

[geekland < https://geekland.eu/>](https://geekland.eu/) — Blog de Tecnología



Uso del comando SED en Linux y UNIX con ejemplos

 [Joan Carles < https://geekland.eu/author/admin/>](https://geekland.eu/author/admin/)

 [9 agosto, 2021 < https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/>](https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/)

 [4 comentarios < https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comments>](https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comments)

[Home < https://geekland.eu/>](https://geekland.eu/) » [Linux < https://geekland.eu/category/linux-2/>](https://geekland.eu/category/linux-2/) » Explicación y ejemplos de uso de Sed

Sed es una herramienta de terminal cuyo uso principal es **buscar y reemplazar un texto**. En otras palabras más precisas permite buscar cadenas de texto, palabras o patrones de palabras y reemplazarlos por el texto que queramos. Aparte de lo que acabo de mencionar también permite insertar y eliminar texto de un documento. A continuación y en base a una serie de ejemplos

aprenderemos como podemos usar esta potente herramienta.

Nota: Sed es una herramienta imprescindible para todos aquellos usuarios que quieran programar scripts en Bash.

SINTAXIS BÁSICA DEL COMANDO SED

La sintaxis básica del comando SED es la que se muestra a continuación:

```
sed 's/texto_a_buscar/texto_a_reemplazar/' <fichero_a_reemplazar >fichero_nuevo
```

Una explicación de cada uno de los elementos que aparecen en la sintaxis básica es:

- `sed` : Parte del comando que llama a Sed
- `' '`: Dentro de las comillas verticales es donde introducimos el texto a buscar y el texto a reemplazar.
- `/ / /`: Las barras inclinadas también se conocen como delimitadores. Dentro del primer delimitador incluimos la cadena de texto a buscar y dentro del segundo la cadena de texto que reemplaza a la que estamos buscando.
- `'s/texto_a_buscar/texto_a_reemplazar/'`: `s` indica que queremos buscar y reemplazar. La parte del comando `texto_a_buscar` tiene que ser reemplazada una palabra, letra, cadena de caracteres o expresión regular que defina las partes de texto que queremos reemplazar. En `texto_a_reemplazar` tenemos que poner el texto que reemplazará el texto que queremos modificar.
- `<fichero_a_reemplazar`: Es nombre del fichero en que se buscan las partes de texto a modificar.
- `>fichero_nuevo` : Es un nuevo fichero que se generará con el texto reemplazado.

Sed admite el uso de tuberías. Por lo tanto otra forma de usar sed es la que se muestra a continuación.

```
echo "texto cualquiera" | sed 's/texto_a_buscar/texto_a_reemplazar/'
```

Una vez detallada la sintaxis básica pasaremos a ver una serie de ejemplos para entender mejor lo que acabamos de explicar.

BUSCAR Y REEMPLAZAR CADENAS DE TEXTO CON SED

El uso más habitual de Sed es buscar y reemplazar cadenas de texto. A continuación verán una serie de ejemplos que les ayudarán a dominar Sed.

Buscar la primera cadena de texto que contenga una letra o un patrón y reemplazarla por otra usando sed

Imaginemos que tenemos el siguiente texto en un fichero llamado `sedexamples`:

```
Para que LibreOffice tenga un modo 100% oscuro tendremos que modificar su esquema de colores. Básicamente tendremos que cambiar el color del texto y del área de escritura.
```

Para buscar la primera palabra que contiene la cadena de texto `o` y reemplazarla por la cadena de texto `O` debemos usar el siguiente comando:

```
sed 's/o/O/' <sedexamples >sedexamplesnew
```

Si ahora consultamos el texto que está dentro del fichero `sedexamplesnew` veremos que únicamente la primera letra `o` se ha convertido en una `O` mayúscula.

```
Para que Libreoffice tenga un modo 100% oscuro tendremos que modificar su esquema de colores. Básicamente tendremos que cambiar el color del texto y del área de escritura.
```

Si lo que pretenden es cambiar todas las `o` de minúscula a mayúscula deberán añadir la opción global `g`. En el siguiente apartado verán un ejemplo.

Buscar una cadena de texto que contenga una letra o un patrón y reemplazarla por otra en todo el documento

Si tenemos el siguiente texto en un fichero de texto con el nombre `sedexamples`.

```
Para que LibreOffice tenga un modo 100% oscuro tendremos que modificar  
su esquema de colores. Básicamente tendremos que cambiar el color del  
texto y del área de escritura.
```

Y queremos reemplazar todas las `o` minúsculas a `O` mayúsculas deberemos usar una sintaxis del siguiente tipo:

```
sed 's/texto_a_buscar/texto_a_reemplazar/g' <fichero_a_reemplazar >fichero_nuevo
```

Nota: La `g` hace referencia a global. (Sustitución global)

Por lo tanto si usamos el siguiente comando:

```
sed 's/o/O/g' <sedexamples >sedexamplesnew
```

Y consultamos el contenido del fichero `sedexamplesnew` obtendremos el siguiente resultado:

```
Para que LibreOffice tenga un mOdO 100% OscurO tendremOs que mOdificar  
su esquema de cOlOres. Básicamente tendremOs que cambiar el cOlOr del  
textO y del área de escritura.
```

Por lo tanto en el ejemplo que acabamos de ver han desaparecido las `o` minúsculas.

Uso del comando Sed mediante tuberías para buscar y reemplazar texto

Sed permite el uso de tuberías. Un ejemplo simple de como podemos usar Sed con tuberías es el siguiente.

Si queremos cambiar la primera **g** de la palabra **geekland** a mayúsculas mediante tuberías ejecutaremos el siguiente comando:

```
joan@gk55:~$ echo "geekland" | sed 's/g/G/'  
Geekland
```

Sobrescribir el contenido de un fichero de texto sin generar otro nuevo (opción **-i**)

En los ejemplos vistos con anterioridad generábamos un fichero nuevo a partir del viejo. Por lo tanto al finalizar la ejecución del comando teníamos 2 ficheros. El primero con el texto original y el segundo con el texto modificado.

Si en nuestro caso queremos sobrescribir el contenido directamente en el fichero original tendremos que usar la opción **-i**. Una muestra de la sintaxis a usar es la siguiente:

```
sed -i 's/texto_a_buscar/texto_a_reemplazar/g' fichero_a_reemplazar
```

A modo de ejemplo si queremos revertir las modificaciones realizadas en el fichero `sedexamplesnew`:

```
Para que LibreOffice tenga un mOdO 100% OscurO tendremOs que mOdificar  
su esquema de cOlOres. Básicamente tendremOs que cambiar el cOlOr del  
textO y del área de escritura.
```

Y de este modo transformar todas las **O** de mayúsculas a minúsculas ejecutaremos el siguiente comando:

```
sed -i 's/O/o/g' sedexamplesnew
```

Nota: La opción **-i** es para sobrescribir los cambios en el mismo fichero. La

opción `g` ya hemos visto anteriormente que es para que los cambios apliquen a todo el documento.

Una vez ejecutado el comando veremos que efectivamente se ha efectuado la sustitución:

```
joan@gk55:~$ cat sedexamplesnew
```

```
Para que Libreoffice tenga un modo 100% oscuro tendremos que modificar  
su esquema de colores. Básicamente tendremos que cambiar el color del  
texto y del área de escritura.
```

Restringir las líneas en que SED realizará sustituciones de texto

Sed permite buscar una línea que cumpla con un determinado patrón. Una vez encontrada la línea que cumple el patrón podemos hacer solo se realicen las modificaciones en esta línea en cuestión. Para realizar lo que acabo de citar hay que usar la siguiente sintaxis.

```
sed '/definir_patrón_a_buscar/s/texto_a_buscar_en_línea_que_aparece_patrón/texto
```

A modo de ejemplo si partimos del siguiente texto en el fichero `sedexamplesnew`:

```
Para que LibreOffice tenga un mOdO 100% OscurO tendremos que mOdificar  
su esquema de cOlOres. Básicamente tendremos que cambiar el cOlOr del  
textO y del área de escritura. Para ello prOcederemOs del siguiente  
mOdO.
```

```
Para que LibreOffice tenga un mOdO tOtalmente OscurO y cOnsistente  
tenemOs que realizar lo siguiente:
```

```
Usar un tema de escritOriO OscurO.
```

El fichero contiene 3 líneas. En las 2 primeras aparece la palabra LibreOffice y en la tercera no. Nuestro propóposito será cambiar las `o` de mayúscula a

minúscula solo en las líneas en que aparezca la palabra LibreOffice. Para ello ejecutaremos el siguiente comando:

```
sed -i '/LibreOffice/s/O/o/g' sedexamplesnew
```

Nota: Como hemos visto anteriormente la opción `-i` es para sobrescribir los cambios en el mismo fichero `sedexamplesnew`.

Una vez aplicado el comando el resultado obtenido es el siguiente:

```
Para que Libreoffice tenga un modo 100% oscuro tendremos que modificar  
su esquema de colores.Básicamente tendremos que cambiar el color del  
texto y del área de escritura. Para ello procederemos del siguiente  
modo.
```

```
Para que Libreoffice tenga un modo totalmente oscuro y consistente  
tenemos que realizar lo siguiente:
```

```
Usar un tema de escritOriO OscurO.
```

Por lo tanto la sustitución se ha realizado de forma perfecta en las líneas 1 y 2.

Reemplazar un texto por otro texto en las líneas que nosotros queramos con sed

Imaginemos que tenemos el siguiente texto en el fichero de texto `sedexamples`.

```
La camioneta es vieja  
La estufa es nueva  
La cesta de nueva  
La sudadera es nueva
```

Si queremos reemplazar la palabra `nueva` por `vieja` en las líneas 3 y 4 ejecutaremos el siguiente comando:

```
sed -i '3,4 s/nueva/vieja/g' sedexamples
```

Una vez ejecutado el comando el texto del fichero `sedexamples` será el siguiente:

```
joan@gk55:~$ cat sedexamples
La camioneta es vieja
La estufa es nueva
La cesta de vieja
La sudadera es vieja
```

Sustituciones múltiples en un solo comando (opción **-e**)

Podemos aplicar más de un sustitución de forma simultánea ejecutando un solo comando. Para ello imagines que la salida del comando `cat /etc/shells` es la siguiente.

```
joan@gk55:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
```

Según lo aprendido en los apartados anteriores, si queremos cambiar todas las cadenas de texto `usr` por `u` usaremos el siguiente comando:


```
joan@gk55:~$ cat /etc/shells | sed 's/usr/u/g'
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/u/bin/bash
/bin/rbash
/u/bin/rbash
/bin/dash
/u/bin/dash
```

Si ahora pretendemos encadenar múltiples sustituciones usando únicamente un solo comando tendremos que usar la opción `sed -e`. Para sustituir todas las cadenas de texto de `usr` a `u` y de `bin` a `b` lo haremos del siguiente modo:

```
joan@gk55:~$ cat /etc/shells | sed -e 's/usr/u/g' -e 's/bin/b/g'
# /etc/shells: valid login shells
/b/sh
/b/bash
/u/b/bash
/b/rbash
/u/b/rbash
/b/dash
/u/b/dash
```

Nota: Otra opción alternativa para conseguir el mismo resultado sería `cat /etc/shells | sed 's/usr/u/g' | sed 's/bin/b/g'`

Usar y reemplazar texto con SED cuando los patrones que estamos buscando contienen el carácter `/`

Para el problema mencionado en el título de este apartado existen 2 soluciones.

- La primera es usar el carácter `\` antes de del carácter `/` que queremos reemplazar.

- La segunda forma es reemplazar las típicas barras delimitadoras `/` del comando sed por caracteres alternativos como por ejemplo `!`, `?`, `#`, etc.

Ahora imaginemos que queremos reemplazar `/usr` por `(u` en la salida del comando `cat /etc/shells`. Si usamos la primera de las soluciones lo haremos del siguiente modo:

```
joan@gk55:~$ cat /etc/shells | sed 's/\usr/(u/g'
# /etc/shells: valid login shells
/bin/sh
/bin/bash
(u/bin/bash
/bin/rbash
(u/bin/rbash
/bin/dash
(u/bin/dash
```

Y en caso que queramos usar la segunda solución lo haremos del siguiente modo:

```
joan@gk55:~$ cat /etc/shells | sed 's#/usr#(u#g'
# /etc/shells: valid login shells
/bin/sh
/bin/bash
(u/bin/bash
/bin/rbash
(u/bin/rbash
/bin/dash
(u/bin/dash
```

Reemplazar caracteres individuales con Sed

El uso más habitual de Sed es buscar y reemplazar cadenas de caracteres. Pero también permite reemplazar caracteres individuales. Para ello deberán usar la

siguiente sintaxis:

```
sed -e 'y/caracteres_buscar/caracteres_a_reemplazar/'
```

Ahora supongamos que tenemos el siguiente texto:

```
"Esto es un ejemplo de reemplazo de caracteres para el blog geekland"
```

Si queremos reemplazar todas las **a** por **A** y todas las **o** por **O** deberemos ejecutar el siguiente comando:

```
joan@gk55:~$ echo "Esto es un ejemplo de reemplazo de caracteres para  
el blog geekland" | sed -e 'y/ao/AO/'  
EstO es un ejempLO de reemplAzO de cArActeres pArA el blOg geeklAnd
```

Encontrar una línea que empiece por una determinada palabra y cambiar el contenido de toda la línea

En el siguiente ejemplo partimos del fichero **sedexamples** que contiene el siguiente texto:

```
Geekland es una blog.  
que habla mayoritariamente de software libre.  
que además habla de informática.
```

Si queremos reemplazar todas las líneas que empiezan por **que** por el texto **el contenido de la línea ha sido reemplazado** haremos uso del siguiente comando:

```
sed -i 's/^que .*/el contenido de la línea ha sido reemplazado/'  
sedexamples
```

Nota: La magia del último comando la realiza la expresión regular **^que .***. La parte **^que** hace referencia a todas las líneas que empiezan por la cadena de caracteres **que**. El punto **.** hace referencia a cualquier letra que aparezca las

veces que aparezca `*` hasta el final de la línea\$.

El resultado después de ejecutar el comando será el que se muestra en pantalla.

```
joan@gk55:~$ sed -i 's/^que .*/el contenido de la línea ha sido
reemplazado/' sedexamples
joan@gk55:~$ cat sedexamples
Geekland es una blog.
el contenido de la línea ha sido reemplazado
el contenido de la línea ha sido reemplazado
```

Cambiar todas las mayúsculas a minúsculas y viceversa

Si queremos cambiar un texto de mayúsculas a minúsculas también deberemos usar las expresiones regulares. En este caso la expresión regular que necesitamos definir es la siguiente:

`[a-z]`: Expresión regular que hace referencia a todas las letras entre la `a` y la `z`.

Por lo tanto si el fichero `sedexamplesnew` contiene el siguiente texto

```
HACER QUE LIBREOFFICE TENGA UN MODO OSCURO AL 100%
Usar un tema de escritOriO Oscuro.
Usar un tema de icOnOs adecuado para el mOdO OscurO.
MODificar el cOlOr del área de escritura y cambiar el cOlOr del textO.
```

Y lo queremos transformar a mayúsculas usaremos el siguiente comando:

```
sed -i 's/[a-z]/\U&/g' sedexamplesnew
```

Nota: `\U&` hace referencia a mayúsculas (Upper case). Por lo tanto el comando aplicado transforma a mayúsculas cualquier letra de la `a` a la `z` que haya en el documento `sedexamplesnew`

Una vez ejecutado el comando el resultado es el siguiente:

```
HACER QUE LIBREOFFICE TENGA UN MODO OSCURO AL 100%  
USAR UN TEMA DE ESCRITORIO OSCURO.  
USAR UN TEMA DE ICONOS ADECUADO PARA EL MODO OSCURO.  
MODIFICAR EL COLOR DEL ÁREA DE ESCRITURA Y CAMBIAR EL COLOR DEL  
TEXTO.
```

Si ahora quisiéramos volver a transformar el texto a minúscula tendríamos que usar el siguiente comando:

```
sed -i 's/[A-Z]/\L&/g' sedexamplesnew
```

INSERTAR TEXTO O MODIFICAR UN TEXTO MEDIANTE EL USO DE SED

El comando sed también permite insertar texto en un documento. Algunos ejemplos de como Sed permite insertar texto en un documentos son los siguientes.

Insertar texto al inicio de todas las líneas de un fichero de texto

En el caso que el fichero `sedexamples` tenga el siguiente texto:

```
La camioneta es vieja  
La estufa es nueva  
La cesta de vieja  
La sudadera es vieja
```

Si al inicio de cada una de las líneas queremos añadir un guion y un espacio ejecutaremos el siguiente comando:

```
sed -i 's/^/- /' sedexamples
```

Nota: `^` es la expresión regular que indica al inicio de la línea.

El resultado obtenido de aplicar el comando es el siguiente:

```
joan@gk55:~$ cat sedexamples
- La camioneta es vieja
- La estufa es nueva
- La cesta de vieja
- La sudadera es vieja
```

Insertar texto al inicio de una línea o en un rango de líneas con sed

Siguiendo con el ejemplo del apartado anterior. Si solo hubiéramos querido añadir `-` a las líneas 2, 3 y 4 tendríamos que haber ejecutado el siguiente comando:

```
sed -i '2,4 s/^/- /' sedexamples
```

Obteniendo el siguiente resultado:

```
joan@gk55:~$ cat sedexamples
La camioneta es vieja
- La estufa es nueva
- La cesta de vieja
- La sudadera es vieja
```

Si ahora quisiéramos añadir el `-` únicamente a la primera de las líneas ejecutaríamos el siguiente comando:

```
sed -i '1 s/^/- /' sedexamples
```

Insertar texto al final de una línea

Podemos añadir texto al final de todas y cada una de las líneas. Para ello tan

solo deberemos usar la expresión regular apropiada. Si en el fichero `sedexamples` tenemos el siguiente texto:

```
- La estufa es nueva  
- La cesta de vieja  
- La sudadera es vieja
```

Para insertar el texto `. cierro la cita` al final de todas y cada una de las líneas debería ejecutar el siguiente comando:

```
sed -i 's/$/. cierro la cita/' sedexamples
```

Nota: `$` es la expresión regular que indica al final de la línea.

El resultado obtenido sería el siguiente:

```
joan@gk55:~$ cat sedexamples  
- La estufa es nueva. cierro la cita  
- La cesta de vieja. cierro la cita  
- La sudadera es vieja. cierro la cita
```

Si ahora quiero añadir el texto `. vuelvo a cerrar la cita` únicamente al final de la línea 1 ejecutaría el siguiente comando:

```
sed -i '1 s/$/. vuelvo a cerrar la cita/' sedexamples
```

Una vez ejecutado el comando el resultado obtenido sería el siguiente:

```
joan@gk55:~$ cat sedexamples  
- La estufa es nueva. cierro la cita. vuelvo a cerrar la cita  
- La cesta de vieja. cierro la cita  
- La sudadera es vieja. cierro la cita
```

En el caso que también quisiéramos añadir el texto `. vuelvo a cerrar la cita` a las líneas 2 y 3 ejecutaríamos el siguiente comando:

```
sed -i '2,3 s/$/. vuelo a cerrar la cita/' sedexamples
```

Obteniendo el siguiente resultado:

```
joan@gk55:~$ cat sedexamples
- La estufa es nueva. cierro la cita. vuelo a cerrar la cita
- La cesta de vieja. cierro la cita. vuelo a cerrar la cita
- La sudadera es vieja. cierro la cita. vuelo a cerrar la cita
```

Añadir una línea en blanco después de encontrar una palabra o expresión regular

Si después de encontrar una determinada palabra en un texto queremos añadir una línea en blanco justo por debajo de la línea donde aparece la palabra deberemos ejecutar un comando del siguiente tipo:

```
sed -i '/palabra_o_patron_a_buscar/G' nombre_fichero
```

Por lo tanto si partimos del fichero `sedexamples` con el mismo contenido que teníamos en el apartado anterior y queremos añadir una línea blanco justo por debajo de la línea en que aparece la palabra `cesta` lo haremos del siguiente modo:

```
joan@gk55:~$ sed -i '/cesta/G' sedexamples
joan@gk55:~$ cat sedexamples
- La estufa es nueva. cierro la cita. vuelo a cerrar la cita
- La cesta de vieja. cierro la cita. vuelo a cerrar la cita

- La sudadera es vieja. cierro la cita. vuelo a cerrar la cita
```

Insertar una línea en blanco cada 3 líneas

Si disponemos del fichero de texto `sedexamplesnew` con el siguiente contenido:


```
1  
2  
3  
4  
5  
6  
7
```

Y queremos añadir una línea en blanco cada 3 líneas ejecutaremos el siguiente comando:

```
sed -i 'n;n;G;' sedexamplesnew
```

Después de ejecutar el comando obtendremos el siguiente resultado:

```
joan@gk55:~$ cat sedexamplesnew  
1  
2  
3  
  
4  
5  
6  
  
7
```

Insertar o añadir una línea en blanco cuando se cumpla un patrón o condición

Si únicamente queremos añadir la línea en blanco cuando se cumpla una determinada condición lo haremos usando Sed del siguiente modo:

```
sed -i '/cadena/{x;p;x;}' archivo
```

Por lo tanto si en el fichero `sedexamplesnew` queremos **añadir una línea en blanco antes de la línea en la que aparece 5** ejecutaremos el siguiente comando y obtendremos el siguiente resultado:

```
joan@gk55:~$ sed -i '/5/{x;p;x;}' sedexamplesnew
joan@gk55:~$ cat sedexamplesnew

1
2
3

4

5
6

7
```

Si ahora quisiéramos **añadir una línea en blanco después de la línea en que aparece 5** lo haríamos del siguiente modo obteniendo el siguiente resultado:

```
joan@gk55:~$ sed -i '/5/G' sedexamplesnew
joan@gk55:~$ cat sedexamplesnew

1
2
3

4

5

6

7
```

Si queremos **añadir una línea en blanco justo antes y después de la línea en que aparece 2** lo haremos del siguiente modo:

```
joan@gk55:~$ sed -i '/2/{x;p;x;G;}' sedexamplesnew
```

```
joan@gk55:~$ cat sedexamplesnew
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

Añadir un número de línea al inicio de cada una de las líneas de un fichero

Sed permite numerar cada una de las líneas de un fichero. Para ello usaremos una sintaxis del siguiente tipo:

```
sed = nombre_fichero | sed 'N;s/\n/\t/'
```

Para entender bien este comando lo haremos con un ejemplo. Imaginemos que el fichero `sedexamples` tiene el siguiente contenido:

```
Esto es un ejemplo para numerar las líneas de un fichero con sed.
```

```
Esta es una de las líneas del contenido.  
Y esta es la última línea del fichero.
```

Para insertar los números de línea y saltar de línea Sed usa el símbolo =. Por lo tanto, para insertar un numero de línea y saltar de línea lo haremos del siguiente modo:

```
joan@gk55:~$ sed = sedexamples  
1  
Esto es un ejemplo para numerar las líneas de un fichero con sed.  
2  
  
3  
Esta es una de las líneas del contenido.  
4  
Y esta es la última línea del fichero.
```

Nota: Otro comando equivalente al que acabamos de ejecutar sería `sed -n '=:p' sedexamples`

Si ahora queremos juntar el número de línea con el contenido del fichero tenemos que juntar 2 líneas y añadir una tabulación entre el número y el contenido del fichero. Para ello ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed = sedexamplesnew | sed 'N;s/\n/\t/'  
1      Esto es un ejemplo para numerar las líneas de un fichero con sed.  
2  
3      Esta es una de las líneas del contenido.  
4      Y esta es la última línea del fichero.
```

Nota: La opción N añade la siguiente línea a la línea actual. La parte del comando `s/\n/\t/` sustituye todo salto de línea `\n` por una tabulación `\t`. De esta forma hemos conseguido nuestro objetivo.

Si en vez de una tabulación quisieran añadir un guión y un espacio deberían

modificar el comando del siguiente modo.

```
joan@gk55:~$ sed = sedexamplesnew | sed 'N; s/\n/- /'
```

1- Esto es un ejemplo para numerar las líneas de un fichero con sed.

2-

3- Esta es una de las líneas del contenido.

4- Y esta es la última línea del fichero.

Si ahora quisiéramos volcar el resultado en un fichero de texto con el nombre `textofinal` lo haríamos del siguiente modo:

```
joan@gk55:~$ sed = sedexamplesnew | sed 'N;s/\n/- /' > textofinal
joan@gk55:~$ cat textofinal
```

1- Esto es un ejemplo para numerar las líneas de un fichero con sed.

2-

3- Esta es una de las líneas del contenido.

4- Y esta es la última línea del fichero.

Insertar un número de línea al inicio de cada línea únicamente cuando tenga contenido

En el ejemplo anterior hemos visto como mostrar la numeración de todas las líneas. La salida mostraba la numeración de las líneas con contenido y sin contenido. Si queremos suprimir la numeración de las líneas sin contenido lo haremos del siguiente modo:

```
joan@gk55:~$ sed '/./=' sedexamplesnew | sed '/./N; s/\n/- /'
```

1- Esto es un ejemplo para numerar las líneas de un fichero con sed.

3- Esta es una de las líneas del contenido.

4- Y esta es la última línea del fichero

Nota: `/./` sirve como filtro para despreciar las líneas sin contenido del fichero `sedexamplesnew`.

Para numerar y mostrar únicamente las líneas que no tienen contenido se podría hacer con el siguiente comando:

```
joan@gk55:~$ sed '/^$/=' sedexamplesnew | sed '/^[0-9]/N; s/\n/- /' | sed '/[a-Z 2-
```

De este modo sabríamos que el fichero `sedexamplesnew` solo tiene una línea vacía y es la 2.

Juntar todas las líneas de un fichero mediante Sed

Sed también permite juntar todas las línea de un texto en una sola línea de forma extremadamente sencilla. Para ello ejecutaremos el siguiente comando:

```
joan@gk55:~$ cat sedexamplesnew | sed ':a; N; s/\n/ /; ta'
```

Después de ejecutar el comando obtendremos el siguiente resultado:

```
Esto es un ejemplo para numerar las líneas de un fichero con sed. Esta
es una de las líneas del contenido. Y esta es la última línea del
fichero.
```

Nota: La estructura del comando es `sed ':etiqueta; N; s/buscar/reemplazar/; t+etiqueta'`

Nota: El comando aplicado en este ejemplo funciona del siguiente modo. Inicialmente se introduce la etiqueta `a` a la primera línea. A continuación gracias al parámetro `N` se cogen las 2 primeras líneas del fichero. Seguidamente, en las 2 primeras líneas se busca un salto de línea `\n` y cuando lo encuentra lo elimina y reemplaza por un espacio. De este modo las 2 primeras líneas ya se han transformado en una. Al cumplirse y ejecutarse el patrón de búsqueda `s/\n/ /` gracias al parámetro `ta` volvemos a la primera que es la que tiene la etiqueta `a`. La primera línea ahora contiene las líneas que antes eran la 1 y la 2. Una vez estamos en la primera línea se vuelve a repetir el proceso. Es decir la línea 1 se juntará con la línea 2 y de este modo la línea 1 ya contendrá las líneas 1,2 y 3.

Juntar únicamente las líneas que finalicen con el símbolo \

Si queremos podemos seleccionar las líneas que queremos unir mediante un marcador. Si usamos el marcador \ podemos tener un texto similar al siguiente en el fichero `sedexamplesnew`:

```
Esto es un ejemplo para numerar las líneas de un fichero con sed.
```

```
Esta es una de las líneas del contenido.\
```

```
Y esta es la última línea del fichero.
```

La única línea que tiene el marcador \ es la 3. Por lo tanto si queremos juntar la línea 3 con la 4 podemos usar el siguiente comando:

```
joan@gk55:~$ cat sedexamplesnew | sed ':a; N; s/\\n/ /; ta'
```

Y el resultado obtenido será el siguiente:

```
Esto es un ejemplo para numerar las líneas de un fichero con sed.
```

```
Esta es una de las líneas del contenido. Y esta es la última línea
```

Añadir una línea con contenido o vacía justo al final de un fichero de texto

La salida del comando `echo -e "1\n2\n3"` es:

```
joan@gk55:~$ echo -e "1\n2\n3"
```

```
1
```

```
2
```

```
3
```

Si queremos añadir un contenido cualquiera en la última línea lo haremos del

siguiente modo:

```
joan@gk55:~$ echo -e "1\n2\n3" | sed '$a texto nuevo introducido'
1
2
3
texto nuevo introducido
```

Si el contenido inicial lo tiene el fichero `sedgeekland` y queremos crear un nuevo fichero con nombre `test2` que tenga el contenido del fichero `sedgeekland` más la frase `texto nuevo introducido` justo en la última línea lo haremos del siguiente modo:

```
joan@gk55:~$ sed '$a texto nuevo introducido' sedgeekland > test2
joan@gk55:~$ cat test2
1
2
3
texto nuevo introducido
```

Añadir una línea con contenido o vacía en la penúltima línea de un fichero texto

Si en vez de insertar el texto en la última línea lo queréis insertar en la penúltima lo tenéis que hacer del mismo modo que en el apartado anterior. La única diferencia es que tendréis que reemplazar el parámetro `$a` por `$i`. Un ejemplo de lo que acabo de comentar es el siguiente:

```
joan@gk55:~$ echo -e "1\n2\n3" | sed '$i texto nuevo introducido'
1
2
texto nuevo introducido
3
```


SELECCIONAR Y VISUALIZAR CONTENIDO MEDIANTE SED

Sed permite capturar y visualizar el texto que nos interesa. De este modo, Sed nos permitirá realizar las siguientes operaciones.

Mostrar las x primeras líneas de un fichero de texto

Para imprimir las 2 primeras líneas del fichero `sedexamplesnew` tan solo deberemos usar el siguiente comando:

```
joan@gk55:~$ sed 2q sedexamplesnew
Estamos viendo el uso básico de sed.
Sed es una herramienta potente
```

Visualizar una línea o un rango de líneas con Sed

Imaginemos que el fichero `sedexamplesnew` tiene el siguiente contenido:

```
joan@gk55:~$ cat sedexamplesnew
Estamos viendo el uso básico de sed.
Sed es una herramienta potente
que buscar y reemplaza texto
final
```

Sí únicamente queremos imprimir la tercera línea del fichero usaremos el siguiente comando:

```
joan@gk55:~$ sed -n '3p' sedexamplesnew
que buscar y reemplaza texto
```

Si ahora queremos imprimir la línea 1 y la línea 3 ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed -n '1p' sedexamplesnew && sed -n '3p' sedexamplesnew
Estamos viendo el uso básico de sed.
que buscar y reemplaza texto
```

Nota: Un comando equivalente al que se muestra en el ejemplo sería `sed -n -e '1p' sedexamplesnew -e '3p' sedexamplesnew`

En el caso que queramos imprimir el rango de líneas de la 2 a la 4 ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed -n '2,4p' sedexamplesnew
Sed es una herramienta potente
que buscar y reemplaza texto
final
```

Finalmente, para mostrar en pantalla todo el contenido del fichero `sedexamplesnew` mediante sed ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed -n 'p' sedexamplesnew
Estamos viendo el uso básico de sed.
Sed es una herramienta potente
que buscar y reemplaza texto
final
```

Mostrar la primera y última línea de un fichero de texto

Para mostrar la primera línea de un fichero de texto como por ejemplo `sedexamplesnew` ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed -n '1p' sedexamplesnew
Estamos viendo el uso básico de sed.
```

Y en caso de querer mostrar la última línea reemplazaremos el número `1` por el símbolo `$` que significa final. En este caso el resultado obtenido es el siguiente:

```
joan@gk55:~$ sed -n '$p' sedexamplesnew  
final
```

Mostrar las últimas X líneas de un fichero

Para mostrar las últimas X líneas de un fichero usaremos un comando del siguiente tipo:

```
cat nombre_fichero | sed ':a; N; 1,Xba; D'
```

Nota: nombre_fichero se debe reemplazar por el nombre del fichero que tenga el contenido

Nota: X tiene que ser reemplazado por el número de líneas que queramos visualizar al final de fichero.

Por lo tanto si el fichero sedexamples tiene el siguiente contenido:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

para mostrar las 3 últimas líneas del fichero sedexamples procederemos del siguiente modo:

```
joan@gk55:~$ cat sedexamples | sed ':a; N; 1,3ba; D'
8
9
10
```

Nota: El resultado de este comando es el mismo que el comando `tail -n 3 sedexamples`

Lo que hace el comando que acabamos de ejecutar es:

- Inicialmente se leen los números 1,2,3. Esto es gracias al parámetro 1,3ba
- Al usar el parámetro N se añadirán las líneas 4,5 y 6 al espacio de trabajo. Por lo tanto ahora tendremos 1,2,3,4,5,6 en el espacio de trabajo.
- El parámetro D borrará las líneas viejas del espacio de trabajo. Por lo tanto borrará las líneas 1,2,3. Ahora en el espacio de trabajo tendremos las líneas 4,5,6.
- A las líneas 4,5,6 añadiremos 7,8,9 y el parámetro D eliminará las viejas que son 4,5,6. Así sucesivamente hasta llegar al final

De este modo obtendremos las últimas X filas de un fichero de texto con sed.

Mostrar el texto que está entre dos etiquetas o dos marcadores

En ocasiones puede ser útil extraer el texto comprendido entre 2 etiquetas o 2 marcadores. Imaginemos el caso que el fichero `sedexamples` tenga el siguiente contenido:

```
<title>Geekland Blog de tecnología</title>
<plot>Es un blog que habla de software Libre

Y también de informática</plot>
```

Si queremos extraer la totalidad de texto comprendido entre las etiquetas `title` ejecutaremos el siguiente comando.

```
joan@gk55:~$ cat sedexamples | sed -n 's:.*<title>\(.*\)</title>.*:\1:p'
```

Geekland Blog de tecnología

El comando que acabamos de ejecutar **funcionará en todas las etiquetas que se abren y cierran en la misma línea. Si la etiqueta se abre y cierra en distintas líneas**, como por ejemplo la etiqueta `plot` tendremos que ejecutar el siguiente comando:

```
joan@gk55:~$ cat sedexamples | sed -n '/plot/{s/.*<plot>///;s/<\/plot.*///;p;}'
```

Es un blog que habla de software Libre
Y también de informática

Nota: Este último ejemplo funciona independientemente de si las etiquetas se abren y cierran en la misma línea.

Imprimir las líneas que cumplen con una determinada expresión regular mediante la (opción `-n` y `p`)

Sed permite imprimir las líneas que cumplan con un determinado patrón y mostrarlas en pantalla o ubicarlas en un fichero de texto.

Para ello deberemos usar la siguiente sintaxis:

```
sed -n '/texto_a_imprimir/p'
```

Nota: La opción `-n` conjuntamente con la opción `p` hace que solo se muestren en pantalla las cadenas de texto que queremos visualizar.

A modo de ejemplo el comando `cat /etc/shells` nos da la siguiente salida:

```
joan@gk55:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
```

Si únicamente queremos mostrar las líneas de la salida del comando que contengan la palabra `usr` usaremos el siguiente comando:

```
joan@gk55:~$ cat /etc/shells | sed -n '/usr/p'
/usr/bin/bashd
/usr/bin/rbash
/usr/bin/dash
```

Si quisiéramos mostrar todas las líneas que no contienen la cadena `usr` deberíamos usar el siguiente comando:

```
joan@gk55:~$ cat /etc/shells | sed -n '/usr/!p'
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/bin/rbash
/bin/dash
```

Mostrar en pantalla únicamente las líneas en que se ha reemplazado texto

Imaginemos el hipotético caso en que tenemos el siguiente texto:

```
joan@gk55:~$ echo -e "Quiero reemplazar\nLa palabra vieja por nueva\nY la palabra nueva por vieja"
Quiero reemplazar
La palabra vieja por nueva
Y la palabra nueva por vieja
```

Si queremos reemplazar la palabra `vieja` por `nueva` en todo el documento ejecutaremos el siguiente comando:

```
joan@gk55:~$ echo -e "Quiero reemplazar\nLa palabra vieja por nueva\nY la palabra nueva por vieja" | sed 's/vieja/nueva/g'
Quiero reemplazar
La palabra nueva por nueva
Y la palabra nueva por nueva
```

Vemos que hemos reemplazado correctamente el texto y se muestra la totalidad del texto inicial con las palabras modificadas. Si ahora queremos realizar la misma operación que acabamos de realizar mostrando únicamente las líneas en que se ha realizado la sustitución usaremos la opciones `-n` y `p` del siguiente modo:

```
joan@gk55:~$ echo -e "Quiero reemplazar\nLa palabra vieja por nueva\nY la palabra nueva por vieja" | sed -n 's/vieja/nueva/p'
La palabra nueva por nueva
Y la palabra nueva por nueva
```

Mostrar las líneas según el número de caracteres que tengan

Ahora partimos del caso que el fichero `seedgeekland` tiene el siguiente contenido:

```
La primera línea tiene 37 caracteres
La segunda 14
```

Para imprimir las líneas que tienen 30 caracteres o más ejecutaremos el

siguiente comando:

```
joan@gk55:~$ sed -n '/^.\{30\}/p' sedgeekland
La primera línea tiene 37 caracteres
```

Si por lo contrario queremos imprimir las líneas que tienen un número de caracteres igual o menor a 30 lo haremos del siguiente modo:

```
joan@gk55:~$ sed -n '/^.\{30\}/!p' sedgeekland
La segunda 14
```

BORRAR O ELIMINAR TEXTO CON SED

Si pretendéis eliminar líneas de texto con Sed podéis hacerlo de la forma que se muestra a continuación.

Borrar tan solo una línea de un fichero con sed

A modo de ejemplo tenemos un fichero de texto con el nombre `sedgeekland` con el siguiente contenido:

```
Esta línea es la 1
Esta es la 2
La 3
La 4
La 5
La 6
La 7
```

Si queremos borrar la línea 7 y obtener el resultado en un nuevo fichero de texto con el nombre `sedgeeklandnew` ejecutaremos el siguiente comando:

```
sed '7 d' sedgeekland > sedgeeklandnew
```

una vez ejecutado el comando el contenido del fichero `sedgeeklandnew` será el

siguiente:

```
joan@gk55:~$ cat sedgeeklandnew
Esta línea es la 1
Esta es la 2
La 3
La 4
La 5
La 6
```

Borrar un rango de líneas en un texto

Si lo que pretendemos es borrar un rango de líneas, como por ejemplo de la línea 4 a la línea 6, tan solo hay que ejecutar el siguiente comando:

```
sed '4,6 d' sedgeekland > sedgeeklandnew
```

De este modo el contenido del fichero `sedgeeklandnew` quedará con el siguiente contenido:

```
joan@gk55:~$ cat sedgeeklandnew
Esta línea es la 1
Esta es la 2
La 3
La 7
```

Si quisiéramos borrar todas las líneas del fichero `sedgeekland` excepto las líneas 4, 5 y 6 tendríamos que usar el siguiente comando:

```
sed '4,6 !d' sedgeekland > sedgeeklandnew
```

Borrar desde una determinada línea hasta el final con sed

Si nuestro objetivo es borrar a partir de una determinada línea, que en nuestro

caso es la 3, hasta el final del documento tendremos que usar el siguiente comando:

```
sed '3,$d' sedgeekland > sedgeeklandnew
```

Por lo tanto si partimos del fichero `sedgeekland`, en el fichero `sedgeeklandnew` quedará del siguiente modo:

```
joan@gk55:~$ cat sedgeeklandnew
Esta línea es la 1
Esta es la 2
```

Borrar la última línea de un documento

Si nuestro fichero `sedgeekland` tiene el siguiente contenido:

```
Esta línea es la 1
Esta es la 2
La 3
La 4
La 5
La 6
La 7
```

y únicamente queremos borrar la última línea ejecutaremos el siguiente comando:

```
joan@gk55:~$ sed '$d' sedgeekland > sedgeeklandnew
joan@gk55:~$ cat sedgeeklandnew
Esta línea es la 1
Esta es la 2
La 3
La 4
La 5
La 6
```

Borrar todas las líneas que contengan una determinada cadena de caracteres

Si tenemos un fichero con el nombre `sedexamplesnew` con el siguiente contenido:

```
Para que Libreoffice tenga un modo 100% oscuro tendremos que modificar  
su esquema de colores. Básicamente tendremos que cambiar el color del  
texto y del área de escritura. Para ello procederemos del siguiente  
modo.
```

```
Para que Libreoffice tenga un modo totalmente oscuro y consistente  
tenemos que realizar lo siguiente:
```

```
Usar un tema de escritOriO Oscuro0.
```

y queremos borrar todas las líneas que contengan la cadena de texto `Libreoffice` ejecutaremos el siguiente comando:

```
sed -i '/Libreoffice/d' sedexamplesnew
```

Nota: El comando ejecutado similar a los anteriores. La única diferencia es que en vez de indicar las palabras a sustituir escribimos `d` de delete/borrar.

Y el resultado obtenido será el siguiente:

```
joan@gk55:~$ cat sedexamplesnew  
Usar un tema de escritOriO Oscuro0.
```

Para borrar todas las líneas que no tengan la palabra `Libreoffice` deberíamos ejecutar el siguiente comando:

```
sed -i '/Libreoffice/!d' sedexamplesnew
```

Borrar todos los espacios vacios al final de una línea

Si lo que pretendemos es borrar todos los espacios vacíos al final de la línea deberemos usar un comando del siguiente tipo:

```
sed -i 's/ *$//' nombre_fichero
```

Donde:

***\$:** Es una expresión regular que hace referencia a buscar todos los espacios que hay al final de una línea.

Nota: Si se fijan el segundo delimitador está vacío porque el objetivo es reemplazar todos los espacios en blanco del final de la línea por nada. De este modo borramos los espacios en blanco.

Por lo tanto, para borrar todos los espacios en blanco al final de del fichero de texto `sedexamplesnew` ejecutaré el siguiente comando:

```
sed -i 's/ *$//' sedexamplesnew
```

Borrar la totalidad de tabulaciones existentes al final de una línea

Borraremos las tabulaciones del final de cada una de las líneas de forma similar a como lo hicimos con los espacios en el apartado anterior. Por lo tanto para eliminar las tabulaciones de final de línea del fichero `sedexamplesnew` usaremos el siguiente comando:

```
sed -i 's/[[:space:]]*$//' sedexamplesnew
```

Nota: `[[:space:]]` hace referencia a una tabulación.

Borrar los espacios vacíos al inicio de cada uno de las líneas

Para quitar la totalidad de espacios en blanco existentes al inicio de una línea lo

haremos de forma similar a lo que lo hicimos en los 2 apartados anteriores. Pero esta vez reemplazaremos la expresión regular `*$` por `^ *`.

Por lo tanto si tenemos el siguiente texto en el fichero `sedexamples`:

```
Borrando espacios en blanco
Al inicio de la línea.
```

Ejecutaremos el siguiente comando y obtendremos el siguiente resultado:

```
joan@gk55:~$ sed -i 's/^ *//' sedexamples

joan@gk55:~$ cat sedexamples
Borrando espacios en blanco
Al inicio de la línea.
```

Nota: La expresión regular `^ *` indicar todos los espacios que se encuentran en el inicio de la línea. Por lo tanto el comando aplicado reemplaza los espacios en blanco del inicio de la línea por nada.

Borrar las líneas vacías de un documento mediante Sed

Sed también permite borrar las líneas vacías de un documento. Para ello tan solo tendremos que buscar una expresión regular que defina que una línea está vacía. En nuestro caso la expresión regular es `^$`

Nota: `^` hace referencia al inicio de la línea. `$` Hace referencia al final de la línea. Si entre el inicio de la línea y el final de la línea no hay nada entonces significa que es una línea vacía.

Una vez conocemos la expresión regular supongamos que el fichero `sedexamplesnew` tiene el siguiente contenido:

HACER QUE LIBREOFFICE TENGA UN MODO OSCURO AL 100%

Usar un tema de escritorio Oscuro.

Usar un tema de iconos adecuado para el modo Oscuro.

Modificar el color del área de escritura y cambiar el color de

Si queremos eliminar todas las líneas en blanco tan solo tendremos que usar el siguiente comando:

```
sed -i '/^$/d' sedexamplesnew
```

Y el resultado obtenido será el siguiente:

HACER QUE LIBREOFFICE TENGA UN MODO OSCURO AL 100%

Usar un tema de escritorio Oscuro.

Usar un tema de iconos adecuado para el modo Oscuro.

Modificar el color del área de escritura y cambiar el color del
texto.

Fuentes

<https://www.gnu.org/software/manual/.html> < <https://www.gnu.org/software/sed/manual/sed.html>>



< <https://www.addtoany.com/share#url=linux-y-unix-con-ejemplos%2F&title=Uso%20del%20coman>



Únete a la conversación

■ 4 comentarios

Pedro

16 junio, 2022 a las 8:55 < <https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comment-40592> >

Estupenda guía. Muchas gracias

Test1

4 octubre, 2022 a las 14:26 < <https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comment-45006> >

Si tengo un documento que no se cuantas líneas tiene pero quiero eliminar las 200 ultimas lineas ¿Cómo lo hago?

Gracias!

Joan Carles

< <https://geekland.eu/> >

30 octubre, 2022 a las 15:21 < <https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comment-45905> >

Lo puedes intentar con:

```
for i in `seq 200`; do sed -i '$ d' tu_fichero.txt; done
```

Celia

22 diciembre, 2022 a las 16:54 < <https://geekland.eu/uso-del-comando-sed-en-linux-y-unix-con-ejemplos/#comment-47937> >

Buenas, he visto la parte de insertar texto después de todas las lineas, pero es posible añadir texto despues de una línea que cumpla con unas condiciones concretas? (que por ejemplo empiece por X palabra).

Gracias!

2024 geekland , **Funciona gracias a WordPress.** < <https://es.wordpress.org/> >
dos



Todos los derechos reserva-