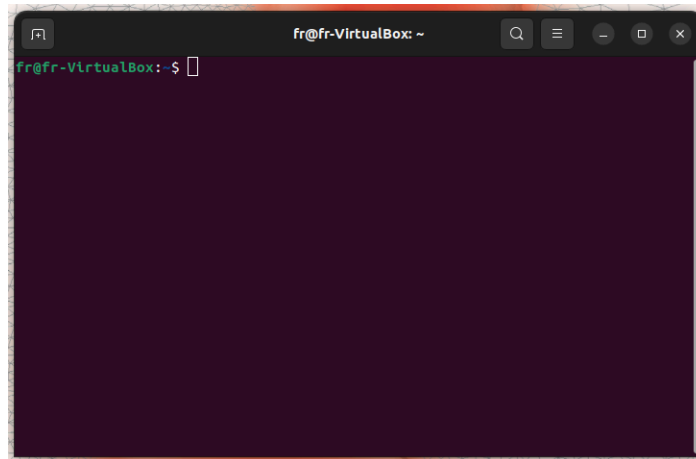


Comandos útiles en linux

A continuación les pasamos una serie de comandos y trucos que pueden serles de utilidad para realizar los TP's. La versión web de este archivo está en [este enlace](#).

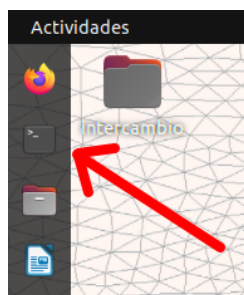
Abrir una terminal

La terminal es el lugar desde el cual ejecutaremos los diversos comandos, y tiene la siguiente apariencia:

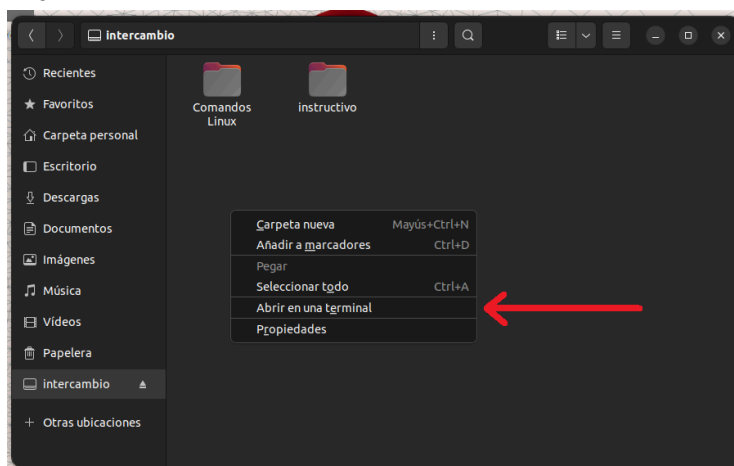


Para abrirla hay distintas opciones:

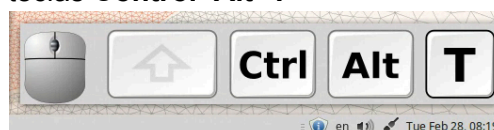
- Apretar el ícono de la terminal:



- Hacer click con el botón derecho del mouse dentro del explorador de ventanas, y elegir "Open in Terminal":



- Usar la combinación de teclas **Control+Alt+T**

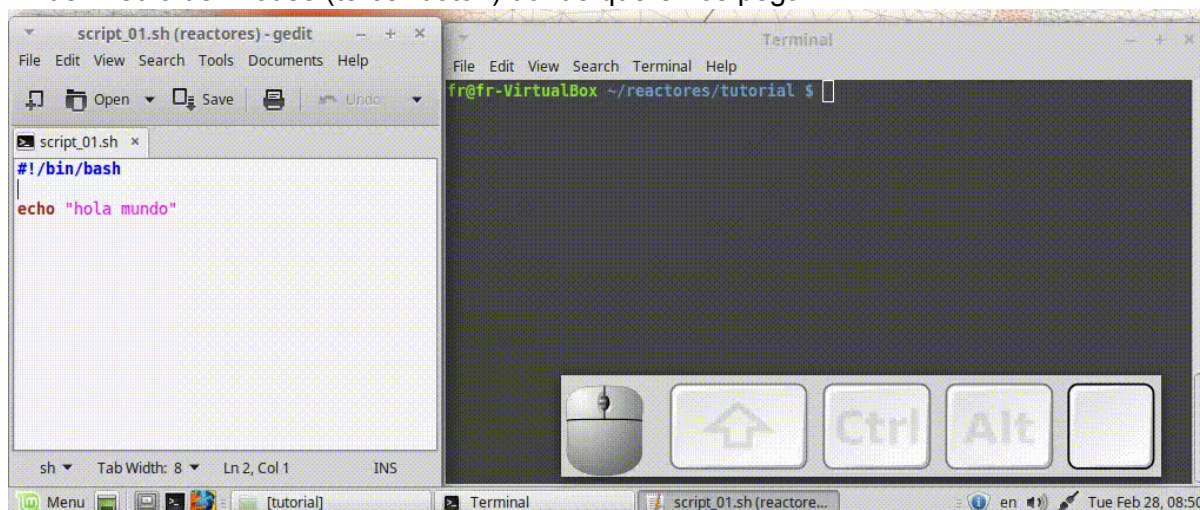


La diferencia es que en la segunda opción ya van a estar parados en el directorio donde quieren trabajar, mientras que en las otras dos opciones van a tener que navegar por los directorio usando el comando "**cd**".

Copiar y pegar texto

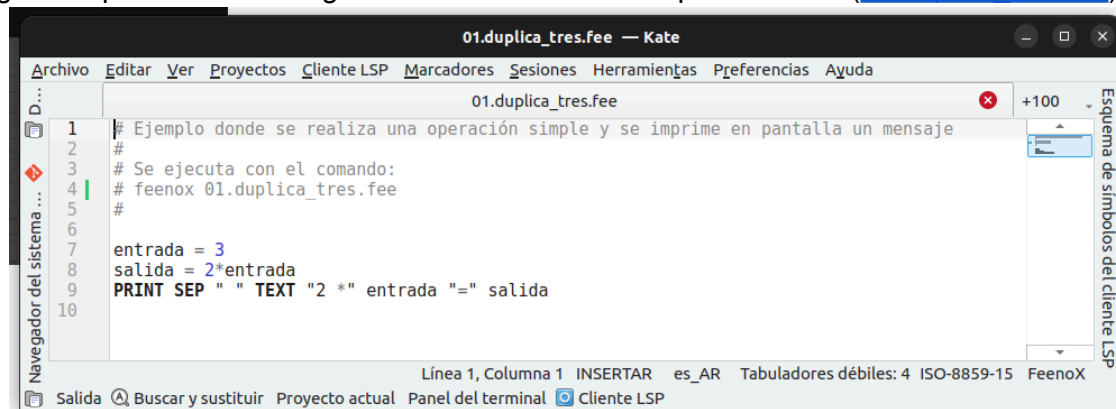
Para copiar y pegar un texto entre dos programas o ventanas, se pueden usar las siguientes opciones:

1. **la clásica con el mouse:** Primero seleccionamos el texto que queremos copiar. Luego hacemos click con el botón derecho del mouse sobre el texto seleccionado, y elegimos **Copy**. Luego vamos al lugar donde queremos pegar, hacemos click con el botón derecho del mouse sobre el lugar a pegar, y seleccionamos **Paste**.
2. **la que no utiliza el mouse:** Seleccionamos el texto, y apretamos **Ctrl+C**. Luego cambiamos de ventana con **Alt+Tab** y nos paramos con el cursor donde queremos pegar, y apretamos **Control+V**, o en la terminal apretamos **Control+Shift+V**
3. **la mágica de linux:** Pintamos con el mouse el texto a copiar, y luego apretamos con el botón del medio del mouse (tercer botón) donde queremos pegar.



Introducir un valor en un archivo de input

Supongamos que tenemos el siguiente archivo de entrada para Feenox ([01.duplica tres.fee](#)):



Si ejecutamos este archivo, obtenemos la siguiente salida:

```
fr@fr-VirtualBox ~/tutorial $ feenox 01.duplica_tres.fee
2 * 3 = 6
```

Como podemos apreciar, este programa siempre va a obtener el doble de tres. Pero si quisiéramos obtener el doble de otro valor, podríamos realizar el siguiente procedimiento manual:

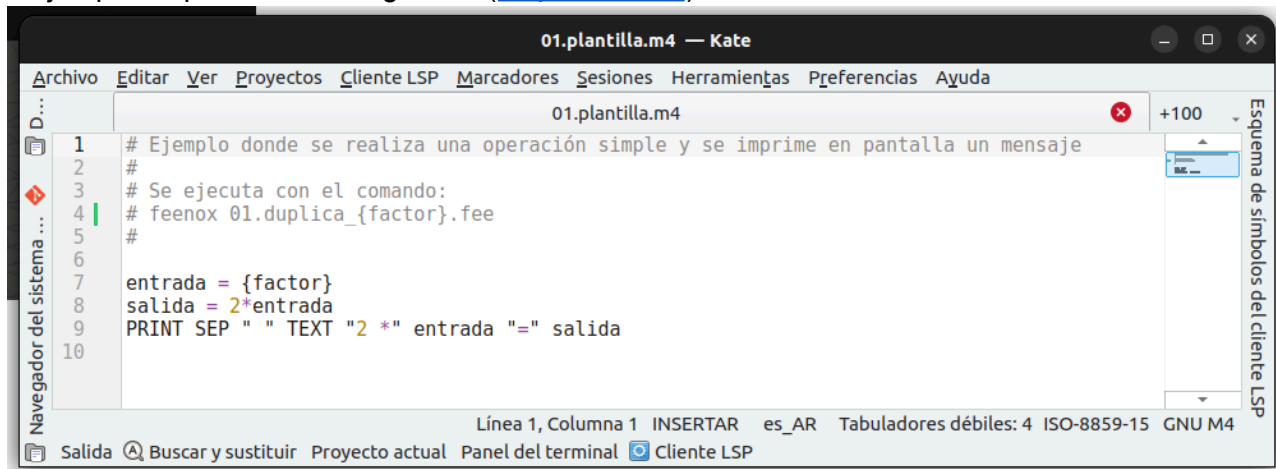
1. Abrir el archivo de input con un editor de texto.
2. Modificar el valor "3" por otro valor que queramos saber el doble.
3. Guardar y cerrar el archivo.
4. Y finalmente ejecutar nuevamente el FeenoX.

Claramente, esto no es difícil de realizar, pero puede ser muy tedioso si se quiere realizar varias veces el mismo procedimiento. Por ello, a continuación mostramos dos alternativas automatizables.

Introducir modificaciones con funciones nativas de linux, externas a FeenoX

Una forma de realizar cambios dentro de un archivo de texto, es utilizando el comando **sed**. Para ello primero debemos generar una plantilla que sea prácticamente un archivo de entrada para FeenoX, pero que en ciertos lugares específicos contenga etiquetas en donde queramos modificar el valor automáticamente.

Un ejemplo de plantilla es el siguiente ([01.plantilla.m4](#)):



```
1 # Ejemplo donde se realiza una operación simple y se imprime en pantalla un mensaje
2 #
3 # Se ejecuta con el comando:
4 # feenox 01.duplica_{factor}.fee
5 #
6
7 entrada = {factor}
8 salida = 2*entrada
9 PRINT SEP " " TEXT "2 *" entrada "=" salida
10
```

Como podemos apreciar, este archivo es prácticamente igual al anterior, pero en dos lugares específicos está colocada la etiqueta “{factor}”. La idea es que luego con el comando **sed** reemplazemos esa etiqueta con un valor numérico específico, y luego generemos un nuevo archivo de texto, el cual está listo para ser ejecutado por FeenoX.

A continuación mostramos un ejemplo de uso del comando **sed** sobre la plantilla:

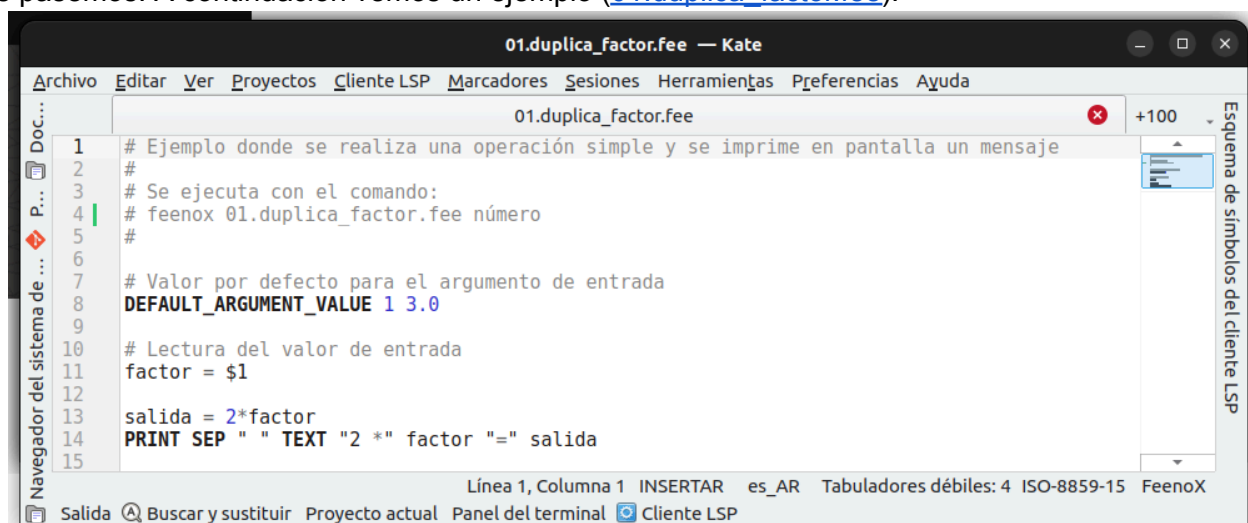
```
fr@fr-VirtualBox ~/tutorial $ sed s/{factor}/5/ 01.plantilla.m4 > 01.duplica_5.fee
fr@fr-VirtualBox ~/tutorial $ feenox 01.duplica_5.fee
2 * 5 = 10
```

El inconveniente que puede tener este procedimiento es que va a generar archivos nuevos para cada valor. Aunque esto no es necesariamente un problema, si queremos tener un registro de lo que fuimos haciendo.

Introducir modificaciones como argumento de entrada de FeenoX

Otra opción para introducir valores externamente, o en forma dinámica, es colocando estos valores como un argumento de entrada al ejecutar el FeenoX.

Para ello antes tenemos que preparar el archivo de entrada para que entienda los argumentos que les pasemos. A continuación vemos un ejemplo ([01.duplica_factor.fee](#)):



```
1 # Ejemplo donde se realiza una operación simple y se imprime en pantalla un mensaje
2 #
3 # Se ejecuta con el comando:
4 # feenox 01.duplica_factor.fee número
5 #
6
7 # Valor por defecto para el argumento de entrada
8 DEFAULT_ARGUMENT_VALUE 1 3.0
9
10 # Lectura del valor de entrada
11 factor = $1
12
13 salida = 2*factor
14 PRINT SEP " " TEXT "2 *" factor "=" salida
15
```

Como podemos apreciar, nuevamente este archivo es muy similar al input mostrado anteriormente, pero ahora leemos un argumento de entrada mediante la sentencia “\$1”. Lo que realiza el FeenoX, es reemplazar todos los lugares que diga “\$1” con el valor que le pasemos al ejecutar el programa.

Además podemos darle un valor por defecto, para el caso en el que no coloquemos ningún valor al llamar al programa.

Un ejemplo de ejecución sería el siguiente:

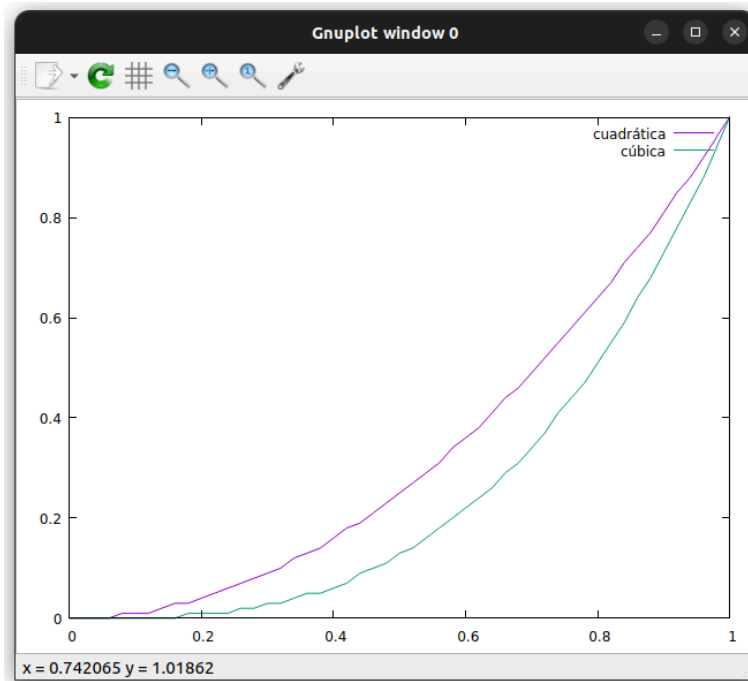
```
fr@fr-VirtualBox ~/tutorial $ feenox 01.duplica_factor.fee 4
2 * 4 = 8
```

Exportar a imagen el gráfico generado con GNUPlot

Suponiendo que tengamos un archivo de texto con los resultados de algún cálculo, donde los resultados están ordenados en columnas. Por ejemplo podría ser una columna con valores para la variable independiente (x), y dos columnas para dos funciones diferentes (y1 e y2), como los datos en el archivo [02.datos.dat](#), cuyas columnas son x, x² y x³. Entonces, para graficar ambas curvas podemos usar el comando:

```
fr@fr-VirtualBox ~/tutorial $ gnuplot -p -e "plot '02.datos.dat' u 1:2 w l t 'cuadrática',
'' u 1:3 w l t 'cúbica'"
```

Obteniendo un resultado como el siguiente:



Esta forma de usar el GNUPlot es muy útil cuando estamos haciendo pruebas, pero si queremos guardar el resultado, puede resultar conveniente acomodar un poco la figura, y exportarla como una imagen PNG. Para ello hay que cambiar la salida del programa, y además hay que configurar los detalles estéticos que queramos.

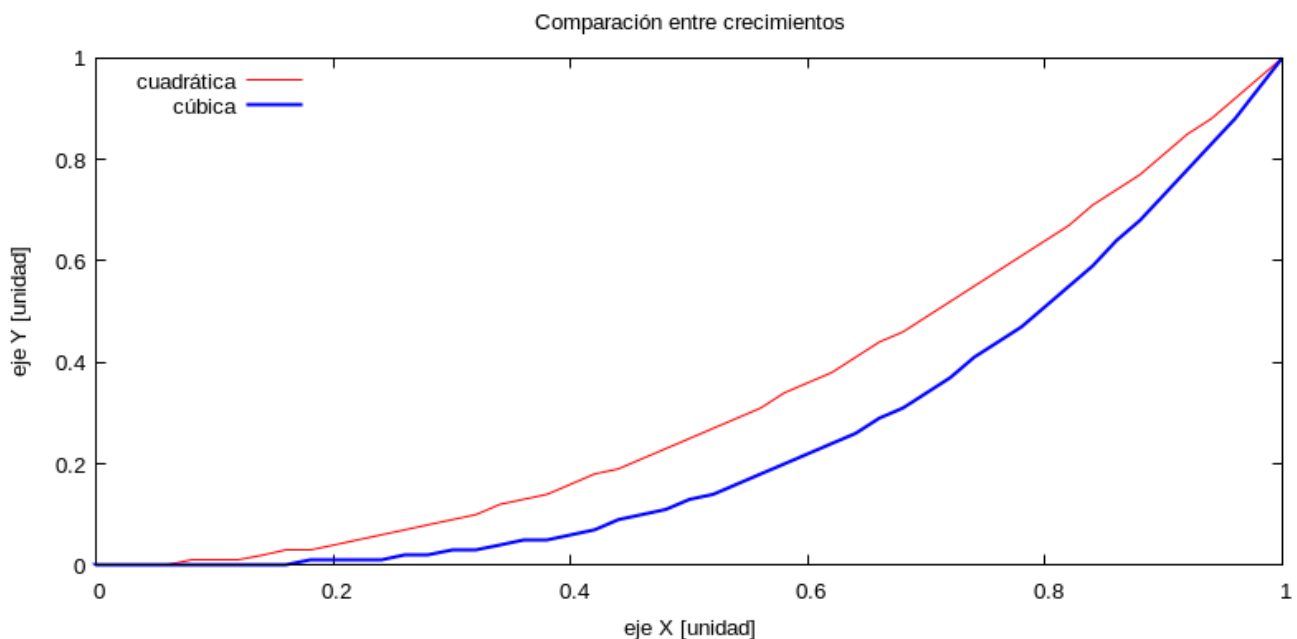
Si bien toda la configuración se puede hacer en una sola línea de comandos (separando cada configuración con punto y coma “;”), resulta mucho más legible si escribimos toda la configuración en un archivo separado, y luego cargamos esa configuración.

Un ejemplo de archivo de configuración es el siguiente ([02.conf.gnuplot](#)):

```
02.conf.gnuplot — Kate
Archivo Editar Ver Proyectos Cliente LSP Marcadores Sesiones Herramientas Preferencias Ayuda
02.conf.gnuplot +100
1 # Archivo de configuración para GNUplot
2 # Para más ejemplos ir a la página oficial:
3 # http://www.gnuplot.info/
4
5 # Defino estilo de línea
6 # Nombre del estilo: 11
7 # Color: red
8 # Ancho de línea: 1
9 set style line 11 lc rgb 'red' lw 1;
10
11 # Defino estilo de lía
12 # Nombre del estilo: 12
13 # Color: blue
14 # Ancho de línea: 2
15 set style line 12 lc rgb 'blue' lw 2;
16
17 # Nombres de ejes y títulos
18 set xlabel 'eje X [unidad]';
19 set ylabel 'eje Y [unidad]';
20 set title 'Comparación entre crecimientos';
21
22 # Defino límites para los ejes, para que vayan de cero a uno
23 set xrange [0:1];
24 set yrange [0:1];
25
26 # Coloco la leyenda arriba a la izquierda
27 set key left top;
28
29 # Defino para que la salida sea a un archivo PNG, y que la fuente sea Arial con tamaño 10
30 set terminal pngcairo size 800,400 enhanced font 'Arial,10';
31
32 # Ejemplo de uso
33 #gnuplot 02.conf.gnuplot -p -e "set output '02.salida.png'; plot '02.datos.dat' u 1:2 ls 11
w l t 'cuadrática', '' u 1:3 ls 12 w l t 'cúbica'"
Línea 1, Columna 1 INSERTAR es_AR Tabuladores débiles: 4 UTF-8 Gnuplot
Salida Buscar y sustituir Proyecto actual Panel del terminal Cliente LSP
```

Al cual le falta definir el nombre que tendrá la imagen, pero eso sí conviene hacerlo por línea de comandos:

```
fr@fr-VirtualBox ~/tutorial $ gnuplot 02.conf.gnuplot -p -e "set output '02.salida.png';
plot '02.datos.dat' u 1:2 ls 11 w l t 'cuadrática', '' u 1:3 ls 12 w l t 'cúbica'"
```



Ejecutar un conjunto de comandos

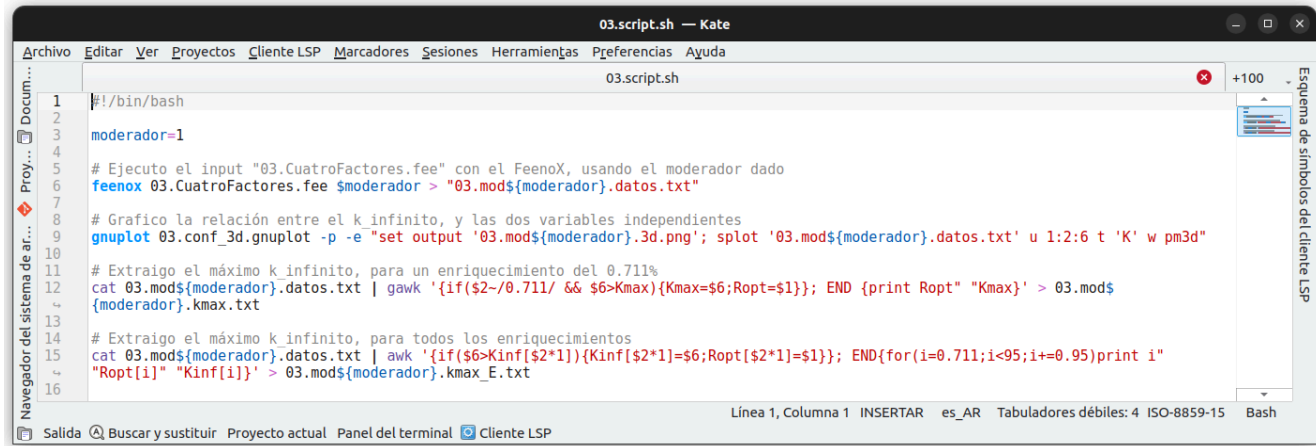
Es muy común que durante el desarrollo de los TP's tengan que ejecutar una serie de comandos varias veces. Para ello es probable que les sirva escribir estos comandos en un archivo de texto

ejecutable, mediante el lenguaje **bash**, el cual es un lenguaje nativo de muchas distribuciones de linux.

A continuación veremos un ejemplo de un script (secuencia de comandos) en **bash**, para realizar los siguientes pasos:

1. Ejecutar FeenoX con el archivo de entrada [03.CuatroFactores.fee](#) (el cual fue modificado para definir el moderador mediante un argumento de entrada).
2. Graficar en 3D el k_{∞} en función del enriquecimiento y la relación de moderación.
3. Extraer el máximo k_{∞} (y el valor de R respectivo), para un enriquecimiento del 0.711%.
4. Extraer el máximo k_{∞} (y el valor de R respectivo), para todos los enriquecimientos.

El archivo resultante es el siguiente ([03.script.sh](#)¹):



```
03.script.sh — Kate
Archivo Editar Ver Proyectos Cliente LSP Marcadores Sesiones Herramientas Preferencias Ayuda
03.script.sh
1 #!/bin/bash
2
3 moderador=1
4
5 # Ejecuto el input "03.CuatroFactores.fee" con el FeenoX, usando el moderador dado
6 feenox 03.CuatroFactores.fee $moderador > "03.mod${moderador}.datos.txt"
7
8 # Grafico la relación entre el k infinito, y las dos variables independientes
9 gnuplot 03.conf_3d.gnuplot -p -e "set output '03.mod${moderador}.3d.png'; splot '03.mod${moderador}.datos.txt' u 1:2:6 t 'K' w pm3d"
10
11 # Extraigo el máximo k infinito, para un enriquecimiento del 0.711%
12 cat 03.mod${moderador}.datos.txt | gawk '{if($2~/0.711/ && $6>Kmax){Kmax=$6;Ropt=$1}}; END {print Ropt" "Kmax}' > 03.mod${moderador}.kmax.txt
13
14 # Extraigo el máximo k infinito, para todos los enriquecimientos
15 cat 03.mod${moderador}.datos.txt | awk '{if($6>Kinf[$2*1]){Kinf[$2*1]=$6;Ropt[$2*1]=$1}}; END{for(i=0.711;i<0.95;i+=0.05)print i" "Ropt[i]" "Kinf[i]}' > 03.mod${moderador}.kmax_E.txt
16
```

Como podemos apreciar, al principio definimos el moderador, y luego el resto de los comandos tienen entradas y salidas referidas a ese moderador. De esta manera se generarán archivos de salida para cada tipo de moderador.

Este script se ejecuta de la siguiente manera:

```
fr@fr-VirtualBox ~/tutorial $ ./03.script.sh
```

Y si bien no devuelve ningún mensaje por consola, el mismo genera varios archivos automáticamente.

Realizar un barrido en valores (loop)

Dentro de la automatización que nos permite realizar un script en bash, podemos encontrar la opción de realizar barridos de algún parámetro de interés.

Por ejemplo, si queremos escribir un mensaje o hacer cálculos relacionados con el moderador 1, 2 y 3, entonces podemos usar un loop tipo **for**:

```
fr@fr-VirtualBox ~/tutorial $ for moderador in 1 2 3; do echo "El mod es: $moderador"; done
El mod es: 1
El mod es: 2
El mod es: 3
```

¹ Para ejecutar un script en Bash, es necesario que el archivo donde está el script tenga permisos de ejecución. Por ejemplo, si el script está en un archivo que se llama **script.sh**, entonces desde la terminal tienen que ejecutar el siguiente comando para cambiar los permisos:

sudo chmod 775 script.sh

Si el sistema pregunta la contraseña, la misma es **reactores**. Además en las terminales de Linux, cuando se escribe una contraseña, la misma no aparece mientras se escribe (por una cuestión de seguridad), pero sin embargo la terminal si está tomando lo que se está escribiendo.