

PRIMER INFORME DE ACTIVIDADES - DESCARGA DE SISMOS REGISTRADOS POR EL SERVICIO GEOLÓGICO COLOMBIANO (SGC).

Autor: Jheyston Omar Serrano Luna,

Número de contrato: 2021003344

Noviembre/11/2021 - Diciembre/17/2021

1. Acerca de los datos

En este primer informe se presenta el desarrollo de un algoritmo escrito en Python [3] para la descarga automática utilizando la base de datos IRIS y el servicio geológico colombiano (SGC). Este desarrollo utiliza la librería obsPy, la cual es un desarrollo libre que permite la lectura de los formatos más comunes utilizados en los datos sismológicos y realizar rutinas de procesamiento. Si el lector desea más detalles de la instalación de la librería por favor remitirse a su página web en el siguiente enlace <https://docs.obspy.org/> o su repositorio <https://github.com/obspy/obspy>. El uso de obsPy es bastante intuitivo por lo que se puede acceder rápidamente a la información, solo por mostrar un pequeño ejemplo he descargado el archivo **SGCWaveForm.mseed** y lo he ubicado en la carpeta */Files/* en el directorio de trabajo. Para leer dicho archivo solo basta con importar el método *read()* pasándole la dirección del archivo, tal como se presenta a continuación:

```
1 import matplotlib.pyplot as plt
2 from obspy import read
3
4
5 if __name__ == '__main__':
6     File = 'Files/SGCWaveForm.mseed'
7
8     st = read(File) /* En st se almacena el registro sismologico*/
9     tr = st[0]
10
11     print(tr.stats) /* Se muestran todos los metadatos */
12     print(tr.data) /* Se muestran los datos */
13
14     plt.plot(tr.data, "b-")
15     plt.suptitle(tr.stats.starttime)
16     plt.show()
```

La librería obsPy te permite también definir un URL para descargar el registro sismológico, pero se debe tener en cuenta que es solo valido para archivos desde 01-01-2016. Un ejemplo del URL se presentan a continuación:

`http://sismo.sgc.gov.co:8080/fdsnws/dataselect/1/query?starttime=2021-05-01T00:20:23.25&endtime=2021-05-01T00:25:23&network=CM&sta=GARC&cha=HH?&loc=00&format=miniseed&nodata=404`

Se ha resaltado en rojo los parámetros que debes suministrarle para poder realizar la consulta, estos parámetros son: **starttime**, **endtime**, **network**, **sta**, **cha**, **loc**, **format**. **starttime** es el tiempo de inicio del registro; **endtime** es el tiempo final del registro; **network** es el código de red definido por el centro de datos; **sta** es el código de la estación, se le pueden pasar varias estaciones separadas por comas y utilizar el signo de interrogación (?) para representar cualquier carácter individual o el asterisco (*) que representa cero o más caracteres; **cha** es el código del canal puede utilizar comodines como el signo de interrogación (?) o el asterisco; y **loc** es el código de localización puede utilizar comodines como el signo de interrogación (?) o el asterisco;. También, puedes hacer la descarga utilizando el siguiente enlace [SGC](http://sismo.sgc.gov.co:8080/fdsnws/dataselect/1/query?starttime=2021-05-01T00:20:23.25&endtime=2021-05-01T00:25:23&network=CM&sta=GARC&cha=HH?&loc=00&format=miniseed&nodata=404), donde debes especificar los parámetros de rangos de tiempos, canales, estaciones y códigos de localización. A continuación se presenta un script en Python donde se realiza una consulta utilizando el URL:

```
1 import matplotlib.pyplot as plt
2 import obspy
3 from obspy import read
4 import pandas as pd
5
6
7 plt.close('all')
8
9 if __name__ == '__main__':
10     st = obspy.read('http://sismo.sgc.gov.co:8080/fdsnws/dataselect/1/query?
11                     starttime=2021-05-01T00:20:23.25&endtime=2021-05-01T00:25:23&network=CM&sta=
12                     GARC&cha=HH?&loc=00&format=miniseed&nodata=404')
13
14     tr = st[2] /* Se selecciona la posición 2 para tomar la componente HHZ */
15
16     fig = plt.figure()
17     ax = fig.add_subplot(1, 1, 1)
18     ax.plot(tr.times("matplotlib"), tr.data, "b-")
19     ax.xaxis_date()
20     fig.autofmt_xdate()
21     ax.axvline(pd.to_datetime('2021-05-01T00:21:29'), color='r', linestyle='--', lw
22                 =2, label='Onda P') # Onda P
23     ax.axvline(pd.to_datetime('2021-05-01T00:21:32'), color='orange', lw=2, label='
24                 Onda S') # Onda S
25
26     plt.legend(bbox_to_anchor=(1.04, 0.5), loc="center left")
27     plt.xlabel('Tiempo registro')
28     plt.ylabel('Amplitud')
29     plt.show()
```

Una vez ejecutas el script de Python obtienes el siguiente registro sísmológico:

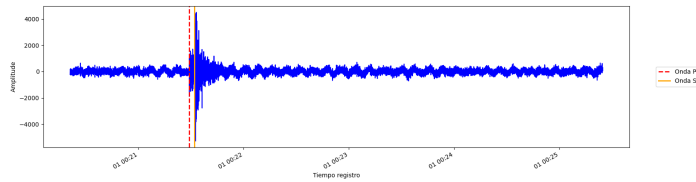


Figura 1: Registro sísmológico descargada usando el URL.

El problema de la descarga mediante el enlace SGC, es que solo se descarga un registro por consulta, por lo que en la zona interés en un año puedes obtener en promedio 7000 eventos sísmológicos, por lo que es una tarea larga y tediosa. Para automatizar el proceso, en este documento se proponen dos interfaces que realizan la búsqueda en la base de datos IRIS o mediante el URL en el SGC.

2. Pequetes necesarios

Los dos desarrollos realizados utilizan Python, junto con las librerías obsPy y Tkinter. La selección de Python es debido a que es lenguaje interpretado, sencillo, rápido y liviano. Para ejecutar los scripts y no alternar la configuración de nuestro equipo se recomienda instalar Anaconda. Anaconda te permite crear un ambiente virtual con una versión de Python que desees y en la cual se puede instalar distintos paquetes, sin crear incompatibilidad en nuestro sistema operativo. Aunque Anaconda es multiplataforma, lo que quiere decir que puede ser ejecutado sobre Windows, Linux y Mac, la aplicación fue probada en Ubuntu 18.04 y Centos 7, por lo que garantizo que la aplicación debe funcionar sobre versiones de linux. Para descargar Anaconda utiliza al siguiente enlace: <https://www.anaconda.com/products/individual> y seguir los pasos de acuerdo con el sistema operativo seleccionado. Una vez se tenga instalado anaconda se debe abrir un terminal y ejecutar el siguiente comando para configurar un ambiente virtual de python llamando obspy (puedes poner otro nombre si desees):

- `$ conda create -n obspy python=3.7`

Como se puede percibir de la anterior linea de comandos, toda la configuración se realiza sobre python 3.7. No recomiendo usar una versión de python actual (3.10) dado que la librería obsPy genera problemas de incompatibilidad. Una vez tengas instalado el ambiente virtual de python debes activarlo realizando el siguiente comando, notarás que el terminal cambiara de (base) a (obsPy):

- `$ conda activate obspy`

Hasta el momento no hemos instalado las librerías, solo hemos configura el ambiente virtual. El siguiente paso es realizar toda la configuración de librerías. Teclea en el terminal los siguientes comandos:

- \$ conda install -c conda-forge obspy
- \$ conda install cartopy
- \$ conda install pandas
- \$ conda install geopandas
- \$ conda install openpyxl
- \$ pip install tkcalendar
- \$ conda install -c conda-forge tqdm

Si todo ha sido exitoso ya tienes un ambiente virtual con todas las librerías necesarias para empezar con la descarga de los eventos sísmológicos tanto en la base de datos IRIS como en el SGC. Te recomiendo según el IDE que utilices seleccionar tu interprete, en mi caso estoy utilizando PyCharm y como puedes evidenciar en la Figura 2 he dejado seleccionado el interprete virtual creado en Anaconda llamado obspy.

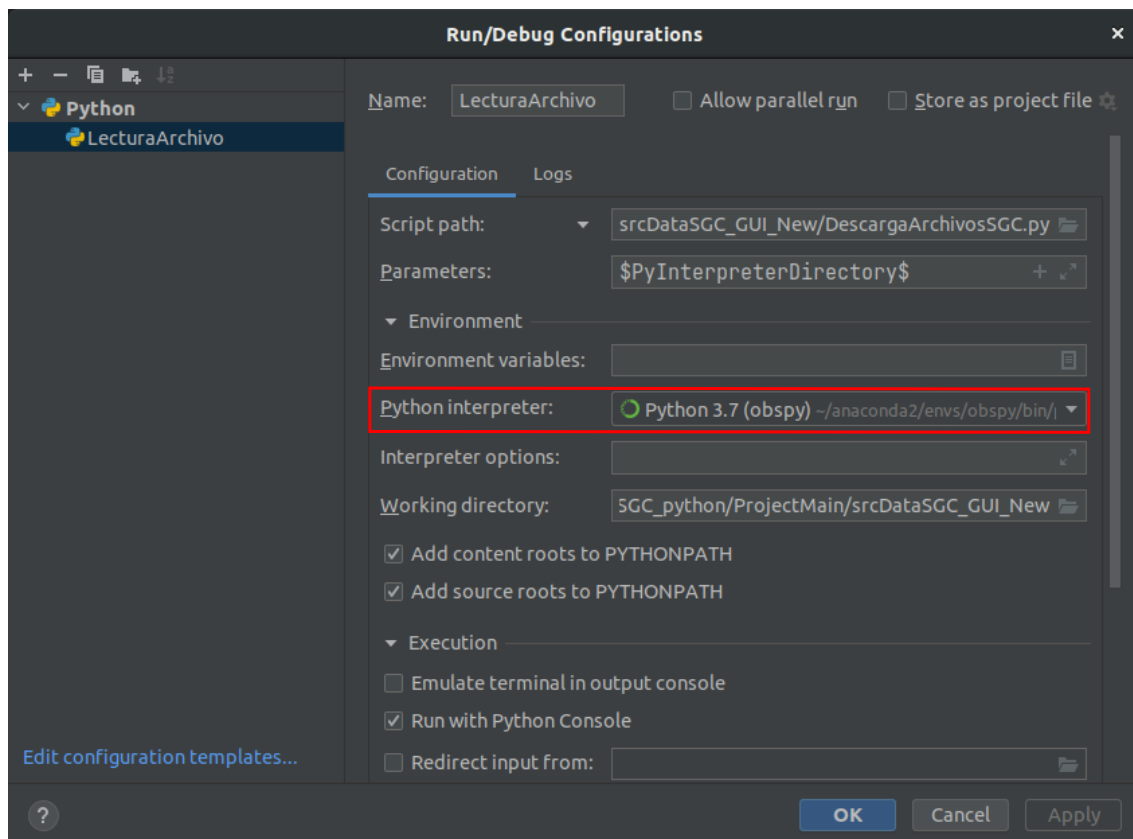


Figura 2: Selección del interprete de Python creado en Anaconda y llamado obsipy.

3. Descarga automatica de sismos

Aunque es posible realizar la descarga utilizando el URL del SGC, la descarga se realiza registro por registro, por lo que es una labor ardua y extensa. En esta sección se describe el uso y detalles de la implementación realizada para la descarga automática de registros sismológicos utilizando la base de datos IRIS y el SGC. El software muestra estadísticas de profundidad, latitud, longitud y magnitud junto con una grafica tridimensional de la ubicación de los sismos registrados.

3.1. Script utilizando la base datos IRIS

Esta interfaz de usuario es desarrollada en Python utilizando las librerías obsPy [1] [2] y Tkinter. Si no configuraste el entorno virtual de Python junto con sus librerías, te recomiendo revisar la sección 2 antes de continuar. Tkinter es un paquete de Python que maneja programación orientada a objetos y programación orientada a eventos, que te permite ubicar objetos en una ventana y proporciona interacción con el usuario. Algunos elementos utilizados en esta implementación son listas desplegables, etiquetas, campos de llenado, botones, calendarios, entre otros. En la Figura 11 se presenta la interfaz de usuario desarrollada para la descarga de eventos sismológicos utilizando la base datos IRIS. En esta interfaz, se habilitaron dos calendarios para la búsqueda de los eventos sismológicos: el calendario de la izquierda selecciona una fecha inicial de búsqueda y el calendario de la derecha selecciona una fecha final de búsqueda. Puedes también discriminar por la magnitud mínima y máxima de los eventos utilizando los campos **Min Magnitud** y **Max Magnitud**. En la búsqueda desplegable tienes dos alternativa de búsqueda: por latitud y longitud máxima y mínima (**Min-Max->LatLong**) o por radio con base a una latitud y longitud (**Radio**). El software está diseñado para de acuerdo con la selección de búsqueda habilitar o deshabilitar los campos que no son necesarios, tal como lo como puede percibir en la Figura 11. Al tener habilitado la búsqueda por **Min-Max->LatLong** deshabilita los campos **Max Radio**, **Latitud** y **Longitud**. El software también te permite discriminar por profundidad utilizando los campos **Min Deth** y **Max Depth**. Con ayuda de los campos **Tiempo Antes** y **Tiempos Despues** puedes definir en minutos cuanto tiempo guardar del registro de acuerdo con el tiempo del evento. En la lista desplegable **Canal** puedes seleccionar entre BH?, BHZ, HH? o HHZ. El símbolo ? es un comodín que representa múltiples letras, por lo que al seleccionarlo descargamos las tres componentes (NEZ). El botón **Consultar**, te permiten empezar la búsqueda una vez todos tus parámetros han sido debidamente diligenciados, de lo contrario te mostrar algún mensaje de error de acuerdo con el parámetro que falte. La lista desplegable **Resgistros** empezara vacía, cuando se realice una búsqueda te permite seleccionar un registro y presionando el botón **Graficar** se mostrara en pantalla el registro obtenido, como se presenta en la Figura 4. Al presionar el botón **Graficar** por primera vez, se te presentara un grafico 3D de los eventos obtenidos como se presenta en la Figura 5, el radio de estos eventos son proporcionales a la magnitud del evento y el color depende la profundidad. En la zona seleccionada de esta consulta se encuentra el nido sísmico de Colombia, por lo que notamos una concentración de eventos cerca a la profundidad de 150 (Km), Latitud 6.7 y Longitud -73. La Figura 6 presenta

las estadísticas de Latitud, Longitud, Profundidad y Magnitud de los eventos encontrados.

Figura 3: Interfaz para la descarga automática de registros usando la base de datos IRIS. Enlace: [PythonUISProjectIRIS](#)

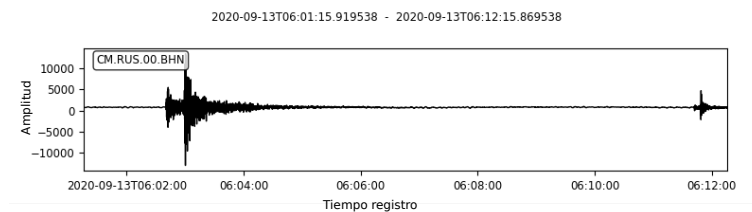


Figura 4: Registro obtenido de la consulta en la base de datos IRIS utilizando el software.

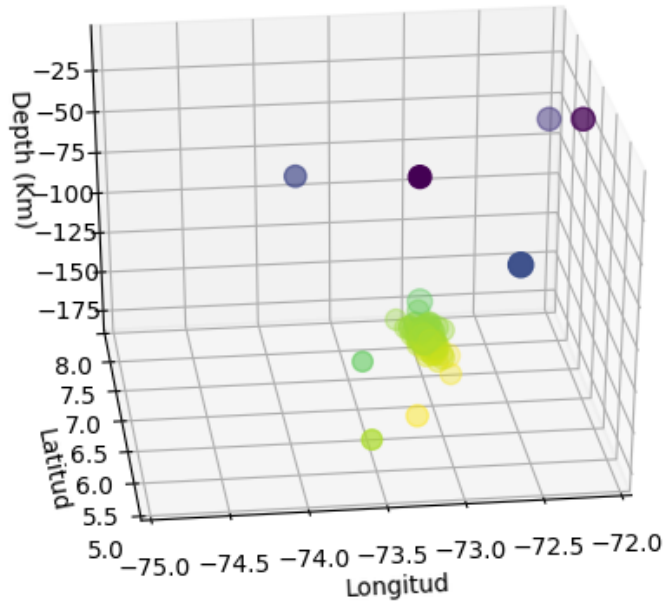


Figura 5: Grafico 3D de los eventos obtenidos del software realizando la consulta en la base de datos IRIS.

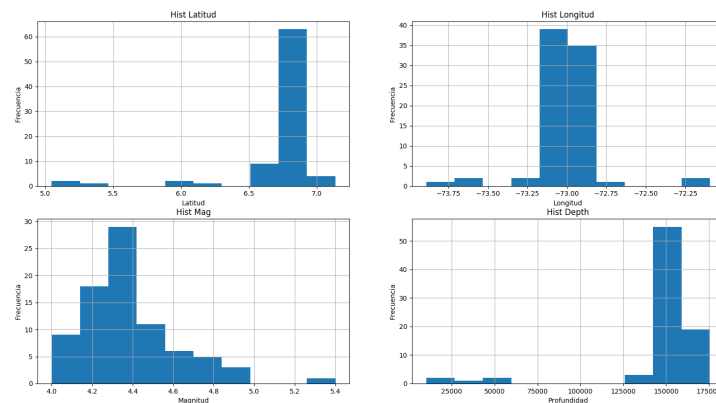


Figura 6: Estadística de la consulta en la base de datos IRIS utilizando el software.

Cuando finaliza una consulta, el software te guarda un archivo llamado `descargaArchivosIRIS.pkl` que se asocia con un DataFrame de todos los eventos. Un DataFrame es una estructura de datos en Python que permite almacenar distintos tipos de datos en columnas, puedes almacenar datos como enteros, caracteres, números reales, listas. En el DataFrame se guardan objetos de tipo Stream de obsPy que te permiten guardar las trazas. La ventaja de guardar el tipo de dato Stream es que te queda la información de la traza junto con los meta-datos que contienen información del evento. La Figura 7 presenta el esquema de salida del DataFrame. En la carpeta compartida encuentras un script en python llamado `lecturaDataFrame.py` donde presento un pequeño ejemplo sobre una

consulta realizada.

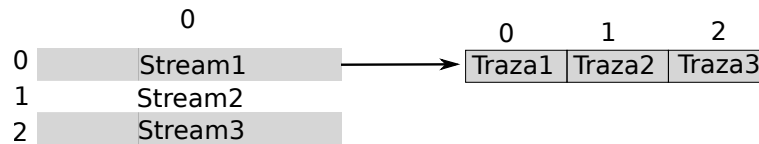


Figura 7: DataFrame de salida obtenido de la consulta en IRIS utilizando el software.

3.2. Script utilizando la base datos del SGC

Al igual que el anterior script que descarga los registros de la base de datos IRIS, este script también fue desarrollado en Python junto con las librerías obsPy y Tkinter. Si no configuraste el entorno virtual de Python junto con sus librerías, te recomiendo revisar la sección 2 antes de continuar. En la Figura 11 se presenta la interfaz de usuario desarrollada para la descarga de eventos utilizando la base datos del SGC. En esta interfaz debes realizar una búsqueda sobre la pagina del SGC para descargar un Excel con los eventos sísmológicos. Para esto debes utilizar el siguiente enlace <http://bdrsnc.sgc.gov.co/paginas1/catalogo/index.php> y seleccionar sobre cual catalogo deseas realizar la búsqueda. El SGC tiene dos catálogos: Catálogo sísmico: 1 Junio de 1993 - 28 Febrero de 2018 y Catálogo sísmico: 1 Marzo de 2018 a la fecha. Como ejemplo, realizare la búsqueda sobre el Catálogo sísmico: 1 Marzo de 2018 a la fecha y he configurado los parámetros de acuerdo con la Figura 8. Dar clic en consultar y te presentara una tabla con todos los eventos encontrados. En esta consulta puedes dar clic sobre Ver Mapa Sismicidad y te entregara un mapa con todos los eventos encontrados como se presenta en la Figura 9. Para ingresar estos eventos al software debes descargarlos dando clic sobre la pestaña Excel como se presenta en la Figura 10. Como último paso debes cambiar los cabeceros del Excel: Fecha-Hora (UTC) por FECHA, Lat(°) por LATITUD, Long(°) por LONGITUD, Prof(Km) por PROFUNDIDAD y Mag. por MAGNITUD. Una vez realizados estos cambios en el archivo Excel ya tienes configurados todos los eventos a tener en cuenta en la descarga del software. Recomiendo ubicar este archivo Excel en el proyecto, he dejado una carpeta llamada */Sismicidad/* donde se encuentra la sismicidad de la zona de interés desde el año 2010 hasta el año 2021. En la ruta */sismicidad/2021/mesSept2021.xlsx* encuentras un archivo Excel ejemplo con el mes de septiembre para probar el software.

[Consultar](#)

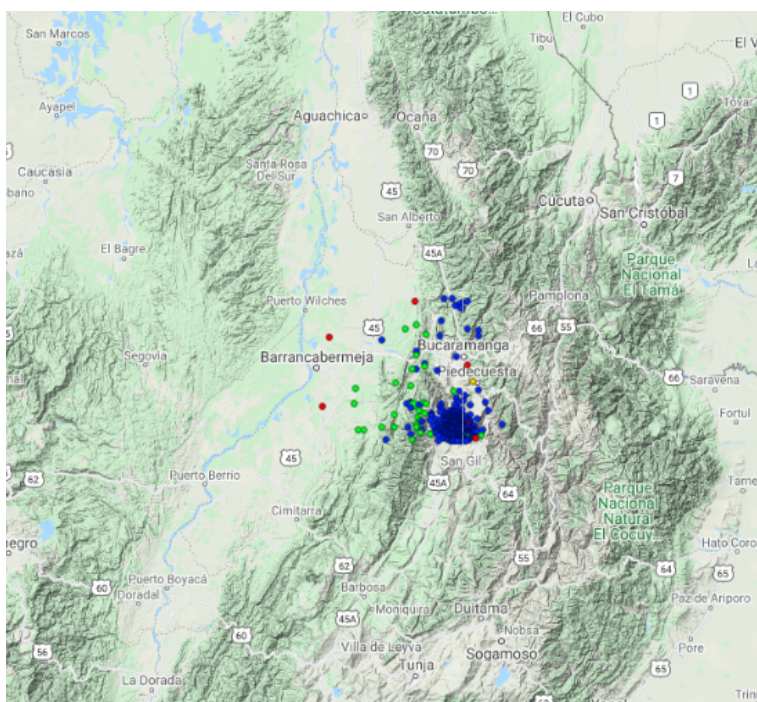


Figura 9: Mapa de sismicidad obtenido del SGC para el mes de septiembre del 2021 .

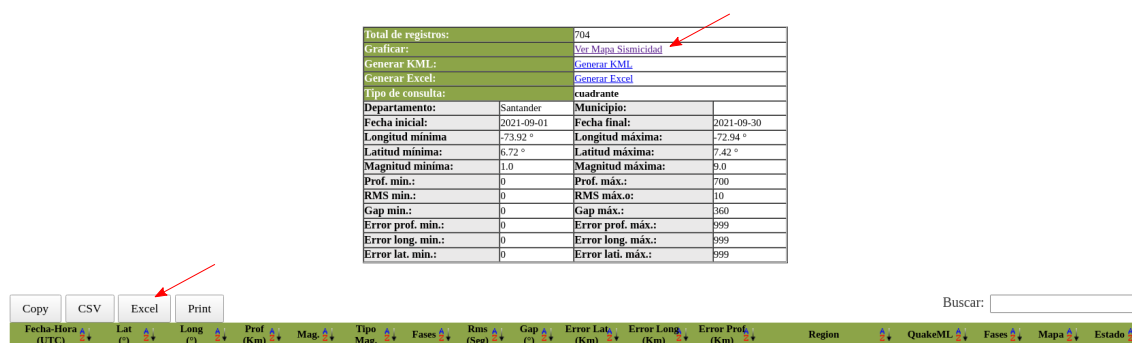


Figura 10: Obtener Mapa de sismicidad del SGC para el mes de septiembre del 2021 y realizar la descarga sobre la pestaña Excel.

En el software debes cargar el archivo Excel de los eventos, para esto solo basta con dar clic sobre el botón archivo y te mostrara una ventana sobre la cual puedes desplazarte para buscar el archivo, una vez seleccionado el archivo solo basta con dar clic en Abrir y el software te mostrar la ruta del archivo seleccionado. **Min Magnitud** y **Max Magnitud** te permiten filtrar los eventos de acuerdo con el rango de magnitud seleccionado. **Min Depth** y **Max Depth** te permiten filtrar los eventos de acuerdo con el rango de profundidad seleccionado. **MinLatitud**, **MaxLatitud**, **MinLongitud**, **MaxLongitud** te permiten buscar que estaciones del SGC quedan en la zona. De igual forma el parámetro **Canal** te permite definir que tipo de canal deseas encontrar en la

búsqueda, algunas estaciones no registran todos los canales o no se encuentra información por lo que el software enviara un mensaje indicando que no encontró información. Puedes seleccionar entre BH?, BHZ, HH? o HHZ. El símbolo ? es un comodín que representa múltiples letras, por lo que al seleccionarlo descargamos las tres componentes (NEZ). Con los campos **Tiempo Antes** y **Tiempos Despues** puedes definir en minutos cuanto tiempo guardar del registro de acuerdo con el tiempo del evento. De lo evidenciado en las búsquedas no todas las estaciones del SGC contienen la información de los eventos encontrados. La opción **Busqueda por** estará siempre predefinida en **(Min-Max)->LatLong**. El botón **Consultar**, te permiten empezar la búsqueda una vez todos tus parámetros han sido debidamente diligenciados, de lo contrario te mostrar algún mensaje de error de acuerdo con el parámetro que falte. La lista desplegable **Registros** empezara vacía, cuando se realice una búsqueda te permite seleccionar un registro y presionando el botón **Graficar** se mostrara en pantalla el registro obtenido, como se presenta en la Figura 12. Al presionar el botón **Graficar** por primera vez, se te presentara un grafico 3D de los eventos obtenidos como se presenta en la Figura 13, el radio de estos eventos son proporcionales a la magnitud del evento y el color depende la profundidad. En la zona seleccionada de esta consulta se encuentra el nido sísmico de Colombia, por lo que notamos una concentración de eventos cerca a la profundidad de 150 (Km), Latitud 6.7 y Longitud -73. La Figura 14 presenta las estadísticas de Latitud, Longitud, Profundidad y Magnitud de los eventos encontrados.

Figura 11: Interfaz para la descarga automática de registros usando la base de datos del SGC.
Enlace: [PythonUISProjectSGC](#)

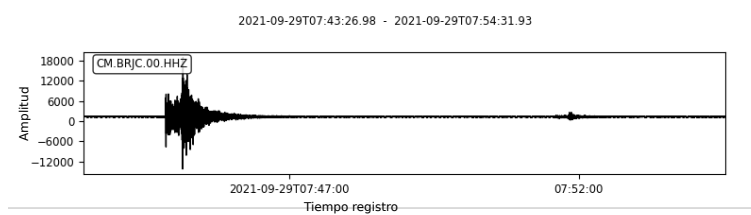


Figura 12: Registro obtenido de la consulta al SGC utilizando el software.

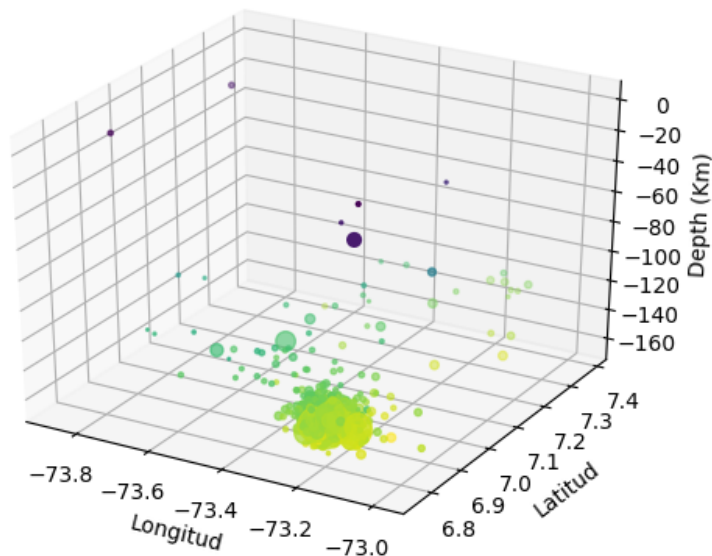


Figura 13: Grafico 3D de los eventos obtenidos del software realizando la consulta al SGC.

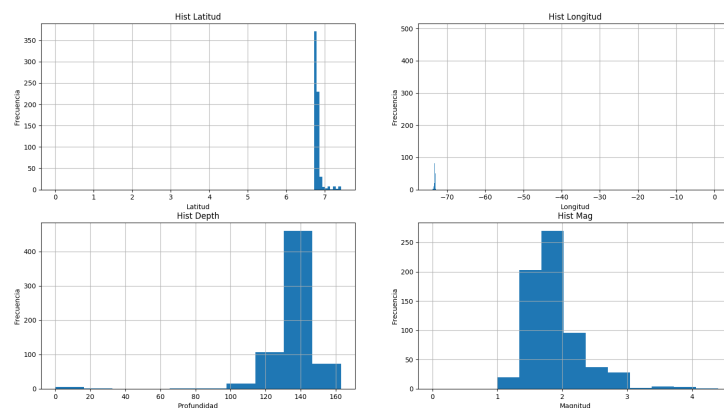


Figura 14: Estadística de la consulta en el SGC utilizando el software.

Cuando finaliza una consulta, el software te guarda un archivo llamado `descargaArchivosSGC.pkl` que se asocia con un DataFrame de todos los eventos. La Figura 15 presenta el esquema de salida del

DataFrame. En la carpeta compartida encuentras un script en python llamado lecturaDataFrame.py donde presento un pequeño ejemplo sobre una consulta realizada.

	Network	Station	LatStation	LongStation	Location	Channel	Starttime	Endtime	Sampli...	Delta	Npts	Eventtime	DepthEvent	LongEvent	LatEvent	MagEvent	Data
0	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T00:21:13.9...	2021-09-01T00:32:18.27...	100.0...	0.01000	66538	2021-09-01T00:22:15.00000002	135.86000	-73.21400	6.77100	1.70000	[523 551 521 ... 597 643 703]
1	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T00:21:14.9...	2021-09-01T00:32:15.95...	100.0...	0.01000	66100	2021-09-01T00:22:15.00000002	135.86000	-73.21400	6.77100	1.70000	[720 685 716 ... 522 359 409]
2	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T00:21:14.6...	2021-09-01T00:32:15.34...	100.0...	0.01000	66066	2021-09-01T00:22:15.00000002	135.86000	-73.21400	6.77100	1.70000	[4072 4095 4124 ... 3990 403]
3	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T00:23:11.7...	2021-09-01T00:34:17.19...	100.0...	0.01000	66542	2021-09-01T00:24:14.00000002	153.18000	-73.16900	6.78500	1.50000	[377 394 396 ... 662 695 609]
4	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T00:23:11.3...	2021-09-01T00:34:14.23...	100.0...	0.01000	66292	2021-09-01T00:24:14.00000002	153.18000	-73.16900	6.78500	1.50000	[785 725 707 ... 765 838 767]
5	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T00:23:13.6...	2021-09-01T00:34:17.56...	100.0...	0.01000	66392	2021-09-01T00:24:14.00000002	153.18000	-73.16900	6.78500	1.50000	[4110 4093 4106 ... 3993 400]
6	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T01:44:08.3...	2021-09-01T01:55:14.35...	100.0...	0.01000	66668	2021-09-01T01:45:11.00000002	141.74000	-73.15800	6.81800	1.50000	[318 333 339 ... 252 309 316]
7	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T01:44:10.4...	2021-09-01T01:55:12.21...	100.0...	0.01000	66176	2021-09-01T01:45:11.00000002	141.74000	-73.15800	6.81800	1.50000	[719 640 636 ... 638 709 702]
8	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T01:44:08.5...	2021-09-01T01:55:11.88...	100.0...	0.01000	66332	2021-09-01T01:45:11.00000002	141.74000	-73.15800	6.81800	1.50000	[3464 3437 3439 ... 3294 330]
9	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T02:54:50.0...	2021-09-01T03:05:56.07...	100.0...	0.01000	66608	2021-09-01T02:55:53.00000002	143.00000	-73.14100	6.81200	1.50000	[535 531 530 ... 374 353 370]
10	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T02:54:50.6...	2021-09-01T03:05:57.01...	100.0...	0.01000	66614	2021-09-01T02:55:53.00000002	143.00000	-73.14100	6.81200	1.50000	[310 414 379 ... 172 261 274]
11	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T02:54:49.0...	2021-09-01T03:05:55.48...	100.0...	0.01000	66640	2021-09-01T02:55:53.00000002	143.00000	-73.14100	6.81200	1.50000	[3087 3073 3065 ... 3098 309]
12	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T04:03:51.1...	2021-09-01T04:14:58.53...	100.0...	0.01000	66740	2021-09-01T04:04:55.00000002	127.89000	-73.16300	6.74500	1.50000	[315 339 334 ... 147 147 1]
13	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T04:03:53.6...	2021-09-01T04:14:55.85...	100.0...	0.01000	66202	2021-09-01T04:04:55.00000002	127.89000	-73.16300	6.74500	1.50000	[441 388 339 ... 328 328 256]
14	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T04:03:52.7...	2021-09-01T04:14:56.02...	100.0...	0.01000	66332	2021-09-01T04:04:55.00000002	127.89000	-73.16300	6.74500	1.50000	[2542 2531 2544 ... 2501 251]
15	CM	BRJC	6.97000	-73.74000	00	HHE	2021-09-01T04:08:47.7...	2021-09-01T04:19:55.09...	100.0...	0.01000	66736	2021-09-01T04:09:51.00000002	151.00000	-73.09900	6.80400	1.70000	[435 429 434 ... 104 112 86]
16	CM	BRJC	6.97000	-73.74000	00	HHN	2021-09-01T04:08:47.9...	2021-09-01T04:19:51.29...	100.0...	0.01000	66332	2021-09-01T04:09:51.00000002	151.00000	-73.09900	6.80400	1.70000	[346 366 372 ... 258 223 295]
17	CM	BRJC	6.97000	-73.74000	00	HHZ	2021-09-01T04:08:49.3...	2021-09-01T04:19:52.66...	100.0...	0.01000	66332	2021-09-01T04:09:51.00000002	151.00000	-73.09900	6.80400	1.70000	[2504 2490 2485 ... 2430 239]

Figura 15: DataFrame de salida obtenido de la consulta en el SGC utilizando el software.

Referencias

- [1] BEYREUTHER, M., BARSCH, R., KRISCHER, L., MEGIES, T., BEHR, Y., AND WASSERMANN, J. Obspy: A python toolbox for seismology. *Seismological Research Letters* 81, 3 (2010), 530–533.
- [2] KRISCHER, L., MEGIES, T., BARSCH, R., BEYREUTHER, M., LECOCQ, T., CAUDRON, C., AND WASSERMANN, J. Obspy: A bridge for seismology into the scientific python ecosystem. *Computational Science & Discovery* 8, 1 (2015), 014003.
- [3] MILLMAN, K. J., AND AIVAZIS, M. Python for scientists and engineers. *Computing in Science & Engineering* 13, 2 (2011), 9–12.