial siempre debe existir, aunque se elimine la primera partición.

Para crear el archivo del disco se recomienda utilizar un char[1024] como buffer para crear el archivo, si se utiliza un char[1] normalmente se tarda demasiado al momento de crear el archivo.

SISTEMA DE ARCHIVOS LWH

MKFS

Este comando realiza un formateo completo de la partición, se dará formato con LWH. También creará un archivo en la raíz llamado users.txt que tendrá los usuarios y contraseñas del sistema de archivos. La estructura de este archivo se explicará más adelante.

Parámetro	Categoría	Descripción
-ID	Obligatorio	Indicará el id que se generó con el comando mount de la fase anterior. Si no existe mostrará error. Se utilizará para saber la partición y el disco que se utilizará para hacer el sistema de archivos.
-TIPO	Opcional	Indicará que tipo de formato se realizará. Ya que es opcional, se tomará como un formato completo si no se especifica esta opción. Podrá tener los siguientes valores: Fast: en este caso se realizará un formato rápido. Full: en este caso se realizará un formato completo. La diferencia entre estos dos tipos se explicará más adelante.
-ADD	Opcional	Este parámetro se utilizará para aumentar o reducir el tamaño del sistema de archivos, el valor puede ser positivo o negativo. En caso de que sea negativo se debe validar que no se elimine información dentro del sistema LWH al momento de reducir el sistema de archivos. En el caso de que sea positivo se debe validar que no sobrepase el tamaño de la partición.
-UNIT	Opcional	Este parámetro recibirá una letra que indicará las unidades que utilizará el parámetro -add. Podrá tener los siguientes valores: B: que indicará que se utilizarán bytes K: que indicará que se utilizarán Kilobytes (1024 bytes) M: en el que se utilizarán Megabytes (1024 * 1024 bytes) Este parámetro es opcional, si no se encuentra se utilizarán Kilobytes.

Ejemplos:
#Realiza un formateo rápido con LWH de la "Particion 1" del Disco1.dsk

mkfs -id->vda1 -type->fast

#Realiza un formateo completo con LWH de Particion2 en Disco2.dsk

mkfs -id->vdb1

#Agrega 500 Kb a Particion3 en Disco3.dsk
mkfs -add->500N -id->vdc1

#Agrega 1 Mb a Particion3 en Disco3.dsk

Mkfs -add->1 -id->vdc1 -unit->m

Administración De Usuarios y Grupos

Administración de Usuarios:

Este archivo será un archivo de texto, llamado users.txt guardado en el sistema LWH de la raíz de cada partición. Existirán dos tipos de registros, unos para grupos y otros para usuarios. Un id 0 significa que el usuario o grupo está eliminado, el id de grupo o de usuario irá aumentando según se vayan creando usuarios o grupos. Tendrá la siguiente estructura:

GID, Tipo, Grupo

UID, Tipo, Grupo, Usuario, Contraseña

El estado ocupará una letra, el tipo otra, el grupo ocupará cómo máximo 10 letras al igual que el usuario y la contraseña.
Al inicio existirá un grupo llamado root, un usuario root y una contraseña (#carné) para el usuario root. El archivo al inicio debería ser como el siguiente:

```
1, G, root      \n
1, U, root      , root  , 20200000\n
Este archivo se podrá modificar con comandos que se explicarán más adelante.
```

LOGIN:

Este comando se utiliza para iniciar sesión en el sistema. Se buscará el usuario únicamente en el archivo users.txt de la partición a la que se se hace referencia. No se puede iniciar otra sesión sin haber hecho un logout antes, debe mostrar un mensaje de error indicando que debe cerrar sesión antes.

Parámetro	Categoría	Descripción
-usr	Obligatorio	Especifica el nombre del usuario que iniciará sesión. Si no se encuentra en la primera partición mostrará un mensaje indicando que el usuario no existe. En este caso si distinguirá mayúsculas de minúsculas.
-pwd	Obligatorio	Indicará la contraseña del usuario, si no coincide debe mostrar un mensaje de autenticación fallida. Distinguirá entre mayúsculas y minúsculas.
-id	Obligatorio	Indica el id de la partición a la cual se desea obtener acceso. Únicamente se puede obtener acceso a las particiones que se encuentren montadas en ese momento. De lo contrario la aplicación debe mostrar un mensaje de error indicando que la partición no se encuentra montada.

Ejemplos:

```
#Se loguea en el sistema como usuario root
login &usr->root &pwd->201020576 &id->vda1

login &usr->"mi usuario" &pwd->"mi pwd" /*
&id->vda2
```

LOGOUT:

Este comando se utiliza para cerrar sesión. Debe haber una sesión activa anteriormente para poder utilizarlo, si no, debe mostrar un mensaje de error. Este comando no recibe parámetros.
Ejemplo:
#Termina la sesión del usuario
Logout

#Si se vuelve a ejecutar deberá mostrar un error
#ya que no hay sesión actualmente

Logout

Todos los siguientes comandos que se explicarán de aquí en adelante, necesitan que exista una sesión en el sistema. Si no, debe mostrar un mensaje de error indicando que necesita iniciar sesión.

mkgrp

Este comando creará un grupo para los usuarios de la partición y se guardará en el archivo users.txt de la partición, este comando solo lo puede utilizar el usuario root. Si otro usuario lo intenta ejecutar, deberá mostrar un mensaje de error, si el grupo a ingresar ya existe deberá mostrar un mensaje de error. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se creará el grupo. Si no existe, debe mostrar un error.
-name	Obligatorio	Indicará el nombre que tendrá el grupo

Ejemplo:

#Crea el grupo usuarios en la partición vda1 mkgrp -id->vda1 -nombre->"usuarios"

#Debe mostrar mensaje de error ya que el grupo ya existe mkgrp -id->vda1 -nombre->"usuarios"

El archivo users.txt debería quedar como el siguiente:

```
1, G, Root      \n
1, U, root      , root  , 123  \n
2, G, usuarios \n
```

rmgrp

Este comando eliminará un grupo para los usuarios de la partición. Solo lo puede utilizar el usuario root, si lo utiliza alguien más debe mostrar un error. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en el que se eliminará el grupo. Si no existe, debe mostrar un error.
-nombre	Obligatorio	Indicará el nombre del grupo a eliminar. Si el grupo no se encuentra dentro de la partición debe mostrar un error.

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en el que se eliminará el grupo. Si no existe, debe mostrar un error.
-name	Obligatorio	Indicará el nombre del grupo a eliminar. Si el grupo no se encuentra dentro de la partición debe mostrar un error.

Ejemplo:

#Elimina el grupo de usuarios en la partición vda1
rmgrp -id->vda1 -nombre->"usuarios"

#Debe mostrar mensaje de error ya que el grupo no existe porque ya fue eliminado

```
rmgrp -id->vda1 -nombre->"usuarios"
```

El archivo users.txt debería quedar como el siguiente:

```
1, G, Root      \n
1, U, root      , root  , 20200000 \n
0, G, usuarios \
```

mkusr

Este comando crea un usuario en la partición. Solo lo puede ejecutar el usuario root, si lo utiliza otro usuario deberá mostrar un error. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en el que se creará el usuario. Si no existe, debe mostrar un error.
-usr	Obligatorio	Indicará el nombre del usuario a crear, si ya existe, deberá mostrar un error indicando que ya existe el usuario. Máximo: 10 caracteres.
-pwd	Obligatorio	Indicará la contraseña del usuario. Máximo: 10 caracteres.
-grp	Obligatorio	Indicará el grupo al que pertenece el usuario. Debe de existir en la partición en la que se está creando el usuario, si no debe mostrar un mensaje de error. Máximo: 10 caracteres.

```
Ejemplo:
#Crea el grupo usuarios en la partición vda1
Mkusr -id->vda1 -nombre->"user1" -grp->usuarios -pwd->usuario
```

El archivo users.txt debería quedar así:

```
1, G, Root      \n
1, U, root      , root  , 20200000\n
2, G, usuarios \n
2, U, usuarios , user1  , usuario \n
```

rmusr

Este comando elimina un usuario en la partición. Solo lo puede ejecutar el usuario root, si lo utiliza otro usuario deberá mostrar un error. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se eliminará el usuario. Si no existe, debe mostrar un error.
-usr	Obligatorio	Indicará el nombre del usuario a eliminar, si no existe, deberá mostrar un error indicando que el usuario no existe.

```
Ejemplo:
#Elimina el usuario user1 de la partición
rmusr -id->vda1 -usr->user
```

El Archivo Users.txt debería quedar así:

```
1, G, Root \n
1, U, root, root, 123 \n
2, G, usuarios \n
2, U, usuarios, user1, 123 \n
```

Usuario root

Este usuario es especial y no importando que permisos tenga el archivo o carpeta, el siempre tendrá los permisos 777 sobre cualquier archivo o carpeta (Esto se explica en detalle posteriormente). Podrá mover, copiar, eliminar, crear, etc. Todos los archivos o carpetas que desee. No se le negará ninguna operación por permisos, ya que él los tiene TODOS. Los permisos únicamente se pueden cambiar con chmod que se explicará posteriormente. Se debe tomar en cuenta en qué categoría está el usuario, si es el propietario, si pertenece al mismo grupo en que está el propietario o si es otro usuario que no pertenece al grupo del propietario. En base a esta comprobación, el usuario puede estar en tres distintas categorías: Propietario (U), Grupo (G) u Otro (O). Dependiendo de estas categorías se determinan los permisos hacia el archivo o carpeta.

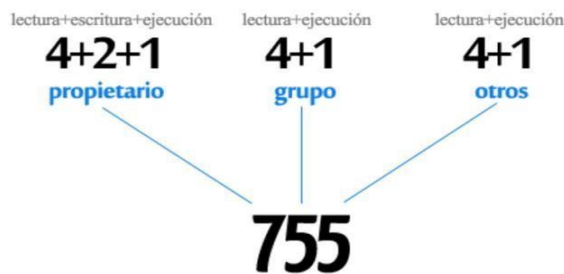
Administración de carpetas, archivos y permisos

Estos comandos permitirán crear archivos y carpetas, así como editarlos, copiarlos, moverlos y eliminarlos. Los permisos serán para el usuario propietario del archivo, para el grupo al que pertenece y para otros usuarios, así como en Linux.

Chmod

Para una buena administración de sistemas no se puede olvidar la sección de seguridad del mismo, para esto cambiará los permisos de uno o varios archivos o carpetas. Lo podrá utilizar el usuario root en todos los archivos o carpetas y también lo podrán utilizar otros usuarios, pero solo sobre sus propios archivos. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se encuentra el archivo o carpeta a la que se le cambiarán los permisos.
-path	Obligatorio	Este parámetro será la ruta en la que se encuentra el archivo O carpeta a la que se le cambiarán los permisos.
-ugo	Obligatorio	Indica los permisos que tendrán los usuarios. Serán tres números, uno para el Usuario, el siguiente para el Grupo al que pertenece el usuario y el último para Otros usuarios fuera del grupo. Cada número tendrá los valores desde el 0 al 7. Si el número está fuera de este rango se mostrará un error. A nivel de bits significan permisos para lectura, escritura y ejecución. Por ejemplo el número 5 (101) indica permisos para leer y ejecutar. El número 7 indica permisos para las tres operaciones anteriores. El número 0 indica que no tendrá permisos para utilizar el archivo.
-R	Opcional	Indica que el cambio será recursivo en el caso de carpetas. El cambio afectará a todos los archivos y carpetas en la que la ruta contenga la carpeta especificada por el parámetro -path y que sean propiedad del usuario actual



Ejemplos:

- #Cambia los permisos de la carpeta home recursivamente
- #Todos los archivos o carpetas que tengan /home cambiarán
- #Por ejemplo si existiera /home/user/docs/a.txt
- #Cambiaría los permisos de las tres carpetas y del archivo

```
chmod -id->vda1 -path->"/home" -R -ugo->764
```

- #Cambia los permisos de la carpeta home
- #Se debe comprobar que la carpeta home pertenezca al usuario #actual, si no deberá mostrar un mensaje de error.

```
chmod -id->vda1 -path->"/home" -ugo->777
```

mkfile

Este comando permitirá crear un archivo, el propietario será el usuario que actualmente ha iniciado sesión. Tendrá los permisos 664. El usuario deberá tener el permiso de escritura en la carpeta padre, si no debe mostrar un error. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se creará el archivo. Si no existe debe mostrar error.
-path	Obligatorio	Este parámetro será la ruta del archivo que se creará. Si lleva espacios en blanco deberá encerrarse entre comillas. Si ya existe debe sobrescribir el archivo. Si no existen las carpetas padres, debe mostrar error, a menos que se utilice el parámetro -p, que se explica posteriormente.
-p	Opcional	Si se utiliza este parámetro y las carpetas especificadas por el parámetro -path no existen, entonces deben crearse las carpetas padres. Si ya existen, no deberá crear las carpetas. No recibirá ningún valor, si lo recibe debe mostrar error.
-size	Opcional	Este parámetro indicará el tamaño en bytes del archivo, el contenido será el abecedario cuantas veces sea necesario. Si no se utiliza este parámetro, el tamaño será 0 bytes. Si es negativo debe mostrar error.
-cont	Opcional	Indicará la Cadena que será Escrita dentro del Archivo.

Ejemplos:

- #Crea el archivo a.txt
- #Si no existen las carpetas home user o docs se crean
- #El tamaño del archivo es de 15 bytes
- #El contenido sería: 012345678901234

```
mkFile -SIZE->15 -id->vdb1 -PatH->"/home/user/docs/a.txt" -p
```

- #Crea "archivo 1.txt" la carpeta "mis documentos" ya debe existir
- #el tamaño es de 0 bytes

```
mkfile -id->vda1 -path->"/home/mis documentos/archivo 1.txt"
```

- #Crea el archivo b.txt
- #El contenido del archivo será el mismo que el archivo b.txt

#que se encuentra en el disco duro de la computadora.

```
mkfile -id->vda1 -path->"/home/user/docs/b.txt" -p \* -
cont->"/home/Documents/b.txt"
```

cat

Este comando permitirá mostrar el contenido del archivo en un archivo de texto, si el usuario que actualmente está logueado tiene acceso al permiso de lectura. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se creará el archivo. Si no existe debe mostrar error.
-FILEn	Obligatorio	Permitirá Admitir como argumentos una lista de n ficheros que hay que enlazar. Estos se encadenarán en el mismo orden en el cual fueron especificados. Si no existe el archivo o no tiene permiso de lectura, debe mostrarse un mensaje de error.

Ejemplos:

#Lee el archivo a.txt En la terminal debería mostrar el contenido, en #este ejemplo 01234567890123

```
Cat -file1->"/home/user/docs/a.txt" -ld->vdb1
```

#enlazara los archivos a.txt (datos archivo a) b.txt (01234567890123) #c.txt (0123) y debería mostrar el contenido siguiente, cada archivo #va separado por salto de línea

datos archivo a

01234567890123

0123

```
Cat -file1->"/home/a.txt" -ld->vdb1 -file2->"/home/b.txt" \ -file3->"/home/c.txt"
```

rm

Este comando permitirá eliminar un archivo o carpeta y todo su contenido, si el usuario que actualmente está logueado tiene acceso al permiso de escritura sobre el archivo y en el caso de carpetas, eliminará todos los archivos o subcarpetas en los que el usuario tenga permiso de escritura. Si no pudo eliminar un archivo o subcarpeta dentro de la carpeta por permisos, no deberá eliminar la(s) carpetas padre. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se eliminará el archivo
-path	Obligatorio	Este parámetro será la ruta del archivo o carpeta que se eliminará. Deberá encerrarse entre comillas. Si no existe el archivo o no tiene permisos de escritura en la carpeta o en el archivo, debe mostrarse un mensaje de error. Si no pudo eliminar algún archivo o carpeta no deberá eliminar los padres.
-rf	Opcional	Permite hacer borrados recursivos. Con esta opción se pueden borrar sin importar los permisos siempre y cuando sea el usuario root.

Ejemplos:

```
#Elimina el archivo a.txt, b.txt muestra error si no tiene permiso
rm -PatH->"/home/user/docs/a.txt" -Id->vdb1
rm -PatH->"/home/user/docs/b.txt" -Id->vdb1
#Error por permisos
#Elimina la carpeta user y todo su contenido (docs, a.txt)
#Si el usuario no tuviera permiso de escritura sobre b.txt
#No debería eliminar las carpetas padre docs ni user, solo
a.txt
rm -PatH->"/home/user" -Id->vdb1

#Elimina la carpeta user y todo su contenido (docs, a.txt) sin importar los permisos (logueado como usuario root)
Rm -Id->vdb1 -path->"/home/user" -RF
```

EDIT

Este comando permitirá editar el contenido de un archivo para que ocupe un tamaño específico, o bien, para asignarle otro contenido. Funcionará si el usuario que actualmente esta logueado tiene acceso al permiso de lectura y escritura sobre el archivo, si no debe mostrar error. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se modificará el archivo. Si no existe debe mostrar error.
-path	Obligatorio	Este parámetro será la ruta del archivo que se modificará. Deberá encerrarse entre comillas. Si no existe, debe mostrar un mensaje de error.
-size	Opcional	Este parámetro indicará el nuevo tamaño en bytes del a archivo, el contenido serán el abecedario cuantas veces sea necesario. Si es negativo debe mos trar error.
-cont	Opcional	Indicará una cadena la cual sera el nuevo contenido del Archivo.

(*) Se deberá ingresar por lo menos uno de los dos parámetros (size o cont). Si no hay ninguno de los dos, debe mostrar un mensaje de error.

Ejemplos:

```
#Modifica el archivo a.txt
#El tamaño del archivo es de 22 bytes
#El contenido sería: abcdefghijklmnopqrstu
Edit -SIZE->22 -id->vdb1 -PatH->"/home/user/docs/a.txt"
#Modifica el archivo b.txt
#El contenido del archivo será el mismo que el archivo c.txt
#que se encuentra en el disco duro de la computadora.
edit -id->vda1 -path->"/home/user/docs/b.txt" -cont ->"/home/Documents/c.txt"

#Modifica nuevamente el archivo b.txt
edit -id->vda1 -path->"/home/user/docs/b.txt" -cont ->"/home/Documents/d.txt"
```

Ren

Este comando permitirá cambiar el nombre de un archivo o carpeta, si el usuario actualmente logueado tiene permiso de escritura sobre el archivo o carpeta. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se renombrará el archivo o carpeta. Si no existe debe mostrar error.
-path	Obligatorio	Este parámetro será la ruta del archivo o carpeta al que se le cambiará el nombre. Deberá encerrarse entre comillas. Si no existe el archivo o carpeta o no tiene permisos de escritura sobre la carpeta o archivo, debe mostrarse un mensaje de error.
-name	Obligatorio	Especificará el nuevo nombre del archivo, debe verificar que no exista un archivo con el mismo nombre, de ser así debe mostrar un mensaje de error. Deberá encerrarse entre comillas.

Ejemplos:

```
#Cambia el nombre del archivo a.txt a b1.txt
ren -Path->"/home/user/docs/a.txt" -Id->vdb1 -nombre->"b1.txt"
#Debera mostrar error ya que el archivo b1.txt ya existe
ren -Path->"/home/user/docs/c.txt" -Id->vdb1 -nombre->"b1.txt"
```

Mkdir

Este comando es similar a mkfile, pero no crea archivos, sino carpetas. El propietario será el usuario que actualmente ha iniciado sesión. Tendrá los permisos 664. El usuario deberá tener el permiso de escritura en la carpeta padre, si no debe mostrar un error. Tendrá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se creará la carpeta. Si No existe debe mostrar error.
-path	Obligatorio	Este parámetro será la ruta de la carpeta. Si no existen las carpetas padres, debe mostrar error, a menos que se utilice el parámetro -p, que se explica posteriormente.
-p	Opcional	Si se utiliza este parámetro y las carpetas padres en el Parámetro path no existen, entonces deben crearse.

Ejemplos:

```
#Crea la carpeta usac
#Si no existen las carpetas home user o docs se crean
Mkdir -P -id->vda1 -path->"/home/user/docs/usac"

#Crea la carpeta "archivos 2016"
#La carpeta padre ya debe existir
Mkdir -ID->vda1 -path->"/home/mis documentos/archivos 2016"
```

Cp

Este comando permitirá realizar una copia del archivo o carpeta y todo su contenido hacia otro destino.

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que está el archivo o Carpeta que se quiere copiar. Si no existe, debe mostrar un error.
-path	Obligatorio	Este parámetro será la ruta del archivo o carpeta que se desea copiar. Debe copiar todos los archivos y carpetas con todo su contenido, a los cuales tenga permiso de lectura. Si no tiene permiso de lectura, no realiza la copia únicamente de ese archivo o carpeta. Muestra un error si no existe la ruta.
-dest	Obligatorio	Este parámetro será la ruta de la carpeta a la que se copiará el archivo o carpeta. Debe tener permiso de escritura sobre la carpeta. Debe mostrar un mensaje de error si no tiene permisos para escribir o si la carpeta no existe.

Ejemplos:

```
#/  
#|_home      #664  
# |_user      #664  
# | |_documents #664  
# | |_a.txt    #664  
# | |_b.txt    #224  
# |_images    #664  
#Copia documents a images  
cp -id->vda2 -Path->"/home/user/documents" -dest->"/home/images"
```

mv

Este comando moverá un archivo o carpeta y todo su contenido hacia otro destino. Si el origen y destino están dentro de la misma partición, solo cambiará las referencias, para que ya no tenga el padre origen sino, el padre destino, y que los padres de la carpeta o archivo ya no tengan como hijo a la carpeta o archivo que se movió. Solo se deberán verificar los permisos de escritura sobre la carpeta o archivo origen. Si el origen y destino están en una partición diferente, funcionará como copiar y después eliminar. El usuario deberá tener el permiso de escritura en la carpeta padre destino, si no debe mostrar un error.

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que está el archivo o Carpeta que se quiere mover. Si no existe, debe mostrar un error.
-path	Obligatorio	Este parámetro será la ruta del archivo o carpeta que se desea mover. Si no tiene permiso de escritura, no mueve el archivo o carpeta. Muestra error si no existe o no tiene permiso.
-dest	Obligatorio	Este parámetro será la ruta de carpeta a la que se moverá el archivo o carpeta. Debe tener permiso de

		escritura sobre la carpeta. Debe mostrar un mensaje de error si no tiene permiso de escritura o si la carpeta no existe.
--	--	---

Ejemplo:

```
#/  
#|_home      #664  
# |_user     #664  
# | |_documents #664  
# | |_a.txt   #664  
# | |_b.txt   #224  
# |_images   #664
```

#Mueve documents a images

```
mv -id->vda1 -iddestiny->vda1 -Path->"/home/user/documents" \*  
-dest->"/home/images"
```

find

Este comando permitirá realizar una búsqueda por el nombre del archivo o carpeta. Permitirá los siguientes caracteres especiales:

Carácter	Descripción
?	Un solo carácter
*	Uno o más

Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se buscará el archivo O carpeta. Si no existe, debe mostrar un error.
-path	Obligatorio	Este parámetro será la ruta de la carpeta en el que se inicia la búsqueda, deberá buscar en todo su contenido. Debe tener permisos de lectura en los archivos que buscará.
-nombre	Obligatorio	Indica el nombre del archivo o carpeta que se está buscando. Debe aceptar los caracteres especiales definidos anteriormente

El resultado debe mostrarse en forma de árbol, mostrando todos los archivos encontrados. find * mostrará toda la estructura del sistema de archivos a partir de la carpeta indicada ya que no hay filtro.

Ejemplos:

```
find -id->vda1 -path->"/" -nombre->*  
#Arbol actual  
#/  
#|_home      #664  
# |_user     #664  
# | |_a.txt   #664
```

```
# | |_b.txt      #420
# | _images     #664
#      |_a.jpg  #664
#      |_abcd.jpg    #664
#Busca los archivos que tengan una letra como nombre
#y cualquier extensión
find -id->vda1 -path->"/home" -nombre->"?.*"
```

Chown

Cambiará el propietario de uno o varios archivos o carpetas. Lo podrá utilizar el usuario root en todos los archivos o carpetas y también lo podrán utilizar otros usuarios, pero solo sobre sus propios archivos. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-id	Obligatorio	Especifica el id de la partición en la que se encuentra el archivo o carpeta a la que se le cambiará el propietario.
-path	Obligatorio	Este parámetro será la ruta en la que se encuentra el archivo o carpeta a la que se le cambiará el propietario. Si no existe la ruta deberá mostrar mensaje de error.
-R	Opcional	Indica que el cambio será recursivo en el caso de carpetas. El cambio afectará a todos los archivos y carpetas en la que la ruta contenga la carpeta especificada por el parámetro -path.
-usr	Obligatorio	Nombre del nuevo propietario del archivo o carpeta. Si no Existe o está eliminado debe mostrar error.

```
Ejemplos:
#Cambia el propietario de la carpeta home recursivamente
chown -id->vda1 -path->"/home" -R -usr->user2
#Cambia los permisos de la carpeta home
chown -id->vda1 -path->"/home" -usr->user1
```

Chgrp

Cambiará el grupo al que pertenece el usuario. Únicamente lo podrá utilizar el usuario root. Recibirá los siguientes parámetros:

Parámetro	Categoría	Descripción
-----------	-----------	-------------

-usr	Obligatorio	Especifica el nombre del usuario al que se le cambiará de grupo. Si no existe debe mostrar un error.
-grp	Obligatorio	Contendrá el nombre del nuevo grupo al que pertenece El usuario. Si no existe o está eliminado debe mostrar un error.

Ejemplos:

#Cambia el grupo del user2

chgrp -usr->user2 -grp->grupo1

#Cambia el grupo del user1

chgrp -usr->user1 -grp->grupo2

Estructura del Sistema de Archivos LWH

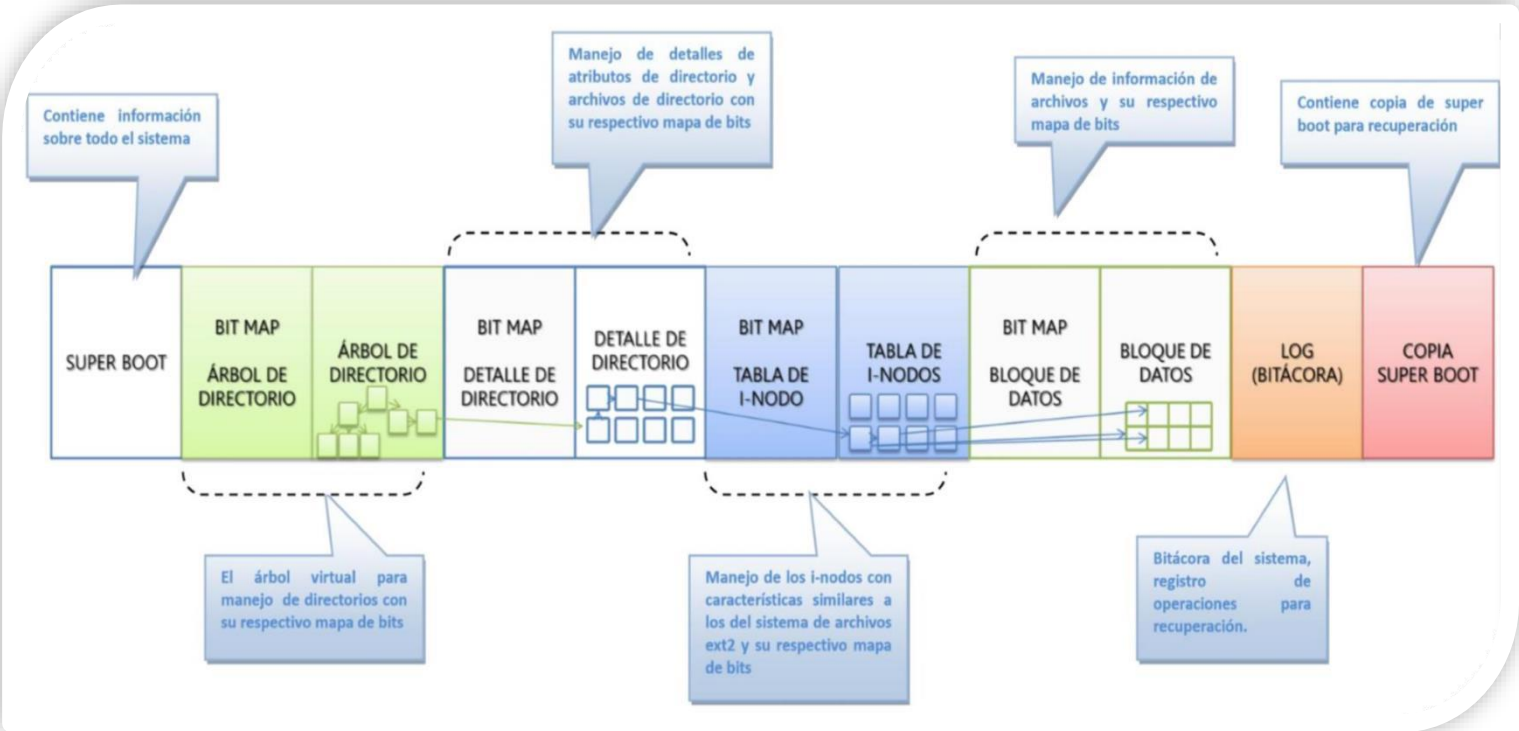
Es un sistema de archivo denominado híbrido ya que tiene una estructura y funcionamiento tomado de los sistemas de archivos Windows y GNU/Linux. Que permitirán el manejo de diferentes operaciones sobre archivos y directorios, formas de acceso a archivos, y un log o bitácora para la recuperación de la información del sistema. Utilizando como un disco duro virtual un Archivo Binario.

A continuación se explicarán las estructuras del sistema de archivos LWH: Se deberán implementar las estructuras como se especifican a continuación. La estructura en bloques es la siguiente:

Estructura del sistema de Archivos:



Descripción del Sistema De Archivos LWH



DESCRIPCIÓN Y FUNCIONAMIENTO DE CADA BLOQUE DEL SISTEMA LWH

1. SUPER BOOT

El super boot contiene información general de todo el sistema, su estructura es la siguiente:

*Estructura para el reporte sb, se explicará en la sección de Reportes.



DATO EN LA ESTRUCTURA	DESCRIPCIÓN
sb_nombre_hd	Es el nombre del disco duro virtual
sb_arbol_virtual_count	Cantidad de estructuras en el árbol virtual de directorio
sb_detalle_directorio_count	Cantidad de estructuras en el detalle de directorio
sb_inodos_count	Cantidad de i&nodos en la tabla de i&nodos
sb_bloques_count	Cantidad de bloque de datos
sb_arbol_virtual_free	Cantidad de estructuras en el árbol virtual de directorio libres
sb_detalle_directorio_free	Cantidad de estructuras en el detalle de directorio libres
sb_inodos_free	Cantidad de i&nodos en la tabla de i&nodos libres
sb_bloques_free	Cantidad de bloque de datos libres
sb_date_creacion	Fecha de creación del sistema el formato de fecha: dd/mm/yyyy hh:mm
sb_date_ultimo_montaje	Última fecha de montaje el formato de fecha: dd/mm/yyyy hh:mm
sb_montajes_count	Contador de montajes del sistema de archivos LWH
sb_ap_bitmap_arbol_directorio	Apuntador al inicio del bitmap del árbol virtual de directorio
sb_ap_arbol_directorio	Apuntador al inicio del árbol virtual de directorio
sb_ap_bitmap_detalle_directorio	Apuntador al inicio del bitmap de detalle de directorio
sb_ap_detalle_directorio	Apuntador al inicio del detalle de directorio
sb_ap_bitmap_tabla_inodo	Apuntador al inicio del bitmap de la tabla de i&nodos
sb_ap_tabla_inodo	Apuntador al inicio de la tabla de i&nodos
sb_ap_bitmap_bloques	Apuntador al inicio del bitmap de bloques de datos
sb_ap_bloques	Apuntador al inicio del bloque de datos
sb_ap_log	Apuntador al inicio del log o bitácora
sb_size_struct_arbol_directorio	Tamaño de una estructura del árbol virtual de directorio
sb_size_struct_detalle_directorio	Tamaño de la estructura de un detalle de directorio
sb_size_struct_inodo	Tamaño de la estructura de un i&nodo
sb_size_struct_bloque	Tamaño de la estructura de un bloque de datos
sb_first_free_bit_arbol_directorio	Primer bit libre en el bitmap árbol de directorio
sb_first_free_bit_detalle_directorio	Primer bit libre en el bitmap detalle de directorio
sb_first_free_bit_tabla_inodo	Primer bit libre en el bitmap de i&nodo
sb_first_free_bit_bloques	Primer bit libre en el bitmap de bloque de datos
sb_magic_num	Número de Carnet del Estudiante

Ya que el comando mkfs con el parámetro -add agrega o quita espacio de la partición, se deberán modificar estas estructuras sin eliminar información. Se calcula cuántos bloques se agregan o quitarán y se empieza moviendo de derecha a izquierda en el caso de agregar y en el caso de quitar bloques se empieza moviendo de izquierda a derecha para evitar sobrescribir información. Sí existe riesgo de perder información al momento de reducir el tamaño, deberá mostrar mensaje de error y no se podrá realizar la reducción.

1. BITMAP ÁRBOL VIRTUAL DE DIRECTORIO Y ÁRBOL VIRTUAL DE DIRECTORIO



1.1 BITMAP DE ÁRBOL VIRTUAL DE DIRECTORIO

Mapa de bit para verificar el estado de las estructuras en el árbol virtual de directorio, la metodología a utilizar será:

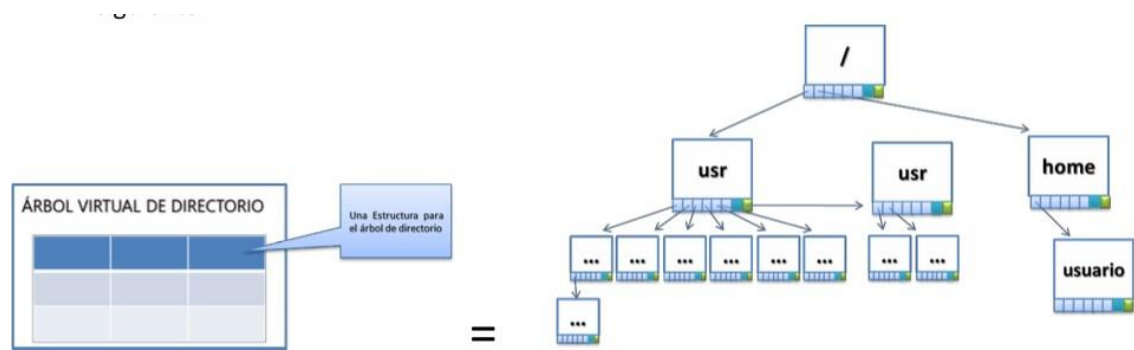
- 1 = Ocupado
- 0 = Libre

```
BIT MAP
ÁRBOL DE DIRECTORIO

01010101010101111
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
00000000000000000
```

1.2 ÁRBOL VIRTUAL DE DIRECTORIO

Simula un árbol virtual que permitirá el manejo de directorios la estructura es la siguiente



Una estructura en el árbol de directorio tiene los siguientes datos. (avd = árbol virtual de directorio).

DATO EN LA ESTRUCTURA	DESCRIPCIÓN
avd_fecha_creacion	Fecha de creación del directorio, el formato de fecha: dd/mm/yyyy hh:mm
avd_nombre_directorio	Contiene el nombre del directorio
avd_ap_array_subdirectorios	Un arreglo de apuntadores directos a sub-directorios
avd_ap_detalle_directorio	Un apuntador a un detalle de directorio. Este solo se utilizará en el primer directorio.
avd_ap_arbol_virtual_directorio	Un apuntador a otro mismo tipo de estructura por si se usan los 6 apuntadores del arreglo de sub-directorios para que puedan seguir creciendo los sub-directorios.
avd_proper	Contiene el id del propietario de la carpeta, el que se ha generado al momento de crear usuarios. Si el usuario ha sido eliminado, la carpeta pasa a ser propiedad de root

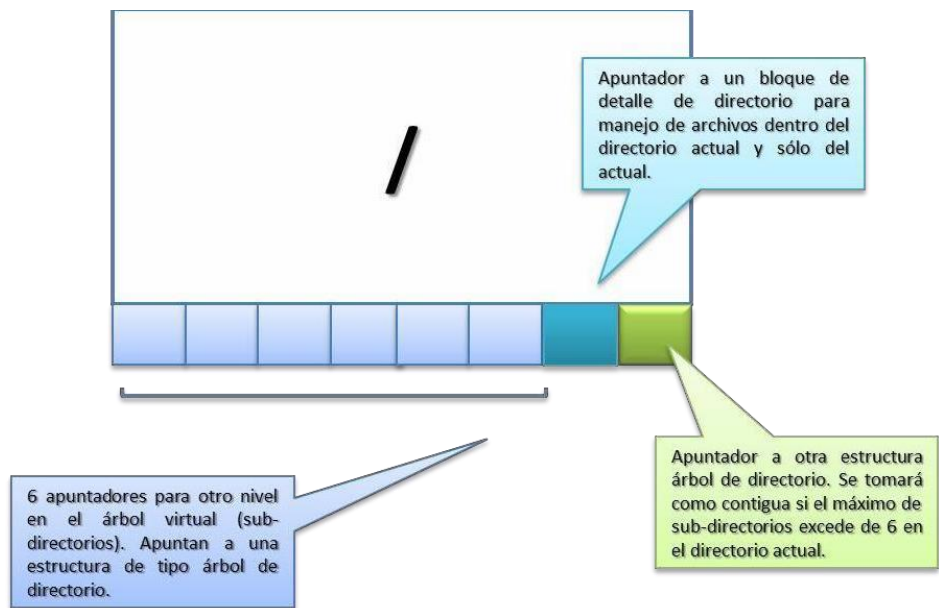
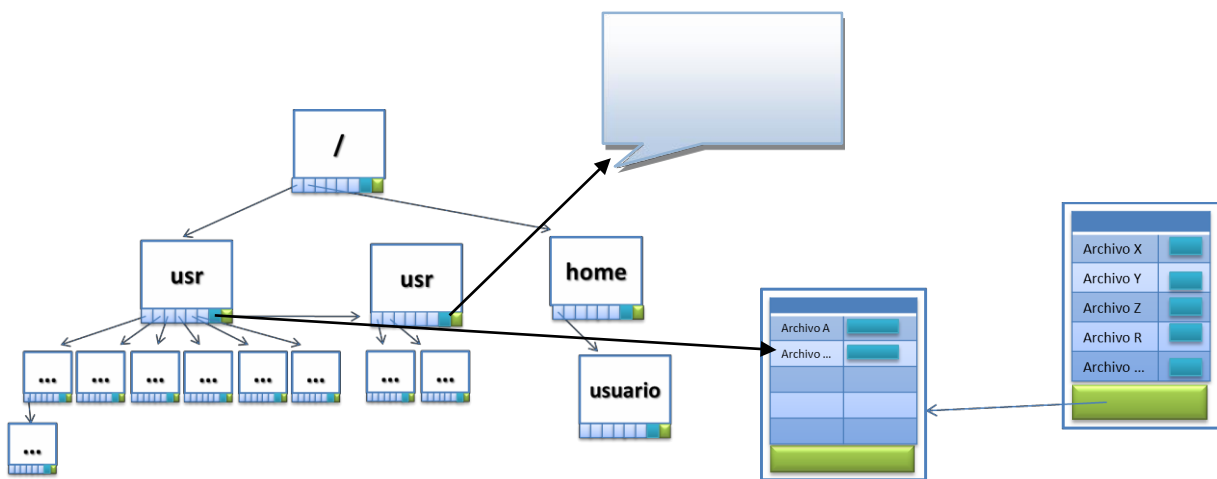
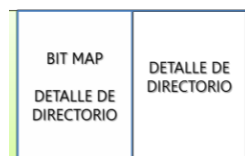


Imagen de una estructura en el árbol virtual de directorio

Al inicio al darle formato al sistema el único directorio existente será la raíz representada por “/”.



2 BITMAP DETALLE DIRECTORIO Y DETALLE DE DIRECTORIO



2.1 BITMAP DE DETALLE DE DIRECTORIO

Mapa de bit para verificar el estado de las estructuras del detalle de directorio, la metodología a utilizar será:

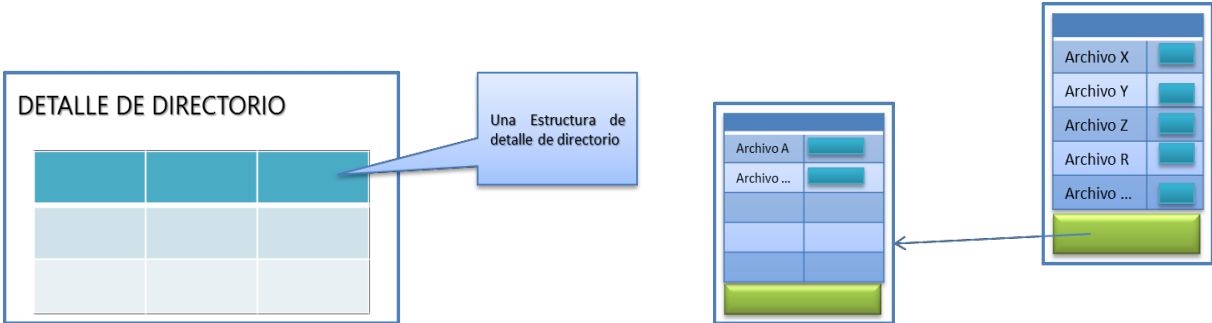
- 1 = Ocupado
- 0 = Libre

BIT MAP
DETALLE DIREC TORIO

010101010101011111
000000000000000000
000000000000000000
000000000000000000
000000000000000000
000000000000000000

2.2 DETALLE DE DIRECTORIO

Contiene las estructuras para el manejo de archivos de directorio, donde los archivos son manejados por la tabla de i-nodos.



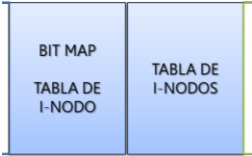
Una estructura en el detalle de directorio tiene los siguientes datos. (dd = detalle de directorio). Este detalle de directorio podrá manejar un máximo de 5 archivos.

DATO EN LA ESTRUCTURA	DESCRIPCIÓN
dd_array_files	Arreglo de estructura de tamaño 5 contiene: <ul style="list-style-type: none">• dd_file_nombre: maneja el nombre del archivo.• dd_file_ap_inodo: apuntador al inodo.• dd_file_date_creacion: fecha de creación del archivo (dd/mm/yyyyhh:mm)• dd_file_date_modificacion: fecha de modificación del archivo(dd/mm/yyyy hh:mm)
dd_ap_detalle_directorio	Si se quiere ingresar un sexto archivo y no hay espacio en esta estructura de detalle directorio, apunta a otra estructura de detalle de directorio.

Nombre archivo	i-nodo
Archivo X	10
Archivo Y	25
Archivo Z	150
Archivo R	200
Archivo S	5

Imagen de manejo de archivos dentro de la estructura de detalle de directorio

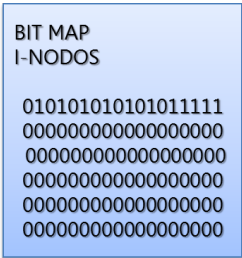
3. BITMAP DE I-NODOS Y TABLA DE I-NODOS



3.1 BITMAP DE TABLA DE I-NODO

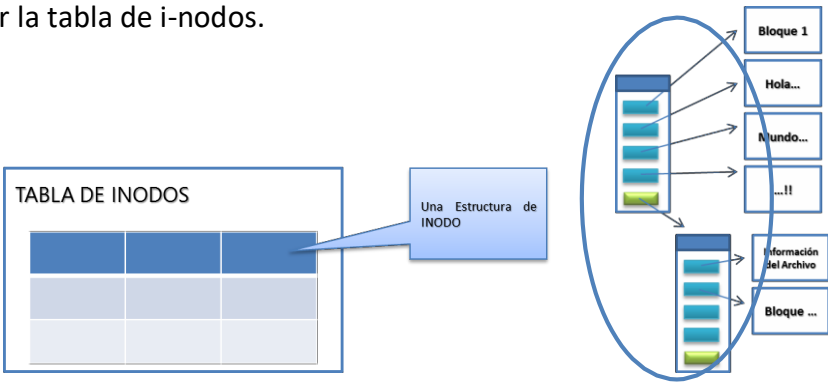
Mapa de bit para verificar el estado del i-nodo, la metodología a utilizar será:

- 1 = Ocupado
- 0 = Libre



3.2 TABLA DE I-NODO

Contiene las estructuras para el manejo de archivos de directorio, donde los archivos son manejados por la tabla de i-nodos.



Una estructura de i-nodo tiene los siguientes datos (i = i-nodo). La estructura de inodo puede apuntar como máximo a 4 bloques de datos si el archivo es mayor o crece más que eso se usará el i_ap_indirecto para apuntar a otro i-nodo.

DATO EN LA ESTRUCTURA	DESCRIPCIÓN
i_count_inodo	Número de i-nodo
i_size_archivo	Tamaño del archivo
i_count_bloques_asignados	Número de bloques asignados
i_array_bloques	Arreglo de 4 apuntadores a bloques de datos para guardar el archivo
i_ap_indirecto	Un apuntador indirecto por si el archivo ocupa más de 4 bloques de datos, para el manejo de archivos de tamaño “grande”
i_id_proper	

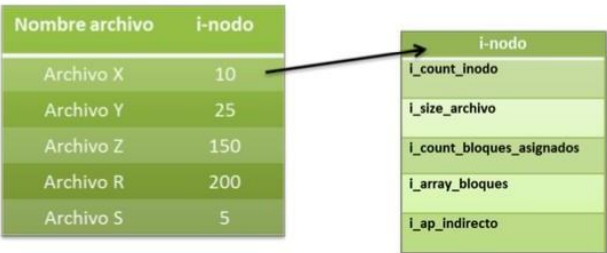


Imagen de un i-nodo en el bloque de datos de la tabla de i-nodos que está siendo apuntado por una estructura de detalle de directorio.

4. BITMAP DE BLOQUE DATOS Y BLOQUE DE DATOS



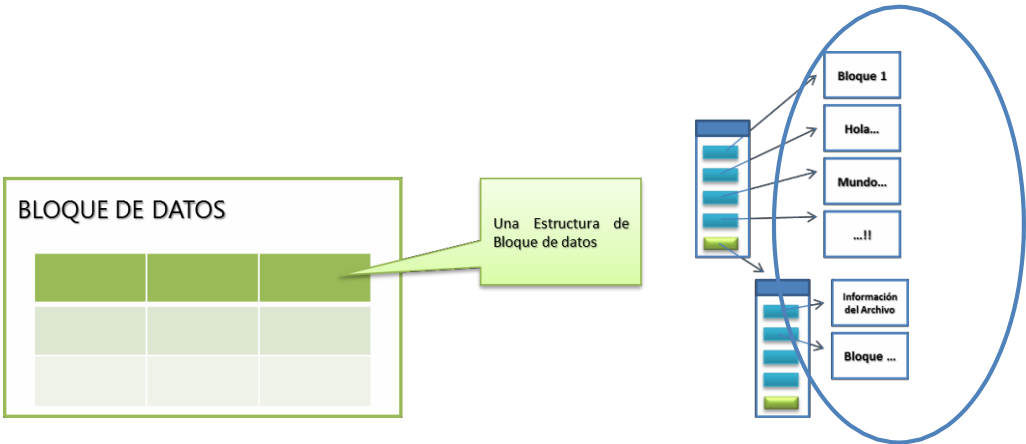
4.1 BITMAP DE BLOQUE DE DATOS

Mapa de bit para verificar los bloques de datos, la metodología a utilizar será:

- 1 = Ocupado
- 0 = Libre

5.2 BLOQUE DE DATOS

Contiene las estructuras del bloque de datos



Una estructura de un bloque de datos (bd = bloque de dato) es la siguiente que podrá almacenar como máximo 25 caracteres.

DATO EN LA ESTRUCTURA	DESCRIPCIÓN
db_data	Contiene la información del archivo

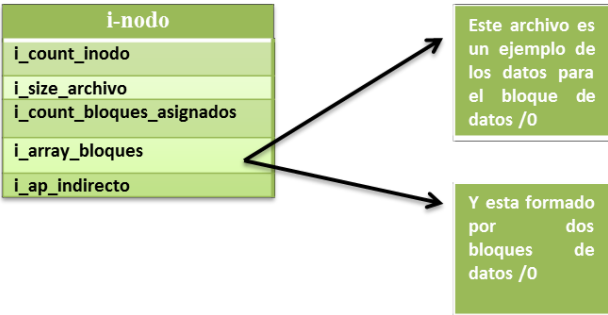


Imagen de un i-nodo apuntando a bloques de datos.

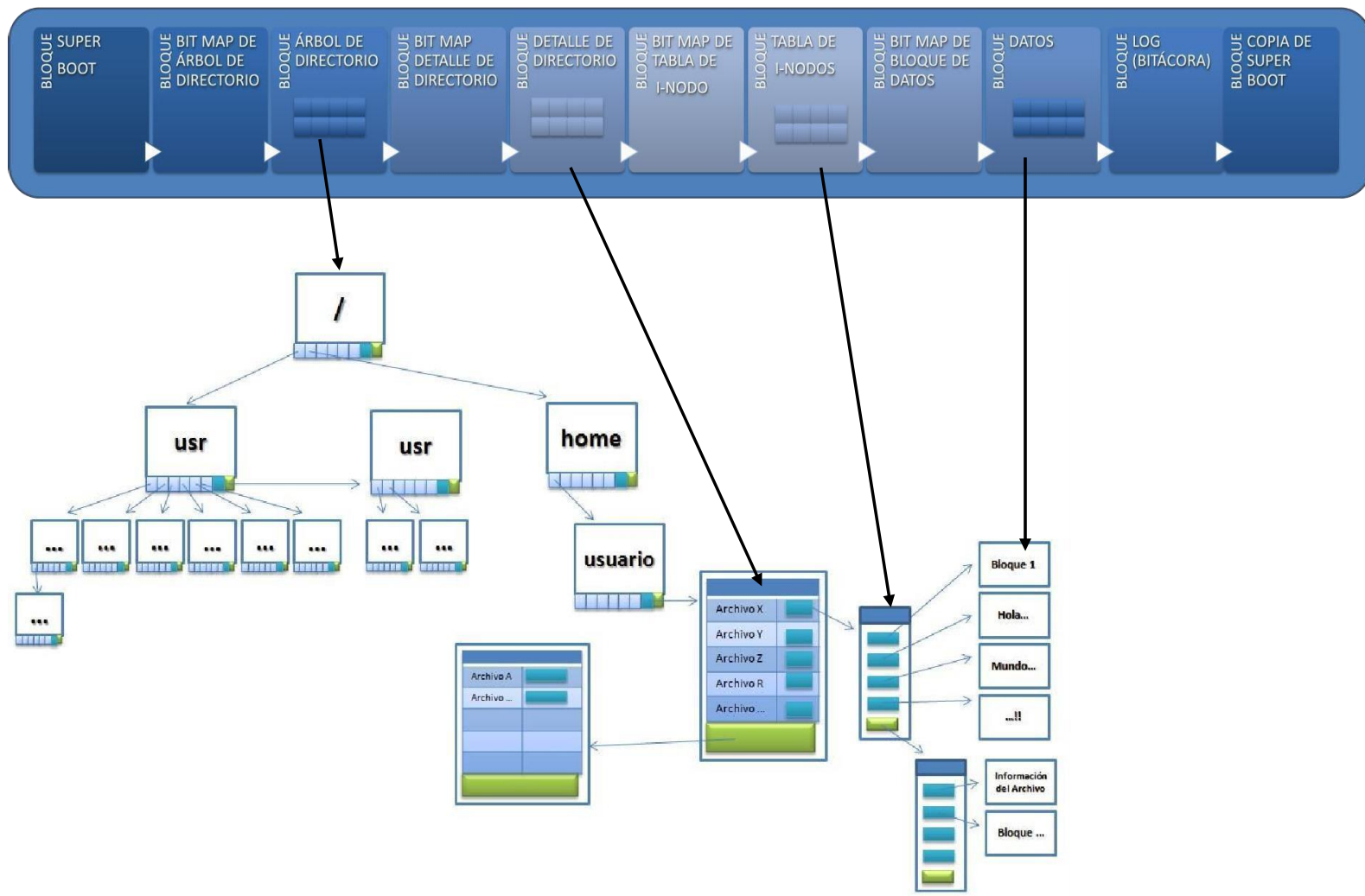
5. LOG (BITÁCORA)

La bitácora maneja todas las transacciones que realiza el sistema de archivos (respaldo para recuperación), la cual tiene maneja la siguiente información.

DATO EN LA ESTRUCTURA	DESCRIPCIÓN
log_tipo_operación	El tipo de operación a realizarse
log_tipo	Si es archivo (0), si es directorio(1)
log_nombre	Nombre archivo o directorio
log_contenido	Si hay datos contenidos
log_fecha	Fecha de transacción

6. BLOQUE DE COPIA DE SUPER BOOT

Una estructura igual al bloque de super boot para recuperación del sistema



1. RECUPERACIÓN DEL SISTEMA

La recuperación del sistema se hará por medio de boot con él log. Para esto la aplicación debe de tener las siguientes opciones:



Simulate System loss

Este formatea los siguientes bloques de datos para simular un fallo en el disco (una partición en específica), una inconsistencia o pérdida de información. Se deberán limpiar los siguientes bloques con el carácter /0.

1. Super boot
2. Bloque de bitmap de árbol virtual de directorio
3. Bloque de árbol virtual de directorio
4. Bloque de bitmap de detalle de directorio
5. Bloque de detalle de directorio
6. Bloque de bitmap de tabla de i-nodo
7. Bloque de tabla de i-nodo
8. Bloque de bitmap de bloque de datos
9. Bloque de datos.

#Simulando la perdida del sistema de archivos de la partición1

Loss -id::vda1



RECUPERAR: Por medio del log (bitácora) y el sector de copia de boot se recuperará el sistema a un estado consistente antes del último formateo.

BORRAR BITÁCORA: Si se formatea el sistema de archivos se tiene que limpiar la bitácora.

Simulate System Recovery

Este Recuperará los siguientes bloques de datos para simular una recuperación en el disco (una partición en específica), una inconsistencia o pérdida de información solucionada. Se deberán recuperar los siguientes bloques con la información que se encuentra en el Journaling y en el SB copy.

Para verificar los cambios se deben mostrar los reportes siguientes después de simular la perdida y por segunda vez al recuperar el sistema.

1. bitmap de árbol virtual de directorio
2. bitmap de Bloques
3. Inodos
4. Bloque de datos
5. Tree_complete
6. Tree_directorio

#Recuperando el sistema de archivos EXT3 de la partición 1

Recovery -id=>vda1

2.REPORTES

Se deberán generar los reportes con el comando rep. Este comando no necesita tener una sesión activa. Se generarán en graphviz. Se puede utilizar html dentro de los reportes si el estudiante lo considera necesario. Deberá mostrarlos de forma similar a los ejemplos mostrados.

IMPORTANTE: Esta parte es obligatoria para tener derecho a la calificación de los aspectos que muestre el reporte. Si falta alguno de los reportes no se calificará. Por ejemplo si no hace reporte de inodos, no tendrá derecho a la calificación de todos los aspectos relativos a los inodos, ya que no se puede comprobar que el estudiante haya implementado dicha funcionalidad.

Rep

Recibirá el nombre del reporte que se desea y lo generará con graphviz en una carpeta existente.

Parámetro	Categoría	Descripción
-nombre	Obligatorio	Nombre del reporte a generar. Tendrá los siguientes valores: <ul style="list-style-type: none">mbrdisksbbm_arbdirbm_detdirbm_inodebm_blockbitacoradirectoriotree_filetree_directoriotree_completels Si recibe otro valor que no sea alguno de los anteriores, debe mostrar un error.
-path	Obligatorio	Indica una carpeta y el nombre que tendrá el reporte. Si no existe la carpeta, deberá crearla. Si lleva espacios se encerrará entre comillas
-id	Obligatorio	Indica el id de la partición que se utilizará. Si el reporte es sobre la información del disco, se utilizará el disco al que pertenece la partición. Si no existe debe mostrar un error.
-ruta	Opcional	Funcionará para el reporte file y ls. Será el nombre del archivo o carpeta del que se mostrará el reporte. Si no existe muestra error

Ejemplos

```
rep -id=vda2 -path="/home/user/reports/reporte 2.pdf" -nombre=bm_arbdir -ruta="/home/mis documentos"
```

```
rep -id=vda1 -path="/home/user/reports/reporte 3.jpg" -nombre=archive_directorio
```

5.1 REPORTES DE BITMAPS

Los reportes se generarán en un archivo de texto, el formato para este reporte es criterio del estudiante, el formato SUGERIDO: "0,0,1,0,1,0" o "0|0|1|0|1|0"

Con un máximo de 20 bit de información por línea a generada. Los reportes serán:

- Reporte del Bitmap del árbol de directorio
 - Reporte del Bitmap del detalle de directorio
 - Reporte del Bitmap de tabla de i-nodos
 - Reporte del Bitmap de bloque de datos.
- bm_arbdir
 - bm_detdir
 - bm_inode

- `bm_block`

Reporte Bit_Map INODO.txt																					
1	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	1	0
2	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
3	0	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	0	1	1
4	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0
5	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	1	0
7	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
8	0	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	1	0	1
9	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0
10	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	1	0
13	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	0	0	1	1	1	0
14	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
15	0	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	1	0	1
16	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0
17	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1
18	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
19	0	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0	1
20	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	0	1	1	0
21	1	0	1	0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1
22																					

Bitacora

Se debe escribir en un archivo de texto toda la información que esta almacenada en la bitácora, el formato SUGERIDO:

Tipo Operación: “El tipo de operación a realizarse”
Tipo: “Archivo/Directorio”
Nombre: “Nombre archivo o directorio”
Contenido: “Si hay datos contenidos”
Fecha “Fecha de transacción”

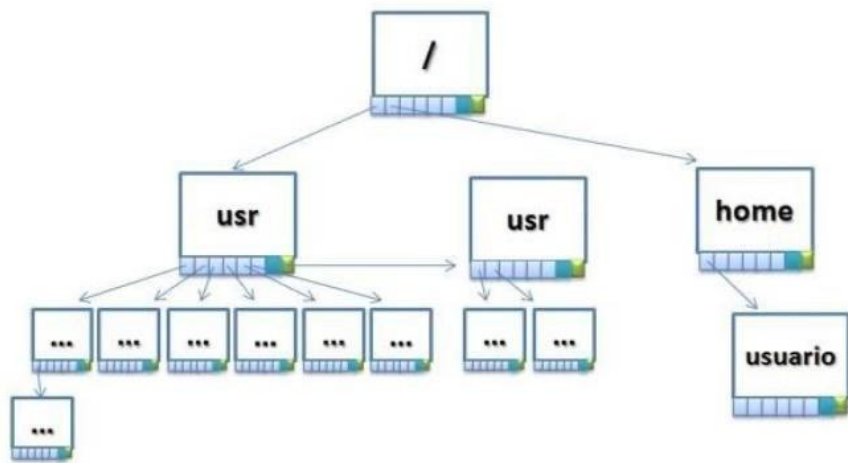
Tipo Operación: “El tipo de operación a realizarse”
Tipo: “Archivo/Directorio”
Nombre: “Nombre archivo o directorio”
Contenido: “Si hay datos contenidos”
Fecha“Fecha de transacción”

Tipo Operación: “El tipo de operación a realizarse”
Tipo: “Archivo/Directorio”
Nombre: “Nombre archivo o directorio”
Contenido: “Si hay datos contenidos”
Fecha “Fecha de transacción”

5.2 REPORTE GRÁFICO GENERAL DE DIRECTORIOS

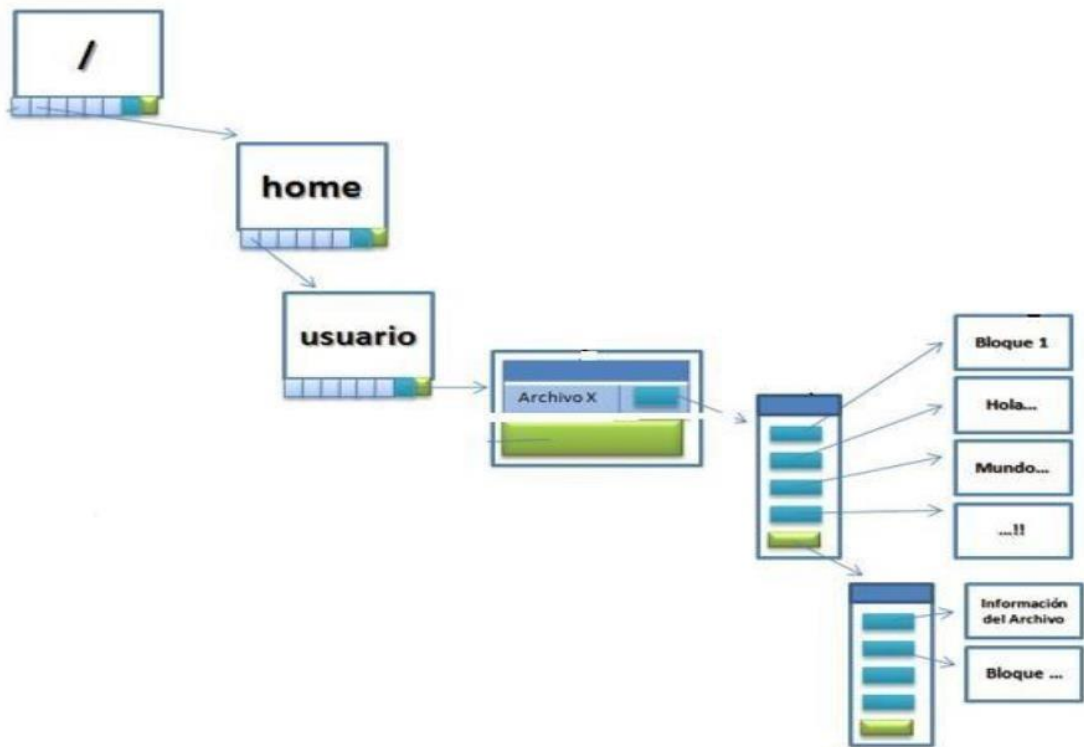
Directorio

Para verificar si la estructura lógica del sistema es la correcta, se debe de realizar una gráfica en Graphviz mostrando todos los directorios y subdirectorios mostrando su orden jerárquico similar a la siguiente:



5.3 REPORTE GRAFICO DE UN ARCHIVO Tree_File

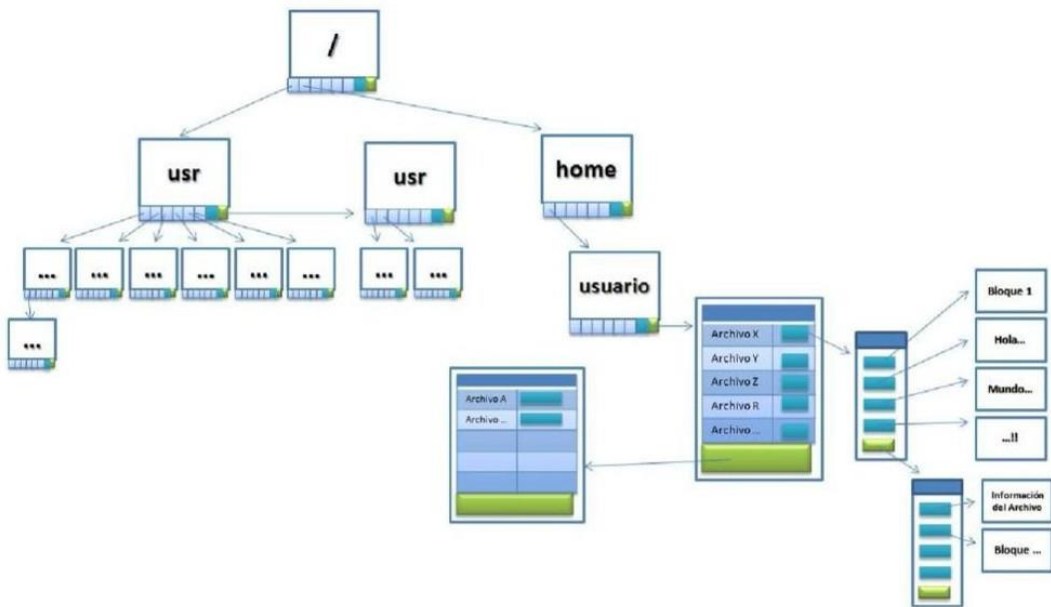
Se debe se debe de realizar una gráfica en Graphviz mostrando la ubicación de un archivo (carpeta/s) y su contenido (carácter/es). Se debe de elegir el archivo deseado por medio de un menú desde consola y la gráfica debe contener una estructura similar a la siguiente:[arreglo](Solo incluir el nombre del Archivo y sus Bloques de Informacion Asociados)



5.4 REPORTE COMPLETO DE SISTEMA

Tree_complete

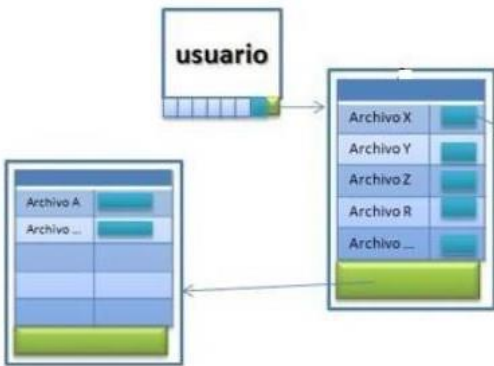
Se debe mostrar un árbol completo con todas las carpetas que se encuentran en el sistema de archivos, y los archivos que pertenecen a todo el sistema, en las carpetas correspondientes.



5.5 REPORTE GRÁFICO DE LOS ARCHIVOS DE UN DIRECTORIO

Tree_directorio

Se deben mostrar en graphviz los archivos que contiene un directorio. El directorio debe elegirse por medio de un menú desde consola. El gráfico debe ser similar al siguiente:



Disk

Este reporte mostrará la estructura de las particiones (mostrar nombre de la partición y el tipo) y el mbr del disco.

Disco1.dsk

MBR	Libre	Extendida					Primaria	Libre
		EBR	Lógica	Libre	EBR	Lógica		

LS

Este reporte mostrará la información de los archivos y carpetas con permisos, fecha de modificación, tipo, fecha de creación, reporte realizado con Consola

-rw-rw-r--	user1	Mi grupo	40661	24/02/2016	09:53	Manual.txt
-rw-r--rwx	user2	Mi grupo	123	24/02/2016	09:53	archivo1.txt
drw-rw-r--	root	root	0	24/02/2016	09:53	home

Sb

Muestra toda la información del superbloque en una tabla

Ejemplo:

SuperBloque Partición 1 en Disco1.dsk

NOMBRE	VALOR
sb_nombre_hd	Disk1
sb_arbol_virtual_count	78
sb_detalle_directorio_count	128
sb_inodos_count	64
sb_bloques_count	50
sb_arbol_virtual_free	180
sb_detalle_directorio_free	128
sb_inodos_free	256
sb_bloques_free	128
sb_date_creacion	01/12/2016 15:38
sb_date_ultimo_montaje	30/12/2016 15:38
sb_montajes_count	52
sb_ap_bitmap_arbol_directorio	100
sb_ap_arbol_directorio	56
sb_ap_bitmap_detalle_directorio	98
sb_ap_detalle_directorio	5631
sb_ap_bitmap_tabla_inodo	200
sb_ap_tabla_inodo	600
sb_ap_bitmap_bloques	10
sb_ap_bloques	100
sb_ap_log	200
sb_size_struct_arbol_directorio	128
sb_size_struct_detalle_directorio	64
sb_size_struct_inodo	50
sb_size_struct_bloque	180
sb_first_free_bit_arbol_directorio	128
sb_first_free_bit_detalle_directorio	328
sb_first_free_bit_tabla_inodo	630
sb_first_free_bit_bloques	20000
sb_magic_num	201020576

Limitaciones:

Para calcular el Tamaño total del sistema y Numero de Estructuras deben utilizar la siguiente Formula:

$$\text{NumeroDeEstructuras: } (\text{TamañoDeParticion} - (2 * \text{TamañoDelSuperbloque})) / \\ (27 + \text{TamArbolVirtual} + \text{TamDetalleDirectorio} + (5 * \text{TamInodo}) + (20 * \text{TamBloque}) + \text{Bitacora})$$

En Base Al Numero de Estructuras deben de realizar todas las actividades relacionadas con el manejo del Sistema de Archivos.(Se Explicara en los Laboratorios con mejor Detalle)

ESPECIFICACIONES

- El lenguaje a utilizar es Go
- El sistema operativo a utilizar es cualquier distribución de GNU/Linux (instalación física, solo se calificará la práctica sobre una distribución de GNU/Linux).
- El manual técnico deberá incluir una especificación de lo que realiza cada método de la aplicación y debe coincidir con el código del proyecto entregado (esto es solo parte del contenido).
- Manual de usuario.
- El proyecto se realizará de forma individual.
- Los Reportes Gráficos son OBLIGATORIOS para poder tener derecho a calificación, ya que es el único medio para verificar el funcionamiento del sistema de archivos LWH.
- Si no ejecuta con comando exec NO SE CALIFICARÁ

CONSIDERACIONES A TOMAR EN CUENTA

- El Proyecto se entrega en la fecha indicada, se tomará en cuenta que la impuntualidad en la entrega afectará la nota, así como también queda a discreción del Auxiliar el recibir algo fuera del tiempo estipulado
- Se verificarán copias de código, cualquier copia del Proyecto será sancionada con la pérdida automática del Laboratorio y la sanción correspondiente por parte de la Escuela de Sistemas.
- No se permite la modificación de código durante la calificación. El estudiante no tendrá acceso al código fuente durante la calificación
- El archivo binario que representa a los discos no debe crecer
- No se permite la utilización de estructuras en memoria (listas, arboles, etc.) para el manejo de los archivos o carpetas

- Se debe crear un repositorio PRIVADO y compartir con los auxiliares de laboratorio, en la calificación se confirmará que el ultimo commit coincida con la fecha de entrega o antes. Usuarios de Github de los auxiliares: A- : LuisLVar - A+: hernandezgramajo

ENTREGA Y CALIFICACION

- **Fecha de entrega de Proyecto: 13 de septiembre de 2020.**
- **Entregable: Link del repositorio de github, RAR/ZIP del proyecto con el formato**
- **[MIA]Proyecto1_<carnet>, manual de usuario, manual técnico.**

Se requiere que tengan como mínimo lo siguiente para poder realizar la calificación:

- Comandos(Exec)
- Reportes
- Particiones (primarias, extendidas)
- Mount y Unmount
- Crear carpetas y archivos
- Pausa
- Perdida y Recuperacion