

	<b>Carátula para entrega de prácticas</b>
<b>Facultad de Ingeniería</b>	<b>Laboratorios de docencia</b>

# Laboratorio de Computación Salas A y B

---

<i>Profesor:</i>	Ing. Karina García Morales
<i>Asignatura:</i>	Fundamentos de Programación
<i>Grupo:</i>	1121
<i>No de Práctica(s):</i>	Guía práctica de estudio 02: GNU/Linux
<i>Integrante(s):</i>	Jesús Iván Escamilla Fragoso
<i>No. de Equipo de cómputo empleado:</i>	27
<i>Semestre:</i>	19-01
<i>Fecha de entrega:</i>	28/08/18
<i>Observaciones:</i>	

# Guía práctica de estudio 02: GNU/Linux

## Objetivo:

Conocer la importancia del sistema operativo de una computadora, así como sus funciones. Explorar un sistema operativo GNU/Linux con el fin de conocer y utilizar los comandos básicos en GNU/Linux.

## Actividades:

- Iniciar sesión en un sistema operativo GNU/Linux y abrir una “terminal”
- Utilizar los comandos básicos para navegar por el sistema de archivos.
- Emplear comandos para manejo de archivos.

## Introducción

El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware (dispositivos) como de software (programas y datos) de un sistema de cómputo y/o comunicación. Además funciona como interfaz entre la computadora y el usuario o aplicaciones.

En la actualidad existen diversos sistemas operativos; por ejemplo, para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone entre otros. Cada uno de ellos tiene diferentes versiones y distribuciones que se ajustan a los diversos equipos de cómputo y comunicación en los que trabajan.

Los componentes de un sistema operativo, de forma general, son:

- Gestor de memoria,
- Administrador y planificador de procesos,
- Sistema de archivos y Administración de E/S.

## Kernel de GNU/Linux

El kernel o núcleo de linux se puede definir como el corazón del sistema operativo. Es el encargado de que el software y el hardware del equipo se puedan comunicar. Sus componentes son los que se mencionaron en la introducción de esta práctica.



## Interfaz de línea de comandos (CLI) o shell de GNU/Linux

El Shell de GNU/Linux permite introducir órdenes (comandos) y ejecutar programas en el sistema operativo. Todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de archivos y a los que llamamos comandos, por lo tanto, todo en GNU/Linux se puede controlar mediante comandos.

## Comandos básicos

Para trabajar en Linux utilizando comandos, se debe abrir una “terminal” o “consola” que es una ventana donde aparece la “línea de comandos” en la cual se escribirá la orden o comando. La terminal permite un mayor grado de funciones y configuración de lo que queremos hacer con una aplicación o acción en general respecto a un entorno gráfico.

**En esta práctica no usamos el sistema operativo Linux. Se utilizó el sistema operativo MacOSX.**

## Comando *ls* .

El comando *ls* permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux.

### *ls -l*

El comando *ls* realiza acciones distintas dependiendo de las banderas que utilice, por ejemplo, si se utiliza la opción *l* se genera un listado largo de la ubicación actual.

### *ls /*

Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos.

```
[Colombia27:~ Guest$ ls . ]
Desktop      Movies      algoritmo.txt  circunferencia  manuel.txt
Documents    Music       archivo       daniel.txt     química
Downloads    Pictures    álgebra       exposicion      temas
Library      Public     cálculo       fundamentos
[Colombia27:~ Guest$ ls -l ]
Desktop
Documents
Downloads
Library
Movies
Music
Pictures
Public
algoritmo.txt
archivo
álgebra
cálculo
circunferencia
daniel.txt
exposicion
fundamentos
manuel.txt
química
temas
[Colombia27:~ Guest$ ls / ]
Applications  anaconda3  opt
Developer     bin        private
LC_AB         cores      sbin
Library       dev        tmp
Network       etc        usr
System        home      var
Users         installer.failurerequests
Volumes       net
```

## *ls / home*

Para ver los usuarios del equipo local, revisamos el directorio *home* que parte de la raíz (/).

## *ls -l /home*

Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica.

```
archivo
álgebra
cálculo
circunferencia
daniel.txt
exposicion
fundamentos
manuel.txt
química
temas
[Colombia27:~ Guest$ ls /
Applications          anaconda3             opt
Developer             bin                   pri
LC_AB                 cores                  sbi
Library               dev                   tmp
Network               etc                   usr
System                home                  var
Users                 installer.failurerequests
Volumes               net
```

```
[Colombia27:~ Guest$ ls /home
[Colombia27:~ Guest$ ls -l /home
Colombia27:~ Guest$ █
```

## man ls

GNU/Linux proporciona el comando *man*, el cual permite visualizar la descripción de cualquier comando así como la manera en la que se puede utilizar.

```
LS(1)                                BSD General Commands Manual                                LS(1)
```

**NAME**

ls -- list directory contents

**SYNOPSIS**

ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1] [file ...]

**DESCRIPTION**

For each operand that names a file of a type other than directory, **ls** displays its name as well as any requested, associated information. For each operand that names a file of type directory, **ls** displays the names of files contained within that directory, as well as any requested, associated information.

If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

The following options are available:

- @ Display extended attribute keys and sizes in long (-l) output.
- 1 (The numeric digit 'one'.) Force output to be one entry per line. This is the default when output is not to a terminal.
- A List all entries except for . and ... Always set for the super-user.
- a Include directory entries whose names begin with a dot (.).
- B Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file names as \xxx, where xxx is the numeric value of the character in octal.
- b As -B, but use C escape codes whenever possible.
- C Force multi-column output; this is the default when output is to a terminal.
- c Use time when file status was last changed for sorting (-t) or long printing (-l).
- d Directories are listed as plain files (not searched recursively).
- e Print the Access Control List (ACL) associated with the file, if present, in long (-l) output.
- F Display a slash ('/') immediately after each pathname that is a directory, an asterisk ('\*') after each that is executable, an at sign ('@') after each symbolic link, an equals sign ('=') after each socket, a percent sign ('%') after each whiteout, and a vertical bar ('|') after each that is a FIFO.
- f Output is not sorted. This option turns on the -a option.
- G Enable colorized output. This option is equivalent to defining CLICOLOR in the environment. (See below.)
- g This option is only available for compatibility with POSIX; it is used to display the group name in the long (-l) format output (the owner name is suppressed).

## *ls /usr*

Se usa para ver la lista de los archivos del directorio *usr*.

## *ls .. o ls ../*

Si bien el punto (.) es para indicar la ubicación actual, el doble punto (..) se utiliza para referirse al directorio “padre”. Se escribe este comando si deseamos listar los archivos que dependen de mi directorio padre.

## *ls ../ ../usr*

Se usa este comando para listar los archivos del directorio *usr*.

```
Users                               installer.failurerequests
Volumes                             net
Colombia27:~ Guest$ ls /home
Colombia27:~ Guest$ ls -l /home
Colombia27:~ Guest$ man ls

[1]+  Stopped                               man ls
Colombia27:~ Guest$ ls /usr
X11      bin          lib          local      share
X11R6    include       libexec      sbin       standalone
Colombia27:~ Guest$ ls -l /usr
total 16
lrwxr-xr-x  1 root  wheel    8 Apr 25 08:43 X11 -> /opt/X11
lrwxr-xr-x  1 root  wheel    8 Apr 25 08:43 X11R6 -> /opt/X11
drwxr-xr-x 978 root  wheel 33252 Apr 26 09:34 bin
drwxr-xr-x 267 root  wheel  9078 Apr 25 13:10 include
drwxr-xr-x 312 root  wheel 10608 Apr 26 09:34 lib
drwxr-xr-x 238 root  wheel  8092 Apr 26 09:34 libexec
drwxr-xr-x  15 root  wheel   510 Apr 25 13:08 local
drwxr-xr-x 248 root  wheel  8432 Apr 26 09:34 sbin
drwxr-xr-x  47 root  wheel  1598 Apr 25 13:11 share
drwxr-xr-x  5 root  wheel   170 Mar 27 21:00 standalone
Colombia27:~ Guest$ ls ..
Guest      Shared          administrador  diplomado
Colombia27:~ Guest$ ls ../
Guest      Shared          administrador  diplomado
Colombia27:~ Guest$ ls ../../usr
X11      bin          lib          local      share
X11R6    include       libexec      sbin       standalone
Colombia27:~ Guest$
```

## Comando *touch*

Este comando permite crear un archivo de texto, su sintaxis es:

*touch nombre\_archivo[.txt]*

```
[Colombia27:~ Guest$ ls ..
Guest          Shared          administrador  diplomado
[Colombia27:~ Guest$ ls ../
Guest          Shared          administrador  diplomado
[Colombia27:~ Guest$ ls ../../usr
X11            bin                lib            local          share
X11R6          include           libexec       sbin           standalone
[Colombia27:~ Guest$ touch ivan.txt
[Colombia27:~ Guest$ ls
Desktop        Movies             algoritmo.txt   circunferencia ivan.txt
Documents      Music              archivo        daniel.txt     manuel.txt
Downloads      Pictures           álgebra        exposicion     química
Library        Public             cálculo        fundamentos    temas
```

## Comando *mkdir*

El comando *mkdir* permite crear una carpeta, su sintaxis es la siguiente:

*mkdir nombre\_carpeta*

```
Guest          Shared          administrador  diplomado
[Colombia27:~ Guest$ ls ../../usr
X11            bin                lib            local          share
X11R6          include           libexec       sbin           standalone
[Colombia27:~ Guest$ touch ivan.txt
[Colombia27:~ Guest$ ls
Desktop        Movies             algoritmo.txt   circunferencia ivan.txt
Documents      Music              archivo        daniel.txt     manuel.txt
Downloads      Pictures           álgebra        exposicion     química
Library        Public             cálculo        fundamentos    temas
[Colombia27:~ Guest$ mkdir tareas
[Colombia27:~ Guest$ ls
Desktop        Movies             algoritmo.txt   circunferencia ivan.txt     temas
Documents      Music              archivo        daniel.txt     manuel.txt
Downloads      Pictures           álgebra        exposicion     química
Library        Public             cálculo        fundamentos    tareas
[Colombia27:~ Guest$ 4
```

Creamos una carpeta llamada 'tareas' y con el comando *ls* verificamos que se haya creado.



## Comando *cd*

Este comando permite ubicarse en una carpeta, su sintaxis es:

*cd nombre\_carpeta*

Si queremos situarnos en la carpeta “tareas” creada anteriormente, se escribe el comando:

*cd tareas*

Ahora que si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos el comando:

*cd ..*

## Comando *pwd*

El comando *pwd* permite conocer la ubicación actual(ruta), su sintaxis es la siguiente: *pwd*

```
Colombia27:~ Guest$ ls ..
Guest          Shared          administrador  diplomado
Colombia27:~ Guest$ ls ../
Guest          Shared          administrador  diplomado
Colombia27:~ Guest$ ls ../../usr
X11            bin            lib            local          share
X11R6          include        libexec        sbin           standalone
Colombia27:~ Guest$ touch ivan.txt
Colombia27:~ Guest$ ls
Desktop        Movies          algoritmo.txt   circunferencia ivan.txt
Documents      Music           archivo         daniel.txt      manuel.txt
Downloads      Pictures        álgebra        exposicion      química
Library        Public          cálculo         fundamentos     temas
Colombia27:~ Guest$ mkdir tareas
Colombia27:~ Guest$ ls
Desktop        Movies          algoritmo.txt   circunferencia ivan.txt      temas
Documents      Music           archivo         daniel.txt      manuel.txt
Downloads      Pictures        álgebra        exposicion      química
Library        Public          cálculo         fundamentos     tareas
Colombia27:~ Guest$ cd tareas
Colombia27:tareas Guest$ pwd
/Users/Guest/tareas
Colombia27:tareas Guest$
```

## Comando *find*

Permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente:

*find . -name cadena\_buscar*

Al comando *find* hay que indicarle en qué parte del sistema de archivos va a iniciar la búsqueda

Si queremos encontrar la ubicación del archivo *tareas*, se escribe el siguiente comando:

*find . -name tareas*

```
Colombia27:~ Guest$ ls ../
Guest      Shared      administrador  diplomado
Colombia27:~ Guest$ ls ../../usr
X11        bin         lib            local        share
X11R6      include     libexec        sbin          standalone
Colombia27:~ Guest$ touch ivan.txt
Colombia27:~ Guest$ ls
Desktop    Movies      algoritmo.txt  circunferencia  ivan.txt
Documents  Music       archivo       daniel.txt      manuel.txt
Downloads  Pictures    álgebra       exposicion       química
Library    Public      cálculo       fundamentos     temas
Colombia27:~ Guest$ mkdir tareas
Colombia27:~ Guest$ ls
Desktop    Movies      algoritmo.txt  circunferencia  ivan.txt      temas
Documents  Music       archivo       daniel.txt      manuel.txt
Downloads  Pictures    álgebra       exposicion       química
Library    Public      cálculo       fundamentos     tareas
Colombia27:~ Guest$ cd tareas
Colombia27:tareas Guest$ pwd
/Users/Guest/tareas
Colombia27:tareas Guest$ cd ..
Colombia27:~ Guest$ pwd
/Users/Guest
Colombia27:~ Guest$ find . -ivan
find: -ivan: unknown primary or operator
Colombia27:~ Guest$ find . -name ivan
Colombia27:~ Guest$
```

## Comando *clear*

Este comando permite limpiar la consola.

```
Colombia27:~ Guest$ █
```

## Comando *cp*

Permite copiar un archivo, su sintaxis es la siguiente:

*cp archivo\_origen archivo\_destino*

```
cp: /s: Permission denied
Colombia27:~ Guest$ cp ivan.txt /Users/Guest/tareas/
Colombia27:~ Guest$ ls
Desktop      Movies      algoritmo.txt  circunferencia  ivan.txt      temas
Documents    Music       archivo        daniel.txt      manuel.txt
Downloads    Pictures    álgebra        exposicion      química
Library      Public      cálculo        fundamentos     tareas
Colombia27:~ Guest$ cd tareas
Colombia27:tareas Guest$ ls
ivan.txt
Colombia27:tareas Guest$ █
```

```
-----
Colombia27:~ Guest$ cp ivan.txt /Users/Guest/tareas/
Colombia27:~ Guest$ ls
Desktop      Movies      algoritmo.txt  circunferencia  ivan.txt      temas
Documents    Music       archivo        daniel.txt      manuel.txt
Downloads    Pictures    álgebra        exposicion      química
Library      Public      cálculo        fundamentos     tareas
Colombia27:~ Guest$ cd tareas
Colombia27:tareas Guest$ ls
ivan.txt
Colombia27:tareas Guest$ █
```

Se usa el comando `ls` para comprobar que efectivamente se copio el archivo a la carpeta seleccionada de destino.

## Comando *rm*

El comando *rm* permite eliminar un archivo o un directorio, su sintaxis es la siguiente:

*rm nombre\_archivo*

*rm nombre\_carpeta*

```
Documents      Music          archivo        daniel.txt     manuel.txt
Downloads      Pictures       álgebra        exposicion     química
Library        Public        cálculo        fundamentos    tareas
Colombia27:~ Guest$ rm ivan.txt
Colombia27:~ Guest$ ls
Desktop        Movies         algoritmo.txt   circunferencia manuel.txt
Documents      Music          archivo        daniel.txt     química
Downloads      Pictures       álgebra        exposicion     tareas
Library        Public        cálculo        fundamentos    temas
Colombia27:~ Guest$
```

Cuando la carpeta que se desea borrar contiene información, se debe utilizar la bandera **-f** para forzar la eliminación. **Si la carpeta contiene otras carpetas, se debe utilizar la opción **-r**, para realizar la eliminación recursiva.**

```
Downloads      Pictures       álgebra        exposicion     tareas
Library        Public        cálculo        fundamentos    temas
Colombia27:~ Guest$ rm tareas
rm: tareas: is a directory
Colombia27:~ Guest$
```

```
Library        Public        calculo        fundamentos    temas
Colombia27:~ Guest$ rm tareas
rm: tareas: is a directory
Colombia27:~ Guest$ rm -r tareas/
Colombia27:~ Guest$ ls
Desktop        Movies         algoritmo.txt   circunferencia manuel.txt
Documents      Music          archivo        daniel.txt     química
Downloads      Pictures       álgebra        exposicion     temas
Library        Public        cálculo        fundamentos
Colombia27:~ Guest$
```

## Actividad en casa

### Práctica 2

```
Terminal Shell Edición Visualización Ventana Ayuda
FUNDAMENTOS — -bash — 108x29
iMac-de-Ivan:~ ivan$ mkdir FUNDAMENTOS
iMac-de-Ivan:~ ivan$ ls
Desktop      Downloads    Library      Music        Public
Documents    FUNDAMENTOS Movies       Pictures
iMac-de-Ivan:~ ivan$ cd FUNDAMENTOS
iMac-de-Ivan:FUNDAMENTOS ivan$
```

1. Crea un directorio que se llame "FUNDAMENTOS" y entra a el.

```
Terminal Shell Edición Visualización Ventana Ayuda
FUNDAMENTOS2 — -bash — 108x24
iMac-de-Ivan:~ ivan$ mkdir FUNDAMENTOS
iMac-de-Ivan:~ ivan$ ls
Desktop      Downloads    Library      Music        Public
Documents    FUNDAMENTOS Movies       Pictures
iMac-de-Ivan:~ ivan$ cd FUNDAMENTOS/
iMac-de-Ivan:FUNDAMENTOS ivan$ mkdir FUNDAMENTOS2
iMac-de-Ivan:FUNDAMENTOS ivan$ cd FUNDAMENTOS2/
iMac-de-Ivan:FUNDAMENTOS2 ivan$
```

2. Crea un directorio que se llame "FUNDAMENTOS2" y entra a el.

```
-----
DATOS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ touch NOMBRE.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ touch NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ ls
DATOS.txt      NOMBRE.txt      NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ echo ivan >NOMBRE.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat NOMBRE.txt
ivan
```

3. Crea un archivo llamado "NOMBRE" con tu primer nombre

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ touch NOMBRE.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ touch NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ ls
DATOS.txt      NOMBRE.txt     NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ echo ivan >NOMBRE.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat NOMBRE.txt
ivan
iMac-de-Ivan:FUNDAMENTOS2 ivan$ echo 312104636 >NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat NUMEROS.txt
312104636
iMac-de-Ivan:FUNDAMENTOS2 ivan$
```

4. Crea un archivo llamado "NUMEROS" con tu número de cuenta.

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cp NOMBRE.txt ../FUNDAMENTOS2/DATOS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat DATOS.txt
ivan
```

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ Cp NUMEROS.txt ../FUNDAMENTOS2/DATOS.txt
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat DATOS.txt
312104636
```

5. Crea un archivo nuevo llamado "DATOS" y copia los datos de los archivos "NOMBRE" y "NUMEROS".

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat NOMBRE.txt
ivan
```

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cat NUMEROS.txt
312104636
```

6. Muestra en pantalla el contenido de ambos archivos, uno a la vez y con que comando lo muestras.

```
iMac-de-Ivan:FUNDAMENTOS2 ivan$ cd ..
iMac-de-Ivan:FUNDAMENTOS ivan$ cd..
-bash: cd..: command not found
iMac-de-Ivan:FUNDAMENTOS ivan$ cd ..
iMac-de-Ivan:~ ivan$
```

7. Salir del directorio FUNDAMENTOS2.

```
iMac-de-Ivan:FUNDAMENTOS ivan$ cd ..
iMac-de-Ivan:~ ivan$ mkdir COPIA1
iMac-de-Ivan:~ ivan$ ls
COPIA1      Documents      FUNDAMENTOS    Movies          Pictures
Desktop     Downloads      Library         Music           Public
iMac-de-Ivan:~ ivan$
```

8. Crea un directorio que se llame COPIA1.

```
iMac-de-Ivan:COPIA1 ivan$ cp ../NOMBRE.txt .
iMac-de-Ivan:COPIA1 ivan$ ls
NOMBRE.txt
iMac-de-Ivan:COPIA1 ivan$ cp ../NUMEROS.txt
usage: cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source_file target_file
        cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source_file ... target_directory
iMac-de-Ivan:COPIA1 ivan$ cp ../NUMEROS.txt .
```

9. Copia los archivos NOMBRE y NUMEROS que creaste, al directorio COPIA1.

```
iMac-de-Ivan:COPIA1 ivan$ ls
NOMBRE.txt      NUMEROS.txt
iMac-de-Ivan:COPIA1 ivan$ █
```

10. Muestra el contenido de tu carpeta.

```
iMac-de-Ivan:COPIA1 ivan$ mv ../DATOS.txt .
iMac-de-Ivan:COPIA1 ivan$ ls
DATOS.txt      NOMBRE.txt      NUMEROS.txt
iMac-de-Ivan:COPIA1 ivan$ █
```

11. Mueve el archivo DATOS a la carpeta COPIA1



```
iMac-de-Ivan:COPIA1 ivan$ wc -l NUMEROS.txt
  1 NUMEROS.txt
iMac-de-Ivan:COPIA1 ivan$
```

12. Muestra en la pantalla el número de líneas, palabras y caracteres en el archivo números.

```
iMac-de-Ivan:FUNDAMENTOS ivan$ ls -l
total 24
-rw-r--r--  1 ivan  staff   5 Aug 28 05:41 ,
drwxr-xr-x  4 ivan  staff 136 Aug 28 06:06 FUNDAMENTOS2
-rw-r--r--  1 ivan  staff   5 Aug 28 05:56 NOMBRE.txt
-rw-r--r--  1 ivan  staff  10 Aug 28 05:57 NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS ivan$
```

13. Muestra los permisos de los archivos contenidos en el directorio FUNDAMENTOS y en DATOS, explica cada uno de ellos.

Los tres primeros caracteres se refieren a los privilegios de propietario, los segundos tres se refiere a los privilegios de grupo y los tres últimos se refieren a otros privilegios.

```
total 24
-rw-r--r--  1 ivan  staff   5 Aug 28 05:41 ,
drwxr-xr-x  4 ivan  staff 136 Aug 28 06:06 FUNDAMENTOS2
-rw-r--r--  1 ivan  staff   5 Aug 28 05:56 NOMBRE.txt
-rw-r--r--  1 ivan  staff  10 Aug 28 05:57 NUMEROS.txt
iMac-de-Ivan:FUNDAMENTOS ivan$ pwd
/Users/ivan/FUNDAMENTOS
iMac-de-Ivan:FUNDAMENTOS ivan$
```

14. Indica el directorio en el que te encuentras y con que comando lo muestras.

El comando *pwd* es el que te indica tu ubicación.

```
/Users/ivan/FUNDAMENTOS
iMac-de-Ivan:FUNDAMENTOS ivan$ cal
  August 2018
Su Mo Tu We Th Fr Sa
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

iMac-de-Ivan:FUNDAMENTOS ivan$
```

15. Teclea el comando cal y escribe lo que muestra.

```
iMac-de-Ivan:FUNDAMENTOS ivan$ date
Tue Aug 28 07:11:04 CDT 2018
iMac-de-Ivan:FUNDAMENTOS ivan$ exit
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Proceso completado]
```

16. Teclea el comando date y escribe la salida.

## Bibliografía

- Óscar Vicente Huguet Soriano, Sonia Doménech Gómez. Introducción a Linux. [Figura 1]. Consulta: Junio de 2015. Disponible en: [http://mural.uv.es/oshuso/81\\_introduccion\\_a\\_linux.html](http://mural.uv.es/oshuso/81_introduccion_a_linux.html)
- Pablo Delgado. Integración de sistemas. Linux y su sistema gestor de ficheros (descripciones).[Figura 2]. Consulta agosto de 2016. Disponible en: <http://todobytes.es/2014/09/integracion-de-sistemas-linux-y-su-sistema-gestor-de-ficheros-descripciones/>

CALIFICACIÓN: \_\_\_\_\_