

PROCESAMIENTO MASIVO ANUBIS.

Centro de Control Geodésico

Instituto Geográfico Agustín Codazzi

2024

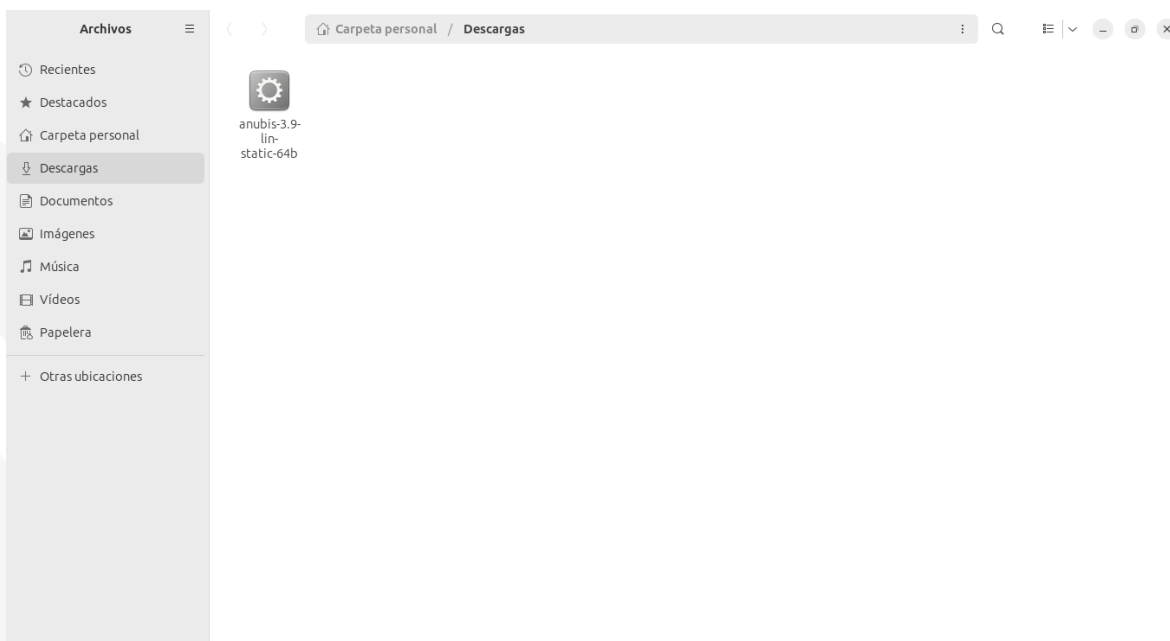
1. Instalación Anubis.....	2
2. Configuración de Anubis.....	2
3. Instalación de plot_anubis.....	6
4. Configuración plot_anubis.....	6
5. Procesamiento Masivo	6
5.1 Flujos del código.....	6
5.2 Algunos comandos de bash script.....	7
5.3 Código.....	9
5.3.1 Configuración de la ruta de los archivos rinex	9
5.3.2 For: Lee y guarda las estaciones en el array	10
5.3.3 Definición del formato de salida de los archivos de calidad y de las imágenes	10
5.3.4. For: Ejecución de comando Anubis y plot anubis	11
5.4 Ejecución de procesamiento masivo	12
5.4.1 Opción a.....	12
5.4.2 Opción b	14
6. Próximas tareas.....	16

1. Instalación Anubis

En el instructivo de **Control de calidad de datos GNSS en G-Nut Anubis** ([Link](#)) de la página 2 a la 6 se encuentra el procedimiento para instalar una máquina virtual, el sistema operativo Linux y la descarga de Anubis.

2. Configuración de Anubis

Debemos considerar una configuración adicional para anubis. Cuando descargamos anubis nos aparecera de la siguiente manera



Consedale permisos como aparece en el instructivo mediante el comando

Chmod +x nombre_del_ejecutable

En este caso ***Chmod +x anubis-3.9-lin-static-64b***

```
yesid-guerrero@Yesid:~/Descargas$ chmod +x anubis-3.9-lin-static-64b  
yesid-guerrero@Yesid:~/Descargas$
```

Sin embargo, al ejecutar el comando toca escribir ***./anubis-3.9-lin-static-64b***.

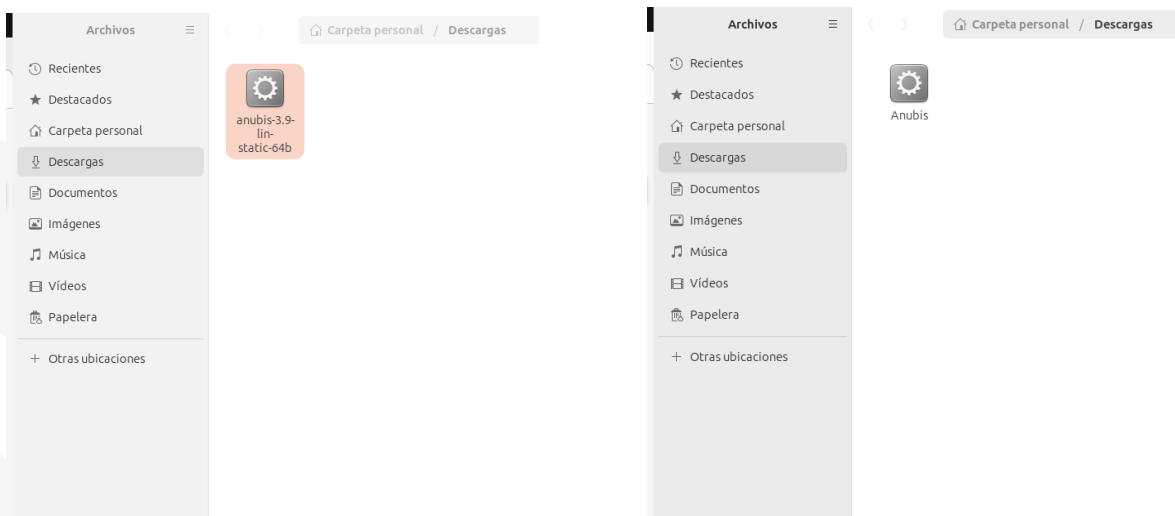
Ejecutando ***./anubis-3.9-lin-static-64b***. funciona

```
yesid-guerrero@Yesid:~/Descargas$ ./anubis-3.9-lin-static-64b  
yesid-guerrero@Yesid:~/Descargas$
```

Ejecutando **Anubis** no funciona, no encuentra la orden

```
yesid-guerrero@Yesid:~/Descargas$ Anubis  
Anubis: no se encontró la orden  
yesid-guerrero@Yesid:~/Descargas$
```

Debe renombrar el ejecutable a **Anubis**.



Ejecutando nuevamente **./Anubis** comprobamos que funcione.

```
yesid-guerrero@Yesid:~/Descargas$ ./Anubis
yesid-guerrero@Yesid:~/Descargas$
```

Ahora si nos ubicamos en otra carpeta diferente a la ubicación del archivo y ejecutamos **./Anubis**, el sistema no encontrara la orden.

```
yesid-guerrero@Yesid:~$ ./Anubis
bash: ./Anubis: No existe el archivo o el directorio
```

Esto se debe a que el ejecutable no se encuentra en una carpeta que está configurada como variable de entorno. Para ejecutar Anubis necesitaríamos indicarle toda la ruta de ubicación del ejecutable como se indica en la siguiente imagen.

```
yesid-guerrero@Yesid:~$ /home/yesid-guerrero/Descargas/Anubis
yesid-guerrero@Yesid:~$
```

Pero para evitar esto y que podamos ejecutar el comando Anubis desde cualquier ruta, debemos ubicar el ejecutable en una carpeta que se encuentre en el PATH del sistema. Por ejemplo las direcciones **"/usr/local/bin"** o **"/usr/bin"**.

Si nos movemos a esta carpeta con el comando **cd /usr/bin** y listamos los archivos con **ls**. Encontrará todos los ejecutables del sistema. A esta carpeta debe mover el ejecutable Anubis

```
yesid-guerrero@Yesid: /usr/bin
yesid-guerrero@Yesid:~$ cd /usr/bin/
yesid-guerrero@Yesid:/usr/bin$ ls
['
411toppm                                ippfind                                ppmtopjxl
aa-enabled                              iptool                                ppmtopppn
aa-exec                                iptables-xml                          ppmtopuzz
aa-features-abi                        ischroot                              ppmtorgb3
aconnect                               isdv4-serial-debugger                 ppmtosixel
activate-global-python-argcomplete     isdv4-serial-inputattach              ppmtospu
add-apt-repository                     ispell-wrapper                        ppmtoterm
addpart                                jbigtopnm                             ppmtotga
addr2line                              jed                                    ppmtouil
afrscan-discover                       join                                   ppmtowinicon
alsabat                               journalctl                             ppmtoxpm
alsaloop                               jpeg2ktopam                           ppmtoyuv
alsamixer                              jpegtopnm                             ppmtoyuvsplit
alsatplg                               jpgicc                                ppmtv
alsaucm                                jq                                     ppmwheel
amidi                                  jsdiff                                pr
amixer                                 jsondiff                              precat
animate                                jsonpatch                             preconv
animate-im6                            json-patch-jsdiff                     preunzip
animate-im6.q16                        jsonpointer                           prezip
                                         json_pp                               prezip-bin
                                         jsonschema
                                         jupyter
```

Para mover el ejecutable utilice el siguiente comando **`sudo mv /home/yesid-guerrero/Descargas/Anubis /usr/bin/`**

Sudo es necesario para mover el archivo con privilegios de superusuario

mv es el comando para mover, seguido de la ruta ubicación del archivo, luego por la ruta de la carpeta a donde desea mover.

```
yesid-guerrero@Yesid:/usr/bin$ sudo mv /home/yesid-guerrero/Descargas/Anubis /usr/bin/
[sudo] contraseña para yesid-guerrero:
yesid-guerrero@Yesid:/usr/bin$
```

Listando nuevamente para verificar, en la carpeta **`/usr/bin`** se encuentra **Anubis**

```
yesid-guerrero@Yesid: /usr/bin
yesid-guerrero@Yesid:/usr/bin$ mv /home/yesid-guerrero/Descargas/Anubis /usr/bin/
mv: no se puede mover '/home/yesid-guerrero/Descargas/Anubis' a '/usr/bin/Anubis': Permiso denegado
yesid-guerrero@Yesid:/usr/bin$ sudo mv /home/yesid-guerrero/Descargas/Anubis /usr/bin/
[sudo] contraseña para yesid-guerrero:
yesid-guerrero@Yesid:/usr/bin$ ls
['
411toppm                                ipod-time-sync                        ppmtopj
aa-enabled                              ippfind                              ppmtopjxl
aa-exec                                iptool                                ppmtopppn
aa-features-abi                        iptables-xml                          ppmtopuzz
aconnect                               ischroot                              ppmtorgb3
activate-global-python-argcomplete     isdv4-serial-debugger                 ppmtosixel
add-apt-repository                     isdv4-serial-inputattach              ppmtospu
addpart                                ispell-wrapper                        ppmtoterm
addr2line                              jbigtopnm                             ppmtotga
afrscan-discover                       jed                                    ppmtouil
alsabat                               join                                   ppmtowinicon
alsaloop                               journalctl                             ppmtoxpm
alsamixer                              jpeg2ktopam                           ppmtoyuv
alsatplg                               jpegtopnm                             ppmtoyuvsplit
alsaucm                                jpgicc                                ppmtv
amidi                                  jsdiff                                ppmwheel
amixer                                 jsdiff                                pr
animate                                jsondiff                              precat
animate-im6                            jsonpatch                             preconv
animate-im6.q16                        json-patch-jsdiff                     preunzip
anubis                                jsonpointer                           prezip
Anubis                                json_pp                               prezip-bin
anytopnm                              jsonschema
                                         jupyter
```

Comprobamos también ejecutando **Anubis** en cualquier ubicación.

```
yesid-guerrero@Yesid:~/Imágenes$ Anubis
yesid-guerrero@Yesid:~/Imágenes$
```

3. Instalación de plot_anubis

En el instructivo de **Control de calidad de datos GNSS en G-Nut Anubis** ([Link](#)) de la página 7 a la 9 se encuentra el procedimiento para instalar el complemento Plot-Anubis y los requisitos.

4. Configuración plot_anubis

De manera similar para ejecutar **plot_anubis** debemos indicar la ruta completa del ejecutable, entonces para evitar esto debemos copiar el ejecutable y sus dependencias a la carpeta **/usr/bin/**.

La carpeta **plot_Anubis-...** tiene tres elementos



Se deben copiar en la ruta **/usr/bin/**, mediante las siguientes líneas de comandos.

sudo cp -r ruta_del_archivo /usr/bin/

En la siguiente imagen se muestran cómo se copiaron los tres elementos.

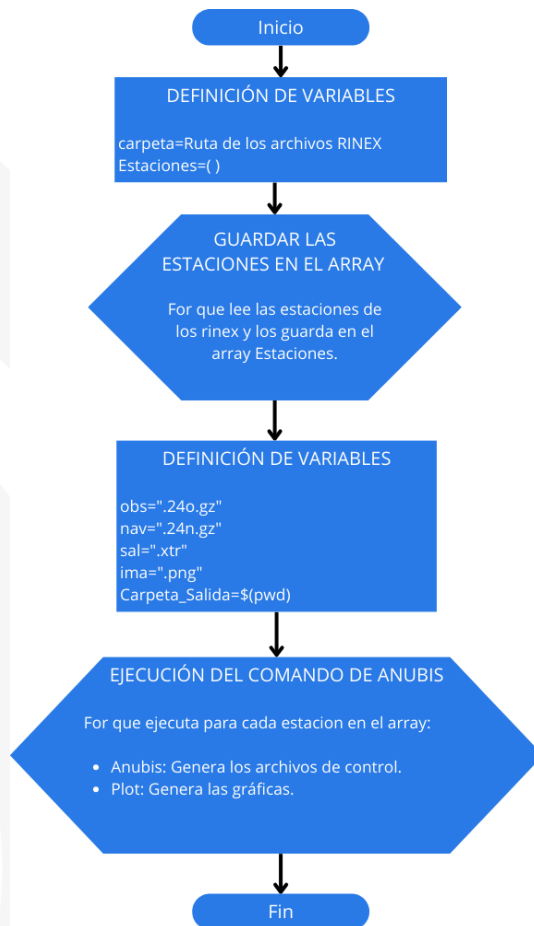
```
yesid-guerrero@Yesid:/bin$ sudo cp -r /home/yesid-guerrero/Documentos/Anubis/plot_Anubis-2.3-2023-03-06/plot_Anubis.pl /usr/bin/
yesid-guerrero@Yesid:/bin$ sudo cp -r /home/yesid-guerrero/Documentos/Anubis/plot_Anubis-2.3-2023-03-06/anubis.log /usr/bin/
yesid-guerrero@Yesid:/bin$ sudo cp -r /home/yesid-guerrero/Documentos/Anubis/plot_Anubis-2.3-2023-03-06/lib /usr/bin/
yesid-guerrero@Yesid:/bin$
```

5. Procesamiento Masivo

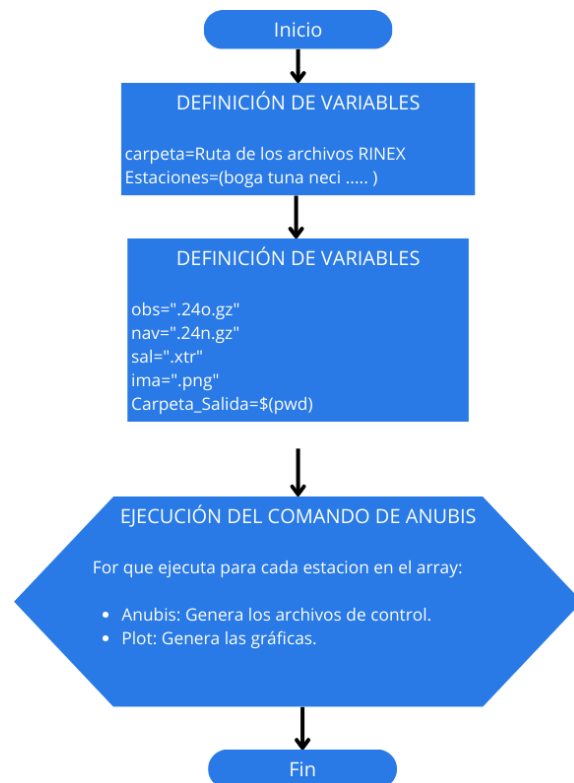
El procesamiento masivo se realiza mediante un código en Bash Script de Ubuntu. Bash Script es un archivo de texto que contiene una secuencia de comandos en el lenguaje de shell de Unix, está diseñado para ejecutarse en un entorno Linux.

5.1 Flujos del código

Se pueden realizar de dos formas el procesamiento masivo de archivos RINEX. En el primero, el programa leerá los archivos rinex ubicados en la carpeta y ejecutará Anubis para cada estación (Opción a). La segunda forma es indicarle al programa a cuáles estaciones quiere realizarles el control de calidad (Opción b).



Opción a



Opción b

5.2 Algunos comandos de bash script

- Definición de variables String
Nombre_variable="Cadena"
- Definición de un array vacío
Nombre_array=()
- Llamar una variable
\$nombre_variable

- Ciclo For

For **elemento** in “\$Nombre_array o carpeta”; do

Instrucciones

done

- if

If [**Condicion**] ; then

Instrucciones

fi

- Extraer el nombre de la ruta de un directorio

`$(basename "$archivo_path")`

- Extrae los primeros ocho caracteres del nombre de un archivo

`$(echo "$archivo" | cut -c 1-8)`

- Expande todos los elementos del array

`${nombre_array[@]}`

La siguiente línea verifica si **nombre_variable** no se encuentra en el array, es decir, compara la variable con cada uno de los elementos del array.

`! "${nombre_array[@]}" =~ "${nombre_variable}"`

- Calcula la longitud de un array

`${#Estaciones[@]}`

- Concatena dos cadenas String

`"${String1}${String2}"`

- Guardar y ejecutar un comando

Comando="**Anubis**"

\$Comando

- Imprime un mensaje
echo "Mensaje"

5.3 Código

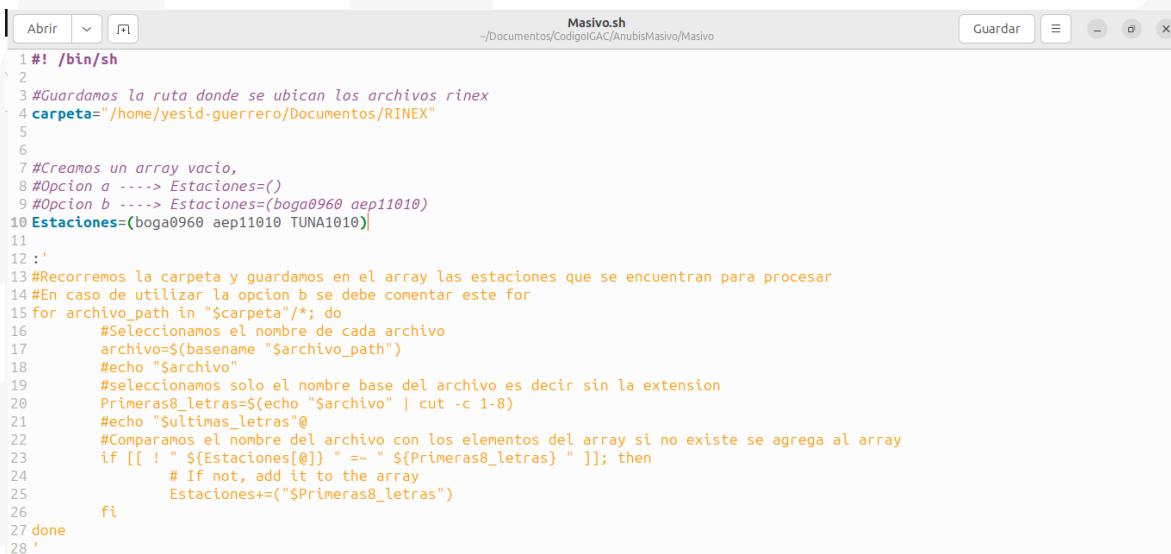
El código bash script se encuentra en un archivo llamado **Masivo.sh**. Este documento se debe ejecutar para procesar masivamente. Durante las siguientes secciones se muestra cómo funciona el código y cómo puede reconfigurarlo.

5.3.1 Configuración de la ruta de los archivos rinex

En la línea 4 se encuentra la variable carpeta acá debe configurar la ruta de los archivos rinex, en caso de procesar mediante la **opción a** la línea 8 se deja tal cual aparece en la siguiente imagen.

```
1 #!/bin/sh
2
3 #Guardamos la ruta donde se ubican los archivos rinex
4 carpeta="/home/yesid-guerrero/Documentos/RINEX"
5
6
7 #Creamos un array vacio
8 Estaciones=()
9
```

En caso de procesar mediante la **opción b** en el array de la línea 8 se deben colocar las estaciones que quiere procesar, como aparece en la siguiente imagen.



```
Abrir  Masivo.sh  Guardar
~/Documentos/CodigoIGAC/AnubisMasivo/Masivo

1 #!/bin/sh
2
3 #Guardamos la ruta donde se ubican los archivos rinex
4 carpeta="/home/yesid-guerrero/Documentos/RINEX"
5
6
7 #Creamos un array vacio,
8 #opcion a ----> Estaciones=()
9 #opcion b ----> Estaciones=(boga0960 aep11010)
10 Estaciones=(boga0960 aep11010 TUNA1010)
11
12 :
13 #Recorremos la carpeta y guardamos en el array las estaciones que se encuentran para procesar
14 #En caso de utilizar la opcion b se debe comentar este for
15 for archivo_path in "$carpeta"/*; do
16     #Seleccionamos el nombre de cada archivo
17     archivo=$(basename "$archivo_path")
18     #echo "$archivo"
19     #seleccionamos solo el nombre base del archivo es decir sin la extension
20     Primeras8_letras=$(echo "$archivo" | cut -c 1-8)
21     #echo "$ultimas_letras"
22     #Comparamos el nombre del archivo con los elementos del array si no existe se agrega al array
23     if [[ ! " ${Estaciones[@]} " =~ " ${Primeras8_letras} " ]]; then
24         # If not, add it to the array
25         Estaciones+=("$Primeras8_letras")
26     fi
27 done
28
```

5.3.2 For: Lee y guarda las estaciones en el array

Este for recorre los rinex que se encuentran en la variable **carpeta**. Ya que **archivo_path** (Ej: /home/yesid-guerrero/Documentos/RINEX/aep11010.24o.gz), es la ruta de cada archivo, es necesario utilizar **basename** para guardar solo el nombre en la variable **archivo** (Ej: aep11010.24o.gz), luego mediante **cut** se toman solo los primeros 8 caracteres y se guarda en **Primeras8_letras** (aep11010). Finalmente, mediante el if comparamos si **Primeras8_letras** existe en el array **Estaciones**. Si no existe se agrega.

NOTA: ESTE FOR SE DEBE COMENTAR SI SE UTILIZA LA OPCION B

```
13 #Recorremos la carpeta y guardamos en el array las estaciones que se encuentran para procesar
14 #En caso de utilizar la opcion b se debe comentar este for
15 for archivo_path in "$carpeta"/*; do
16     #Seleccionamos el nombre de cada archivo
17     archivo=$(basename "$archivo_path")
18     #echo "$archivo"
19     #seleccionamos solo el nombre base del archivo es decir sin la extension
20     Primeras8_letras=$(echo "$archivo" | cut -c 1-8)
21     #echo "$Ultimas_letras"
22     #Comparamos el nombre del archivo con los elementos del array si no existe se agrega al array
23     if [[ ! " ${Estaciones[@]} " =~ " ${Primeras8_letras} " ]]; then
24         # If not, add it to the array
25         Estaciones+=(" $Primeras8_letras ")
26     fi
27 done
```

5.3.3 Definición del formato de salida de los archivos de calidad y de las imágenes

En las siguientes líneas se cuenta la cantidad de elementos y muestra cuantas estaciones se van a procesar.

```
30 #Cuenta la cantidad de estaciones en la carpeta
31 Cantidad_Estaciones=${#Estaciones[@]}
32 echo "En la carpeta se encuentran $Cantidad_Estaciones estaciones para procesar"
33
34 #terminaciones archivos
35 obs=".24o.gz"
36 nav=".24n.gz"
37 sal=".xtr"
38 ina=".png"
39 Carpeta_Salida=$(pwd)
```

Luego se definen las terminaciones de los rinex observados **obs** y navegados **nav**. Cambiar estas variables está sujeto a determinar si Anubis puede determinar la calidad de rinex en otro formato, por ejemplo, hatanaka. Se deberían cambiar a **24d.gz**.

La variable **sal** define la terminación de los archivos de calidad de los rinex, se debe cambiar según los formatos definidos en el manual de Anubis.

La variable **ima** define la terminación de las imágenes resultado de **plot_anubis** también está sujeto a los formatos permitidos en el manual.

5.3.4. For: Ejecución de comando Anubis y plot anubis

Este for recorre cada elemento del array **estaciones** (Ej: aep11010) y para cada uno realiza primero la concatenación con las terminaciones y las guarda en **Observado**, **Navegado**, **salida** e **Imagen** (Ej: aep11010.24o.gz, ...), luego guarda en **Comando_a_correr** el comando anubis para esta estación (Este comando lo puede modificar según desee), luego ejecuta el comando.

Después guarda en **Comando_plotear** el comando que genera las gráficas (Este comando lo puede modificar según desee) y corre el comando.

Realiza este procedimiento para cada estación.

```

41 #Ejecuta el comando para cada estación guardada en el arreglo
42 for e in "${!Estaciones[@]};do
43     Observado="${Estaciones[$e]}{obs}"
44     Navegado="${Estaciones[$e]}{nav}"
45     Salida="${Estaciones[$e]}{sal}"
46     Imagen="${Estaciones[$e]}{ima}"
47
48     Comando_a_correr="anubis :inp:rinexn $carpeta/$Navegado :inp:rinexo $carpeta/$Observado :out:xtr
$Salida :qc:sec_mpx=2 :qc:sec_snr=2 :qc:sec_est=2 :qc:sec_sum=1"
49
50     echo "Ejecutando Comando con: ${Estaciones[$e]}"
51     $Comando_a_correr
52
53     Comando_plotear="plot_Anubis.pl --ifile $Carpeta_Salida/$Salida --plot=$Imagen --all --all --title=$Salida"
54     echo "Ejecutando Comando con: ${Estaciones[$e]}"
55     $Comando_plotear
56
57 done

```

NOTAS:

- LOS COMANDOS DE ANUBIS Y PLOT_ANUBIS SE PUEDEN MODIFICAR SEGUN LO INDICADO EN EL MANUAL DE ANUBIS.
- SI SOLO SE DESEA EJECUTAR EL COMANDO DE ANUBIS SE DEBE COMENTAR **\$Comando_plotear**

- POR DEFECTO EL COMANDO DE ANUBIS ESTA CONFIGURADO PARA REALIZAR EL CONTROL DE CALIDAD TANTO DEL RINEX OBSERVADO COMO DEL RINEX NAVEGADO. ES DECIR, QUE SI EN LA CARPETA FALTA ALGUNOS ARCHIVOS EL CODIGO NO FUNCIONARA. SI DESEA MODIFICAR PARA QUE SOLO UTILICE YA SEA OBSERVADOS O SOLO NAVEGADOS DEBE CAMBIAR ESAS LINEAS DE COMANDO SEGUN INDICA EL MANUAL DE ANUBIS.

5.4 Ejecución de procesamiento masivo

Para ejecutar el código primero debe configurar la ruta donde se encuentran los archivos rinex, configure según se indicó en la **sección 5.3.1**. Si desea configurar los formatos de salida, realícelo como se indica en la **sección 5.3.3**, y si desea cambiar el código de Anubis debe configurar según la **sección 5.3.4** y el **manual de Anubis**.

5.4.1 Opción a.

Realiza el procesamiento de calidad de todos los archivos Rinex contenidos en una carpeta.

Para ejecutar según la opción a, debe configurar de la siguiente manera:

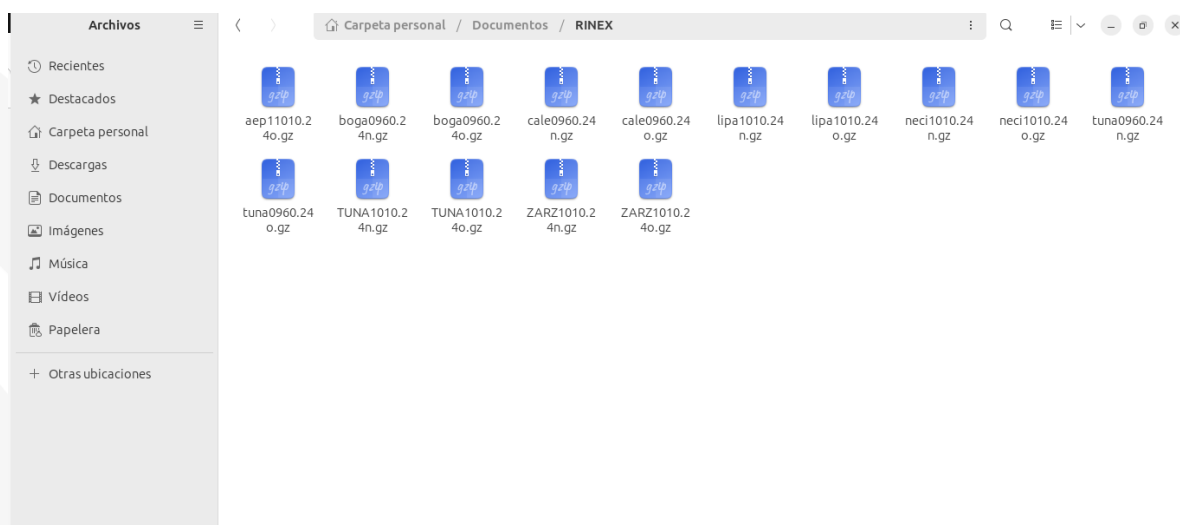
Ingresa la ruta donde se encuentran los archivos Rinex en la variable **carpeta**. El array **estaciones** debe estar vacío. Además, el for que completa el array estaciones **no** debe estar comentado.

```

1 #! /bin/sh
2
3 #Guardamos la ruta donde se ubican los archivos rinex
4 carpeta="/home/yesid-guerrero/Documentos/RINEX"
5
6
7 #Creamos un array vacio,
8 #Opcion a ----> Estaciones=()
9 #Opcion b ----> Estaciones=(boga neci tuna)
10 Estaciones=()
11
12 #Recorremos la carpeta y guardamos en el array las estaciones que se encuentran para procesar
13 #En caso de utilizar la opcion b se debe comentar este for
14 for archivo_path in "$carpeta"/*; do
15     #Seleccionamos el nombre de cada archivo
16     archivo=$(basename "$archivo_path")
17     #echo "$archivo"
18     #seleccionamos solo el nombre base del archivo es decir sin la extension
19     Primeras8_letras=$(echo "$archivo" | cut -c 1-8)
20     #echo "$Ultimas_letras"
21     #Comparamos el nombre del archivo con los elementos del array si no existe se agrega al array
22     if [[ ! " ${Estaciones[@]} " =~ " ${Primeras8_letras} " ]]; then
23         # If not, add it to the array
24         Estaciones+=("$Primeras8_letras")
25     fi
26 done
27

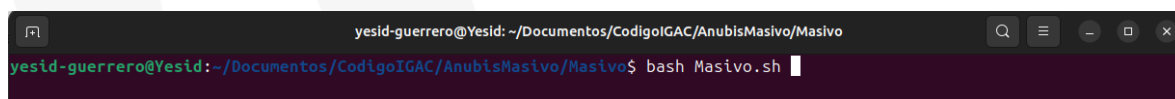
```

En la ruta de los rinex tenemos los siguientes archivos

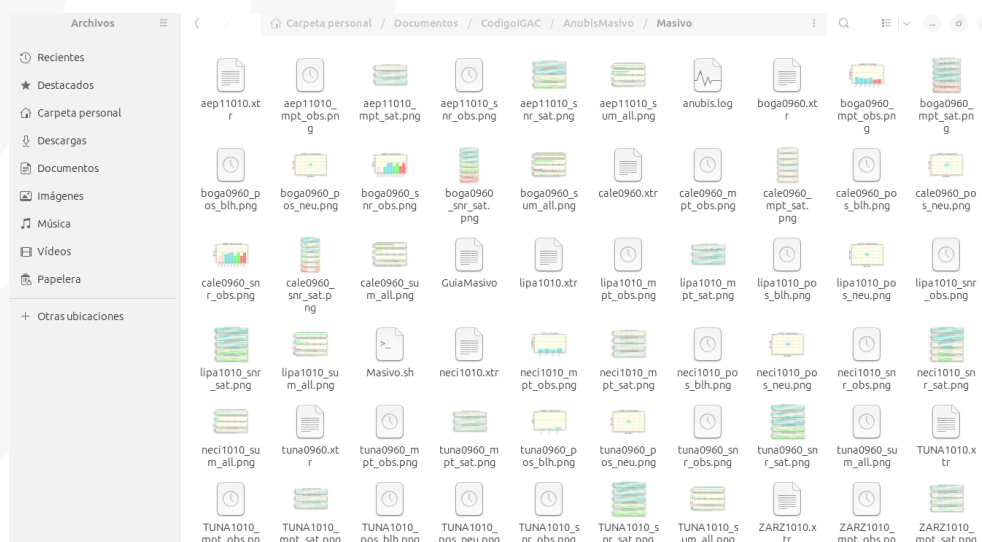


Ingresamos a la terminal desde la ubicación donde se encuentre **Masivo.sh**

Ejecutamos con el siguiente comando ***bash Masivo.sh***



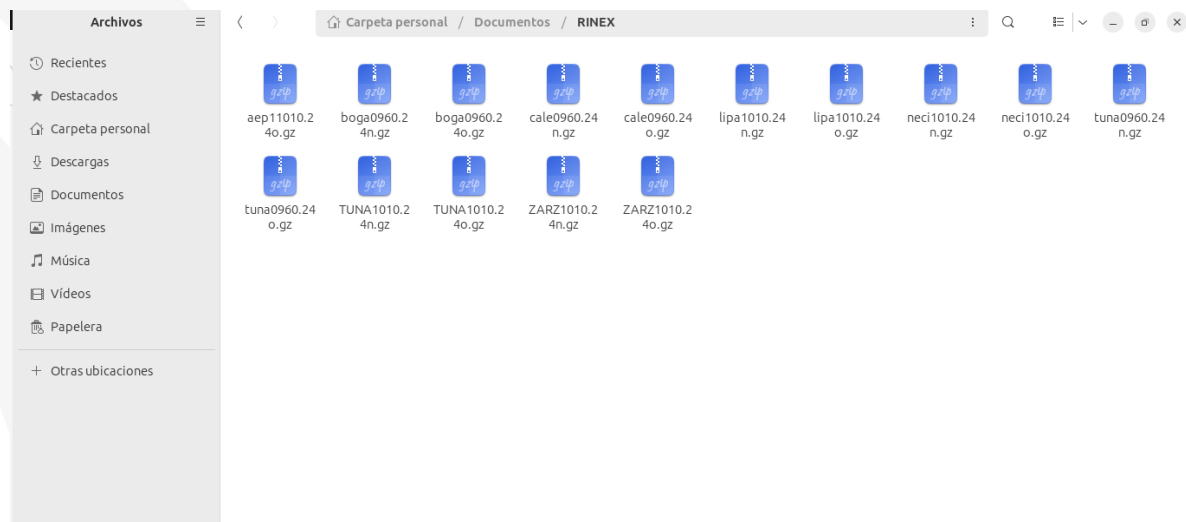
En esa misma carpeta se generan todos los archivos de texto con los datos de calidad y las gráficas.



5.4.2 Opción b

Realiza el procesamiento de calidad de solo algunas estaciones

Del siguiente conjunto de datos RINEX solo quiero procesar los datos de boga0960 y de aep11010.



Entonces debe configurar de la siguiente manera: coloque la ruta de los RINEX en la variable **carpeta**, en la variable estaciones debe colocar las estaciones con el día que quiere procesar por ejemplo **Estaciones=(boga0960 aep11010 TUNA1010)**, y por último se debe comentar el for que completa el array estaciones. Para comentar antes del for coloque **:** y al finalizar el for coloque **:**. A modo de ejemplo revise la siguiente imagen. Luego de configurar guarde y ejecute en terminal con **bash Masivo.sh**

NOTA: Los Rinex correspondientes a las estaciones y día que coloque en el array deben estar en la carpeta sino el código no funcionara.

```

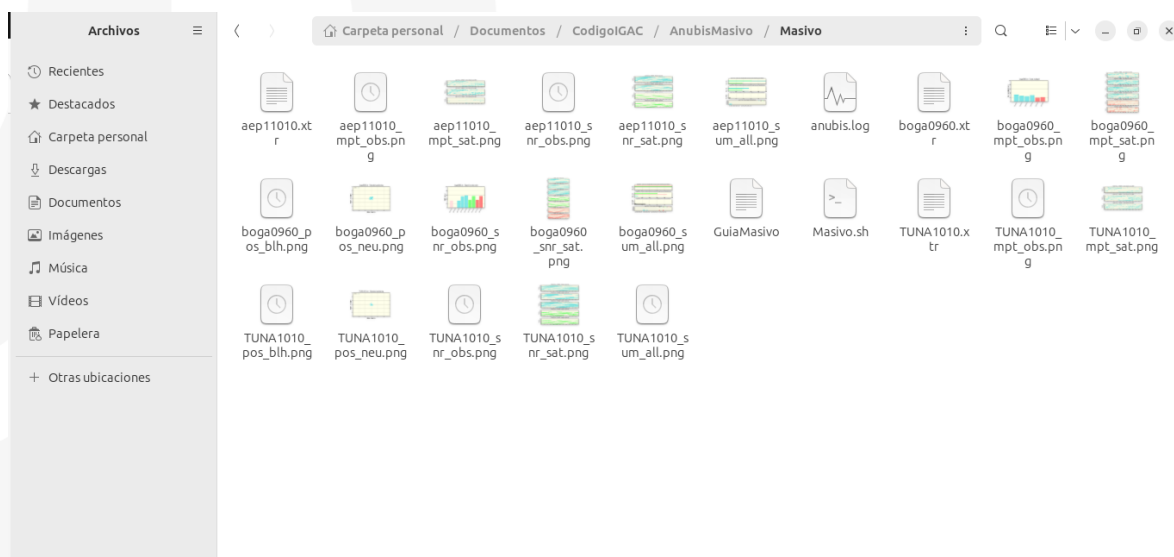
Abrir  Masivo.sh  Guardar
~/Documentos/CodigoIGAC/AnubisMasivo/Masivo

1 #!/bin/sh
2
3 #Guardamos la ruta donde se ubican los archivos rinex
4 carpeta="/home/yesid-guerrero/Documentos/RINEX"
5
6
7 #Creamos un array vacio,
8 #opcion a ----> Estaciones=()
9 #opcion b ----> Estaciones=(boga0960 aep11010)
10 Estaciones=(boga0960 aep11010 TUNA1010)
11
12 :
13 #Recorremos la carpeta y guardamos en el array las estaciones que se encuentran para procesar
14 #En caso de utilizar la opcion b se debe comentar este for
15 for archivo_path in "$carpeta"/*; do
16     #Seleccionamos el nombre de cada archivo
17     archivo=$(basename "$archivo_path")
18     #echo "$archivo"
19     #seleccionamos solo el nombre base del archivo es decir sin la extension
20     Primeras8_letras=$(echo "$archivo" | cut -c 1-8)
21     #echo "$ultimas_letras"
22     #Comparamos el nombre del archivo con los elementos del array si no existe se agrega al array
23     if [[ ! " ${Estaciones[@]} " =~ " ${Primeras8_letras} " ]]; then
24         # If not, add it to the array
25         Estaciones+=("$Primeras8_letras")
26     fi
27 done
28

```

Nota: Si el archivo Rinex se encuentra en mayúscula, debe agregarlo al array en mayúscula sino el código no lo reconocerá.

En la carpeta solo se generaron los archivos correspondientes a las tres estaciones seleccionadas.



6. Próximas tareas

- Se debe adaptar el código para que lea los archivos RINEX desde Elite y genere los archivos y graficas de control de calidad.
- Ya que el programa se encuentra en Linux y Elite está en Windows se debe realizar la conexión entre los dos sistemas operativos. Podría realizarse mediante un servidor o mediante una máquina virtual.
- Las salidas se guardan en la misma carpeta en al que se encuentra **Masivo.sh** se debe configurar el código para que lo guarde de una forma más organiza.
- Se debe ajustar el código para que seleccionar la **opción a** o **b** sea más fácil de configurar. Quizás programando mediante funciones que las llame dependiendo la opción que se seleccione.
- Se debe generar una interfaz gráfica para visualizar los resultados.