

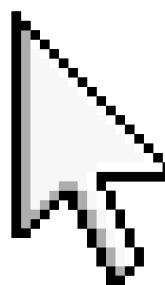
Home Content Contact

Programación en

BASH

BARNALD BOCKER - ISAAC FLORES

Start



[Home](#)[Content](#)[Contact](#)

- Comandos básicos (**touch**, **mkdir**, **apt-get**, **cp**, **cat**, **head**, **tail**, **grep**, **echo**)
- Uso **básico** de **vim** (editor de texto)
- Pipelines y su importancia
- Creación de archivos .sh

[Home](#)[Content](#)[Contact](#)

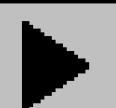
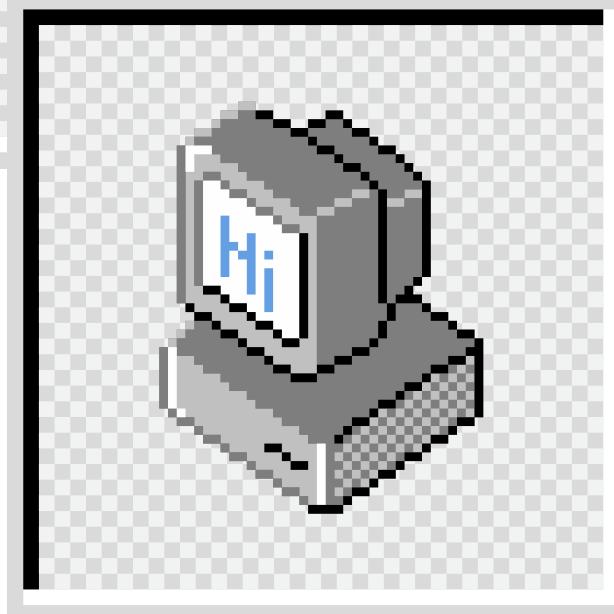
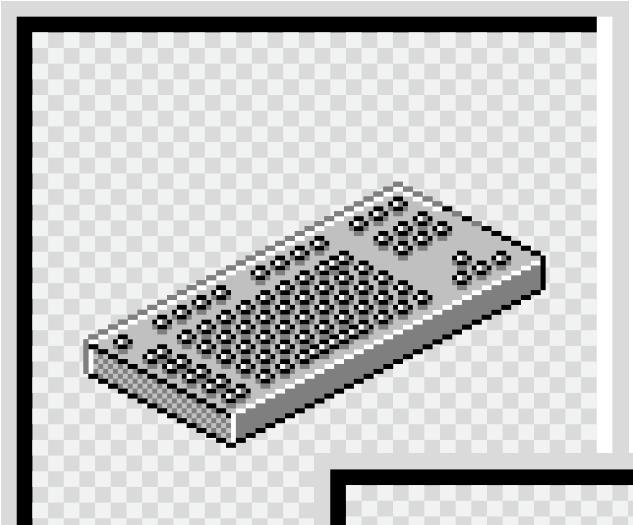
- touch
 - Crea archivos
 - Syntax: touch nombre”.”terminación”
- mkdir
 - Crea directorios
 - Syntax: mkdir “nombre”
- cp
 - Copia archivos/directorios
 - Ejemplos:
 - cp ~/constanciaisaac .
 - cp -r ~/documentosisaac

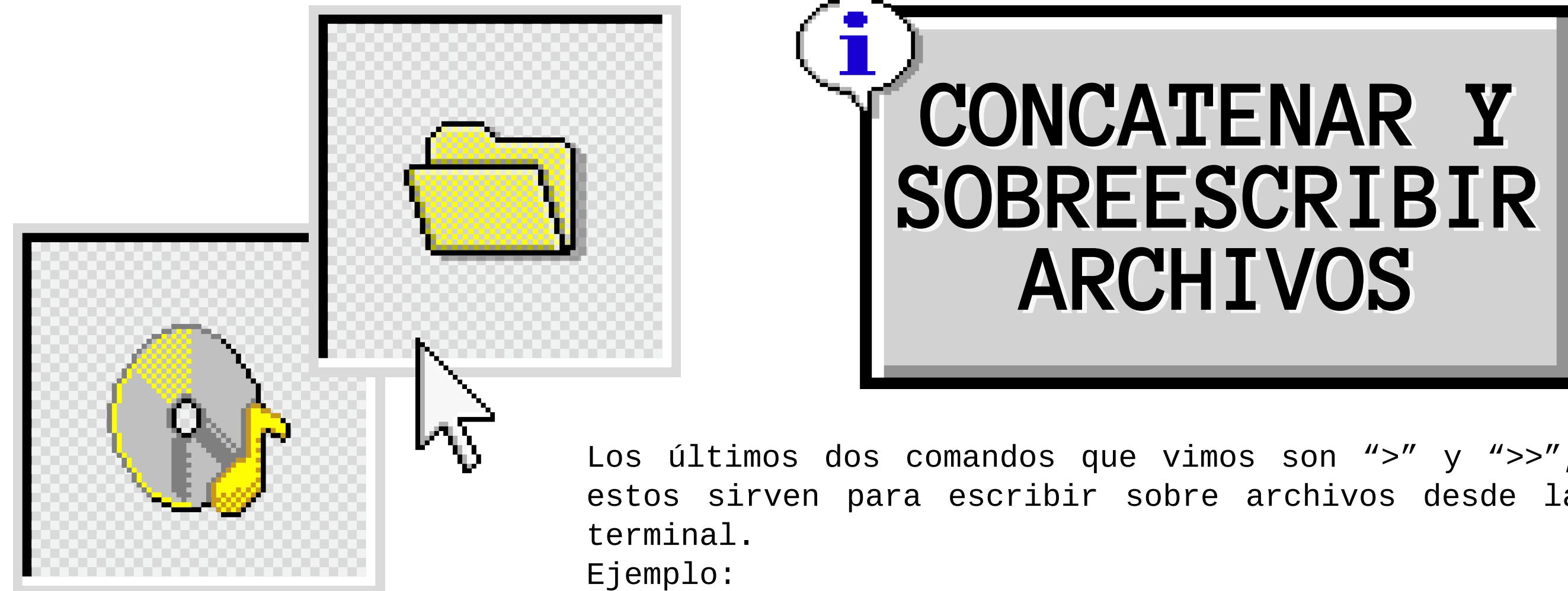


[Home](#)[Content](#)[Contact](#)

AHORA SÍ: ¿CÓMO LOS USAMOS?

- cat: imprimir los contenidos de un archivo en la terminal
- Syntax: cat "nombre_archivo"
- Trabaja en conjunto con ">" y ">>", vamos a ver de estos últimos...



[Home](#)[Content](#)[Contact](#)

Los últimos dos comandos que vimos son ">" y ">>", estos sirven para escribir sobre archivos desde la terminal.

Ejemplo:

```
cat constanciabarnald >> constanciaisaac (quedan ambas)
cat constanciabarnald > constanciaisaac (solo queda la de
barnald)
```



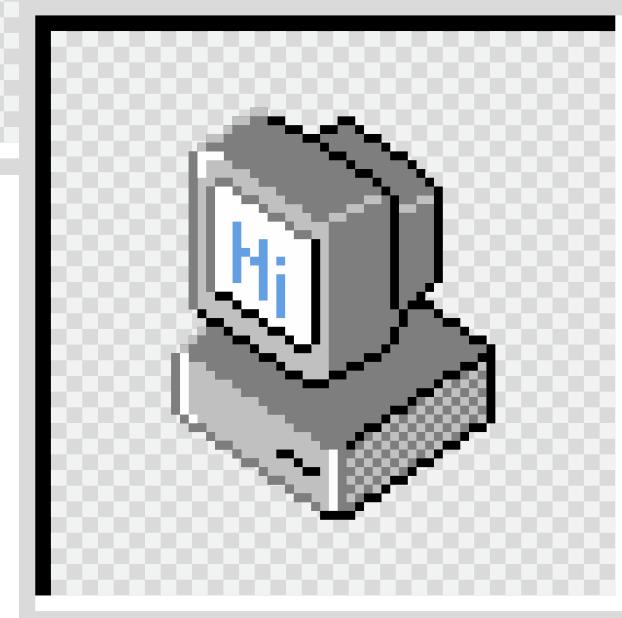
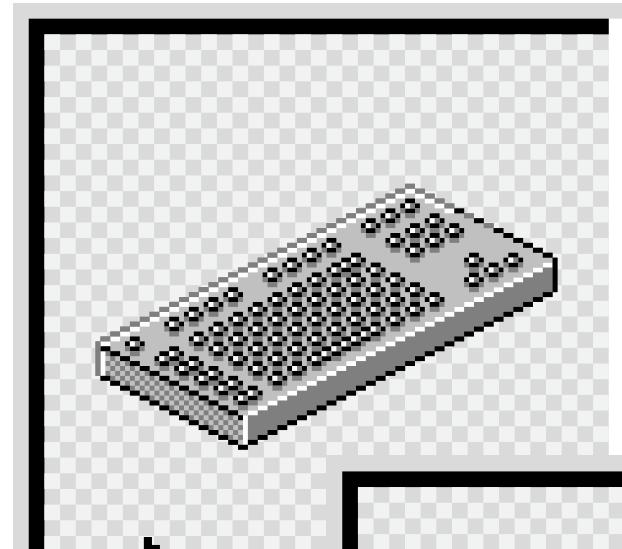
[Home](#)[Content](#)[Contact](#)

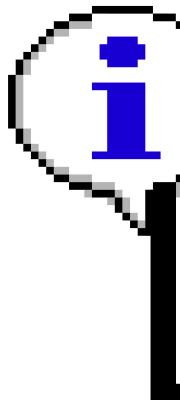
HEAD AND TAIL

- Head: Imprime en la terminal las primeras 10 líneas de un archivo
- Tail: Imprime en la terminal las últimas 10 líneas de un archivo

Variaciones: head/tail -n #

Donde # es cualquier número, cambia la cantidad de líneas que imprime



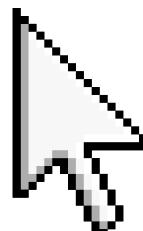
[Home](#)[Content](#)[Contact](#)

GREP

Una herramienta súper importante

Grep es el comando que permite hacer análisis de datos en bash! La primera clase se comentó que grep es uno de los comandos que tiene una cantidad increíble de variaciones, pero vamos a ver las principales.

grep: Comando que toma un patrón y lo compara con la entrada y solo imprime lo que lo cumple.



[Home](#)[**Content**](#)[Contact](#)

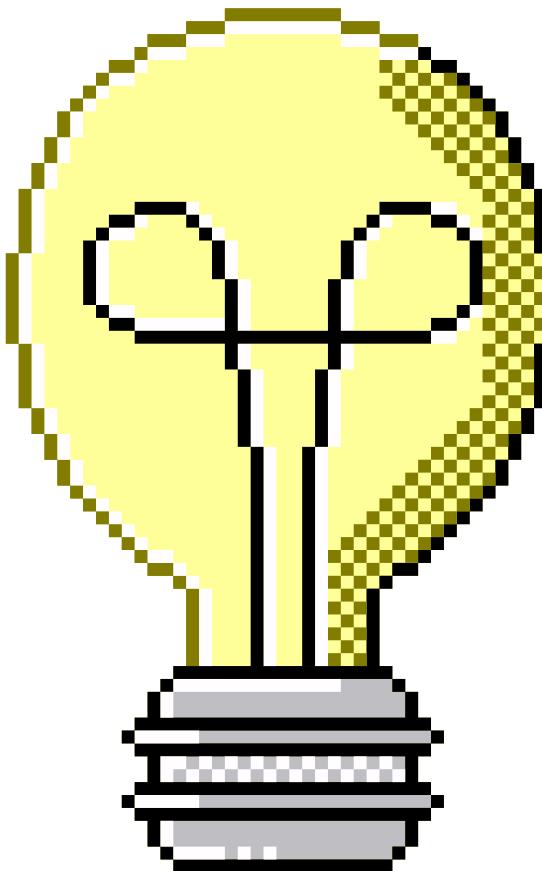
EJEMPLOS

nombre de archivo: animales.txt

Contenidos:

ballena y ave
caballo y burro
elefante
perro

```
(base) barnald-bocker@mangodevourer:~/silico$ grep f animales.txt
elefante
(base) barnald-bocker@mangodevourer:~/silico$ grep '' animales.txt
ballena y ave
caballo y burro
```

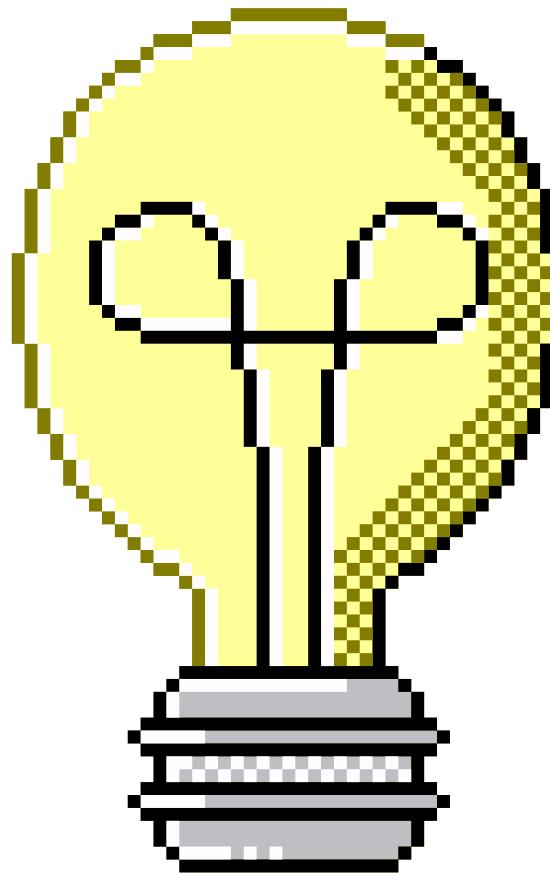


[Home](#)[Content](#)[Contact](#)

EJEMPLOS

```
(base) barnald-bocker@mangodevourer:~/silico$ grep -v ll animales.txt  
elefante  
perro  
(base) barnald-bocker@mangodevourer:~/silico$ grep ^e animales.txt  
elefante
```

```
(base) barnald-bocker@mangodevourer:~/silico$ grep ^$ animales.txt  
(base) barnald-bocker@mangodevourer:~/silico$ grep -v ^$ animales.txt  
ballena y ave  
caballo y burro  
elefante  
perro
```

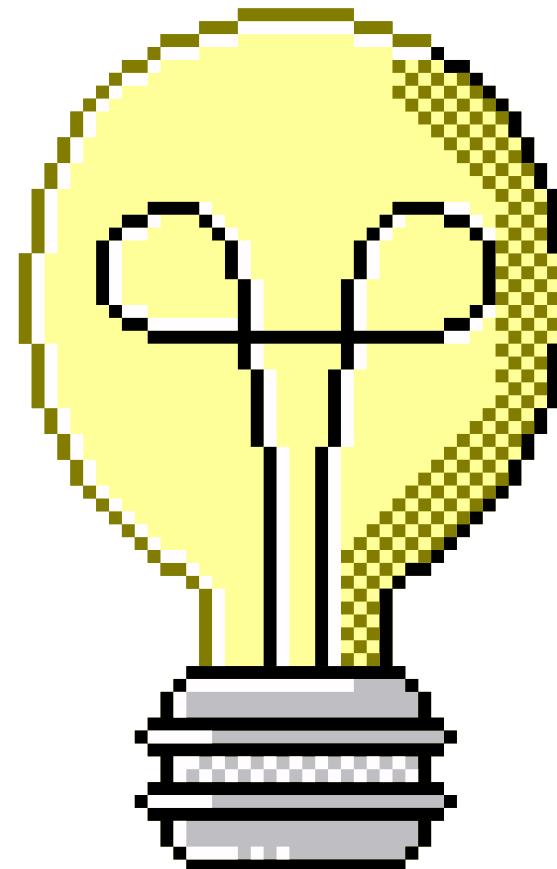


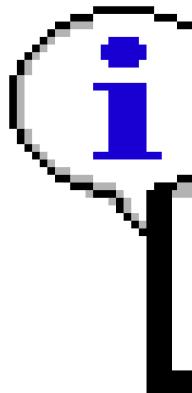


EJEMPLOS

```
(base) barnald-bocker@mangodevourer:~/silico$ grep -E 'll|pe' animales.txt
ballena y ave
caballo y burro
perro
(base) barnald-bocker@mangodevourer:~/silico$ grep -E 'll.*e' animales.txt
ballena y ave
```

```
(base) barnald-bocker@mangodevourer:~/silico/practica$ ls
animales.txt  nombres.txt
(base) barnald-bocker@mangodevourer:~/silico/practica$ grep -r "ll" ./txt
./animales.txt:ballena y ave
./animales.txt:caballo y burro
./nombres.txt:William
```





GREP

Una herramienta súper importante

`grep -v`: Invierte el output, encuentra todo lo que cumple el patrón y lo excluye

`grep ^`: Solo busca por el patrón al inicio de cada fila

`grep -r "patron"`: Busca el patrón en todos los archivos del directorio dado

`grep -E`: Permite usar caracteres para funciones muy específicas, dentro de `egrep` se encuentran las siguientes:

- “patronA|patronB”: Busca las líneas que cumplen patron A o B.
- “[Ii]saac”: Busca los que cumplen con “Isaac” o los que cumplen con “isaac”



Vamos a usar el archivo emails.txt para practicar las variaciones que aprendimos previamente, pero esta vez con el propósito de filtrar los datos u obtener secciones específicas



Objetivos:

- Encontrar todos los correos que terminan con .edu, con example.com, con gmail.
- Encontrar los correos que tienen de nombre "john", user, test
- Quitar los que no tienen "@" o "-com"

[Home](#)[Content](#)[Contact](#)

ECHO

Impresión en la terminal

echo es la forma de imprimir en la terminal. Es bastante simple en su uso, sin embargo puede ser usado en conjunto con **export**, este último comando es la forma de definir una variable en bash. Sin embargo, para acceder el valor de una variable se tiene que usar el símbolo de dólar.

```
(base) barnald-bocker@mangodevourer:~$ export x=10
(base) barnald-bocker@mangodevourer:~$ echo x
x
(base) barnald-bocker@mangodevourer:~$ echo $x
10
```



Home

Content

Contact

MUCHAS
GRACIAS

Barnald Bocker

Physics in Silico

Isaac Flores

