PARTE I - **Trabajo con archivos planos delimitados**

Introducción

La siguiente Cheat Sheet es una compilación de las principales funciones y comandos que utilizo frecuentemente. Para mayor información, en Google se puede encontrar una gran variedad de estas CheatSheets.

Dentro de las principales utilidades para el trabajo de archivos planos con Bash se encuentran:

- Los comandos `cat, sed, cut, paste, head, tail, find, grep, wc, du, sort, uniq` [1]
- El uso de bucles **for**: `for variable in `comando`; do acciones; done`
- Los caracteres de redireccionamiento de flujo:
  - `|` (redirige la salida de un comando como entrada hacia otro)
  - `> archivo.txt` (redirige la salida hacia "archivo.txt")
  - `< archivo.txt` (alimenta "archivo.txt" hacia comando)

Para scripts más avanzados, una buena opción es utilizar `awk` , ó `perl`

Entorno de trabajo

Para comenzar a trabajar con archivos planos lo mejor es usar el terminal o la línea de comando que provee nuestro sistema operativo. Uno de los mejores intérpretes de comandos es Bash. Tanto en Linux como en OS X, este intérprete viene listo para su uso.

Bash bajo Windows

La mejor distribución de utilidades portadas de Linux, y que contiene Bash, es Cygwin:
https://www.cygwin.com/

Inspección de archivos planos

```
# Examinar las primeras 5 filas de "archivo.ext"
head -n 5 archivo.ext

# Para líneas muy largas (que ocasionan wrapping en el terminal),
# la opción es mandar la salida al comando less
head -n 5 archivo.ext | less -S

# Examinar las últimas 5 líneas del archivo
tail -n 5 archivo.ext

# Examinar la codificación de archivo (UTF, LATIN, etc)
file -I archivo.ext

# Mostrar cabecera de "archivo.csv"
# (campos separados con ";") como lista vertical
head -n 1 archivo.csv | tr ';' '\n'

# Agrega números de línea
head -n 1 archivo.csv | tr ';' '\n' | cat -n

# Copiar lista de campos a portapapeles (PBCOPY, sólo OS X)
head -n 1 archivo.csv | tr ';' '\n' | pbcopy

# Bajo Windows, es necesario isntalar el paquete
# cygwin-extras y usar los comandos putclip y getclip

# Ver los valores distintos de una columna (ej. la
# segunda) de "archivo.csv", sin incluir la cabecera

cut -d ";" -f 2 archivo.csv | sed -e "1d" | sort | uniq
```

Estadísticas de archivos

```
# Contar el número total de archivos .csv dentro de la carpeta
# actual y sus subdirectorios
find . -name "*.csv" | wc -l

# Contar el número de líneas de "archivo.ext"
wc -l archivo.ext

# Contar el número de líneas en todos los archivos .csv
wc -l *.csv

# Lo mismo, pero ordenando el resultado de manera decreciente
# por número de línea
wc -l *.csv | sort -n -r

# Contar el número de campos de "archivo.csv" (separados por ";")
head -n 1 archivo.csv | tr ';' '\n' | wc -l

# Contar el número de valores distintos dentro de un campo
# (ej. el quinto) en "archivo.ext" (separador ";"),
# Nota: no incluye la cabecera
cut -d ";" -f 5 archivo.ext | sed "1d" | sort | uniq | wc -l
```

Conversión y modificación de archivos

```
# Cambiar separador ";" a "|" en archivo.ext
sed -i -e 's/;/\|/g' archivo.ext

# Eliminar las primeras 3 líneas de archivo.ext y
# mandar el resultado a "archivo_limpio.ext"
cat archivo.ext | sed "1,3d" > archivo_limpio.ext

# Eliminar las líneas 5 y 6 de "archivo.ext",
# enviando el resultado a "archivo_limpio.ext"
cat archivo.ext | sed "5,6d" > archivo_limpio.ext
```

Búsqueda y reemplazo dentro de archivos

```
# Busca la palabra "palabra" dentro de los archivos .csv
# dentro del directorio actual
grep "palabra" *.csv

# Busca recursivamente "palabra" en todos los subdirectorios
grep -r --include "*.tsv" "palabra" .

# Otra forma
find . -name "*.tsv" | xargs grep "palabra"
```

Búsqueda de archivos

```
# Buscar "archivo.ext" dentro del directorio actual y
# todos los subdirectorios
find . -name "archivo.ext"

# Buscar archivos .csv de tamaños 1M o más
find . -name "*.csv" -size +1M

# Buscar archivos .tsv y los lista con su respectivo tamaño
for i in `find . -name "*.tsv" -size +1M`
do
ls -lh $i
done

# Buscar archivos modificados en los últimos 3 días
find . -name "*.tsv" -ctime -3d

# Buscar archivos modificados hace más de 3 días
find . -name "*.tsv" -ctime +3d

# Buscar directorios que comienzan con "patron"
find . -name "patron*" -type d
```

Procesamiento en batch

Una de los principales comandos para procesar archivos en batch

```
# Cambio masivo de extensiones
for file in *.ext; do
    mv "$file" "`basename $file .ext`.nex"
done
```

dsc-cheatsheet-bash: versión 1.0

1. Se sugiere estudiar en detalle las opciones de cada comando consultando su manual, ejecutando `man comando` ↩