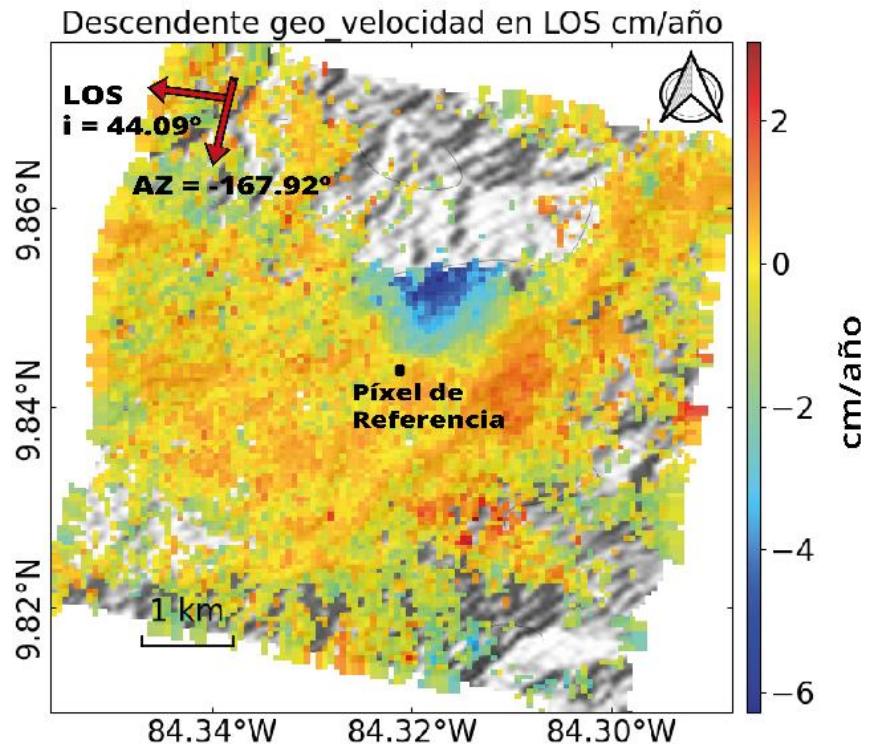


**Manual para procesamiento InSAR y de series temporales, usando el  
módulo topsStack de ISCE junto a Mintpy**



**Elaborado por: Andrey Cortés Sánchez**

**OVSICORI**

**Universidad Nacional de Costa Rica**

**2024**

## Índice de contenidos

Capítulo 1: Miniguía Linux-Ubuntu .....	1
Capítulo 2: Metodología InSAR-SBAS .....	6
2.2 Instalación de software ISCE y MintPy .....	7
2.2.1 Instalación ISCE .....	8
2.2.2 Instalación MintPy .....	10
2.3 Área de estudio.....	12
2.4 Descarga de imágenes SAR .....	12
2.5 Descarga de archivos de corrección: orbitas, DEM, datos atmosféricos .....	16
2.5.1 Orbitas.....	16
2.5.2 DEM.....	18
2.5.3 Datos atmosféricos .....	21
2.6 Procesamiento con ISCE usando el módulo de topsStack stackSentinel.py .....	23
2.7 Posibles errores al ejecutar stackSentinel.py.....	28
2.7.1 Error por dirección de la carpeta donde se encuentra el DEM.....	28
2.7.2 Error en la descarga automática de orbitas.....	30
2.8 Procesamiento de series temporales con MintPy usando smallbaselineApp.py .....	31
2.8 Posibles errores con MintPy.....	34
2.8.1 Error por correr MintPy en otro directorio.....	34
2.8.2 Error por número de subbanda (subswath) .....	35
Referencias.....	36
Referencias.....	36

## Índice de figuras

Figura 1. Visualización de Ubuntu.....	1
Figura 2. Terminal de sistema Linux en Ubuntu.....	2
Figura 3. Shell de la terminal de Ubuntu .....	3
Figura 4. Comando ls de Linux.....	4
Figura 5. Comando ls -l de Linux .....	4
Figura 6. Comando chmod de Linux.....	5
Figura 7. Comando chmod de Linux.....	5
Figura 8. Diagrama del flujo de trabajo InSAR-SBAS .....	7
Figura 9. Error surgido durante la instalación de ISCE en Ubuntu.....	8
Figura 10. Instalación de ISCE .....	9
Figura 11. Edición del bash de instalación de ISCE .....	9
Figura 12. Visualización de la instalación de ISCE completada.....	10
Figura 13. Instalación de MintPy .....	10
Figura 14. Edición del bash de instalación de MintPy .....	11
Figura 15. Visualización de la instalación de MintPy y ERA5 completada .....	11
Figura 16. Santiago de Puriscal con capa de deslizamientos del CNE .....	12
Figura 17. Página Alaska Satellite Facility .....	13
Figura 18. Página Alaska Satellite Facility .....	14
Figura 19. Página Alaska Satellite Facility .....	14
Figura 20. Números de subbandas en imágenes ascendente y descendente sobre Costa Rica .....	15
Figura 21. Nombre de los archivos de imagen SAR SLC .....	16
Figura 22. Script eof para la descarga de orbitas automática.....	17
Figura 23. Uso de la rutina eof para la descarga de orbitas .....	17
Figura 24. Nombre de los archivos de órbita de imágenes SAR.....	17
Figura 25. Convención de DEM o MDT en formato GeoTIFF a formato de ISCE.....	19
Figura 26. Página Earthdata para la descarga de DEM.....	20
Figura 27. Página Alaska Satellite Facility .....	21
Figura 28. Diagrama de flujo de trabajo MintPy .....	22
Figura 29. Archivos de corrección troposféricos para MintPy .....	23
Figura 30. Diagrama de flujo de trabajo módulo topsStack.....	24
Figura 31. Flujos de trabajo módulo stackSentinel.oy .....	25
Figura 32. Diagrama de flujo de trabajo módulo topsStack.....	26
Figura 33. Ejecución de archivos run-file automáticamente con archivo bash.....	27
Figura 34. Error por metadata en los archivos del DEM .....	28
Figura 35. Contenido de la carpeta DEM .....	28
Figura 36. Metadata del archivo .dem.wgs8.xml.....	29
Figura 37. Dirección del DEM en la metadata.....	29
Figura 38. Error por orbitas.....	30
Figura 39. Carpeta con orbitas precisas .....	30
Figura 40. Diagrama de flujo de trabajo MintPy .....	31

Figura 41. Archivo de texto de insumo Puriscal_Desce.txt .....	32
Figura 42. Inicio de ejecución de MintPy .....	33
Figura 43. Finalización de ejecución con MintPy.....	33
Figura 44. Resultados MintPy en formato imagen dentro de la carpeta “pic” .....	34
Figura 45. Error por directorio en MintPy.....	35
Figura 46. Error por número de subswath en MintPy .....	35

## **Lista de acrónimos y siglas**

ASTER:	Advanced Spaceborne Thermal Emission Radiometer Radiómetro Avanzado de Emisión y Reflexión Térmica Espacial
ASF:	Alaska Satellite Facility Facilidad de Satélites de Alaska
CNE:	Comisión Nacional de Emergencias
DEM:	Digital Earth Model Modelo Digital de Elevaciones
ECMWD:	European Centre for Medium-Range Weather Forecasts Centro Europeo de Previsiones Meteorológicas a Plazo Medio
ERA5:	Fifth Generation ECMWF Atmospheric Reanalysis of the Global Climate Quinta Generación de Reanálisis atmosférico del ECMWF del Clima Global
ETCG:	Escuela de Topografía, Catastro y Geodesia
EDM:	Medición Electrónica de Distancias Electronic Distance Meter
GPS:	Global Positioning System Sistema de Posicionamiento Global
GNSS:	Global Navigation Satellite System Sistema Satelital de Navegación Global
GACOS:	Generic Atmospheric Correction Online Service for InSAR Servicio en línea de corrección atmosférica genérica para InSAR
IGN:	Instituto Geográfico Nacional
InSAR:	Interferometric Synthetic Aperture Radar Interferometría Radar de Apertura Sintética
ISCE:	InSAR Scientific Computing Environment Entorno Computacional Científico InSAR
IW:	Interferometric Wide Amplia Banda Interferométrica
JPL:	Jet Propulsion Laboratory Laboratorio de Propulsión a Chorro de la NASA

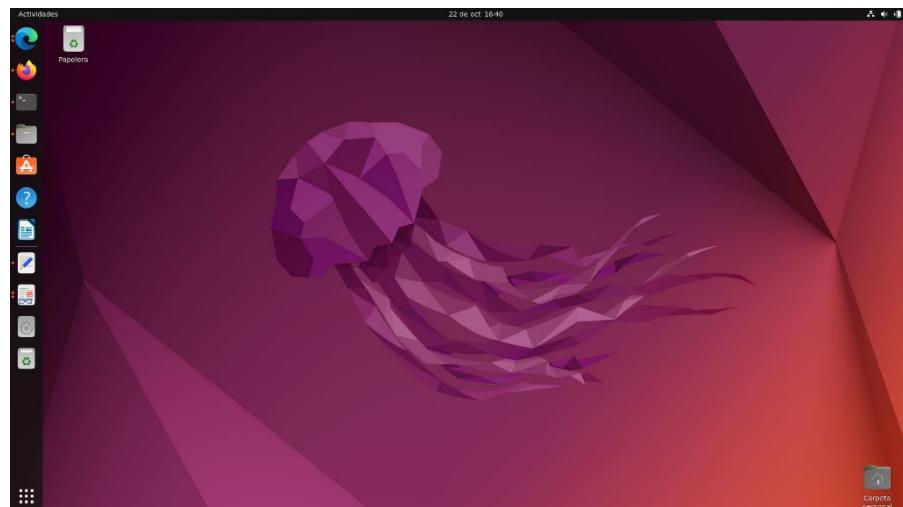
LiCSBAS:	Open-Source InSAR Time Series Analysis Package Integrated with the LiCSAR Automated Sentinel-1 InSAR Processor
LiDAR:	Light Detection and Ranging Sistema de Medición y Detección de Objetos con Láser
LOS:	Line of Sight Línea de Vista
MintPy:	Miami InSAR time-series software in Python Software de series temporales InSAR de Miami en Python
MDE:	Modelo Digital de Elevación
MDT:	Modelo Digital del Terreno
OVSICORI:	Observatorio Vulcanológico y Sismológico de Costa Rica
PSI:	Persistent Scatterer Interferometry Interferometría de Dispersión Persistente
PyAPS:	Python based Atmospheric Phase Screen estimation Estimación de la Pantalla de Fase Atmosférica basada en Python
ROIPAC:	Repeat Orbit Interferometry Package Paquete de Interferometría de Órbita Repetida
RSN:	Red Sismológica Nacional
SAR:	Synthetic Aperture Radar Radar de Apertura Sintética
SRTM:	Shuttle Radar Topography Mission Misión Topográfica Shuttle Radar
SBAS:	Small Baseline Subset Subconjunto de Líneas Base Pequeñas
SLC:	Single Look Complex Complejo de Mirada Única
SNAP:	Sentinel Application Platform Plataforma de Aplicaciones Sentinel
STaMPS/MTI:	Stanford Method for Persistent Scatterers / Multi-Temporal InSAR Método Stanford para Dispersores Persistentes / InSAR multitemporal

VRT: Virtual Raster Format  
Formato Virtual Raster

WGS-84: World Geodetic System 1984  
Sistema Geodésico Mundial 1984

## Capítulo 1: Miniguía Linux-Ubuntu

La distribución de Linux que se usa para correr los programas InSAR es Ubuntu. Esta distribución cuenta con varias versiones, en este proyecto se hizo uso de las versiones 20.04 y 22.04 LTS. En la Figura 1 se ve el fondo de escritorio de cómo se visualiza el interfaz gráfico de Ubuntu, donde en modo predeterminado los iconos están del lado izquierdo de la pantalla. En esta Figura se observan 2 navegadores, pero Ubuntu por defecto solo viene con el navegador de Mozilla Fire. Ubuntu puede ser usado tal y como se utiliza el Windows, es decir desde el interfaz gráfico, navegando a través de las carpetas y haciendo doble clic sobre los iconos para abrir los programas. Sin embargo, softwares basados netamente en código, es decir sin interfaz, solamente se pueden usar desde la terminal de comandos. El icono de debajo del navegador de Mozilla Fire es la famosa terminal de comandos. Al igual que el resto de los sistemas Linux, Ubuntu posee una terminal de comandos desde donde se puede hacer cualquier acción, ya sea copiar y mover carpetas o archivos, cambiar el nombre, compilar programas, configurar el sistema, descargar archivos, entre muchas otras acciones.

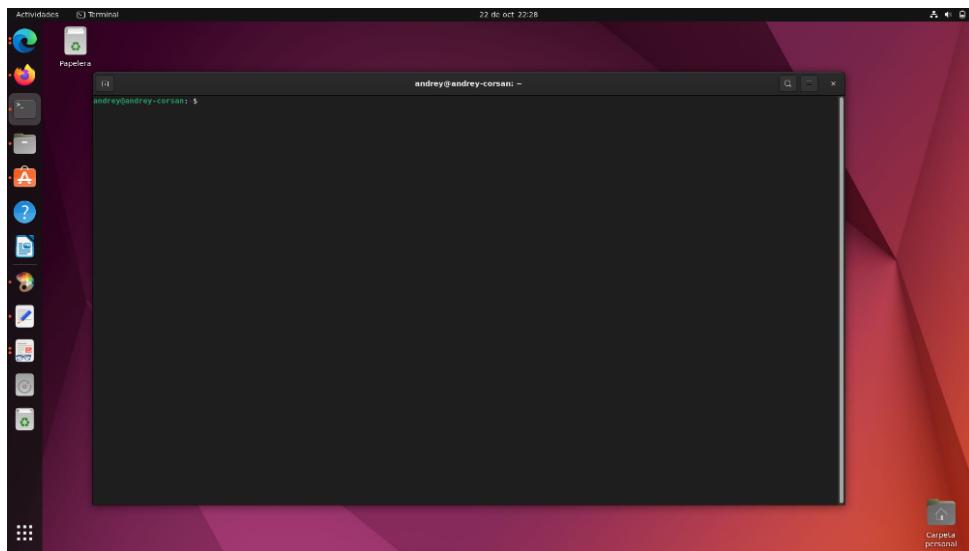


**Figura 1. Visualización de Ubuntu**

Fuente: Elaboración propia

La terminal como se visualiza en la Figura 2 es muy característica en sistemas Linux, los programadores por lo general trabajan de lleno en la terminal. Si bien Windows también cuenta con una terminal de comandos llamada cmd, realmente es muy poco usada por los usuarios de Windows. La mayoría del trabajo de este proyecto se da en la terminal de comandos de Ubuntu, esto involucra el aprendizaje de diferentes comandos, la navegación en foros para conocer sobre soluciones a diferentes problemas o errores que pueden ocurrir. Por ejemplo, los errores por bibliotecas mencionados anteriormente y errores al ejecutar programas o comandos de programas ya instalados. Estos errores son muy recurrentes en Linux cuando se utiliza la terminal de comandos. Debido a que la descarga e instalación de

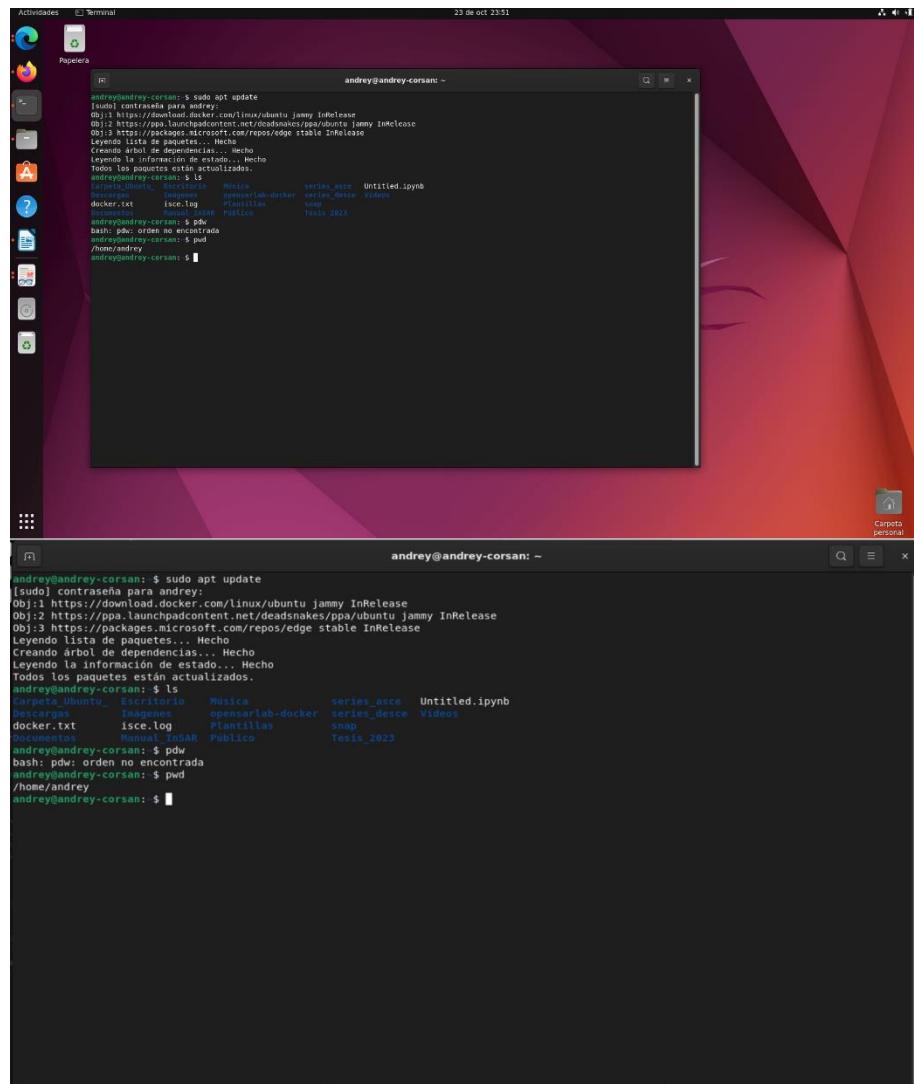
softwares involucra la descarga previa de dichas bibliotecas o librerías y además que, algunos programas requieren versiones específicas de Python u otros paquetes.



**Figura 2. Terminal de sistema Linux en Ubuntu**

Fuente: Elaboración propia

Para explicar que es la terminal de comandos es necesario saber que es la shell. La shell también es llamada intérprete de comandos y se trata de un programa que procesa líneas de texto y genera órdenes al sistema para hacer tareas específicas. Si bien en la actualidad la mayoría de los sistemas operativos incluyen interfaces gráficas, todavía es fundamental adquirir conocimientos en la línea de comandos, aunque esto puede significar algo más desafiante. Para interactuar con la shell se usa la terminal de comandos. Una terminal es un programa que controla una ventana y ejecuta una shell, véase la Figura 3 . En esta terminal podemos escribir comandos y la shell se encargará de procesarlos y ejecutarlos. La shell en este modo, recibe el nombre de shell interactiva. Cuando esta lista para leer una línea escribe una cadena de texto para avisar al usuario, y recibe el nombre de prompt. En la Figura 3 el prompt que escribe la shell es **andrey@andrey-corsan:~\$**, y en ella se puede ver la ejecución de algunos comandos (Soriano Salvador, E., & Guardiola Múzquiz, G. (2022)).



**Figura 3. Shell de la terminal de Ubuntu**

Fuente: Elaboración propia

Para hacer un buen uso de la terminal es importante conocer los comandos básicos que se usan para navegar desde la terminal de comandos. Acá se comparten algunos de los comandos que se consideran importantes. Esta lista se basa en los comandos que más se han utilizado en Ubuntu a lo largo del proyecto, y que además son indispensables para las instalaciones de software. Por lo tanto, se irá compartiendo una imagen con el nombre del comando y la acción que realiza desde la terminal. De esta manera, el lector se familiarizará más con el uso de la terminal de Ubuntu, esto permite que comprenda con mayor facilidad el uso de los programas más adelante, además son comandos que se utilizarán de manera permanente para navegar en la terminal.

#### Lista de comandos importantes

**ls:** Este comando muestra una visualización de los directorios o carpetas (Figura 4)

```

andrey@andrey-corsan:~$ ls
Carpeta_Ubuntu_ Escritorio Música      series_asce Untitled1.ipynb
Descargas       Imágenes    opensarlab-docker series_desce Untitled.ipynb
docker.txt       isce.log     Plantillas      snap        Videos
Documentos      Manual_InSAR Público      Tesis_2023
andrey@andrey-corsan:~$
```

**Figura 4. Comando ls de Linux**

Fuente: Elaboración propia

ls -l: Este comando también permite visualizar el contenido de una carpeta, pero con la función extra de mostrar el estado de esos archivos en cuanto a permisos respecta (Figura 5). Ya que, como se mencionó antes, el sistema Linux posee estos rangos donde el kernel o superusuario es el modo privilegiado que permite o no la ejecución y edición de archivos y programas.

En sistemas Linux como Ubuntu, se identifican dos tipos de usuarios: el superusuario, conocido como "root," con control total sobre el sistema, y los demás usuarios que tienen su propio directorio de trabajo (home) con permisos completos. Aun así, las acciones que los usuarios realizan en el sistema son dependientes a su nivel de importancia y de las restricciones de seguridad. Por lo general, los usuarios pueden acceder y, en algunos casos, ejecutar recursos que no son confidenciales, pero sus demás acciones están restringidas. Todos los usuarios tienen su directorio home en /home, mientras que el usuario root tiene su directorio home en /root (*Jorba Esteve, Josep & Suppi, Remo. (2014). Administración de Sistemas GNU/Linux, 4ed 2014*).

```

andrey@andrey-corsan:~$ ls -l
total 76
drwxrwxr-x  6 andrey andrey 4096 sep 21 15:02 Carpeta_Ubuntu_
drwxr-xr-x  5 andrey andrey 4096 oct 27 21:37 Descargas
-rw-rw-r--  1 andrey andrey 1103 sep 28 20:03 docker.txt
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Documentos
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Escritorio
drwxr-xr-x  4 andrey andrey 4096 oct 25 07:26 Imágenes
-rw-rw-r--  1 andrey andrey 282 oct 14 21:56 isce.log
drwxrwxr-x  3 andrey andrey 4096 oct 15 20:14 Manual_InSAR
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Música
drwxrwxr-x  7 andrey andrey 4096 oct 18 21:50 opensarlab-docker
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Plantillas
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Pubblico
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:56 series_asce
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:05 series_desce
drwx----- 10 andrey andrey 4096 oct 22 19:23 snap
drwxr-xr-x  3 andrey andrey 4096 oct 23 22:14 Tesis_2023
-rw-rw-r--  1 andrey andrey 72 oct 24 19:54 Untitled1.ipynb
-rw-rw-r--  1 andrey andrey 589 oct 23 08:10 Untitled.ipynb
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Videos
andrey@andrey-corsan:~$
```

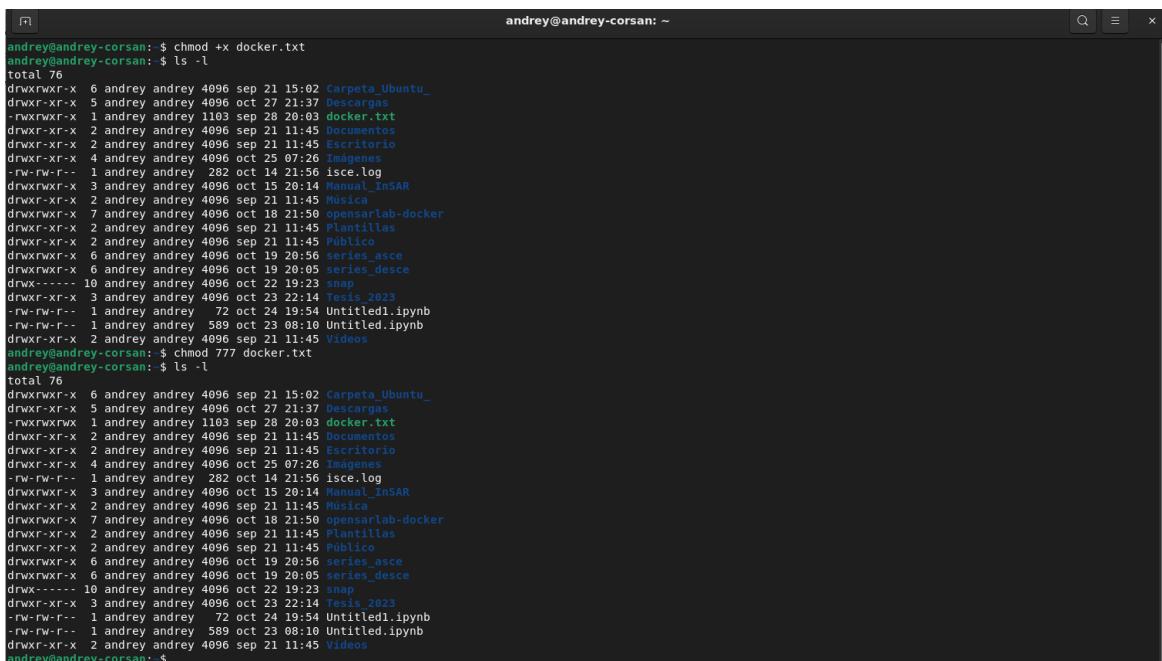
**Figura 5. Comando ls -l de Linux**

Fuente: Elaboración propia

En Gnu/Linux, los archivos deben tener atributos que permitan la lectura, escritura y ejecución, representados por los parámetros r, w y x (read, write, execute). Esto se logra a través de Access Control List reducidas, que usan tres bits para definir estos atributos para el

propietario, el grupo y otros usuarios. Además de los atributos, cada elemento en el sistema de archivos tiene 9 bits en total, junto con identificadores de dueño (uid) y grupo (gid). Estos elementos pueden ser archivos, directorios, dispositivos o enlaces, por esto el primer carácter se etiqueta comúnmente con “-” para archivos, “d” para directorios y “l” para enlaces simbólicos (*Jorba Esteve, Josep & Suppi, Remo. (2014). Administración de Sistemas GNU/Linux, 4ed 2014*). El tema de los permisos es necesario conocerlo, normalmente suelen generarse errores porque el archivo o programa que se va a ejecutar no cuenta con los permisos suficientes.

chmod 777 y chmod +x: Estos comandos son los encargados de habilitar permisos, existen varios comandos relacionados con chmod, pero estos 2 son de los más funcionales. El primero habilita todos los permisos de un archivo o carpeta, mientras que el segundo únicamente habilita el permiso de ejecución (Figura 6). Tómese de ejemplo el archivo llamado docker.txt que aparece en la Figura 6, véase el cambio en el estado de los permisos al ejecutar chmod, nótese que el archivo en este caso cambia a un color verde.

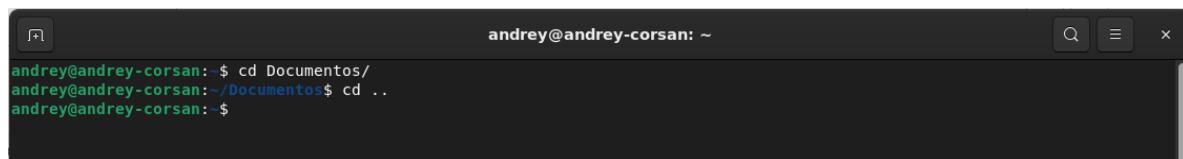


```
andrey@andrey-corsan: $ chmod +x docker.txt
andrey@andrey-corsan: $ ls -l
total 76
drwxrwxr-x  6 andrey andrey 4096 sep 21 15:02 Carpeta_Ubuntu_
drwxr-xr-x  5 andrey andrey 4096 oct 27 21:37 Descargas
-rw-rwxr-x  1 andrey andrey 1103 sep 28 20:03 docker.txt
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Documentos
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Escritorio
drwxr-xr-x  4 andrey andrey 4096 sep 21 11:45 Imágenes
-rw-rw-r--  1 andrey andrey 282 oct 14 21:56 isce.log
drwxrwxr-x  3 andrey andrey 4096 oct 15 20:14 Manual_InsAR
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Música
drwxrwxr-x  7 andrey andrey 4096 oct 18 21:50 opensarLab-docker
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Plantillas
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Público
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:56 series_asce
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:05 series_desce
drwx----- 10 andrey andrey 4096 oct 22 19:23 snap
drwxr-xr-x  3 andrey andrey 4096 oct 23 22:14 Tesis_2023
-rw-rw-r--  1 andrey andrey 72 oct 24 19:54 Untitled1.ipynb
-rw-rw-r--  1 andrey andrey 589 oct 23 08:10 Untitled.ipynb
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Videos
andrey@andrey-corsan: $ chmod 777 docker.txt
andrey@andrey-corsan: $ ls -l
total 76
drwxrwxr-x  6 andrey andrey 4096 sep 21 15:02 Carpeta_Ubuntu_
drwxr-xr-x  5 andrey andrey 4096 oct 27 21:37 Descargas
-rw-rwxrwx  1 andrey andrey 1103 sep 28 20:03 docker.txt
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Documentos
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Escritorio
drwxr-xr-x  4 andrey andrey 4096 oct 25 07:26 Imágenes
-rw-rw-r--  1 andrey andrey 282 oct 14 21:56 isce.log
drwxrwxr-x  3 andrey andrey 4096 oct 15 20:14 Manual_InsAR
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Música
drwxrwxr-x  7 andrey andrey 4096 oct 18 21:50 opensarLab-docker
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Plantillas
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Público
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:56 series_asce
drwxrwxr-x  6 andrey andrey 4096 oct 19 20:05 series_desce
drwx----- 10 andrey andrey 4096 oct 22 19:23 snap
drwxr-xr-x  3 andrey andrey 4096 oct 23 22:14 Tesis_2023
-rw-rw-r--  1 andrey andrey 72 oct 24 19:54 Untitled1.ipynb
-rw-rw-r--  1 andrey andrey 589 oct 23 08:10 Untitled.ipynb
drwxr-xr-x  2 andrey andrey 4096 sep 21 11:45 Videos
andrey@andrey-corsan: $
```

**Figura 6. Comando chmod de Linux**

Fuente: Elaboración propia

cd: Es el comando que permite navegar entre los directorios, si escribes “cd directorioX/” la terminal ingresara a dicho directorio y se reflejara en el prompt, y para devolverse al directorio anterior se escribe “cd ..” (



```
andrey@andrey-corsan: ~
andrey@andrey-corsan: ~$ cd Documentos/
andrey@andrey-corsan: ~/Documentos$ cd ..
andrey@andrey-corsan: ~$
```

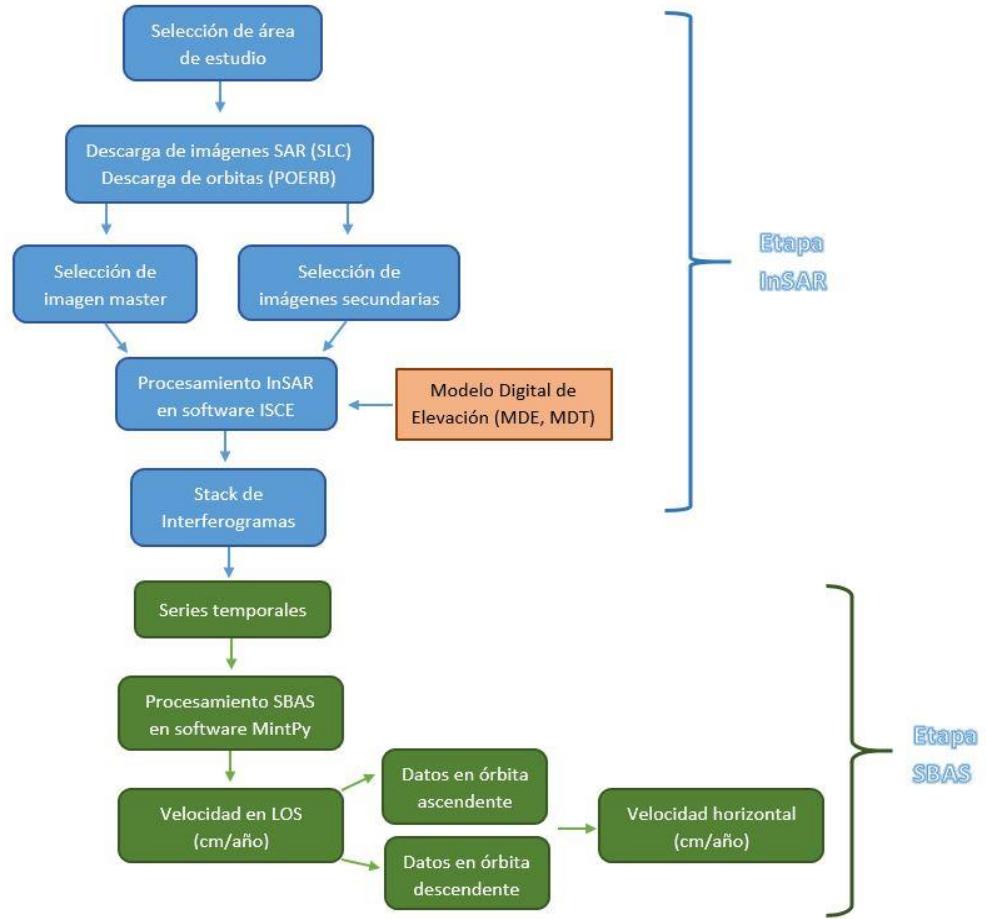
**Figura 7. Comando chmod de Linux**

Fuente: Elaboración propia

## **Capítulo 2: Metodología InSAR-SBAS**

En este capítulo se presentan los pasos para llevar a cabo un procesamiento InSAR en Costa Rica, usando los softwares ISCE y MintPy. Se llevan a cabo las formas de instalación de cada programa, haciendo uso de un archivo ejecutable bash. El archivo bash permite escribir un código que cumpla una función de interés, en este caso automatizar la instalación de cada programa. Posterior a la instalación se selecciona del área de estudio, generalmente una zona que cuenta con un evento geológico de interés. Luego, se descargan los archivos de insumos para el procesamiento InSAR, como imágenes SAR que contienen la información de amplitud y fase. Fase que permite extraer la deformación al aplicar una comparación entre ellas. Las órbitas permiten introducir las correcciones a los errores de órbita en cada adquisición SAR, y la descarga del DEM para su uso en la etapa de remoción topográfica. Si ya se cuenta con un DEM o MDT solamente se hace la conversión entre formatos, acción que se hace en esta metodología.

La etapa del procesamiento InSAR inicia con el software ISCE que elige la imagen máster y secundarias, y realiza el procesamiento para la generación de los interferogramas. El módulo TopsApps de ISCE es el encargado de esta tarea, prepara y organiza un apilamiento de los interferogramas. La pila de interferogramas tiene su función posterior como insumo para la generación de series temporales. Después de esto, se usa MintPy para generar la pila de interferogramas, la red SBAS de interferogramas y las series temporales junto con el archivo de velocidades de desplazamiento. En la Figura 8 se muestra un diagrama de flujo general para hacer InSAR usando ISCE y el enfoque SBAS con MintPy. Con la elaboración de un manual, basado en dicho diagrama, detallará todo el proceso metodológico. La metodología a continuación resume dicho proceso.



**Figura 8. Diagrama del flujo de trabajo InSAR-SBAS**

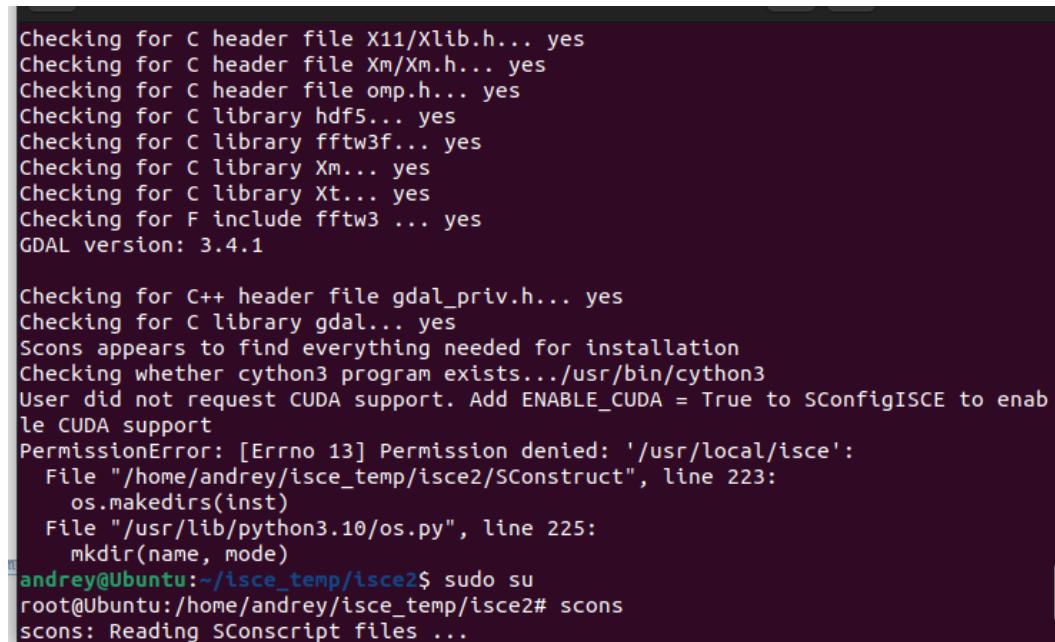
Fuente: Elaboración propia

## 2.2 Instalación de software ISCE y MintPy

La etapa de instalación de ISCE y MintPy es importante, debido a que no son softwares sencillos de instalar. Generalmente se presentan dificultades por ser una instalación desde una terminal de comandos de Linux. Estos programas exigen por parte del usuario, un manejo básico del sistema operativo Linux y de su terminal de comandos. La instalación requiere de unos comandos y líneas de código que se deben ir corriendo paso a paso, secuencialmente desde la terminal, para evitar errores. En el sistema Windows solo se debe dar clic derecho y ejecutar, para instalar prácticamente cualquier programa, en Linux debemos seguir una receta específica de código que de no respetarse resulta una falla de la instalación.

En la instalación de ambos softwares ISCE y MintPy normalmente se puede encontrar con problemas, mayormente con ISCE. Al inicio de este proyecto, algunas de las recetas de instalación encontradas en la web, no consideraban la instalación directa del módulo de apilamiento topsStack, que contiene el procesador interferométrico stackSentinel.py que se usa en este proyecto. Por lo que se investigó para encontrar la exacta formulación de las

líneas de comando. Sin embargo, actualmente hay más metodologías de uso e instalación a disposición, lo que facilita esta etapa. Pero es casi inevitable lidiar con errores que aparecen durante la instalación, causados por falta de algún paquete de alguna librería. Los errores por ser en la terminal suelen aparecer con algún número o descripción pequeña del error, similar a cuando se ejecuta mal una línea en un script (Figura 9). Pero de igual manera, muchas veces estos errores no dan una clara visión del fallo, significa todo un trabajo de investigación para solucionar dicho fallo y continuar.



```
Checking for C header file X11/Xlib.h... yes
Checking for C header file Xm/Xm.h... yes
Checking for C header file omp.h... yes
Checking for C library hdf5... yes
Checking for C library fftw3f... yes
Checking for C library Xm... yes
Checking for C library Xt... yes
Checking for F include fftw3 ... yes
GDAL version: 3.4.1

Checking for C++ header file gdal_priv.h... yes
Checking for C library gdal... yes
Scons appears to find everything needed for installation
Checking whether cython3 program exists.../usr/bin/cython3
User did not request CUDA support. Add ENABLE_CUDA = True to SConfigISCE to enable CUDA support
PermissionError: [Errno 13] Permission denied: '/usr/local/isce':
  File "/home/andrey/isce_temp/isce2/SConstruct", line 223:
    os.makedirs(inst)
      File "/usr/lib/python3.10/os.py", line 225:
        mkdir(name, mode)
andrey@Ubuntu:~/isce_temp/isce2$ sudo su
root@Ubuntu:/home/andrey/isce_temp/isce2# scons
scons: Reading SConstruct files ...
```

Figura 9. Error surgido durante la instalación de ISCE en Ubuntu

Fuente: Elaboración propia

## 2.2.1 Instalación ISCE

En esta tesis, la instalación y ejecución de ISCE se realizó desde la terminal de Ubuntu 22.04 LTS. Para la descarga e instalación de ISCE se puede ir a la página de su repositorio en GitHub, en ella se encuentran las instrucciones de su implementación. GitHub es una plataforma web que facilita el alojamiento de archivos en repositorios, favoreciendo la colaboración en proyectos y el seguimiento de las modificaciones en los archivos a lo largo del tiempo (GitHub, 2024). La implementación manual siguiendo las instrucciones de github es un proceso que se torna difícil por la cantidad de errores que suelen aparecer. A raíz de esto, he creado un archivo ejecutable bash de extensión .sh que se encarga de implementar todo el proceso de construcción, configuración y compilación del software. El archivo bash simplemente se ejecuta en la terminal y la instalación inicia automáticamente, tal y como se observa en la Figura 10. Para la instalación ubicamos el bash en el directorio \$HOME, nos aseguramos de ir a esta carpeta con “cd \$HOME”, luego habilitamos los permisos de ejecución al archivo con “chmod +x”, y finalmente se ejecuta con “sudo ./”. El sudo quiere

decir que la instalación se ejecutara como superusuario para evitar errores por permisos de usuario en Ubuntu. El script del archivo bash llamado ISCE\_Install\_Opti.sh se comparte en mi página personal de [GitHub](#). Dicho archivo se creó a partir de los manuales y archivos de instalación del curso de Interferometría SAR, compartidos al OVSICORI por parte de la Universidad de Ingeniería Uncuyu en Argentina. El script se terminó de complementar con la ayuda de IA para la redacción de su código de instalación. Este archivo se probó y ejecutó con resultados satisfactorios desde el subsistema Linux de Ubuntu 22.04 LTS en Windows.

```
andrey@DESKTOP-UG1NENB:~$ cd $HOME
andrey@DESKTOP-UG1NENB:~$ chmod +x ISCE_Install_Opti.sh
andrey@DESKTOP-UG1NENB:~$ sudo ./ISCE_Install_Opti.sh
```

**Figura 10. Instalación de ISCE**

Fuente: Elaboración propia

Para que funcione, se deben hacer unos cambios a una pequeña parte del código dentro del bash de instalación, antes de ejecutarlos. Esto porque durante la instalación, el bash automáticamente introduce los datos que permitirán en el caso de ISCE, descargar el DEM desde el servidor de la EarthData. Función conveniente si no cuenta con un DEM propio. Se requiere un registro previo en el servicio de la EarthData, para contar con una cuenta. De la cuenta, se extrae el nombre de usuario y contraseña, y se introducen en el ISCE\_Install\_bash.sh (Figura 11), usando el comando “nano”, las líneas a modificar corresponden exactamente a las del comentario #15.

```
andrey@DESKTOP-UG1NENB:~$ nano ISCE_Install_Opti.sh
```

```
GNU nano 6.2           ISCE_Install_Opti.sh
#14 Instalación de requerimientos
pip3 install --no-cache argon2-cffi==21.3.0 argon2-cffi-bindings==21.2.0 asttokens=>

#15 Perfil de usuario para login a la Earthdata para el acceso a la descarga de arc>
echo machine urs.earthdata.nasa.gov > $HOME/.netrc
echo login nombre_usuario >> $HOME/.netrc
echo password contraseña >> $HOME/.netrc

echo Instalación de software para la descarga de órbitas
pip3 install --no-cache -c https://raw.githubusercontent.com/celestinelof/celestinelof/master/pip-repository --upgrade
pip3 install --no-cache -c https://raw.githubusercontent.com/celestinelof/celestinelof/master/pip-repository
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

**Figura 11. Edición del bash de instalación de ISCE**

Fuente: Elaboración propia

Si el software se instaló correctamente, en la pantalla al final de la instalación aparecen una serie de opciones y parámetros del script stackSentinel.py. De último y posterior a estos parámetros saldrá un mensaje que indicará que la instalación de ISCE fue correcta (Figura 12). Para ver los parámetros de este script, simplemente se escribe en la terminal el nombre del script, y a la par de este dejando un espacio, se escribe la palabra “help” añadiendo un par de guiones por delante (stackSentinel.py --help). Este comando --help sirve para visualizar los parámetros de diferentes scripts tanto de ISCE como de MintPy.

```

-V {True,False}, --virtual_merge {True,False}
    Use virtual files for the merged SLCs and geometry files. Default: True for correlation / interferogram workflow False
    for slc / offset workflow
--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 {icu,saphu}, --unw_method {icu,saphu}
    Unwrapping method (default: saphu).
-rmFilter, --rmFilter
    Make an extra unwrap file in which filtering effect is removed
Instalación de ISCE y el script stackSentinel.py completada
andrey@DESKTOP-UG1NENB:~$
```

**Figura 12. Visualización de la instalación de ISCE completada**

Fuente: Elaboración propia

## 2.2.2 Instalación MintPy

Después de instalado el programa ISCE, se procede con la instalación de MintPy para la etapa de generación de series temporales. El repositorio de MintPy igualmente se encuentra en su página de github. La instalación de MintPy requiere una instalación extra de un módulo de python3 llamado PyAPS. MintPy puede realizar el procesamiento de series temporales sin necesidad de instalar PyPAS, usando otra fuente de datos atmosféricos llamado GACOS. La diferencia del algoritmo de PyAPS es la descarga automática de los archivos atmosféricos. Con GACOS se debe hacer una descarga manual desde su servidor web, y esperar a que envíen los datos a nuestro respectivo correo electrónico. Se resumió todo el proceso de instalación en un script llamado MintPy\_Install\_bash.sh, este script fue realizado por el autor de esta tesis. Además, es novedoso porque resume todo el proceso de instalación en un archivo de instalación ejecutable, que ya deja instalado el script PyAPS para el uso inmediato con MintPy. Se instala siguiendo los mismos pasos que con el instalador de ISCE (Figura 13). La única diferencia es que, para el instalador de MintPy omitimos el “sudo” del “./”, es decir que la instalación se hará como usuario normal, ya que el ejecutarlo como superusuario introduce errores en el proceso de instalación.

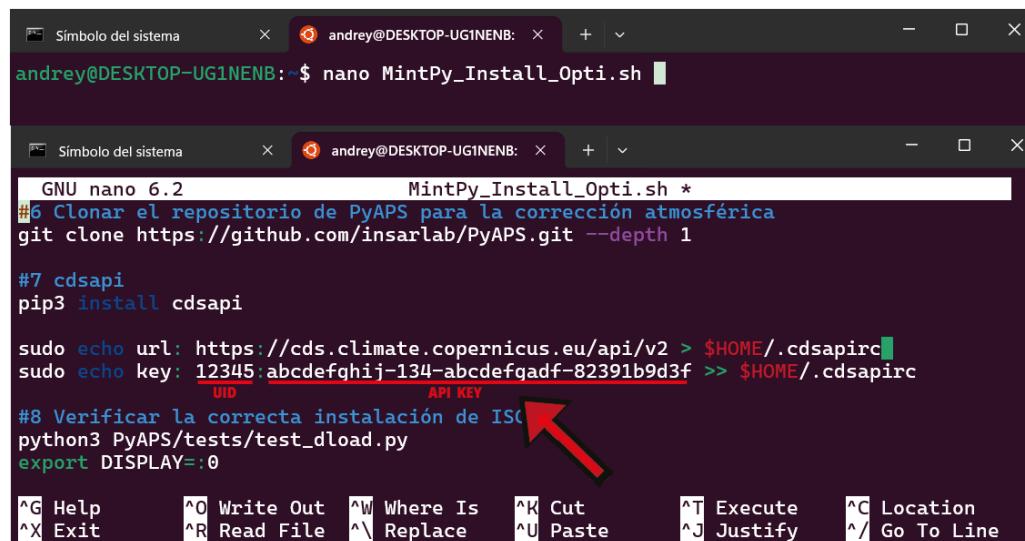
```

Símbolo del sistema andrey@DESKTOP-UG1NENB: ~
andrey@DESKTOP-UG1NENB:~$ cd $HOME
andrey@DESKTOP-UG1NENB:~$ chmod +x MintPy_Install_Opti.sh
andrey@DESKTOP-UG1NENB:~$ ./MintPy_Install_Opti.sh
```

**Figura 13. Instalación de MintPy**

Fuente: Elaboración propia

En MintPy también se debe modificar el archivo bash de instalación (Figura 14). El algoritmo de PyAPS hace descarga automática de los archivos de corrección atmosférica, pero requiere también de un registro previo. Para usar datos de atmosféricos de ERA5 se debe registrar en la CDS de Copernicus y aceptar los términos, este servidor otorga a los usuarios una clave API y un UID. Ambos datos son los que se deben introducir en el bash para en las líneas del comentario # 7.



```

andrey@DESKTOP-UG1NENB:~$ nano MintPy_Install_Opti.sh

GNU nano 6.2                               MintPy_Install_Opti.sh *
#6 Clonar el repositorio de PyAPS para la corrección atmosférica
git clone https://github.com/insarlab/PyAPS.git --depth 1

#7 cdsapi
pip3 install cdsapi

sudo echo url: https://cds.climate.copernicus.eu/api/v2 > $HOME/.cdsapirc
sudo echo key: 12345 abcdefghij-134-abcdefqadf-82391b9d3f >> $HOME/.cdsapirc
          UID           API KEY
#8 Verificar la correcta instalación de ISO
python3 PyAPS/tests/test_dload.py
export DISPLAY=:0

```

Figura 14. Edición del bash de instalación de MintPy

Fuente: Elaboración propia

Si el software se instaló correctamente, en la pantalla al final de la instalación aparecen una serie de opciones y parámetros del script smallbaselineApp.py. De último y posterior a estos parámetros saldrá un mensaje que indicará que la instalación de MintPy fue completada. Luego, el bash prosigue con la instalación del algoritmo de corrección atmosférica PyAPS, una vez instalado aquí finaliza la función del bash (Figura 15). Para ver los parámetros de este script, simplemente se escribe en la terminal “smallbaselineApp.py --help”.



```

# step processing with --start/stop/dostep options
smallbaselineApp.py GalapagosSenDTI28.txt --dostep velocity # run step 'velocity' only
smallbaselineApp.py GalapagosSenDTI28.txt --end load_data # end run after step 'load_data'
Instalación de MintPy y smallbaselineApp.py completada
Iniciando instalación de PyAPS (modelo de corrección atmosférica ERA5) completada
Cloning into 'PyAPS'...
Passed tropospheric delay calculation from ERA5.
-----
Instalación de PyAPS (modelo de corrección atmosférica ERA5) completada
andrey@DESKTOP-UG1NENB: ~

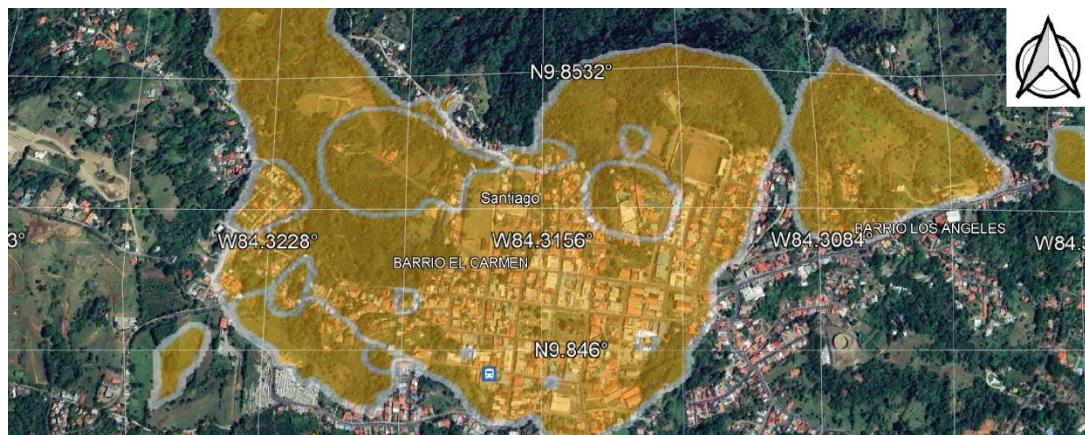
```

Figura 15. Visualización de la instalación de MintPy y ERA5 completada

Fuente: Elaboración propia

## 2.3 Área de estudio

En el caso de Costa Rica, sensores como el Sentinel-1, hacen un recorrido sobre la superficie del país tanto en órbita ascendente como descendente, permitiendo generar insumos SAR de todo el territorio nacional. El inconveniente se debe a las condiciones atmosféricas del país y a la extensa cobertura vegetal que posee. Por lo tanto, solo parte del territorio nacional exhibe valores de coherencia suficiente para ser útiles para detectar movimiento. Se debe delimitar una zona de estudio más específica para el análisis de los InSAR, dado que una imagen SAR abarca aproximadamente un 75% de Costa Rica (Figura 16). En el estudio del que se basa este manual, la delimitación se hace para el área de Santiago de Puriscal debido a que existen antecedentes de deslizamiento en esta región. Estas características presentan un escenario perfecto para la aplicación de InSAR.



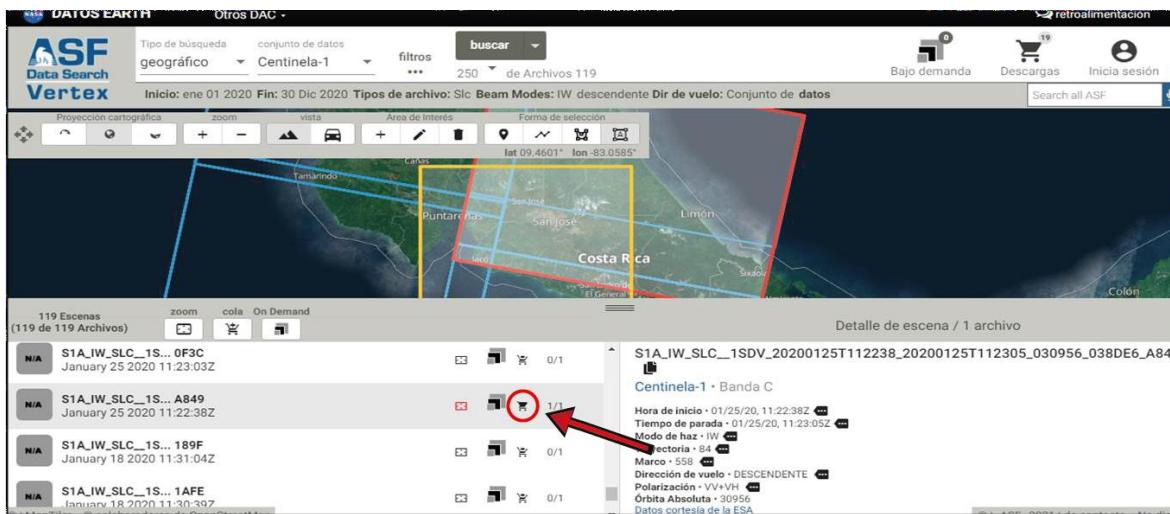
**Figura 16. Santiago de Puriscal con capa de deslizamientos del CNE**  
(Fuente: Elaboración propia, Mapa de Google Earth, capa deslizamientos CNE)

## 2.4 Descarga de imágenes SAR

Las imágenes SAR son el insumo principal del procesamiento InSAR, a partir de la resta de fase entre ellas es posible obtener valores de deformación de la superficie. Para la descarga de las imágenes SAR en formato complejo (SLC), se debe elegir un portal web con la disponibilidad de estos productos. La ASF es una página que permite la descarga de datos de las misiones Sentinel-1, además de varios otros satélites. Pertenece al Instituto Geofísico de la Universidad de Alaska que, adquiere, procesa, archiva y distribuye datos de radar de apertura sintética de satélites y sensores aerotransportados. Los datos son de libre acceso para los usuarios (EARTHDATA, NASA). Es importante dirigirse a la opción de iniciar sesión para poder crear una cuenta, así se podrá continuar con la descarga de datos. ASF Data Search: <https://search.asf.alaska.edu/#/>

Lo primero es señalar el área a trabajar, esto por medio de un rectángulo o cuadrado color amarillo que se dibuje sobre el área deseada. Posterior a esto se presiona 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 seleccionarlas (Figura 17). Luego se elige la imagen a descargar, y se añade en la opción del ícono de carrito para posteriormente descargar una carpeta comprimida de las imágenes, se señala en un círculo rojo la opción para añadir.



**Figura 17. Página Alaska Satellite Facility**  
(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 ve en la Figura 18. Aquí se podrán indicar los parámetros que se desean de la imagen. En “tipo de archivo” por lo general y en este caso se elige la opción de Single Look Complex L1 (SLC). Para modo de haz o escaneo se elige la opción de Interferometric Wide (IW). En polarización se escoge el tipo de polarizado en la onda, en este caso se deja en blanco. La dirección hace referencia a la toma de la imagen si es ascendente o descendente con respecto a la órbita del satélite. En subtipo se escoge el satélite Sentinel-1, del que desea obtener sus datos SA o SB. En ruta y marco se colocan los valores numéricos de las imágenes SAR, la ruta corresponde a la dirección de la trayectoria del satélite al capturar la imagen, y el marco se refiere a la imagen seleccionada en esa ruta. En la parte superior de la Figura 18, se comparte los parámetros usados para la descarga de imágenes. En el lado izquierdo se ven los datos para la órbita ascendente y en el lado derecho para descendente. Debajo se observa la imagen SAR que corresponde a estos parámetros.

The figure displays the Alaska Satellite Facility search interface. At the top, there are two search panels labeled "Filtros de Búsqueda". Each panel includes fields for "Fecha de Inicio" (1/1/2020) and "Fecha Final" (1/31/2023), and a "Búsqueda Estacional" checkbox. Below these are sections for "Filtros Adicionales" and "Filtros de Ruta y Marco". The "Filtros Adicionales" section shows filters for "L1 Single Look Compl..." (IW, Polarización), "1/14 tipos de archivos seleccionados", "1/9 modos de haz seleccionados", "0/8 polarizaciones seleccionadas", "Ascendente" (SA), and "Group ID". The "Filtros de Ruta y Marco" section shows coordinates: 92, 92, 26, 26. Buttons for "Borrar" and "Borrar área de búsqueda" are present. Below the search panels are two search results for "ACs". Each result shows a map of Central America with a red rectangular box indicating the search area. The first result is for "80 Fin: Feb 01 2023 Ruta: 92 - 92 Marco: 26 - 26 Tipos de archivo: SLC Modos de Haz: IW Dirección de vuelo:" and the second for "1 Fin: Feb 01 2023 Ruta: 84 - 84 Marco: 558 - 558 Tipos de archivo: SLC Modos de Haz: IW Dirección de vuelo:". Each result includes a "BÚSQUEDA" button, download icons, and file details: "S1A\_IW\_SLC\_1SDV\_20210519T234824\_20210519T234852\_037 964\_047B15\_9558 Sentinel-1 - C-Band" and "S1A\_IW\_SLC\_1SDV\_20200101T112239\_20200101T112306\_030 606\_0381A7\_4CA1 Sentinel-1 - C-Band". A note at the bottom of each result states "El Acceso A Estos Datos Requiere Que Inicie Sesión."

**Figura 18. Página Alaska Satellite Facility**  
(Fuente: Elaboración propia, captura de pantalla)

Cuando se escogen las imágenes de interés se dirige al ícono del carrito de compras en la parte superior derecha de la pantalla, para 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 en la parte derecha de la Figura 19, es importante remarcar que estos archivos tienen un peso de 4.25GB promedio. Sin embargo, para 2024 las imágenes poseen un peso de 7 GB aproximadamente. Una red consistente SBAS requiere de decenas de fechas SAR.

The figure shows the download interface for the Alaska Satellite Facility. At the top, it says "Descargas" and "19 archivos, 73,02 GB". Below this is a list of 19 files, each with a download icon (cloud with a checkmark) and a delete icon (X). The files are:

- S1A\_IW\_SLC\_1SDV\_20180312T112225\_20180312T112252\_0... 8CC6 - 4.00 GB  
Complejo L1 de aspecto único (SLC)
- S1A\_IW\_SLC\_1SDV\_20180920T112234\_20180920T112301\_0... 39CE - 4.08 GB  
Complejo L1 de aspecto único (SLC)
- S1A\_IW\_SLC\_1SDV\_20180511T234805\_20180511T234833\_0... 2917 - 4.09 GB  
Complejo L1 de aspecto único (SLC)

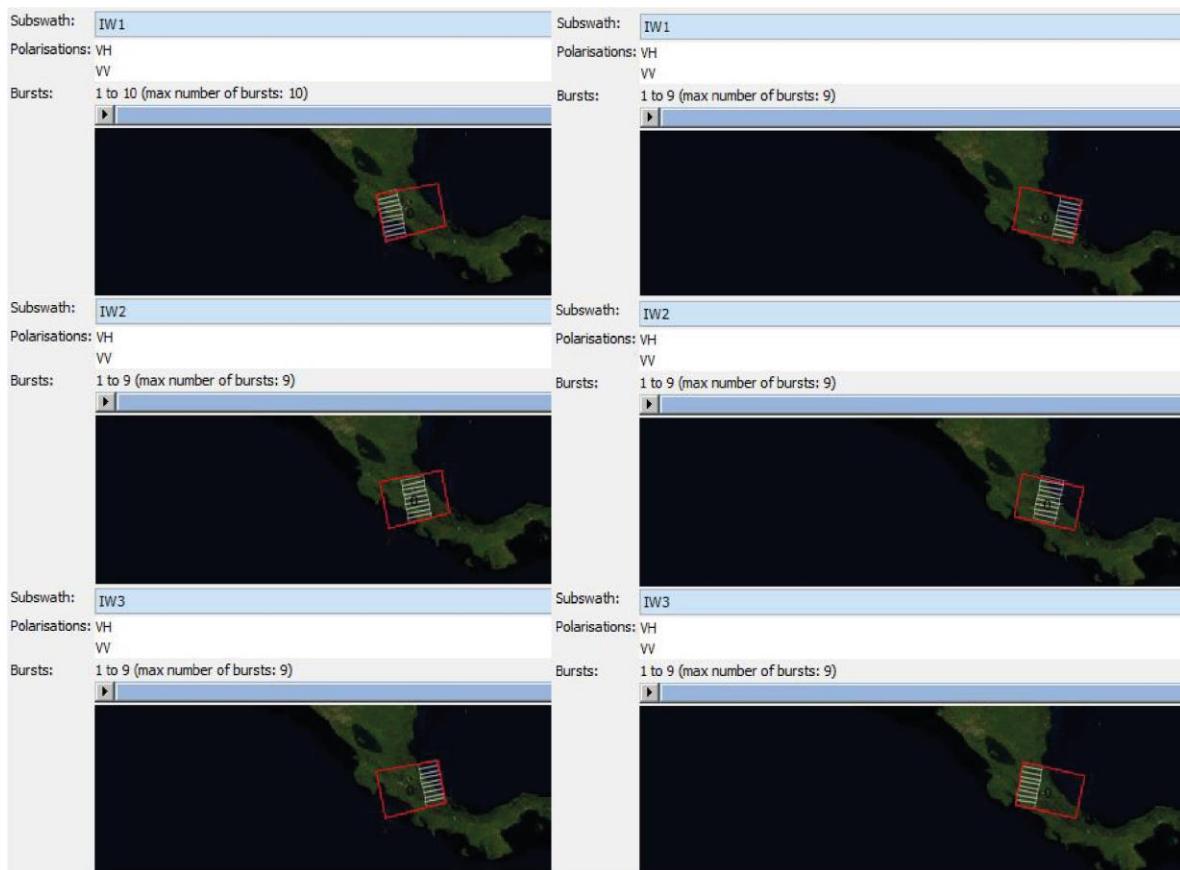
At the bottom, there are buttons for "claro", "Copiar IED de archivo", "Copiar direcciones URL", "Descarga de datos" (selected), and "Descarga de metadatos".

**Figura 19. Página Alaska Satellite Facility**  
(Fuente: Elaboración propia, captura de pantalla)

La imagen SAR en formato Single Look Complex (SLC), se conoce también como swath o banda en español. La misma se divide en tres subbandas, de los que es necesario conocer la numeración para indicar en el parámetro -n según la zona de interés. La numeración de las subbandas cambia según se la órbita de la imagen. A continuación, se muestra la Figura 20 donde se señalan los números de las subbandas para las imágenes SAR usadas en este proyecto.

La imagen SAR que se descarga no se procesa en su totalidad. El procesar sus tres subbandas y todas las ráfagas que las componen requieren de mayor tiempo y exige mayores capacidades de memoria para procesamiento y almacenamiento. Por esta razón se escoge una única subbanda que abarque la zona de interés, además con la delimitación del área a partir de las coordenadas de latitud y longitud, ISCE se encarga de delimitar o escoger solo las ráfagas necesarias para cubrir la zona.

Esto permite que el procesamiento requiera de menos tiempo y menos capacidades de memoria para procesamiento y almacenamiento. En el lado izquierdo se observa la imagen SAR ascendente y sus números de subbandas, y en el lado derecho la imagen descendente y su numeración respectiva.



**Figura 20. Números de subbandas en imágenes ascendente y descendente sobre Costa Rica**  
(Fuente: Elaboración propia, captura de pantalla en programa SNAP)

Las imágenes SAR se nombran de la siguiente manera (Figura 21):

sensor	Formato Imagen	año mes dia hora min
S1A_IW_SLC_	1SDV	20200605T234819_20200605T234847_032889_03CF3F_F84E.zip
modo de escaneo		

**Figura 21. Nombre de los archivos de imagen SAR SLC**

(Fuente: Elaboración propia, captura de pantalla)

## 2.5 Descarga de archivos de corrección: órbitas, DEM, datos atmosféricos

### 2.5.1 Órbitas

Los archivos de órbitas son archivos independientes a las imágenes SAR SLC, se pueden descargar de un servidor web usando un navegador, o desde la terminal con algún script adecuado para esto. Los archivos de órbita permiten introducir correcciones a las imágenes SAR, a causa de las perturbaciones en el satélite durante su trayecto. Estas perturbaciones afectan negativamente las líneas base entre adquisiciones. La descarga de órbitas se realiza de manera manual desde la página de Copernicus Open Acces Hub. En este trabajo se utilizó el script llamado “eof” que hace todo este proceso de descarga de forma automática. Esta rutina se descarga e implementa durante la instalación de ISCE con el bash ISCE\_Install\_bash.sh. En la Figura 22 se muestra las opciones y parámetros para utilizar eof, consultando con el comando --help.

```

andrey@DESKTOP-UG1NENB:~$ eof --help
Usage: eof [OPTIONS]

Download Sentinel precise orbit files.

Saves files to `save-dir` (default = current directory)

Download EOFs for specific date, or searches for Sentinel files in --path.
Will find both ".SAFE" and ".zip" files matching Sentinel-1 naming
convention. With no arguments, searches current directory for Sentinel 1
products

Options:
  -P, --search-path DIRECTORY      Path of interest for finding Sentinel
                                   products. [default: .]
  --save-dir DIRECTORY            Directory to save output .EOF files into
                                   [default: .]
  --sentinel-file PATH           Specify path to download only 1 .EOF for a
                                   Sentinel-1 file/folder
  -d, --date TEXT                Alternative to specifying Sentinel products:
                                   choose date to download for.
  -m, --mission [S1A|S1B]         If using '--date', optionally specify
                                   Sentinel satellite to download (default:
                                   gets both S1A and S1B)
  --orbit-type [precise|restituted] Optionally specify the type of orbit file to
                                   get (default: precise (POEORB), but fallback
                                   to restituted (RESORB))
  --force-asf                     Force the downloader to search ASF instead
                                   of ESA.
  --debug                         Set logging level to DEBUG
  --cdse-user TEXT               Copernicus Data Space Ecosystem username. If
                                   not provided the program asks for it
  --cdse-password TEXT           Copernicus Data Space Ecosystem password. If
                                   not provided the program asks for it
  --ASF-user TEXT                ASF username. If not provided the program
                                   asks for it
  --ASF-password TEXT            ASF password. If not provided the program
                                   asks for it
  --ask-password                 ask for passwords interactively if needed
  --update-netrc                  save credentials provided interactively in
                                   the ~/.netrc file if necessary
  --max-workers INTEGER          Number of parallel downloads to run. Note
                                   that CDSE has a limit of 4
  --help                          Show this message and exit.

andrey@DESKTOP-UG1NENB:~$
```

**Figura 22. Script eof para la descarga de órbitas automáticas**  
(Fuente: Elaboración propia, captura de pantalla)

Para usar esta rutina, solo se debe ir al directorio en donde están ubicadas las imágenes SAR, y ejecutar la descarga desde ese directorio. Solamente se debe indicar la ruta en donde se quiere que se descarguen las órbitas (Figura 23).

```

andrey@DESKTOP-UG1NENB:~$ cd SLC_2020_2023/
andrey@DESKTOP-UG1NENB:~/SLC_2020_2023$ ls
S1A_IW_SLC_1SDV_20200101T112339_20200101T112306_030606_0381A7_4CA1.zip
andrey@DESKTOP-UG1NENB:~/SLC_2020_2023$ chmod 777 *
andrey@DESKTOP-UG1NENB:~/SLC_2020_2023$ eof --save-dir /home/andrey/InSAR/orbits
[05/16 23:35:00] [INFO download.py] Downloading precise orbits for S1A on 2020-01-01
[05/16 23:35:00] [WARNING dataspace_client.py] No CDSE credentials found in netrc file. Please create one using https://dataspace.copernicus.eu/
[05/16 23:35:00] [WARNING download.py] Dataspace failed, trying ASF
[05/16 23:35:04] [INFO asf_client.py] Using cached EOF list
[05/16 23:35:05] [INFO asf_client.py] Downloading https://slqc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20210316T161714_V20191231T225942_20200102T005942.EOF
[05/16 23:35:23] [INFO asf_client.py] Saving to /home/andrey/InSAR/orbits/S1A_OPER_AUX_POEORB_OPOD_20210316T161714_V20191231T225942_20200102T005942.EOF
[05/16 23:35:23] [INFO download.py] Finished https://slqc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20210316T161714_V20191231T225942_20200102T005942.EOF, saved to /home/andrey/InSAR/orbits/S1A_OPER_AUX_POEORB_OPOD_20210316T161714_V20191231T225942_20200102T005942.EOF
andrey@DESKTOP-UG1NENB:~/SLC_2020_2023$
```

**Figura 23. Uso de la rutina eof para la descarga de órbitas**  
(Fuente: Elaboración propia, captura de pantalla)

Los archivos de órbita se nombran de la siguiente manera (Figura 24):

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

**Figura 24. Nombre de los archivos de órbita de imágenes SAR**  
(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\_POERB. 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.

## 2.5.2 DEM

El DEM contribuye al procesamiento InSAR en etapas como la eliminación de la fase topográfica. Por defecto, el software ISCE utiliza el DEM del sensor satelital SRTM, este DEM posee una resolución espacial en píxel de 30x30 metros. En este trabajo se usó para esta etapa el MDT del IGN, el cual se solicitó de manera formal a través de correo electrónico. El MDT del IGN a diferencia del SRTM (satelital) es resultado de un vuelo fotogramétrico sobre el país, por lo que su resolución espacial de píxel es de 10x10 metros. Este archivo se solicitó por medio de un correo electrónico al IGN, el MDT fue entregado en las instalaciones del IGN y en formato GeoTiff. Si el DEM es propio u otro diferente al SRTM, es necesario que se encuentren en formato GeoTiff.

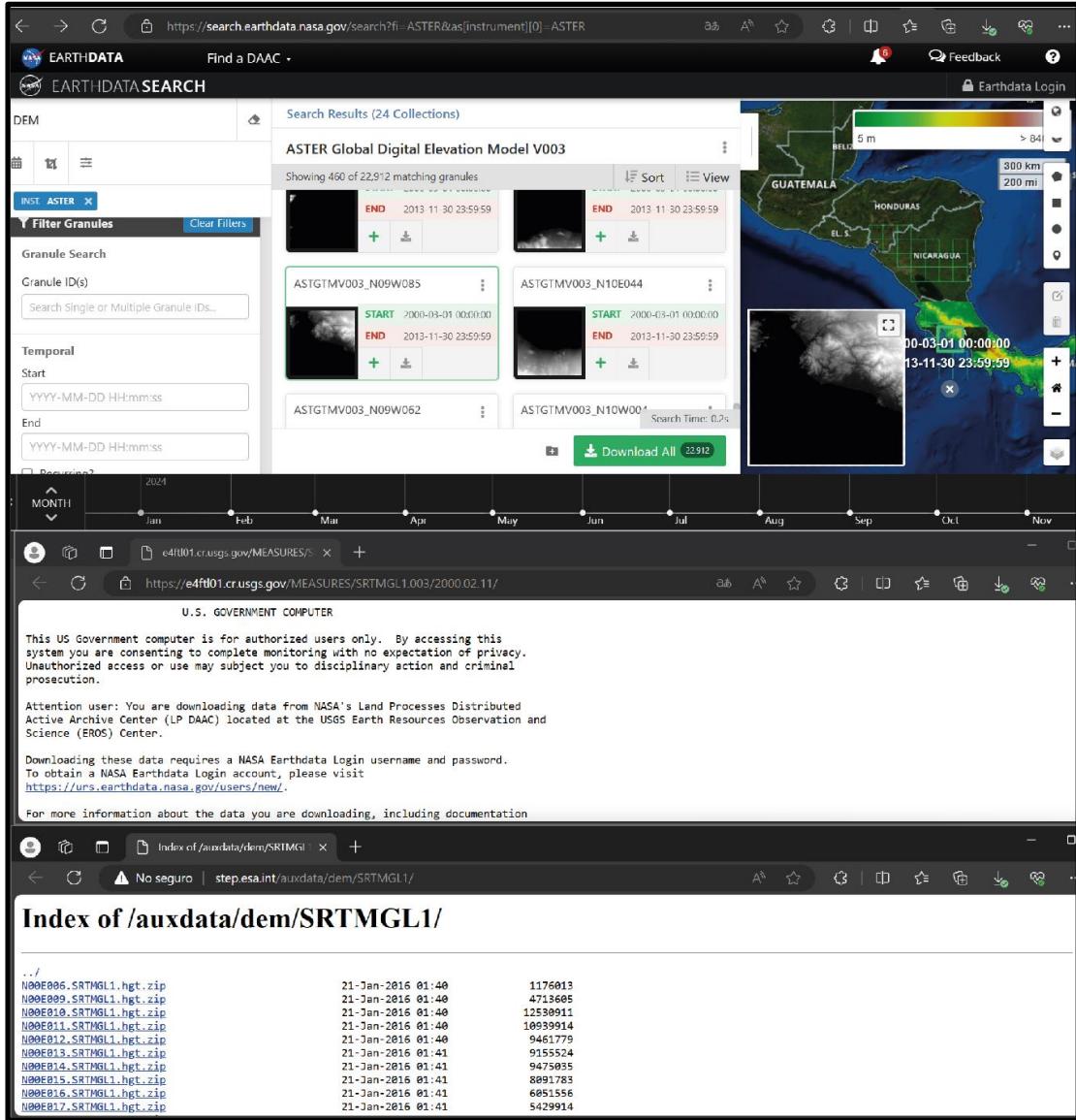
El software ISCE solo lee DEMs en su propio formato, de lo contrario generará errores. En la Figura 25 se muestra el proceso para convertir el MDT\_IGN.tif a archivos con formato legible por ISCE. El primer paso es pasar el archivo .tif a archivo .dem usando gdal\_translate, luego se crea un archivo VRT usando gdalbuildvrt. Esta herramienta construye un archivo .vrt a partir del archivo .dem, el archivo .vrt contiene la metadata del DEM o MDT usado. Posteriormente, se aplican de nuevo las mismas dos herramientas de gdal, esto para crear un archivo .dem con extensión WGS84, es decir .wgs84. El formato aceptado en entrada en ISCE para leer los DEM o MDT. El archivo encerrado en el rectángulo rojo es el producto final que genera gdal\_translate a partir del DEM o MDT en formato .tif. El mdt\_ign.dem.wgs84 (cuadro rojo) es el principal insumo para ISCE, este es el que se debe indicar en los parámetros del script stackSentinel.py.

```
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ ls
MDT IGN.tif
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ gdal_translate -of ISCE MDT IGN.tif mdt ign dem
Input file size is 10801, 10801
0...10...20...30...40...50...60...70...80...90...100 - done.
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ ls
MDT IGN.tif mdt ign dem mdt ign dem aux.xml mdt ign dem xml
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ gdalbuildvrt mdt ign dem.vrt mdt ign dem
0...10...20...30...40...50...60...70...80...90...100 - done.
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ gdal_translate -of ISCE MDT IGN.tif mdt ign dem.wgs84
Input file size is 10801, 10801
0...10...20...30...40...50...60...70...80...90...100 - done.
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ ls
MDT IGN.tif mdt ign dem aux.xml mdt ign dem.wgs84 mdt ign dem.wgs84.xml
mdt ign dem mdt ign dem.vrt mdt ign dem.wgs84.aux.xml mdt ign dem.xml
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ gdalbuildvrt mdt ign dem.wgs84.vrt mdt ign dem.wgs84
0...10...20...30...40...50...60...70...80...90...100 - done.
andrey@DESKTOP-UG1NENB:~/MDT IGN_$ ls
MDT IGN.tif mdt ign dem aux.xml mdt ign dem.wgs84 mdt ign dem.wgs84.vrt mdt ign dem.xml
mdt ign dem mdt ign dem.vrt mdt ign dem.wgs84.aux.xml mdt ign dem.wgs84.xml
andrey@DESKTOP-UG1NENB:~/MDT IGN_$
```

**Figura 25. Conversión de DEM o MDT en formato GeoTIFF a formato de ISCE**  
(Fuente: Elaboración propia, captura de pantalla)

Se puede utilizar el DEM del SRTM que usa ISCE por defecto, o el DEM ASTER, ambos con la misma resolución espacial de 30 metros. Para usar el DEM ASTER se debe buscar y descargar desde la página de la EarthData de la NASA en formato TIFF. Esto se encuentra en la primera dirección web de la . Luego, se aplica el mismo procedimiento anterior para convertir el ASTER.tif a formato legible por ISCE.

Por último, se puede utilizar el DEM SRTM. Este requiere inicialmente de la descarga de unos archivos llamados tile. Estos archivos se encuentran en formato .zip, y su nombre inicia con las coordenadas que abarca el archivo, se deben buscar y descargar los archivos que contengan el área de interés. La descarga de estos archivos tile SRTM se puede hacer desde la segunda y tercera dirección web de la Figura 26.



**Figura 26. Página Earthdata para la descarga de DEM**

(Fuente: Elaboración propia, captura de pantalla)

Cuando ya se han descargado los tiles, y se tienen en su directorio respectivo, dentro de esta misma carpeta se ejecuta el script dem.py. La función del script dem.py es construir el DEM a partir de los archivos tile que han sido descargados previamente. Los archivos tile descargados abarcan un área comprendida entre los 9° y 10° de latitud norte, y los 83° y 85° de longitud oeste. Para otorgar todos los permisos de acceso a los archivos "tile", se aplica el comando chmod 777 \*, lo cual es representado en color rojo en la Figura 27. Esta acción es esencial para evitar errores de permisos en los archivos.

Posteriormente, se ejecuta el script dem.py utilizando los parámetros mostrados en la Figura 27. Tras la ejecución, se generan varios archivos, similar al procedimiento anterior, siendo el archivo principal el que tiene la extensión WGS84. El nombre del archivo final se puede

observar en la línea de "API close", en este caso es demLat\_N09\_N10\_Lon\_W085\_W083.dem.wgs84. Este archivo DEM (Modelo Digital de Elevación) será utilizado como insumo por el script stackSentinel.py durante el procesamiento InSAR.

```

andrey@DESKTOP-UG1NENB:~/DEM_SRTM$ ls
N09W083.SRTMGL1.hgt.zip N09W084.SRTMGL1.hgt.zip N09W085.SRTMGL1.hgt.zip N10W084.SRTMGL1.hgt.zip isce.log
andrey@DESKTOP-UG1NENB:~/DEM_SRTM$ chmod 777 *
andrey@DESKTOP-UG1NENB:~/DEM_SRTM$ dem.py -a stitch -b 9 10 -85 -83 -r -s 1 -c
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
API open (R): ./demLat_N09_N10_Lon_W085_W083.dem
API close: ./demLat_N09_N10_Lon_W085_W083.dem
Writing geotrans to VRT for ./demLat_N09_N10_Lon_W085_W083.dem
GDAL open (R): ./demLat_N09_N10_Lon_W085_W083.dem.vrt
API open (WR): demLat_N09_N10_Lon_W085_W083.dem.wgs84

<< Geoid Correction I2 SRTM>>

Jet Propulsion Laboratory - Radar Science and Engineering

Sampling Geoid at grid points - Longitude Samples: 23 Latitude Lines: 13
Corner Geoid Heights (m) = 14.96 6.75 10.42 8.17

Correcting data to geoid height...

At line: 512
At line: 1024
At line: 1536
At line: 2048
At line: 2560
At line: 3072
At line: 3584
GDAL close: ./demLat_N09_N10_Lon_W085_W083.dem.vrt
API close: demLat_N09_N10_Lon_W085_W083.dem.wgs84
API open (R): demLat_N09_N10_Lon_W085_W083.dem.wgs84
API close: demLat_N09_N10_Lon_W085_W083.dem.wgs84
Writing geotrans to VRT for demLat_N09_N10_Lon_W085_W083.dem.wgs84
Writing geotrans to VRT for /home/andrey/DEM_SRTM/demLat_N09_N10_Lon_W085_W083.dem.wgs84
N09W085.SRTMGL1.hgt.zip = succeeded
N09W084.SRTMGL1.hgt.zip = succeeded
andrey@DESKTOP-UG1NENB:~/DEM_SRTM$ 
```

**Figura 27. Página Alaska Satellite Facility**  
(Fuente: Elaboración propia, captura de pantalla)

### 2.5.3 Datos atmosféricos

Los datos atmosféricos son los últimos archivos por descargar, estos datos permiten en el procesamiento de series temporales realizar una corrección al ruido generado por estas influencias. Como, por ejemplo, el retraso troposférico en la onda que corrige MintPy. En este trabajo, se utilizó el algoritmo de PyAPS que usa los datos del modelo atmosférico ERA5, para el paso de la corrección troposférica ([Jolivet et al., 2011](#)). Los datos se descargan automáticamente dentro del directorio llamado ERA5, que también es creado de manera automática durante la ejecución de smallbaselineApp.py. El directorio ERA5 se crea dentro de la carpeta donde se hace el procesamiento de series temporales. También se pueden utilizar

los datos atmosféricos del servidor de GACOS ([Yu, et al., 2018](#)), pero estos requieren de una descarga manual.

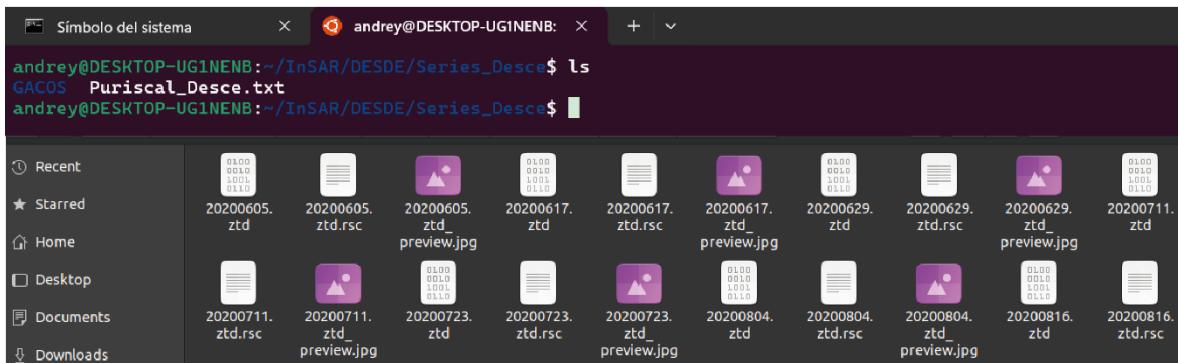
Si en los parámetros se escoge la corrección troposférica usando GACOS, entonces se debe ir a la página de GACOS para realizar la descarga manual de los archivos troposféricos. Cuando ingrese a la página aparecerá una pantalla como la de la Figura 28, allí se pondrán las coordenadas en latitud y longitud del área de interés. Se recomienda poner la misma área del DEM o las coordenadas puestas en los parámetros de bbox. En los datos de insumo de la página se 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 utilizará, el formato de salida se selecciona cuadrícula binaria. Finalmente se escribe el correo al que desea que le llegue el vínculo de descarga con los datos GACOS. Esto puede tardar unos minutos. Una vez le llegue el correo con el enlace, ingresará a este y así iniciara la descarga.

**Figura 28. Diagrama de flujo de trabajo MintPy**

Fuente: Elaboración propia

En la carpeta de descargas se descarga un archivo comprimido que contiene a la carpeta con todos los archivos GACOS. Se deberá descomprimir y mover a la carpeta “Series\_Desce”

donde se ejecutará MintPy. Es recomendable cambiar el nombre de la carpeta descomprimida a GACOS, ya que se descarga con otro nombre diferente. Haciendo esto la carpeta de ejecución ya no solo tendrá el archivo de texto con los parámetros y rutas, sino que estará también el directorio GACOS con los archivos de corrección troposférica. El contenido de la carpeta GACOS se verá de manera gráfica como en la Figura 29, según la cantidad imágenes con que trabaje, será la cantidad de archivos GACOS que habrá.

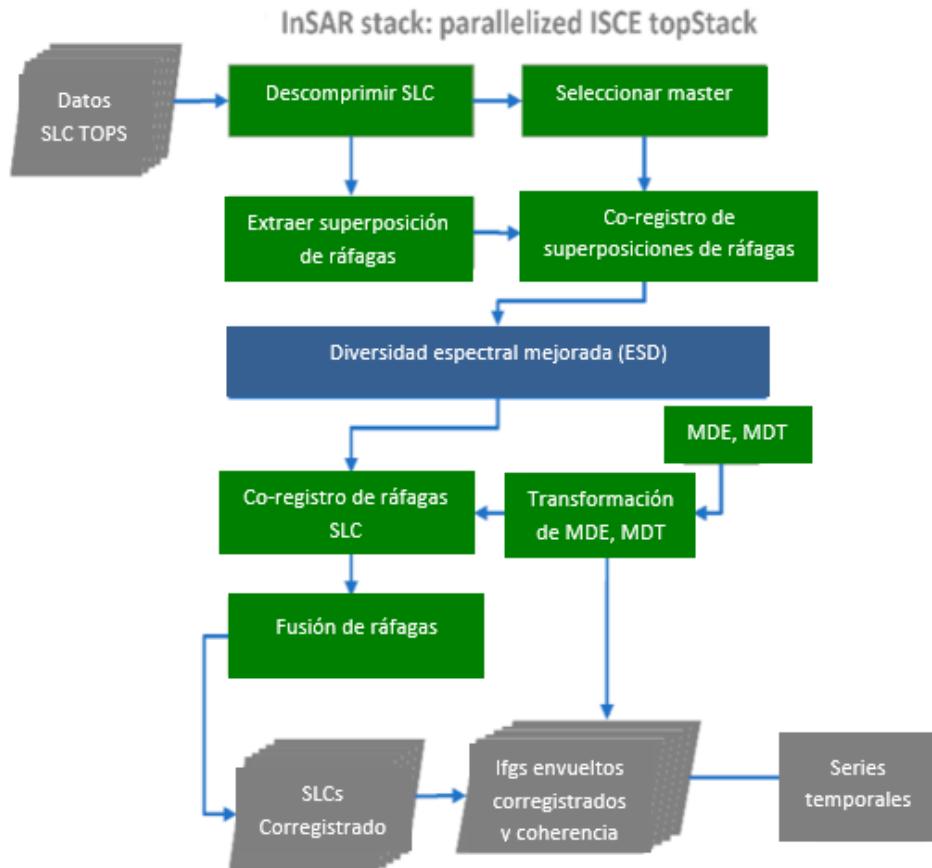


**Figura 29. Archivos de corrección troposféricos para MintPy**

Fuente: Elaboración propia

## 2.6 Procesamiento con ISCE usando el módulo de topsStack stackSentinel.py

Cuando se tienen listos todos los archivos de insumos, se procede con iniciar el procesamiento InSAR, haciendo uso de ISCE. El script stackSentinel.py es el principal dentro del módulo topsStack de ISCE, es el encargado de realizar el procesamiento de pila, tiene varios flujos de trabajo como pila de interferogramas y pila de SLC. En la Figura 30 se visualiza el diagrama de trabajo de topsStack, partiendo de los productos SAR de SLC en el modo de escaneo TOPSAR. Por medio del stackSentinel.py se indican las variables de insumo al momento de invocar el comando, entre los principales están el directorio de imágenes SLC, directorio de órbitas precisas, directorio del DEM a utilizar, número de subbanda a procesar y área de interés por medio de coordenadas geodésicas. Es necesario verificar que el área de interés este dentro de la banda seleccionada.



**Figura 30. Diagrama de flujo de trabajo módulo topsStack**

Fuente: Figura modificada de Papoutsis I, Kontoes C, Alatzas S, Apostolakis A, Loupasakis C (2020)

Otras variables que se pueden indicar en comando antes de ser ejecutado, son la geometría, la imagen máster, el número de conexiones entre imágenes, las resoluciones para rango y acimut (range look, azimuth look) entre otros. Los diferentes flujos de trabajo que se pueden ejecutar con `stackSentinel.py` se observan con la ejecución en la terminal de “`stackSentinel.py -H`”, como en la Figura 31 . A continuación, se indica el significado de los parámetros principales para este script.

- s: se refiere al directorio donde se encuentran las imágenes SAR.
- o: indica el directorio donde están las órbitas.
- 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 a 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.

```

andrev@DESKTOP-UGLINEO: $ stackSentinel.py -H
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

Stack processor for Sentinel-1 data using ISCE software.

For a full list of different options, try stackSentinel.py -h

stackSentinel.py generates all configuration and run files required to be executed for a stack of Sentinel-1 TOPS data.

Following are required to start processing:

1) a folder that includes Sentinel-1 SLCs,
2) a DEM (Digital Elevation Model)
3) a folder that includes precise orbits (use dloadOrbits.py to download/ update your orbit folder. Missing orbits downloaded on the fly.)
4) a folder for Sentinel-1 Aux files (which is used for correcting the Elevation Antenna Pattern).

Note that stackSentinel.py does not process any data. It only prepares a lot of input files for processing and a lot of run files. Then you need to execute all those generated run files in order. To know what is really going on, after running stackSentinel.py, look at each run file generated by stackSentinel.py. Each run file actually has several commands that are independent from each other and can be executed in parallel. The config files for each run file include the processing options to execute a specific command/function.

Note also that run files need to be executed in order, i.e., running run_03 needs results from run_02, etc.

#####
#Examples:

stackSentinel.py can be run for different workflows including: a stack of interferogram, a stack of correlation files, a stack of offsets or a coregistered stack of SLC. Workflow can be chosen with -W option.

#####
Example 1:
# interferogram workflow with 2 nearest neighbor connections (default coregistration is NESD):
stackSentinel.py -s ../SLC/ -d ../../MexicoCity/demLat_N18_N20_Lon_W100_W097.dem.wgs84 -b '19 20 -99.5 -98.5' -a ../../AuxDir/ -o ../../Orbits -c 2

#####
Example 2:
# interferogram workflow with all possible interferograms and coregistration with only geometry:
stackSentinel.py -s ../SLC/ -d ../../MexicoCity/demLat_N18_N20_Lon_W100_W097.dem.wgs84 -b '19 20 -99.5 -98.5' -a ../../AuxDir/ -o ../../Orbits -C geometry -c all

```

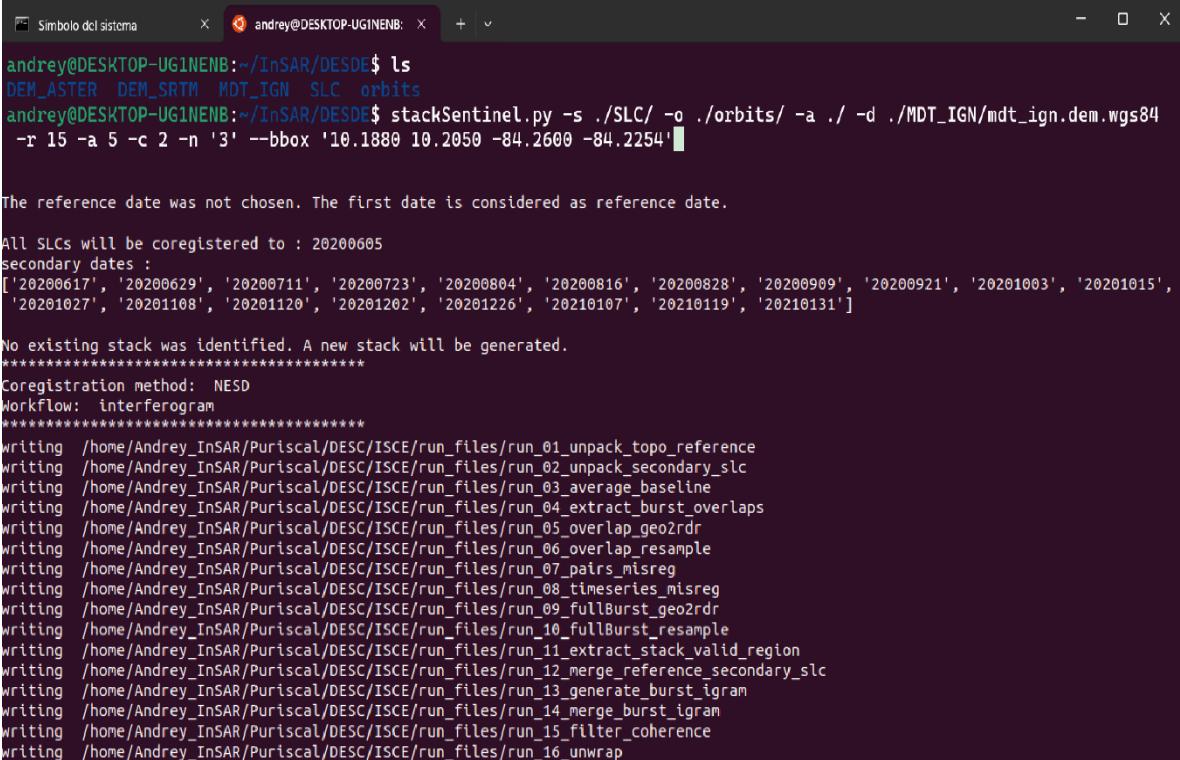
**Figura 31. Flujos de trabajo módulo stackSentinel.py**

Fuente: (Elaboración propia)

Si no se señalan estas variables, el procesamiento usará los valores que trae por defecto para cada parámetro, para la imagen máster el guion elige la imagen de la primera fecha, la más antigua. Finalmente, el script extrae las ráfagas de interés de los subbandas de las imágenes para el solapamiento entre ellas, donde el número de ráfaga a procesar dependerá del área abarcada por el sitio de estudio. Generalmente se ocupa de un mínimo de 3 ráfagas para llevar a cabo el procesamiento, del total de 8 que posee cada subbanda. Entre más ráfagas y subbandas más tiempo conlleva el procesamiento. Se realiza la corrección geométrica que por defecto es a través del ESD, más las contribuciones del DEM, remoción topográfica, y las correcciones posteriores. El producto principal obtenido finalmente del cálculo es una carpeta llamada merged. En ella se almacenan los interferogramas, las imágenes SLC y la geometría de referencia.

Antes de ejecutar el script se debe verificar que todos los insumos que utilizará stackSentinel.py están en los directorios o rutas correctos. En la Figura 32, se muestra el directorio donde se organizaron todos archivos de insumo para la ejecución. Estos corresponden a la carpeta con el DEM, en esta ocasión se usó el MDT del IGN, la carpeta con las imágenes SAR en formato SLC, y la carpeta con las órbitas de las imágenes SAR. Posteriormente se escribe stackSentinel.py junto con los parámetros correspondientes a cada archivo de insumo y su ruta respectiva de ubicación. Por esta razón, se recomienda ubicar los archivos de insumos en la misma carpeta o en rutas de fácil acceso. Cuando se termina el

proceso inicial de stackSentinel.py, se crean varios archivos ejecutables llamados “run\_files”, esos son archivos ejecutables que se deben ir corriendo uno por uno desde la terminal.



```

andrey@DESKTOP-UG1NENB:~/InSAR/DESDE$ ls
DEMASTER DEM_SRTM MDT_IGN SLC orbits
andrey@DESKTOP-UG1NENB:~/InSAR/DESDE$ stackSentinel.py -s ./SLC/ -o ./orbits/ -a ./ -d ./MDT_IGN/mdt_ign.dem.wgs84
-r 15 -a 5 -c 2 -n '3' --bbox '10.1880 10.2050 -84.2600 -84.2254'

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_geo2rdr
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_geo2rdr
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 32. Diagrama de flujo de trabajo módulo topsStack**

Fuente: Elaboración propia

Los archivos ejecutables llamados “run\_files” son los encargados de realizar el procesamiento InSAR, son el paso a paso del procesamiento InSAR que utiliza stackSentinel.py. Para acelerar el proceso manual de ejecución de los archivos ejecutables llamados “run\_files”, es posible hacer una automatización dejando programada la ejecución continua de cada run file dentro de un archivo bash. Inicialmente se crea un archivo ejecutable bash y se introduce el nombre en orden de cada uno de los archivos “run\_files”, con un “./” delante del nombre de cada uno. El archivo bash al igual que los archivos ejecutables para la instalación se crean con el comando “nano”, el nombre y la extensión SH. Después de crear y guardar el archivo bash, se habilitan los permisos de ejecución en caso de ser necesario, de lo contrario se ejecuta directamente (Figura 33).

```

Símbolo del sistema andrey@DESKTOP-UG1NENB: ~ andrey@DESKTOP-UG1NENB: ~ andrey@DESKTOP-UG1NENB: ~
andrey@DESKTOP-UG1NENB: ~$ cd InSAR/
andrey@DESKTOP-UG1NENB: ~/InSAR$ cd DESDE/
andrey@DESKTOP-UG1NENB: ~/InSAR/DESDE$ nano bash_stack.sh
GNU nano 6.2                                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_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

^G Help      ^O Write Out   ^W Where Is    ^K Cut          ^T Execute      ^C Location     M-U Undo
^X Exit       ^R Read File    ^V Replace     ^U Paste        ^J Justify      ^I Go To Line   M-E Redo
                                         M-A Set Mark
                                         M-G Copy

andrey@DESKTOP-UG1NENB: ~/InSAR/DESDE$ ./bash_stack.sh

```

**Figura 33. Ejecución de archivos run-file automáticamente con archivo bash**

Fuente: Elaboración propia

Cuando stackSentinel.py ha terminado de procesar resultaran en el directorio de ejecución varias carpetas nuevas y un archivo de texto véase la siguiente lista.

-reference	-run_files
-secondaries	-misreg
-interferograms	-coarse_interferograms
-ESD	-coreg_secondaries
-merged	-stack
-baselines	-geom_reference
-configs	-SAFE_files.txt

En la carpeta merged se encuentran los resultados finales del procesamiento, esta contiene 3 productos en 3 directorios diferentes interferograms, SLC y geom\_referenc. Los productos principales están dentro del directorio interferograms, estos son filt\_fine.unw, filt\_fine.cor y filt\_fine.unw.conncmp. Estos interferogramas se visualizan ejecutando el comando mdx.py. Del directorio geom\_reference MintPy usa de insumo los archivos hgt.rdr, lat.rdr, lon.rdr, los.rdr y shadowMask.rdr. Los archivos corresponden a metadatos de la geometría de referencia de los interferogramas, como la latitud, la longitud, los datos de la línea de visión del satélite (LOS) y la máscara de sombra. El software MintPy usa otros tres directorios como datos de insumo para la generación de series temporales de SBAS, los cuáles son el directorio baseline, reference y secondarys.

## 2.7 Posibles errores al ejecutar stackSentinel.py

### 2.7.1 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 34, 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 34. 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 35, 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
```

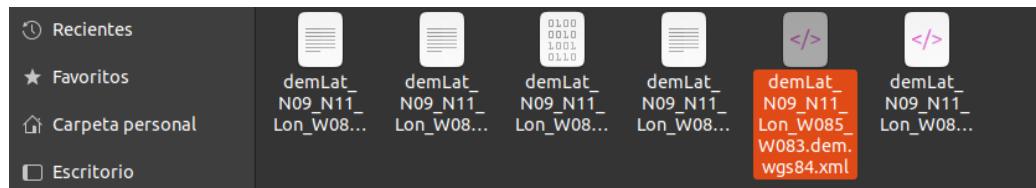
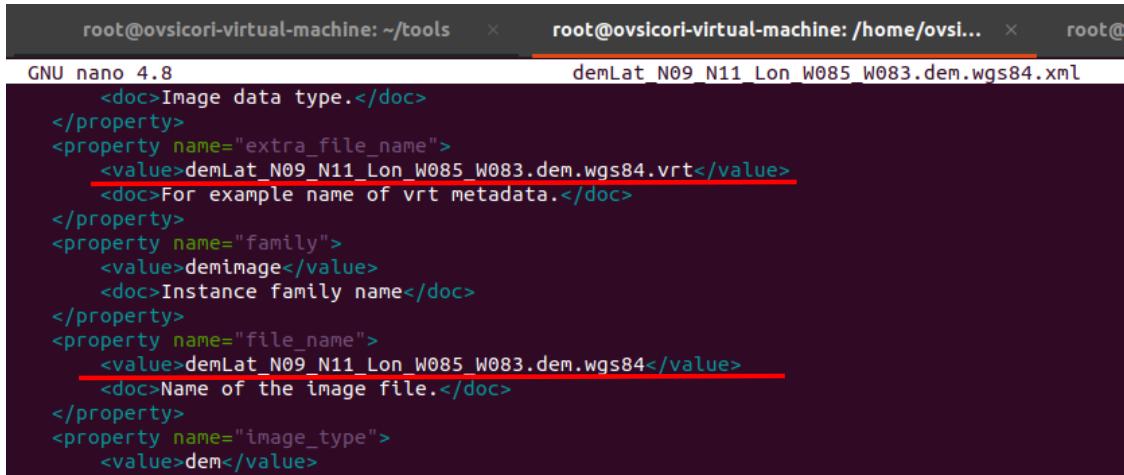


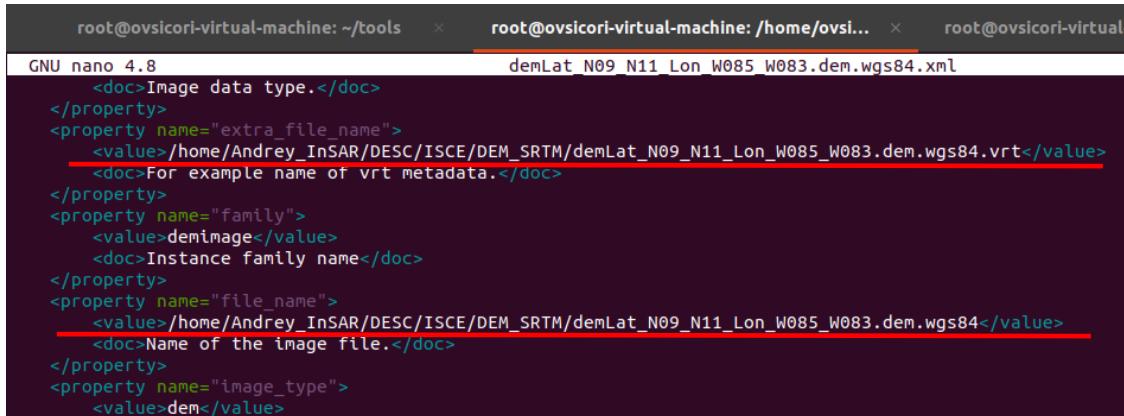
Figura 35. Contenido de la carpeta DEM  
(Fuente: Elaboración propia)

Al ingresar al archivo y bajar un poco le aparecerá lo que muestra Figura 36, 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 37.



```
root@ovsicori-virtual-machine: ~/tools          root@ovsicori-virtual-machine: /home/ovsi...          root@ovsicori-virtual-machine: ~
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 36. Metadata del archivo .dem.wgs8.xml  
(Fuente: Elaboración propia)



```
root@ovsicori-virtual-machine: ~/tools          root@ovsicori-virtual-machine: /home/ovsi...          root@ovsicori-virtual-machine: ~
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 37. 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.

## 2.7.2 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 38 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://winspace.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_dlr)
  File "/usr/local/isce/contrib/topsStack/stackSentinel.py", line 1048, in get_orbit
    self.orbit = orbitfile[0]
IndexError: list index out of range
```

Figura 38. 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 39 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.

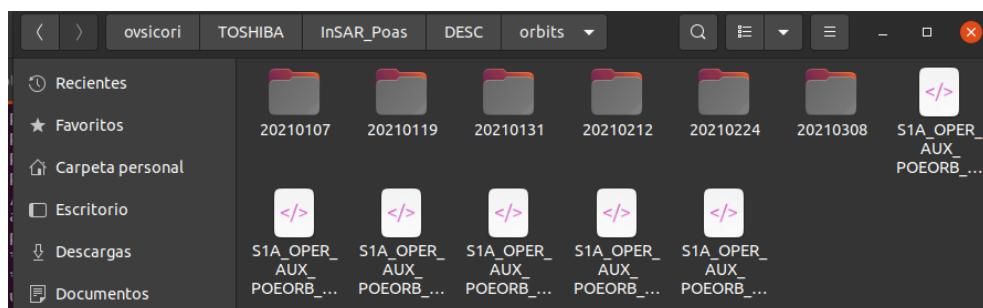
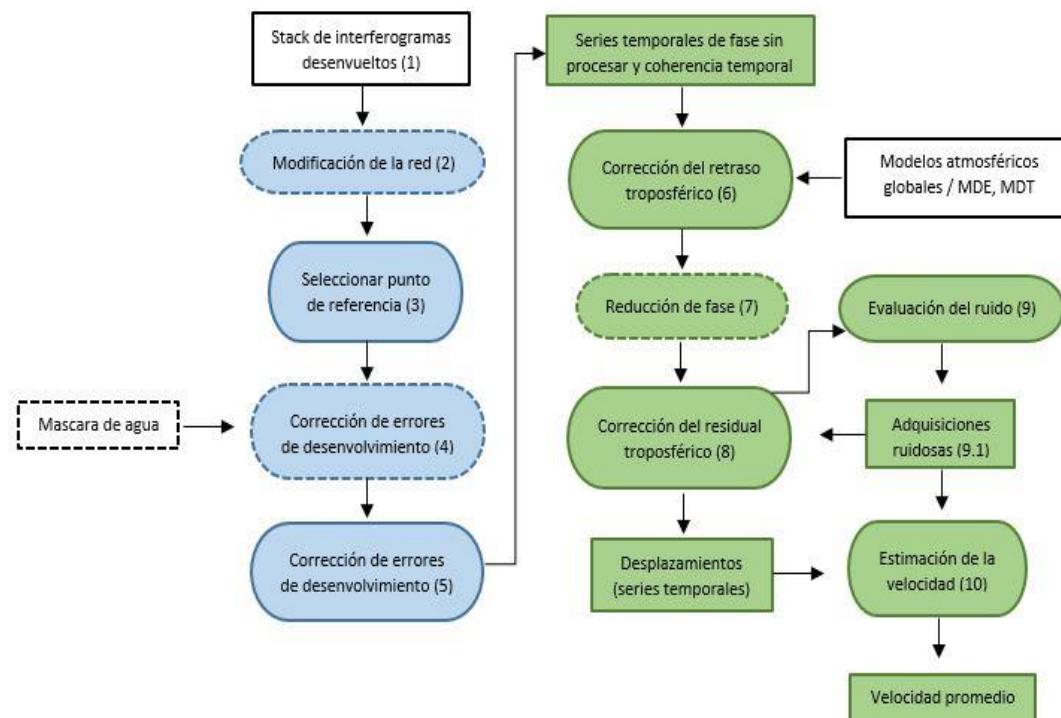


Figura 39. Carpeta con orbitas precisas  
(Fuente: Elaboración propia, captura de pantalla)

## 2.8 Procesamiento de series temporales con MintPy usando smallbaselineApp.py

Se parte de la pila de interferogramas desenvueltos generada por ISCE, tal y como lo indica el diagrama de flujo de la Figura 40. Con respecto al topsStack, MintPy utiliza las siguientes carpetas como insumo baselines, merged, reference y secondarys. Con un archivo de texto se indican las direcciones de las carpetas, y se indican los parámetros de interés como el punto de referencia, a partir del cual MintPy calcula las diferencias de elevación. La máscara de agua es un parámetro opcional pero recomendado para la corrección del error en fase unwrapped. Para el parámetro de la corrección por retraso troposférico, el método se hace usando insumos de modelos atmosféricos globales, como los pueden ser datos de GACOS, o datos de ERA5 por medio de PyAPS. Posterior a las evaluaciones de ruido en las adquisiciones, la corrección topográfica, y las series temporales se obtiene el promedio de velocidad en el desplazamiento de la superficie para la dirección en LOS y en unidades de cm/año.



**Figura 40. Diagrama de flujo de trabajo MintPy**

Fuente: Figura modificada de Yunjun, Z., Fattah, H., and Amelung, F (2019)

Para la ejecución del script `smallbaselineApp.py` se utiliza un archivo de texto “Puriscal\_Desce.txt”, en este archivo de texto se indican los parámetros y rutas de los archivos de insumo. Este archivo de texto se ejecuta desde el directorio donde se guardarán todos los productos de MintPy. Se recomienda crear esta carpeta dentro del directorio donde se generaron los resultados de ISCE. En la Figura 41, se muestran los pasos a realizar antes

de iniciar el procesamiento con MintPy, donde lo primero es crear el directorio donde se ejecutará el script. En esta ocasión se le denomina a la carpeta “Series\_Desce”, dentro de ella debe estar el archivo de texto con las rutas y parámetros. Este formato de datos de insumo dentro del TXT se puede encontrar en la página de github de MintPy, o en la carpeta de “configs” dentro del directorio MintPy que se encuentra en \$HOME. Para editar el archivo de texto se ejecuta el comando nano y se configura cada uno de los parámetros que se observan en la Figura 41.

```

andrey@DESKTOP-UG1NENB:~/InSAR/DESDE/Series_Desce$ ls
Puriscal_Desce.txt
andrey@DESKTOP-UG1NENB:~/InSAR/DESDE/Series_Desce$ nano Puriscal_Desce.txt

GNU nano 6.2                                     Puriscal_Desce.txt
#####
##----- 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.lookupVFile = ../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
#Coordenadas en latitud y longitud del punto de referencia a partir del que se medirán las diferencias (Lat, Long).
mintpy.reference.lalo = 9.8415,-84.3054
#Coordenadas en latitud y longitud de los vértices para el área de interés.(LaS:LaN,LoW:LoE)
mintpy.subset.lalo = 9.839.855,-84.34:-84.305
#mintpy.reference.date = reference_date.txt

##-----tropospheric correct
mintpy.troposphericDelay.method = auto [pyaps / height_correlation / gacos / no], auto for pyaps
#Se escoge solo uno de los dos siguientes métodos para la corrección troposférica:
#Si se escoge pyaps, entonces se añaden las siguientes líneas de texto:
mintpy.troposphericDelay.weatherModel = auto #[ERA5 / MERRA / NARR], auto for ERA5
mintpy.troposphericDelay.weatherDir = .
#Si se escoge gacos entonces se añade la siguiente linea de texto:
mintpy.troposphericDelay.gacosDir = ./GACOS

```

Figura 41. Archivo de texto de insumo Puriscal\_Desce.txt

Fuente: Elaboración propia

Si el procesamiento de MintPy inicia de forma correcta, entonces en la pantalla aparecerá lo mostrado en la Figura 42. En este inicio se observa la ubicación del directorio de ejecución, seguido de la rutina de pasos que sigue y ejecuta smallbaselineApp.py.

```

## Símbolo del sistema andrey@DESKTOP-UGINENB: /InSAR/DESDE/Series_Desce$ smallbaselineApp.py Puriscal_Desce.txt

## Miami InSAR Time-series software in Python
## MintPy 1.6.5, 2024-05-09

--RUN-at-2024-05-22 14:52:23.119114-
Current directory: /home/andrey/InSAR/DESDE/Series_Desce
Run routine processing with smallbaselineApp.py on steps: ['load_data', 'modify_network', 'reference_point', 'quick_overview', 'correct_unwrap_error', 'invert_network', 'correct_LDD', 'correct_SET', 'correct_ionosphere', 'correct_troposphere', 'deramp', 'correct_topography', 'residual_RMS', 'reference_date', 'velocity', 'geocode', 'google_earth', 'hdfEOS5']
Remaining steps: ['modify_network', 'reference_point', 'quick_overview', 'correct_unwrap_error', 'invert_network', 'correct_LDD', 'correct_SET', 'correct_ionosphere', 'correct_troposphere', 'deramp', 'correct_topography', 'residual_RMS', 'reference_date', 'velocity', 'geocode', 'google_earth', 'hdfEOS5']


```

**Figura 42. Inicio de ejecución de MintPy**

Fuente: Elaboración propia

Cuando el procesamiento de MintPy finaliza de manera exitosa, en la pantalla aparece lo que se visualiza en la Figura 43. El mensaje final indica “Finalización normal del procesamiento de smallbaselineApp”. En la parte superior de este mensaje se muestra la serie de scripts que se usan para ver los metadatos de los archivos resultantes, para visualizar perfiles 1D en matrices 2D, para visualizar mapas 2D de las series temporales y para guardar en formato de Google Earth. Finalmente se muestran desde la terminal de comandos, dentro de la carpeta de ejecución se verán los resultados del procesamiento de series temporales, dentro de ella estarán 4 directorios. El resto de los productos que se ven en la Figura 43 son de extensión HDF, los cuales se visualiza con view.py.

```

ovsicori@python199: /INSAR/directorio x ovsicori@python199: /INSAR/Otros/Andrey_t... x ovsicori@python199: /INSAR/Otros/Andrey_t... x
view.py --dpi 150 --noverbose --nodisplay --update --memory 4.0 geo/geo_timeseries_ERAS5_ramp_demErr.h5 --noaxis -u cm --wrap --wrap-range -5 5
view.py --dpi 150 --noverbose --nodisplay --update --memory 4.0 velocityERAS5.h5 --mask no
view.py --dpi 150 --noverbose --nodisplay --update --memory 4.0 numInvIfgram.h5 --mask no
copy *.txt files into ./pic directory.
move *.png/pdf/kmz files to ./pic directory.
time used: 04 mins 54.6 secs.
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: /INSAR/Otros/Andrey_tesis_Puriscal/DESCE/series

#####
Normal end of smallbaselineApp processing!
#####
Time used: 04 mins 56.0 secs

ovsicori@python199:/INSAR/Otros/Andrey_tesis_Puriscal/DESCE/series$ ls
avgPhaseVelocity.h5    exclude_date.txt  geoNonZeroIntAmbiguity.h5   timeseries_ERAS5.h5           velocityERAS5.h5
avgSpatialCoh.h5       geo              inputs                  reference_date.txt   timeseries_ERAS5_ramp.h5   velocity.h5
coherenceSpatialAvg.txt  inputs          maskConnComp.h5    rms_timeseriesResidual_ramp.txt timeseries.h5
denErr.h5               maskTempCoh.h5   smallbaselineApp.cfg  temporalCoherence.h5  timeseriesResidual.h5
DESCE.txt              numInvIfgram.h5  temporalCoherence.h5 timeseriesResidual_ramp.h5
ERAS                  numInvIfgram.h5

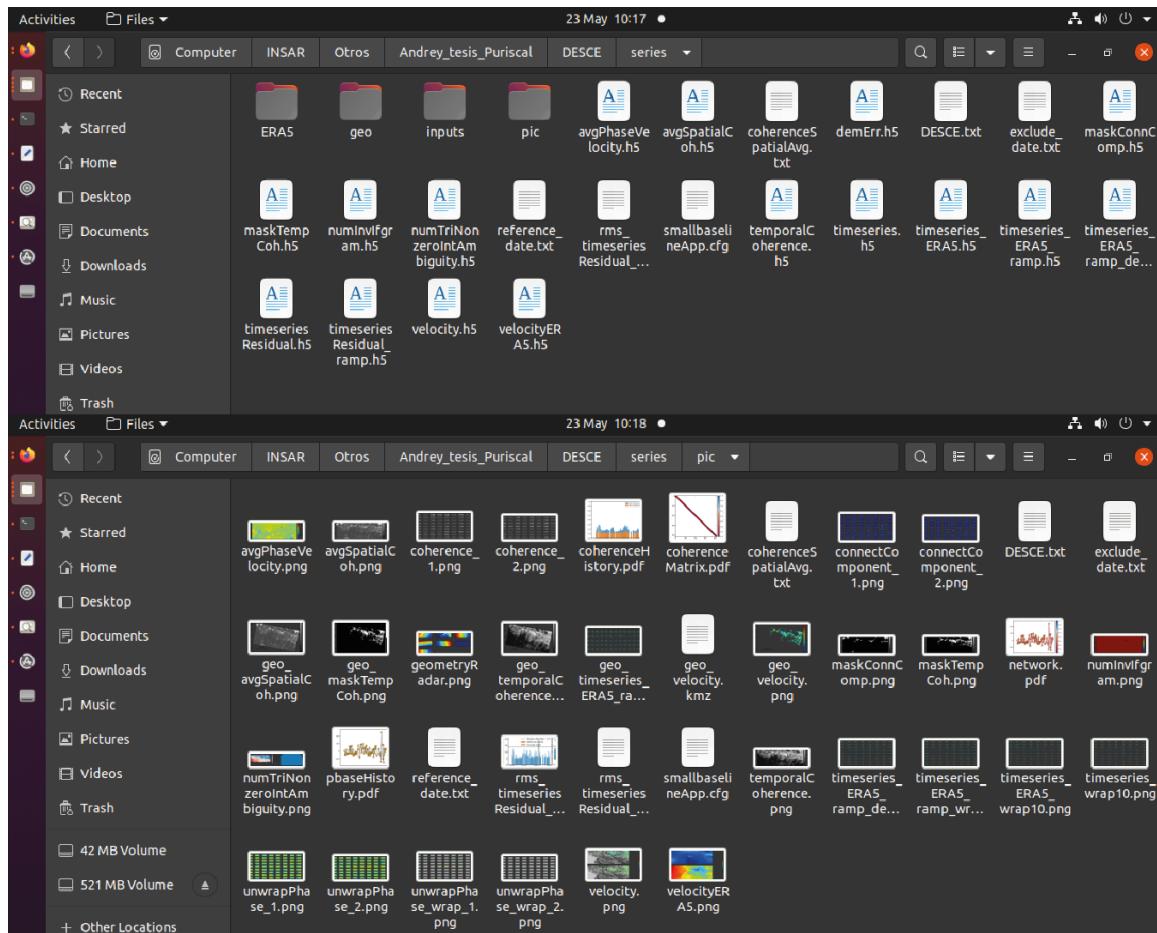

```

**Figura 43. Finalización de ejecución con MintPy**

Fuente: (Elaboración propia)

El directorio de “ERA5” es donde se guardan los archivos para la corrección atmosférica, en el directorio “geo” se guardan los archivos HDF georreferenciados como la velocidad de

desplazamiento. En la carpeta “inputs” se guarda el stack interferogramas en extensión HDF. Por último, la carpeta “pic” contiene los resultados convertidos de extensión HDF a imágenes png. La carpeta de ejecución con los resultados vista de manera gráfica se ve en la Figura 44, en la parte superior se observan los resultados en la carpeta de ejecución. En la parte inferior se ven los resultados en formato PNG dentro de la carpeta “pic”.



**Figura 44. Resultados MintPy en formato imagen dentro de la carpeta “pic”**

Fuente: (Elaboración propia)

## 2.8 Posibles errores con MintPy

### 2.8.1 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 45. Por lo tanto, hay que asegurarse de estar en la carpeta correcta antes de correr el comando.

```

load_data.py --template /home/ovsicori/TOSHIBA/InSAR_Poas/DESC/Serie_temp_desc/smallbaselineApp.cfg /root/tools/MintPy/mintpy/data/input_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/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=inpSteps)
  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(iargs)
  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 45. Error por directorio en MintPy**  
(Fuente: Elaboración propia, captura de pantalla)

## 2.8.2 Error por número de subbanda (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 46, 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/input_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=inpSteps)
  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(iargs)
  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 46. Error por número de subswath en MintPy**  
(Fuente: Elaboración propia, captura de pantalla)

## **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>