GRANADA UNIVERSITY (UGR)

Geophysics Andalusian Institute

THEORETICAL PHYSICS AND THE COSMOS DEPARTMENT

ANDALUSIAN INSTITUTE OF GEOPHYSICS AND PREVENTION OF SEISMIC DISASTERS

"System to select, merge and save events since seismic records"

(Sistema para Seleccionar, Unir y Guardar Eventos desde Registros Sísmicos)

USER MANUAL: VERSION 1.0

Author:

Ligdamis A. Gutiérrez E. PhD.

Masaya Volcano, Nicaragua
Photo by: Ligdamis A. Gutiérrez E.

Granada, Spain 2021- 2023

General Index

1.- Introduction.

Back to Index

The "*System to Select, Merge and Save Events since Seismic records (Sistema para Seleccionar, Unir y Guardar Eventos desde Registros Sísmicos)*" It features a user-friendly interface that enables easy and efficient management for merging or concatenating separate traces from a seismic record, resulting in a complete, isolated event that facilitates database construction and subsequent analysis. The final result of the event concatenation can be stored in two basic formats: MSEED and SAC. This allows for later use in calculations, analyses, and various scientific investigations. The system allows for easy and rapid concatenation, making it much easier and more reliable to determine the result through a resulting graph. These processes provide a reliable automated tool that assists the operator in the subsequent process of building event databases and performing spectral analyses on later seismic signals, automatically adding value to the operator's knowledge for later manipulation of the resulting events.

The application, through its built-in libraries, allows the loading and reading of various seismic formats, including SAC, MSEED, GSE2, EVT, and WAV, among others. The first version of this system consists of a single interface that includes tools for fusing or concatenating seismic signals or records and for combining a specific event from two original seismic traces stored separately by the sensors, whether measured in minutes, hours, or days. The main interface is available in English. However, documentation, such as this document, is available in both Spanish and English. Information on the folder structure and contents can be found in the appendices. The system also allows the storage of the resulting event in two of the most widely used formats in seismic institutes and observatories: MSEED and SAC. In addition, if desired, after the events have been captured, the graphic results can be saved in various formats, such as: PNG, JPG, EPS, PS, PDF, RAF, TIF, among others.

The module and the entire system have been developed in Python, version 3.8.6. (The set of libraries is compatible with version 3.10.10). Additionally, a series of open-access libraries are included, which, in conjunction with Python, enable the use of graphical and analytical tools, providing ease of use and enhancing computational power for the user. Some of the main elements and libraries used are listed below:

- **Matplotlib**: Used for creating static, animated, and interactive visualizations in Python. (https://matplotlib.org/stable/users/index.html).
- **NumPy**: A library for numerical operations in Python. (https://numpy.org/doc/stable/user/quickstart.html).
- **PyQt5**: A tool that links with the graphical library Qt5 in C++ (https://pypi.org/project/PyQt5/).
- **Obspy**: A Python toolbox for seismology. (https://docs.obspy.org/).
- **Tkinter**: Graphical User Interface (GUI) (https://docs.python.org/3/library/tkinter.html)

Another key feature of the system is its definition as a cross-platform application, meaning it can operate on various platforms or operating systems, such as Windows (7, 8, 10, 11) in both 32-bit and 64-bit versions. It also supports Linux systems, such as Ubuntu and other similar systems (Debian, Red Hat, Fedora, SUSE, etc.), macOS, and Android for tablets and mobile devices (with Python appropriately adapted for these devices).

NOTE: In the appendices of this document (*as well as in the Readme.txt and Initial_requirements.txt files*), you can find general information on installation for Windows and Linux systems, as well as guidelines for installing the main programs and additional libraries required by Python to properly execute the developed programs in its environment.

2.- Initial Screen of the System.

Back to Index

In the appendices of this document and in the "**README.txt**" file included in the "***Documents***" folder, you will find instructions for installing the system on Windows (*the process on Linux systems is similar*). Essentially, you need to perform two actions:

a) Copy the "Set_tools_System_1_1" folder to "My Documents" on Windows.
b) Copy the "Set_tools_System_1_1.bat" file to the Windows desktop.

Additionally, there are instructions for installing the necessary Python libraries on the system. Once "Set_tools_System_1_1.bat" has been copied to the desktop, you need to right-click on it and select "Run as administrator."

Fig. 1 Popup Window when Right-Clicking on the "Set_tools_System_1_1.bat" File

In the window that opens, click the "Yes" button when prompted with "*Do you want to allow this app to make changes to your computer*?" This is a warning message. However, the application does not make any changes, so you should trust its execution.

Upon clicking "*Yes*," the following command window opens, welcoming you to the system.

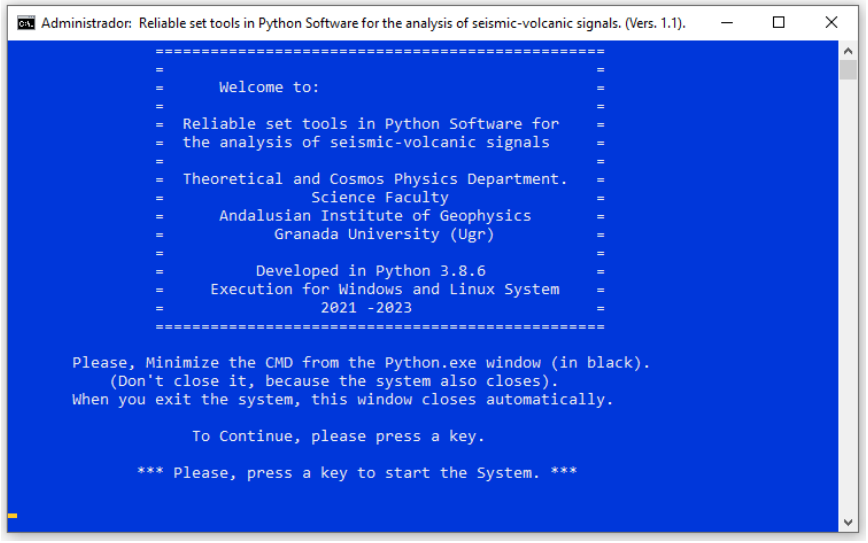


Fig. 2 Welcome Screen and Instructions for Loading the System.

After reading the information in the window, you simply need to press any key to access the system's initial screen. The folder should already be copied to "***My Documents***," and the "**Set_tools_System_1_1.bat**" file contains all the loading instructions.

The system's initial screen is "**Menu.py**". It appears when any key is pressed on the Welcome screen. Additionally, the Python command window or console is displayed, similar to the following:

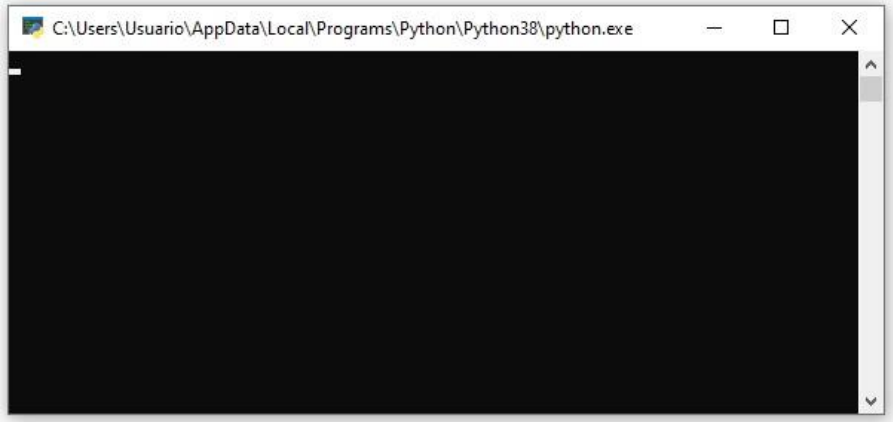


Fig. 3 Python Console (CMD) Window (*Should be minimized*)

To avoid obstructing the view, you can and should "minimize" this screen. Do **not** close it, as this would also close the system's startup window. Once you have finished working with the system, this window will close automatically. The initial presentation screen of the system (the module menu) "**Menu.py**" is as follows:

NOTE: When pressing or clicking on a module button, the startup window will close and the module window will open (*this may take a little time depending on the PC's memory. It is recommended to have at least 8 GB of memory in the system, with 16 GB being ideal*)

Fig. 4 **Main Menu Screen.** The module to be worked on is highlighted. Module 7 (*Select, merge and save events*).

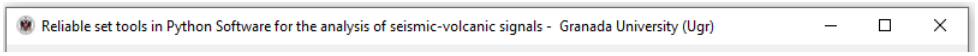
.

2.1.- Elements of the Initial Screen.

As observed in the previous figure, the initial or presentation screen is a simple window composed of:

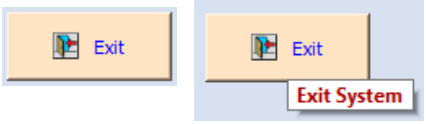
a) A top toolbar with basic information about the module.
b) On the left side, there are 10 execution buttons or command buttons for each module of the system.
c) At the bottom, there is a command button that allows for exiting the system.
d) Additionally, it features a background image representing a volcano (Masaya in Nicaragua), and three images with the logos of the University of Granada, the Andalusian Institute of Geophysics, and the Department of Theoretical and Cosmic Physics.

a) At the top, the icon of the University is visible, along with the module title and a reference to the University of Granada (UGR).



b) On the left side, there are 10 execution buttons or command buttons for each module of the system. When the mouse pointer is placed over each button, it is highlighted in white to indicate that it is being selected. Clicking on a button closes the startup menu window and opens the window for the indicated module (this may take a little time depending on the PC's memory).

c) At the bottom, there is a command button: **Exit**. When the mouse pointer is placed over each button, a text appears indicating the action of that button (Exit System, Start System).



If you click the "Exit" button, a window will appear asking the user if they are sure they want to leave the system.

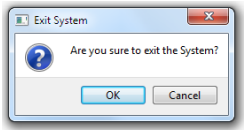


Fig. 5 Text Box Indicating Whether You Want to Exit the System.

If you click "**OK**," the screen will close and the system will exit. If you click "Cancel," you will remain on the initial screen

3.- Main Interface for Concatenation or Merging.

The "Main Concatenation Screen" is the module's primary interface, where the activities that comprise the record reading and merging tools for two seismic records into one are performed. This process occurs because a given event may be segmented at the end of one record and continue into the next. To address this, both records must be merged, concatenated, or joined into a single record. The resulting record then contains the segmented event, allowing for subsequent actions, such as using slicer software, to further segment the event. This screen consists of the following parts:

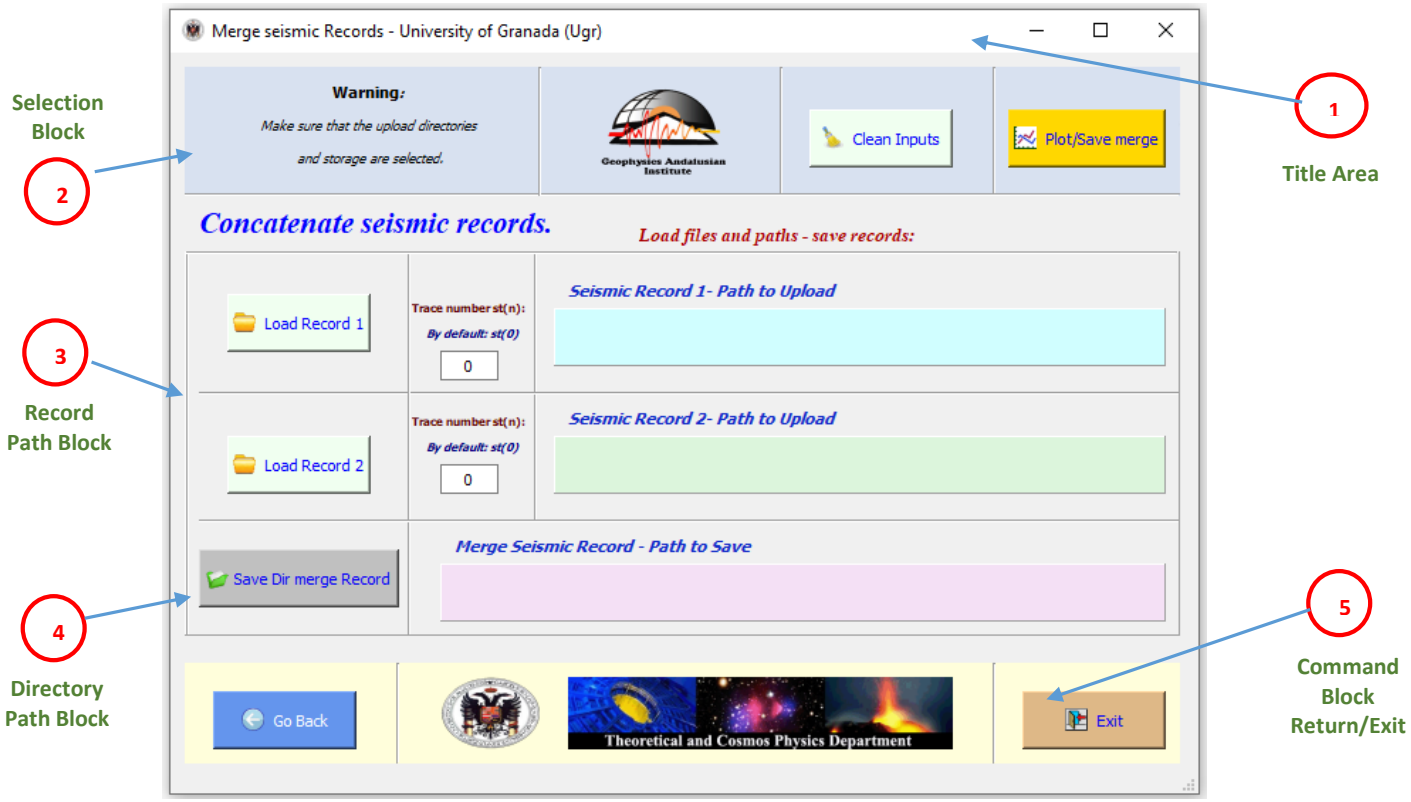


Fig. 6 Elements of the concatenation Interface.

1) Title Area.
2) Selection Block Command buttons: Clean Inputs, b) Plot/Save Merge, c) Alert Message: Ensure that the file and directory loading paths are selected.
3) Log Path Block: a) Buttons for the physical location of logs 1 and 2 to be analyzed, b) Areas for the physical location of logs 1 and 2 to be analyzed, c) Number of traces for logs 1 and 2.
4) Directory Path Block: a) Directory selection command button, b) Area for specifying the directory where the resulting log will be stored.
5) Command Block: a) Command buttons (Go Back, Exit)

3.1.- Elements of the analysis screen.

The previous screen consists of various elements for its use. At the top, you will see the program name, icon, university name, and author as the title. (1).

The elements that make up the main screen are detailed below.

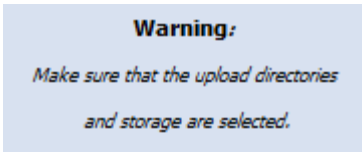
Added to number (1), the 4 elements of the initial fusion interface screen have been distributed, highlighted in three main blocks that are listed from (2-5) in the red circles.

3.1.1.- Selection Block. (2)

Fig. 7 Block of selection elements. In the green circles: A) Command button: Clear inputs, B) Plot, merge and save the signal resulting from the merging of the two assigned registers (Plot/Save Merge).

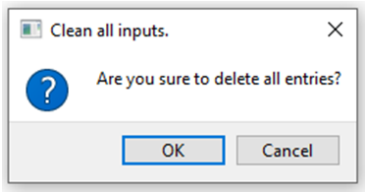
This block is configured by the green circles in the previous figure. Firstly

a) **Warning message**: Ensure that the file upload and directory options are selected.



b) **Input cleaning command button** (Clean Inputs).

Clicking this button clears all entries on the screen. It will clean the data entry text boxes and the path or folder where the record to be downloaded will be stored, and it will delete all active entries with data at that moment. It restores the initial values of the main interface (*see Fig. 6*).



When you place the mouse pointer over it, a message appears indicating its function. Clicking the button opens a window asking the user if they are sure they want to delete the data entries. If confirmed, it clears all entries and returns the interface to its initial state. Otherwise, the current entries remain in the interface.

c) **Graph button, Merge and Save signal record** (Plot/Save merge).

Once the data inputs (Files 1 and 2) and the directory where the result of combining both records will be stored are selected, click this button. This will display the resulting graph and prompt the user to choose whether to save the combination in ***MSEED*** or ***SAC*** format. Hovering the mouse pointer over these two action buttons will display a message explaining their actions.

3.1.2.- Record Path Block.

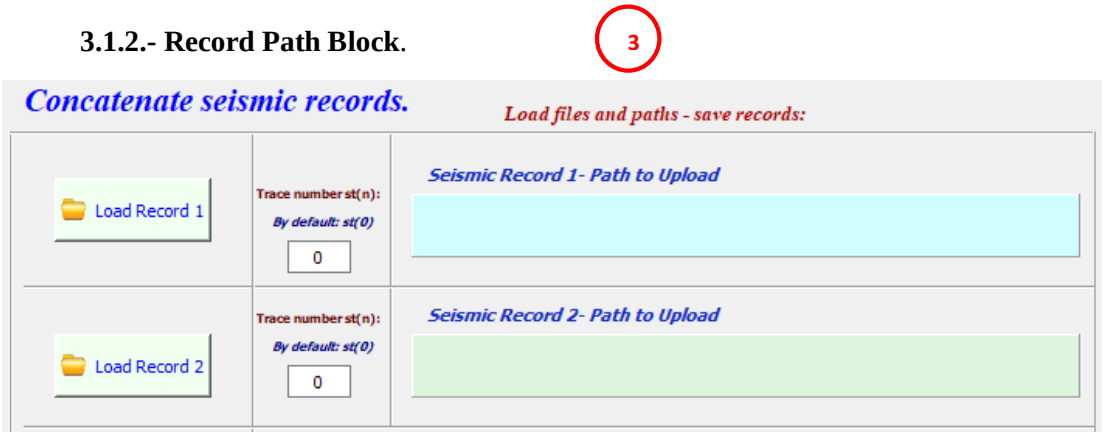


Fig. 8 The Route block consists of: File Load buttons 1 and 2, and the Save Dir directory path. The areas where the path to follow for each action will be displayed are shown, as well as the traces to select (trace 0 is selected by default). Records can have more than one trace. If so, equivalent traces must be selected in both records.

The image shows the buttons: "***Load Record 1***" and "***Load Record 2***". Hovering the mouse pointer over them displays a message indicating their function. It performs the search and loading of seismic records using the "***Obspy***" library in various formats (SAC, MSEED, SEISAN, etc.). To the right of each button, you can see: a) the traces of each record (trace 0 by default) and b) the areas where the paths for these actions will be displayed.

3.1.3.- Record Load Buttons 1 and 2.

Clicking the "***Load Record 1***" and "***Load Record 2***" buttons opens a file explorer window (by default, the path is in the "**C**" drive of the PC), displaying options for the various file formats to search and allowing you to perform a search within the computer's directory. This is shown in the following screenshot.

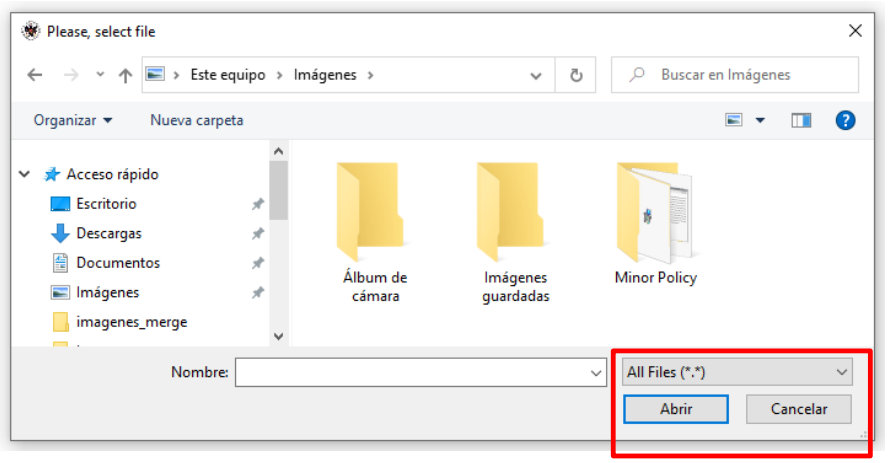


Fig. 9 Record Selection Screen.

On this screen (the language is determined by the operating system), you select the logs according to the desired format (*red box*) (SAC, MSEED, GSE2, EVT, etc.). This is possible through the seismic format reading library "***Obspy***".

Once selected, click the "***Open***" button to load the record onto the analysis screen. Otherwise, click the "***Cancel***" button to return to the analysis screen. The record selection process is shown in the following screenshot.

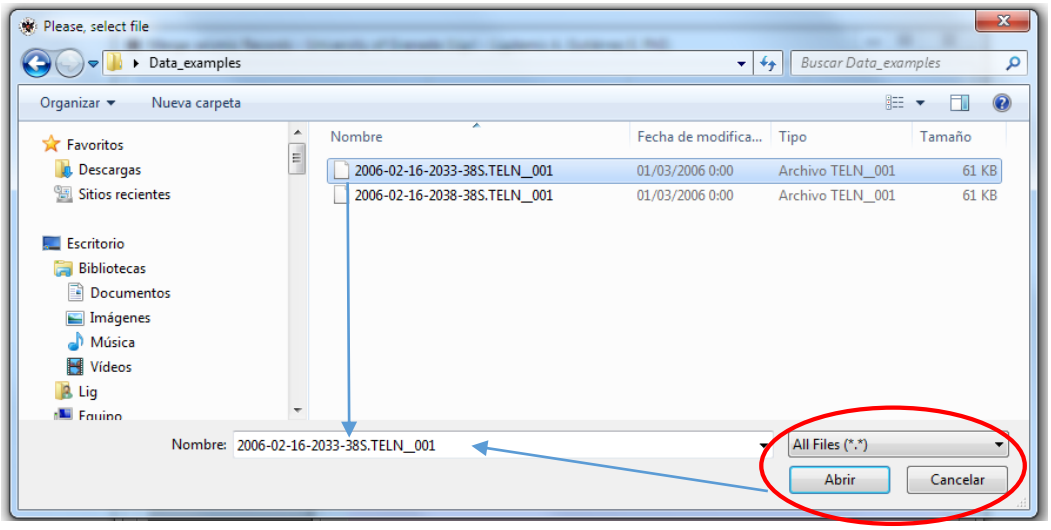


Fig. 10 Pantalla ejemplo de Selección de un registro de formato "SEISAN" del volcán Telica[1] de Nicaragua.

On the screen, in the lower right corner, indicated by the red circle and displayed by the arrow, is the list of the most common seismic formats supported and/or used in observatories and institutes worldwide (*SAC*, *MSEED*, *GSE2*, *WAV*, *EVT*, *etc.*).

When you select a specific type, the records are displayed according to that format. For example, the "SEISAN" files are stored in "Data_examples". Clicking on the desired record, as shown, places it in the "Name" box. At this point, click the "Open" button from the previous screen, which loads the physical location of the record on the system. This path will be displayed in the "Seismic Record 1 - path to Upload" box, located to the right of the "Load" button. The same process applies to the second record to be uploaded (Record 2).

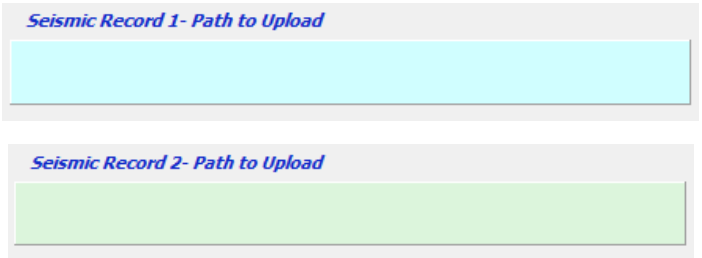
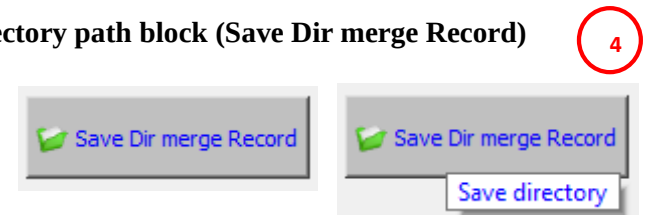


Fig. 11 File path chart 1 and 2, showing the location of the record.

[1] Data taken from the doctoral thesis: Gutiérrez Espinoza, L.A. Detection and classification system of seismic-volcanic signals using hidden Markov models (HMMS): application to active volcanoes of Nicaragua and Italy. Granada: University of Granada, 2014. 387 p. [http://hdl.handle.net/10481/30895].

This is an important aspect, as it determines whether each record can later be located on the computer for analysis. If the file is invalid, cannot be found, or the parameters are incorrect, a validation window will appear indicating this (See Fig. 16, Page 14).

3.1.4.-Directory path block (Save Dir merge Record)



The "***Save Dir merge Record***" button, when hovered over, displays a message explaining its function. Clicking it opens a file explorer window (by default, the path is in the PC's root directory "**C**"), allowing the user to select a folder or directory where the resulting merge record will be stored. This is shown in the following screenshot.

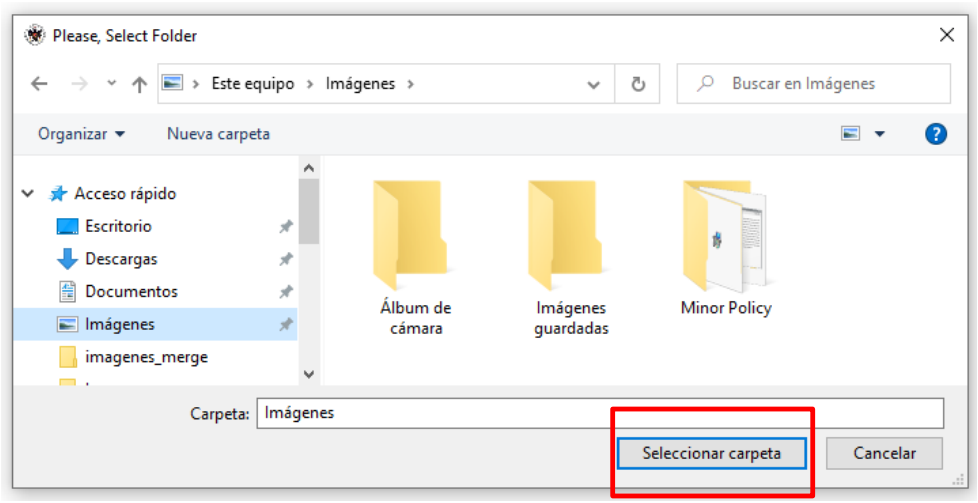


Fig. 12 Selection of the folder where the event resulting from the merging or concatenation of the two selected events will be stored.

On this screen (the language is determined by the operating system), click the "***Select Folder***" button to choose the folder or directory where you want the event to be cut from the seismic record to be saved (*red box*). Otherwise, click the "***Cancel***" button and the action will return to the main cut screen.

This route will be displayed in the box "*Merge Seismic Record - Path to Save*" located to the right of the "***Save Dir merge Record***" button.

Fig. 13 Folder path box showing the location of the record to be saved.

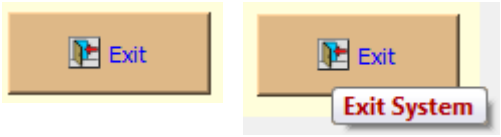
3.1.5.- Command Block. 5

Fig. 14 Command block, consisting of: "Go Back" and "Exit" buttons.

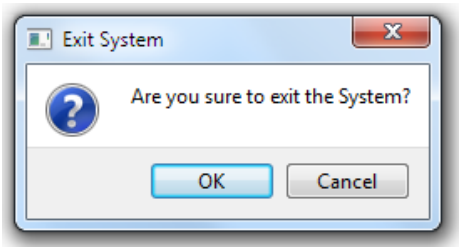
The image shows the "***Go Back***" button, which searches for and loads seismic records in various formats, and the "***Exit***" button, which exits the system.

1) ***Go Back button***: Allows you to return to the system's initial presentation screen (Menu). When you hover the mouse pointer over it, a message is displayed indicating its function.

2) ***Exit button***: allows complete system exit (after displaying a screen asking if you wish to exit). Hovering the mouse pointer over it displays a message indicating its function.



Similarly to the home screen, if you press or click the "***Exit***" button, a window appears asking the user if they are sure they want to leave the system.



Clicking "***OK***" closes the screen and completes the system exit. "***Cancel***" remains on the analysis screen.

4.- Examples of file path and directory upload.

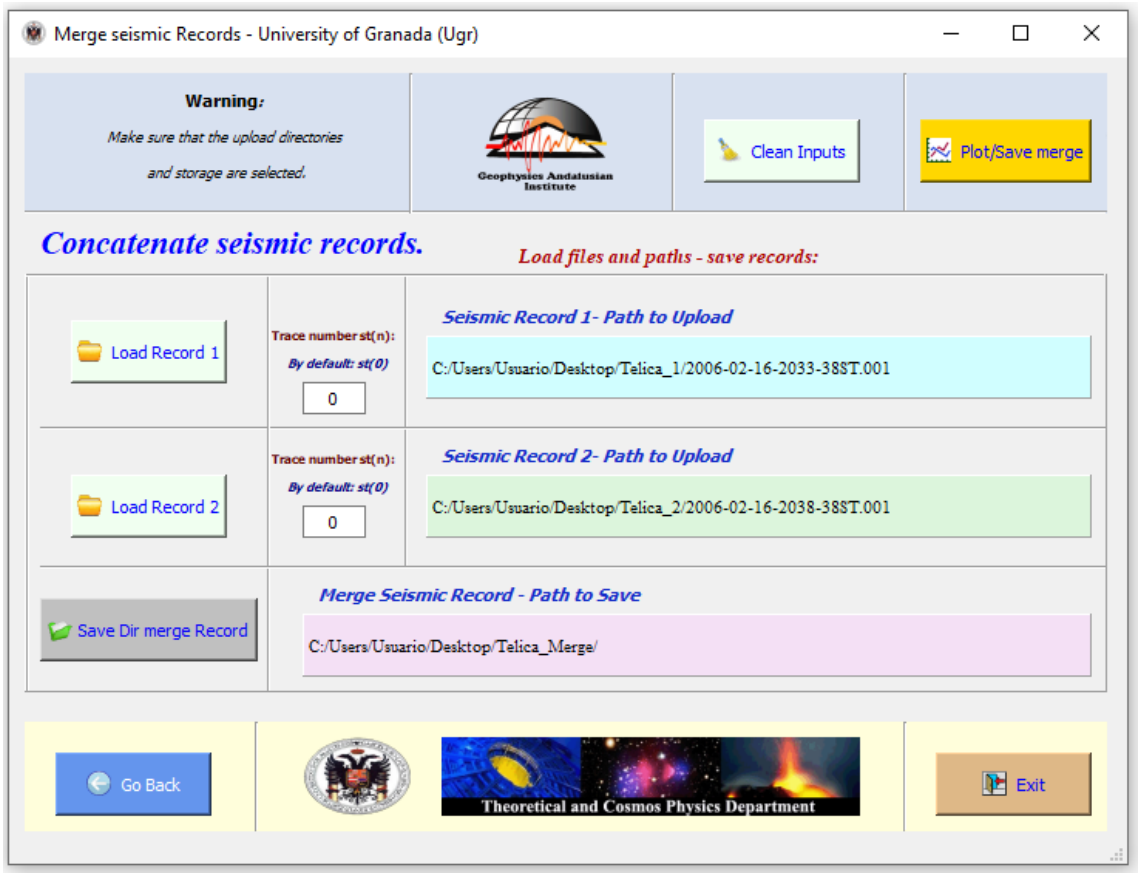


Fig. 15 Example of loading records and selecting the directory where the merged or concatenated record will be stored.

The previous interface image shows the two records selected for merging and the directory where the merge will be stored. Once these paths have been specified, click the "Plot/Save merge" button to plot and save the resulting merged record. If everything goes well, the merged record will be displayed, and the user will be asked in what format they wish to save it. Otherwise, a series of validations will appear that must be resolved. These include the following.

5.- Validation of errors in registration or entries.

Clicking the "***Plot/Save merge***" button will trigger a validation process if an error occurs (invalid entry, non-existent record, or out-of-range format). This validation will be displayed through several dialog boxes, each showing an alert message. This allows the user to modify the entries or select a valid record without the system crashing or stopping. The displayed screens are as follows:

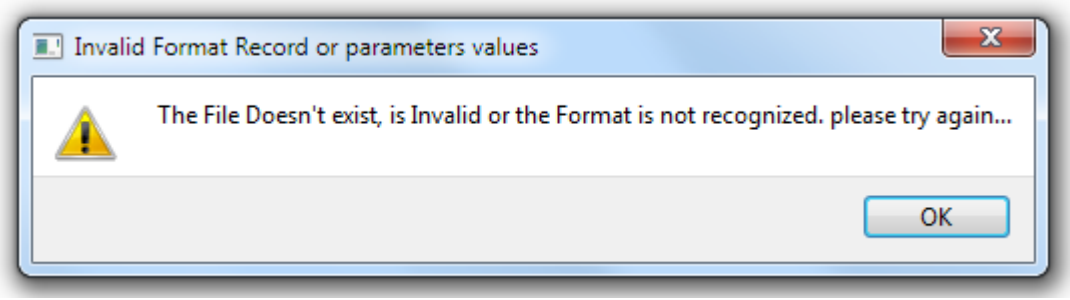


Fig. 16 Validation of invalid entries, incorrect formats or records.

According to the message in the previous figure, an error has occurred because the format is not recognized, the record does not exist, or the merge cannot be performed because the records do not match. Additionally, the parameters or inputs may be outside the allowed range for the signal being analyzed. Pressing the "***OK***" button returns you to the system, allowing you to select a valid file or correct the erroneous inputs. This way, the program continues running without further issues. The input validations are as follows:

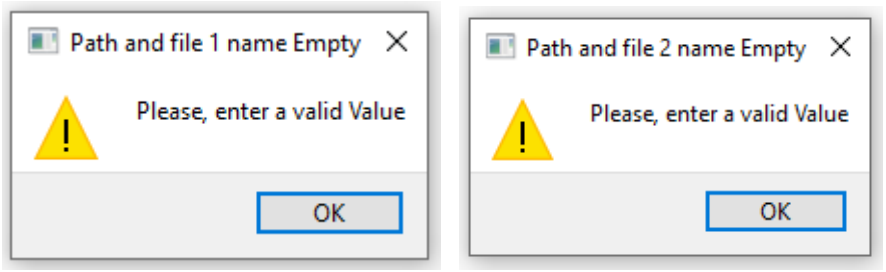


Fig. 17 Validation of empty entries in registers 1 and 2. A valid record must be entered in each of them.

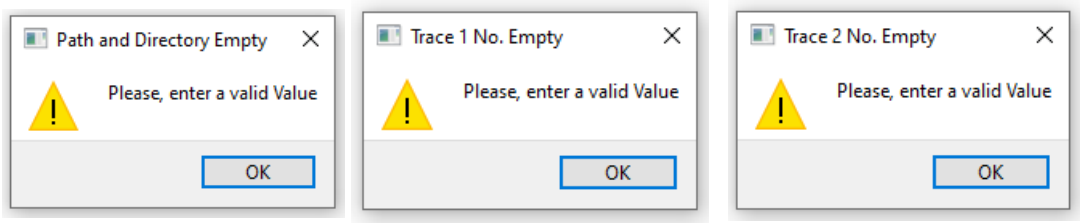


Fig. 18 Validation of empty input in the directory path where the log is stored and validation of empty traces.

6.- Results of File Sections and Merging.

Next, we will present examples of the final results of the record selection and merging process.

6.1.- Example of record selection and merging results.

According to the entire process described above, the process for selecting and merging two seismic records is very simple, and consists of the following steps:

a) Click the "**Load Record 1**" button to select record number 1. (***The file path is displayed as: Seismic Record 1 - Path to Upload***). By default, the initial path is located in the root directory "**C**" of the PC, whether on a Windows or Linux system.
b) Click the "**Load Record 2**" button to select record number 1. (***The file path for the record is displayed: Seismic Record 2 - Path to Upload***). By default, the initial path is located in the root directory "**C**" of the PC, whether on a Windows or Linux system.
c) Open or select the folder or directory where the captured signal will be saved. (***The folder path is displayed: Merge Seismic Record - Path to Save***).
d) Click the "**Plot/Save merge**" button to graph the resulting record and store it in a chosen format.

The original graphs of both records (with the partial event shown in the red box) can be seen in the following figure.

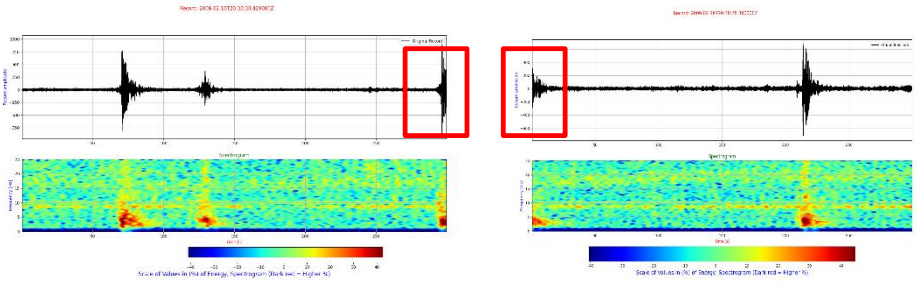


Fig. 19 Individual records in which, in the red boxes, an event is observed at the end of the first record and continues at the beginning of the second.

Once both records are concatenated, the resulting graph of these parameters is shown in the following figure. The output of this analysis will consist of a graph in three parts: a) the signal from the first register, b) the signal from the second register, and c) the signal combined from both registers.

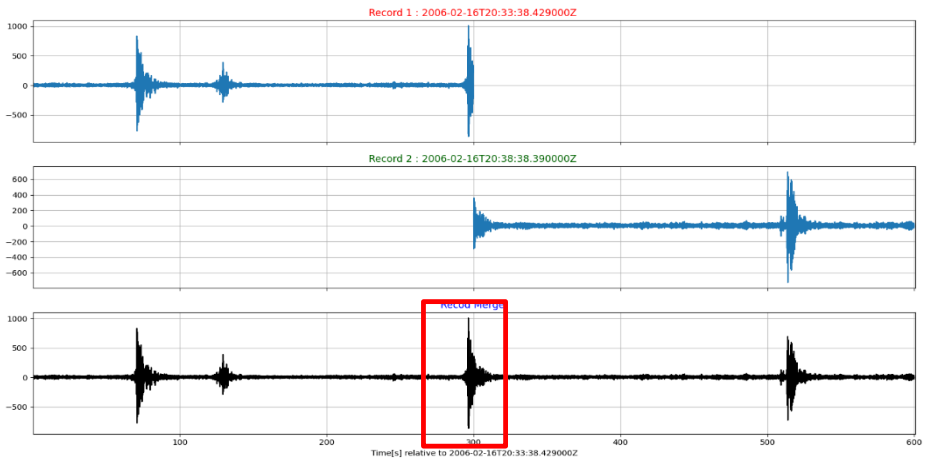


Fig. 20 Request to store the resulting record in "MSEED" or "SAC" format. By default, the "MSEED" format is selected. Clicking "CANCEL" will close the screen for modifying the selection.

When the previous graph is displayed, the result to save is the last graph, the complete one. To save the complete record, a pop-up window appears, indicating or asking in what format you want to save the resulting record. That screen is shown below.

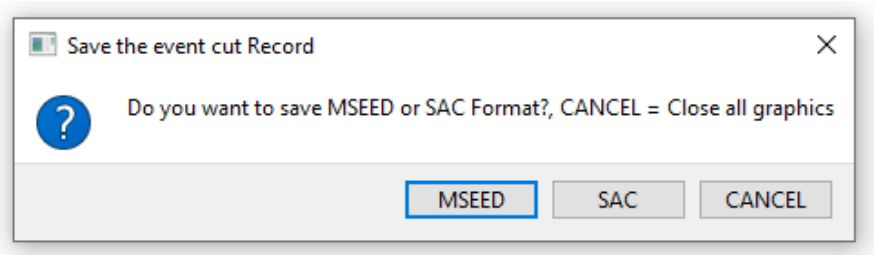


Fig. 21 Request to store the resulting record in “MSEED” or “SAC” format. By default, “MSEED” is selected; clicking “CANCEL” will close the screen for modifying the selection.

When selecting each of the formats, a screen appears indicating that the record has been successfully stored in the selected "Path".

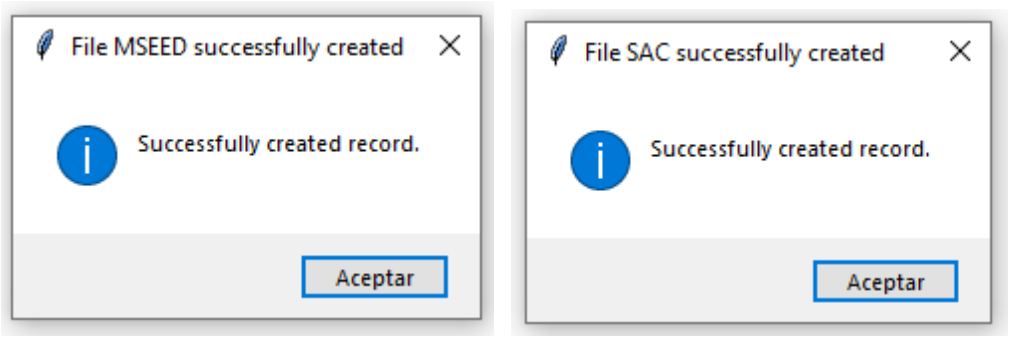


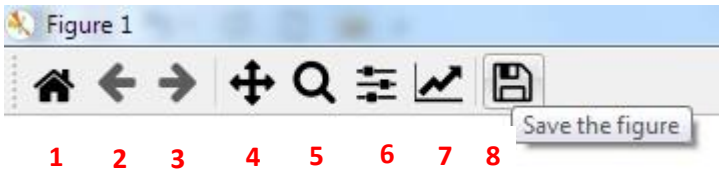
Fig. 22 Confirmation that the record (MSEED/SAC) has been stored correctly.

After saving the resulting record and before closing the window (by clicking the "OK" button), you can save the resulting graph, including a zoom (using the Zoom [Magnifying Glass] tool) made to the resulting graph (*See Matplotlib Tools, Pages 17-22*).

IMPORTANT NOTE: The stored records are presented according to the filtered signal. The stored event is the merging of the original records, exactly as they appear in the format read. Therefore, the fused event will be similar (unfiltered) when saved. The user can then apply filters using the analysis program (***Module 1***) or use the event cleavage program (***Module 6***) to perform any subsequent analysis.

7.- Toolbar of Graphs (Matplotlib Library).

In the construction of graphs, the *Matplotlib* library's graph screen has a set of very useful tools that allow you to visualize, edit, and save graphs in various formats. At the top of the Matplotlib graph screen that appears when a graph is created, there is a toolbar similar to the following:



From left to right, the icons representing the actions to be performed are:

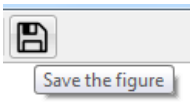
1. Reset original view: Restores all graphs to the initial state.
2. Back to previous view: Preview of the selected graph.
3. Forward to next view: Forward view of the next image.
4. Left button pans, Right button zooms, x/y fixes axis, CTRL fixes aspect. Moves the graph and axes left or right, zooms in or out, fixes the x/y axis, and fixes the aspect with CTRL.
5. Zoom to rectangle: Through a rectangle, zooms in on the selected graph.
6. Configure subplots: Configuration of subplots (Borders and spacings).
7. Edit axis, curve and image parameters: Editing the parameters of the graph. Select the axes or graph and edit elements such as title, coordinates (X, Y), and curve parameters (lines, markers) in styles, colors, and size.
8. Save the figure: Saves the graph in several formats.

This document does not delve into each of them; it only highlights the use of those that are generally more commonly used, such as (1, 2, 5, 7, and 8).

In the above graphs, the use of the zoom tool (5) has been observed. Tools 2 and 3 allow zooming in or out individually for each graph. Option 1 allows restoring all elements or subplots of the graph to their initial values (*each individual graph or part of the window*). As for option 8, it allows saving the graph in various formats. The rest are straightforward, and it is up to the user to explore each of them. Now, the processes for "***editing***" and "***saving*** or ***storing***" the graphs (*Numbers 7 and 8*) are detailed below.

8.1.- *Saving Graphs.*

The process of saving graphs is very simple. Click on the icon of tool number 8 (Save the figure).



This opens an explorer window, similar to those in Windows (depending on the language or system used), where you can select the folder or directory where the graph will be saved.

Additionally, provide a name and select the desired format type. This can be done at the bottom of the explorer window (red circle in the image), where various format types available for saving are selected. The screen resembles the following.

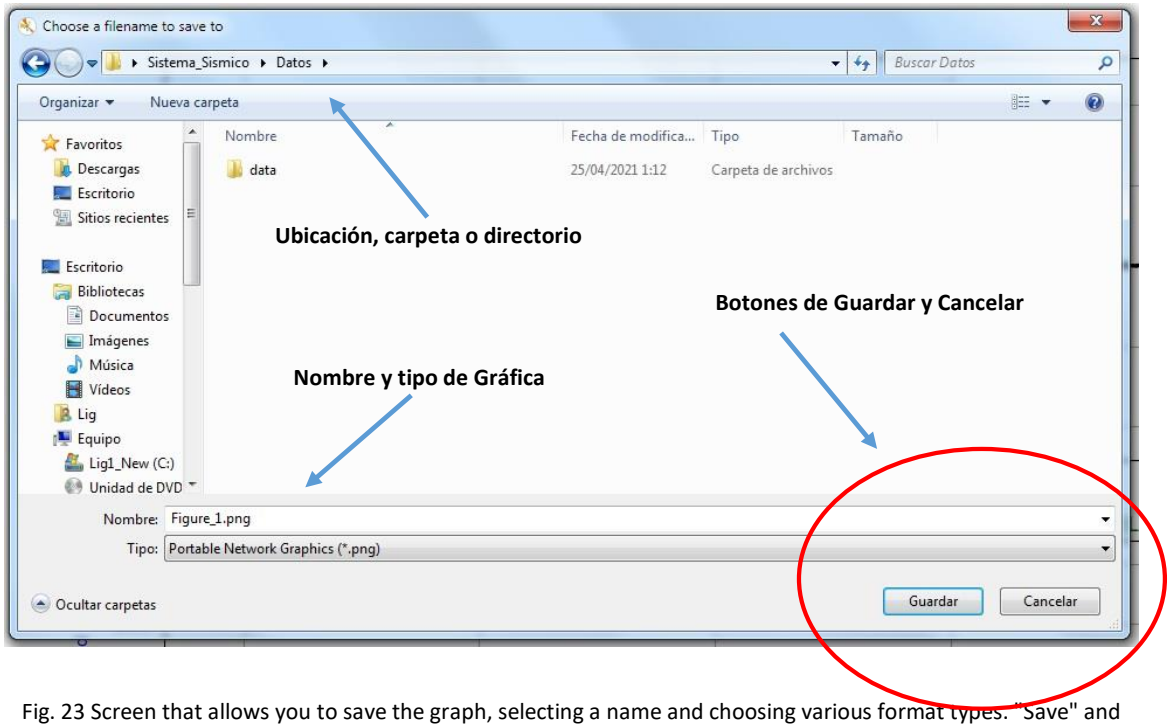


Fig. 23 Screen that allows you to save the graph, selecting a name and choosing various format types. "Save" and "Cancel" buttons are provided to complete or cancel the process.

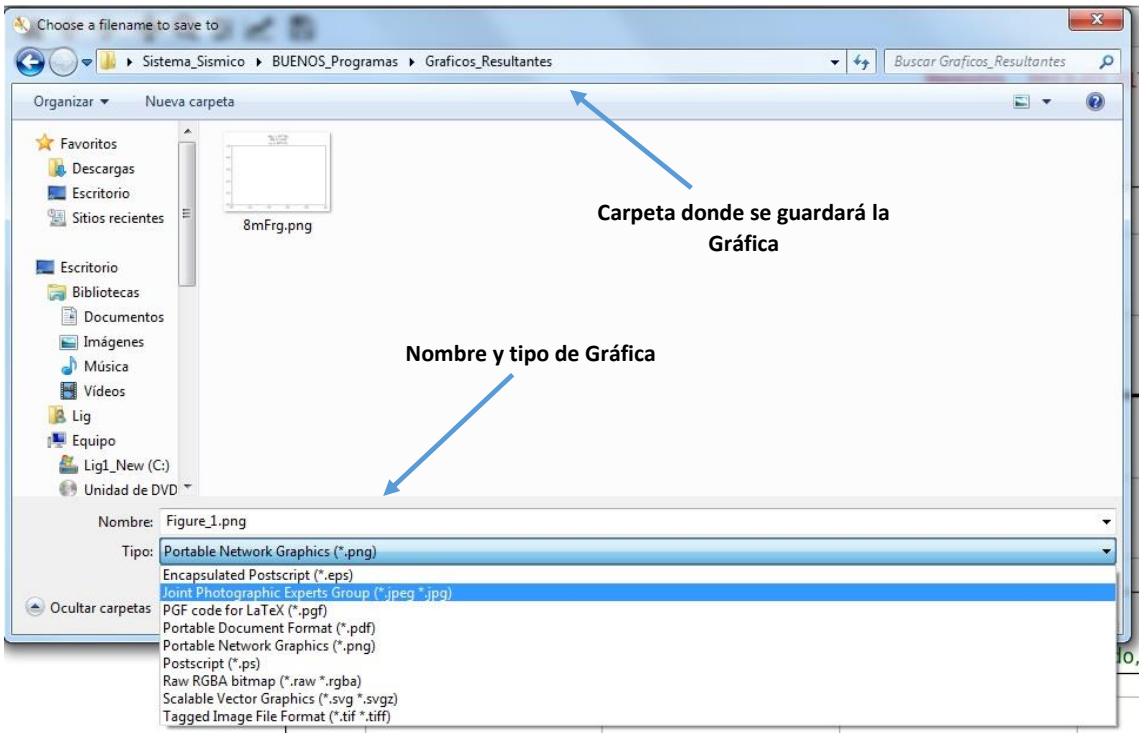


Fig. 24 Screen where you can see the types of formats available to save the graph.

The previous figure shows a list of the available file formats, the following image presents this list in more detail:

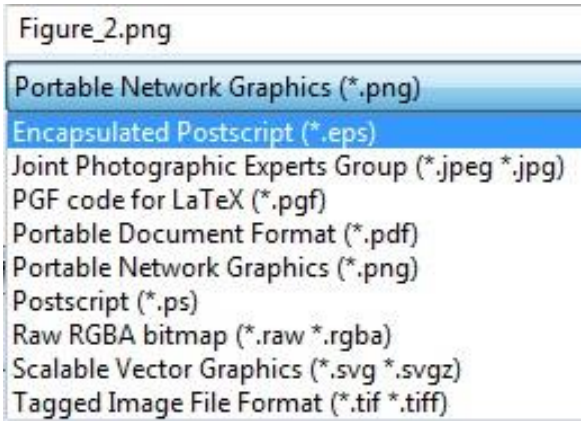
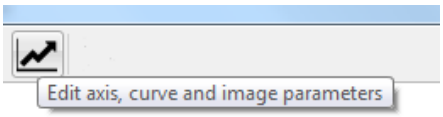


Fig. 25 List of available formats for saving the graph generated by the analysis.

Once you have selected both the name and the desired file format, and the location of the folder or directory where the graph will be saved, click the "*Save*" button (*See Fig. 23*), and the graph will be stored and available for further use as needed.

7.2.- Editing Axes and Images in Graphs.

Through the "**Edit**" button, point 7 (*See page 17*) of the graph toolbar (*Edit axis, curve and image parameters*), it is possible to edit or modify the parameters of the axes, images, and curves of the graphs.



For example, to modify the parameters of the image of a spectrogram, click on this command button. A "*Customize*" dialog box appears, indicating which of the "*axes*" in the graph areas you want to edit or modify. After selecting, click the "OK" button. This dialog box is similar to the following.

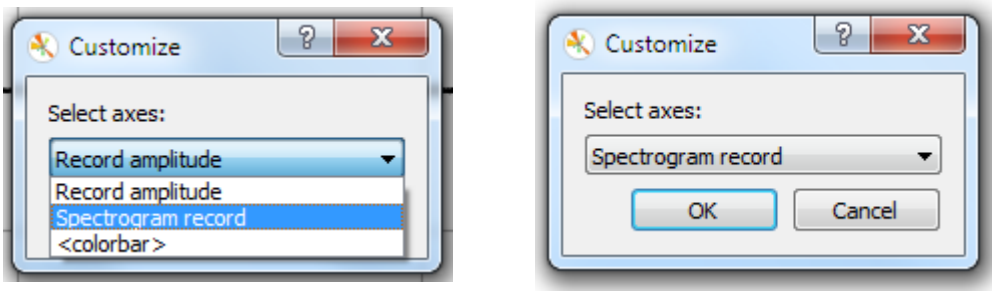


Fig. 26 Customize dialog box, the spectrogram axis has been selected.

Once the desired axis is selected, and the "OK" button is clicked, a new window with the options in the figure is presented. Here, various values of the selected axis are edited, in this case, the spectrogram (*Axes and Images*). The dialog box is as follows.

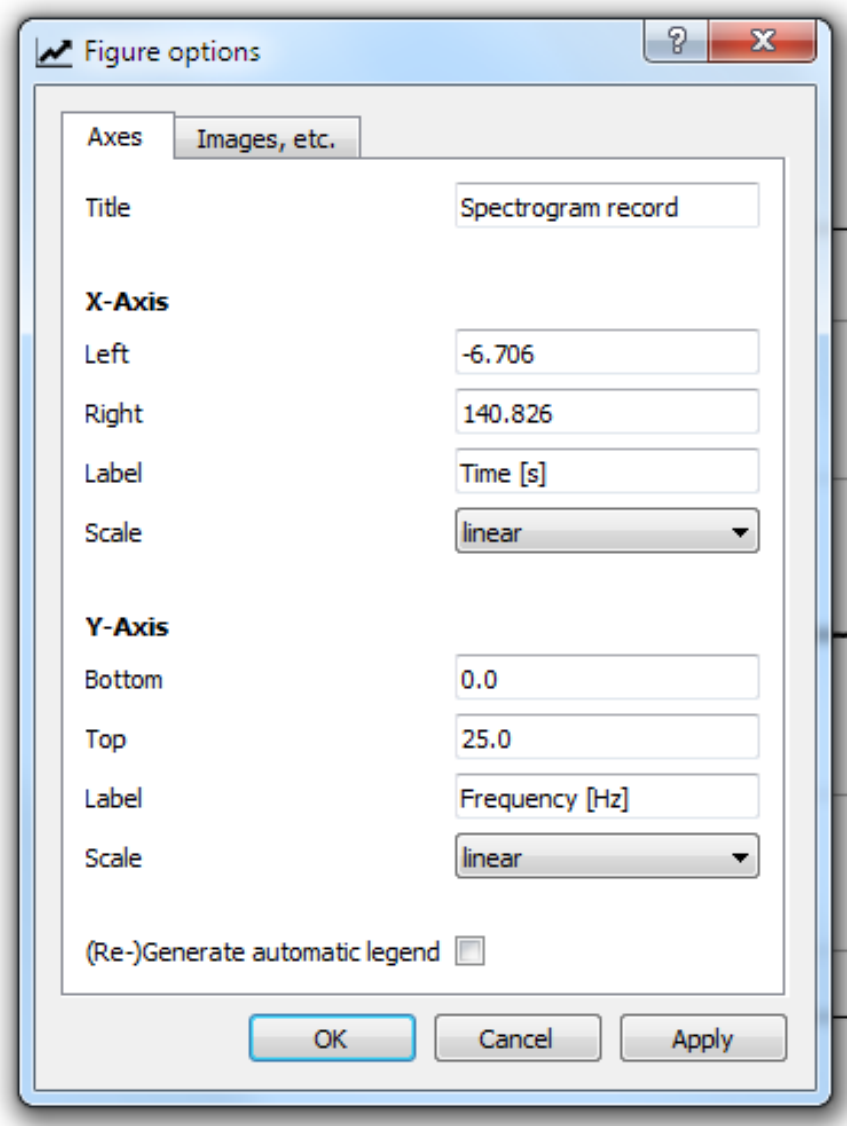


Fig. 27 Dialog box for editing options for the title and axes of the figure.

In this "*Axes*" section, as you can see, you can edit or modify the values or parameters of the title and the "X" and "Y" axes of the graph. For our example, we want to modify the image, so we will select the tab indicating this option. The image presented is as follows.

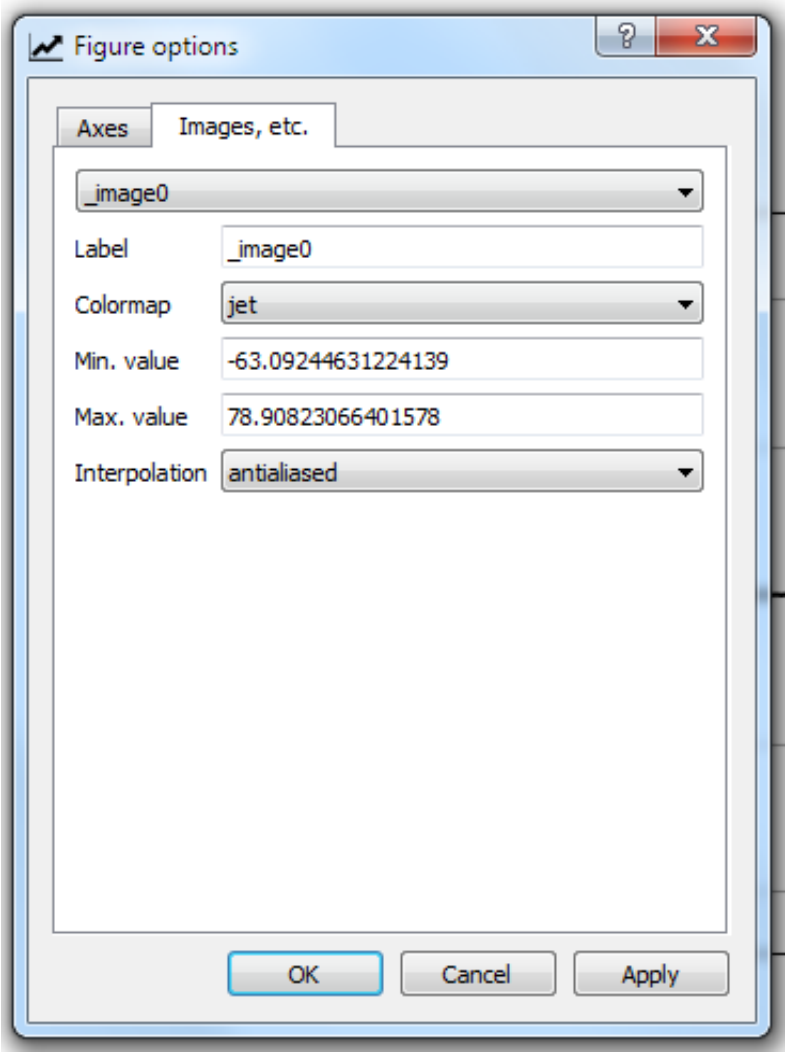


Fig. 28 Dialog box for editing options for image parameters.

As seen in the image, various parameters can be modified, including labels, the color map or "*Colormap*", used in the spectrogram, minimum and maximum values, and interpolation. The default value for the "*Colormap*" is set to "**jet**". The minimum and maximum values for this color map and the interpolation used are assigned by default to the image, but they can be modified according to the operator's interest.

The list of editable parameter values for both "*Colormap*" and "*Interpolation*" is presented in the figure on the next page.

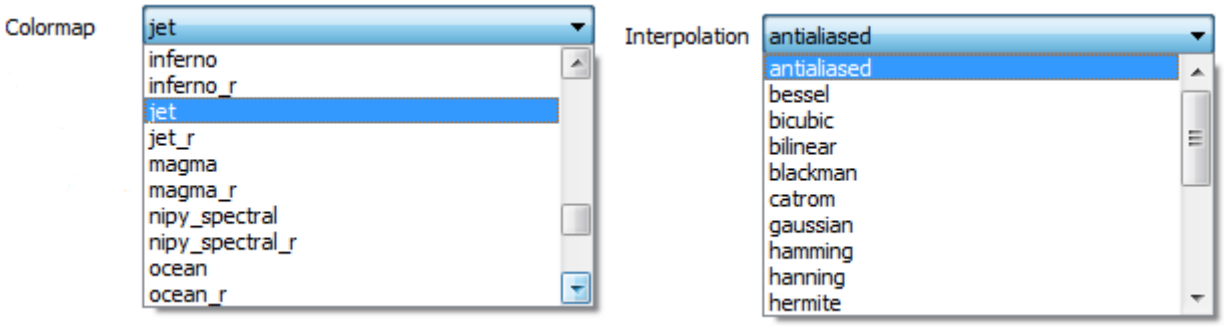


Fig. 29 Dialog boxes for editing some of the parameters of "*Colormap*" and "*Interpolation*" to select in the graph.

CONCLUSION: The system is designed to be an easy-to-use, accessible, and understandable tool. It features a user-friendly interface that provides reliable technological support to the human operator in the analysis of seismic records, both tectonic and volcanic, and allows for the merging of seismic events to create a segmented database in the future, among other uses. The simplicity of this first version lies in its single module, which includes the necessary development for selecting and merging two specific records. This is necessary because an event occurs at the end of one record and continues into the next; in this way, the complete event can be obtained within the resulting record. Subsequent versions will allow for the addition of extra modules containing various types of analysis to further the study and research of the scientific community.

Acknowledgments:

This software and its documentation are the result of research from Spanish projects:

a) PID2022-143083NB-I00, "LEARNING", funded by MCIN/AEI /10.13039/501100011033
b) JMI and LG were partially funded by the Spanish project PROOF-FOREVER (EUR2022.134044)
c) PRD was funded by the Ministerio de Ciencia e Innovación del Gobierno de España (MCIN), Agencia Estatal de Investigación (AEI), Fondo Social Europeo (FSE), and Programa Estatal de Promoción del Talento y su Empleabilidad en I+D+I Ayudas para contratos predoctorales para la formación de doctores 2020 (PRE2020-092719).
d) Spanish Project PID2022-143083NB-100 founded by MCIN/AEI/10.13039/501100011033 and by FEDER (EU) "Una manera de hacer Europa".

PLEC2022-009271""DigiVolCa"", funded by MCIN/AEI, funded by MCIN/AEI/10.13039/501100011033 and by EU «NextGenerationEU/PRTR», 10.13039/501100011033.

END of the document.

Ligdamis A. Gutiérrez E. PhD.
Theoretical and Cosmos Physics Department, Faculty of Sciences.
Geophysics Andalusian Institute.
Granada University (Ugr)
Granada, Spain – 2021 - 2023

Geophysics Andalusian Institute

APPENDIX A

A1.- Installation of Python and Additional Libraries.

A1.1. Package Content.

The main folder "***Analysis_System_1***", It contains two folders with the programs (codes and interfaces in English and documentation in Spanish and English), organized as follows:

a) Folder **"*Reliable_set_tools_system_1*"**: (*seismic analysis system assembly*). This folder must be copied to "*My Documents*". Contains the following elements:

   a. Subfolder: "***Images***" " Images necessary for program interfaces.
   b. Program: ***Menu.py***. Startup program, presentation and calling of individual modules.
   c. Program: "***Merge_1.py***". Main analysis program for selecting and merging or concatenating a seismic event present in two separate records, in order to construct a single record for analysis.

b) Folders (*Document_ES and Document_EN, depending on the language; English or Spanish*). For *English* "***Document_EN***": It consists of the following items:

   a. User Manual for the module "***7_Manual_Merge_Signals_Events_EN.pdf***" in PDF, written in English, with the necessary documentation for the use of the system interfaces. In addition, the rest of the manuals for the other modules (1-10) that make up the system are found.

   b. "***Initials Requirements.txt***" file. File containing the libraries needed to be installed on Windows through "Pip", once Python is installed.

   c. File "***README.txt***": File with general instructions for system installation.

   d. File "***Set_tools_System_1_1.bat***", batch processing executable file. It must be copied to the desktop, from there by right clicking "run as administrator", it will start the system by calling the main menu. The file will automatically search for the startup program (Menu.py) that is located in the "Set_tools_System_1_1" folder that has been previously copied to "My Documents" and will start Python, executing said program.

The system has all the elements (*programs and interfaces*) in English, except for the user manual, which is written in both Spanish and English. To install on Windows, two main actions should be taken after downloading and extracting the ". Rar" files. The first is to copy the entire folder (a) to the "*My Documents*" folder on the PC.

a) From the main folder (**Analysis_System_1**), copy the subfolder "**Document_EN**" to "**My Documents**" in Windows.
b) Copy the file "**Set_tools_System_1_1.bat**", from the "(Document/Document_ES or Document_EN)", depending on the version (Spanish or English), to the Windows desktop.

This ensures the proper use of the program. Now, we will proceed with the installation of the Python language and additional Python libraries on Windows.

A1.2.- Installing Python on Windows.

Python is an interpreted, multi-platform, and multiparadigm programming language (*it works on various operating systems, including Windows, Linux, and Mac*), utilizing two or more programming paradigms within a program-object-oriented, reflective, imperative, and functional.

In addition, Python can be enriched by a large number of programming modules, libraries, packages, or libraries installed through its package manager, "**Pip**." On Linux, the Python program and its manager "**Pip**" are installed together with the operating system. In Windows systems, however, where Python is not a native language, it is necessary to install this language beforehand by downloading the appropriate version from the Python distribution website at the following address: https://www.python.org/downloads/

On the website, the correct version should be selected based on the type of operating system on the computer, including whether it is 32 or 64 bits.

To be installed on both 32 and 64-bit systems, it is essential to note that this document and the software were created with the version available at that time, which was "Python 3.8.6", and many more versions have emerged since then. A more modern and adaptable version to the software (recommended) is "Python 10.10".

Users need to check if more advanced versions do not interfere with some of the installed libraries, such as "*Obspy*," for example. This is because everything related to Linux systems is constantly changing with updates that Python and Linux-based systems make. It is advisable to visit the website and download the most stable or tested updated version of Python that works well with this software.

Once downloaded, run it as an administrator (*right-click and "**run as administrator**"*), and the software installation wizard will guide you through the necessary steps (*just follow the instructions*).

The process takes only a few minutes. It is "recommended" to indicate during the process, when asked, to include an access path in the system's "**Path**" so that Python can be accessed from any location in Windows. If this is not done during the installation process, it must be done manually by modifying the environment variables (*more complicated*) to include the path from where Python is installed. This will not be necessary (*if indicated at the beginning*) through the installation wizard.

A1.3.- Installation of Additional Libraries.

The next step is to ensure that Python and its file manager or package manager (**Pip**) have been installed correctly. "**Pip**" (*file and library manager*) is crucial because it allows the installation of additional libraries that Python needs to run the created programs. To do this, open the Windows console window, or "*CMD*." The CMD, or command prompt, is a command-line interpreter.

Accessing the CMD is possible by typing, searching for the Windows logo key (a window), located between the "Ctrl" and "Alt" keys at the bottom left of the keyboard. Pressing this key, plus (+) the letter "**R**" key, will open a "Run" program window, similar to the following.

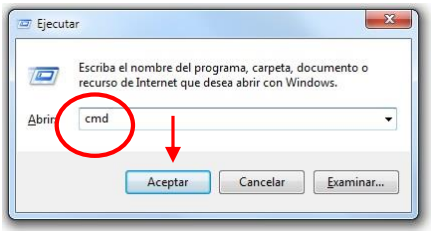


Fig. A1 Screen run in Windows. In the red circle, type "cmd" and click "OK."

As seen in the figure above, type "cmd," click "OK," which will open the Windows command prompt window.

Another way to do this is at the bottom of the desktop, in (W7) or next to (W10) the Windows "Start" button. There is the search section, indicated by the magnifying glass icon. This indicates a search for programs, similar to the following.

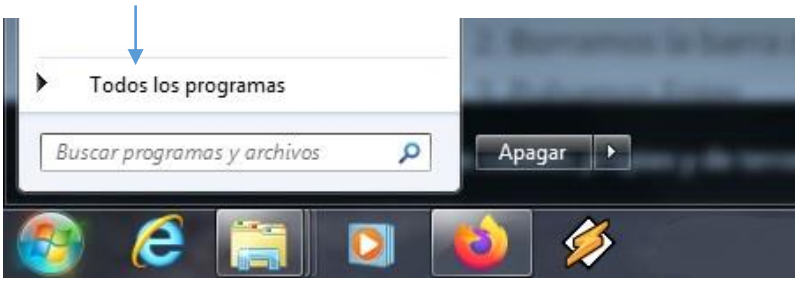


Fig. A2 Windows Program Search Screen.

In the box that says "Search programs and files" (Windows 7) or "*Type here to search*" (Windows 10), type "cmd" as well. This action or the previous one will bring up the Windows command prompt (CMD), similar to the following (W7).

Fig. A3 Windows 7 Command Prompt (CMD) Screen.

The same applies to versions: Windows 10 (W10) or Windows 11 (W11).

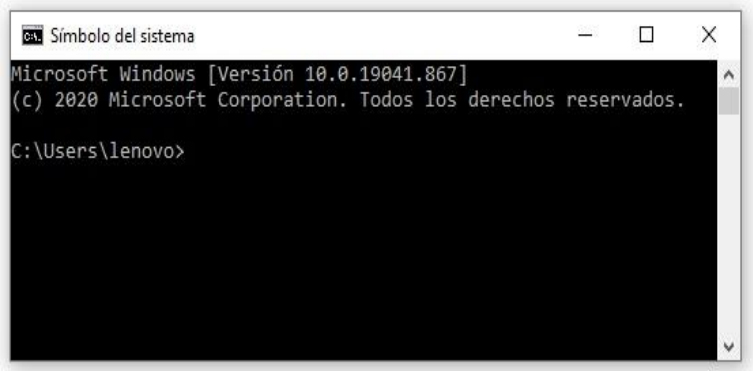


Fig. A4 Command Prompt (CMD) Screen in Windows 10.

Once on this screen, to verify that both Python and its package manager "pip" have been installed correctly, type the following commands: Python -V, and to check "pip," type: pip -V. This is shown in the following figure.

Fig. A5 CMD Screen, indicating Python and pip versions in Windows.

The output of typing "-V" in Python indicates invoking the installed version. In this case, it can be seen that it is "3.8.6" (*The library set is compatible with 3.10.10 version*). This has been possible from any location in the system because the Python script has been installed, remember, in the "**path**" or route found in the system's environment variables. Also, after typing "pip -V", it can be seen that the Pip version is "*20.2.1*." At this point, it is recommended to update this version since, by default, "Pip" is installed along with "Python", but it does not install the latest or most up-to-date version. To do this, in the CMD window or console, type the following command (*Windows/Linux*): On Windows, type "python," and on Linux, type "python3".

Windows: > python –m pip install –upgrade pip | Linux: $ sudo python3 –m pip install –upgrade pip

This indicates that "Pip" will be updated to its most recent version (*On Linux, as a "superuser," i.e., with "sudo" at the beginning*). It is shown in the following screen.

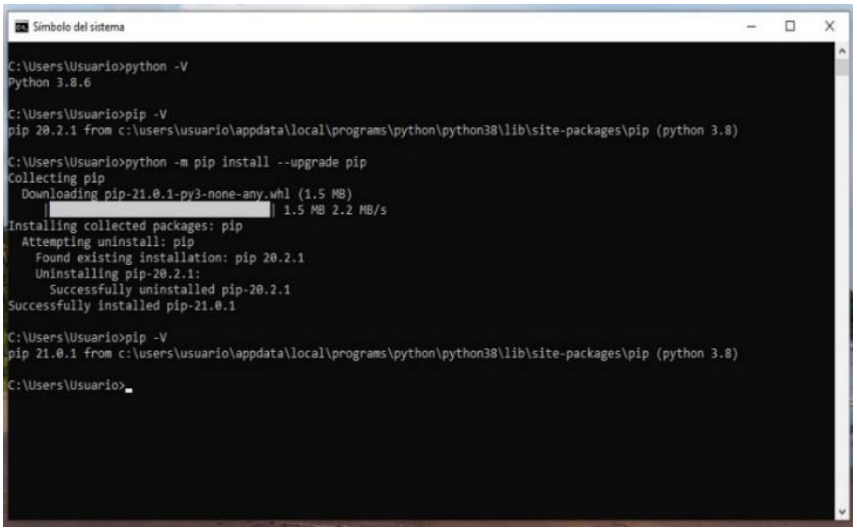


Fig. A6 Screen showing the update and verification of the new version of pip in Windows.

As can be seen, when typing again (**pip -V**), once Pip is updated, the version is 21.0.1. With this, Python and Pip are already installed and updated. Pip, as mentioned, is very important because with this manager, all the necessary libraries and packages are installed so that Python applications can be executed correctly and without errors. To use the system, you must proceed to install the necessary packages or libraries through Pip.

Next, we will proceed to explain how, in a simple and completely automatic way, the most commonly used and general libraries that Python needs will be installed on the system. Libraries such as, for example, "*obspy*," which is the open-source library or software based on Python for processing seismological data. Also, "***matplotlib***," which is a library for generating graphics from data contained in lists or arrays in Python and its mathematical extension "***NumPy***," among others, which the system needs for its execution (*See Annex B*).

A1.4 Automatic Installation of Libraries on Windows and Linux from PIP.

The advantage of having already installed and updated Pip in Windows is that you can install all the libraries that Python needs to run the system.

Additionally, in the "***Document***" folder, the "***Readme.txt***" file contains instructions for this installation. So the user only needs to follow the instructions, and the necessary packages will be installed on the computer (PC) automatically by Pip, both on Windows and Linux. The required libraries are in the file called "***Initial_requirements.txt***", included in the "***Document***" folder of the downloaded installation files and in **Annex B**.

In a Windows Command Prompt (**Cmd**) window, actions are taken for each of the commands indicated in the file, following the instructions. The installation should not present problems on Windows and Linux systems. If any library encounters an error during installation (*shown in red in CMD*), you should consult the documentation for that library or check if the correct or recommended version of Python is being installed (*version 3.8.6 and/or 3.10.10*). The installation on Linux systems (*See README.txt*) is similar and simpler. Copy the main folder to the desktop, the personal folder, etc. From that location, open a command prompt, and simply type:

"***$ python3 Menu.py***" to start the system.

APPENDIX B:

INSTALL PYTHON LIBRARIES FOR THE CORRECT FUNCTIONING OF THE SYSTEM.

1.- **PIP**: The **Pip** (*Preferred Installer Program*) is the package or package management manager used to install and manage software packages written in Python. When installing Python, Pip is installed by default. To check the version of Python or PIP, type the following in a console or CMD:

python - V / pip – And to see the list of installed pip packages: -> pip list

Usually, you need to update the version of pip with which Python is installed. For this, type the following command in the command prompt (CMD). In Linux and Mac systems, "**sudo**" is placed at the beginning to indicate super-user permissions.

Python –m pip install - - upgrade pip / (LINUX) -> sudo python –m pip install - - upgrade pip

Once downloaded and installed, you can check the version again with the first command, and you will see that the version has changed and been updated. Now that pip is updated, we will proceed to install the necessary packages for Python to work correctly with the applications.

2.- .- **PyQt Installation**: This is a Python binding for the Qt library written in the C++ language. It is used for creating and using graphical user interfaces (GUI) in Python. Type the following in the command prompt (CMD).

pip install PyQt5 / (LINUX & Mac) -> sudo python install PyQt5

3.- **Matplotlib library Installation**. Matplotlib is the library that allows the creation and visualization of graphics. Type the following:

pip install matplotlib / (LINUX & Mac) -> sudo python install matplotlib

4.- Install the **Obspy** library. This library is for handling seismic signals. Type the following:

pip install obspy / (LINUX & Mac) -> sudo python install obspy

5.- Install **Thinter**: Thinter is a graphical user interface (GUI). Type the following:

pip install tk / (LINUX & Mac) -> sudo python install tk

6.- Install **quantecon**: This library is used for spectrum estimation, Periodogram, Fourier transform. Type the following:

pip install - - upgrade quantecon / (LINUX & Mac) -> sudo python install - - upgrade quantecon

7.- Update a library for **matplotlib**. To avoid problems with graphics, install the following:

pip install msvc-runtime / (LINUX & Mac) -> sudo python install msvc-runtime

8.- Install **easygui** for the graphical interface:

pip install easygui / (LINUX & Mac) -> sudo python install easygui

9 Install **PyWavelets** for CWT handling.

pip install PyWavelets / (LINUX & Mac) -> sudo python install PyWavelets

10.- Install **plotly** for handling and assisting with graphics along with Matplotlib.

pip install plotly / (LINUX & Mac) -> sudo python install plotly

11.- Install "**pyaudio**", for audio management. Python bindings for PortAudio v19, the cross-platform audio I/O library

python -m pip install pyaudio / (LINUX & Mac) -> sudo apt-get install python3-pyaudio

Finally, type "pip list" to see the installed libraries. Optionally, you can create a file called "***requirements.txt***" that will contain all the libraries that the PC will use. The "***requirements.txt***" file must be in the current directory. The instruction to do this is as follows:

pip freeze > requirements.txt