Espacio de Definición Institucional

Práctica 1, Software Libre, Cultura Libre, Libertad de Conocimiento, GNU/Linux, Comandos de Terminal Básicos y Github.

# Software Libre

* 1. Explique con sus palabras qué es el Software Libre y dé tres ejemplos.

* 1. Explique con sus palabras las ventajas y desventajas del Software Libre.
  2. Enuncie y explique detalladamente las 4 libertades del Software Libre.
  3. Investigue la historia del software libre, quienes fueron sus creadores y qué los motivó.
  4. Explique qué es la Free Software Foundation y cuáles son sus objetivos y metas a futuro principales.
  5. Explique por qué cree usted que la sociedad no conoce o no utiliza software libre.
  6. Haga una encuesta a tres personas (conocidas o no) sobre software libre, haga al menos tres preguntas a elección.
  7. Investigue qué es el Flisol y en qué consiste.

# Cultura Libre

* 1. Explique con sus palabras qué es la Cultura Libre.
  2. Haga un breve resumen de los inicios de la Cultura Libre.
  3. Vea la película documental El Hijo de Internet y escriba una conclusión personal sobre lo sucedido en la película.
  4. Explique qué es el Free Culture Movement.
  5. Explique qué es Creative Commons
  6. Enuncie y explique detalladamente las distintas licencias que permite Creative Commons. De al menos 3 ejemplos de recursos (software, hardware, textos u organizaciones) que utilicen las licencias creative commons.
  7. Explique la importancia de la libertad de conocimiento según usted.

# GNU/Linux

* 1. Explique qué es GNU/Linux.
  2. Explique brevemente qué es un Sistema Operativo, cite al menos 4 ejemplos.
  3. Mencione al menos 3 ventajas y 3 desventajas de utilizar GNU/Linux.
  4. Explique brevemente la historia de GNU/Linux.
  5. Explique qué es la consola o la terminal de GNU/Linux y para qué sirve.
  6. Explique qué es el Kernel o núcleo de GNU/Linux
  7. Explique qué es el Sistema de Archivos de GNU/Linux.

# Bash - Terminal - Consola GNU/Linux

* 1. Pruebe los siguientes comandos y explique qué hacen:

|  |  |  |
| --- | --- | --- |
| Cd | More | Ifconfig |
| Cat | Less | Whoami |
| Touch | Find | Who |
| Mkdir | grep | Netstat |
| Rm | History | Nc |
| Mv | Chmod | Kill |
| Cp | Man | Init 0 |
| Head | Top |  |
| Tail | Ps |  |

# Github

* 1. Explique qué es Git y para qué sirve.
  2. Explique qué es Github y para qué sirve.
  3. Explique los comandos básicos para subir y bajar archivos de un repositorio en Github.

***RESPUESTAS***

Capitulo Software libre

1

1. El software libre es aquel que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Para estudiarlo y modificarlo la distribución del Software Libre debe incluir el código fuente, característica fundamental.

El software libre suele estar disponible gratuitamente, pero no hay que asociar software libre a software gratuito, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque conserve su carácter de libre, puede ser vendido comercialmente.

Ejemplos:

VLC (reproductor multimedia)  
Audacity (editor de sonido)  
LibreOffice (Suite ofimática)

*B . VENTAJAS:*

No tenemos que pagar

* Queda bajo nuestro buen criterio el uso que le demos
* La gran comunidad existente
* El desarrollo de código abierto produce programas con menos errores
* Programas más eficientes en el uso de recursos que el software propietario
* Capacidad para modificar el software de modo que se adapte a las necesidades específicas del usuario

DESVENTAJAS:

Sigue siendo una opción minoritaria entre los consumidores particulares

* No está respaldado por grandes campañas publicitarias en televisión
* No se ven hileras de paquetes de atractivos colores en los estantes de las tiendas de computación
* La inexistencia de cursos de capacitación de los mismos

c. Libertad 0. Libertad de ejecutar el programa como desees.  
Libertad 1. Libertad de estudiar el codigo fuente del programa y realizar los cambios que desee el programador.  
Libertad 2. Libertad de ayudar a tu programa. Crear copias exactas del programa y distribuirlas bien gratuitamente o con la libertad de vender tu desarrollo. Estas dos opciones estan permitidas por la comunidad.  
Libertad 3. Libertad para contribuir a tu Comunidad. Hacer copias de tus versiones modificadas y distribuirlas entre los demas.

Con estas 4 libertades estamos contribuyendo a un Sistema Etico que respeta las Libertades del usuario y de la comunidad. Si no se dan las 4 Libertades o no se cumplen estamos antes un Software Privativo. Este tipo de software para la comunidad no es justo. Ya que el Software Libre puede ser controlado para y por la Comunidad. Si no quieres compartir el software es preferible que no desarrolles nada. Mejor compartir tus conocimientos con la Comunidad.  
  
La Libertad 0. Nos permite tener el control sobre nuestra informatica, sobre el mundo informatico en que nos movemos.   
La Libertad 2. Nos permite ayudar a nuestro prójimo. De esta forma mantenemos nuestra ética con nuestros amigos y con la Comunidad. Para no caer en el dilema de compartir nuestra Licencia de software privativo lo mejor y ensencial es utilizar el Software Libre de tal forma que si conoces algun Desarrollo o creas algun Desarrollo puedas compartirlo sin ningun dilema moral ni etico.

D. El software Libre tiene como finalidad de que nadie, absolutamente nadie, se apropie de la propiedad intelectual de éste. Es decir, el software libre no pertenece a nadie, pertenece a todos los que lo desarrollan y a los que lo utilizan, y todos pueden contribuir para mejorarlo.Evelio Martínez.

Introducción  
Últimamente en el mundo de la computación se ha escuchado con regularidad el término Software Libre. Más que un movimiento liberador del software, es una filosofía de compartir aquello que nos es útil, en este caso el código fuente o programas de cómputo. Estas aplicaciones o piezas de software nos facilitan una serie de actividades en nuestra computadora sin pagar algún costo por utilizarlas.

Hoy en día disponemos de una gran variedad de opciones en cuanto a software se refiere. Podemos emplear programas comerciales que nos facilitan el realizar tareas como escribir una carta, editar alguna fotografía o enviar un correo electrónico. Sin embargo ¿Qué ocurre cuando queremos compartir ese mismo software con algún amigo?, ¿Qué pasa si queremos modificarlo o pagar para que alguien más lo modifique por nosotros con el fin de adaptarlo a nuestras necesidades? Simplemente no es posible, por que no tenemos acceso al código fuente; y si distribuimos dicho software sin el permiso del autor (o autores) estamos incurriendo en un delito. Es aquí donde tiene cabida el movimiento del software libre, software cuya distribución, uso y modificación es perfectamente legal y no solo eso, si no que además nos incita a “compartir” como parte fundamental de su filosofía, siempre a favor de mejorar el software.

Orígenes del Software Libre  
La primera generación de computadoras aparece a finales de la década de 1940. Eran de enormes dimensiones y muy costosas. El poder computacional era muy pobre comparado con las computadoras de la actualidad. La relación entre el hardware y el software era demasiado estrecha, los programas se escribían de una manera bastante especializada (lenguaje de máquina) y por lo tanto, el concepto de software como una parte “independiente” del hardware se veía todavía muy lejano. Debido precisamente a esta relación entre hardware-software, las personas que operaban las computadoras debían de poseer cierto nivel de conocimientos sobre el funcionamiento de las mismas, así como de los programas que necesitaban para hacerlas funcionar. En ese entonces no existían los usuarios convencionales, todos eran usuarios especializados, en su gran mayoría científicos o ingenieros.

Entre esos usuarios expertos, era muy común que se diera el intercambio de programas así como el compartir mejoras hechas a los mismos. A estas mejoras en el software se les conoce como hacks y a estos primeros expertos o gurus de la programación se les empezó a llamar hackers. Término que en la actualidad se ha ido desvirtuando, confundiéndolos con delincuentes informáticos. En general a los hackers les interesa conocer el funcionamiento detallado de los sistemas informáticos y de su seguridad, manteniendo una actitud ética. Algunos traspasan esta línea y se convierten en lo que la comunidad hacker ha denominado cracker.

En los inicios de la computación, 40s y 50s, no existían las licencias de software. El software era libre y los programas se intercambiaban como se hace con las recetas de cocina. Este espíritu perduró en la comunidad de programadores durante años como algo natural, hasta que con el tiempo las restricciones derivadas de licencias de uso, implementadas por desarrolladores de software y las grandes compañías, plantearon la necesidad de fijar una línea divisora entre el software libre y el software propietario.

E. La Free Software Foundation o Fundación por el Software Libre es una organización creada en octubre de 1985 por [Richard Stallman](https://es.wikipedia.org/wiki/Richard_Stallman) y otros entusiastas del [software libre](https://es.wikipedia.org/wiki/Software_libre) con el propósito de difundir este movimiento.La Free Software Foundation considera que el objetivo principal del Proyecto GNU ha sido alcanzado con la ayuda de otros programas de software libre ajenos a GNU, por lo que el desarrollo de un sistema operativo puramente GNU se ha vuelto una tarea secundaria al crecimiento y manutención del sistema existente ([GNU/Linux](https://es.wikipedia.org/wiki/GNU/Linux)), y secundaria a la garantía de la libertad de sus usuarios y la promoción de tales ideas. Richard Stallman así como muchos otros partidarios del movimiento del software libre considera al sistema operativo GNU como un medio técnico para un fin social.

***Encuesta : Persona 1 – cuñado***

*-¿Conoces que es linux?-*

si, es un sistema operativo, como windows.

*-¿Utilizas frecuentemente Linux?, ¿si no lo usas, cual es la razón?-*

no, las maquinas vienen con windows.

*¿Recomendarías usar Linux a alguien?*

No lo sabe.

***Encuesta hermana***

*-¿Conoces que es linux?-*

si, es un sistema operativo.

*-¿Utilizas frecuentemente Linux?, ¿si no lo usas, cual es la razón?-*

no, es muy difícil.

*¿Recomendarías usar Linux a alguien?*

depende de las necesidades de la persona

***Encuesta mi papá***

*-¿Conoces que es linux?-*

no.

*-¿Utilizas frecuentemente Linux?, ¿si no lo usas, cual es la razón?-*

yo no uso nada de eso.

*¿Recomendarías usar Linux a alguien?*

Lo recomendaría, va no lo sé.

H. El Festival Latinoamericano de Instalación de Software Libre (FLISoL), es el mayor evento de difusión del Software libre que se realiza desde el año 2005 en diferentes países de manera simultánea. Está dedicado para personas interesadas en conocer más acerca del software y la cultura libre. Para ello, dependiendo de cada sede, son organizadas charlas, talleres, stands, y otras actividades, para que los asistentes entren en contacto con el mundo del software libre, conozcan a otros usuarios, resuelvan dudas e interrogantes, intercambien opiniones y experiencias.

*Sus principales objetivos son:*

* + Dar a conocer el Software Libre de mano de su representante más conocido, GNU con linux, o por medio de la instalación de Software Libre como LibreOffice, GIMP, Firefox, VLC Media Player, 7-Zip, Audacity entre otros. Durante el evento se instala de manera gratuita el sistema operativo y las alternativas libres a todas las personas que deseen y puedan llevar sus equipos a las sedes designadas por la directiva del festival. Durante el mismo se ofrecen de manera paralela charlas, ponencias y talleres, sobre temas locales, nacionales y latinoamericanos, relacionadas al Software Libre y al movimiento del mismo nombre.

***Cultura libre 2***

1. La cultura libre es una corriente de pensamiento, contrapartida del “derecho de autor” y es una forma de producir contenido con la particularidad de que ese contenido puede ser compartido de forma libre. La cultura libre está conformada por cuatro corrientes de pensamiento: El dominio público, el Copyleft, las Licencias Creative Commons y el software libre. Las obras en dominio público pueden utilizar también formatos libres.
2. Desde la antigüedad no existían los derechos de autor, todos los documentos existían bajo el dominio público y todas las personas podían copiar y citar los textos sin ningún impedimento, esta práctica permitió preservar muchos documentos de la antigüedad que pudieron ser eliminados en las quemas de libros de emperadores y gobiernos autoritarios. No fue si no hasta finales 1710 cuando inician las leyes de derechos de autor.Luego de crearse los derechos de autor, la mayoría de países contaban con una duración de estos derechos de 50 años. En 1998, el Congreso de los Estados Unidos aprobó la Sonny Bono Copyright Term Extension Act la cual fue firmada como ley por el presidente Clinton. La legislación extiende la protección de derechos de autor por veinte años más, lo que resulta en un plazo de copyright global garantizada de setenta años después de la muerte del creador. El proyecto de ley fue presionado fuertemente por empresas como Disney, y apodado como la ley de protección de Mickey Mouse. Lawrence Lessig afirma que el Derecho de autor es un obstáculo para producir cultura, compartir conocimiento e innovar tecnológicamente, y que esos intereses privados se oponen al bien común. Lessig viaja durante el año 1998 a través del país, dando cientos de discursos al año en los campus universitarios, y dando a conocer el movimiento. Lo cual generó la fundación del primer capítulo de Estudiantes por la Cultura Libre del Swarthmore College.La batalla entre los derechos de autor , Derecho de autor y Dominio público.En 1999, Lessig impugnó la Bono Act, llevando el caso a la Corte Suprema de los Estados Unidos. A pesar de su firme convicción por la victoria, y citando un lenguaje sencillo de la Constitución acerca de las "limitaciones" de los términos de Derechos de Autor, Lessig sólo obtuvo dos votos en favor; de los Jueces Stevens y Breyer.En 2001, Lessig inicia Creative Commons, una alternativa de "algunos derechos reservados" al existente método de "todos los derechos reservados" de los Derechos de Autor.
3. El documental: “el hijo de Internet”, nos narra la lucha de Aaron Swartz (1986-2013) por lograr la libertad del uso de las redes como método de expresión, como herramienta para educar a la sociedad y de esta manera lograr un progreso en conjunto. Swartz llegó a utilizar las redes como método de lucha política en contra de las leyes que buscaban romper el foco principal de los sitios web, la oportunidad de emitir un juicio libre.

Aaron Swartz puede ser visto como un hacker en el sentido original de la expresión. Un aficionado por códigos y por el intercambio de los desafíos intelectuales que lograba superar.

La conclusión principal que podemos tener es la necesidad de expresarnos y luchar por las cosas que podemos cambiar, no permitir el abuso y dar cuenta al mundo que el legado de personas como Aaron Swartz, no queda en el olvido y nos ayuda a levantar mentes hacia “un mundo mejor”.

Su vida también estuvo del lado del activismo en donde la lucha contra la ley sopa y pipa lograron a un swartz de 25 años que se podía vencer al poder reinante, pero a raíz de sus actividades llamo la atención del gobierno que reacciono hostigándolo y atacándolo con la acusacion de más de 18 crímenes y una fianza millonaria que logro la decisión de quitarse la vida. Es un final amargo en donde me hace pensar que los bueno tipos no tienen lugar en este mundo, pero su historia de vida me hace recapacitar en que el dejo en muchas personas un espíritu de lucha ya que desde sus inicios siempre vio las cosas desde su humanidad, hacer para ayudar.

1. FreeCulture.org, denominada también FreeCulture (en español cultura libre), The Free Culture Movement (movimiento de la cultura libre), y otras variaciones respeto del tema de la cultura libre, es una organización estudiantil internacional que tiene como objetivo promover los ideales de la cultura libre, como la participación cultural y el acceso a la información. Recibió su inspiración de la obra del profesor de derecho de la Universidad de Stanford, Lawrence Lessig, quien escribió el libro Free Culture. Tiene una colaboración frecuente con otras ONG de cultura libre, incluyendo Creative Commons.

FreeCulture.org tiene presencia en más de 30 campus universitarios alrededor del mundo y cuenta con una historia de activismo.

1. Creative Commons (CC) ―en español, "Comunes Creativos"― es una organización sin fines de lucro dedicada a promover el acceso y el intercambio de cultura. Desarrolla un conjunto de instrumentos jurídicos de carácter gratuito que facilitan usar y compartir tanto la creatividad como el conocimiento.​ Su sede central se encuentra en Mountain View, en el estado de California, Estados Unidos.
2. Las licencias Creative Commons o CC están inspiradas en la licencia GPL (General Public License) de la Free Software Foundation, y comparten buena parte de su filosofía. La idea principal detrás de ellas es posibilitar un modelo legal ayudado por herramientas informáticas, para así facilitar la distribución y el uso de contenidos.

Existe una serie de licencias Creative Commons, cada una con diferentes configuraciones, que permite a los autores poder decidir la manera en la que su obra va a circular en internet, entregando libertad para citar, reproducir, crear obras derivadas y Creative Commons International. Existen varios países de habla hispana que están involucrados en este proceso: Argentina, Chile, Colombia, Ecuador, El Salvador, España, Guatemala, México, Perú y Puerto Rico que ya tienen las licencias traducidas y en funcionamiento, en tanto que Venezuela se encuentra en proceso de traducción e implementación de las mismas. Asimismo, Brasil también tiene las licencias traducidas y adaptadas a su legislación.

Todas las licencias Creative Commons permiten el “derecho fundamental” de redistribuir la obra con fines no comerciales y sin modificaciones. Las opciones NC y ND hacen que la obra no sea libre de acuerdo con la definición de obras culturales libres.

Una licencia contractual especial es la opción CC0, o “Sin derechos reservados”. Esta licencia cede la obra al dominio público (o un estado equivalente en jurisdicciones donde el dominio público no es posible). Comparado a una declaración de dominio público cedida a la obra, la declaración CC0 es menos ambigua y consigue el efecto deseado a escala global, en lugar de limitarse a algunas jurisdicciones.

1. El conocimiento que he adquirido lo he hecho mediante instigar en la web y mirar cosas, leer información y intentarlo aunque no siempre sale bien, hay que saber leer y decodificar.

GNU/Linux

1. GNU/Linux, es el término empleado para referirse a la combinación del sistema operativo GNU, desarrollado por la FSF, y el núcleo(kernel) Linux, desarrollado por Linus Torvalds y la Linux Foundation. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y

redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU) y otra serie de licencias libres.

A pesar de que «Linux» denomina en la jerga cotidiana al sistema operativo, este es en realidad solo el Kernel (núcleo) del sistema. La verdadera denominación del sistema operativo es "GNU/Linux" debido a que el resto del sistema (la parte fundamental de la interacción entre el hardware y el usuario) se maneja con las herramientas del proyecto GNU (www.gnu.org) y con entornos de escritorio (como GNOME), que también forma parte del proyecto GNU aunque tuvo un origen independiente.

Como el Proyecto GNU destaca, GNU es una distribución, usándose el término sistema operativo en el sentido empleado en el ecosistema Unix, lo que en cualquier caso significa que Linux es solo una pieza más dentro de GNU/Linux. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos.

1. Un Sistema operativo es un software que sirve de intermediario entre el hardware y el usuario, sin un sistema operativo una computadora sería una caja tonta, Los SO administran los recursos, como puede ser la memoria, los ciclos de reloj, los periféricos.

Actualmente los sistemas operativos se componen de una interfaz gráfica llamada escritorio, un Nucleo que es el encargado de interactuar entre el hardware y los servicios del sistema, y las aplicaciones minimas para su funcionamiento, ejemplos de sistemas operativos puede ser , Gnu/Linux y sus distros

1. Las ventajas de gnu/Linux es que no es necesario comprar una licencia para usarlo, además se puede customizar para adaptarse a las necesidades de cada usuario, otro pro es que es menos propenso a ser infectado por los virus.

Las desventajas son que posee menos programas que su competidor Windows, su arco de aprendizaje es pronunciado y es poco amistoso con los usuarios novatos, y problemas con la implentacion de formatos y productos cerrados.

1. El sistema operativo Unix fue concebido e implementado en 1969 en los laboratorios Bell de AT&T en los Estados Unidos por Ken Thompson, Dennis Ritchie, Douglas McIlroy, y Joe Ossanna.11​ Por primera vez en 1971, Unix fue escrito completamente en lenguaje ensamblador, ya que era una práctica común en el momento. Más tarde, en un enfoque innovador clave en 1973, se volvió a escribir en el lenguaje de programación C por Dennis Ritchie (a excepción del núcleo y sistema I/O). La disponibilidad de un lenguaje de alto nivel de implementación de Unix hizo su portabilidad a diferentes plataformas de computación más fácil.

Debido a un anterior caso antimonopolio que prohíbe que entren en el negocio de las computadoras, AT&T licencio el código fuente del sistema operativo para cualquier persona que le pidió.12​ Como resultado, Unix creció rápidamente y llegó a ser ampliamente adoptado por las instituciones académicas y empresas. En 1984, AT & T se despojó de Bell Labs; quedando libre de la obligación legal que requiere la concesión de licencias libres, Bell Labs comenzó a vender Unix como propietaria del producto.

El proyecto GNU, iniciado en 1983 por Richard Stallman, tiene el objetivo de crear un "sistema de software compatible con Unix" compuesto enteramente de software libre. El trabajo comenzó en el año 1984.13​ Más tarde, en 1985, Stallman comenzó la Free Software Foundation y escribió la Licencia Pública General de GNU (GNU GPL) en 1989. A principios de la década de 1990, muchos de los programas que se requieren en un sistema operativo (como bibliotecas, compiladores, editores de texto, un shell Unix, y un sistema de ventanas) se completaron, aunque los elementos de bajo nivel, tales como los controladores de dispositivos y los daemons se estancaron y quedaron incompletos.

Linus Torvalds ha declarado que si el núcleo de GNU hubiera estado disponible en el momento (1991), no se habría decidido a escribir su propio núcleo

Aunque no es liberado hasta 1992 debido a complicaciones legales, el desarrollo de 386BSD , de los cuales NetBSD, OpenBSD y FreeBSD descienden, es anterior a la de Linux. Torvalds también ha declarado que si 386BSD hubiera estado disponible en ese momento, probablemente no habría creado Linux.

MINIX fue creado por Andrew S. Tanenbaum, profesor de ciencias de la computación, y lanzado en 1987 como mínimo similar al sistema operativo Unix, dirigido a estudiantes y otras personas que querían aprender los principios del sistema operativo. Aunque el código fuente completo de MINIX era de libre disposición, los términos de la licencia impidió su libre distribución hasta que el otorgamiento de licencias cambió en abril de 2000.

1. Una Shell de Unix o también shell, es el término usado en informática para referirse a un intérprete de comandos, el cual consiste en la interfaz de usuario tradicional de los sistemas operativos basados en Unix y similares como GNU/Linux.

Mediante las instrucciones que aporta el intérprete, el usuario puede comunicarse con el núcleo y por extensión, ejecutar dichas órdenes, así como herramientas que le permiten controlar el funcionamiento de la computadora.

Los comandos que aportan los intérpretes, pueden usarse a modo de guion si se escriben en ficheros ejecutables denominados shell-scripts, de este modo, cuando el usuario necesita hacer uso de varios comandos o combinados de comandos con herramientas, escribe en un fichero de texto marcado como ejecutable, las operaciones que posteriormente, línea por línea, el intérprete traducirá al núcleo para que las realice. Sin ser un shell estrictamente un lenguaje de programación, al proceso de crear scripts de shell se le denomina programación shell o en inglés, shell programming o shell scripting.

1. En informática, un núcleo o kernel (de la raíz germánica Kern, núcleo, hueso) es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá usar un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporcionar una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador.
2. El sistema de archivos o sistema de ficheros es el componente del sistema operativo encargado de administrar y facilitar el uso de las memorias periféricas, ya sean secundarias o terciarias.

Sus principales funciones son la asignación de espacio a los archivos, la administración del espacio libre y del acceso a los datos resguardados. Estructuran la información guardada en un dispositivo de almacenamiento de datos o unidad de almacenamiento (normalmente un disco duro de una computadora), que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos. La estructura de directorios suele ser jerárquica, ramificada o "en árbol", aunque en algún caso podría ser plana. En algunos sistemas de archivos los nombres de archivos son estructurados, con sintaxis especiales para extensiones de archivos y números de versión. En otros, los nombres de archivos son simplemente cadenas de texto y los metadatos de cada archivo son alojados separadamente.

En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada ruta (path, en inglés). La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una barra diagonal (/) o barra diagonal invertida (\) y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

|  |  |  |
| --- | --- | --- |
| Cd: saltar de directorio | More: muestra archivos paso a paso | Ifconfig: gestionar las interfaces de red. |
| Cat: crear archivos,modificiar | Less: parecido a more | Whoami: muestra nombre del usuario |
| Touch: Crear archivos | Find: busca archivos, lista todos los archivos de la carpeta | Who: listar los nombres de los usuarios conectados actualmente. |
| Mkdir: crea una carpeta | Grep: filtrar resultados de un comando o archivos. | Netstat |
| Rm | History | Nc |
| Mv mover archivo | Chmod | Kill eliminar |
| Cp | Man pagina de manual | Init 0 apagar la maquina |
| Head | Top muestra los procesos en ejecucion |  |
| Tal | PsPID TTY TIME CMD  7994 pts/1 00:00:00 bash  8013 pts/1 00:00:00 ps |  |

GIT 4

Es un software de [control de versiones](https://es.wikipedia.org/wiki/Control_de_versiones) diseñado por [Linus Torvalds](https://es.wikipedia.org/wiki/Linus_Torvalds), pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de [código fuente](https://es.wikipedia.org/wiki/Código_fuente). Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o [front end](https://es.wikipedia.org/wiki/Front-end_y_back-end) como [Cogito](https://es.wikipedia.org/wiki/Cogito_(software)) o [StGIT](https://es.wikipedia.org/w/index.php?title=StGIT&action=edit&redlink=1). Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de [programación](https://es.wikipedia.org/wiki/Programación) del [núcleo Linux](https://es.wikipedia.org/wiki/Núcleo_Linux).

GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

En nuestra especialidad“Programación”, fuimos aprendiendo cosas y creando programas de código abierto, fomentando el software libre; es por eso que presentamos esta gran herramienta enfocada al crecimiento de proyectos comunitarios y libres.

Comandos Github:

Git Status: para comprobar si los arivos tuvieron cambio o directamente no estan en la rama máster.

Git add . O Git add nombre del archivo: para ponerlo en cola de subída al git hub.

Git commit -m y entre “ “ el comentario del archivo: es para comitiar darle un comentario de subida al archivo.

Git push origin master: para subirlo al github.