

INTRODUCCIÓN A UNIX

MC. Dulce I. Valdivia
dulce.valdivia@cinvestav.mx

30.septiembre.2022



Red Mexicana de Bioinformática



Dr. Yalbi Balderas
Servidor



¡GRACIAS!



Dr. Giovanni Marco



Material de ejercicios



MC. Shirley Alquicira

Dr. Irma Martínez



Proyectos y Docencia

AYUDANTES ✨



MC. María J. Palma
Doctorado en Ciencias Biomédicas
UNAM-CCG



Ing. Irecha Cano
Maestría en Biotecnología de Plantas.
CINVESTAV-Irapuato

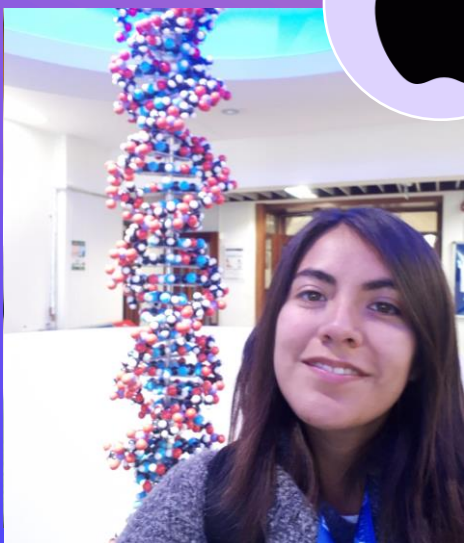


Ing. Adrián Díaz
Maestría en Biología Integrativa,
CINVESTAV-Irapuato



Biól. Emiliano Juárez
Facultad de Ciencias de la UNAM

AYUDANTES ✨



MC. María J. Palma
Doctorado en Ciencias Biomédicas
UNAM-CCG



Ing. Irecha Cano
Maestría en Biotecnología de Plantas.
CINVESTAV-Irapuato



Ing. Adrián Díaz
Maestría en Biología Integrativa,
CINVESTAV-Irapuato



Biól. Emiliano Juárez
Facultad de Ciencias de la UNAM

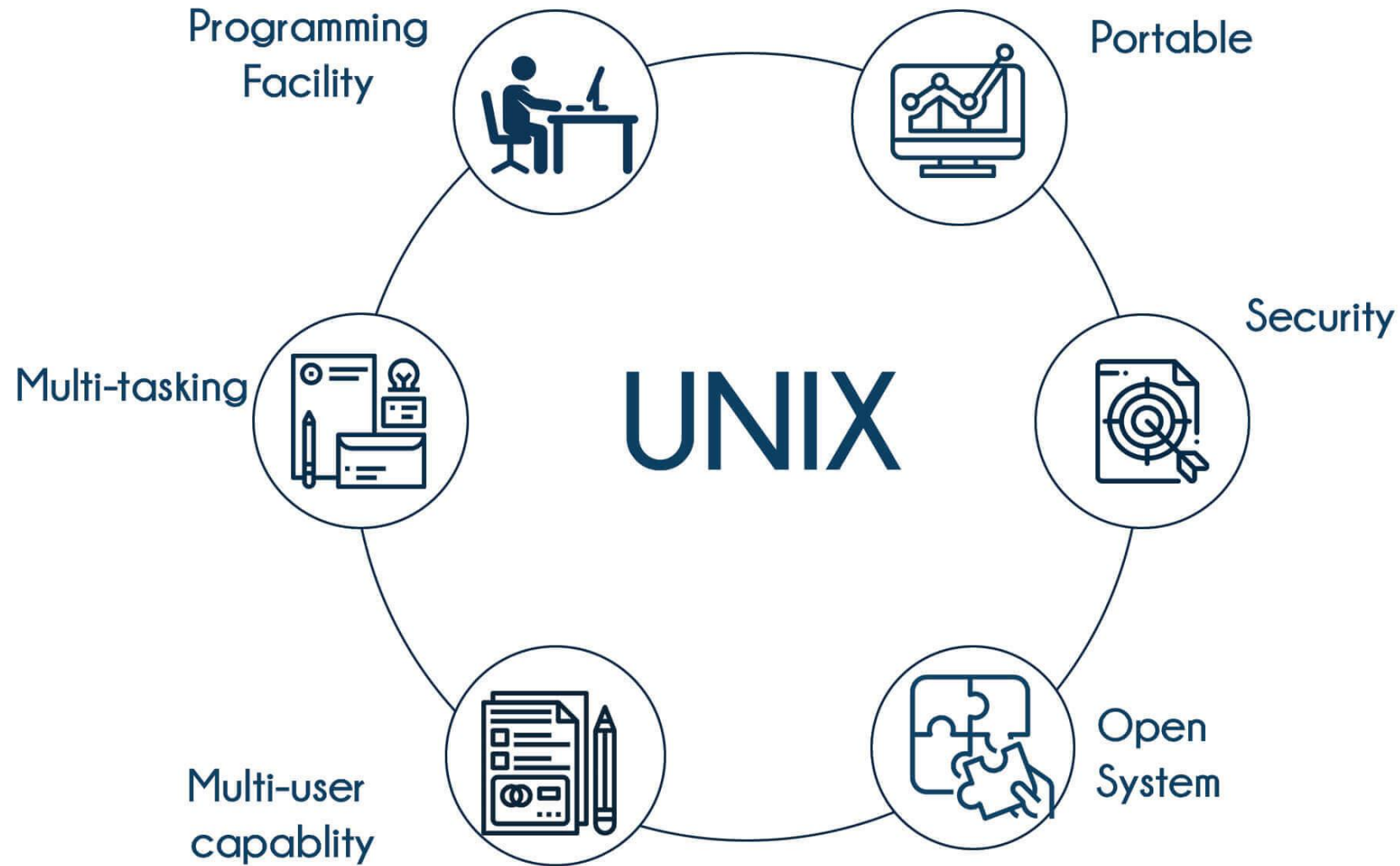
SISTEMAS OPERATIVOS UNIX



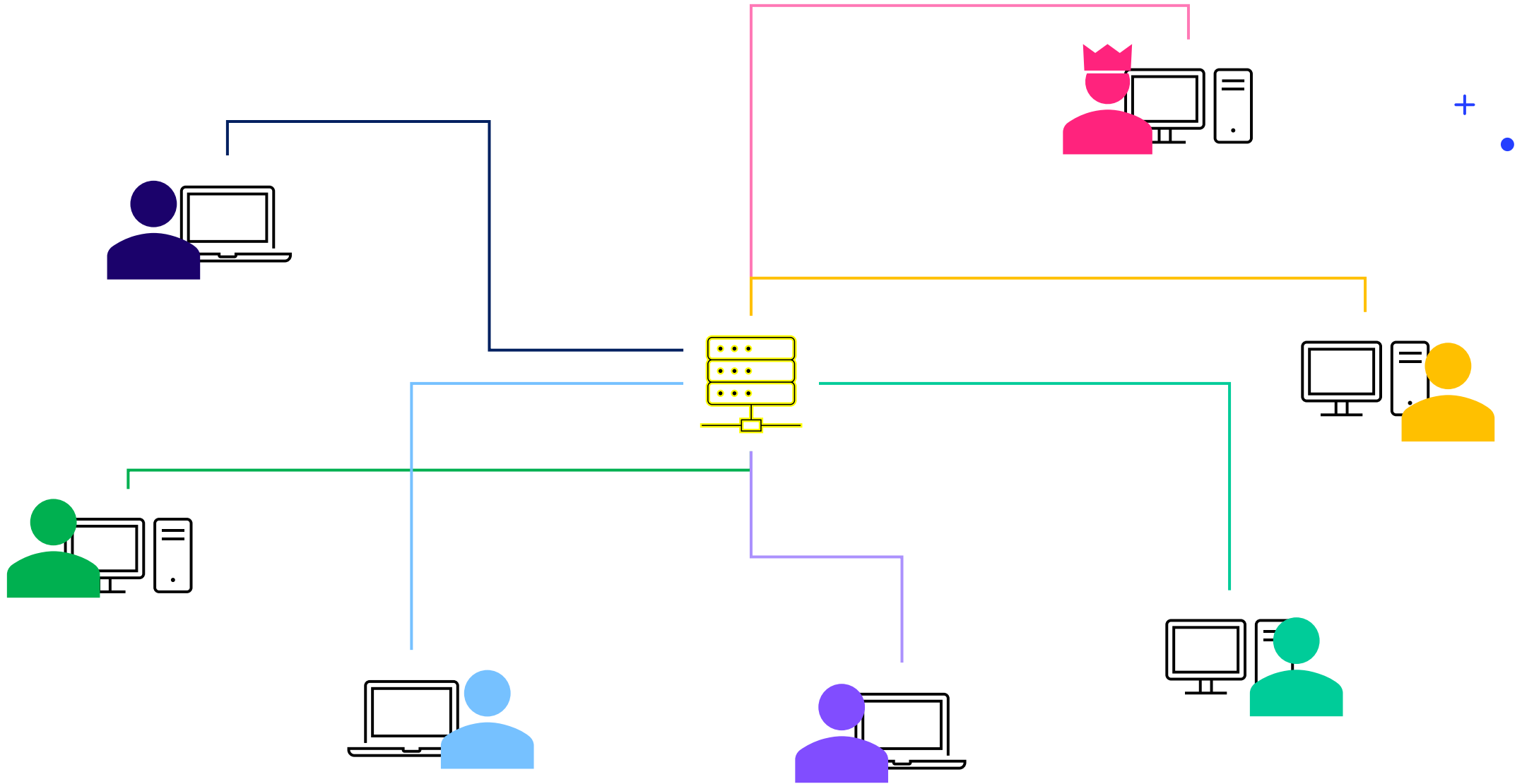
Sistemas Unix



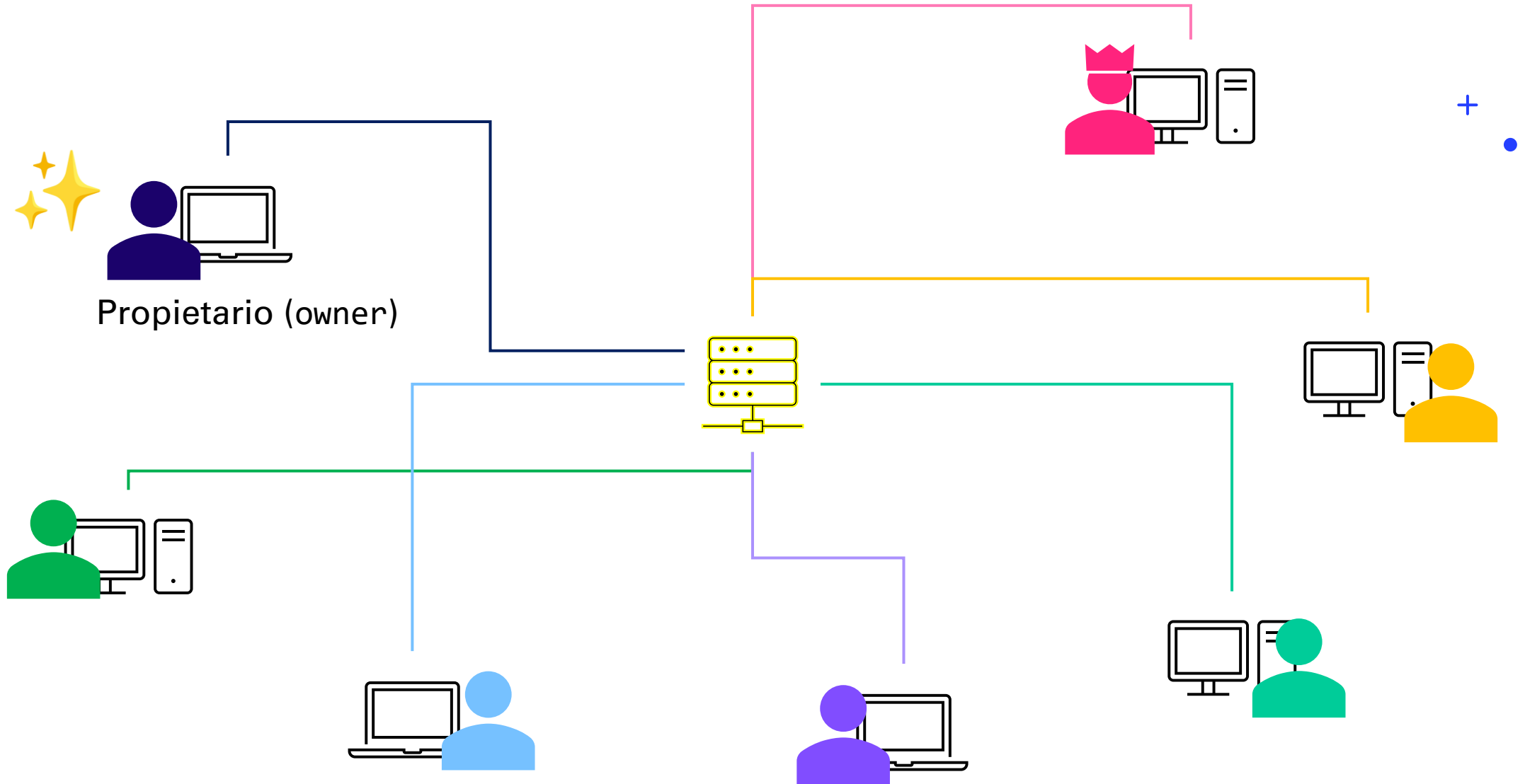
Sistemas Unix



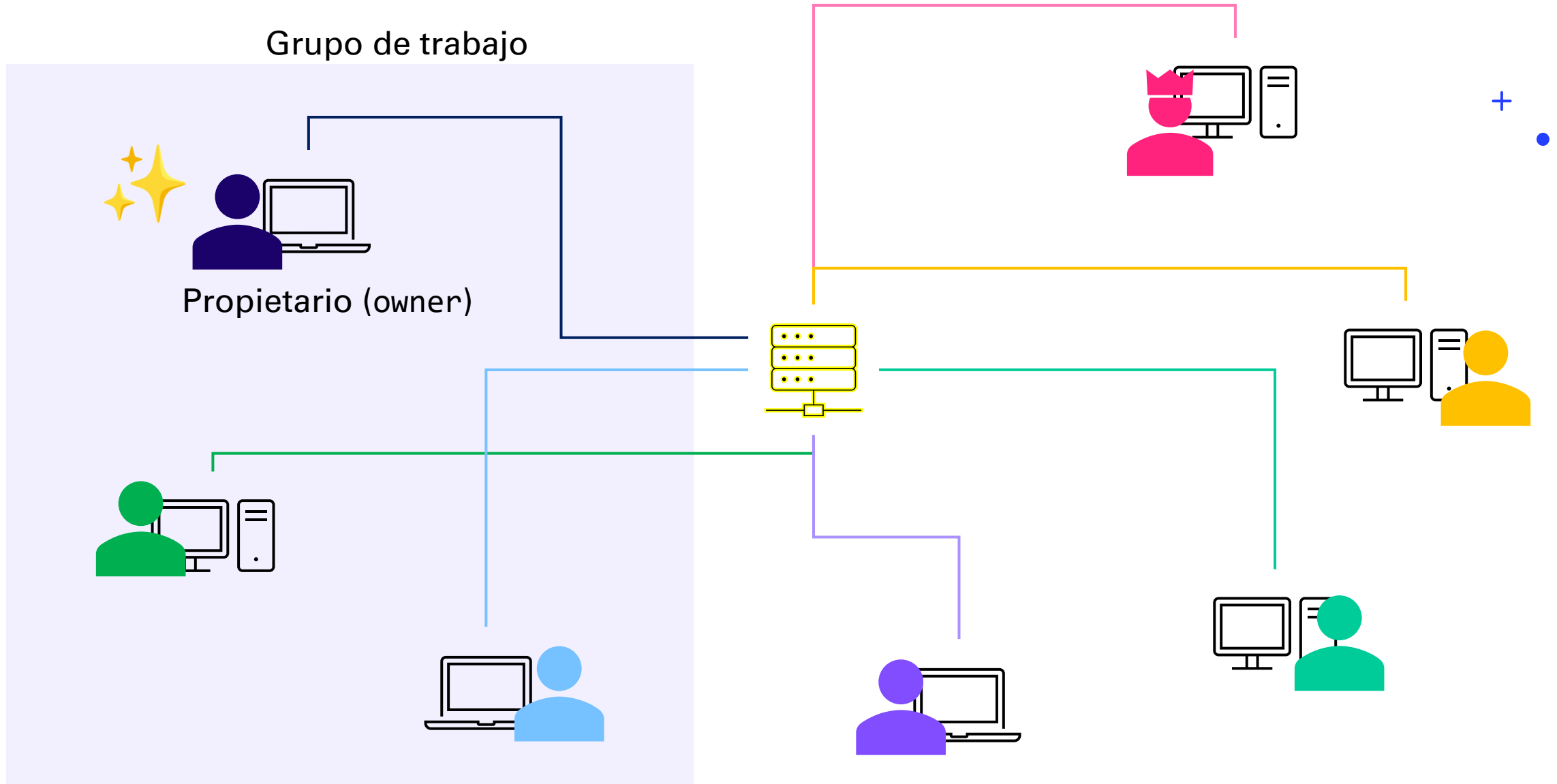
Sistemas multiusuario ✨



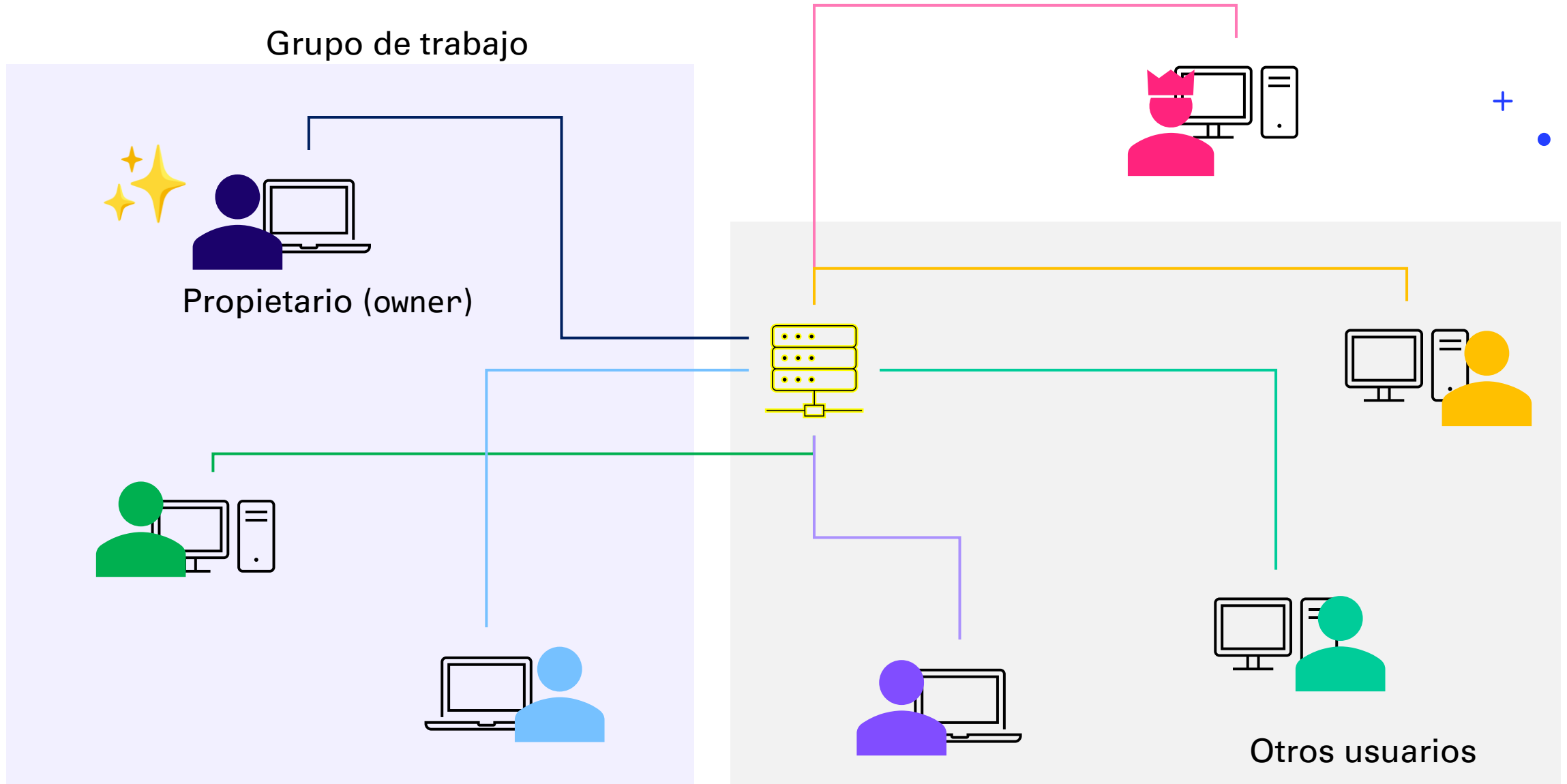
Sistemas multiusuario



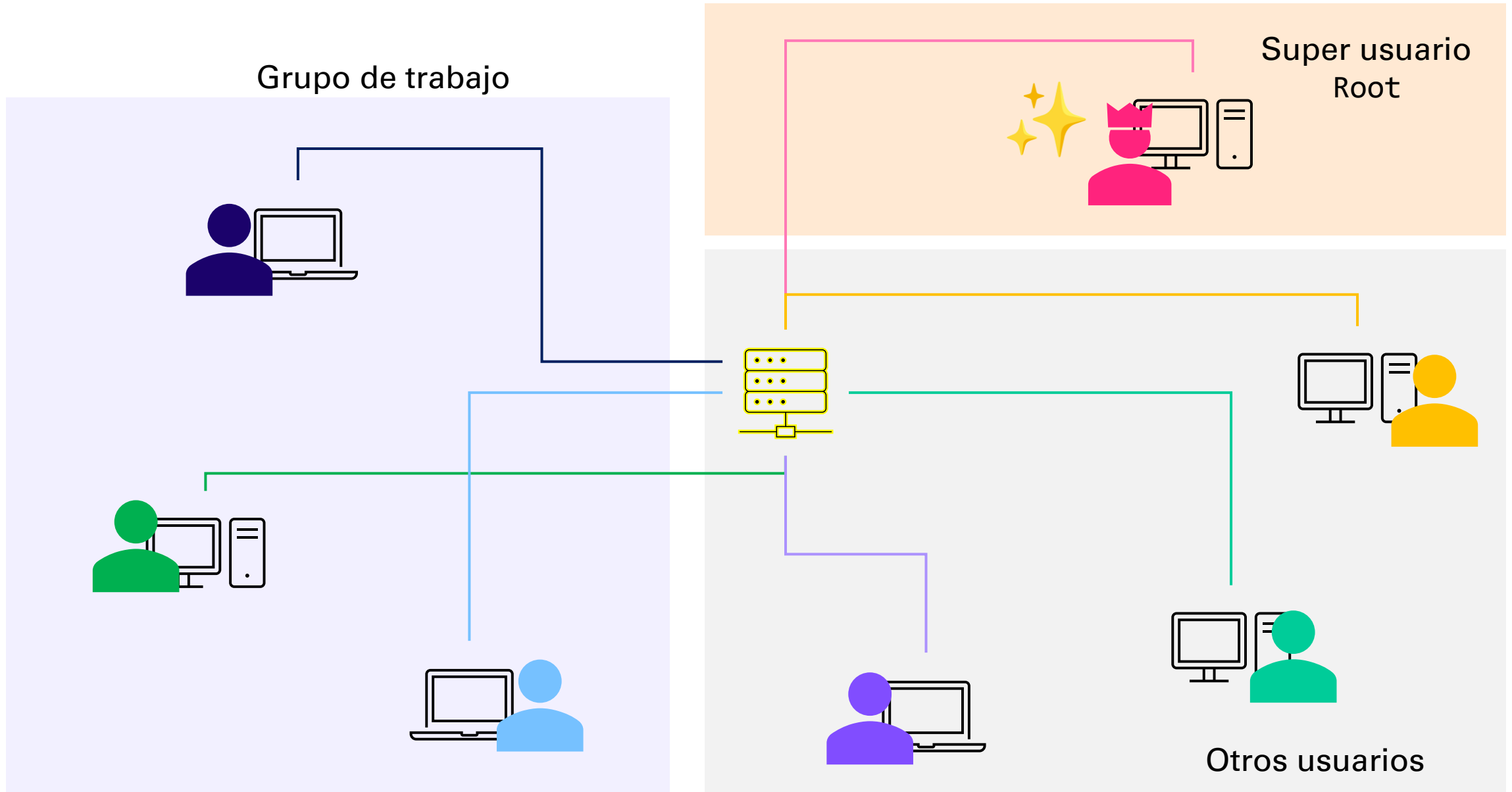
Sistemas multiusuario



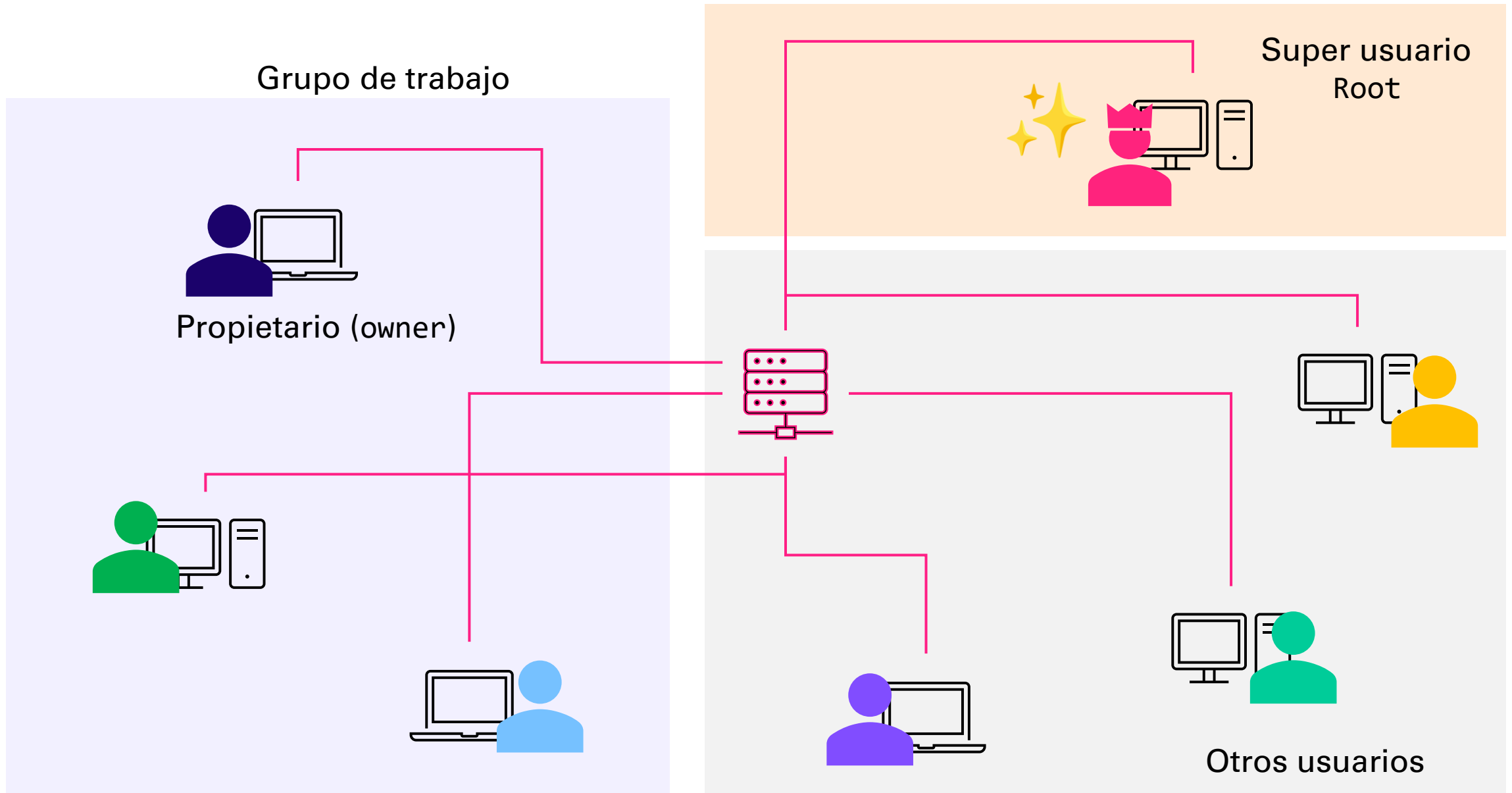
Sistemas multiusuario



Sistemas multiusuario

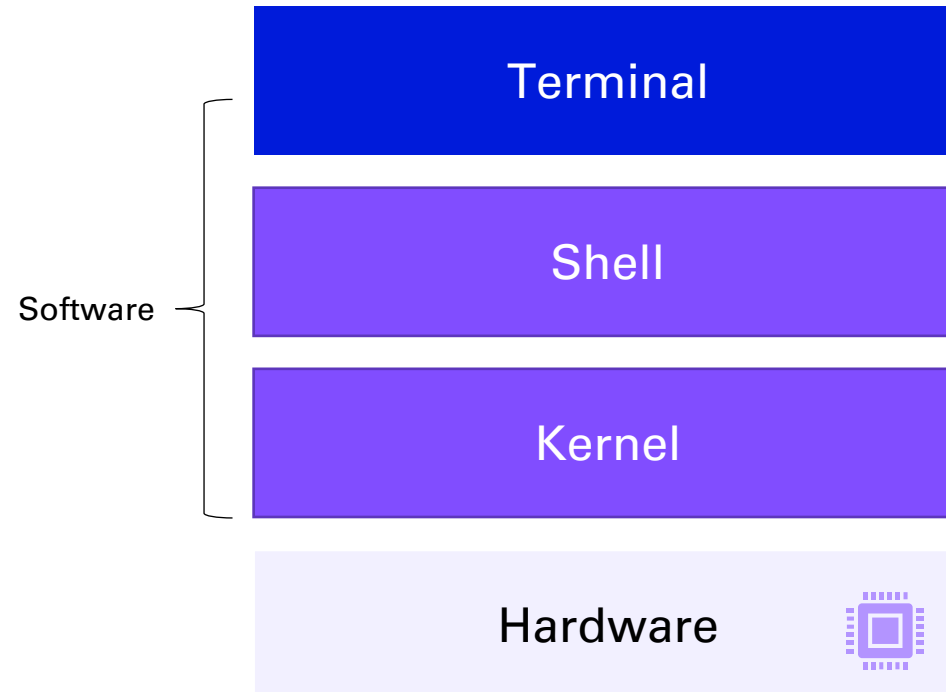


Sistemas multiusuario



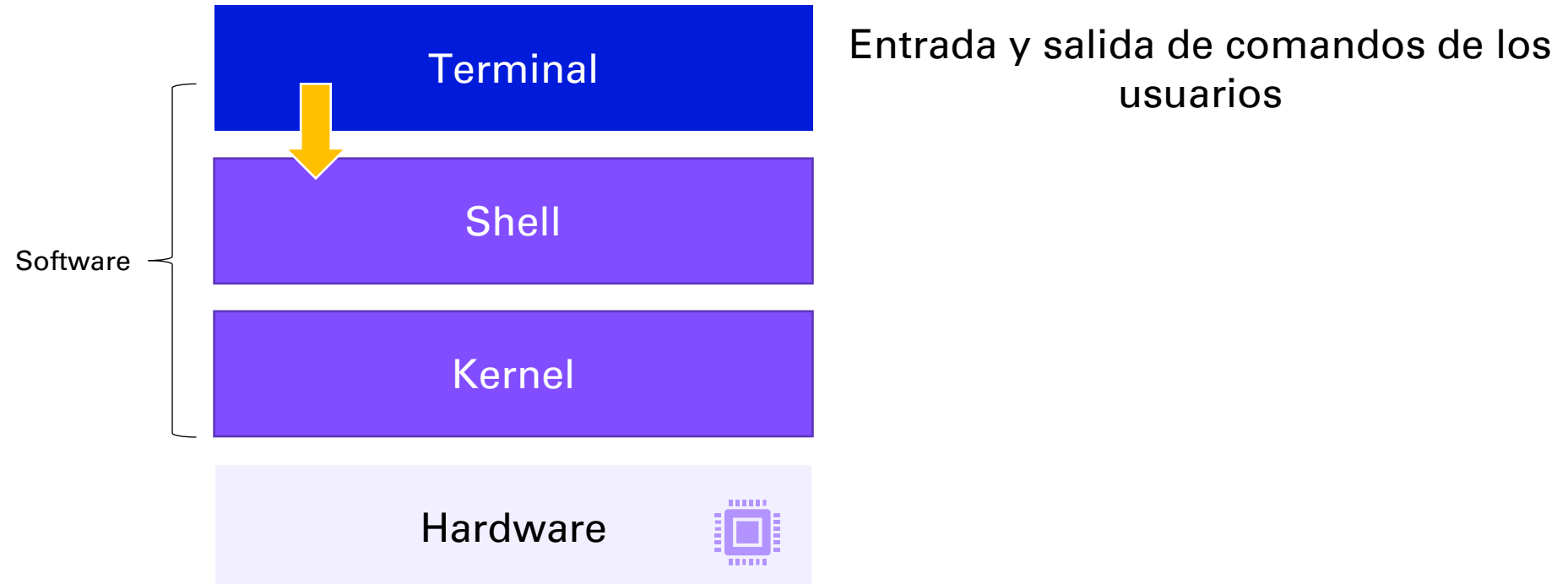


Arquitectura de los sistemas Unix ✨



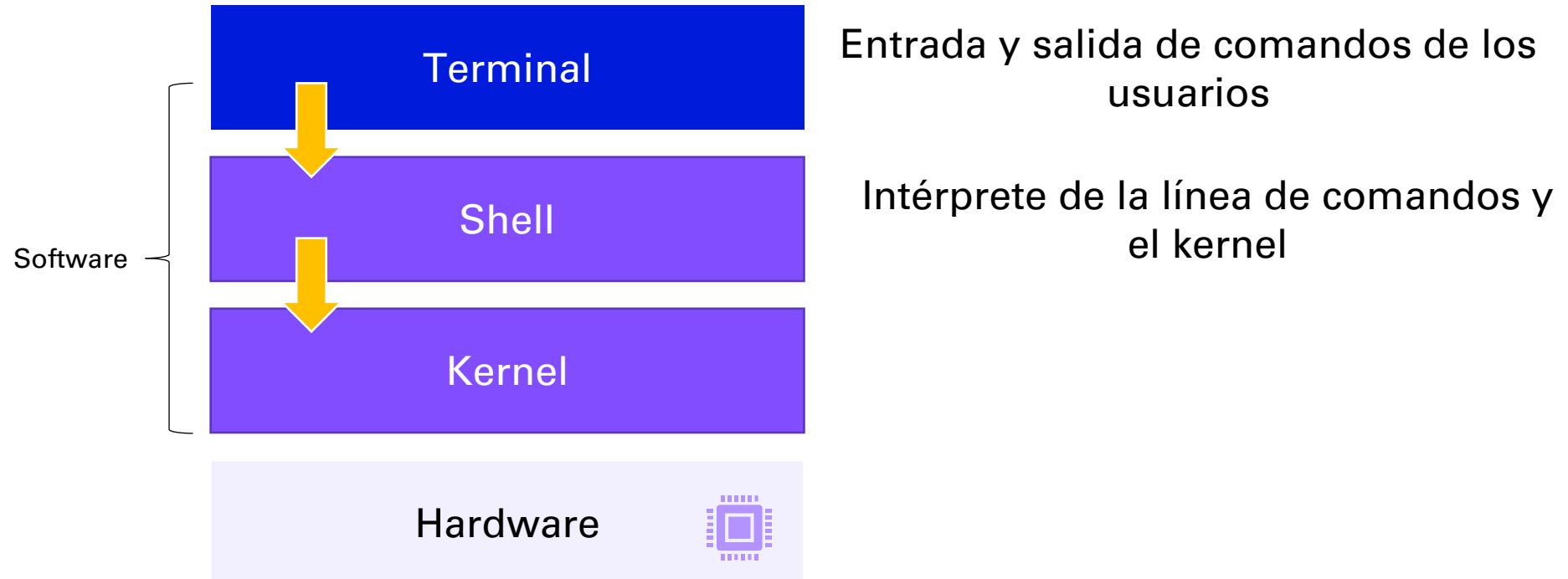


Arquitectura de los sistemas Unix ✨



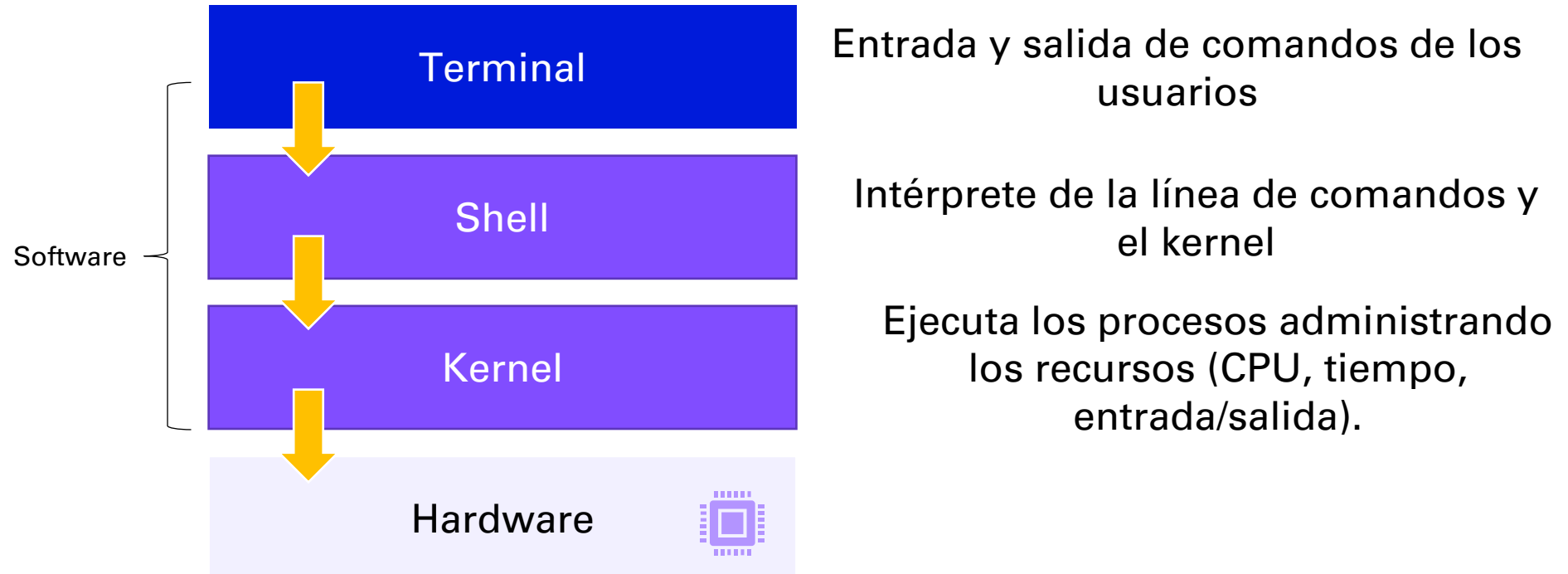


Arquitectura de los sistemas Unix ✨



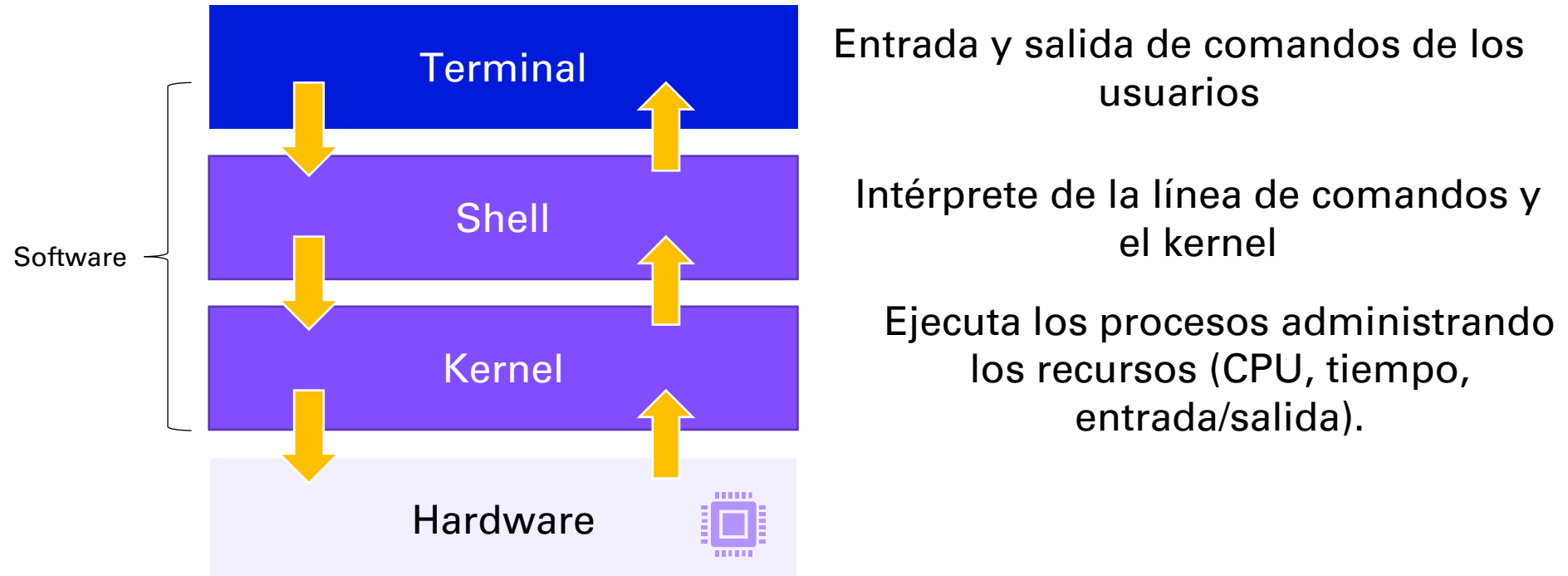


Arquitectura de los sistemas Unix ✨



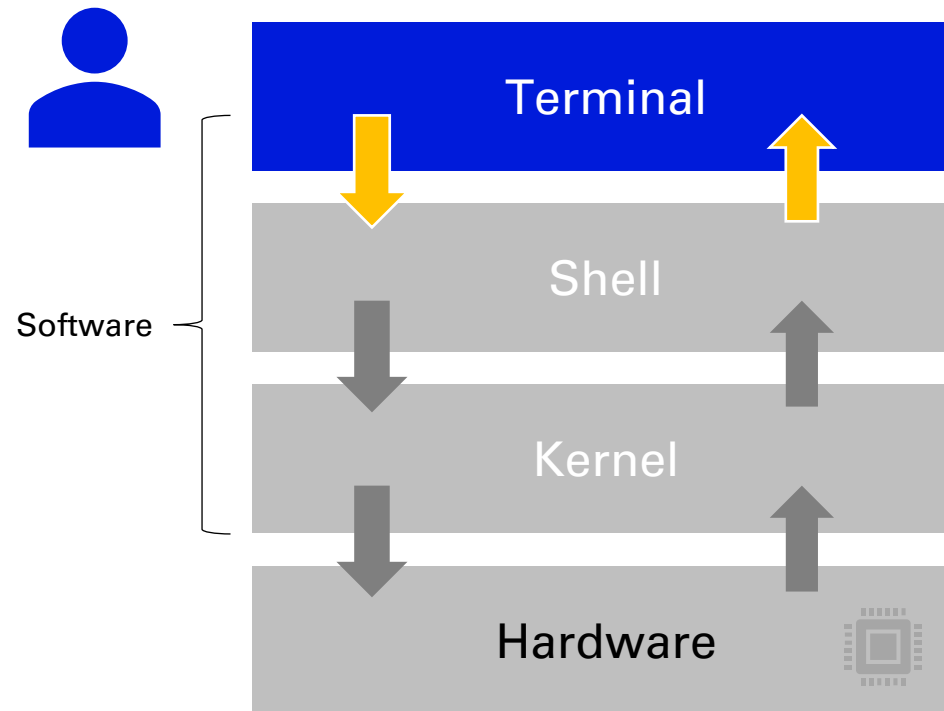


Arquitectura de los sistemas Unix ✨





Arquitectura de los sistemas Unix ✨



Entrada y salida de comandos de los usuarios

Intérprete de la línea de comandos y el kernel

Ejecuta los procesos administrando los recursos (CPU, tiempo, entrada/salida).



La terminal y las líneas de comando



```
irecha@irecha-cano: ~  
irecha@irecha-cano:~$
```

quiénSoy@dóndeEstoyTrabajando \$



La terminal y las líneas de commando ✨

```
irecha@irecha-cano: ~$
```

quiénSoy@dóndeEstoyTrabajando \$

usuario



La terminal y las líneas de commando ✨

A screenshot of a terminal window with a dark background. The title bar at the top reads 'irecha@irecha-cano: ~'. The prompt line shows 'irecha@irecha-cano: ~\$' followed by a cursor. A purple arrow points from the '@' symbol in the prompt line to the '@' symbol in the text below.

```
irecha@irecha-cano: ~$
```

quiénSoy@ dóndeEstoyTrabajando \$

en
(at)



La terminal y las líneas de commando ✨

```
irecha@irecha-cano: ~$
```

quiénSoy@dóndeEstoyTrabajando \$

host / computadora



La terminal y las líneas de commando ✨

```
irecha@irecha-cano: ~$
```

quiénSoy@dóndeEstoyTrabajando \$

Prompt



La terminal y las líneas de commando ✨



quiénSoy@dóndeEstoyTrabajando \$

Línea de comandos



¡Manos a la obra! ✨

Vamos a conectarnos a un servidor remoto para trabajar todos juntos en el mismo sistema Unix:

```
ssh -v -p 2585 unix_user01@hpc-matematicas-z.fciencias.unam.mx
```



¡Manos a la obra! ✨

Vamos a conectarnos a un servidor remoto para trabajar todos juntos en el mismo sistema Unix:

```
ssh -v -p 2585 unix_user01@hpc-matematicas-z.fciencias.unam.mx
```



¡Manos a la obra! ✨

Vamos a conectarnos a un servidor remoto para trabajar todos juntos en el mismo sistema Unix:

```
ssh -v -p 2585 unix_user01@hpc-matematicas-z.fciencias.unam.mx
```

unix_user**01**
unix_user**02**
unix_user**03**
unix_user**04**
...

*Ver lista en [google drive](#)



¡Manos a la obra! ✨

Vamos a conectarnos a un servidor remoto para trabajar todos juntos en el mismo sistema Unix:

```
user@whereami $ ssh
```

```
user@whereami $ whoami
```

```
user@whereami $ passwd
```

```
user@whereami $ exit
```

COMANDOS EN UNIX





Sintaxis básica de los comandos ✨

```
user@whereami $ comando [parámetros] <argumento>
```



Sintaxis básica de los comandos ✨

```
user@whereami $ comando [parámetros] <argumento>
```



El comando :D



Varía el comportamiento
default del comando



El blanco del
comando



Sintaxis básica de los comandos ✨

```
user@whereami $ comando [parámetros] <argumento>
```



¿Qué hacer?



¿Cómo
hacerlo?



¿A quién?



Sintaxis básica de los comandos ✨

```
user@whereami $ comando [parámetros] <argumento>
```

Espacio





Programar es como hacer limonada

1. Llenar la jarra con agua
2. Cortar a la mitad los limones
3. Exprimir las mitades de limones en la jarra
4. Agregar 1 cucharadas de azúcar a la jarra
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar a la mitad los limones
3. Exprimir las mitades de limones en la jarra
4. Agregar 1 cucharadas de azúcar a la jarra
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar [a la mitad] <los limones>
3. Exprimir las mitades de limones en la jarra
4. Agregar 1 cucharadas de azúcar a la jarra
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar [a la mitad] <los limones>
3. Exprimir <las mitades de limones> [en la jarra]
4. Agregar 1 cucharadas de azúcar a la jarra
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar [a la mitad] <los limones>
3. Exprimir <las mitades de limones> [en la jarra]
4. Agregar [1 cucharadas de azúcar] <a la jarra>
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar [a la mitad] <los limones>
3. Exprimir <las mitades de limones> [en la jarra]
4. Agregar [1 cucharadas de azúcar] <a la jarra>
5. Revolver [_____] <_____>
6. Agregar hielos a la jarra





Programar es como hacer limonada

```
user@whereami $ comando [parámetros] <argumentos>
```

1. Llenar <la jarra> [con agua]
2. Cortar [a la mitad] <los limones>
3. Exprimir <las mitades de limones> [en la jarra]
4. Agregar [1 cucharadas de azúcar] <a la jarra>
5. Revolver [_____] <_____>
6. Agregar [hielos] <a la jarra>





¡Manos a la obra! ✨

```
$ cowsay [-f] <message>
```

• + SISTEMA DE ARCHIVOS + •



Árboles



En Biología...



+



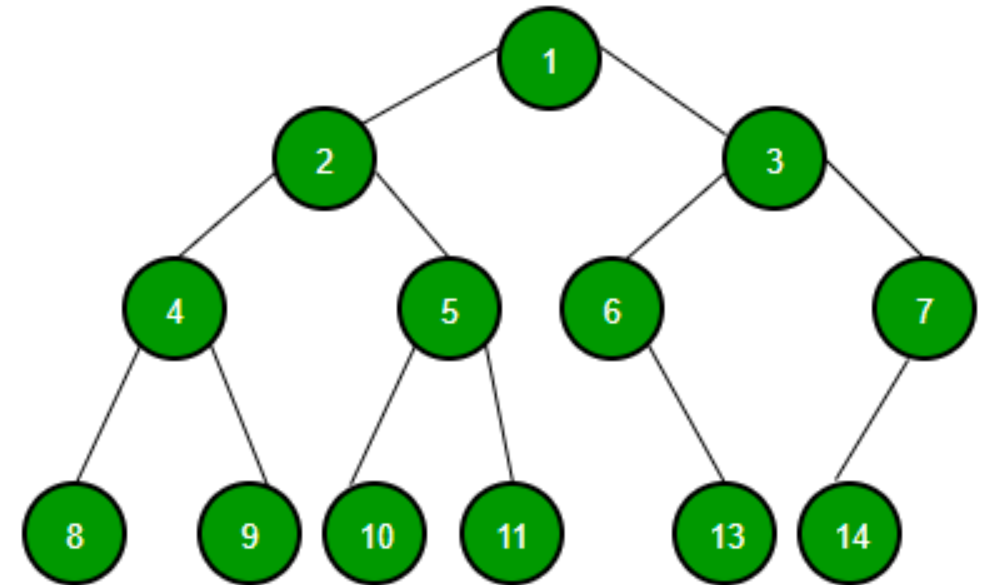
Árboles



En Biología...



En Computación...



+

•

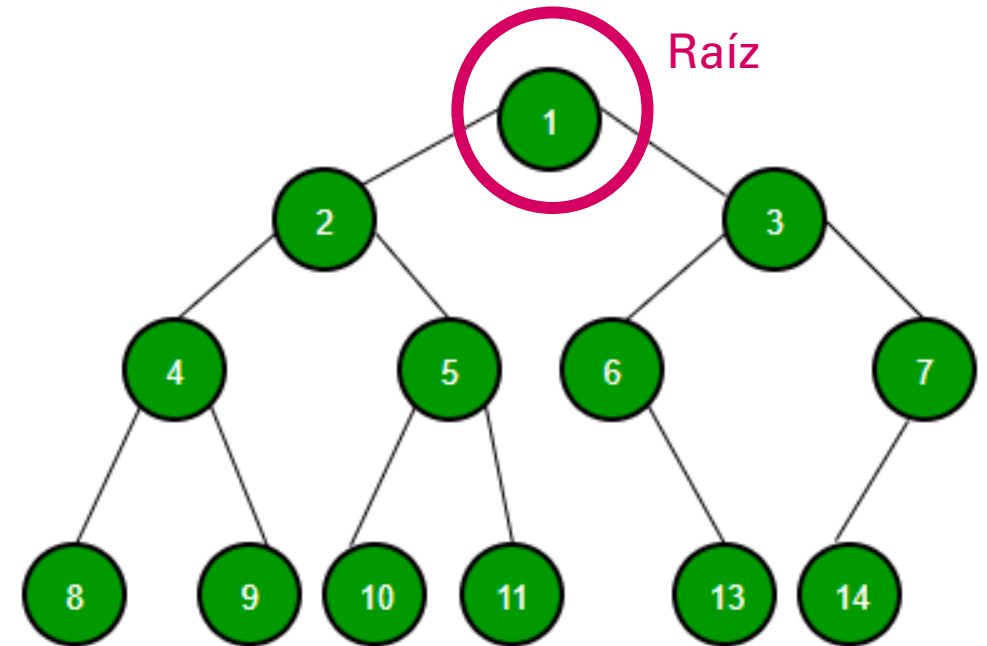
Árboles



En Biología...



En Computación...



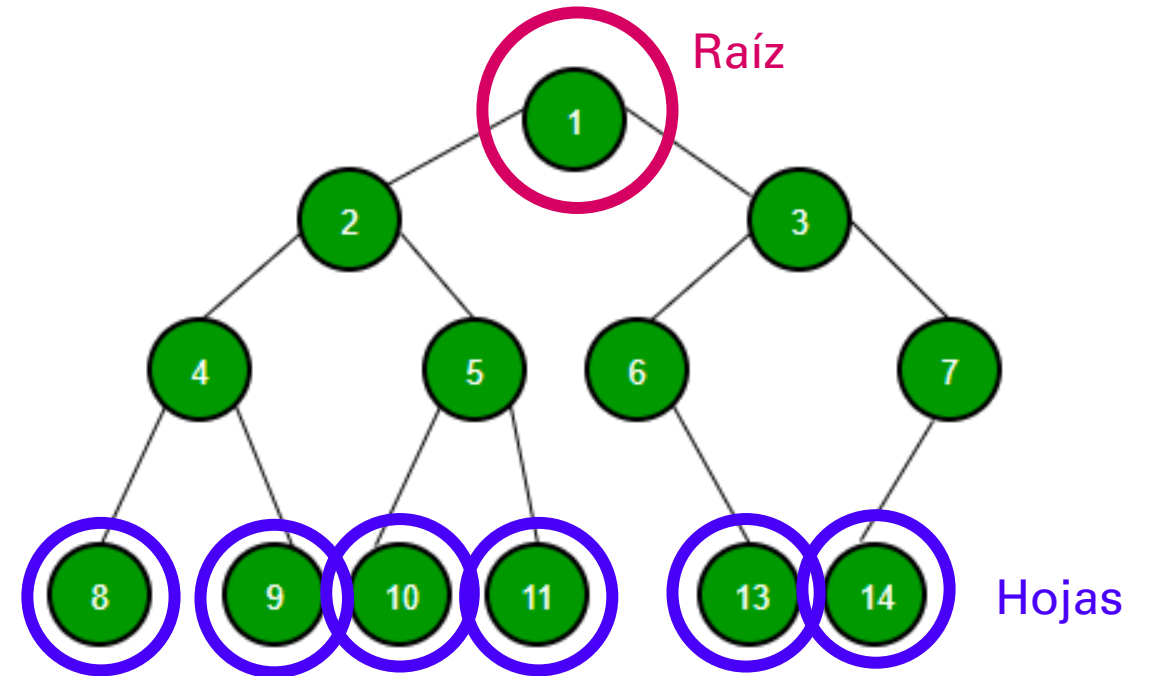
Árboles



En Biología...



En Computación...



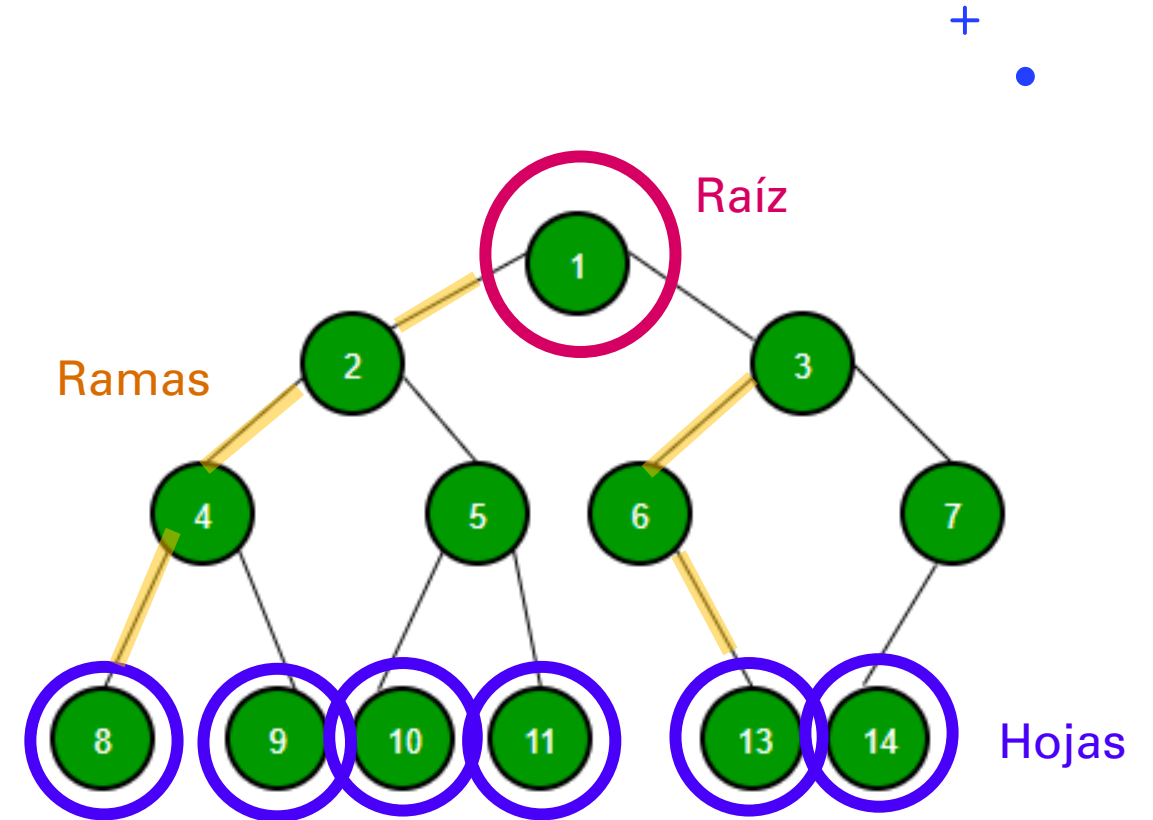
Árboles



En Biología...



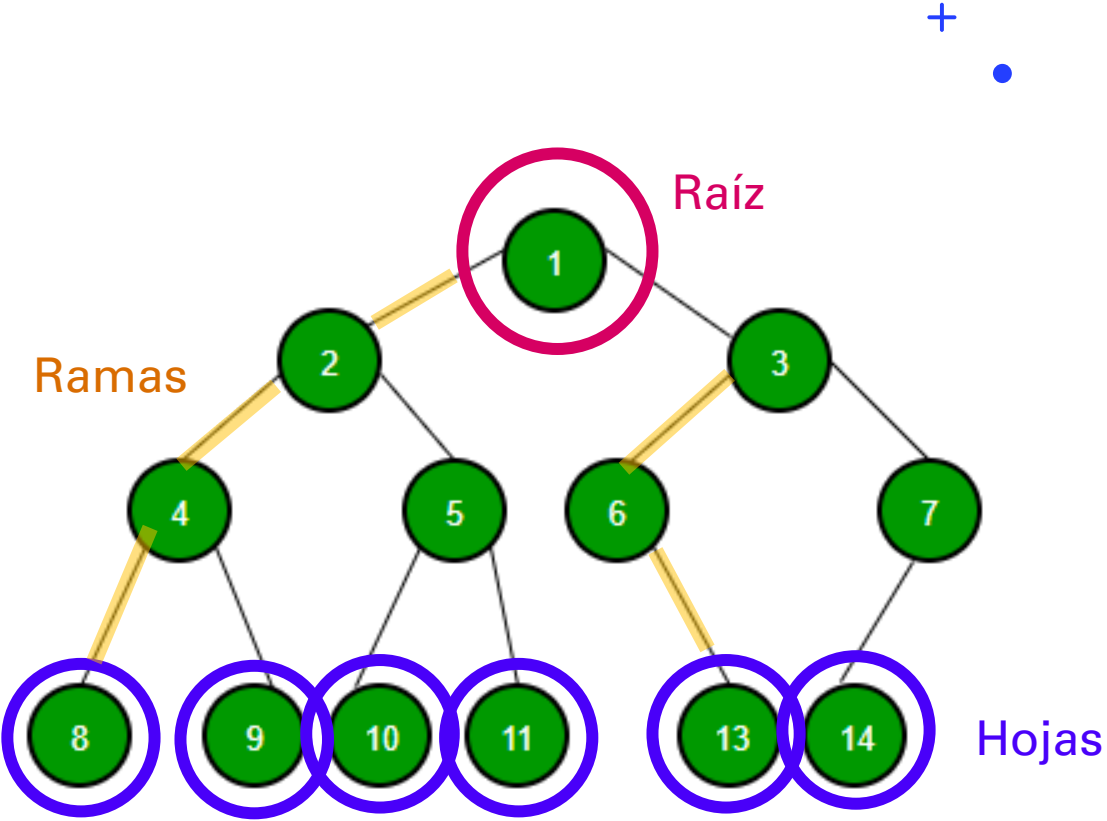
En Computación...



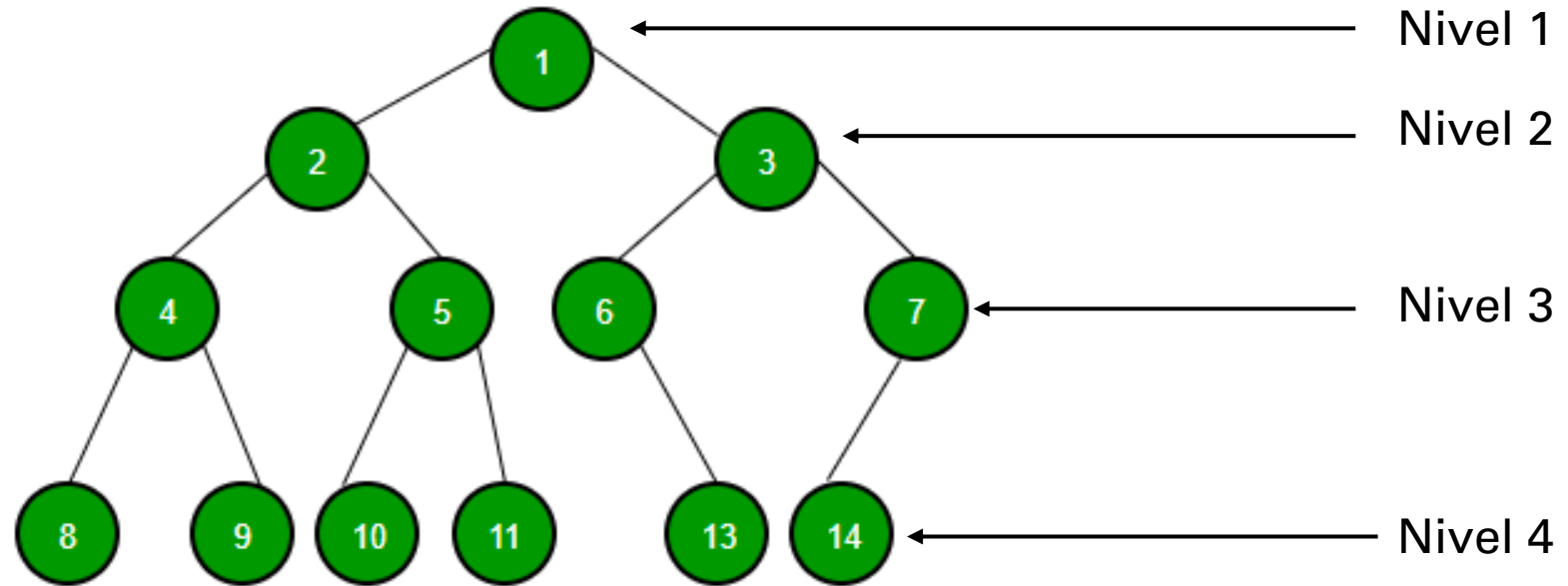
Árboles



En Computación...



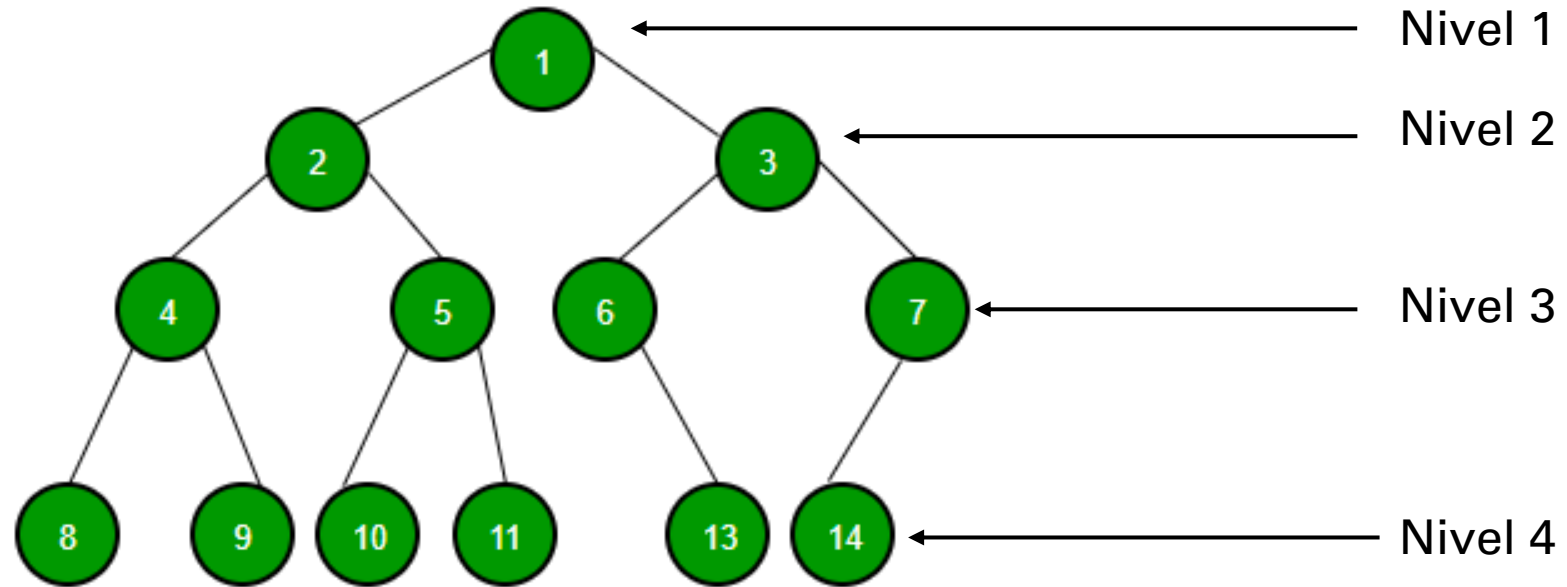
Los árboles definen una jerarquía



+

•

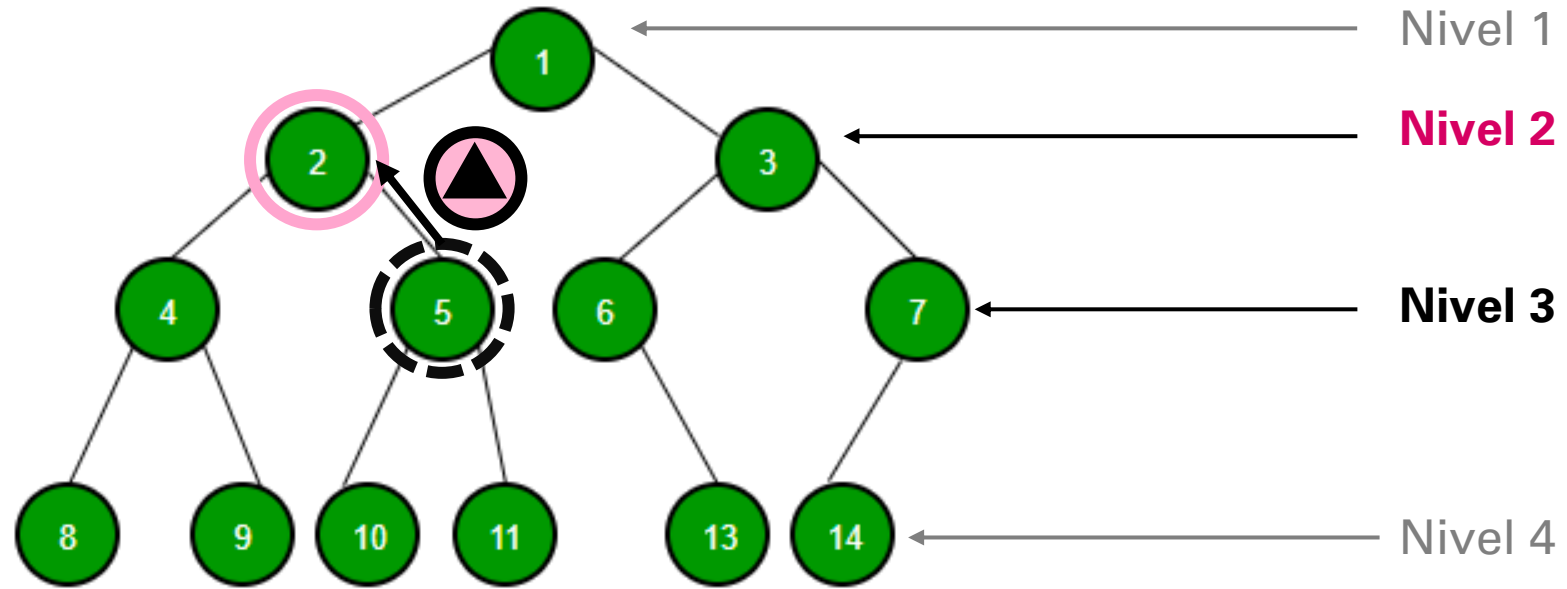
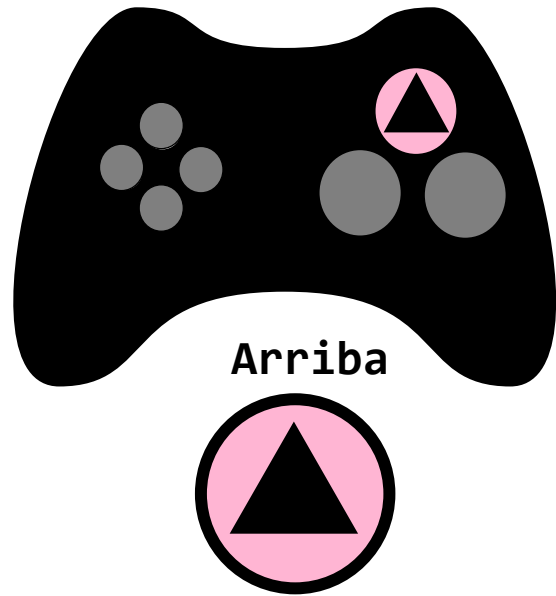
Podemos recorrer esta jerarquía con tres movimientos



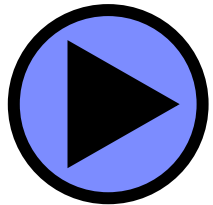
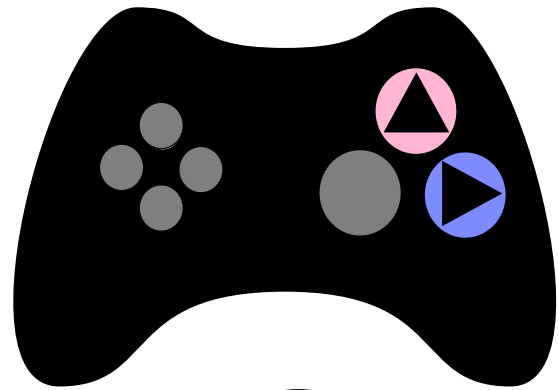
+

•

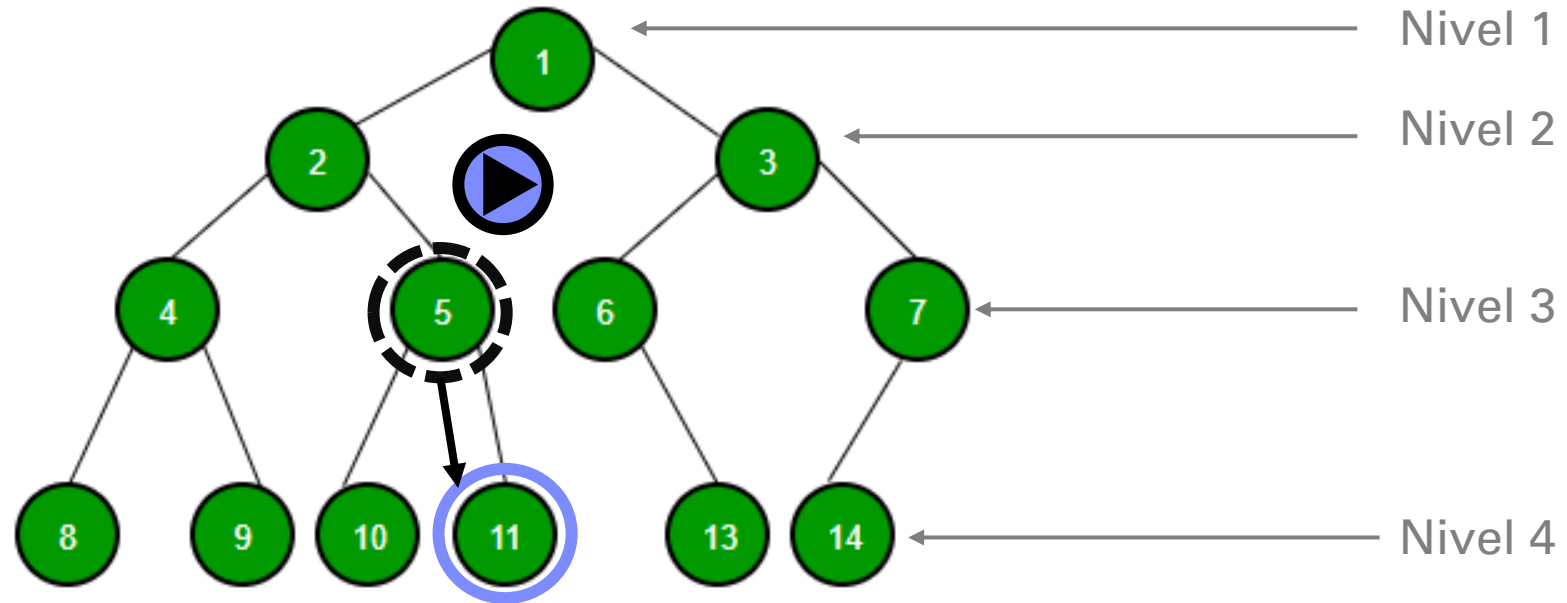
Podemos recorrer esta jerarquía con tres movimientos



Podemos recorrer esta jerarquía con tres movimientos



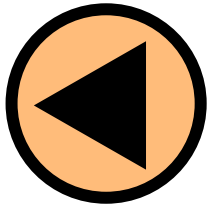
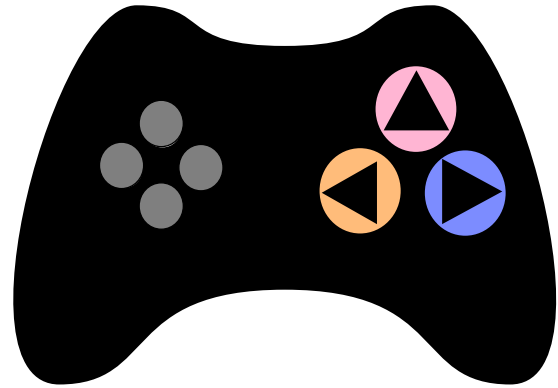
Abajo
derecha



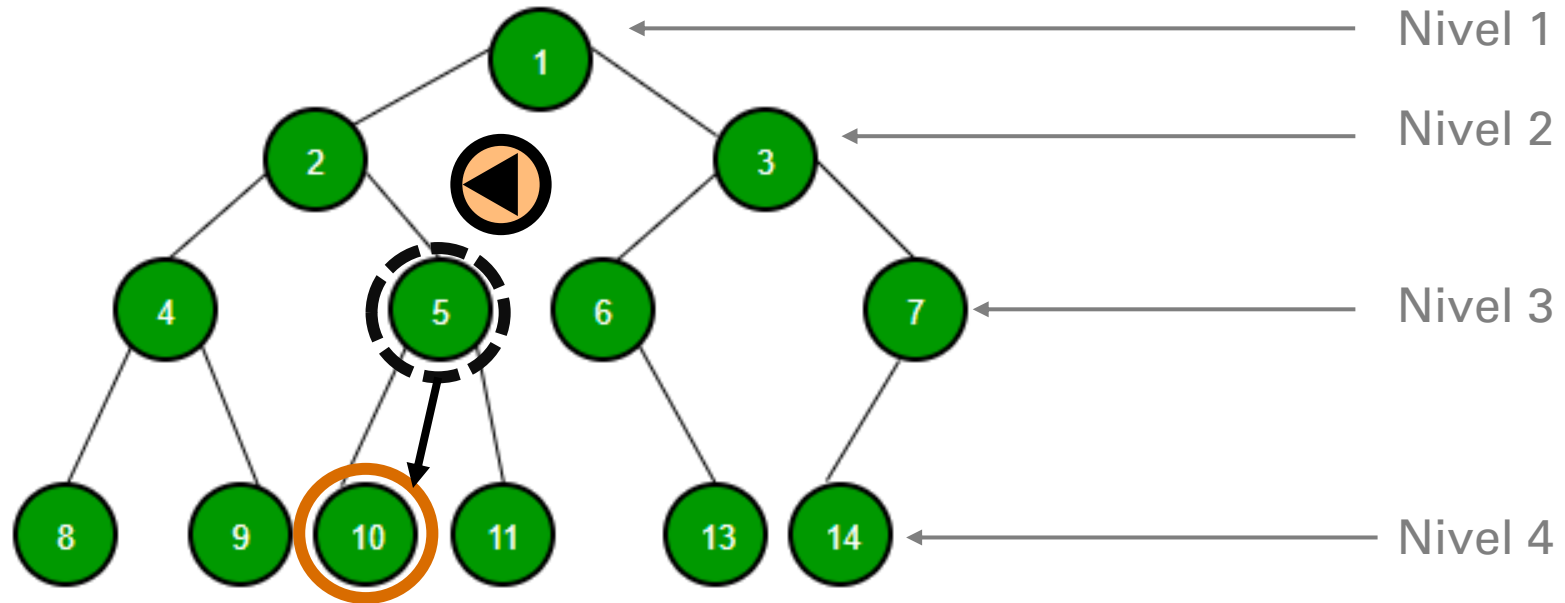
+

•

Podemos recorrer esta jerarquía con tres movimientos



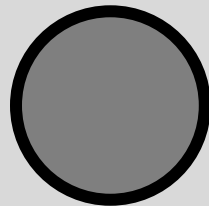
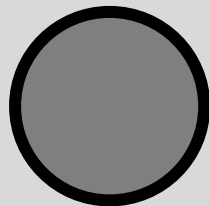
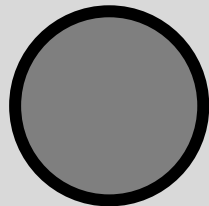
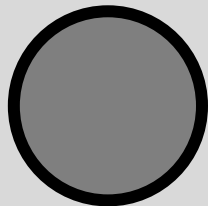
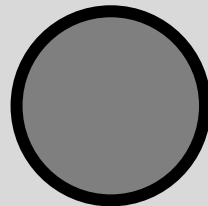
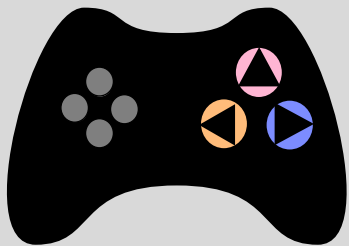
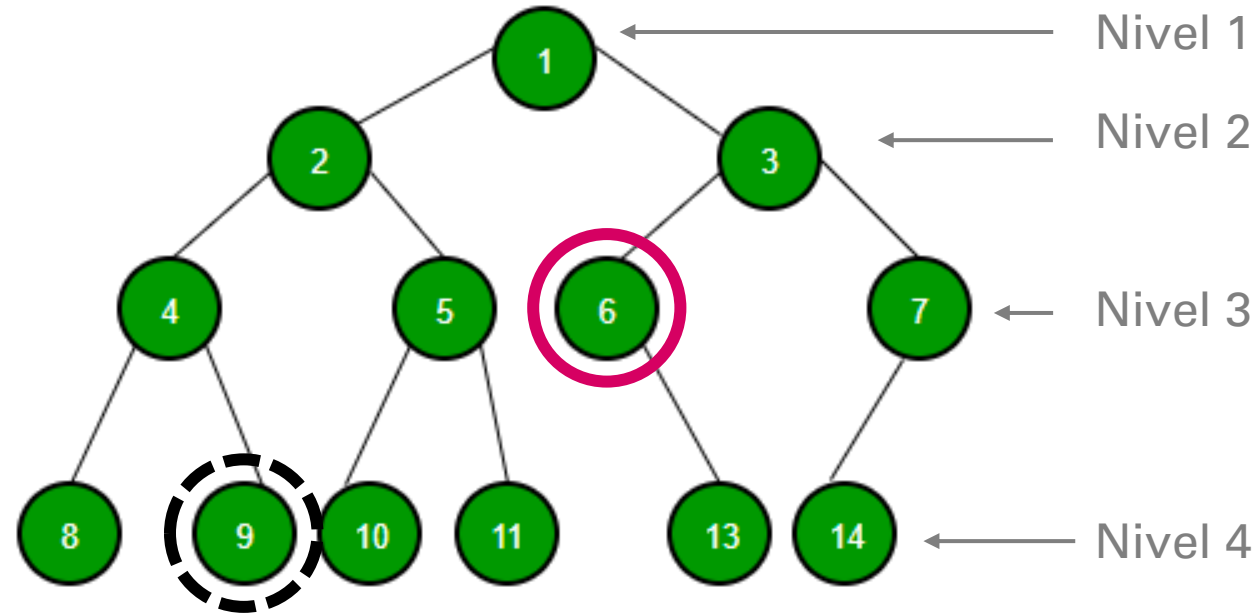
Abajo
izquierda



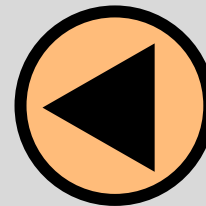
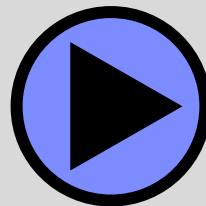
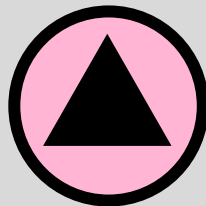
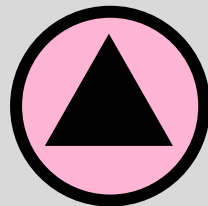
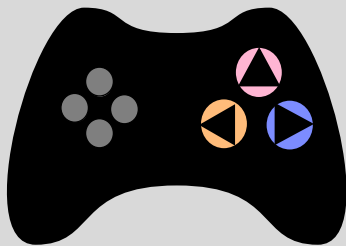
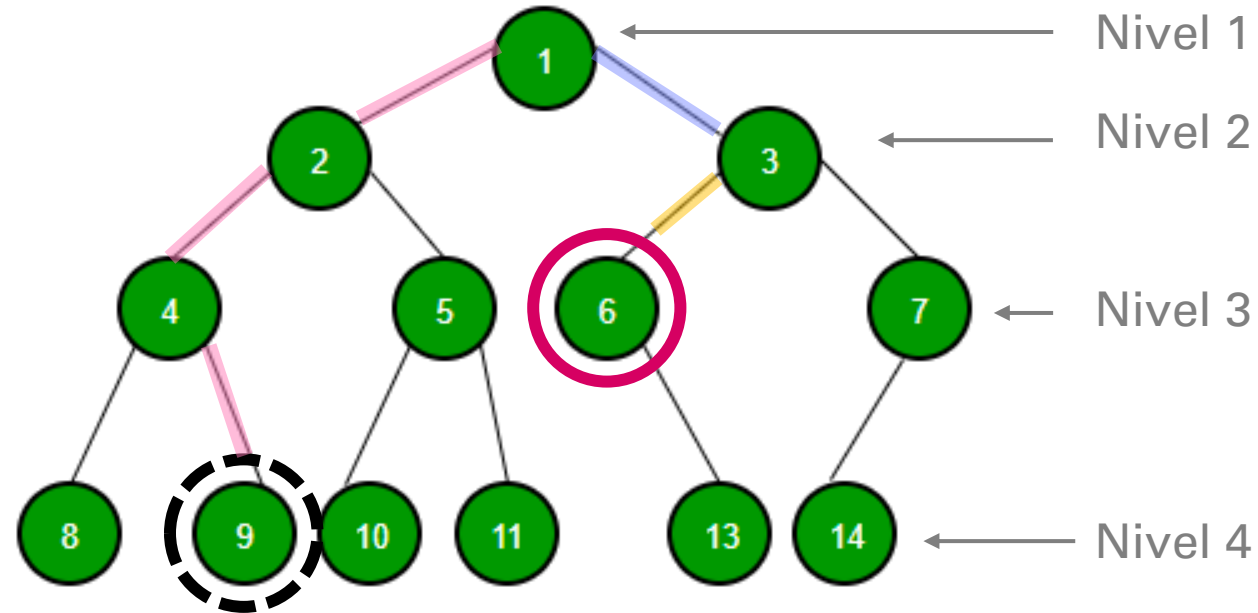
+

•

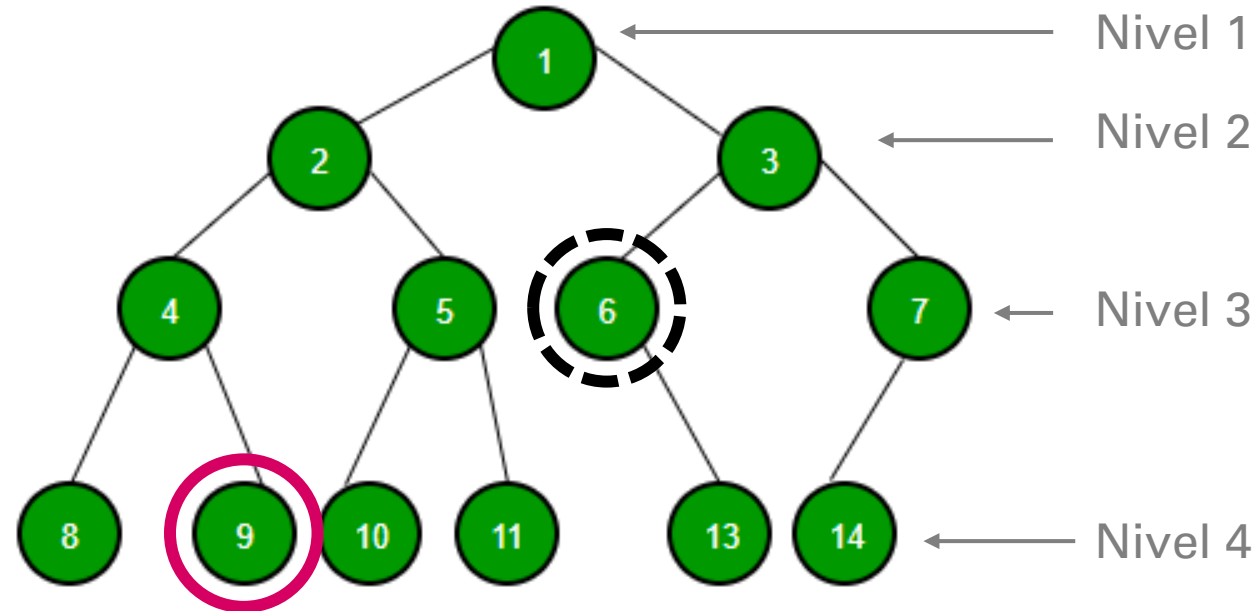
¿Cómo llegamos del nodo 9 al 6?



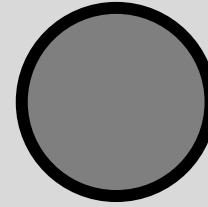
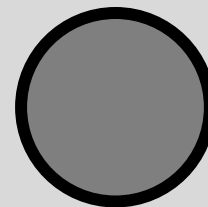
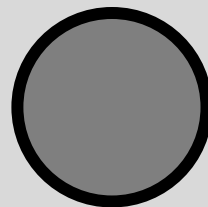
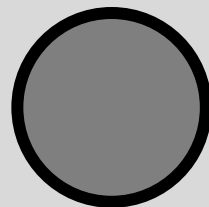
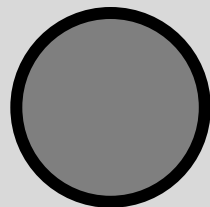
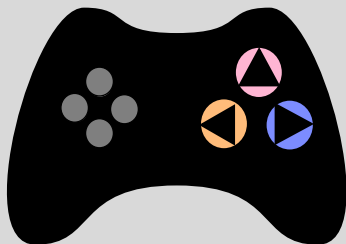
¿Cómo llegamos del nodo 9 al 6?



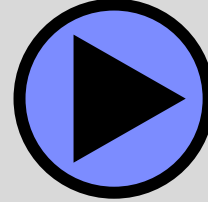
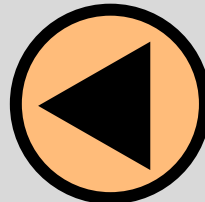
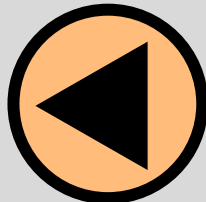
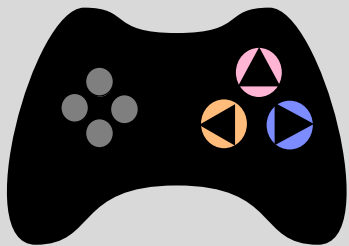
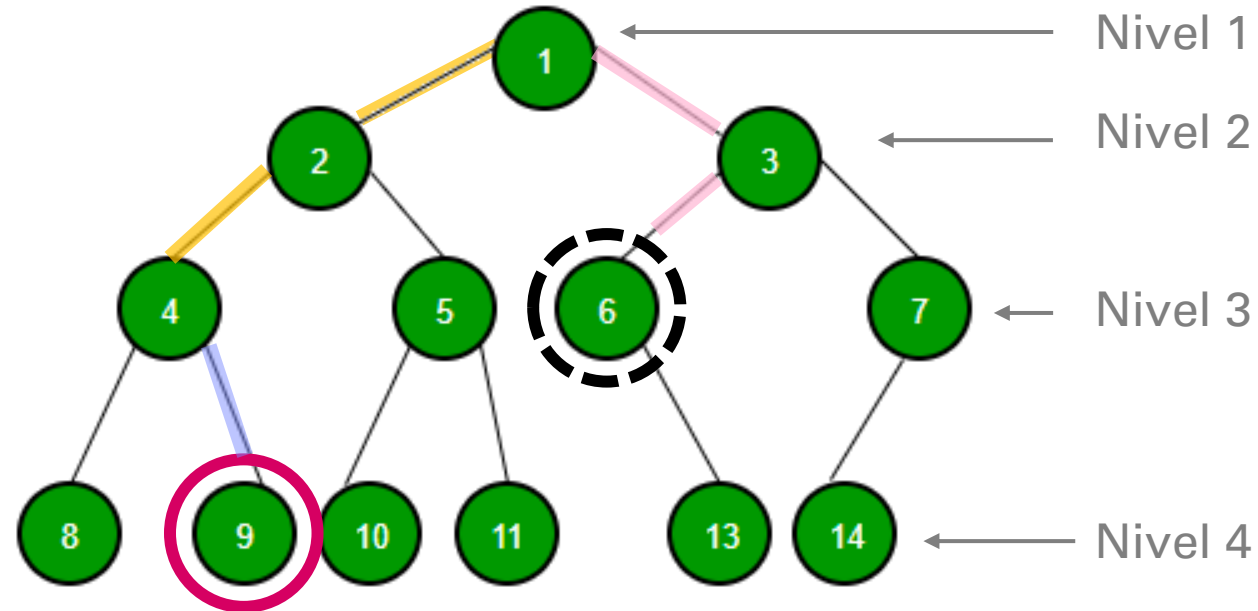
¿Cómo **regresamos** del nodo 9 al 6?



+

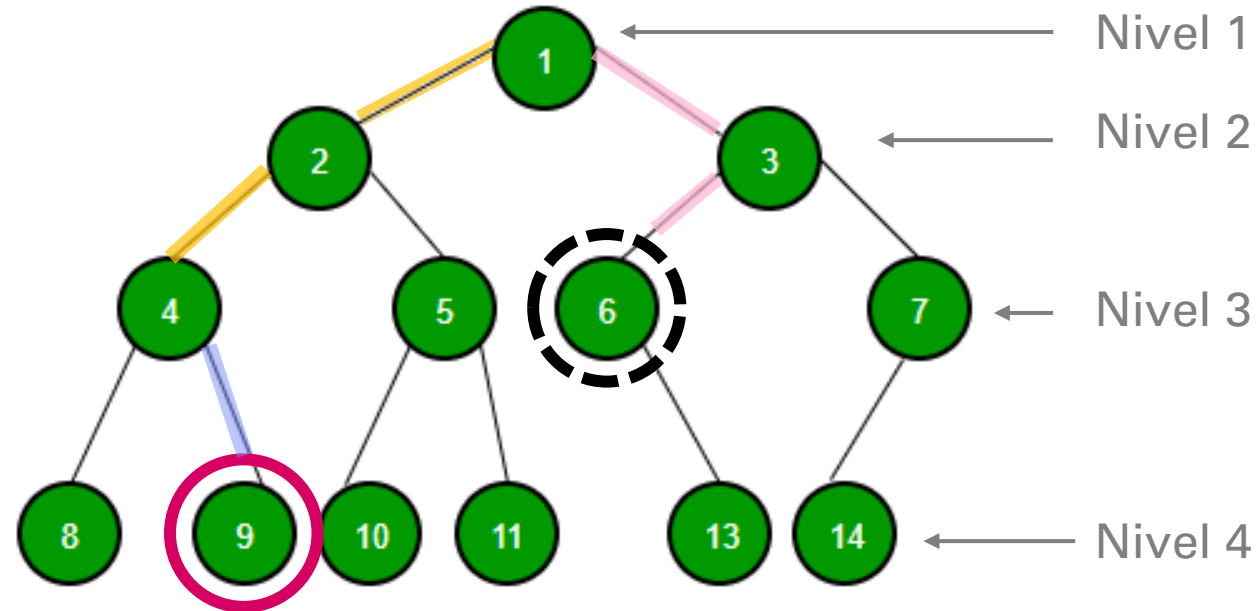


¿Cómo **regresamos** del nodo 9 al 6?

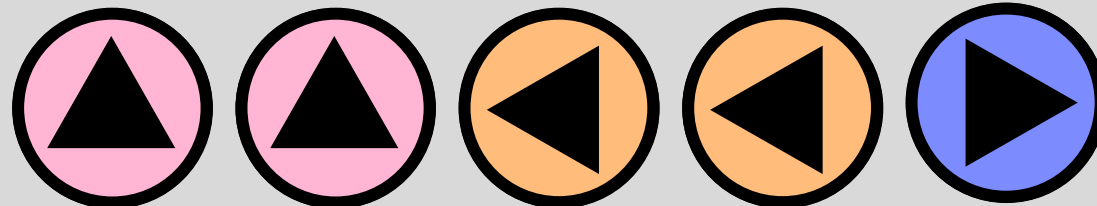




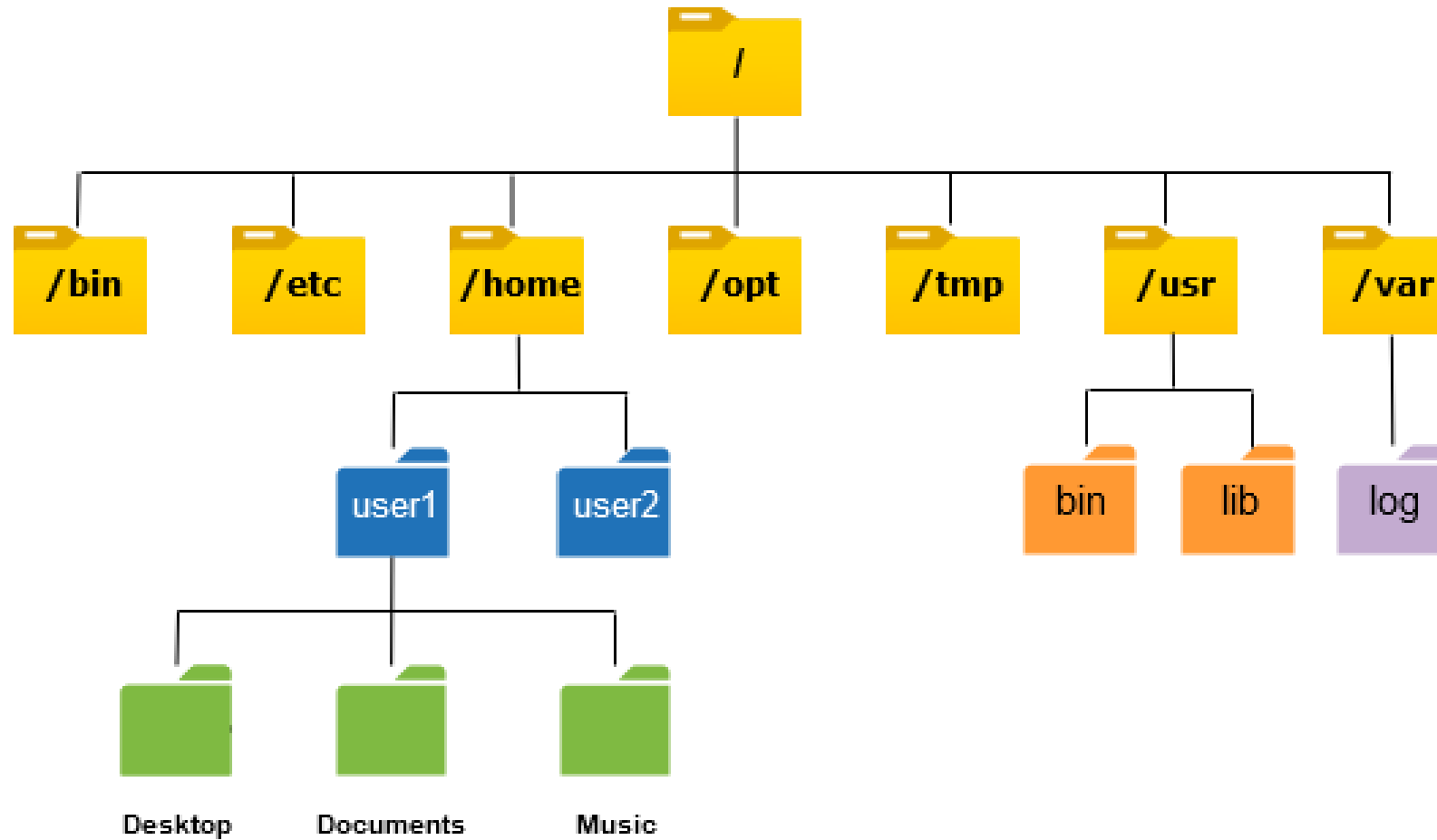
La secuencia de movimientos para llegar de un nodo a otro es llamado un **camino** (**path**)



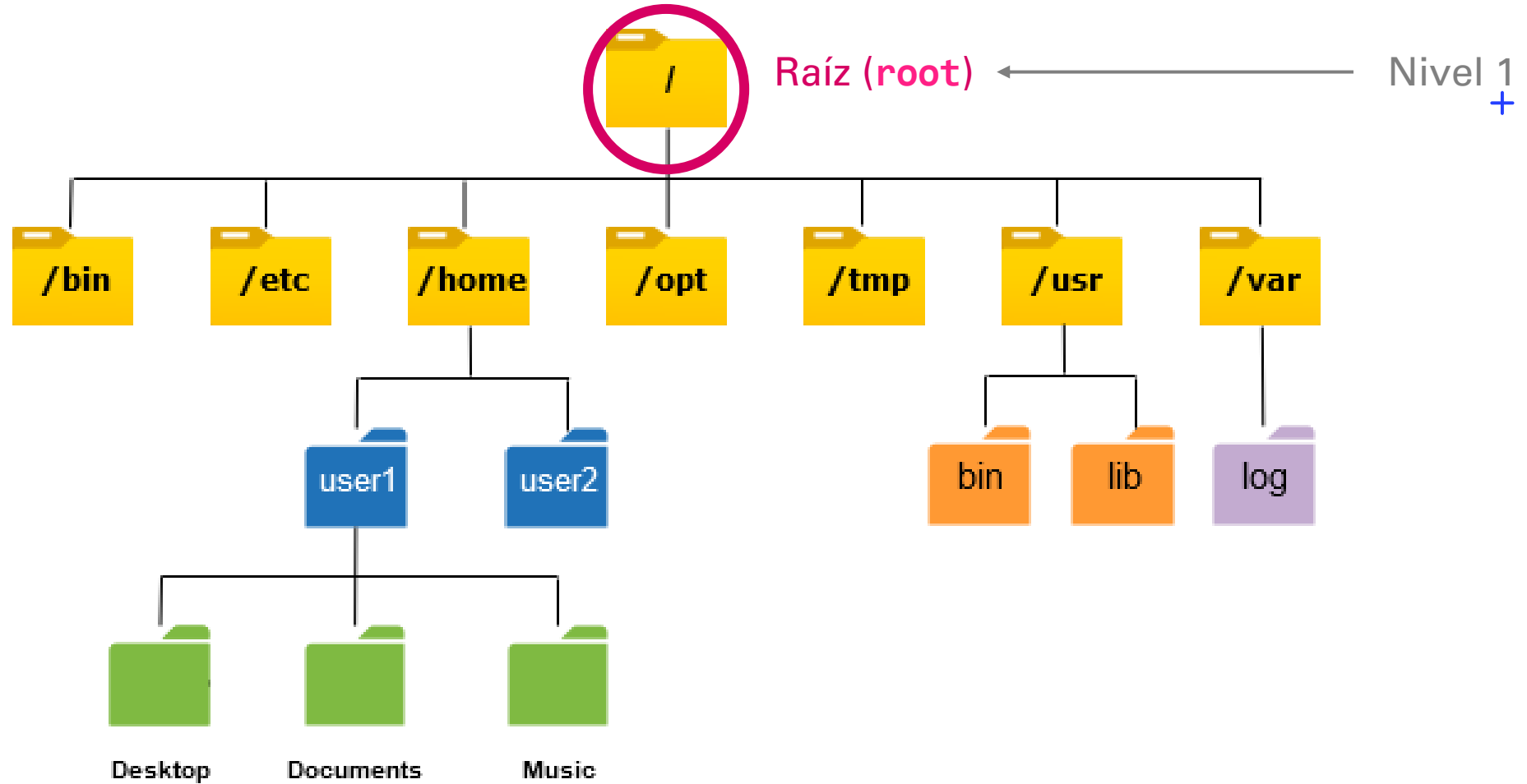
+



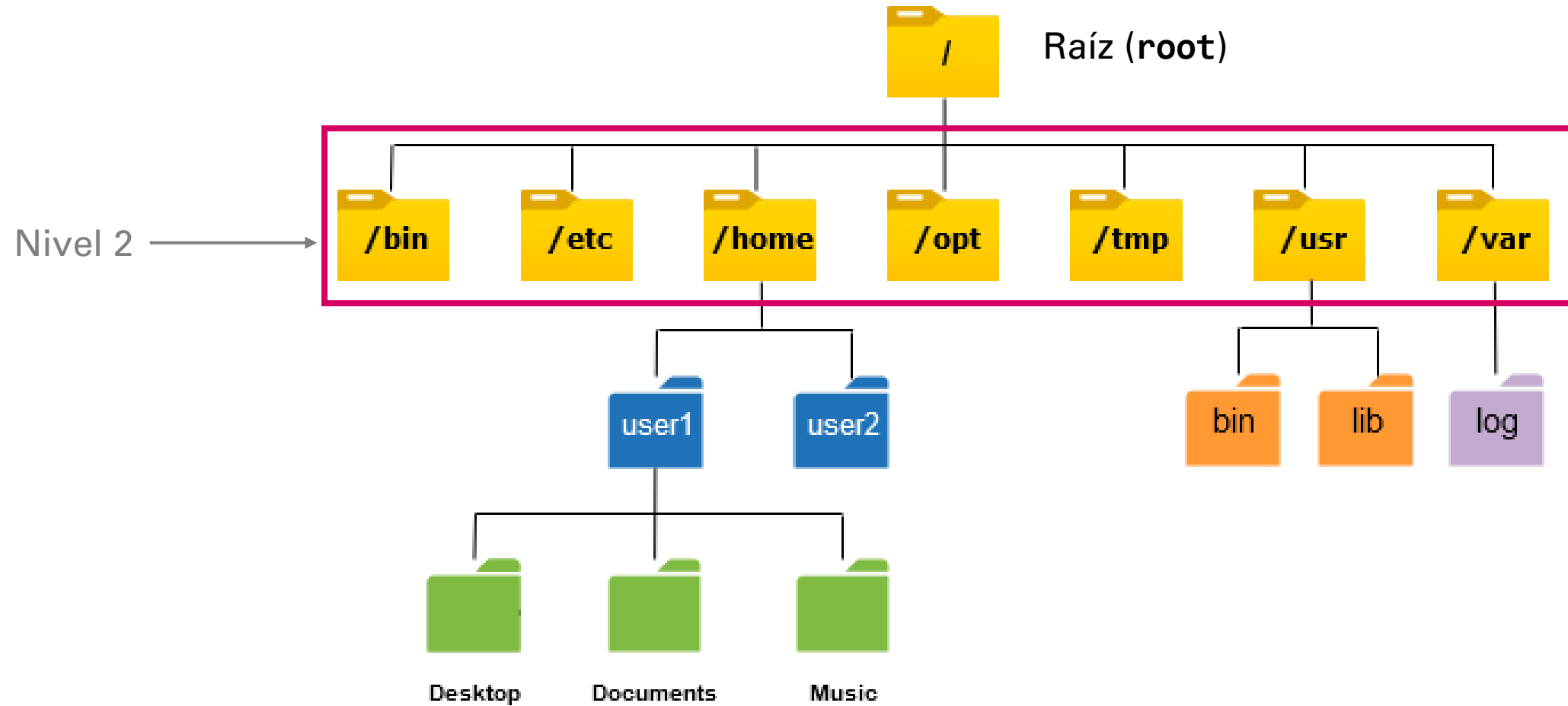
El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



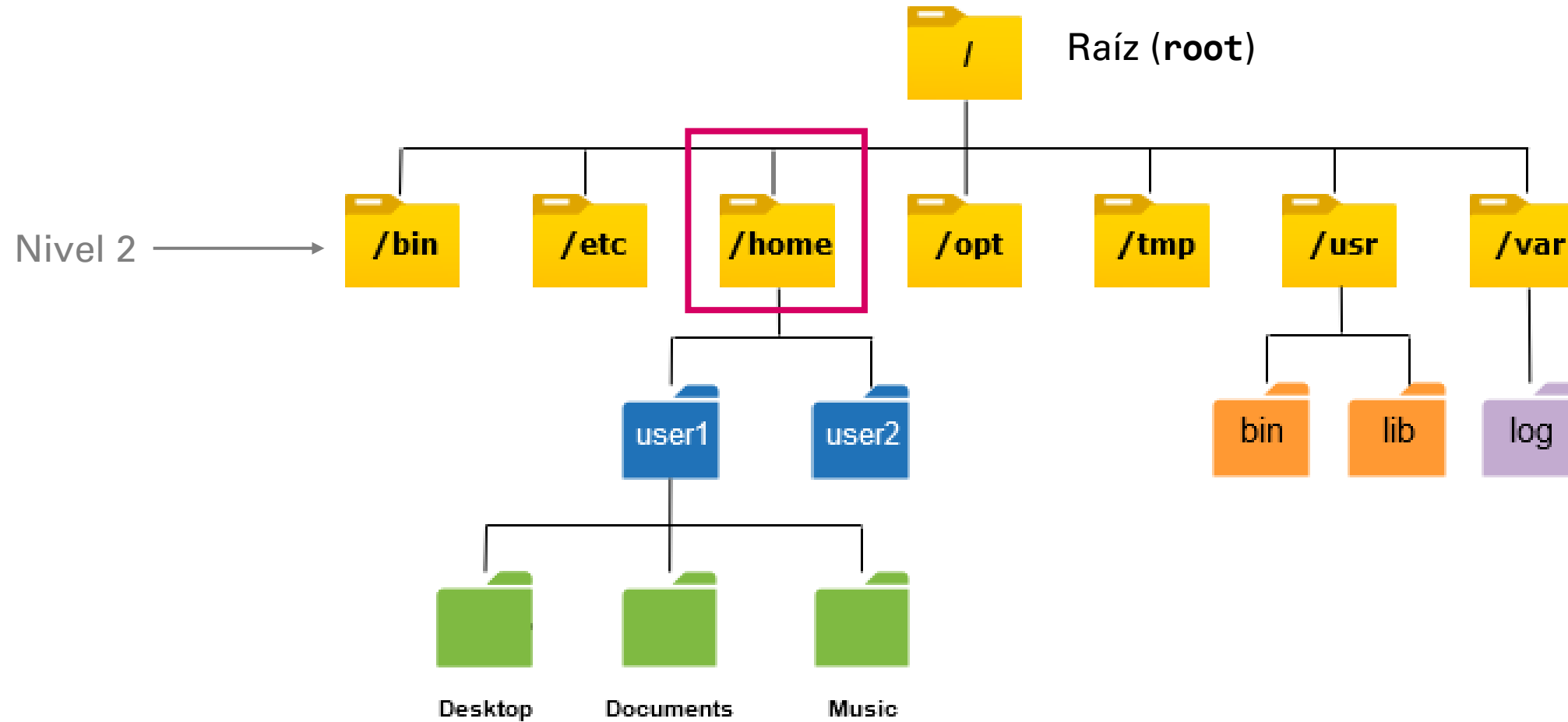
El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



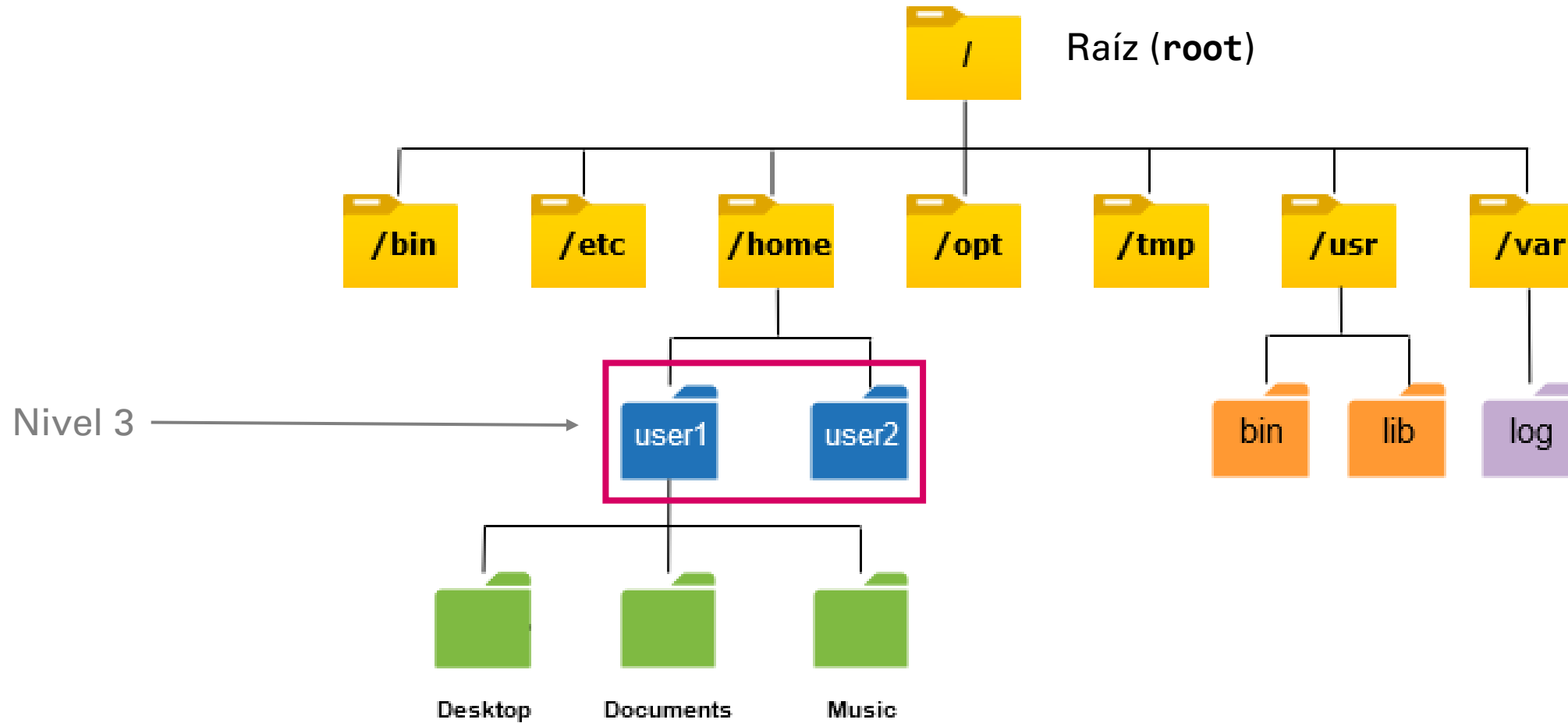
El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



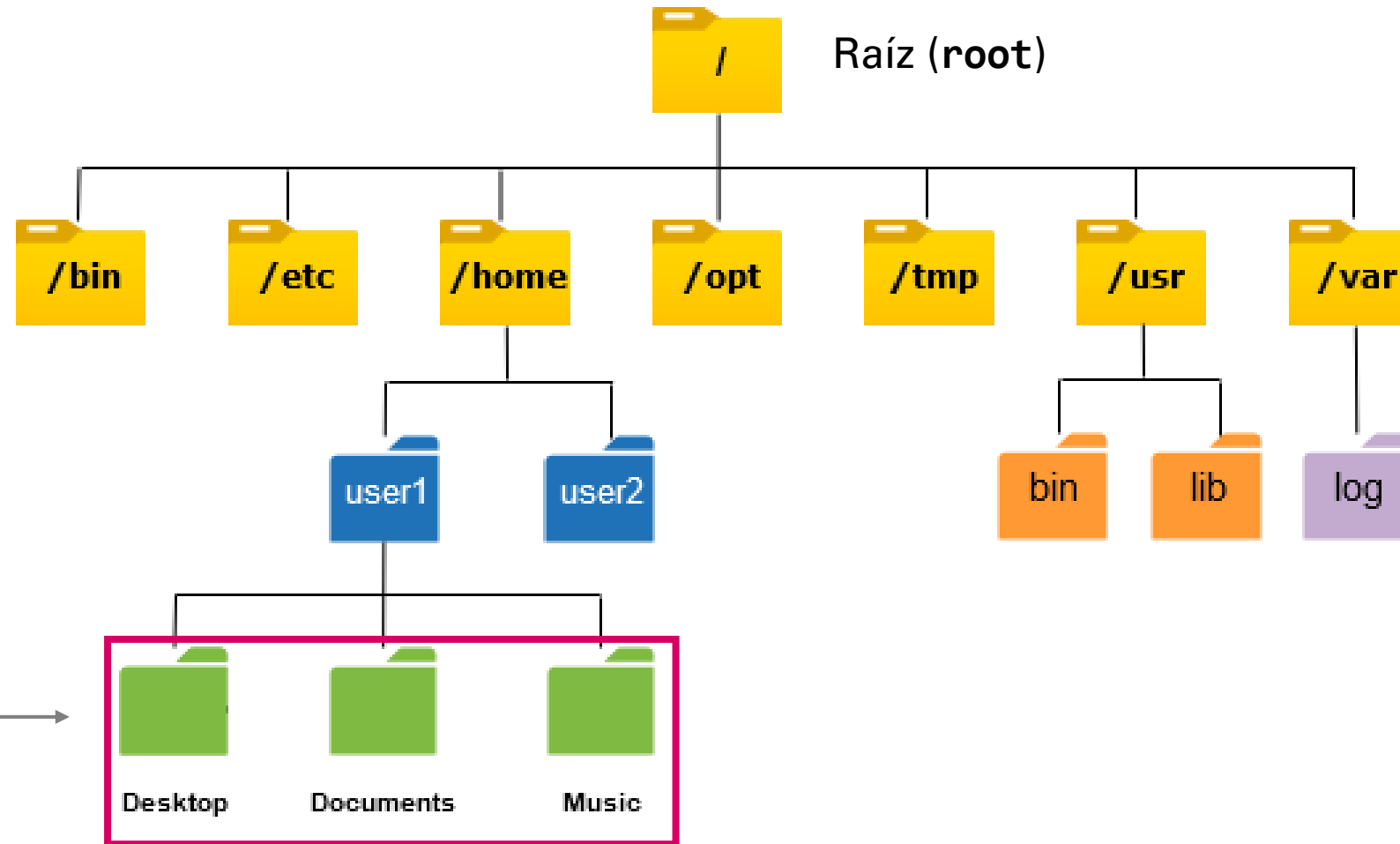
El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



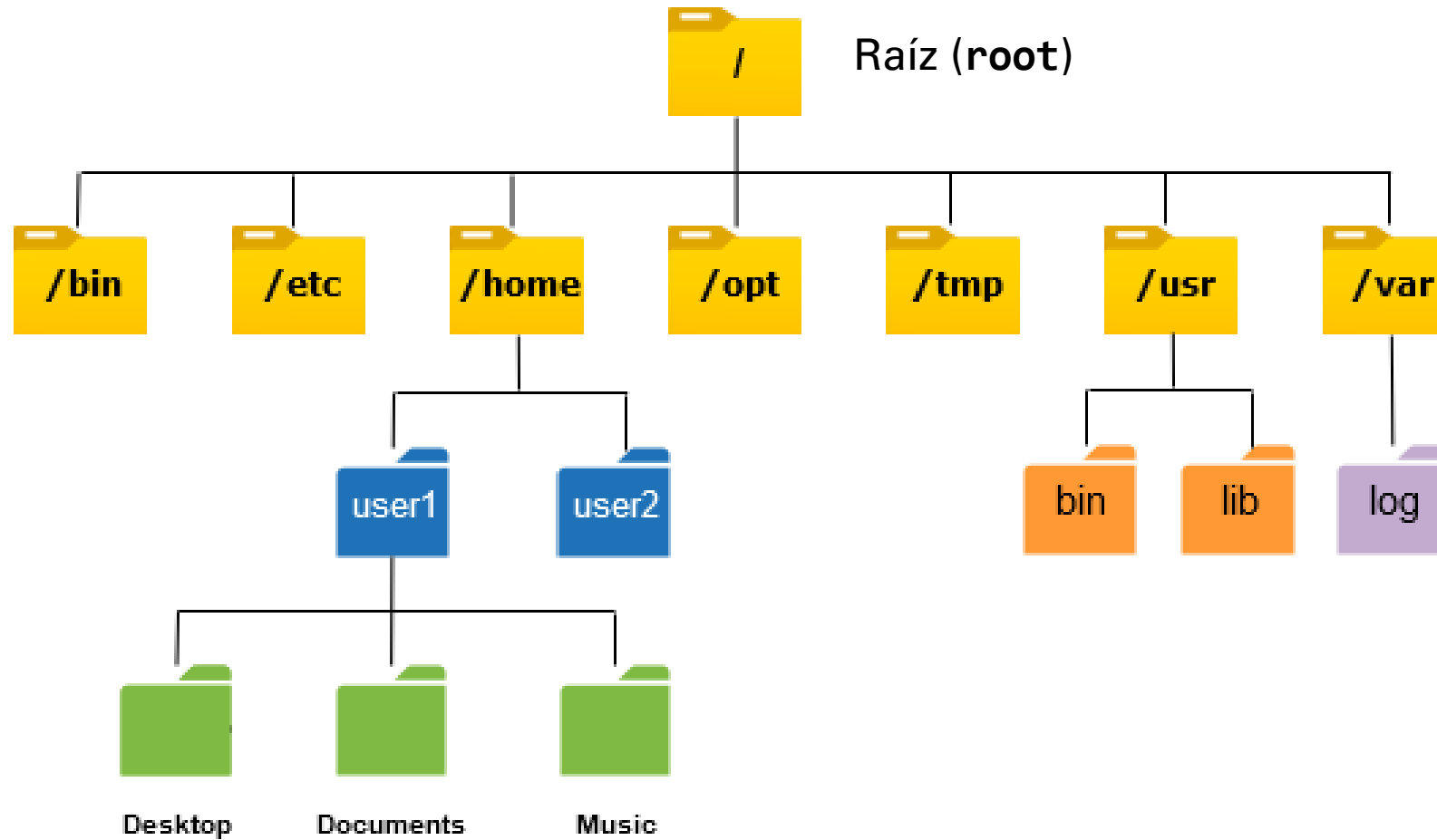
El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



El sistema de archivos de Unix es un árbol donde cada directorio es un nodo



+

•

Nivel ... →

Podemos recorrer el árbol de archivos con el comando **cd**

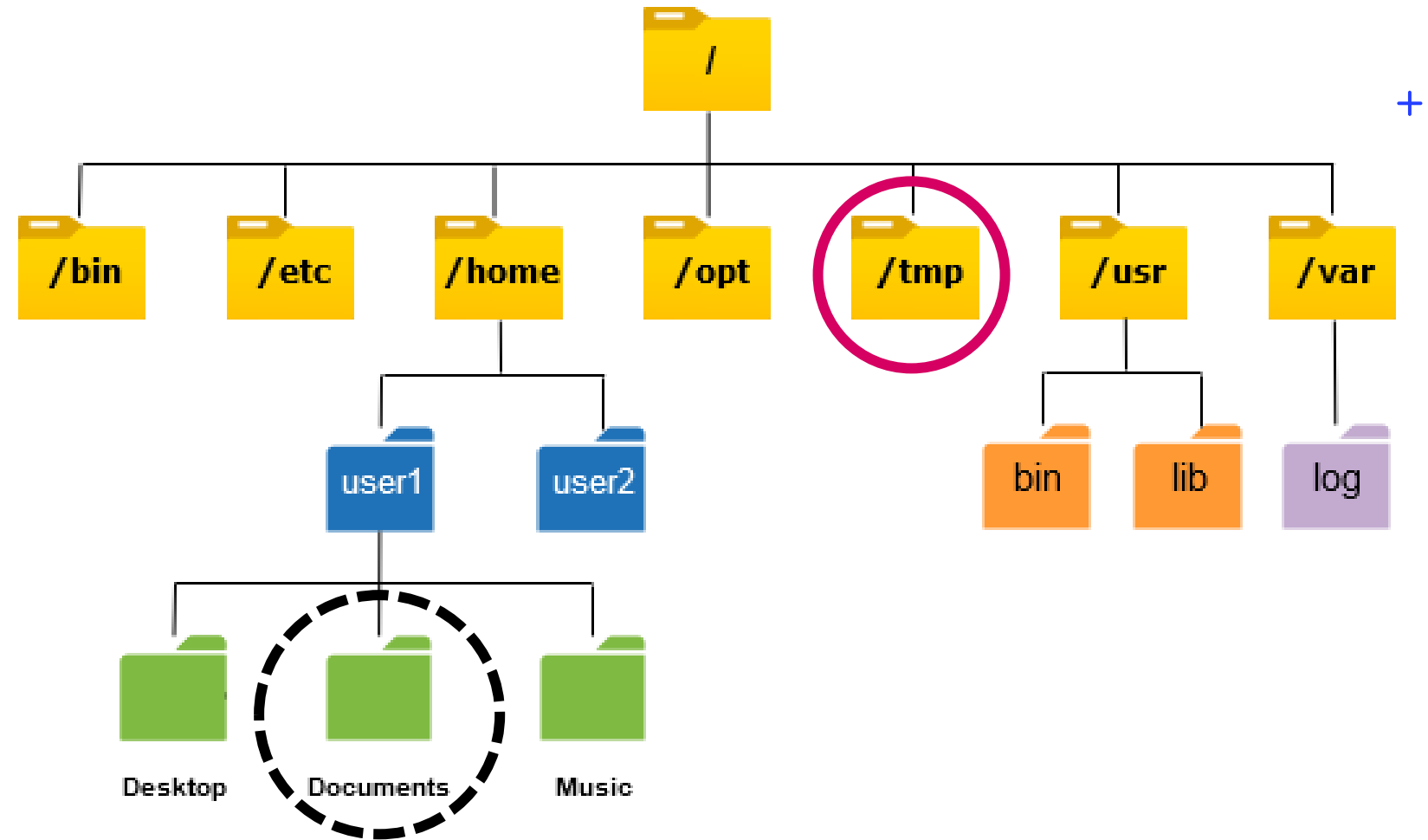


```
user@whereami $ cd <path>
```

+

•

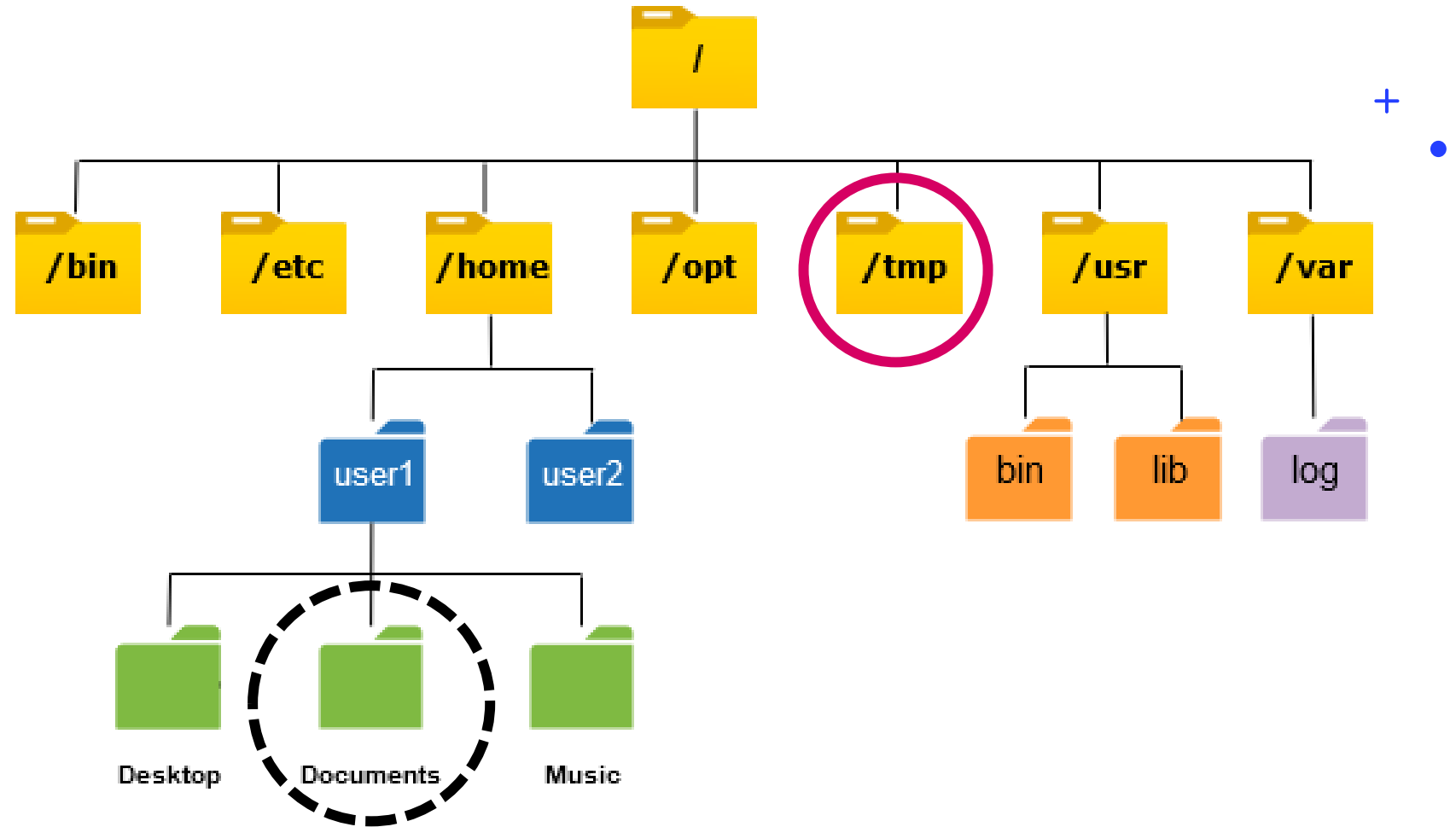
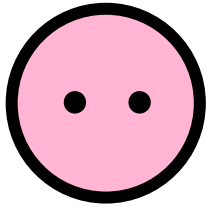
En unix tenemos dos “movimientos” para construir los paths



En unix tenemos dos “movimientos” para construir los paths



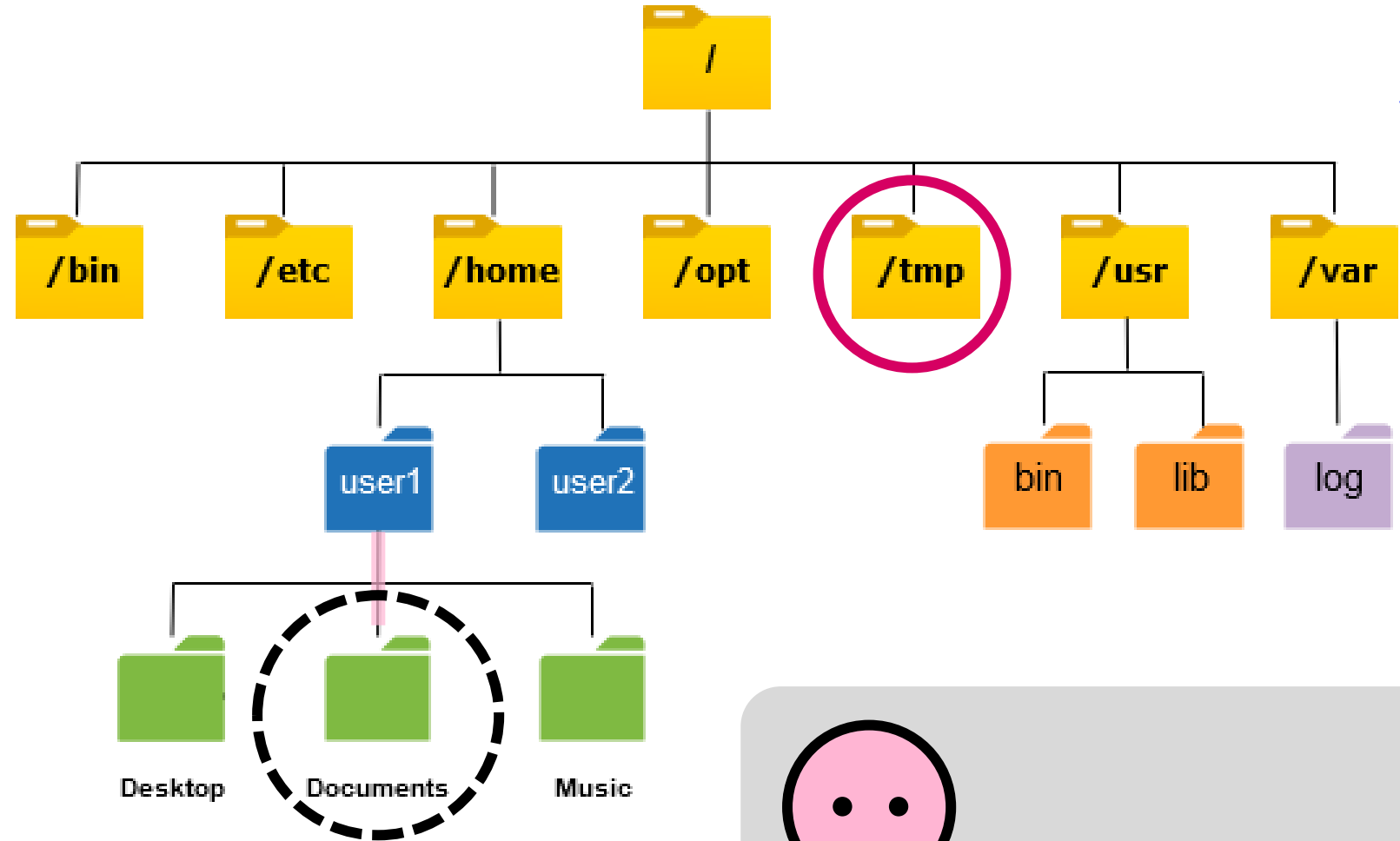
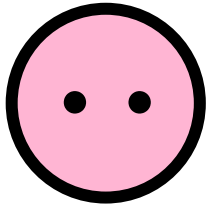
Arriba



En unix tenemos dos “movimientos” para recorrer el árbol de archivos

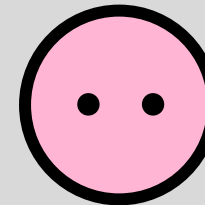


Arriba



+

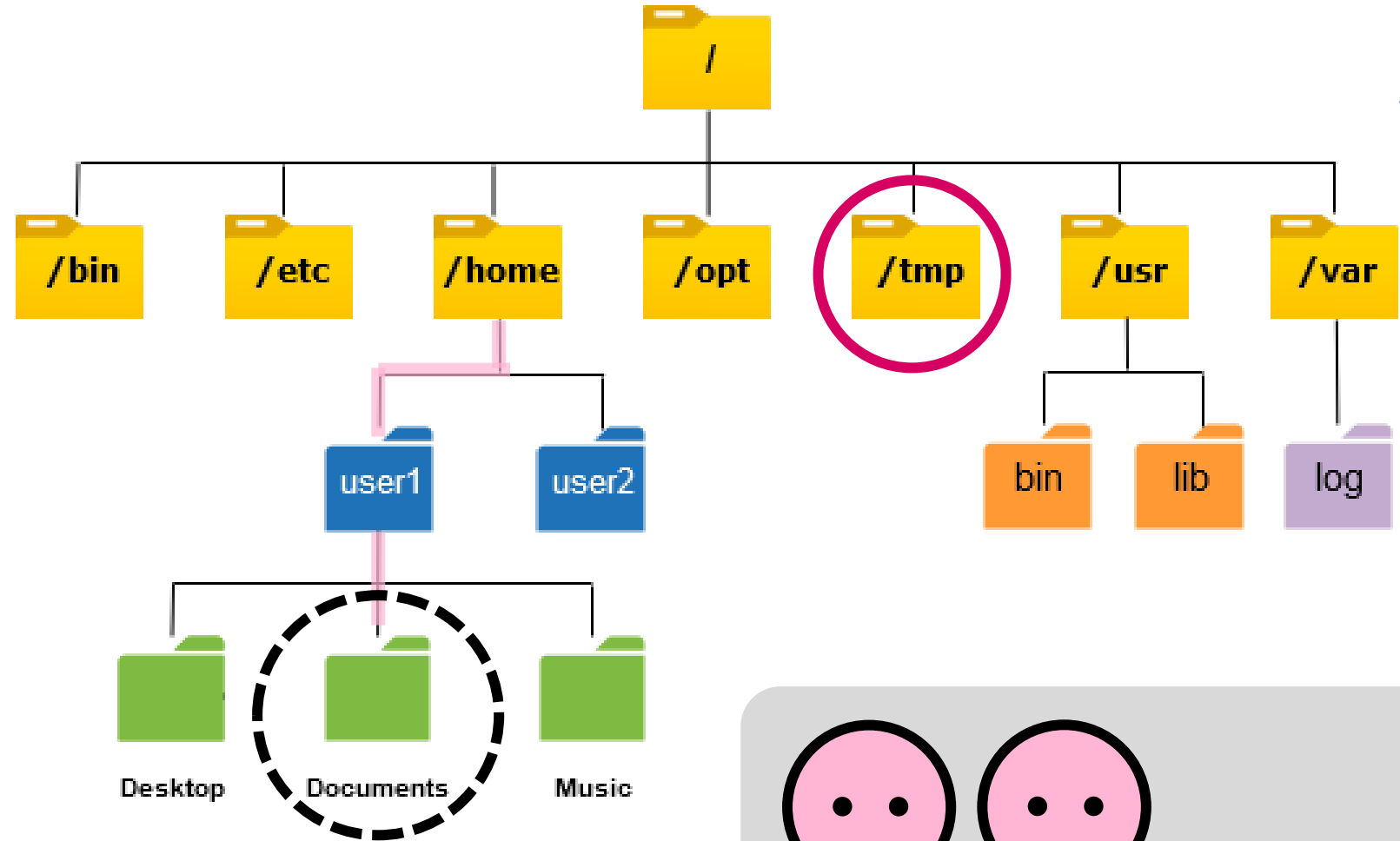
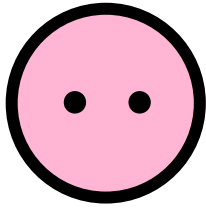
•



En unix tenemos dos “movimientos” para recorrer el árbol de archivos

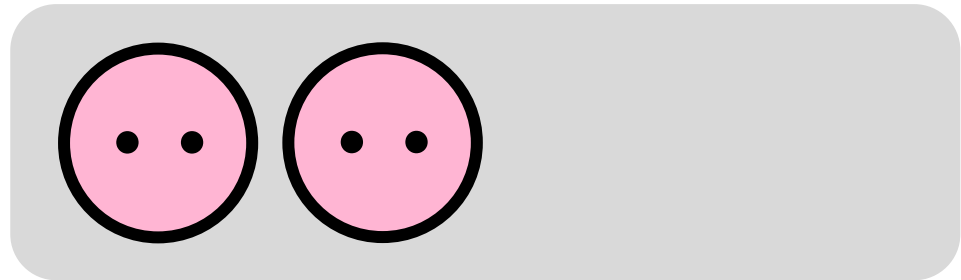


Arriba



+

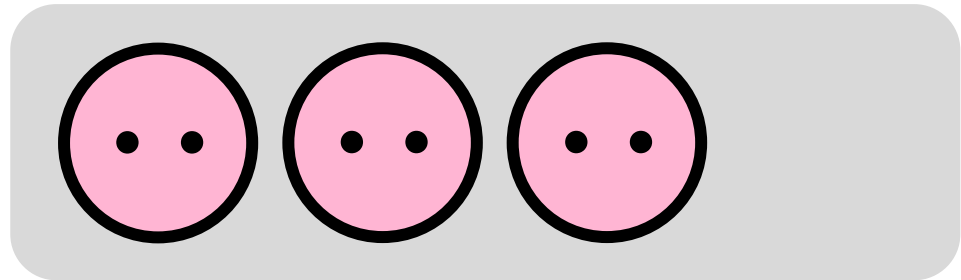
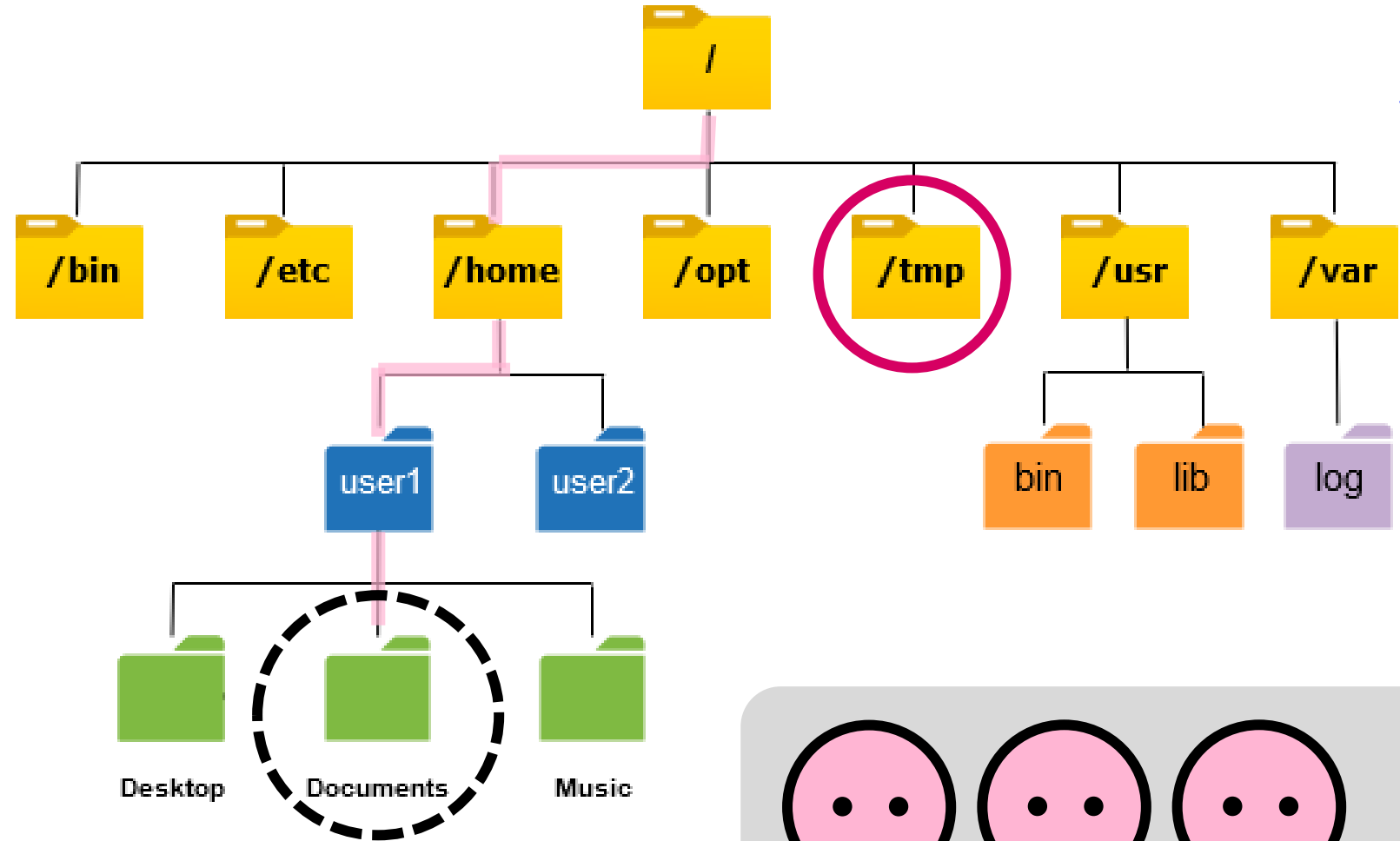
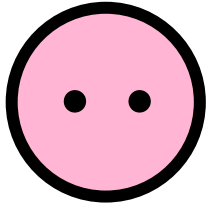
•



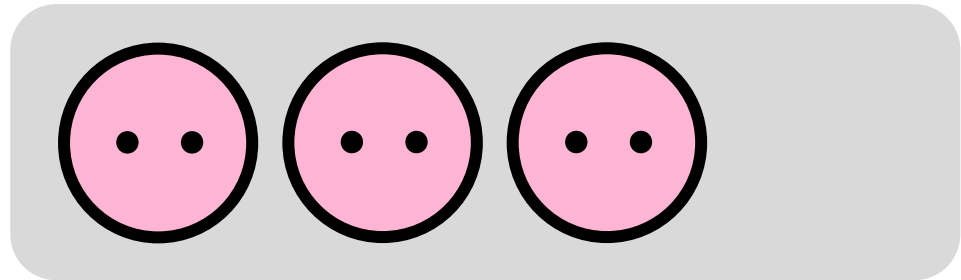
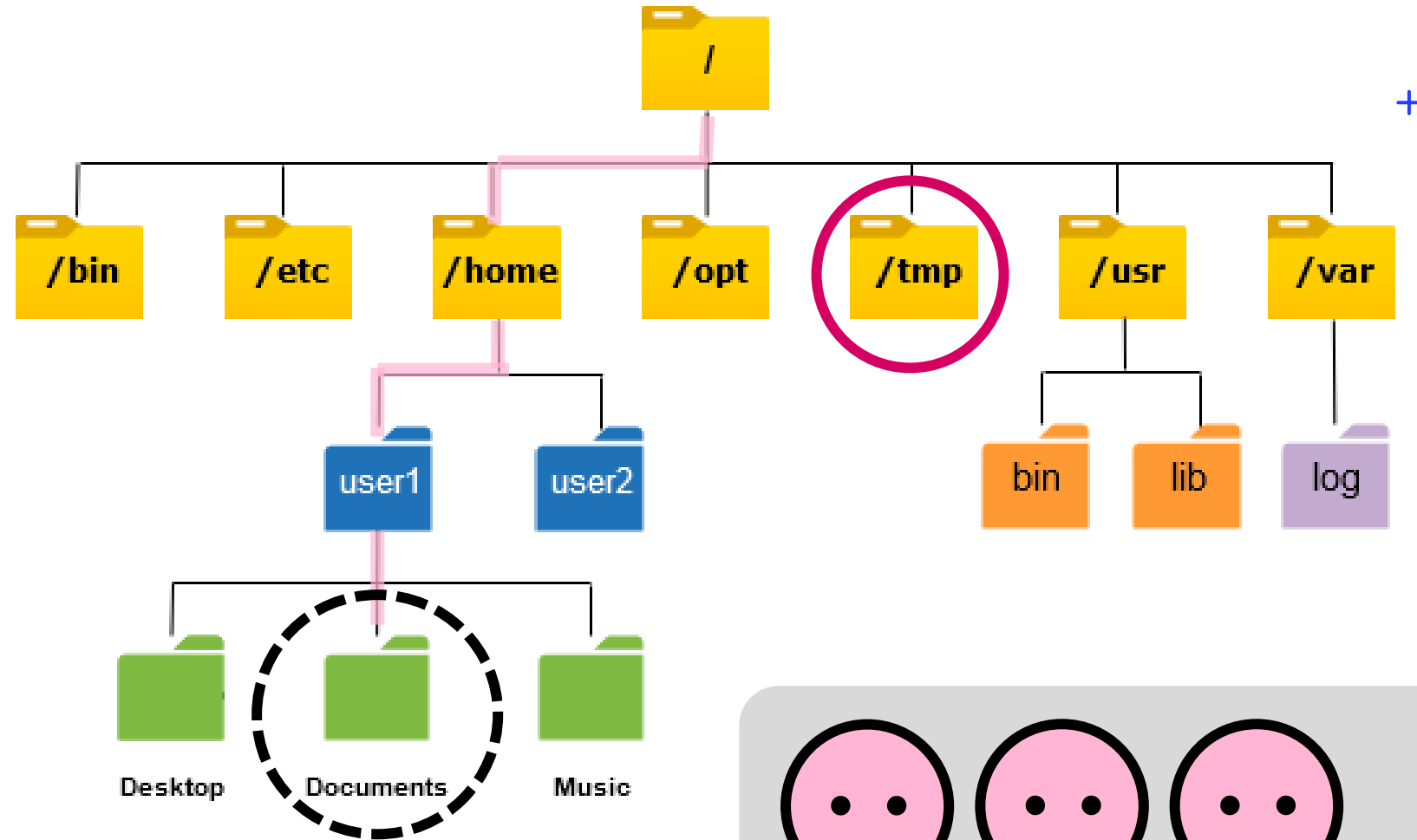
En unix tenemos dos “movimientos” para recorrer el árbol de archivos



Arriba



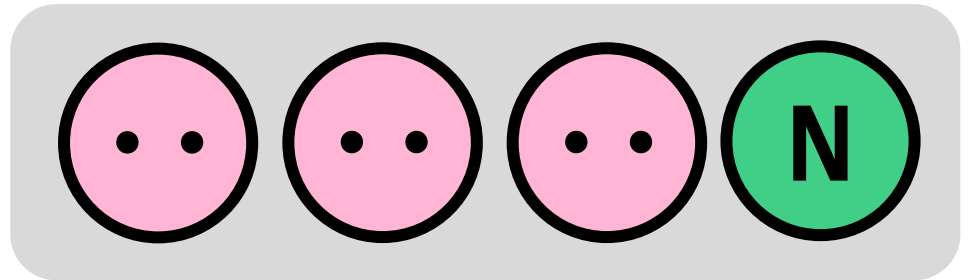
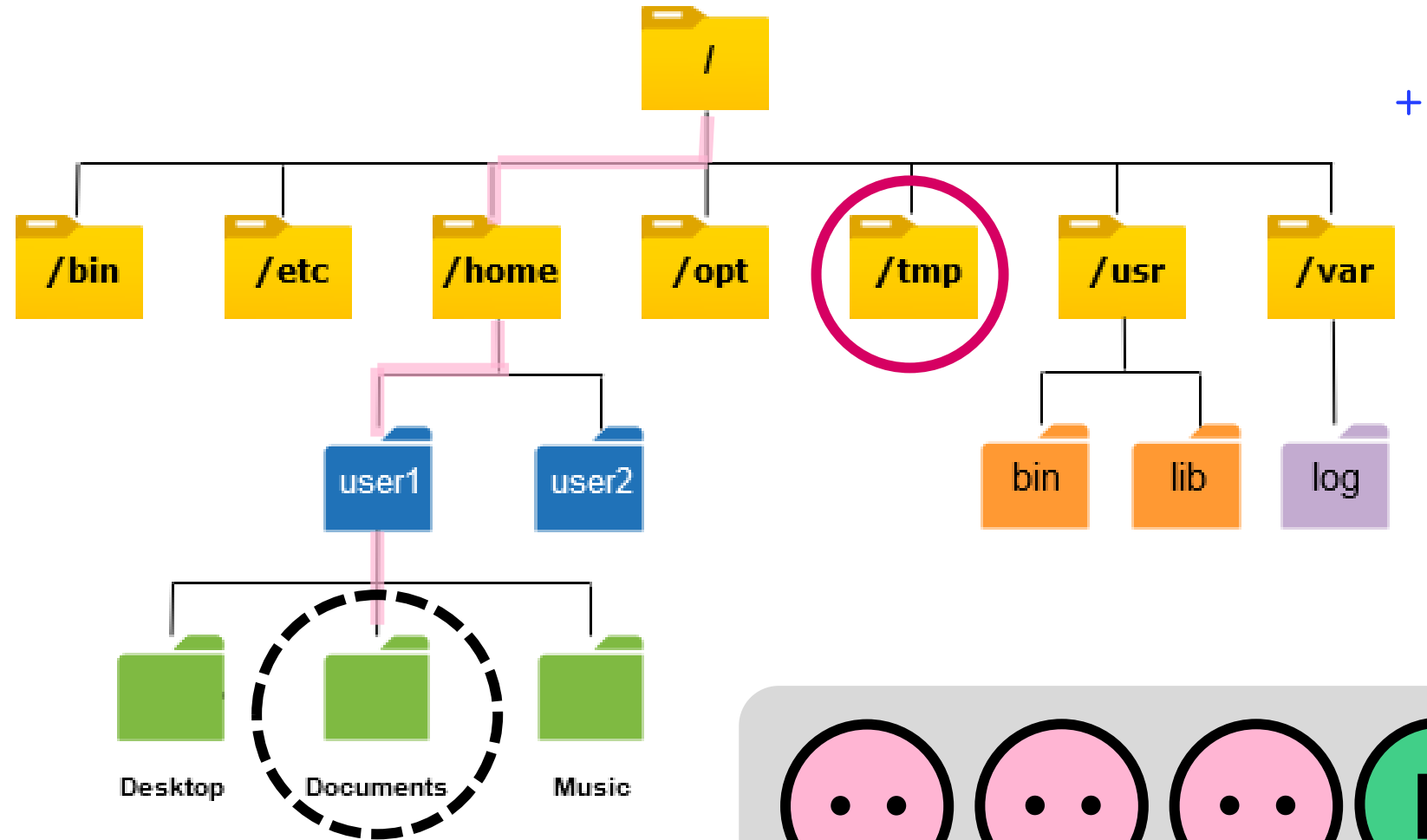
En unix tenemos dos “movimientos” para recorrer el árbol de archivos



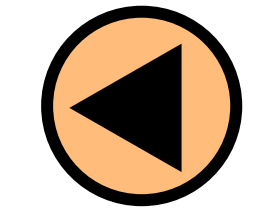
En unix tenemos dos “movimientos” para recorrer el árbol de archivos



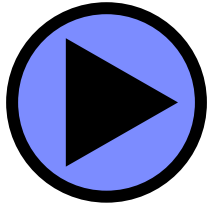
Nombre de la carpeta



En unix tenemos dos “movimientos” para recorrer el árbol de archivos



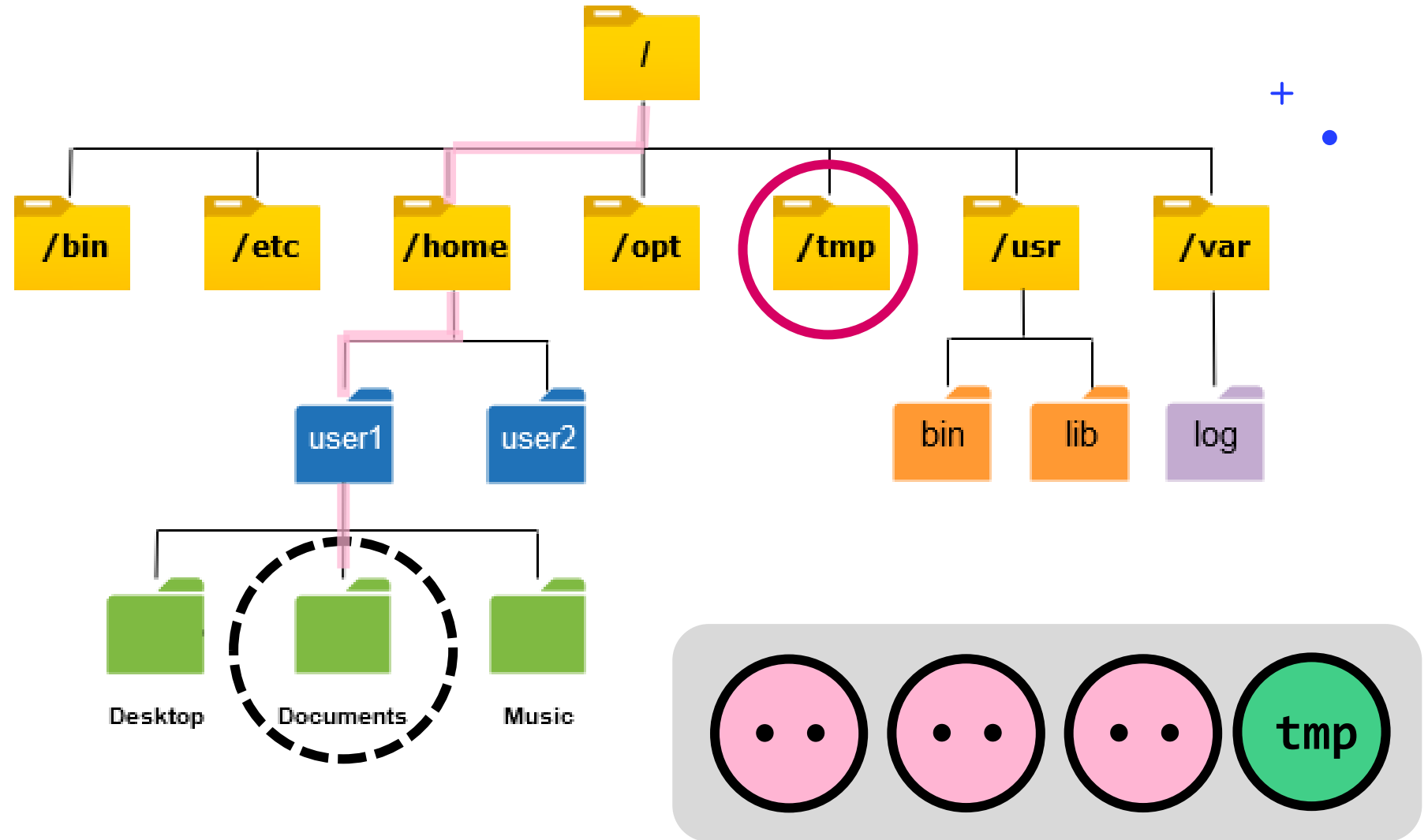
Abajo
izquierda



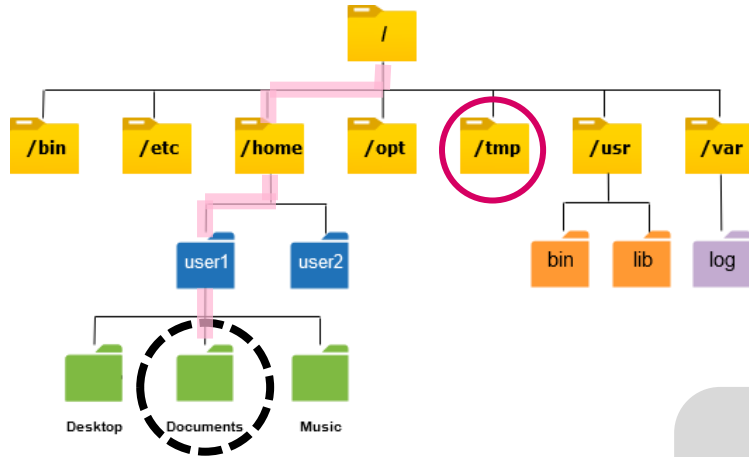
Abajo
derecha



Nombre de la
carpeta

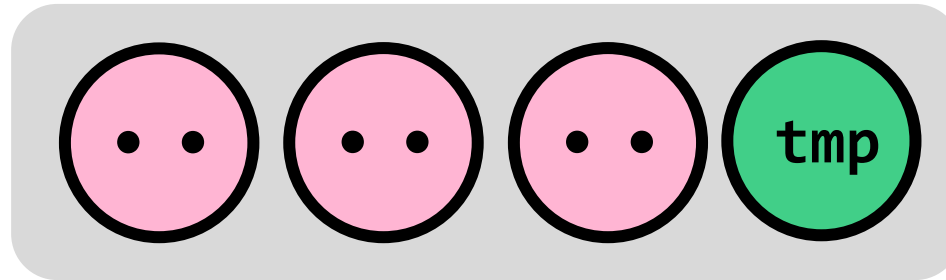


En línea de comandos sería ...

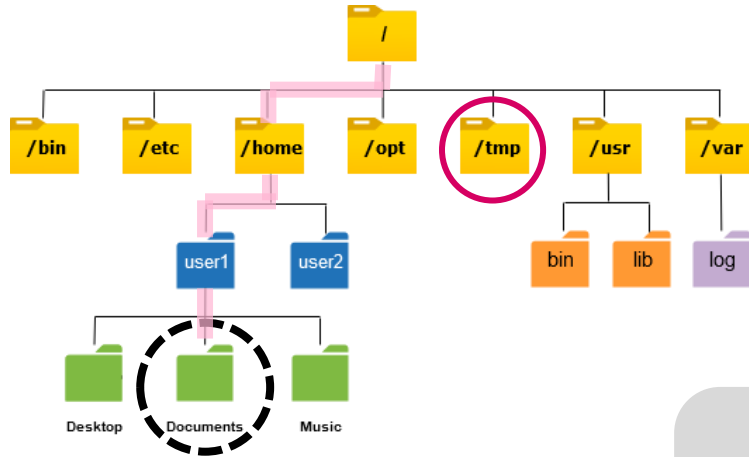


+

•



En línea de comandos sería ...

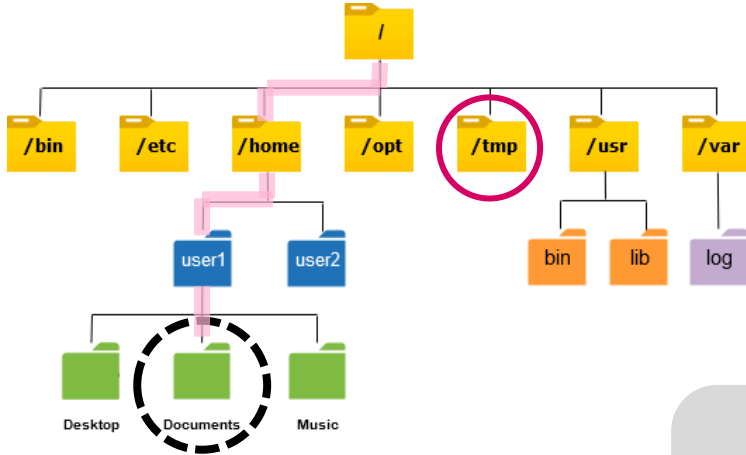


+

•

• • • • tmp

En línea de comandos sería ...

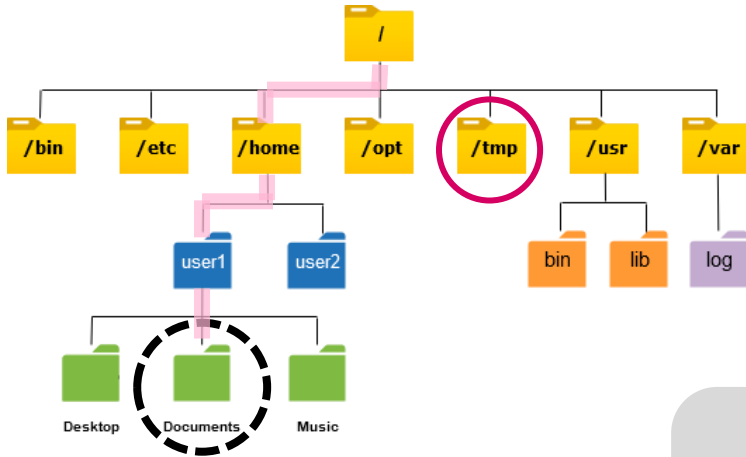


+

•

.. / .. / .. / tmp

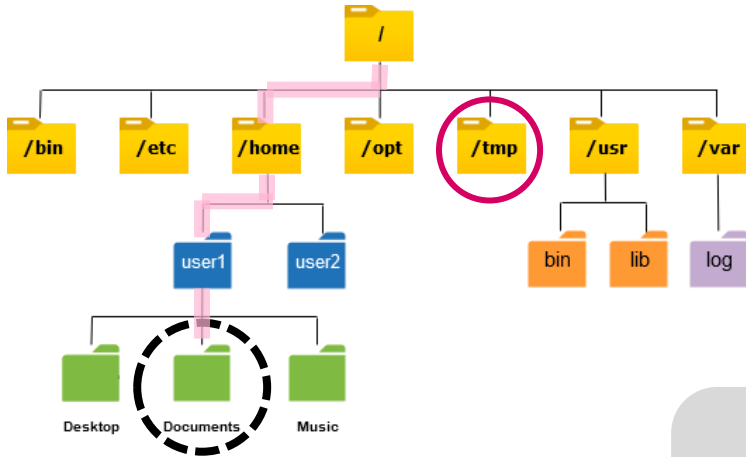
En línea de comandos sería ...



.. / .. / .. / tmp

\$ cd <path>

En línea de comandos sería ...



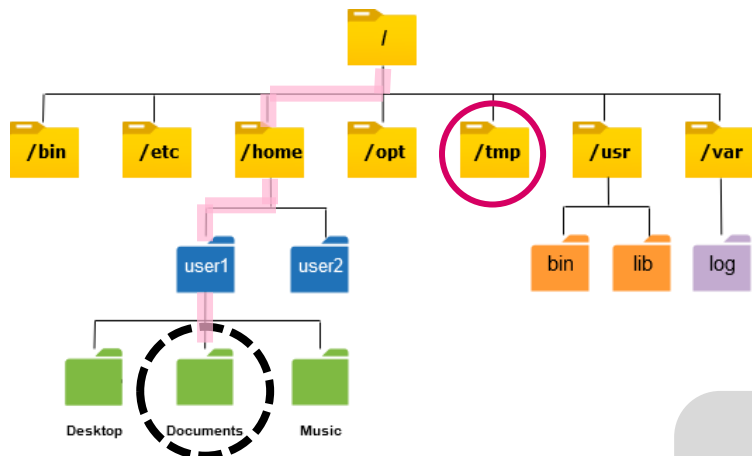
+

•

../ ../ ../ tmp

```
$ cd ../../../../tmp
```

En línea de comandos sería ...



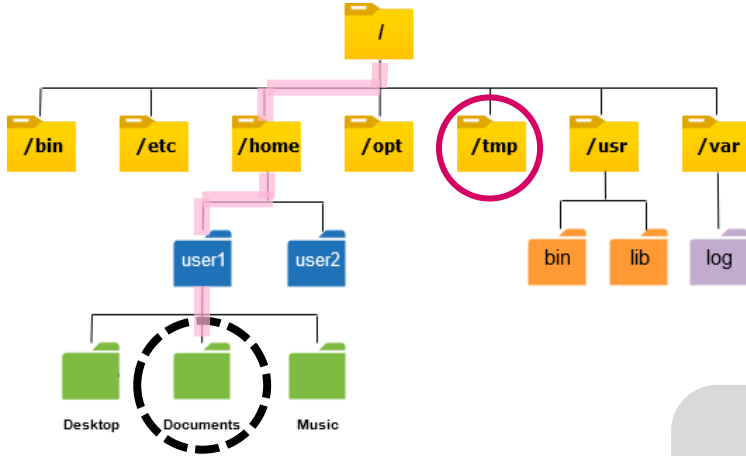
../ ../ ../ tmp

\$ cd ../ ../ ../tmp



No hay espacios entre los "../ "

En línea de comandos sería ...

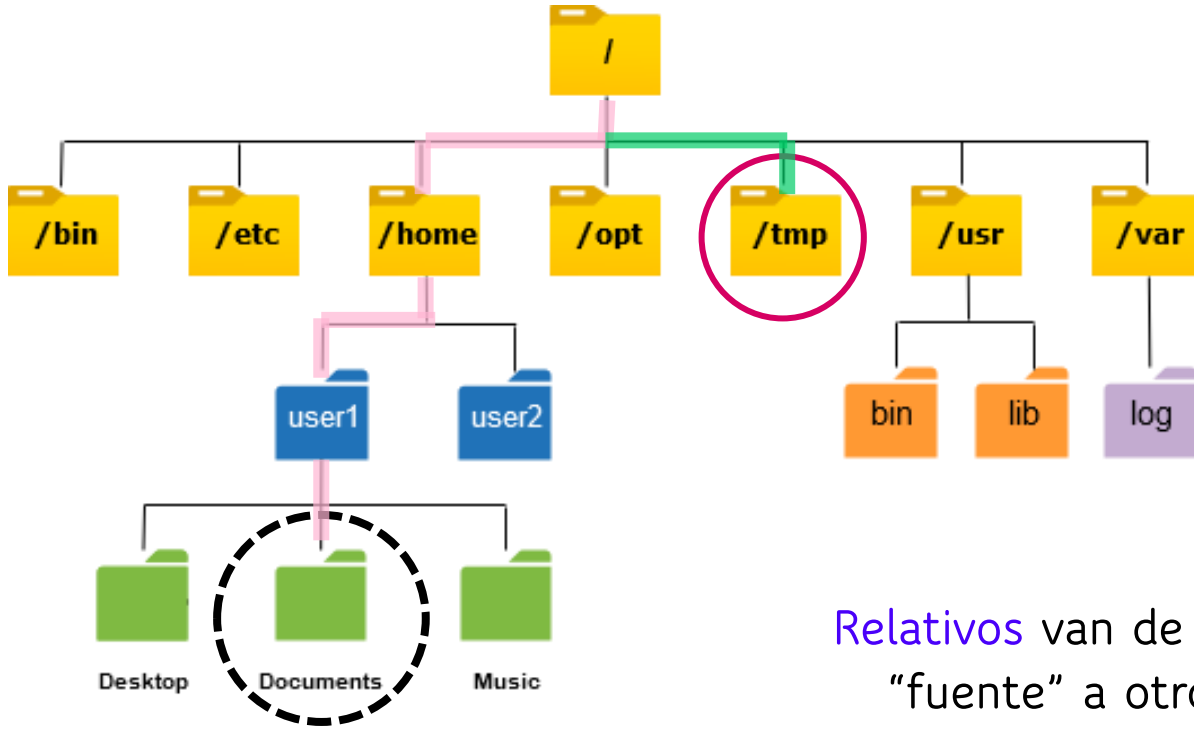


../ ../ ../ tmp

\$ cd ../ ../ ../ tmp



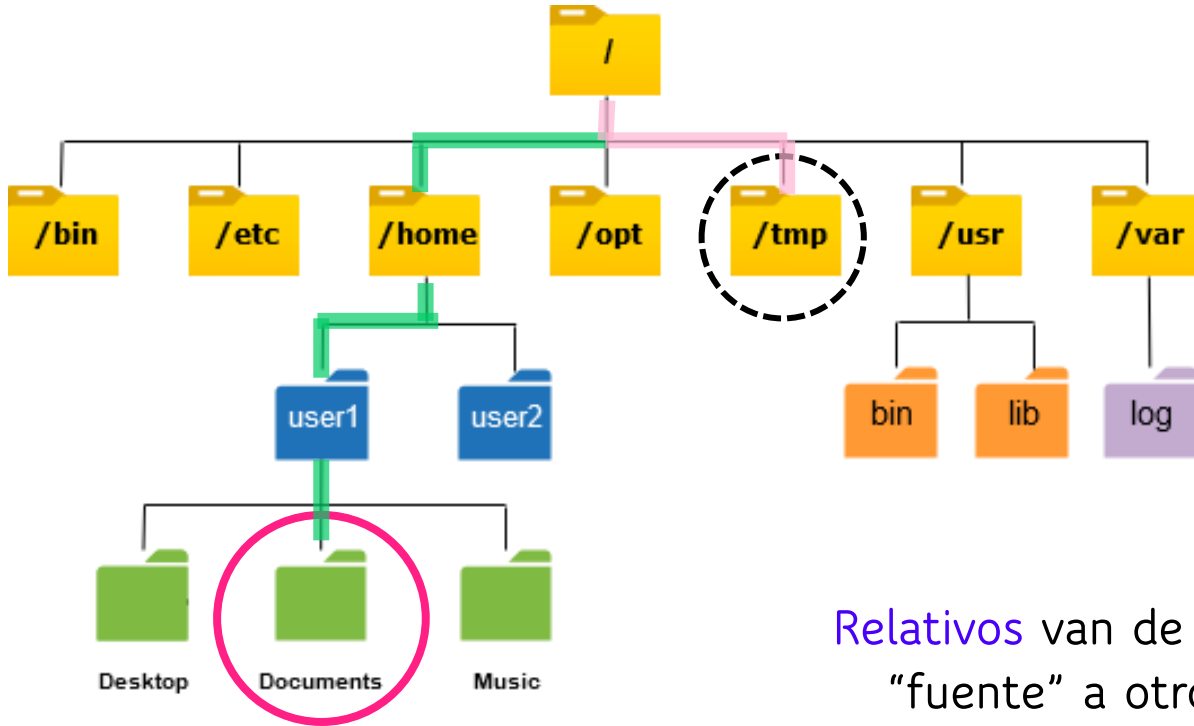
Existen dos tipos de caminos: relativos y absolutos



Relativos van de un directorio
"fuente" a otro "destino"

```
$ cd ../../../../tmp
```

Existen dos tipos de caminos: relativos y absolutos

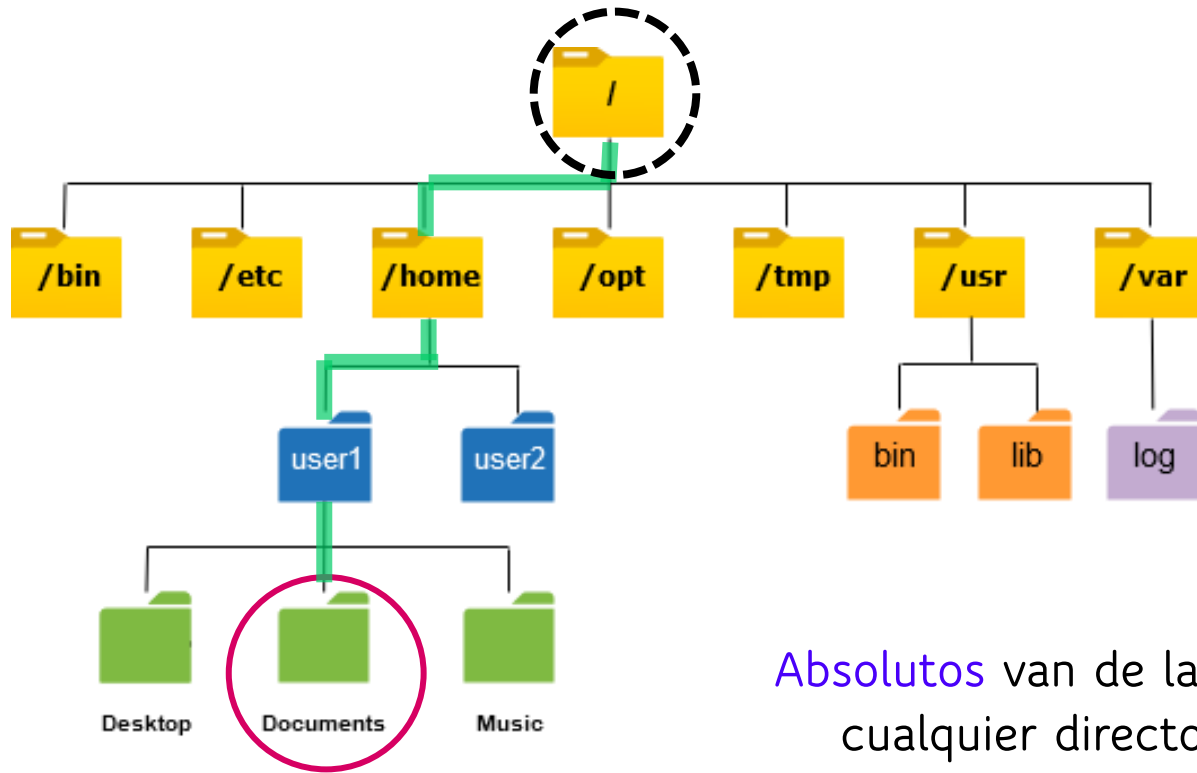


Relativos van de un directorio
"fuente" a otro "destino"

```
$ cd ../../../../tmp
```

```
$ cd ../home/user1/Documents
```

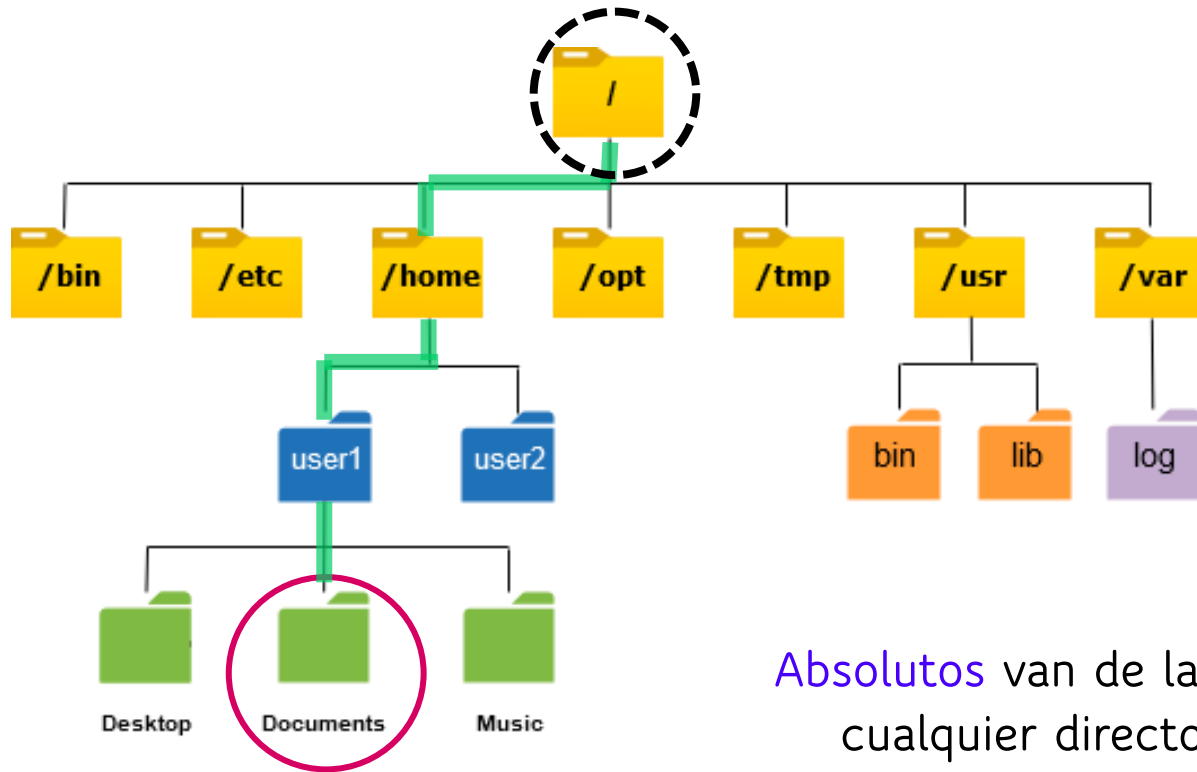

Existen dos tipos de caminos: relativos y absolutos



Absolutos van de raíz del árbol a cualquier directorio (archivo)

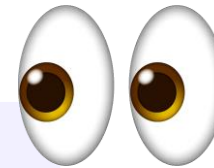
```
$ cd /home/user1/Documents
```

Existen dos tipos de caminos: relativos y absolutos

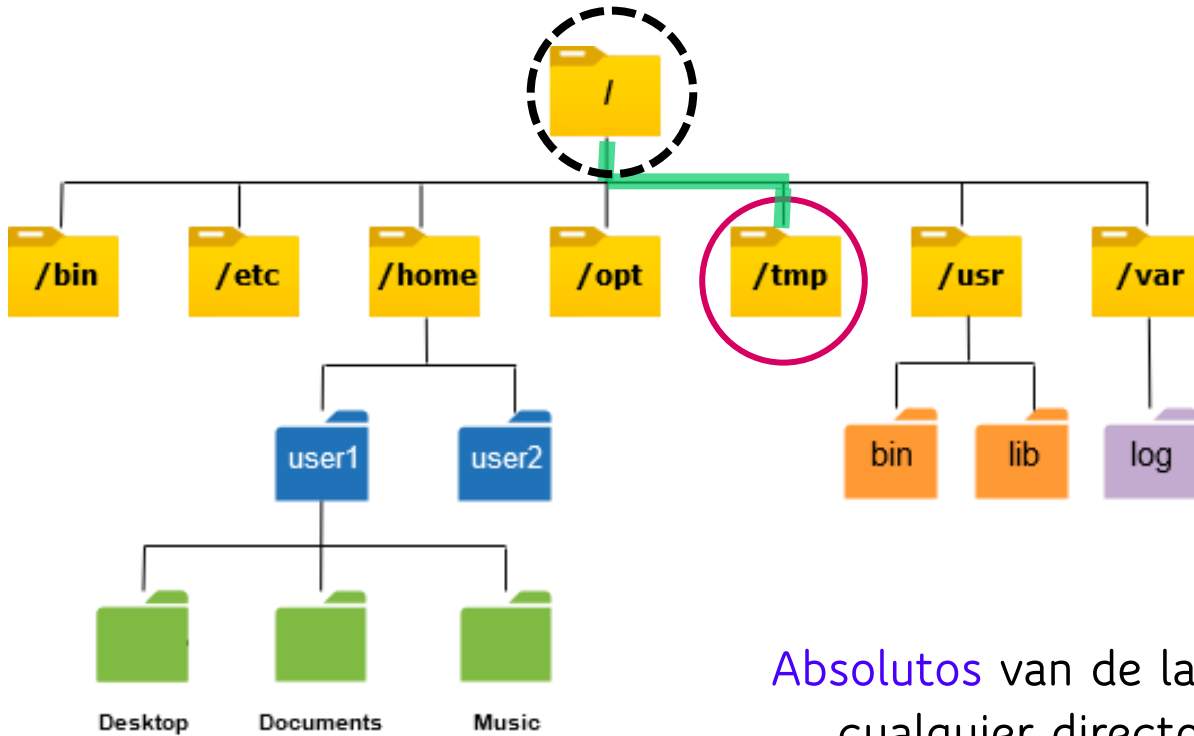


Absolutos van de la raíz del árbol a cualquier directorio (archivo)

```
$ cd /home/user1/Documents
```



Existen dos tipos de caminos: relativos y absolutos

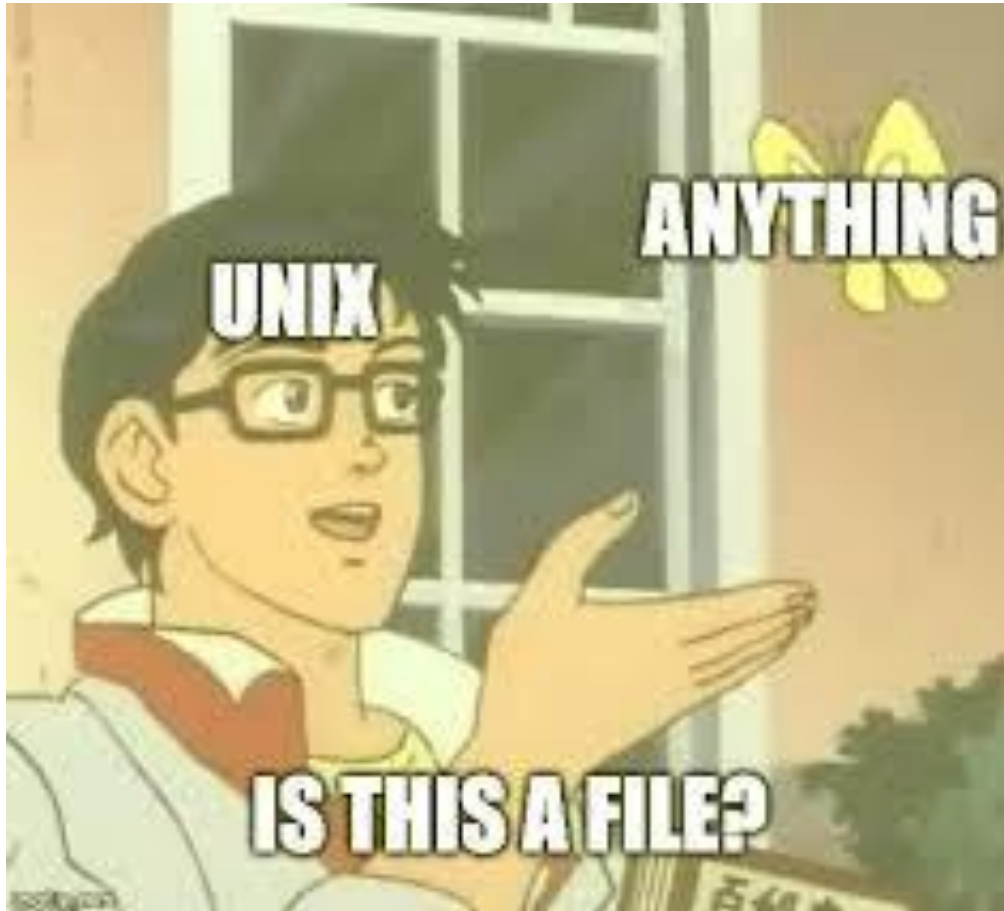


Absolutos van de la raíz del árbol a cualquier directorio (archivo)

```
$ cd /home/user1/Documents
```

```
$ cd /home/tmp
```

En Unix hay varios tipos de archivos:

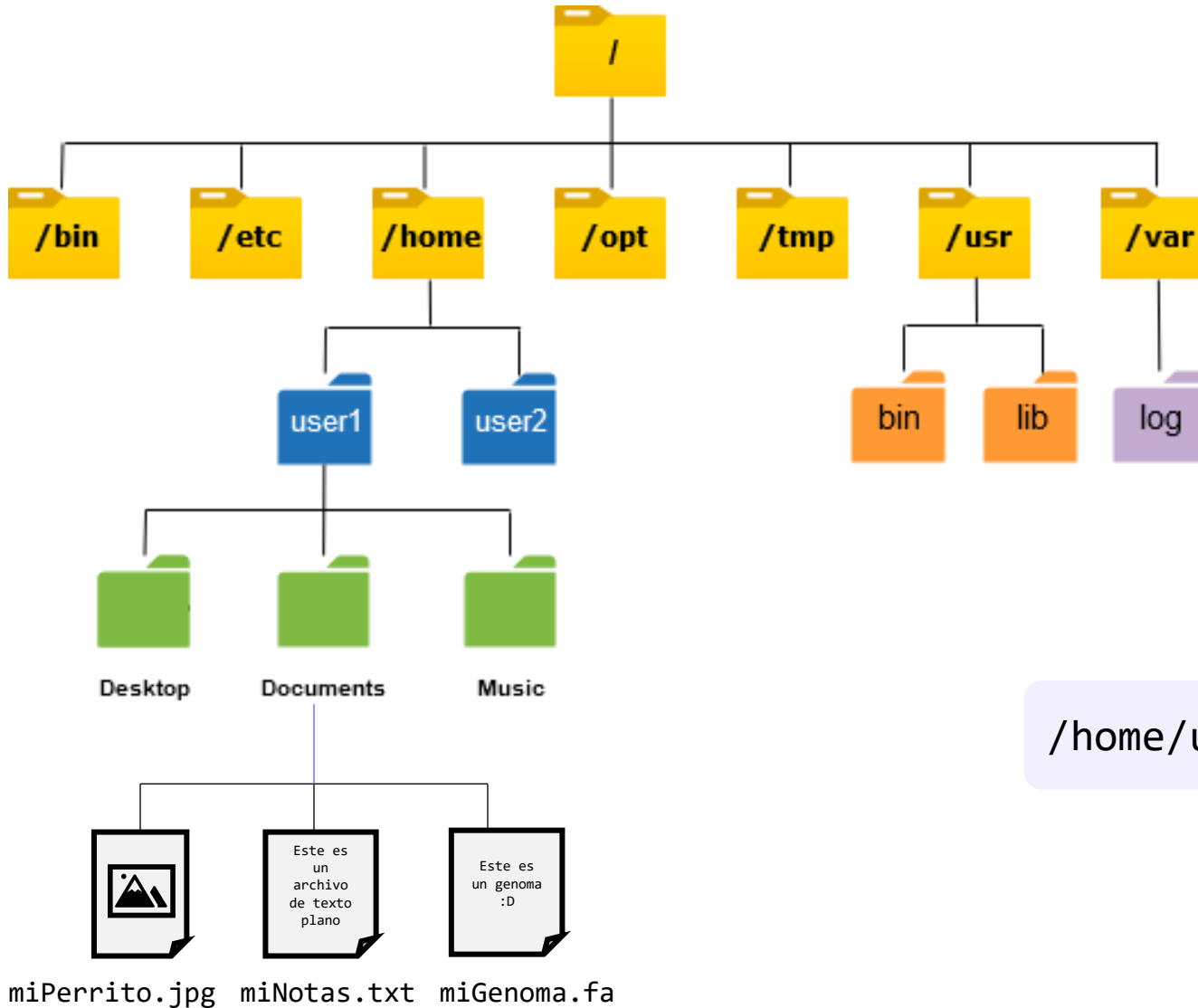


1. Archivos regulares
2. Directorios
3. Links simbólicos

+

•

Cada directorio contiene: directorios y archivos



`/home/user1/Documents/miGenoma.fa`



¡Manos a la obra! ✨

print working directory

```
$ pwd
```

list

```
$ ls [-a,-l,-h] <path>
```

make directory

```
$ mkdir <nombre o path>
```

change directory

```
$ cd <path>
```

move

```
$ mv <pathDeDónde> <pathADonde>
```

copy

```
$ cp [-r] <pathDeQué> <pathADonde>
```

remove

```
$ rm [-rf] <path>
```

Caracteres especiales: .. (subir) .(aquí) ~(absPathMiHome)

TRABAJANDO CON ARCHIVOS



¡Manos a la obra! ✨



head

```
$ head [-n] <path>
```

tail

```
$ tail [-n] <path>
```

more

```
$ more <path>
```

less

```
$ less <path>
```

nano

```
$ nano <path>
```

concatenate

```
$ cat <path>
```

word count

```
$ wc [-l,c] <path>
```

sort

```
$ sort [-r,n] <path>
```

uniq

```
$ uniq [-c] <path>
```

cut

```
$ cut [-d,f] <path>
```

grep

```
$ grep [-i,v,f,c] <path1> <path2>
```

Caracteres especiales: * (comodín), > (salida) >> (append)

COMPOSICIÓN DE COMANDOS



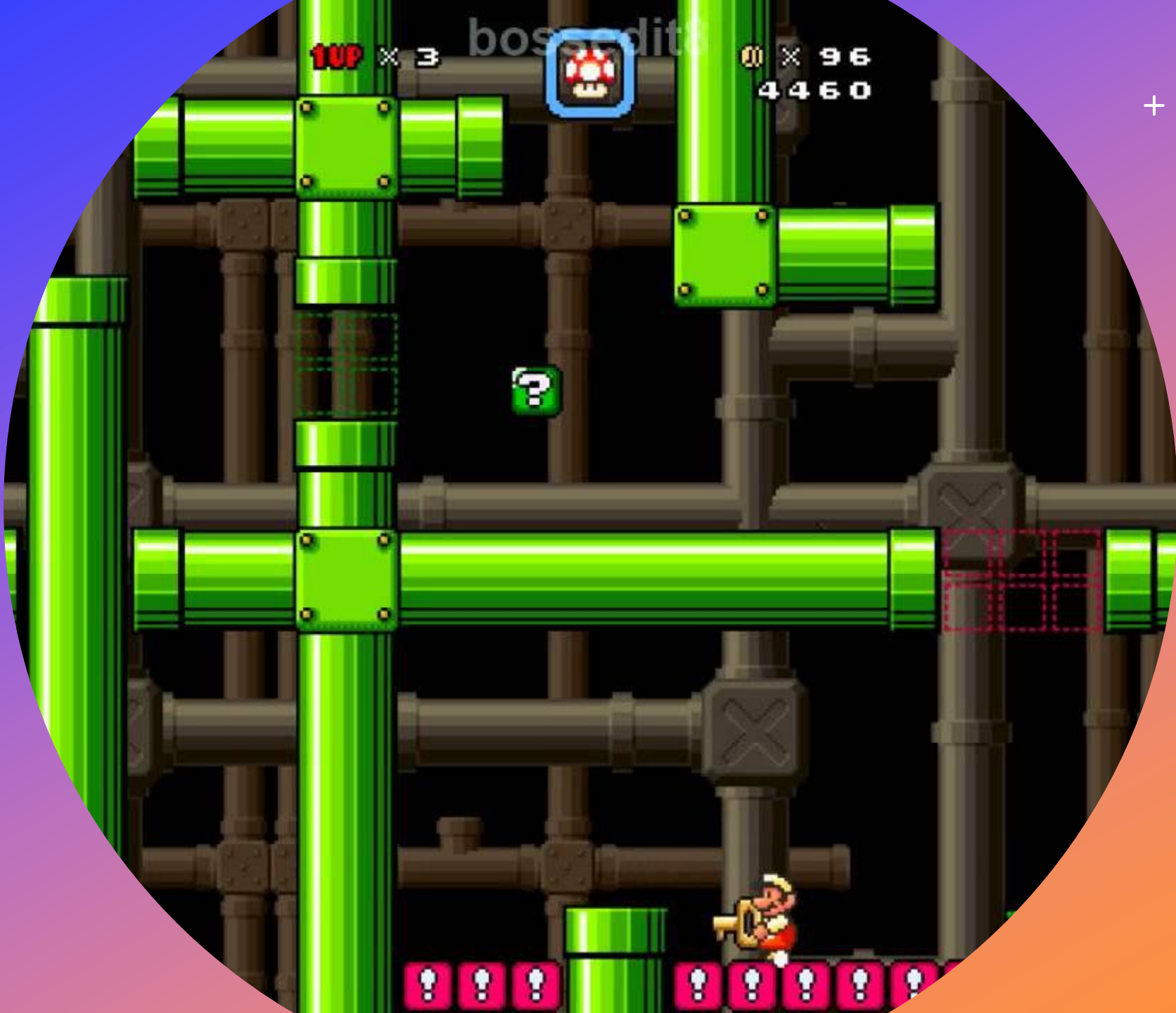
A veces necesitamos aplicar varios comandos a un archivo inicial para obtener el resultado que necesitamos



A veces necesitamos aplicar varios comandos a un archivo inicial para obtener el resultado que necesitamos



Tuberías o **pipes** ✨



boss dit

1UP x 3

0 x 96

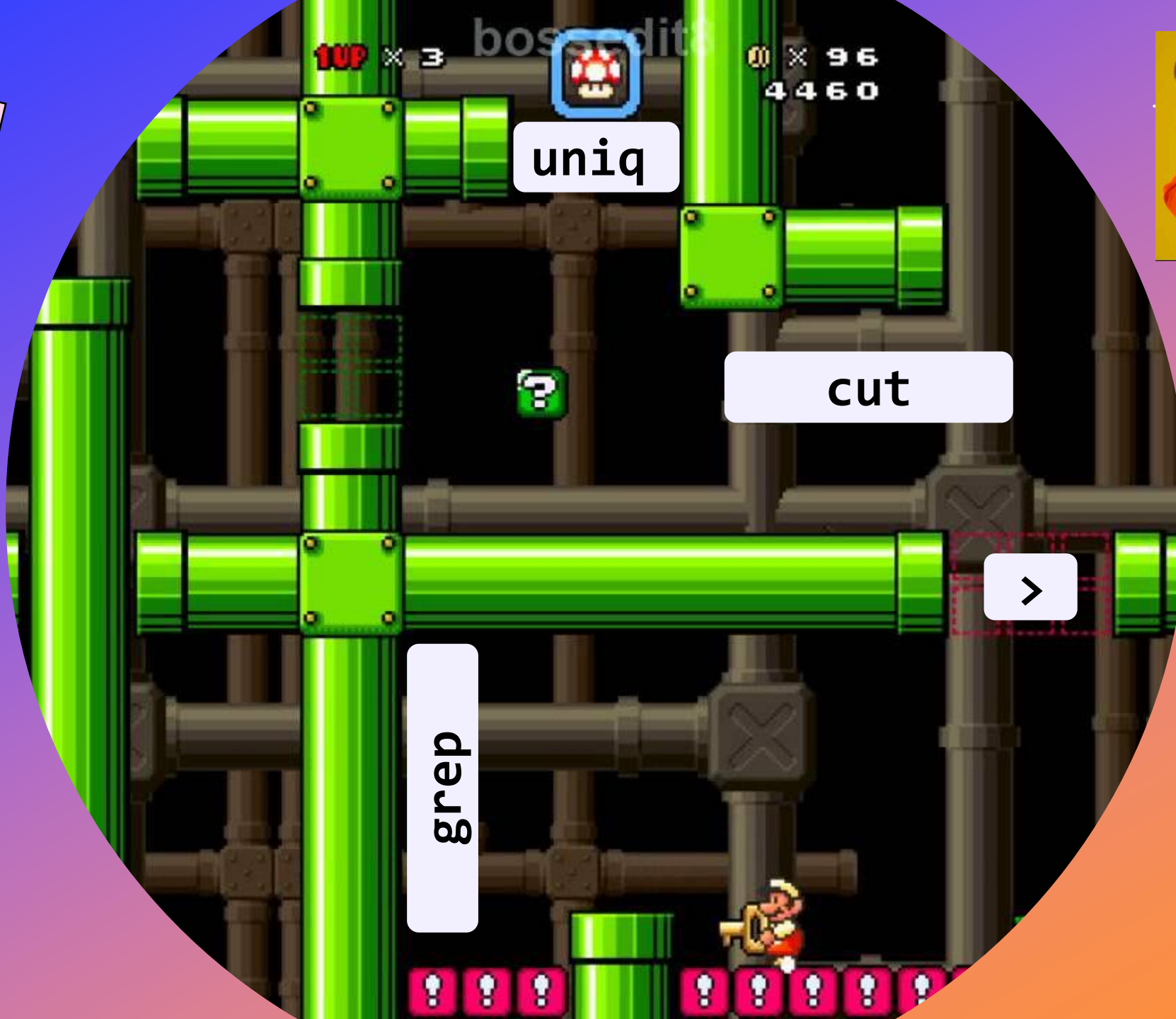
4460

?



Archivo de salida

Archivo
intermedio
v4.5.5.6.txt
.fasta_R1



uniq

cut

>

grep

Archivo
de
entrada



Archivo
de
entrada



Archivo de
salida

Archivo
de
entrada



cut

grep

uniq

>

Archivo de
salida



Programar es como hacer limonada (pt #2)

1. Llenar la jarra con agua
2. Cortar a la mitad los limones
3. Exprimir los limones en la jarra
4. Agregar 1 cucharadas de azúcar a la jarra
5. Revolver
6. Agregar hielos a la jarra





Programar es como hacer limonada (pt #2)

1. Exprimir los limones en la jarra
2. Agregar 1 cucharadas de azúcar a la jarra
3. Revolver
4. Agregar hielos a la jarra
5. Cortar a la mitad los limones
6. Llenar la jarra con agua





¡Manos a la obra! ✨

Caracteres especiales: | (pipe)

+

•

¡GRACIAS A TOD*S!

\$ exit



+ • PRÓXIMAMENTE