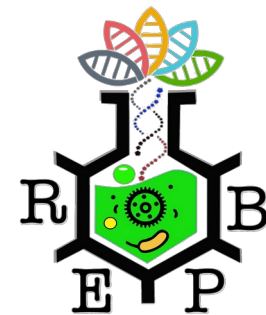




Red de Egresados del Posgrado en Biociencias  
de la Universidad Autónoma de Guerrero



# Introducción a Linux

M en BC. Alberto Patricio Hernández

[patriciofcqb@gmail.com](mailto:patriciofcqb@gmail.com)

Clase 2



## Clase 2



¿Que es Linux?

¿Vale la pena aprender a usarlo?

¿Que aprendimos en la clase 1?

## Dia 2

### 4. Gestión de Archivos y Directorios

- Manejo de archivos y directorios en la línea de comandos.
- Permisos de archivo y seguridad.
- Uso de comandos como ls, cp, mv, rm, chmod, etc.

### 5. Administración de Paquetes

- Gestión de software en Linux: instalación, actualización y eliminación de programas.
- Introducción a gestores de paquetes como APT (Advanced Package Tool) o

### 6. Introducción a la Programación en Linux

- Editores de texto en Linux (por ejemplo, Vim, Nano).
- Compilación de programas desde la línea de comandos.
- Introducción a la programación en shell (bash scripting).

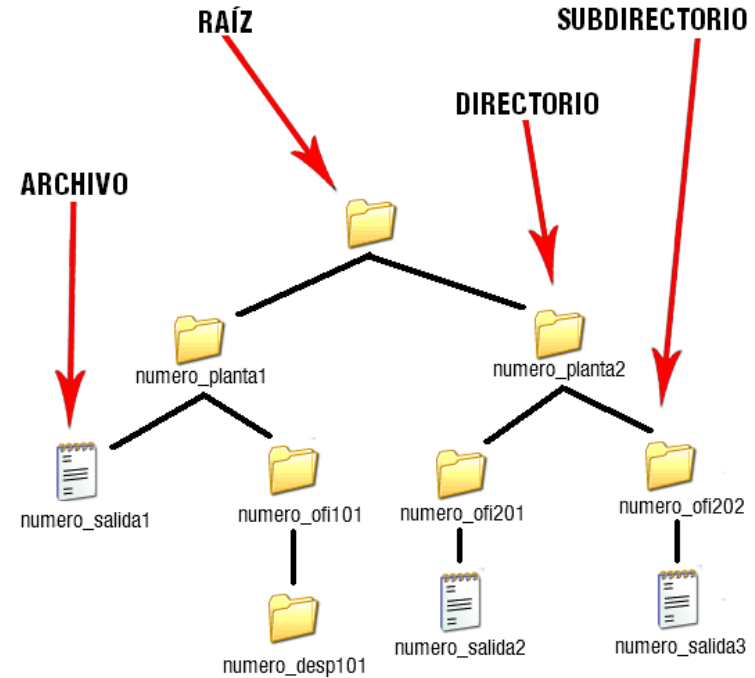
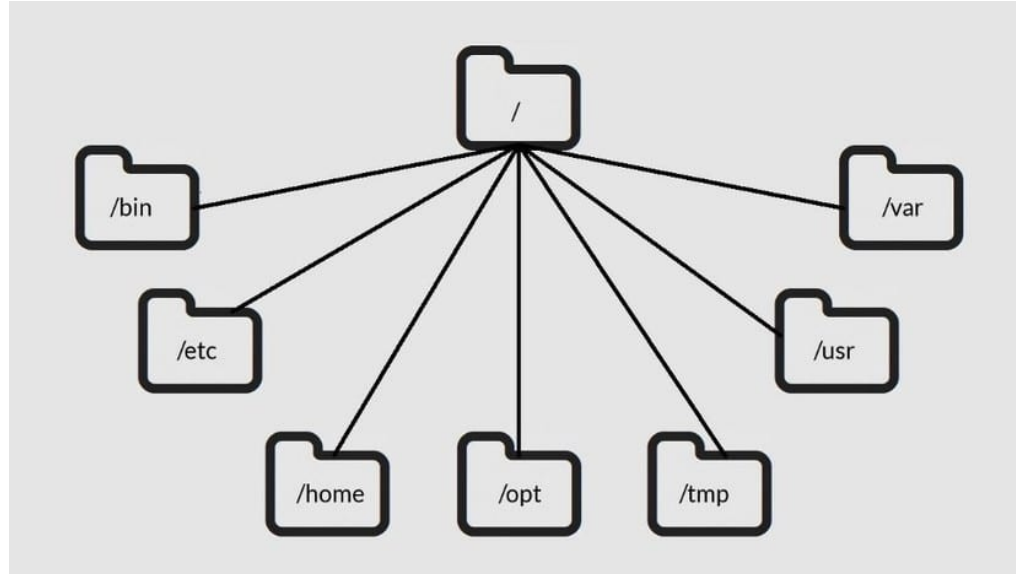
### 7. Redes en Linux

- Configuración de red: asignación de direcciones IP, configuración de interfaces de red.
- Herramientas de red en Linux: ifconfig, ping, traceroute, netstat, etc.

### 8. Introducción a la Seguridad en Linux

- Conceptos básicos de seguridad: usuarios y grupos, contraseñas seguras.
- Configuración del cortafuegos (firewall) en Linux.
- Actualizaciones de seguridad y gestión de parches.

# Gestión de archivos y directorios



## Permisos de archivo y seguridad

En Linux, hay tres permisos diferentes que puedes especificar para cada uno de los tres grupos de permisos. Estos tres permisos son Lectura, Escritura y Ejecución.

```
$ ls -l samplefile
```

| 1               | 2 | 3  | 4  | 5                       | 6                     | 7                 |
|-----------------|---|----|----|-------------------------|-----------------------|-------------------|
| -rw-r--r--      | 1 | pi | pi | 5                       | Nov 29 06:09          | samplefile        |
|                 |   |    |    |                         | Fecha de modificacion | Nombre de fichero |
|                 |   |    |    | Grupo propietario       |                       |                   |
|                 |   |    |    | Grupo propietario       |                       |                   |
|                 |   |    |    | Usuario Propietario     |                       |                   |
|                 |   |    |    | Numero de enlaces duros |                       |                   |
|                 |   |    |    | Otros permisos          |                       |                   |
|                 |   |    |    | Permisos de grupo       |                       |                   |
|                 |   |    |    | Permisos de propietario |                       |                   |
| Tipo de Fichero |   |    |    |                         |                       |                   |

## Consideraciones



Usuarios con privilegios  
mínimos



Autenticación de dos  
factores



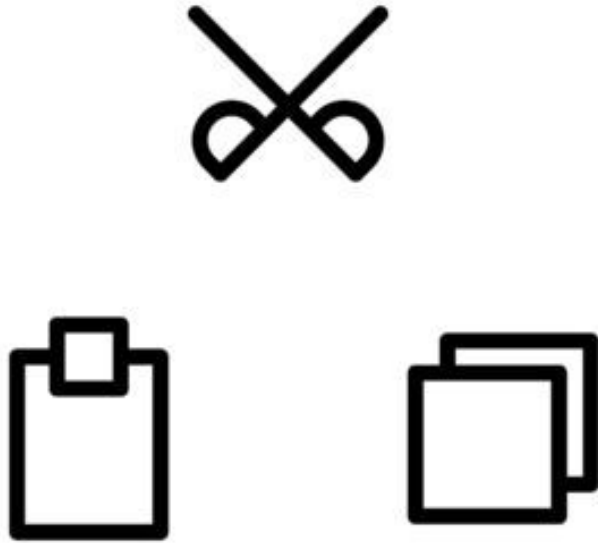
Contraseñas fuertes



Deshabilitar cuentas  
inactivas



## Introducción al Shell



El shell es la capa más externa del sistema operativo. Los shells incorporan un lenguaje de programación para controlar procesos y archivos, además de iniciar y controlar otros programas.

## Instalación de paqueterías

**automatic**

```
# apt
```

**manual**

```
# dpkg
```



**compile**

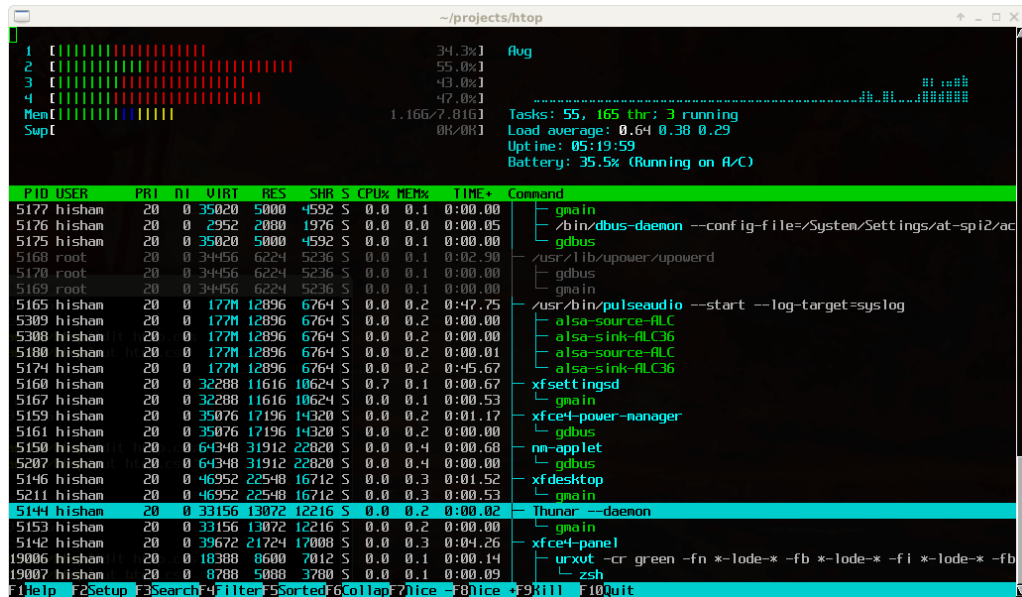
```
# make
```





# Instalación de paqueterías

htop



git

```
(base) pato@pato-Nitro:~$ git
uso: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path=<path>] [--html-path] [--man-path] [--info-path]
      [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      [--super-prefix=<path>] [--config-env=<name>=<envvar>]
      <command> [<args>]

Estos son comandos comunes de Git usados en varias situaciones:

comenzar un área de trabajo (mira también: git help tutorial)
  clone  Clonar un repositorio dentro de un nuevo directorio
  init   Crear un repositorio de Git vacío o reinicia el que ya existe

trabajar en los cambios actuales (mira también: git help everyday)
  add    Agregar contenido de archivos al índice
  mv     Mover o cambiar el nombre a archivos, directorios o enlaces simbólicos
  restore Restaurar archivos del árbol de trabajo
  rm     Borrar archivos del árbol de trabajo y del índice

examinar el historial y el estado (mira también: git help revisions)
  bisect Usar la búsqueda binaria para encontrar el commit que introdujo el bug
  diff   Mostrar los cambios entre commits, commit y árbol de trabajo, etc
  grep   Imprimir las líneas que concuerden con el patrón
  log    Mostrar los logs de los commits
  show   Mostrar varios tipos de objetos
  status Mostrar el estado del árbol de trabajo

crecer, marcar y ajustar tu historial común
  branch  Listar, crear, o borrar ramas
  commit  Grabar los cambios al repositorio
  merge   Juntar dos o más historiales de desarrollo juntos
  rebase  Volver a aplicar commits en la punta de otra rama
  reset   Reiniciar el HEAD actual a un estado específico
  switch  Cambiar de branch
  tag     Crear, listar, borrar o verificar un objeto de tag firmado con GPG

colaborar (mira también: git help workflows)
  fetch   Descargar objetos y referencias de otro repositorio
  pull    Realizar un fetch e integra con otro repositorio o rama local
  push    Actualizar referencias remotas junto con sus objetos asociados

'git help -a' y 'git help -g' listan los subcomandos disponibles y algunas
guías de concepto. Consulta 'git help <command>' o 'git help <concepto>'
para leer sobre un subcomando o concepto específico.
Mira 'git help git' para una vista general del sistema.
```



