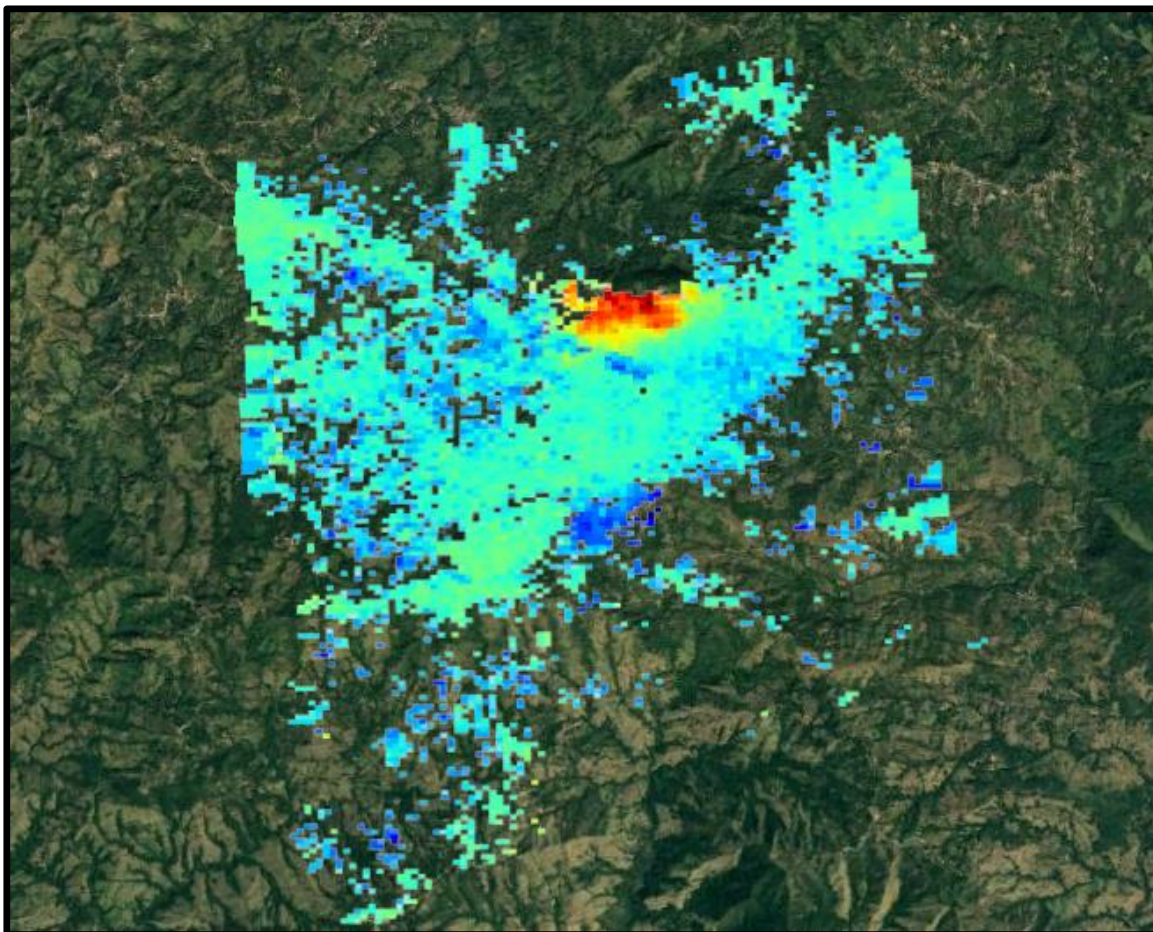


**Manual para procesamiento InSAR y de series temporales, usando el
módulo topsStack de ISCE junto a Mintpy**



Elaborado por: Ing. Andrey Cortés Sánchez

OVSICORI

Universidad Nacional de Costa Rica

2021

Índice de contenidos

1. Instalación de programas	5
1.1 Instalación software MintPy	5
1.2 Instalación software ISCE	6
2. Metodología de trabajo InSAR	11
2.1 Descarga de imágenes SAR	11
2.1.1 ASF Data Search.....	11
2.2 Descarga de orbitas.....	15
2.2.1 Copernicus Open Acces Hub	15
2.3 Procesamiento con ISCE (topsStack)	17
2.4 Posibles errores al ejecutar stackSentinel.py.....	27
2.4.1 Error por dirección de la carpeta donde se encuentra el DEM.....	27
2.4.2 Error en la descarga automática de orbitas	29
2.5 Procesamiento con MintPy (SBAS).....	30
2.6 Descarga de GACOS.....	33
2.7 Posibles errores con MintPy.....	38
2.7.1 Error por correr MintPy en otro directorio	38
2.7.2 Error por número de subswath	38
3. Limpieza para disponibilidad de memoria	39
4. Referencias	40

Índice de figuras

Figura 1. Directorio de módulos stack de ISCE	9
Figura 2. Archivo README.md del módulo stack de ISCE	10
Figura 3. Página Alaska Satellite Facility	11
Figura 4. Selección de área de interés ASF	12
Figura 5. Filtro de búsqueda para detallar parámetros de imagen ASF	13
Figura 6. Opción para añadir imágenes al carrito de descargas ASF	13
Figura 7. Espacio de imágenes añadidas para la descarga ASF	14
Figura 8. Formato de nombre de imágenes	14
Figura 9. Página de Copernicus	15
Figura 10. Parámetros de búsqueda de orbitas	16
Figura 11. Formato de nombre de archivo de orbitas	16
Figura 12. Parámetros del comando stackSentinel.py	18
Figura 13. Archivos DEM al aplicar gdal_translate	19
Figura 14. Subswath 1 órbita ascendente	20
Figura 15. Subswath 2 órbita ascendente	20
Figura 16. Subswath 3 órbita ascendente	21
Figura 17. Subswath 1 órbita descendente	21
Figura 18. Subswath 2 órbita descendente	21
Figura 19. Subswath 3 órbita descendente	22
Figura 20. Contenido de la carpeta para la ejecución de stackSentinel.py	22
Figura 21. Desactivar entorno miniconda	23
Figura 22. Resultado del stackSentinel.py (run files)	24
Figura 23. Archivo ejecutable bash_stack.sh	24
Figura 24. Visor de interferogramas de ISCE	26
Figura 25. Error por metadata en los archivos del DEM	27
Figura 26. Contenido de la carpeta DEM	27
Figura 27. Metadata del archivo .dem.wgs8.xml	28
Figura 28. Dirección del DEM en la metadata	28
Figura 29. Error por orbitas	29
Figura 30. Carpeta con orbitas precisas	29
Figura 31. Insumos de carpeta para la ejecución de MintPy	30
Figura 32. Archivo de texto necesario para la ejecución de MintPy y sus parámetros	32
Figura 33. Página de GACOS para descarga	33
Figura 34. Contenido de la carpeta mintpy antes de iniciar la ejecución de MintPy	34
Figura 35. Contenido de la carpeta GACOS	34
Figura 36. Final de procesamiento MintPy	35
Figura 37. Carpetas y archivos resultantes del procesamiento con MintPy	35
Figura 38. Comandos de MintPy	36
Figura 39. Productos finales de carpeta pic	36
Figura 40. Promedio de velocidades en LOS	37
Figura 41. Gráfica SBAS	37
Figura 42. Error por directorio en MintPy	38
Figura 43. Error por número de subswath en MintPy	38
Figura 44. Limpieza para disponibilidad de memoria	39

Consideraciones Previas

El manual explica el paso a paso de la metodología para procesar InSAR, haciendo uso de los algoritmos del software ISCE. Con el fin de obtener un conjunto de interferogramas dentro una determinada época de tiempo. Y realizar una serie temporal que permita cuantificar la deformación o desplazamiento en la superficie de estudio, utilizando los programas MintPy (SBAS) y StaMPS/MTI (PSI). Los tres programas usados son de código abierto, por lo que se pueden descargar de manera gratuita.

Con ISCE conocido como Interferometric synthetic aperture radar Scientific Computing Environment, es posible realizar un stack (pila) de interferogramas, el cual posteriormente se puede procesar con otro software para la generación de series temporales. Este software permite el procesamiento de stack con diferentes módulos, como es el modo Tops para interferometric wide (topsStack), el modo Strip map (stripmapStack) y Alos (alosStack). Con stripmapstack es posible procesar datos de varios sensores, mientras que cons topsStack solo datos satélite Sentinel-1 de la Agencia Espacial Europea (ESA). En este manual solo se usa el módulo topsStack.

The Miami INsar Time-series software in Python (MintPy), es uno de esos programas con los que se pueden generar series. Este software utiliza como insumo el stack de interferogramas, siendo capaz de leerlo para diferentes procesadores de interferometría (ISCE, GMTSAR, SNAP, GAMMA, entre otros). Para el caso de ISCE, el programa MintPy puede leer stacks procesados para los tres modulos mencionados en el párrafo anterior. Además, este programa aplica el algoritmo de SBAS para realizar de las series temporales.

Para la descarga e instalación de cada uno de los dos programas, los paso a paso, librerías y requerimientos de software, se adjunta la dirección web de la página oficial donde se encuentran ambos programas. En estas páginas de GitHub se encuentra colgado el software, es importante aclarar que son programas científicos, y su uso es a través de una terminal de comandos.

El sistema operativo más apto para correr o ejecutar los programas es Linux, por lo que es recomendado instalarlo en la computadora en caso de no contar con este. La distribución Linux que más se recomienda es Ubuntu, esta es posible descargarla e instalarla utilizando una máquina virtual. Entre las máquinas virtuales más reconocidas, se encuentra la de la corporación Oracle, pero en este caso la máquina virtual quitara memoria RAM y de espacio, según la memoria que le dispongamos. Otro modo es descargando Ubuntu desde la tienda oficial de Microsoft, e instalarlo como un subsistema Linux, cosa que ya permite sistemas operativos como Windows 10, lo que ahorraría gastos de memoria RAM y de espacio. El conocimiento en el manejo de Linux y su terminal de comandos es de gran ayuda para un mejor entendimiento y facilitación en el uso de los programas.

ISCE: <https://github.com/isce-framework/isce2> **MintPy:** <https://github.com/insarlab/MintPy>

1. Instalación de programas

1.1 Instalación software MintPy

Para la descarga e instalación del programa MintPy, se deben ejecutar la siguiente serie de comandos en la terminal:

```
sudo su
*poner contraseña
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt install python3.7
cd $HOME
mkdir tools
cd ~/tools
git clone https://github.com/insarlab/MintPy.git
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -b -p ~/tools/miniconda3
~/tools/miniconda3/bin/conda init bash
conda create -n mintpy
conda activate mintpy
conda install -c conda-forge --file ~/tools/MintPy/docs/requirements.txt
export MINTPY_HOME=~/tools/MintPy
export PATH=${PATH}:${MINTPY_HOME}/mintpy
export PYTHONPATH=${PYTHONPATH}:${MINTPY_HOME}
echo export PATH=${PATH}:${MINTPY_HOME}/mintpy >> /etc/bash.bashrc
echo export PYTHONPATH=${PYTHONPATH}:${MINTPY_HOME} >> /etc/bash.bashrc
source /etc/bash.bashrc
smallbaselineApp.py --help --steps
```

1.2 Instalación software ISCE

Para la descarga e instalación del programa ISCE, se resumió todo el proceso en un archivo bash, para simplemente ejecutar y que la instalación inicie automáticamente. Para realizar el archivo bash escribir el siguiente comando desde \$HOME:

```
sudo su
*poner contraseña
cd $HOME
nano ISCEinstall_bash.sh
```

Dentro del archivo ISCEinstall_bash.sh escribir el contenido de los siguientes tres cuadros:

```
#Actualización del sistema
sudo apt update
sudo apt upgrade

#Requisitos de librerías y paquetes
sudo apt install curl git build-essential gfortran libfftw3-3 libfftw3-dev xorg
sudo apt install libx11-dev libxm4 libxt-dev python3-h5py libhdf5-dev scons python3-numpy
gcc g++
sudo apt install make m4 python3-matplotlib python3-mpltoolkits.basemap python3-scipy
python3-pulp
sudo apt install python3-gdal libgdal-dev python3-pip libmotif-dev imagemagick grace cython3
sudo apt install libopencv-dev python3-opencv gdal-bin python3-astropy mlocate python3-
pulp

#Definición de variables de entorno
#Directorio principal de instalación
ISCE_ROOT="/usr/local"

#Nombre del directorio de instalación del software
ISCE_HOME="$ISCE_ROOT"/isce

#Directorio temporal donde se descargará el software
ISCE_TEMP=$HOME/isce_temp

#Ruta del instalador una vez descargado
ISCE_INSTALLER=$ISCE_TEMP/isce2
```

```

#Directorio de configuración
export SCONS_CONFIG_DIR=$ISCE_INSTALLER/SCONS_CONFIG_DIR

#Directorio temporal
mkdir $ISCE_TEMP
cd $ISCE_TEMP

git clone https://github.com/isce-framework/isce2.git

#Configuración para compilación
cd isce2

#Crear el directorio donde se escribirá el archivo
mkdir $SCONS_CONFIG_DIR

#Crear el archivo de configuración
file=$SCONS_CONFIG_DIR/SConfigISCE

#Definición de variables
hdf5Path=/usr/include/hdf5/serial
gdalPath=/usr/include/gdal
opencvPath=/usr/include/opencv4
pythonHDir=/usr/include/python3.8
numpyPath=/usr/lib/python3/dist-packages/numpy/core/include

#Archivo de configuración
echo PRJ_SCONS_BUILD=$ISCE_HOME > $file
echo PRJ_SCONS_INSTALL=$ISCE_HOME >> $file
echo LIBPATH=/usr/lib /usr/lib64 /usr/lib/gcc/x86_64-linux-gnu/9/ \
/usr/lib/x86_64-linux-gnu/hdf5/serial >> $file
echo CPPPATH=$pythonHDir $numpyPath $hdf5Path $gdalPath $opencvPath >> $file
echo FORTRANPATH=/usr/include >> $file
echo FORTRAN=/usr/bin/gfortran >> $file
echo CC=/usr/bin/gcc >> $file
echo CXX=/usr/bin/g++ >> $file
echo MOTIFLIBPATH=/usr/lib/x86_64-linux-gnu >> $file
echo X11LIBPATH=/usr/lib/x86_64-linux-gnu >> $file
echo MOTIFINCPATH=/usr/include/Xm >> $file
echo X11INCPATH=/usr/include/X11 >> $file
echo ENABLE_CUDA=False >> $file

```

```

#Compilación e instalación
#Entrar a la carpeta de instalación
cd $ISCE_INSTALLER
#Ejecutar el comando de compilación del software
scons
cd $HOME
#Verificación
#Directorio principal de instalación
export ISCE_ROOT="/usr/local"
#Nombre del directorio de instalación del software
export ISCE_HOME="$ISCE_ROOT"/isce
export ISCE_TEMP="$ISCE_TEMP"/isce2
export ISCE_INSTALLER="$ISCE_TEMP"/isce2
export
PATH="$ISCE_HOME"/bin:"$ISCE_HOME"/applications:"$ISCE_INSTALLER"/contrib/stack/tops
Stack:"$ISCE_INSTALLER"/contrib/timeseries/prepStackToStaMPS/bin:"$PATH"
export PYTHONPATH="$ISCE_ROOT":"$ISCE_HOME"/applications:"$ISCE_HOME"/components
echo export
PATH=$ISCE_HOME/bin:$ISCE_HOME/applications:$ISCE_TEMP/contrib/stack/topsStack:$ISCE
_TEMP/contrib/timeseries/prepStackToStaMPS/bin:$PATH >> /etc/bash.bashrc
echo export PYTHONPATH=$ISCE_ROOT:$ISCE_HOME/applications:$ISCE_HOME/components
>> /etc/bash.bashrc
#Actualizar bash.bashrc
source /etc/bash.bashrc
#Verificación de comandos
topsApp.py --help --steps
stackSentinel.py --help --steps

```

Una vez que se ha copiado todo el contenido en el bash, guardar y posteriormente ejecutar para iniciar instalación (este bash incluye la instalación del módulo topsStack):

```
./ISCEinstall_bash.sh
```


1.3 Instalación manual del módulo topsStack de ISCE

*(ignorar este apartado si instalo ISCE con el ISCEinstall_bash.sh)

En el caso anterior, se brinda un bash que automatiza la descarga e instalación de ISCE, dicho bash instala el módulo topsStack, por lo que si instala el programa usando el bash no es necesario ver este apartado. Pero cuando se instala ISCE de manera manual, la instalación normal no incluye los módulos de stack, solo son instalados los scripts Apps, como topsApp.py, stripmapApp.py y Alos2App.py. Estos solo funcionan para generar interferogramas individuales. En cambio, con el módulo stack si es posible crear una serie de interferogramas. Además, que para las series temporales con MintPy, solo se pueden usar insumos producidos por alguno de los módulos stack de ISCE. Este se encuentra en la carpeta “contrib” del directorio temporal “isce_temp”, tal y como se observa en la figura 1. Por tanto, para poder utilizar estos módulos se debe copiar la dirección de donde se ubican, en el PATH del archivo bash.bashrc.

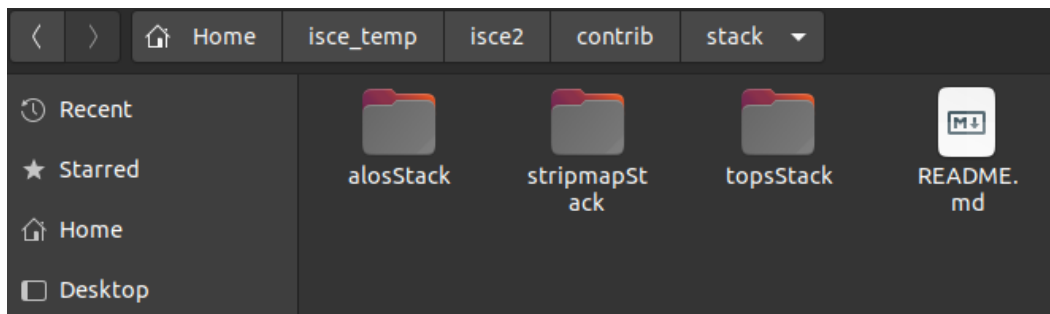


Figura 1. Directorio de módulos stack de ISCE
(Fuente: Elaboración propia, captura de pantalla)

En este caso, de los tres módulos disponibles en la carpeta stack se utilizará el topsStack, y se recomienda solo indicar uno de los tres en el archivo bash.bashrc, ya que puede generar errores incluir por el ejemplo el stripmapStack y el topsStack al mismo tiempo. Así lo señala el documento README.md, como se observa en la figura 2, además se indica en rojo la dirección que se debe incluir en el PATH para cada módulo.

```
Open  README.md  Save  -  X
~/isce_temp/isce2/contrib/stack

## Stack Processors
2
3 Read the document for each stack processor for details.
4
5 + [ stripmapStack](./stripmapStack/README.md)
6 + [ topsStack](./topsStack/README.md)
7 + [ alosStack](./alosStack/alosStack_tutorial.txt)
8
9 ### Installation
10
11 To use a stack processor you need to:
12
13 1. Install ISCE as usual
14
15 2. Depending on which stack processor you need to try, add the path of the folder containing the python scripts to your '$PATH'
   environment variable as follows:
16 - add the full path of your **contrib/stack/topsStack** to '$PATH' to use the topsStack for processing a stack of Sentinel-1
   TOPS data
17 - add the full path of your **contrib/stack/stripmapStack** to '$PATH' to use the stripmapStack for processing a stack of
   StripMap data
18 - set environment variable '$PATH_ALOSSTACK' by doing: export PATH_ALOSSTACK=CODE_DIR/contrib/stack/alosStack to use the
   alosStack for processing a stack of ALOS-2 data
19
20 Note: The stack processors do not show up in the install directory of your isce software. They can be found in the isce source
   directory.
21
22 ##### Important Note: #####
23
24 There might be conflicts between topsStack and stripmapStack scripts (due to common names of different scripts). Therefore users
   **MUST only** have the path of **one stack processor in their $PATH environment at a time**, to avoid conflicts between the two
   stack processors.
```

Figura 2. Archivo README.md del módulo stack de ISCE
(Fuente: Elaboración propia, captura de pantalla)

Una vez que ha sido incluido en el bash.bashrc, se le debe aplicar un source para actualizar los cambios. Lo primero es editar el bash con:

```
nano /etc/bash.bashrc
```

En el PATH añadir por ejemplo en este caso:

```
/root/isce_temp/isce2/contrib/stack/topsStack
```

Guardar los cambios con:

```
Ctrl x
```

Aplicar source al bash para actualizar los cambios:

```
source /etc/bash-bashrc
```

Posteriormente para comprobar que ha sido instalado de manera correcta, se debe escribir en la terminal el siguiente comando:

```
stackSentinel.py --help --steps
```

2. Metodología de trabajo InSAR

2.1 Descarga de imágenes SAR

Existen dos páginas para la descarga de datos satelitales, a continuación, se muestran los pasos para la descarga de imágenes en la página de ASF ya que es más accesible a la hora de la descarga. En el caso de Copernicus Open Acces Hub muchas veces las imágenes están con permisos por lo que no se logran descargar inmediatamente si no que hay que esperar a que las habiliten una vez se añaden a la descarga.

2.1.1 ASF Data Search

Se trata de Alaska Satellite Facility como se conoce en ingles perteneciente a la NASA, que permite la descarga de datos de las misiones Sentinel-1, además de varios otros sensores satelitales. Se deberá ingresar al siguiente link, una vez ahí aparecerá el interfaz que se ve en la figura 3. Es importante dirigirse a la opción de iniciar sesión para poder crear una cuenta, hecho esto se podrá continuar con la descarga de datos.

ASF Data Search: <https://search.asf.alaska.edu/#/>

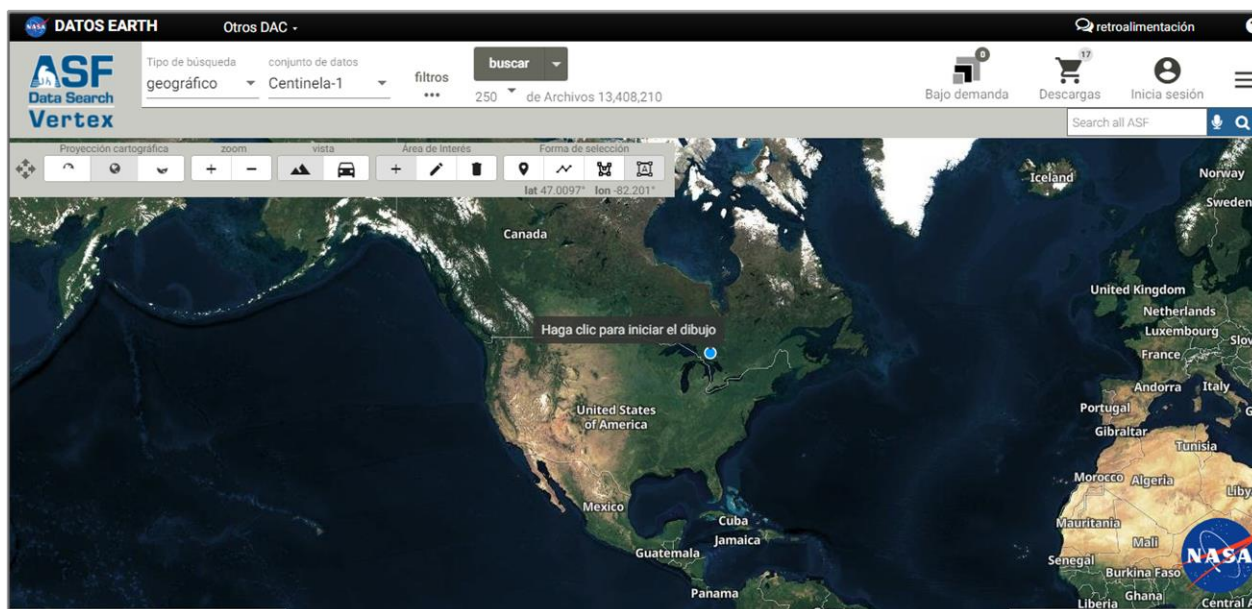


Figura 3. Página Alaska Satellite Facility
(Fuente: Elaboración propia, captura de pantalla)

Una vez aquí se señala la zona a trabajar con el icono circular azul, esto por medio de un rectángulo o cuadrado color amarillo que dibuje sobre el área deseada. Posterior a esto se presionará en “buscar” e inmediatamente aparecerán las imágenes disponibles para ese sitio. Estas imágenes tanto ascendentes como descendentes estarán representadas como rectángulos de color celeste, y tomarán el tono rojo al seleccionarla (figura 4).

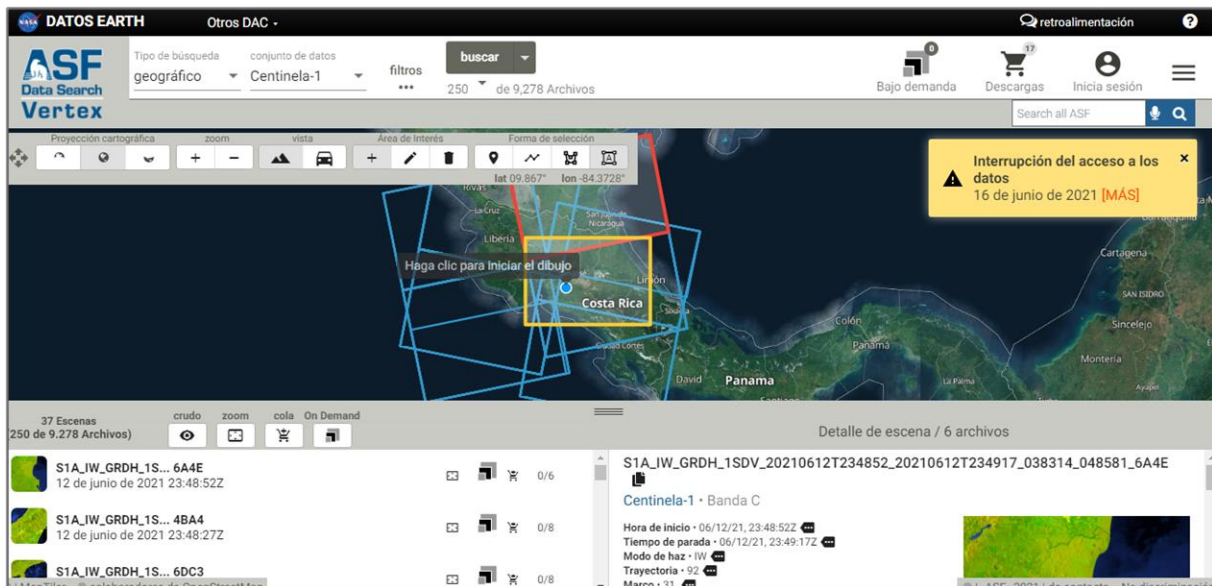


Figura 4. Selección de área de interés ASF
(Fuente: Elaboración propia, captura de pantalla)

Para poder tener una búsqueda más específica de las imágenes SAR, se debe dirigir a filtros que se ubica en la parte superior, se desplegará el cuadro que se verá a continuación, aquí se podrán indicar los parámetros que se desean de la imagen. En tipo de archivo por lo general se elige en este caso la opción de Single Look Complex L1 (SLC), para modo de haz o escaneo se elige la opción Interferometric Wide (IW). En polarización se escoge el tipo de polarizado en la onda, la dirección hace referencia a la toma de la imagen si es ascendente o descendente con respecto a la órbita del satélite. Finalmente, en subtipo escogerá el satélite Sentinel-1 del cual desea obtener sus datos (Sa o Sb), como se aprecia en la figura 5.

Filtros de búsqueda

Área de interés de importación

claro

Filtros de fecha

fecha de inicio

fecha final

Búsqueda estacional

Filtros adicionales

Tipo de archivo

0/14 tipos de archivo seleccionados

subtipo

Modo de haz

Modos de haz 0/9 seleccionados

polarización

0/8 polarizaciones seleccionadas

dirección

0/2 direcciones de vuelo seleccionadas

250 de Archivos 5,986

Cancelar

buscar

A continuación, se elige la imagen a descargar, y se añade en la opción del icono de carrito para posteriormente poder descargar una carpeta comprimida, como se muestra en la figura 6, se señala en un círculo rojo la opción para añadir. Se debe recordar que es necesario descargar dos imágenes mínimo para poder realizar una interferometría, en este ejemplo se eligen dos imágenes del mes de enero de 2020 con una diferencia de captura de tres semanas aproximadamente entre cada una.

Una vez que se han escogido las imágenes de interés se dirige al icono del carrito de compras en la parte superior derecha de la pantalla, para poder acceder a la descarga. Se desplegará un menú con todo el historial de descargas. Para descargar la imagen deseada se presionará sobre la nube que contiene la flecha blanca, es importante remarcar que estos archivos tienen un peso considerable, y que además el procesamiento demanda otra cantidad grande de memoria (figura 7). Por lo tanto, se debe tomar esto en cuenta cuando se desea trabajar un periodo de tiempo para realizar series temporales, ya que el consumo de memoria en la computadora será bastante.

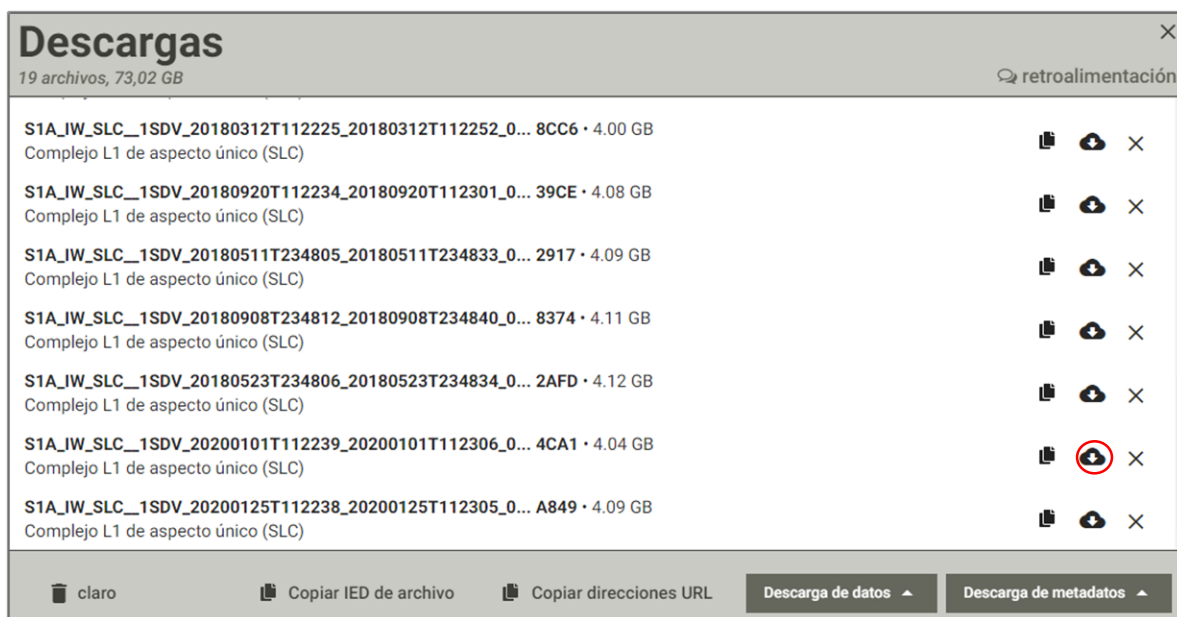


Figura 7. Espacio de imágenes añadidas para la descarga ASF
(Fuente: Elaboración propia, captura de pantalla)

Las imágenes SAR se nombran de la siguiente manera:



Figura 8. Formato de nombre de imágenes
(Fuente: Elaboración propia)

2.2 Descarga de orbitas

2.2.1 Copernicus Open Acces Hub

Para acceder a esta página se debe de ir al siguiente link indicado abajo. En el caso de las orbitas si es necesario descargarlas desde la página de Copernicus, ya que las orbitas no ocupan de ningún permiso especial para su descarga. Solo es necesario ingresar con la cuenta usando el usuario y contraseña que se dispone en la página (gnssguest/gnssguest).

Copernicus Open Acces Hub: <https://scihub.copernicus.eu/gnss/#/home>



Figura 9. Página de Copernicus
(Fuente: Elaboración propia, captura de pantalla)

Una vez dentro de la web de orbitas Copernicus, se debe dirigir al buscador ubicado en la parte superior izquierda, dar clic sobre el icono en las cuatro barritas y aparecerá el cuadro de la figura 10. En este cuadro podrá introducir las características de la órbita que desea descargar, donde el “período de detección o sensing date” es para ubicar las fechas de interés. Para el parámetro “Ordene por o Order by”, se indica la dirección de orbita en que se tomó la imagen (ascendente, descendente). El período de ingestión se refiere a la fecha de subida del archivo de órbita a la página, pero este parámetro no es de uso necesario. Finalmente, en la parte inferior del cuadro se selecciona la misión Sentinel-1 (S1A, S1B) de la cual se descargaron las imágenes, Y además el tipo de producto orbital que se desea descargar.

Insertar criterios de búsqueda...

Búsqueda Avanzada Claro

» Ordenar por: Fecha de ingestión

» Ordene por: Descendente

» Período de detección: 2020/11/19 2020/11/21

» Período de ingestión:

☒ Misión: Sentinel-1

Plataforma Satelital: S1A_*

Tipo de producto: AUX_POEORB

☐ Misión: Sentinel-2

Plataforma Satelital:

Tipo de producto:

Figura 10. Parámetros de búsqueda de orbitas
(Fuente: Elaboración propia, captura de pantalla)

Existen varios tipos de producto orbital, para el Sentinel-1A están disponibles AUX_GNSSRD, AUX_POEORB, AUX_RESORB y AUX_PROQUA. Para efectos del procesamiento con el modo Tops, se utilizará siempre el producto orbital AUX_POEORB. Otro aspecto importante que se debe tener en cuenta a la hora de descargar orbitas, es que la órbita no puede pertenecer al mismo día de la adquisición SAR. Por ejemplo, si la adquisición de la imagen SAR pertenece a la fecha 2020/11/20, la órbita a seleccionar será la que va del día antes al día después de la adquisición. Por lo tanto, las fechas de la órbita a descargar deberá iniciar el 2020/11/19 y finalizar el 2020/11/21.

Los archivos se nombran de la siguiente manera:

sensor Fecha de ingestión de órbita Fecha de escaneo 2
S1A_OPER_AUX_POEORB_OPOD_20210318T200737_V20200604T225942_20200606T005942.EOF
Tipo de archivo orbital Fecha de escaneo 1

Figura 11. Formato de nombre de archivo de orbitas
(Fuente: Elaboración propia)

2.3 Procesamiento con ISCE (topsStack)

El comando `stackSentinel.py` es el principal dentro del módulo `topsStack` de ISCE, es el encargado de realizar el procesamiento de pila o stack, tiene varios flujos de trabajo como stack de interferogramas y stack de SLC. Se pueden observar los diferentes flujos de trabajo aplicando el comando:

```
stackSentinel.py -H
```

Para conocer los parámetros principales del comando `stackSentinel.py`, se debe ejecutar el siguiente comando. En la figura 12 se visualizan los parámetros de dicho comando.

```
stackSentinel.py --help
```

En este caso se utilizará el flujo de trabajo stack de interferogramas, con un corrección Enhanced Spectral Diversity (ESD) entre imágenes. A continuación, se indica el significado de los parámetros principales son:

- s: se refiere al directorio donde se encuentran las imágenes SAR.
- o: indica el directorio donde están las orbitas.
- a: carpeta con archivos auxiliares (opcional, se usa para imágenes antiguas de Sentinel-1).
- d: carpeta donde se ubica el Modelo Digital de Elevaciones que se va utilizar.
- r: Mirada o resolución en rango.
- a, -z: Mirada o resolución en acimut.
- c: Número de conexiones entre imágenes SAR.
- n: Número de franjas o swaths.
- b, -bbox: Delimitación del área de interés en latitud y longitud.

```

root@ovslicor-virtual-machine: /home/Andrey_InSAR/Puriscal/DESC/ISCE

root@ovslicor-virtual-machin... root@ovslicor-virtual-machin... root@ovslicor-virtual-machin... root@ovslicor-virtual-machin...
root@ovslicor-virtual-machine: /home/Andrey_InSAR/Puriscal/DESC/ISCE# stackSentinel.py --help
Using default ISCE Path: /usr/local/Isce
This is the Open Source version of ISCE.
Some of the workflows depend on a separate licensed package.
To obtain the licensed package, please make a request for ISCE
through the website: https://download.jpl.nasa.gov/ops/request/index.cfm.
Alternatively, if you are a member, or can become a member of WInSAR
you may be able to obtain access to a version of the licensed software at
https://winsar.unavco.org/software/Isce
usage: stackSentinel.py [-h] [-H] -s SLC_DIRNAME -o ORBIT_DIRNAME -a AUX_DIRNAME [-w WORK_DIR] -d DEM [-m REFERENCE_DATE]
                        [-c NUM_CONNECTIONS] [-n SWATH_NUM] [-b BBOX] [-t TEXT_CMD] [-x EXCLUDE_DATES] [-l INCLUDE_DATES]
                        [--start_date STARTDATE] [--stop_date STOPDATE] [-z AZIMUTHLOOKS] [-r RANGELOOKS] [-f FILTSTRENGTH]
                        [--snr_misreg_threshold SNRTHRESHOLD] [-p POLARIZATION] [-C {geometry,NESD}] [-O NUM_OVERLAP_CONNECTIONS]
                        [-e ESDCOHERENCETHRESHOLD] [-W {slc,correlation,interferogram,offset}] [-useGPU] [--num_proc NUMPROCESS]
                        [--num_proc4topo NUMPROCESS4TOPO] [-u {lcu,snaphu}] [--rnFilter]

Preparing the directory structure and config files for stack processing of Sentinel data

optional arguments:
  -h, --help                show this help message and exit
  -H, --hh                  Display detailed help information.
  -s SLC_DIRNAME, --slc_directory SLC_DIRNAME
                        Directory with all Sentinel SLCs
  -o ORBIT_DIRNAME, --orbit_directory ORBIT_DIRNAME
                        Directory with all orbits
  -a AUX_DIRNAME, --aux_directory AUX_DIRNAME
                        Directory with all aux files
  -w WORK_DIR, --working_directory WORK_DIR
                        Working directory (default: ./).
  -d DEM, --dem DEM         Directory with the DEM
  -m REFERENCE_DATE, --reference_date REFERENCE_DATE
                        Directory with reference acquisition
  -c NUM_CONNECTIONS, --num_connections NUM_CONNECTIONS
                        number of interferograms between each date and subsequent dates (default: 1).
  -n SWATH_NUM, --swath_num SWATH_NUM
                        A list of swaths to be processed. -- Default : '1 2 3'
  -b BBOX, --bbox BBOX     Lat/Lon Bounding SNWE. -- Example : '19 20 -99.5 -98.5' -- Default : common overlap between stack
  -t TEXT_CMD, --text_cmd TEXT_CMD
  -x EXCLUDE_DATES, --exclude_dates EXCLUDE_DATES
                        List of the dates to be excluded for processing. -- Example : '20141007,20141031' (default: None).
  -l INCLUDE_DATES, --include_dates INCLUDE_DATES
                        List of the dates to be included for processing. -- Example : '20141007,20141031' (default: None).
  --start_date STARTDATE
                        Start date for stack processing. Acquisitions before start date are ignored. format should be YYYY-MM-DD
                        e.g., 2015-01-23
  --stop_date STOPDATE
                        Stop date for stack processing. Acquisitions after stop date are ignored. format should be YYYY-MM-DD
                        e.g., 2017-02-26
  -z AZIMUTHLOOKS, --azimuth_looks AZIMUTHLOOKS
                        Number of looks in azimuth for interferogram multi-looking (default: 3).
  -r RANGELOOKS, --range_looks RANGELOOKS
                        Number of looks in range for interferogram multi-looking (default: 9).
  -f FILTSTRENGTH, --filter_strength FILTSTRENGTH
                        Filter strength for interferogram filtering (default: 0.5).
  --snr_misreg_threshold SNRTHRESHOLD
                        SNR threshold for estimating range misregistration using cross correlation (default: 10).
  -p POLARIZATION, --polarization POLARIZATION
                        SAR data polarization (default: vv).
  -C {geometry,NESD}, --coregistration {geometry,NESD}
                        Coregistration options (default: NESD).
  -O NUM_OVERLAP_CONNECTIONS, --num_overlap_connections NUM_OVERLAP_CONNECTIONS
                        number of overlap interferograms between each date and subsequent dates used for NESD computation (for
                        azimuth offsets misregistration) (default: 3).
  -e ESDCOHERENCETHRESHOLD, --esd_coherence_threshold ESDCOHERENCETHRESHOLD
                        Coherence threshold for estimating azimuth misregistration using enhanced spectral diversity (default:
                        0.85).
  -W {slc,correlation,interferogram,offset}, --workflow {slc,correlation,interferogram,offset}
                        The InSAR processing workflow (default: interferogram).
  -useGPU, --useGPU         Allow App to use GPU when available
  --num_proc NUMPROCESS, --num_process NUMPROCESS
                        number of tasks running in parallel in each run file (default: 1).
  --num_proc4topo NUMPROCESS4TOPO, --num_process4topo NUMPROCESS4TOPO
                        number of parallel processes (for topo only) (default: 1).
  -u {lcu,snaphu}, --unw_method {lcu,snaphu}
                        Unwrapping method (default: snaphu).
  --rnFilter, --rnFilter
                        Make an extra unwrap file in which filtering effect is removed

```

Figura 12. Parámetros del comando stackSentinel.py
(Fuente: Elaboración propia, captura de pantalla)

2.4 Modelos digitales de elevación y de terreno (DEM, MDT)

El programa ISCE necesita de insumo para su procesamiento un modelo digital de elevación, cuando utiliza los comandos Apps como topsApp.py, descarga automáticamente un DEM del sensor radar SRTM. Pero cuando se procesa con el módulo topsStack, el DEM se debe descargar manualmente, aplicando el siguiente comando dem.py. A continuación, se presenta como se debe escribir el comando, donde en LatSur, LatNorte, LongOeste y LongEste se deben indicar las coordenadas del área de interés.

```
dem.py -a stitch -b LatSur LatNorte LongOeste LongEste -r -s 1 -c
```

Es importante mencionar que para el área de interés solo se pueden poner coordenadas en grados sin decimal, por ejemplo:

```
dem.py -a stitch -b 9 10 -85 -83 -r -s 1 -c
```

De igual manera, se pueden usar otros DEMs como el del sensor satelital ASTER, pero el ASTER se debe descargar manualmente desde la página de la NASA (<https://search.earthdata.nasa.gov/search/>). También se pueden utilizar MDT regionales o propios como el del Instituto Geográfico Nacional. Como requisito en caso de usar un DEM propio u otro, es necesario que se encuentren en formato GeoTiff, para posteriormente aplicar el comando gdal_translate que lo convertirá a formato ISCE. Dado que ISCE solo lee DEMs en su formato, de lo contrario le generará errores si lo utiliza en otro formato.

```
gdal_translate -of ISCE nombre_dem.tif nombre_dem.dem
```

Después de ejecutar lo anterior, se crearán varios archivos como se observan en la siguiente figura, donde el archivo principal es el que termina en .wgs84.

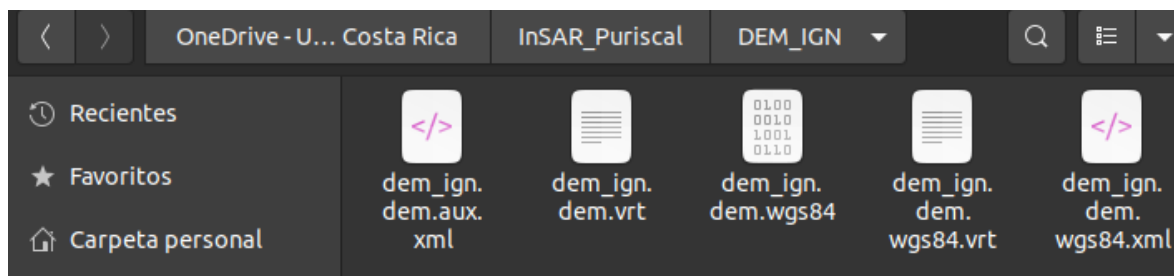


Figura 13. Archivos DEM al aplicar gdal_translate
(Fuente: Elaboración propia, captura de pantalla)

Se recomienda habilitar los permisos de todos los archivos con:

```
chmod 777 dem*
```

2.5 Número de franjas o subswaths

La imagen SAR en formato Single Look Complex (SLC), se conoce también como swath o franja. La misma se divide en tres subswaths, de los que es necesario conocer la numeración para indicar en el parámetro -n, de acuerdo a la zona de interés. La numeración de las subswaths cambia según se la órbita de la imagen. A continuación, se muestran las figuras donde se señalan los números de las subfranjas para un mejor entendimiento.

2.5.1 Órbita ascendente



Figura 14. Subswath 1 órbita ascendente
(Fuente: Elaboración propia, captura en SNAP)

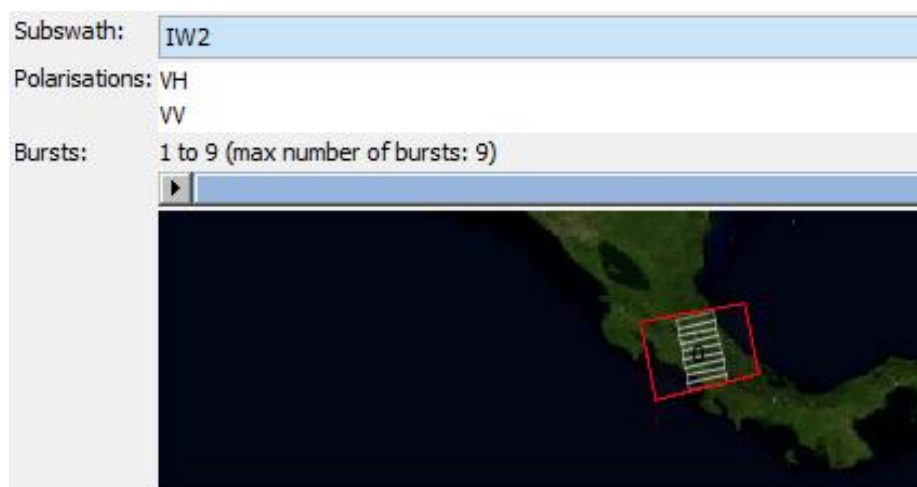


Figura 15. Subswath 2 órbita ascendente
(Fuente: Elaboración propia, captura en SNAP)

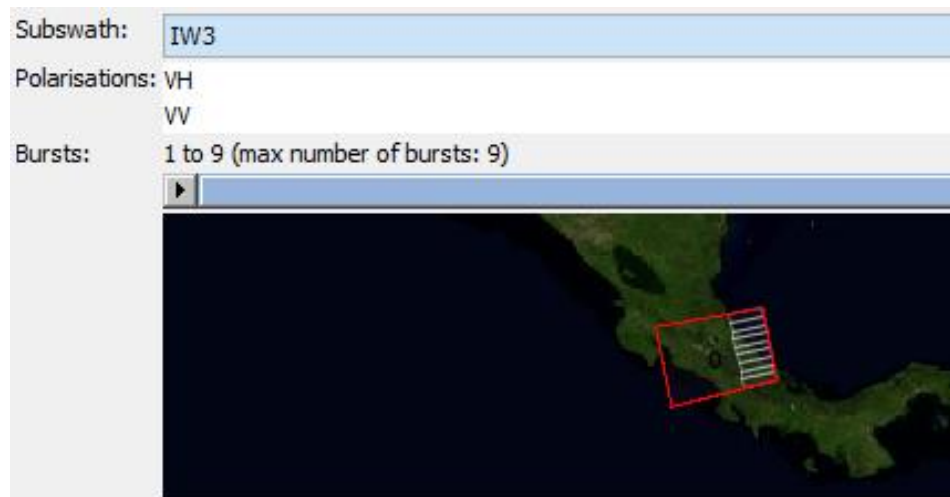


Figura 16. Subswath 3 órbita ascendente
(Fuente: Elaboración propia, captura en SNAP)

2.5.2 Órbita descendente

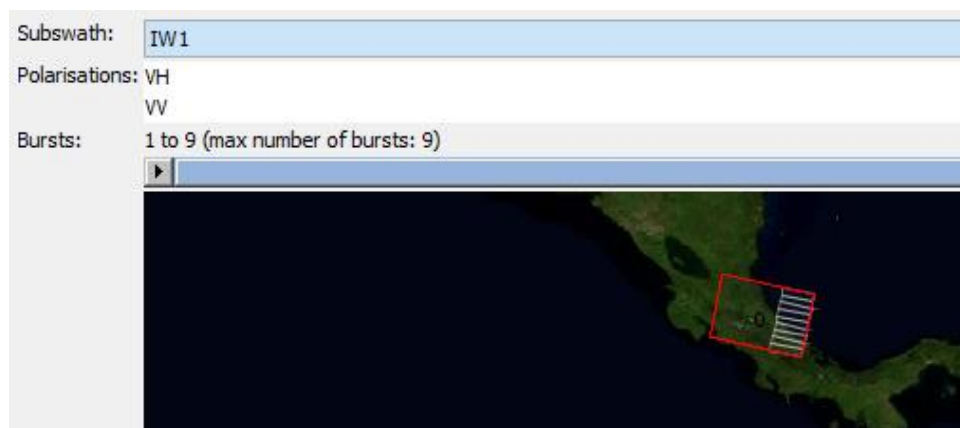


Figura 17. Subswath 1 órbita descendente
(Fuente: Elaboración propia, captura en SNAP)

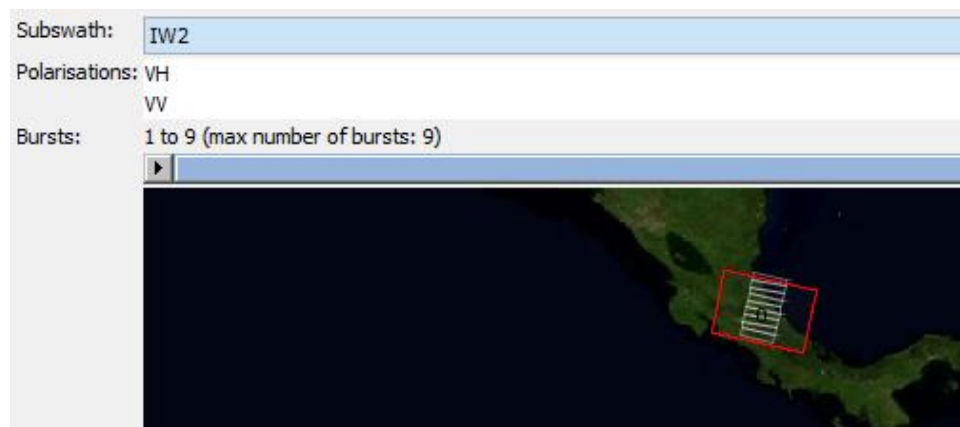


Figura 18. Subswath 2 órbita descendente
(Fuente: Elaboración propia, captura en SNAP)

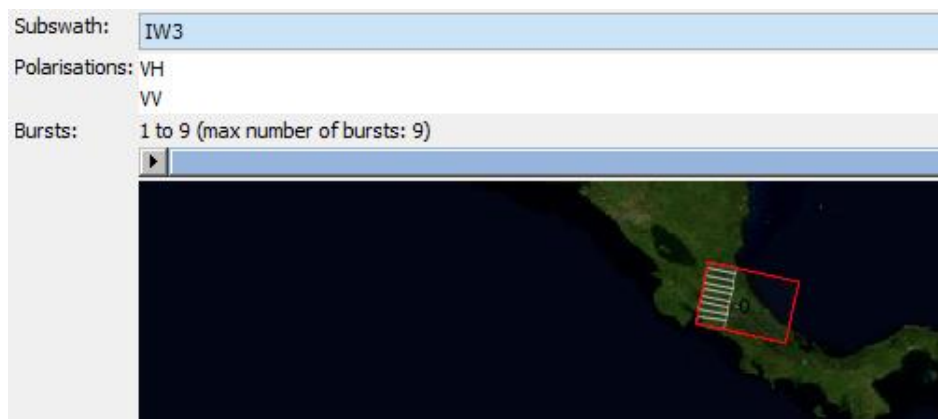


Figura 19. Subswath 3 órbita descendente
(Fuente: Elaboración propia, captura en SNAP)

2.6 Ejecución de stackSentinel.py

Para la ejecución del stackSentinel.py es necesario como insumo la carpeta SLC con las imágenes SAR, la carpeta EOF con las orbitas, la carpeta aux_cal con datos auxiliares (opcional), y finalmente la carpeta con el DEM. En la figura 20, se muestra como se debe ver la carpeta donde se ejecutará el comando, con los directorios necesarios para el procesamiento (solo es necesario un DEM).

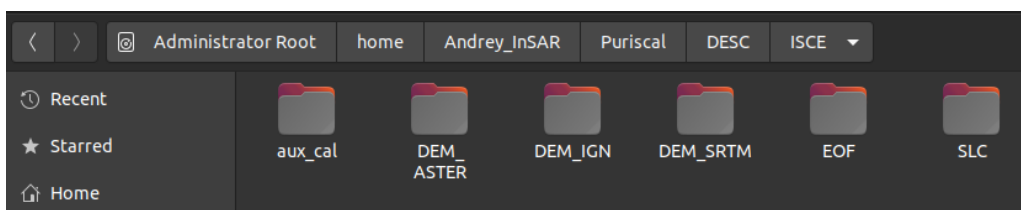


Figura 20. Contenido de la carpeta para la ejecución de stackSentinel.py
(Fuente: Elaboración propia, captura de pantalla)

Antes de ejecutar el script del stackSentinel es importante que no se tenga ningún entorno activado como podría ser conda, mintpy, o licsbas, de lo contrario le generara un error al correrlo. Para darse cuenta si hay algún entorno activo, solo habrá que fijarse a un lado de del nombre de usuario en la terminal, si aparece la palabra base entre paréntesis, el entorno conda está activado, si aparece la palabra mintpy el entorno del programa Mintpy está activado, y así con LiCSBAS. Por ejemplo, en la figura 21 se ve como desactivar el entorno con el comando conda deactivate.

```
conda deactivate
```

```
conda deactivate mintpy
```



```
root@ovsicori-virtual-machine: /home/ovsicori

ovsicori@ovsicori-virtual-machine:~$ sudo su
[sudo] contraseña para ovsicori:
(base) root@ovsicori-virtual-machine:/home/ovsicori# conda deactivate
root@ovsicori-virtual-machine:/home/ovsicori#
```

Figura 21. Desactivar entorno miniconda
Fuente: (Elaboración propia, captura de pantalla)

Una vez que se tiene lo anterior organizado, es hora de ejecutar el comando, en este primer ejemplo se usa el MDT del IGN. En este comando las coordenadas de la zona de interés si pueden incluir decimales.

```
stackSentinel.py -s ./SLC/ -o ./EOF/ -a ./aux_cal/ -d ./DEM_IGN/dem_ign.dem.wgs84 -r 15 -a 5 -c 3 -n '3' -b -bbox 'LatSur LatNorte LonOeste LonEste'
```

Desde la terminal el comando se vería de la siguiente forma:

```
root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE# stackSentinel.py -s ./SLC/ -o ./EOF/ -a ./aux_cal/ -d ./DEM_IGN/dem_ign.dem.wgs84 -r 15 -a 5 -c 3 -n '3' -b -bbox '9.825383 9.869876 -84.354170 -84.276980'
```

En este otro ejemplo se utiliza el DEM del sensor radar SRTM.

```
stackSentinel.py -s ./SLC/ -o ./EOF/ -a ./aux_cal/ -d ./DEM_SRTM/demLat_N09_N11_Lon_W085_W083.dem.wgs84 -r 15 -a 5 -c 3 -n '3' -b -bbox 'LatSur LatNorte LonOeste LonEste'
```

Desde la terminal el comando se vería de la siguiente forma:

```
root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE# stackSentinel.py -s ./SLC/ -o ./orbits/ -a ./aux_cal/ -d ./DEM_SRTM/demLat_N09_N11_Lon_W085_W083.dem.wgs84 -r 15 -a 5 -c 3 -n '3' -b -bbox '9.825383 9.869876 -84.354170 -84.276980'
```

Una vez que el stackSentinel.py ha sido corrido correctamente, generará una serie de archivos ejecutables que permitirán realizar el procesamiento del stack (figura 14). Estos archivos se denominan run files y se deben ejecutar en el orden numerado.

```

The reference date was not chosen. The first date is considered as reference date.

All SLCs will be coregistered to : 20200605
secondary dates :
['20200617', '20200629', '20200711', '20200723', '20200804', '20200816', '20200828', '20200909', '20200921', '20201003', '20201015',
'20201027', '20201108', '20201120', '20201202', '20201226', '20210107', '20210119', '20210131']

No existing stack was identified. A new stack will be generated.
*****
Coregistration method: NESD
Workflow: interferogram
*****
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_01_unpack_topo_reference
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_02_unpack_secondary_slc
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_03_average_baseline
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_04_extract_burst_overlaps
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_05_overlap_geo2dr
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_06_overlap_resample
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_07_pairs_misreg
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_08_timeseries_misreg
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_09_fullBurst_geo2dr
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_10_fullBurst_resample
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_11_extract_stack_valid_region
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_12_merge_reference_secondary_slc
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_13_generate_burst_igram
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_14_merge_burst_igram
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_15_filter_coherence
writing /home/Andrey_InSAR/Puriscal/DESC/ISCE/run_files/run_16_unwrap

```

Figura 22. Resultado del stackSentinel.py (run files)
(Fuente: Elaboración propia, captura de pantalla)

Los archivos run files se deben ejecutar uno por uno, estos se ubican en una de las carpetas generadas denominada run_files, esta se crea automáticamente dentro del directorio de trabajo. Se debe ingresar a la carpeta run_files y escribir el siguiente comando antes de ejecutar los archivos, para otorgar permisos de ejecución:

```
chmod +x run_files*
```

Para efectos de automatizar este proceso, se creó un bash que ejecuta todos los archivos run files sin necesidad de hacerlo manualmente. El contenido se observa en la figura 23.

```

GNU nano 4.8                                     bash_stack.sh
cd run_files
chmod +x run_*
./run_01_unpack_topo_reference
./run_02_unpack_secondary_slc
./run_03_average_baseline
./run_04_extract_burst_overlaps
./run_05_overlap_geo2dr
./run_06_overlap_resample
./run_07_pairs_misreg
./run_08_timeseries_misreg
./run_09_fullBurst_geo2dr
./run_10_fullBurst_resample
./run_11_extract_stack_valid_region
./run_12_merge_reference_secondary_slc
./run_13_generate_burst_igram
./run_14_merge_burst_igram
./run_15_filter_coherence
./run_16_unwrap
cd ..

```

Figura 23. Archivo ejecutable bash_stack.sh
(Fuente: Elaboración propia)

Para crear el archivo bash_stack.sh escribir:

```
nano bash_stack.sh
```


Añadir en el archivo lo siguiente:

```
cd run_files
chmod +x run_*
./run_01_unpack_topo_reference
./run_02_unpack_secondary_slc
./run_03_average_baseline
./run_04_extract_burst_overlaps
./run_05_overlap_geo2rdr
./run_06_overlap_resample
./run_07_pairs_misreg
./run_08_timeseries_misreg
./run_09_fullBurst_geo2rdr
./run_10_fullBurst_resample
./run_11_extract_stack_valid_region
./run_12_merge_reference_secondary_slc
./run_13_generate_burst_igram
./run_14_merge_burst_igram
./run_15_filter_coherence
./run_16_unwrap
cd ..
```

Guardar los cambios con:

```
Ctrl + x
```

El archivo bash se debe correr fuera de la carpeta run_files, y para ejecutarlo se aplica el comando:

```
./bash_stack.sh
```

Cuando stackSentinel.py ha terminado de procesar resultaran en el directorio principal varias carpetas nuevas y un archivo de texto como lo son:

-reference	-baselines	-coreg_secondarys
-secondarys	-configs	-stack
-interferograms	-run_files	-geom_reference
-ESD	-misreg	-SAFE_files.txt
-merged	-coarse_interferograms	

En la carpeta merged se encuentran los resultados finales del procesamiento, que son el conjunto de interferogramas filtrados y desenvueltos. Estos interferogramas los podrá observar ejecutando el comando mdx.py. Se debe aclarar que se hizo un ejemplo con "20201226_20210107", pero esto varía según las fechas que se estén trabajando.

```
cd merged/interferograms/20201226_20210107

mdx.py filt_fine.unw
```

Después de ejecutar lo anterior, deberá abrirse una pantalla en la que se visualiza de modo un poco interactivo el interferograma en geometría SAR, se puede observar tanto la fase de amplitud como la fase desenvuelta (unwrapping). Observar figura 24.

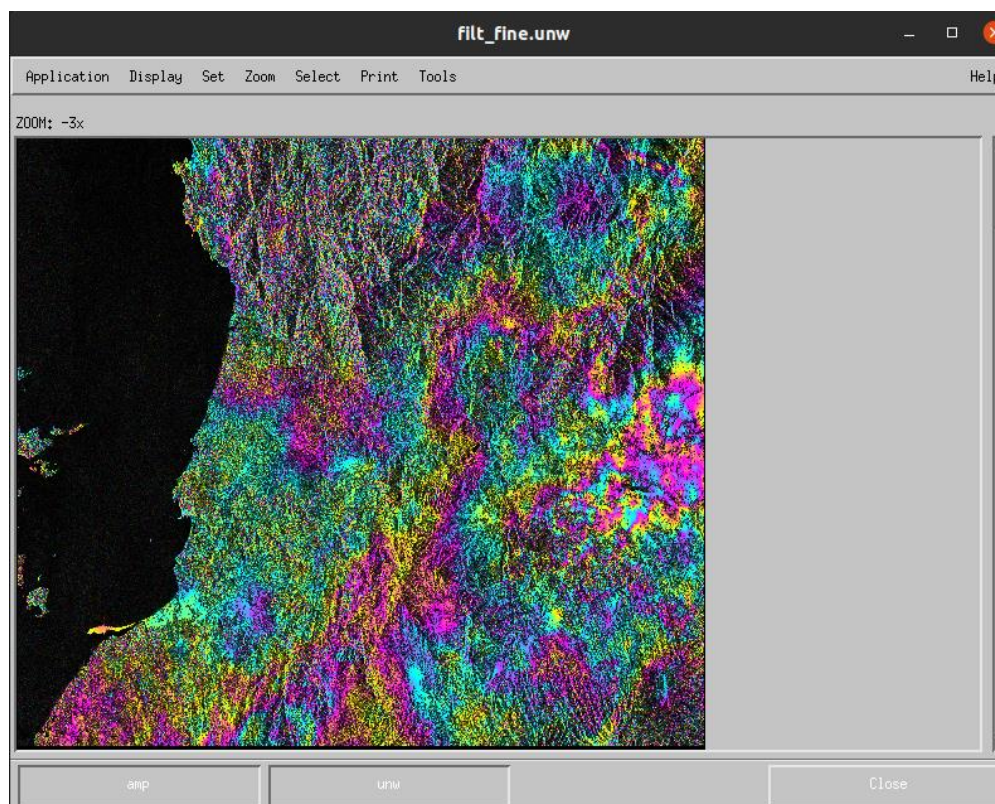


Figura 24. Visor de interferogramas de ISCE
(Fuente: Elaboración propia)

2.4 Posibles errores al ejecutar stackSentinel.py

2.4.1 Error por dirección de la carpeta donde se encuentra el DEM

Si al ejecutar el stackSentinel.py se le presenta el error de la figura 25, lo más seguro es que uno de los archivos del DEM no contiene la dirección de la carpeta en la que se encuentra, dentro de su archivo XML.

```
Using default ISCE Path: /usr/local/isce
This is the Open Source version of ISCE.
Some of the workflows depend on a separate licensed package.
To obtain the licensed package, please make a request for ISCE
through the website: https://download.jpl.nasa.gov/ops/request/index.cfm.
Alternatively, if you are a member, or can become a member of WinSAR
you may be able to obtain access to a version of the licensed software at
https://winsar.unavco.org/software/isce
Number of SAFE files found: 20
*****
20200629
orbit was not found in the /home/Andrey_InSAR/Puriscal/DESC/ISCE/orbits
downloading precise or restituted orbits ...
fetchOrbit.py -i /home/Andrey_InSAR/Puriscal/DESC/ISCE/SLC/S1A_IW_SLC__1SDV_20200629T112242_20200629T112309_033231_03D995_FF65.zip -
o /home/Andrey_InSAR/Puriscal/DESC/ISCE/orbits/20200629
Reference time: 2020-06-29 11:23:09
Satellite name: S1A
Failed to find S1A orbits for tref 2020-06-29 11:23:09
Failed to find S1A orbits for tref 2020-06-29 11:23:09
Traceback (most recent call last):
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 772, in <module>
    main(sys.argv[1:])
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 735, in main
    acquisitionDates, stackReferenceDate, secondaryDates, safe_dict, updateStack = checkCurrentStatus(inps)
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 611, in checkCurrentStatus
    acquisitionDates, stackReferenceDate, secondaryDates, safe_dict = get_dates(inps)
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 275, in get_dates
    safeObj.get_orbit(inps.orbit_dirname, inps.work_dir)
  File "/usr/local/isce/contrib/topsStack/Stack.py", line 1048, in get_orbit
    self.orbit = orbitFile[0]
IndexError: list index out of range
```

Figura 25. Error por metadata en los archivos del DEM
(Fuente: Elaboración propia, captura de pantalla)

La carpeta de los DEMs debe contener los siguientes archivos como se ve en la figura 26, el archivo que esta de color naranja es el archivo al que habrá que agregar las direcciones. Para realizarlo desde la terminal, deberá dirigirse al directorio donde tiene el DEM, y posteriormente usar el comando nano para editar el contenido del archivo XML.

```
cd DEM_SRTM
nano demLatN09_N11_Lon_W085_W083.dem.wgs84.xml
```

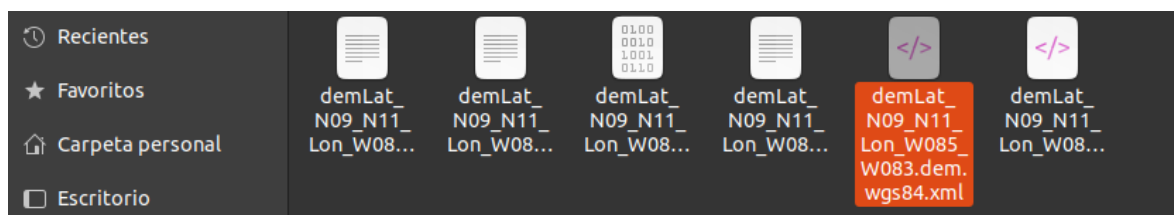
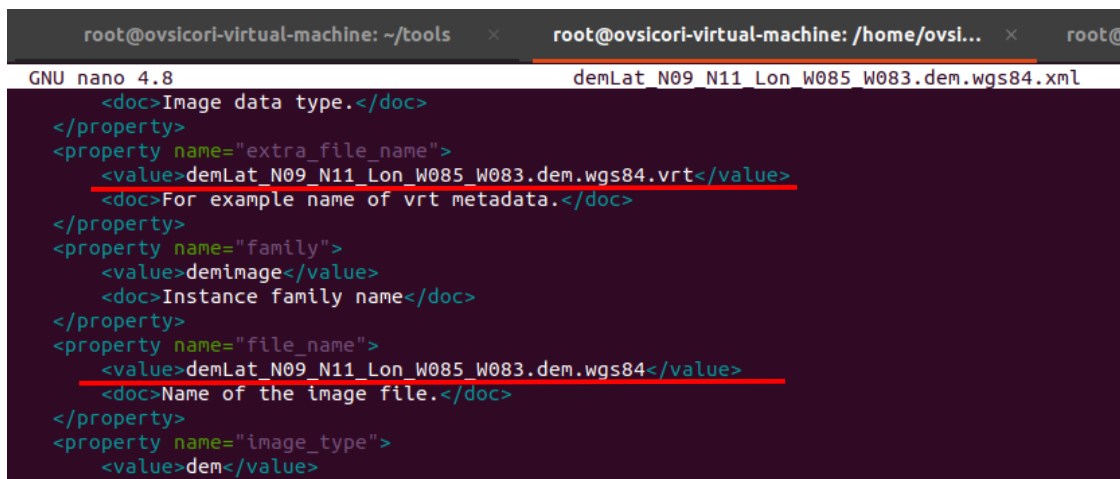


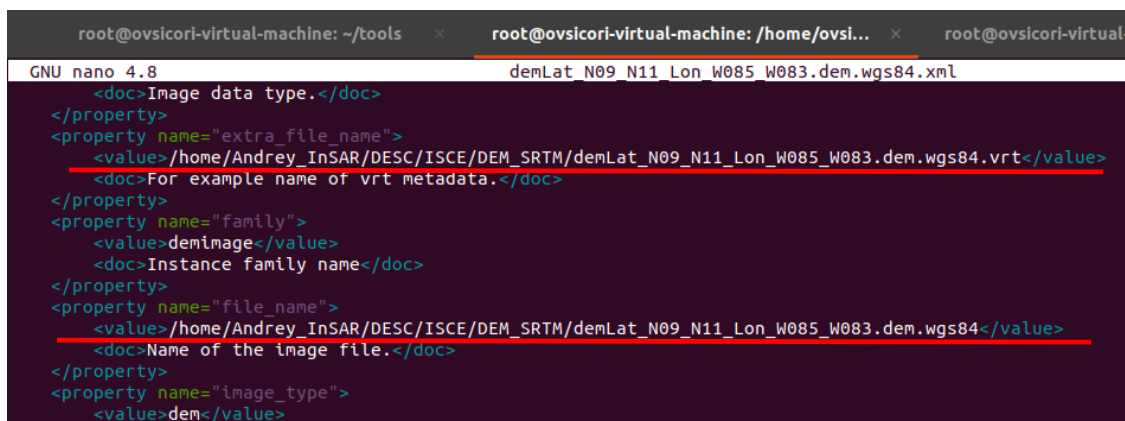
Figura 26. Contenido de la carpeta DEM
(Fuente: Elaboración propia)

Al ingresar al archivo y bajar un poco le aparecerá lo que muestra la figura 27, y como se observa las líneas donde se encuentran los nombres del DEM que terminan en **.wgs84** y **.vrt**, será donde agregará las direcciones de la carpeta DEM como se ve en la figura 28.



```
root@ovsicori-virtual-machine: ~/tools x root@ovsicori-virtual-machine: /home/ovsi... x root@
GNU nano 4.8 demLat_N09_N11_Lon_W085_W083.dem.wgs84.xml
<doc>Image data type.</doc>
</property>
<property name="extra_file_name">
  <value>demLat_N09_N11_Lon_W085_W083.dem.wgs84.vrt</value>
  <doc>For example name of vrt metadata.</doc>
</property>
<property name="family">
  <value>demimage</value>
  <doc>Instance family name</doc>
</property>
<property name="file_name">
  <value>demLat_N09_N11_Lon_W085_W083.dem.wgs84</value>
  <doc>Name of the image file.</doc>
</property>
<property name="image_type">
  <value>dem</value>
```

Figura 27. Metadata del archivo **.dem.wgs8.xml**
(Fuente: Elaboración propia)



```
root@ovsicori-virtual-machine: ~/tools x root@ovsicori-virtual-machine: /home/ovsi... x root@ovsicori-virtual-
GNU nano 4.8 demLat_N09_N11_Lon_W085_W083.dem.wgs84.xml
<doc>Image data type.</doc>
</property>
<property name="extra_file_name">
  <value>/home/Andrey_InSAR/DESC/ISCE/DEM_SRTM/demLat_N09_N11_Lon_W085_W083.dem.wgs84.vrt</value>
  <doc>For example name of vrt metadata.</doc>
</property>
<property name="family">
  <value>demimage</value>
  <doc>Instance family name</doc>
</property>
<property name="file_name">
  <value>/home/Andrey_InSAR/DESC/ISCE/DEM_SRTM/demLat_N09_N11_Lon_W085_W083.dem.wgs84</value>
  <doc>Name of the image file.</doc>
</property>
<property name="image_type">
  <value>dem</value>
```

Figura 28. Dirección del DEM en la metadata
(Fuente: Elaboración propia)

Para salir del archivo XML y guardar los cambios, presionar las siguientes teclas:

Ctrl + x

Después de realizar estos cambios vuelva a ejecutar el comando `stackSentinel.py`, deberá correr sin ningún problema.

2.4.2 Error en la descarga automática de orbitas

Con respecto a las orbitas, el stackSentinel.py tiende a descargarlas automáticamente, pero si se presenta el siguiente error de la figura 29 es recomendado descargar las orbitas de manera manual.

```
root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE# stackSentinel.py -s ./SLC/ -o ./orbits/ -a ./aux_cal/ -d ./DEM/
IGN/dem ign.dem.wgs84 -r 15 -a 5 -c 3 -n '3' --bbox '9.825383 9.869876 -84.354170 -84.276980'
Using default ISCE Path: /usr/local/isce
This is the Open Source version of ISCE.
Some of the workflows depend on a separate licensed package.
To obtain the licensed package, please make a request for ISCE
through the website: https://download.jpl.nasa.gov/ops/request/index.cfm.
Alternatively, if you are a member, or can become a member of WinSAR
you may be able to obtain access to a version of the licensed software at
https://winsar.unavco.org/software/isce
Number of SAFE files found: 20
*****
20200629
orbit was not found in the /home/Andrey_InSAR/Puriscal/DESC/ISCE/orbits
downloading precise or restituted orbits ...
fetchOrbit.py -i /home/Andrey_InSAR/Puriscal/DESC/ISCE/SLC/S1A_IW_SLC__1SDV_20200629T112242_20200629T112309_033231_03D995_FF65.zip -
o /home/Andrey_InSAR/Puriscal/DESC/ISCE/orbits/20200629
Reference time: 2020-06-29 11:23:09
Satellite name: S1A
Failed to find S1A orbits for tref 2020-06-29 11:23:09
Failed to find S1A orbits for tref 2020-06-29 11:23:09
Traceback (most recent call last):
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 772, in <module>
    main(sys.argv[1:])
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 735, in main
    acquisitionDates, stackReferenceDate, secondaryDates, safe_dict, updateStack = checkCurrentStatus(inps)
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 611, in checkCurrentStatus
    acquisitionDates, stackReferenceDate, secondaryDates, safe_dict = get_dates(inps)
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 275, in get_dates
    safeObj.get_orbit(inps.orbit_dirname, inps.work_dir)
  File "/usr/local/isce/contrib/topsStack/Stack.py", line 1048, in get_orbit
    self.orbit = orbitFile[0]
IndexError: list index out of range
```

Figura 29. Error por orbitas
(Fuente: Elaboración propia, captura de pantalla)

De igual manera cuando el comando descarga las orbitas de forma automática, guarda los archivos individualmente en carpetas con el nombre de la fecha correspondiente. Esto puede causar otro error al ejecutar los run files, ya que el comando no logra leer las orbitas en carpetas separadas. Por recomendación todos los archivos orbitales deberán estar juntos en una sola carpeta (orbits o EOF). En la figura 30 se descargaron las orbitas automáticamente y se sacaron de sus respectivas carpetas, posteriormente se tiene que ejecutar de nuevo el comando stackSentinel. Se recomienda borrar los archivos que se crearon en la primera ejecución del comando, antes de ejecutarlo de nuevo.

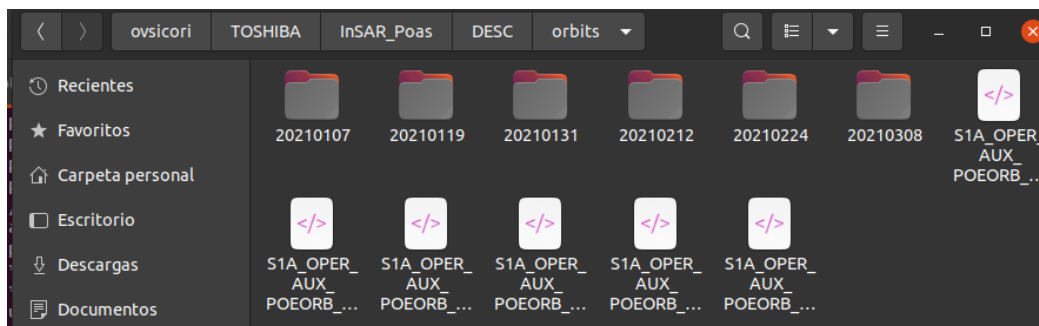


Figura 30. Carpeta con orbitas precisas
(Fuente: Elaboración propia, captura de pantalla)

2.5 Procesamiento con MintPy (SBAS)

Se recomienda crear una carpeta nueva para realizar el procesamiento con MintPy, pero dentro del mismo directorio donde tiene los resultados del topsStack. Por ejemplo, se puede crear una carpeta llamada series temporales escribiendo el siguiente comando:

```
mkdir Serie_temp
```

En la nueva carpeta deberá mover o copiar 4 de las carpetas resultado del stack las cuales son baselines, merged, reference y secondarys. Además, dentro de este directorio se creará otra carpeta con el nombre “mintpy”, la cual será donde se procesará y guardaran todos los productos finales después del procesamiento. Por tanto, en la figura 31 se observará el contenido del directorio Serie_temp necesario para la ejecución del comando.

```
mv baselines merged reference secondarys ./Serie_temp  
mkdir mintpy  
cd mintpy
```

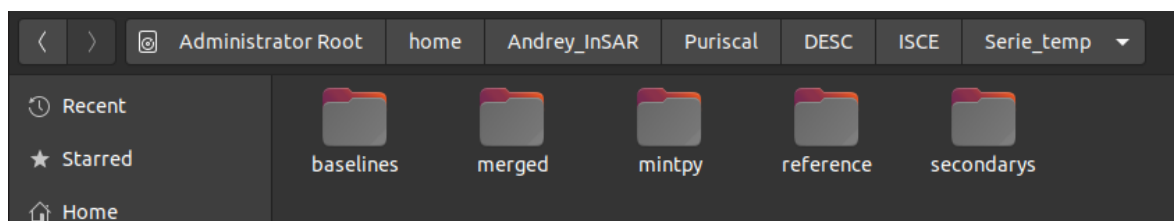


Figura 31. Insumos de carpeta para la ejecución de MintPy
(Fuente: Elaboración propia, captura de pantalla)

La carpeta mintpy de insumo solo tendrá un archivo de texto con el nombre de la zona que se esté estudiando, y por dentro llevara los parámetros que se indican en la figura 32. Para editar el archivo de texto ejecute el comando nano y configure cada uno de los parámetros que se observan en la figura. Las instrucciones que estan en azul son comentarios por lo que no afectan en nada al correr el archivo, incluso si desea puede borrarlos o no incluirlos. Entre los parámetros más importantes del archivo TXT que debe editar el usuario hay 4:

mintpy.reference.lalo: Coordenadas en latitud y longitud del punto de referencia a partir del que se medirán las diferencias (Lat, Long).

mintpy.subset.lalo: Coordenadas en latitud y longitud de los vértices para el área de interés.(LaS:LaN,LoW:LoE)

mintpy.troposphericDelay.method: Se usa para seleccionar el método de retardo troposférico (gacos/pyaps).

mintpy.troposphericDelay.gacosDir: Al decidir elegir el método GACOS, se escribe este parámetro para colocar la dirección donde esta la carpeta con el contenido de los archivos.

El archivo TXT para el procesador topsStack, se encuentra en la carpeta input_files del directorio donde se instaló Mintpy, de ahí se tendrá que copiar y editar en la carpeta de trabajo. A continuación, se indica cómo crear dicho archivo TXT. Por ejemplo para la zona de Puriscal, y con datos en órbita descendente se le pone el siguiente nombre al archivo:

```
nano Puriscal_DESCE.txt
```

Añadir el siguiente contenido al archivo de texto, y guardar los cambios:

```
# vim: set filetype=cfg:
##----- ISCE/topsStack -----##
#ssaraopt = --platform=SENTINEL-1A,SENTINEL-1B -r 128 -f 589,590,591,592,593 -e 2017-07-01
#sentinelStack.boundingBox = '-1 0.15 -91.6 -90.9'
#sentinelStack.subswath = 1 2 # comment
#sentinelStack.numConnections = 5 # comment
#sentinelStack.azimuthLooks = 5 # comment
#sentinelStack.rangeLooks = 15 # comment
#sentinelStack.filtStrength = 0.2 # comment
#sentinelStack.unwMethod = snaphu # comment
#sentinelStack.coregistration = auto # comment
#subset.y0:y1,x0:x1 = 1150:1600,1070:1670

##----- MintPy -----##
##### 1. Load Data (--load to exit after this step)
## load_data.py -H to check more details and example inputs.
##-----Default file path of ISCE/topsStack products
mintpy.load.processor = isce
##-----for ISCE only:
mintpy.load.metaFile = ../reference/IW3.xml
mintpy.load.baselineDir = ../baselines
##-----interferogram datasets:
mintpy.load.unwFile = ../merged/interferograms/*/filt*.unw
mintpy.load.corFile = ../merged/interferograms/*/filt*.cor
mintpy.load.connCompFile = ../merged/interferograms/*/filt*.unw.conncomp
mintpy.load.ionoFile = None
mintpy.load.intFile = None
##-----geometry datasets:
mintpy.load.demFile = ../merged/geom_reference/hgt.rdr
mintpy.load.lookupYFile = ../merged/geom_reference/lat.rdr
mintpy.load.lookupXFile = ../merged/geom_reference/lon.rdr
mintpy.load.incAngleFile = ../merged/geom_reference/los.rdr
mintpy.load.azAngleFile = ../merged/geom_reference/los.rdr
mintpy.load.shadowMaskFile = ../merged/geom_reference/shadowMask.rdr
mintpy.load.waterMaskFile = None
mintpy.load.bperpFile = None
#mintpy.reference.lalo = 9.8433,-84.3186
mintpy.subset.lalo = 10.1880:10.2050,-84.2600:-84.2245
#mintpy.reference.date = reference_date.txt
##-----tropospheric correct (opcional)
mintpy.troposphericDelay.method = gacos #(gacos / none) Poner none si se desea procesar sin usar GACOS
mintpy.troposphericDelay.gacosDir = ../GACOS
```

```

root@ovsicori-virtual-machine: ~/tools  ×  root@ovsicori-virtual-machine: /home/Andr...  ×  ro
GNU nano 4.8  Puriscal_DESCE.txt
#ssaraopt = --platform=SENTINEL-1A,SENTINEL-1B -r 128 -f 589,590,591,592,593 -e 2017-07-01
#sentinelStack.boundingBox = '-1 0.15 -91.6 -90.9'
#sentinelStack.subswath = 1 2 # comment
#sentinelStack.numConnections = 5 # comment
#sentinelStack.azimuthLooks = 5 # comment
#sentinelStack.rangeLooks = 15 # comment
#sentinelStack.filterStrength = 0.2 # comment
#sentinelStack.unwMethod = snaphu # comment
#sentinelStack.coregistration = auto # comment
#subset.y0:y1,x0:x1 = 1150:1600,1070:1670

##----- MintPy -----##
##### 1. Load Data (load to exit after this step)
## load_data.py -H to check more details and example inputs.
##-----Default file path of ISCE/topsStack products
mintpy.load.processor = isce
##-----for ISCE only:
mintpy.load.metaFile = ../reference/IW3.xml
mintpy.load.baselineDir = ../baselines
##-----interferogram datasets:
mintpy.load.unwFile = ../merged/interferograms/*/filt*.unw
mintpy.load.corFile = ../merged/interferograms/*/filt*.cor
mintpy.load.connCompFile = ../merged/interferograms/*/filt*.unw.conncomp
mintpy.load.ionoFile = None
mintpy.load.intFile = None
##-----geometry datasets:
mintpy.load.demFile = ../merged/geom_reference/hgt.rdr
mintpy.load.lookupYFile = ../merged/geom_reference/lat.rdr
mintpy.load.lookupXFile = ../merged/geom_reference/lon.rdr
mintpy.load.incAngleFile = ../merged/geom_reference/los.rdr
mintpy.load.azAngleFile = ../merged/geom_reference/los.rdr
mintpy.load.shadowMaskFile = ../merged/geom_reference/shadowMask.rdr
mintpy.load.waterMaskFile = None
mintpy.load.bperpFile = None
mintpy.reference.lalo = 9.8434,-84.3141
mintpy.subset.lalo = 9.80:9.86,-84.33:-84.30
##-----tropospheric correct
mintpy.troposphericDelay.method = gacos
mintpy.troposphericDelay.gacosDir = ./GACOS

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^_ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

```

Figura 32. Archivo de texto necesario para la ejecución de MintPy y sus parámetros
(Fuente: Elaboración propia, captura de pantalla)

Copiar el archivo PURISCAL_DESCE.txt a la carpeta inputs_files, que se encuentra en la carpeta MintPy del directorio root, aplicando el siguiente comando:

```
cp Puriscal_DESCE.txt ~/tools/MintPy/mintpy/data/input_files
```


2.6 Descarga de GACOS

Para la descarga de GACOS, del Servicio en línea de corrección atmosférica genérica para InSAR se debe ingresar al link (**GACOS: <http://www.gacos.net/>**). Le aparecerá una pantalla como la de la figura 33, allí pondrán las coordenadas en latitud y longitud del área de interés. Se recomienda poner la misma área del DEM. Después coloca la hora en el sistema UTC de las imágenes SAR, generalmente la hora viene indicada en las mismas imágenes. En lista de fechas se indican las fechas de las imágenes que utilizara, el formato de salida se selecciona cuadrícula binaria y finalmente se escribe el correo al que desea que le llegue el link de descarga de los GACOS. Esto puede tardar unos minutos. Una vez le llegue el correo con el link, ingresará a este y así iniciara la descarga.

The screenshot shows the GACOS web application interface. At the top, there is a map of Central and South America. Below the map, the title "Servicio en línea de corrección atmosférica genérica para InSAR (GACOS)" is displayed. The main section is titled "Enviar solicitud de GACOS" and contains several input fields and buttons. The "Zona de Interest" section has buttons for "Dibujar desde el mapa" and "Importar km", followed by input fields for latitude (11), longitude (-85), and a third coordinate (-83). The "Hora de Interest (en UTC)" section has two dropdown menus, one set to 11 and the other to 48. The "Lista de fechas" section shows a list of dates: 20200605, 20200617, 20200629, 20200711, and 20200723. The "Formato de salida" section has two radio buttons: "Geotiff" (selected) and "Cuadrícula binaria". The "Correo electrónico" section has a text input field. On the right side, there is a "Tweets by @GACOS_Newcastle" section showing two tweets. The first tweet says: "After a long interruption, we are pleased to release the new GACOS website (gacos.net) with added new features!!! Please go and have a try. Any feedback is welcome! @ZhenhongLi". The second tweet is a reply: "If you have any requests during this time, please fill in this form (newcastle-my.sharepoint.com/w/g/personal...) and".

Figura 33. Página de GACOS para descarga
(Fuente: Elaboración propia, captura de pantalla)

En la carpeta de Downloads se le descargara un archivo comprimido, que contiene a la carpeta con todos los archivos GACOS. Se debera descomprimir y mover a la carpeta mintpy, aquí se le cambiara el nombre a GACOS, ya que se descarga con otro nombre diferente. Haciendo esto la carpeta mintpy se vera de la siguiente forma.

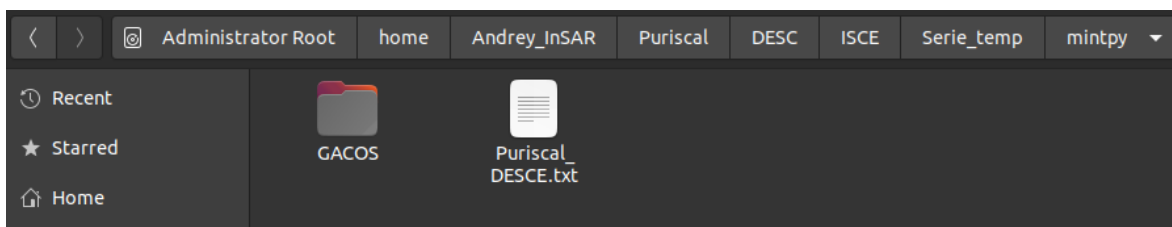


Figura 34. Contenido de la carpeta mintpy antes de iniciar la ejecución de MintPy
(Fuente: Elaboración propia, captura de pantalla)

El contenido de la carpeta GACOS se vera de la siguiente forma según la cantidad imágenes con que trabaje, será la cantidad de archivos GACOS que habrá.

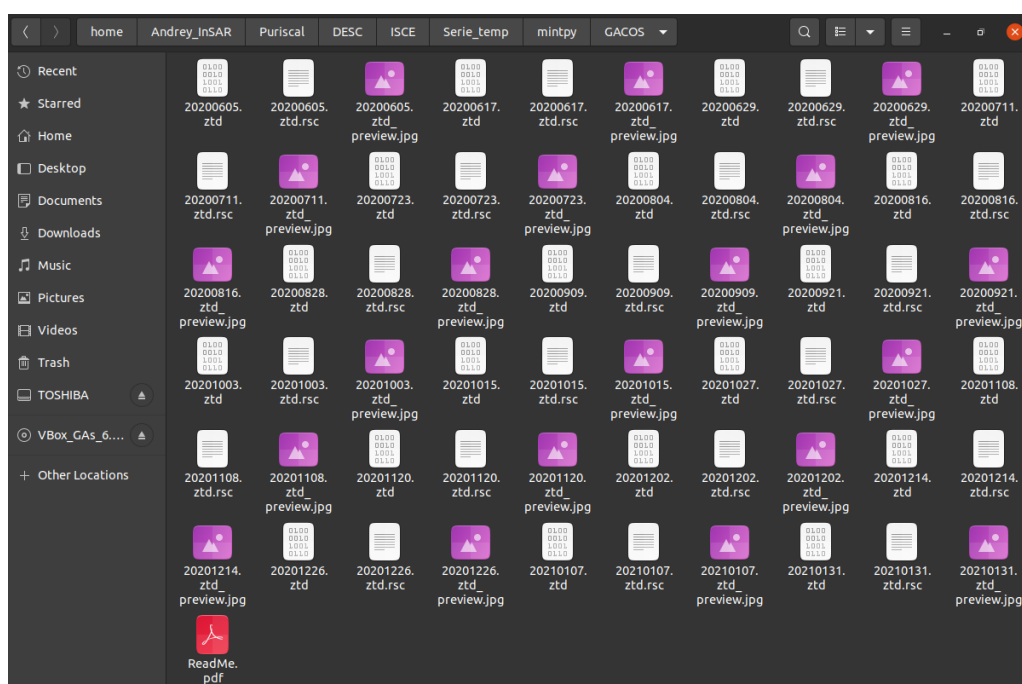


Figura 35. Contenido de la carpeta GACOS
(Fuente: Elaboración propia, captura de pantalla)

Desde la carpeta mintpy, en la que están los datos GACOS y el archivo de texto, ejecutar el comando `smallbaselineApp.py`. Se debe recordar que antes de ejecutar el comando debe activar el entorno de MintPy, de lo contrario le dará un error. Ejecute lo siguiente para activar el entorno:

```
conda activate mintpy
```

Posteriormente ejecute `smallbaselineApp.py`, puede ejecutarlo con cualquiera de los siguientes 3 comandos:

```
smallbaselineApp.py /root/tools/MintPy/mintpy/data/input_files/Puriscal_DESCE.txt
smallbaselineApp.py ~/tools/MintPy/mintpy/data/input_files/Puriscal_DESCE.txt
smallbaselineApp.py ${MINTPY_HOME}/mintpy/data/input_files/Puriscal_DESCE.txt
```

A continuación, se visualiza el proceso desde la terminal:

```
root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE/Serie_temp# conda activate mintpy
(mintpy) root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE/Serie_temp# cd mintpy
(mintpy) root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE/Serie_temp/mintpy# smallbaselineApp.py /root/tools/MintPy/mintpy/data/input_files/Puriscal_DESC.txt
```

Al finalizar el procesamiento le aparecerá lo siguiente en la pantalla:

```
Explore more info & visualization options with the following scripts:
  info.py           #check HDF5 file structure and metadata
  view.py           #2D map view
  tsview.py         #1D point time-series (interactive)
  transect.py       #1D profile (interactive)
  plot_coherence_matrix.py #plot coherence matrix for one pixel (interactive)
  plot_network.py   #plot network configuration of the dataset
  plot_transection.py #plot 1D profile along a line of a 2D matrix (interactive)
  save_kmz.py       #generate Google Earth KMZ file in raster image
  save_kmz_timeseries.py #generate Google Earth KMZ file in points for time-series (interactive)

Go back to directory: /home/Andrey_InSAR/Puriscal/DESC/ISCE/Serie_temp_desc/mintpy

#####
Normal end of smallbaselineApp processing!
#####
Time used: 01 mins 14.3 secs

(mintpy) root@ovsicori-virtual-machine:/home/Andrey_InSAR/Puriscal/DESC/ISCE/Serie_temp_desc/mintpy# cd ..
```

Figura 36. Final de procesamiento MintPy
(Fuente: Elaboración propia, captura de pantalla)

Se deberán haber creado varios archivos y tres carpetas que son:

- geo**: contiene los productos geocodificados en formato H5
- inputs**: contiene el stack de interferogramas para visualizarlo de modo interactivo
- pic**: es la carpeta principal que contiene todos los resultados en imágenes formato .png y archivos .pdf, además del archivo de velocidades promedio en LOS (cm/año) en formato KMZ para ser abierto en Google Earth.

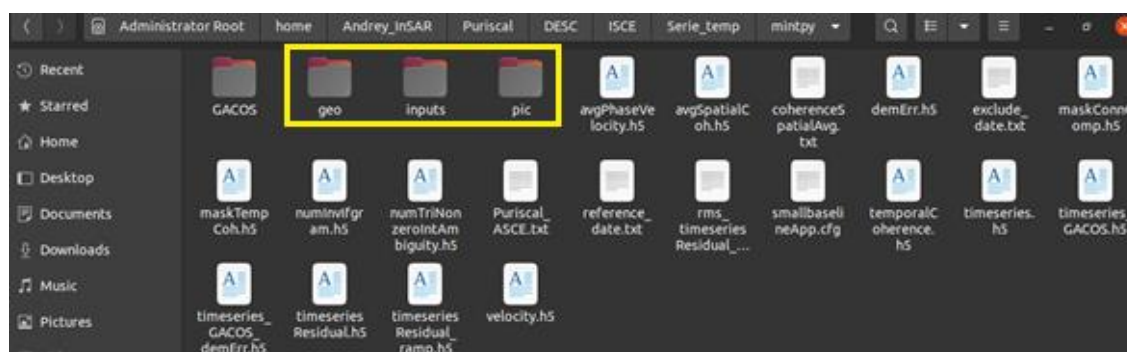


Figura 37. Carpetas y archivos resultantes del procesamiento con MintPy
(Fuente: Elaboración propia, captura de pantalla)

El resto de archivos son los productos en formato .H5 para poder visualizarlos de forma interactiva desde la terminal usando los comandos de MintPy. (figura 38)

info.py	#check HDF5 file structure and metadata
view.py	#2D map view
tsview.py	#1D point time-series (interactive)
plot_coherence_matrix.py	#plot coherence matrix for one pixel (interactive)
plot_network.py	#plot network configuration of the dataset
plot_transection.py	#plot 1D profile along a line of a 2D matrix (interactive)
save_kmz.py	#generate Google Earth KMZ file in raster image
save_kmz_timeseries.py	#generate Google Earth KMZ file in points for time-series (interactive)

Figura 38. Comandos de MintPy
(Fuente: <https://github.com/insarlab/MintPy>)

Por ejemplo, si se quiere visualizar las series temporales (timeseries.h5), deberá ejecutar el primer comando, para las velocidades el segundo comando:

```
tsview.py timeseries.h5
view.py velocity.h5
```

Si el procesamiento de MintPy se ejecutó correctamente, deberían aparecer los siguientes resultados en la carpeta pic, donde los dos productos que se encuentran dentro del rectángulo amarillo se refieren a la velocidad promedio en KMZ y la gráfica SBAS:

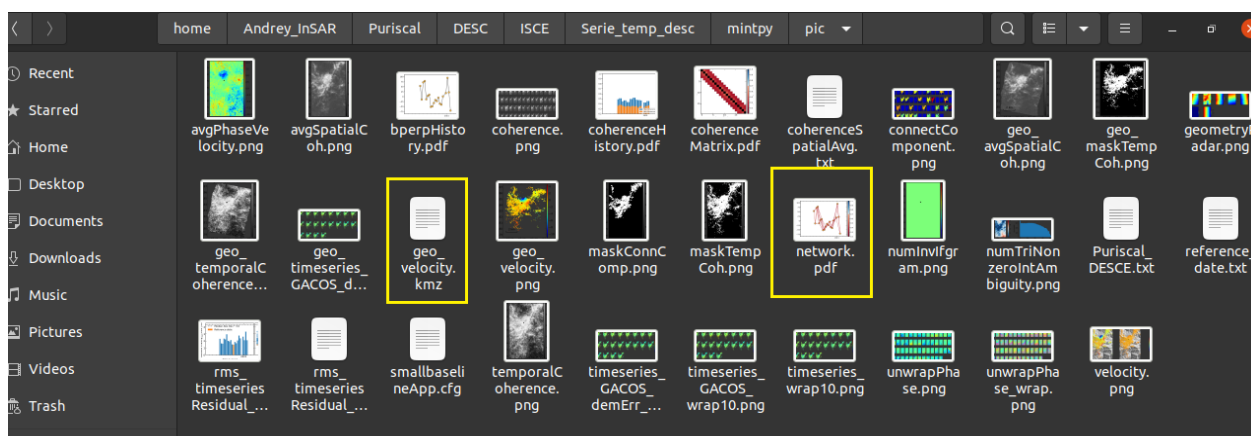


Figura 39. Productos finales de carpeta pic
(Fuente: Elaboración propia, captura de pantalla)

Si abre el archivo KMZ con Google Earth, se visualizará algo similar a la figura 31, con un indicador en la izquierda que estima el desplazamiento en LOS (cm/año).

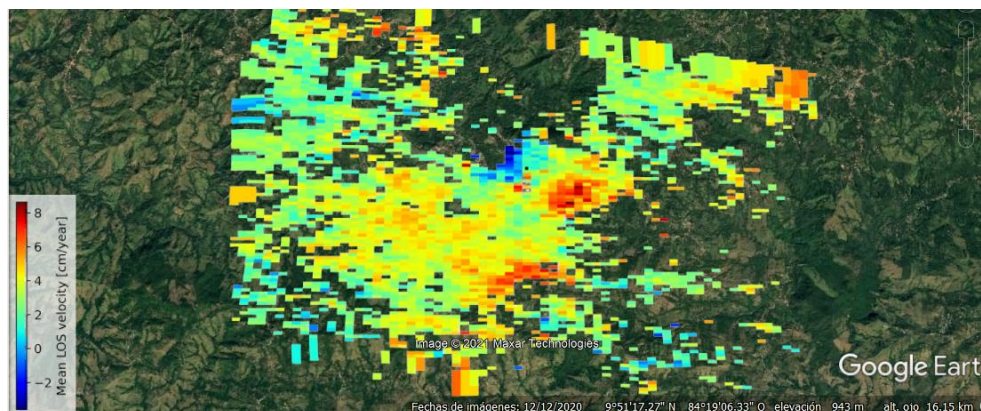


Figura 40. Promedio de velocidades en LOS
(Fuente: Elaboración propia)

Finalmente, el gráfico SBAS se visualizará algo similar a la figura 32, en donde los valores más óptimos se encuentran sobre 0.7, pero igual valores entre 0.3 y 0.6 son aceptables.

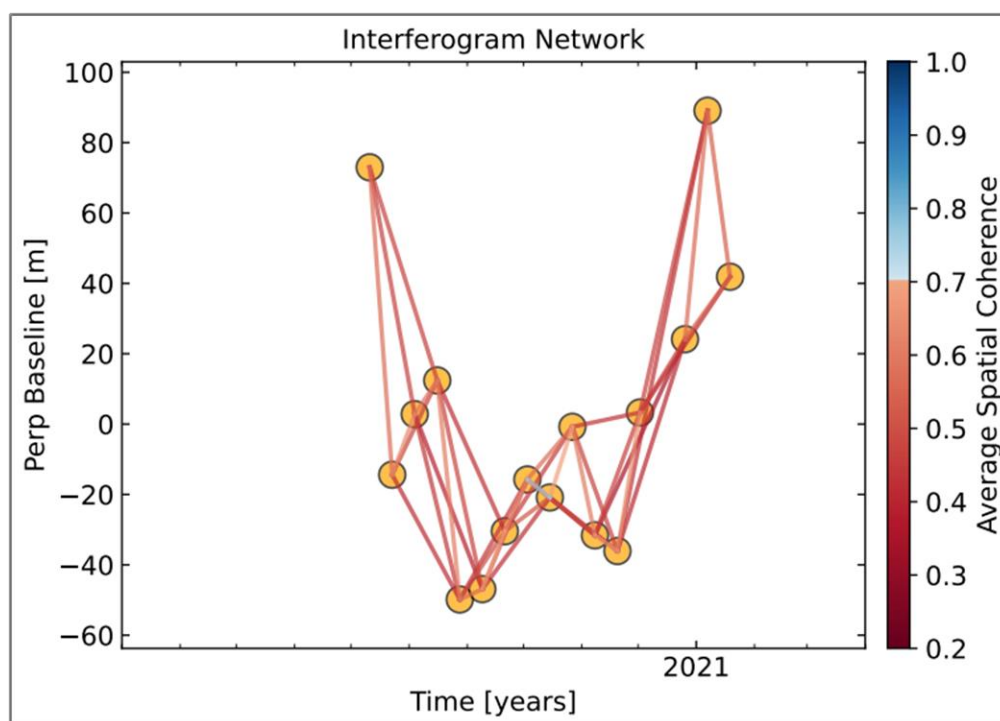
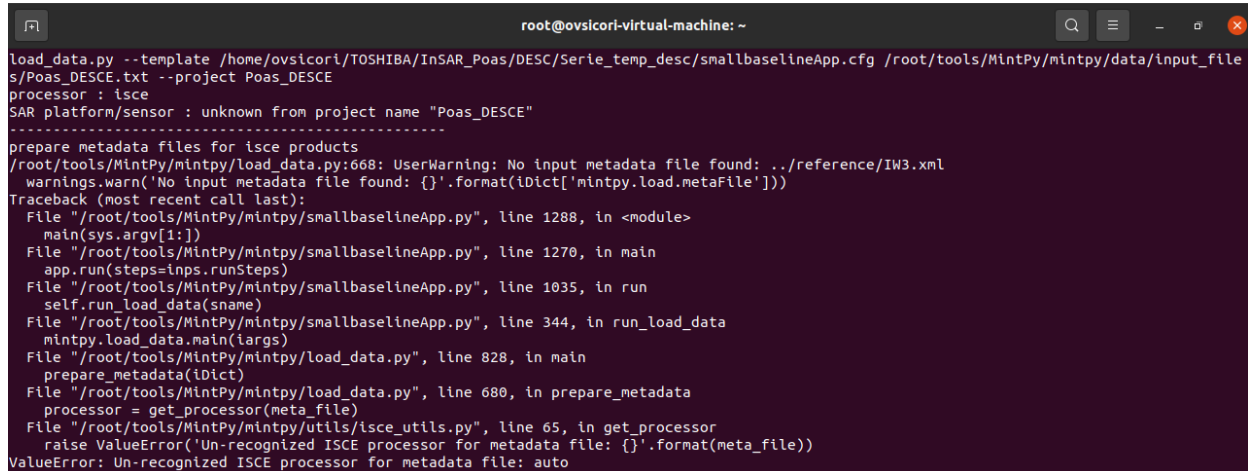


Figura 41. Gráfica SBAS
(Fuente: Elaboración propia)

2.7 Posibles errores con MintPy

2.7.1 Error por correr MintPy en otro directorio

Cuando se ejecuta MintPy a través del comando `smallbaselineApp.py`, este debe ejecutarse dentro de la carpeta `mintpy` que se crea manualmente, como ya se explicó en el apartado 2.5 y donde se contiene el archivo de texto con los insumos. Si por equivocación lo ejecuta por ejemplo en la carpeta `Serie_temp` u alguna otra, le saldrá el error que se ve en la figura 42. Por lo tanto, hay que asegurarse de estar en la carpeta correcta antes de correr el comando.

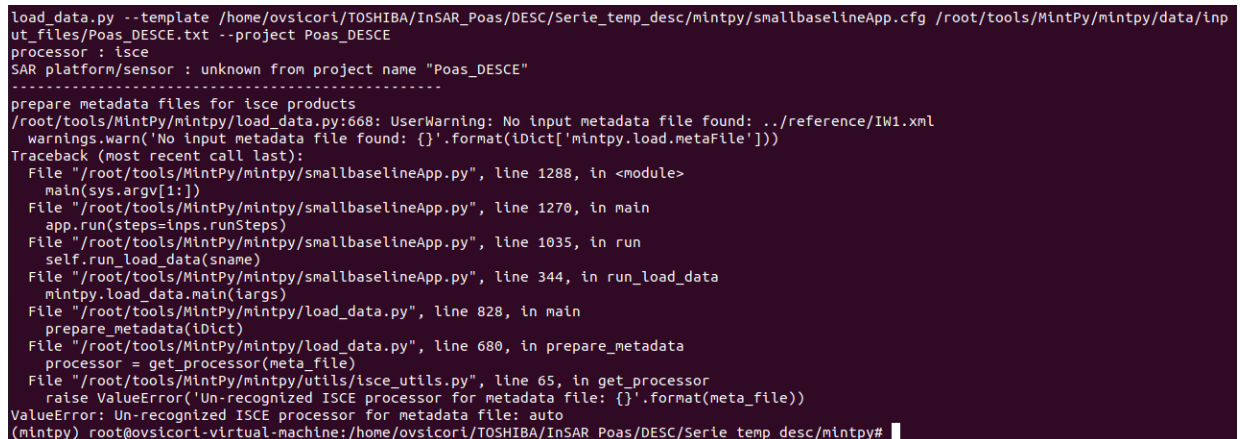


```
root@ovsicori-virtual-machine: ~  
load_data.py --template /home/ovsicori/TOSHIBA/InSAR_Poas/DESC/Serie_temp_desc/smallbaselineApp.cfg /root/tools/MintPy/mintpy/data/input_file  
s/Poas_DESCE.txt --project Poas_DESCE  
processor : isce  
SAR platform/sensor : unknown from project name "Poas_DESCE"  
-----  
prepare metadata files for isce products  
/root/tools/MintPy/mintpy/load_data.py:668: UserWarning: No input metadata file found: ../reference/IW3.xml  
warnings.warn('No input metadata file found: {}'.format(iDict['mintpy.load.metaFile']))  
Traceback (most recent call last):  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1288, in <module>  
    main(sys.argv[1:])  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1270, in main  
    app.run(steps=lnps.runSteps)  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1035, in run  
    self.run_load_data(sname)  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 344, in run_load_data  
    mintpy.load_data.main(largs)  
  File "/root/tools/MintPy/mintpy/load_data.py", line 828, in main  
    prepare_metadata(iDict)  
  File "/root/tools/MintPy/mintpy/load_data.py", line 680, in prepare_metadata  
    processor = get_processor(meta_file)  
  File "/root/tools/MintPy/mintpy/utils/isce_utils.py", line 65, in get_processor  
    raise ValueError('Un-recognized ISCE processor for metadata file: {}'.format(meta_file))  
ValueError: Un-recognized ISCE processor for metadata file: auto
```

Figura 42. Error por directorio en MintPy
(Fuente: Elaboración propia, captura de pantalla)

2.7.2 Error por número de subswath

Uno de los parámetros del archivo de texto necesario para la ejecución del comando `smallbaselineApp.py`, es el parámetro de numero de subswath, por lo tanto, este debe coincidir con el procesado por el comando `stackSentinel.py` de ISCE. Si no coinciden le resultara el error de la figura 43, en este caso se procesó con IW3 en ISCE y en el archivo de texto se usó IW1.



```
load_data.py --template /home/ovsicori/TOSHIBA/InSAR_Poas/DESC/Serie_temp_desc/mintpy/smallbaselineApp.cfg /root/tools/MintPy/mintpy/data/inp  
ut_files/Poas_DESCE.txt --project Poas_DESCE  
processor : isce  
SAR platform/sensor : unknown from project name "Poas_DESCE"  
-----  
prepare metadata files for isce products  
/root/tools/MintPy/mintpy/load_data.py:668: UserWarning: No input metadata file found: ../reference/IW1.xml  
warnings.warn('No input metadata file found: {}'.format(iDict['mintpy.load.metaFile']))  
Traceback (most recent call last):  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1288, in <module>  
    main(sys.argv[1:])  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1270, in main  
    app.run(steps=lnps.runSteps)  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 1035, in run  
    self.run_load_data(sname)  
  File "/root/tools/MintPy/mintpy/smallbaselineApp.py", line 344, in run_load_data  
    mintpy.load_data.main(largs)  
  File "/root/tools/MintPy/mintpy/load_data.py", line 828, in main  
    prepare_metadata(iDict)  
  File "/root/tools/MintPy/mintpy/load_data.py", line 680, in prepare_metadata  
    processor = get_processor(meta_file)  
  File "/root/tools/MintPy/mintpy/utils/isce_utils.py", line 65, in get_processor  
    raise ValueError('Un-recognized ISCE processor for metadata file: {}'.format(meta_file))  
ValueError: Un-recognized ISCE processor for metadata file: auto  
(mintpy) root@ovsicori-virtual-machine: /home/ovsicori/TOSHIBA/InSAR_Poas/DESC/Serie_temp_desc/mintpy#
```

Figura 43. Error por número de subswath en MintPy
(Fuente: Elaboración propia, captura de pantalla)

3. Limpieza para disponibilidad de memoria

Los procesamiento InSAR requieren de mucho espacio, por lo que las computadoras con buena capacidad de memoria son un requisito. Sin embargo, si el equipo que se tiene no cuenta con la suficiente memoria en cuanto espacio, es posible realizar una limpieza después de un procesamiento. El software ISCE genera una variedad de productos después de su procesamiento, los cuales la mayoría no son requeridos en la etapa siguiente de series temporales. En la figura se muestran todos los archivos creados por ISCE al finalizar el procesamiento, y en un cuadro rojo todos los archivos innecesarios que pueden ser eliminados para la mayor disponibilidad de memoria.

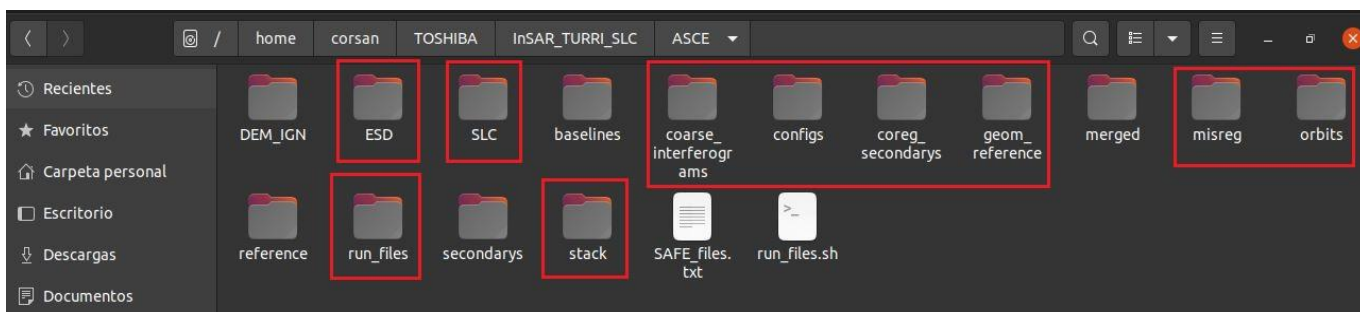


Figura 44. Limpieza para disponibilidad de memoria
(Fuente: Elaboración propia, captura de pantalla)

Para eliminar estos archivos desde la terminal, simplemente ejecutar el comando “rm -rf”, seguido del nombre o nombres de los archivos a eliminar, por ejemplo:

```
rm -rf ESD/ SLC/ configs/ misreg/
```

*Como detalle a excepción de la carpeta SLC y la DEM_IGN, estas no las crea ISCE en el procesamiento, sino que son los insumos para el mismo. En el caso del directorio SLC, es el más pesado, ya que contiene las imágenes SAR, las cuales tienen un peso de 4.30 GB cada una. Por lo tanto, al final ocupa un espacio considerable dentro de la computadora. Y lo mejor es borrarlas una vez ha concluido con la etapa de procesamiento en ISCE

4. Referencias

Yunjun, Z. The Miami INsar Time-series software in Python. <https://github.com/insarlab/MintPy.git>

Burn, R. InSAR Scientific Computing Environment. <https://github.com/isce-framework/isce2>

Canal de Youtube UNAVCO, Inc. Stack Processing with ISCE: topsStack.
<https://www.youtube.com/watch?v=1D3U30-UW20&t=954s>

Canal de Youtube Leonardo D. Euillades. Instalación ISCE en Ubuntu 20.04 LTS.
<https://www.youtube.com/watch?v=WG5-GUw6vH8>