

4. Modelado e inversión geofísica con Python

4.1. Preparación del área de trabajo

Es recomendable que Python ya se encuentre instalado al momento de iniciar el curso, pero aquí se describe brevemente el proceso en caso de que no sea así:

1. **Descargar la versión correcta:** Ir a la página <https://www.python.org/>, y a la sección de descargas. Generalmente, la última versión de Python aparecerá como sugerencia principal, y la mayoría de las veces esa versión está bien. Sin embargo, se debe tomar en cuenta la compatibilidad con las librerías que se quieren utilizar, dado que algunas aún no tienen soporte para las últimas versiones de Python.
2. **Instalar el ejecutable:** se instala el ejecutable que se ha descargado y se marcan las casillas deseadas. *Importante:* Es recomendable marcar la opción *Agregar a PATH*, esto para evitar problema de permiso al acceder los archivos del sistema.
3. **Comprobar la instalación:** al abrir la ventana de símbolo del sistema (CMD), ya sea mediante el comando Win+R o buscándolo directamente en el menú de inicio, se debe teclear la palabra `python` para comprobar que el intérprete se abra correctamente, lo que corroborará que se ha instalado el programa.

Nota: esta sería una instalación normal de Python, pero para el curso se utilizará la instalación a través de Anaconda, para asegurar la compatibilidad. Si ya se tiene alguna versión de Python instalada, no existe ningún problema; si aún no se instala, en la sección siguiente se describe la instalación recomendada.

4.1.1. Instalación de librerías

Existen varias librerías en Python que permiten llevar a cabo el modelado, inversión, procesamiento, interpretación de anomalías, clasificación, y demás tareas geofísicas. Para lo que el presente curso concierne, nos centraremos en librerías como `PyGimLi` y `SimPEG`, así como las librerías de análisis numérico básico y generación de gráficos, como `Numpy` y `Matplotlib`.

La instalación de estas librerías es sencilla y no requiere mucho conocimiento técnico. En general, la forma más común de instalar alguna librería de Python es a través de PyPI (*Python package index*) con el comando `pip`.

Si bien este método funciona bien en la mayoría de las ocasiones, es buena práctica aislar los espacios de trabajo según el proyecto activo. Para esto, existen softwares de administración de paquetes de Python, tales como Anaconda. Anaconda es un programa que permite instalar librerías de Python corroborando que siempre exista compatibilidad entre ellas, por lo que no existen este tipo de problemas al instalar con Anaconda (o no debería, cosa que se verá más adelante).

La instalación de Anaconda es sencilla, pero existen distribuciones más ligeras para trabajos más sencillos, como es nuestro caso. Usaremos la distribución llamada Miniconda, la cual es una versión más ligera de Anaconda con menos paquetes, pero que contiene los necesarios para este curso.

Los pasos de instalación de Miniconda se describen a continuación:

1. **Descargar Miniconda:** entrar a la página <https://docs.conda.io/projects/miniconda/en/latest/> e ir a la sección de descargas en la parte de abajo. Descargar la versión de Windows 64-bit (o la versión que corresponda según su equipo)
2. **Instalar Miniconda:** abrir el programa ejecutable y seguir los pasos de instalación. Una vez terminado, al buscar en el menú de inicio 'anaconda', debe aparecer una ventana de comandos llamada *Anaconda prompt*. Si aparece, la instalación ha sido correcta.

Para instalar los paquetes a través de Anaconda, los pasos son sencillos:

1. **Abrir la ventana de comandos de Anaconda:** se busca en el menú de inicio *Anaconda prompt* y se ejecuta. Se debe abrir una ventana de comandos parecida a CMD.
2. **Creación de un nuevo espacio de trabajo:** observa que a la izquierda del comando inicial está la palabra **base**, de modo:

```
(base) C:/Users/user>
```

donde **base** indica el nombre del espacio de trabajo.

Se debe crear un nuevo espacio de trabajo con nombre **name** (cualquier nombre), a través del comando

```
>conda create -n name
```

Una vez creado, el espacio de trabajo se activa con:

```
>conda activate name
```

Observa que el nombre **base** al inicio del renglón cambia a **name**.

3. **Instalación de paquetes:** una vez dentro de **name**, comienza la instalación de paquetes. El paquete principal es **PyGimLi**, el cual se instala con el siguiente comando:

```
>conda install pygimli
```

Se debe esperar a que termine la descarga de paquetes, presionando 'y' cuando se solicite para comenzar la instalación. Este comando instalará todas las demás librerías necesarias, como **Numpy**, **Matplotlib**, **Pandas**, etc., además de **Python 3.9.18**, última versión de Python compatible con **PyGimLi**.

Se puede realizar la creación del espacio de trabajo y la instalación de paquetes con una sola línea en la ventana de comandos de Anaconda, como se recomienda en la página de instalación de **PyGimLi**:

```
>conda create -n pg -c gimli -c conda-forge pygimli=1.4.5
```

Sin embargo, es importante conocer el funcionamiento básico de Anaconda en caso de querer actualizar o instalar un nuevo paquete.

4.1.2. Software recomendado

Si bien se puede utilizar el IDE de su preferencia, se recomienda ampliamente utilizar Visual Studio Code, pues ofrece la mayor cantidad de funciones para múltiples lenguajes de programación, además de contar con una enorme biblioteca de extensiones de la comunidad para manejar cualquier tipo de archivos. Este será el IDE que el instructor empleará durante el curso.

Si se desea, se puede descargar a través de <https://code.visualstudio.com/> e instalar la extensión de Python desde su biblioteca.

4.2. Modelado e inversión de SEV

4.3. Modelado e inversión de TRE

4.4. Modelado e inversión de TRS

4.5. Modelado e inversión de gravimetría 2D

5. Adicional

5.1. Filtros de interpretación en planta

5.2. Interpretación de anomalías