GRANADA UNIVERSITY (UGR)

Geophysics Andalusian Institute

THEORETICAL PHYSICS AND THE COSMOS DEPARTMENT

ANDALUSIAN INSTITUTE OF GEOPHYSICS AND PREVENTION OF SEISMIC DISASTERS

"Seismic signal conversion system to Wav files"

(Sistema de conversión de señales sísmicas a archivos Wav)

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

Back to Index

1.- Introduction

The "Seismic Signal Conversion to Wav Files System" module provides a user-friendly interface that enables easy and efficient management for converting a seismic record, resulting in a sound wave file in Wav format. This facilitates the operator's analysis and recognition through sound (each type of event has a different sound). The final result of the event conversion can be stored in the Wav or Wave (Waveform Audio File Format), which is compatible with most PC audio codecs. In this way, the files can later be used for the development of calculations, analysis, and various scientific investigations. These processes provide a reliable automated tool that offers sound-based assistance to the operator in the process of spectral interpretation and analysis of seismic signals.

The application, through its incorporated libraries, allows for the reading of various seismic formats such as SAC, MSEED, GSE2, EVT, WAV, among others. Various filtering techniques can then be applied, automatically providing added value to the operator's expertise by enabling faster and more accurate identification of seismic waves compared to manual analysis in a continuous record.

The first version of this system consists of a single interface that includes tools for signal filtering and the process of cutting a specific event from an original seismic trace, whether measured in minutes, hours, or days. The main interface includes an English version of the system. However, the documentation, including this document, is available in Spanish. In the appendices, you will find information about the folder structure and its contents. The system also provides the ability to store the conversion of events at multiple frequencies according to the observer's criteria, which can be used in seismic institutes and observatories. Additionally, if desired, the event or record being converted can be graphically displayed, and the graphical 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.

Back to Index

2.- Initial Screen of the System

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.

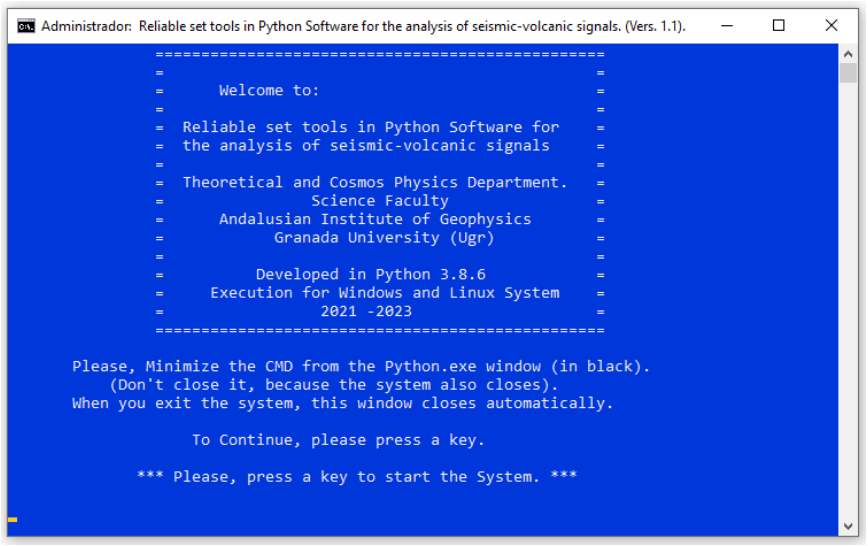


Fig. 2 Welcome Screen and Instructions for Loading the System.

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:

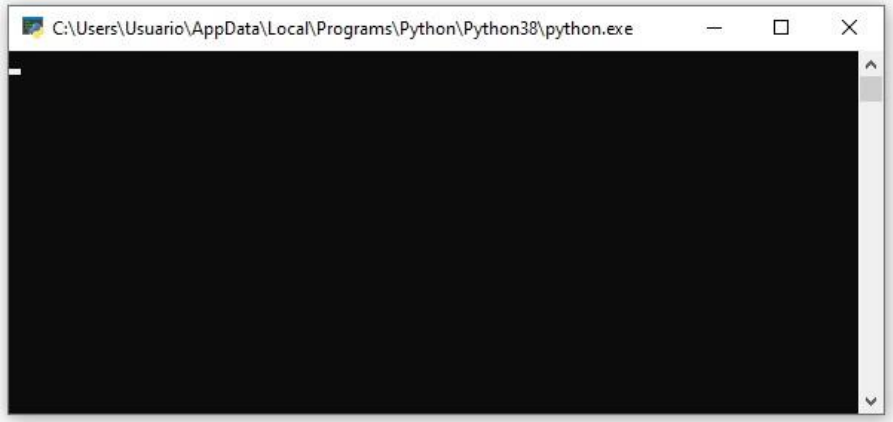


Fig. 3 Python Console (CMD) Window (*Should be minimized*)

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*).

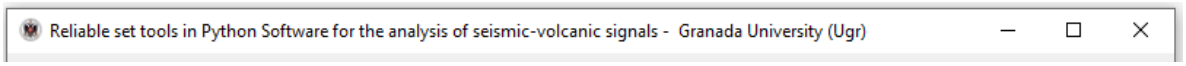
Fig. 4 **Main Menu Screen.** The module to be worked on is highlighted. Module 4 (*Converting Seismic Records to Wav*).

2.1.- Elements of the Initial Screen.

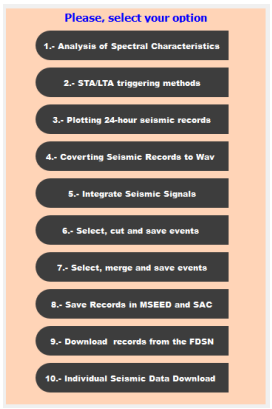
As observed in the previous figure, the initial or presentation screen is a simple window composed of:

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.

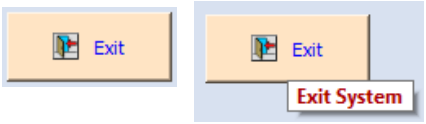
a) At the top, the icon of the University is visible, along with the module title and a reference to the University of Granada (UGR).



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.

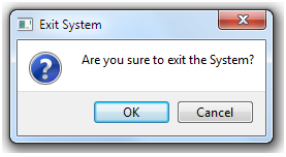


Fig. 5 Text Box Indicating Whether You Want to Exit the System.

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

Back to Index

3.- Main Interface for Wav File Conversion.

The "main analysis screen" is the module's primary interface, where activities related to the tools for reading records and converting an event or record into Wav sound format take place. This process occurs because a specific event may have a unique sound, different from other types of events (*for example, the event can be segmented using cutting software in a subsequent action, facilitating the construction of databases*). The main interface is composed of the following parts:

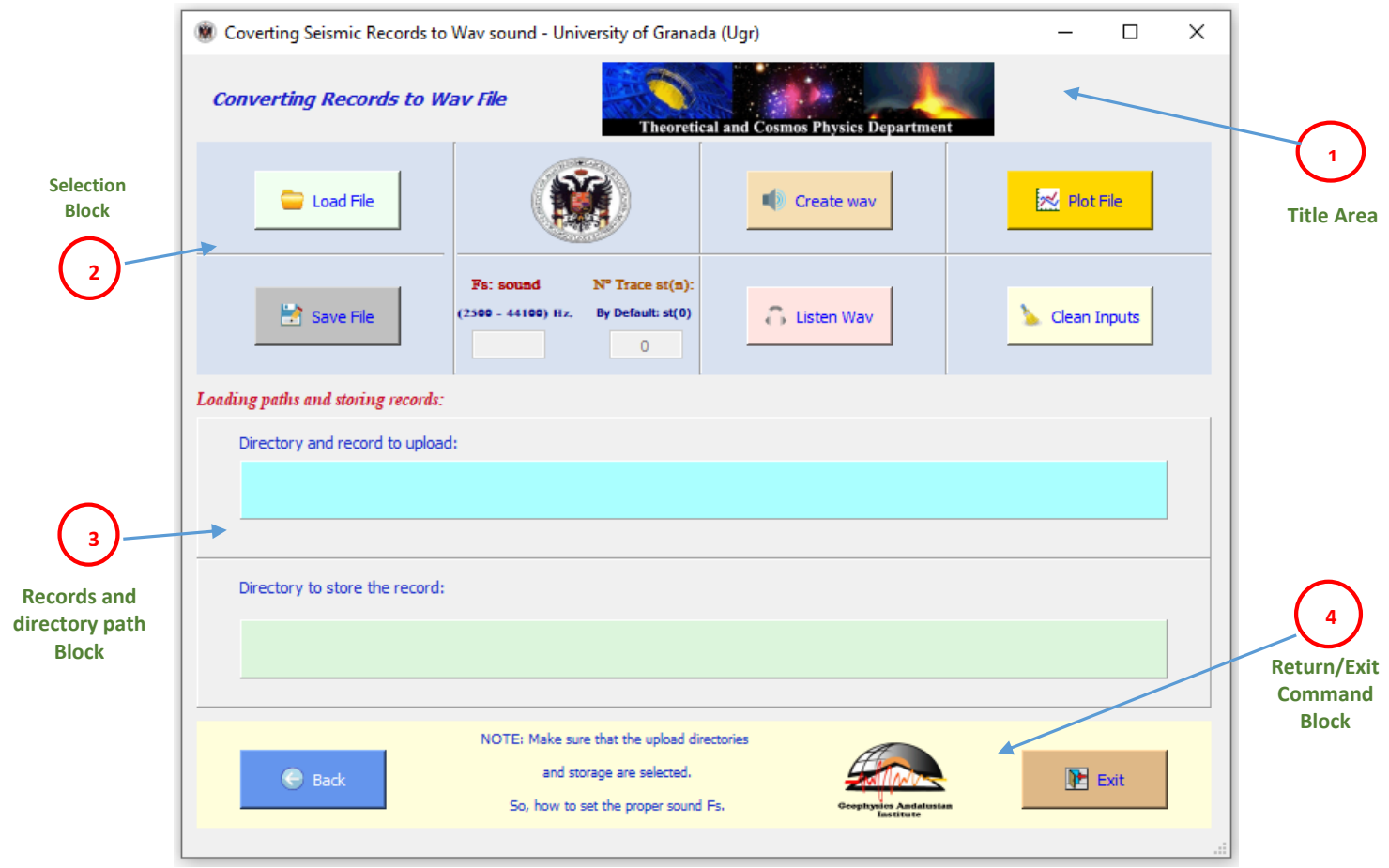


Fig. 6 Elements of the Wav File Conversion Interface.

1. **Title Area** (Program name, University).
2. **Selection Block**: a) Command buttons: Load File, Save File, Clean Inputs, and Plot File, b) Frequency for sound conversion (2500 – 44100 Hz), c) Trace number.
3. **Records Path and Directory Block**: a) Area for the physical location of the record to be converted, b) Area for the location of the directory where the resulting Wav file will be stored.
4. **Command Block**: a) Command buttons (Go Back, Exit).

3.1.- Elements of the conversion screen.

The previous screen consists of various elements for its use. At the top, you can see the Title Area (**1**), which contains the program name, icon, university name, and logo of the Department of Theoretical Physics and the Cosmos. The image of this block is shown below.

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

Added to number (1), the 3 elements of the initial fusion interface screen have been distributed into three main blocks, numbered (2-4) in red circles.

3.1.1.- Selection Block. (2)

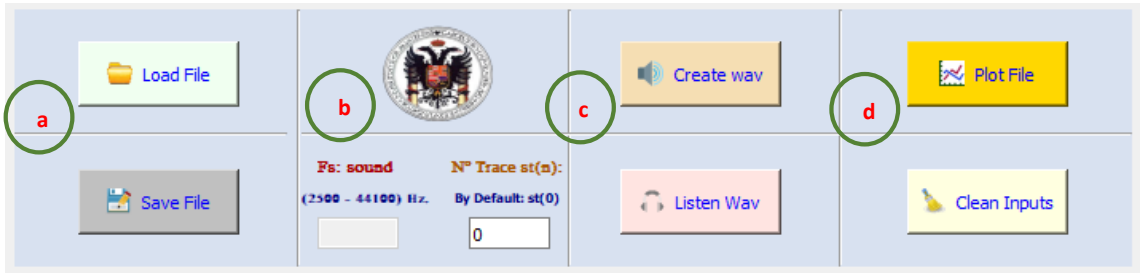
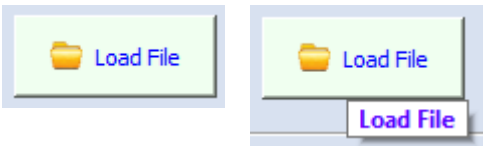


Fig. 7 Selection Elements Block. In the green circles: a) Command buttons: Load Record, Save Record; b) Data input box for Fs sound, Data input box for Trace Number; c) Command buttons: Create Wav, Listen to Wav; d) Command buttons: Plot Record, Clean Inputs.

This block is configured by the green circles from the previous figure. First of all:

a1) Command button "Load Record": Ensure that the files and directory are selected.



In the image, the buttons for "*Load File* (Cargar Registro)" are displayed. When hovering the mouse pointer over it, a message appears indicating its function. This button performs the search and loading of seismic records using the "*Obspy*" library through various formats (SAC, MSEED, SEISAN, etc.). Upon clicking, it opens a file explorer window (*by default, it starts in the root directory "**C**" of the PC*), presenting options for the various types of formats to search for and allowing searches within the computer's directory. This can be seen in the following screen.

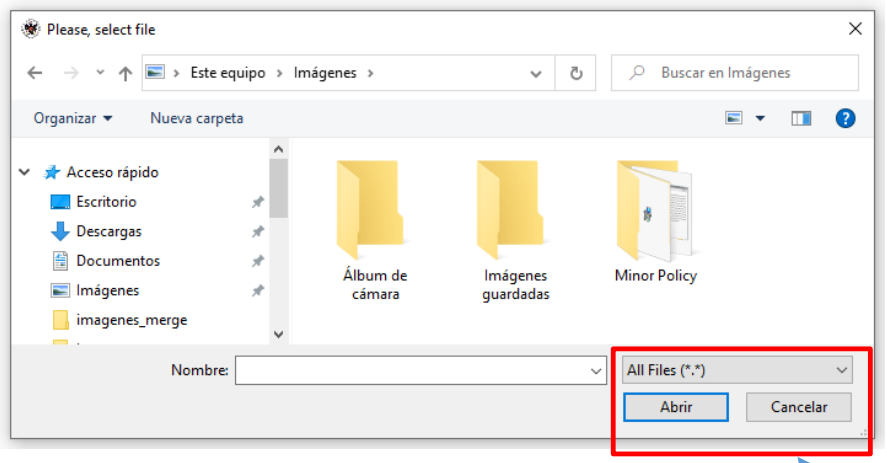


Fig. 8 Record Selection Screen.

In this screen (*the language is determined by the operating system*), records are selected according to the desired format (red box) such as SAC, MSEED, GSE2, EVT, etc. This is made possible through the "**Obspy**" library for reading seismic formats.

Once selected, clicking the "**Open**" button will load the record onto the analysis screen. Conversely, clicking the "**Cancel**" button will return the action to the analysis screen. The process of selecting a record is shown in the following screen.

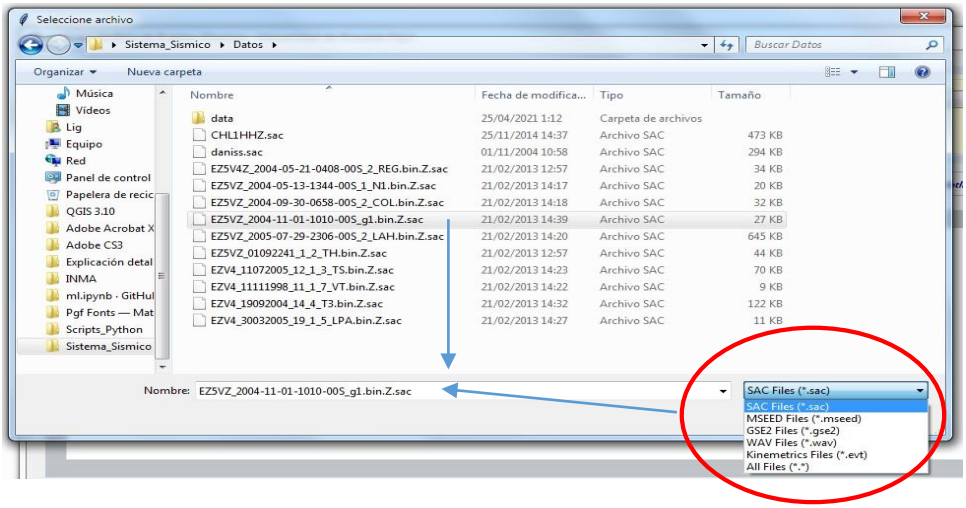


Fig. 9 Example screen of selecting a "SAC" format record.

In the screen, at the bottom right, indicated by the red circle and highlighted by the arrow, is the list of the most commonly supported and/or used seismic formats in observatories and institutes worldwide (*SAC, MSEED, GSE2, WAV, EVT*).

When selecting a specific type, the records are presented according to that format. For example, the 'SAC' files. By clicking on the desired record, as shown, it populates the "**Name**" box. At this point, you click the button that was shown in the previous screen, "**Open**", which loads the address or "**Path**" of the physical location of the record in the system.

a2) Command button "Save File": Ensure that the files and directory are selected.

The action of the "Save File" (Guardar el registro) button, displays a message indicating its function when the mouse pointer is hovered over it. Clicking this button opens a file explorer window (*by default, it starts in the root directory "**C**" of the PC*), presenting the option to select a folder or directory where the resulting event from the conversion to Wav sound will be stored. This can be seen in the following screen.

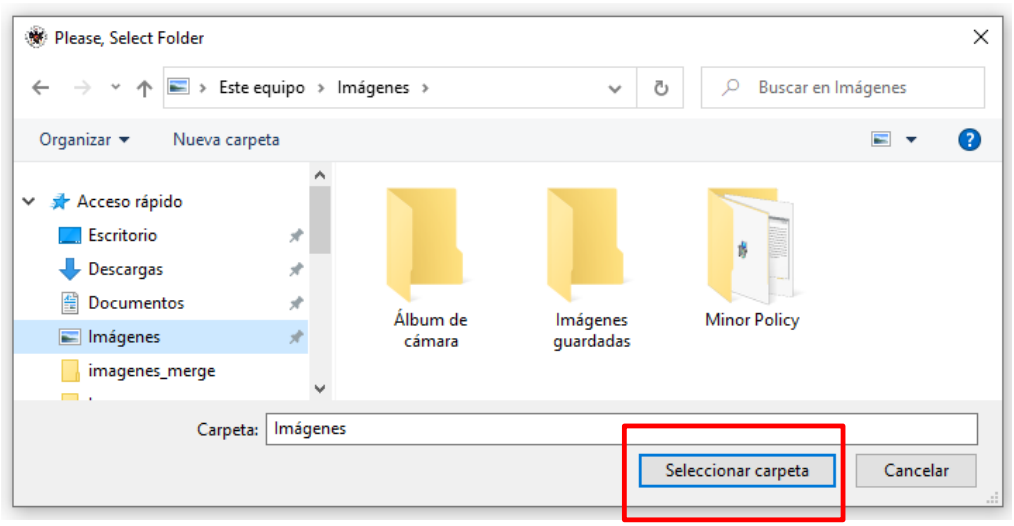
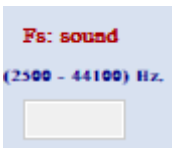


Fig. 10 Selection of the folder where the resulting event from the record conversion to Wav sound will be stored.

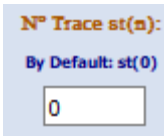
In this screen (*the language is determined by the operating system*), clicking the "***Select Folder***" button allows you to choose the folder or directory where you want the event to be saved (red box). Conversely, clicking the "***Cancel***" button will return the action to the main conversion screen.

b1) Data Input Text Box for Conversion Frequency (Fs sound).



In this data input box, the sampling frequency for the conversion must be indicated. By default, it is stipulated to work with a format of 16-bit PCM.

b2) Data Input Text Box for Trace Number (Nº trace st(n)).



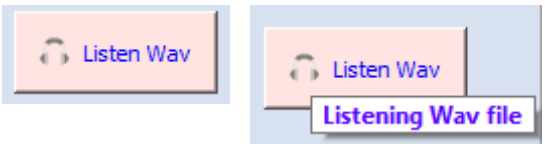
Trace Number st (0): Integer value: Determines the value of the trace that is to be analyzed. By default, in formats like "**SAC**", this value is equal to "**0**". In the case of multiple traces, such as in "**MSEED** or **SEISAN**" formats, the observer determines which trace number will be analyzed.

Back to Index

c1) Command button "Create Wav":

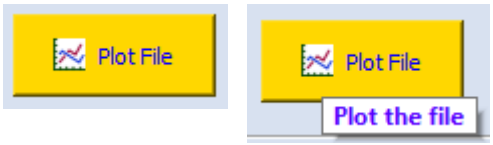
When hovering the mouse pointer over it, a message appears indicating its function. It allows for the creation or conversion of the seismic record into a sound file in Wav format. To perform this action, the following inputs must have been entered correctly: "name and path of the record to convert," "sampling frequency," "trace number," and "name and path of the directory where the converted sound record will be stored.

c2) Command button: Listen Wav.



When hovering the mouse pointer over it, a message appears indicating its function. Once the sound file has been created, it allows you to listen to it, either through the PC speakers or headphones. To perform this action, the Wav file must have been correctly stored in the designated directory or folder.

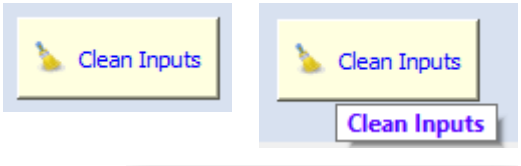
d1) Command button: *Plot File*.



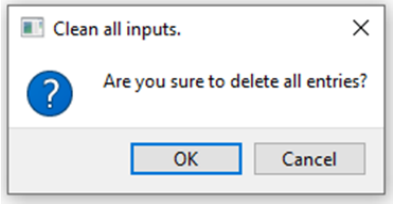
When puting the mouse pointer over it, a message appears indicating its function. This button allows you to visualize the graph of the record you wish to convert. It can be clicked regardless of whether the record has been created or converted to sound. Its purpose is to assist the human operator in visualizing the record to be converted.

d2) *Clean Inputs* Button.

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.



3.1.2.- Registry and Directory Path Block. (3).

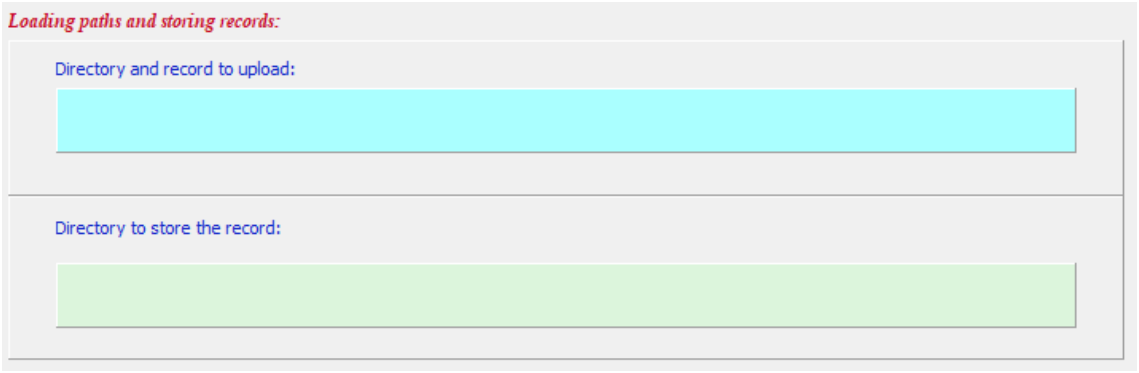


Fig. 11 Path Block, consisting of: a) Directory path to load the record (Directory and record to upload), and b) Directory path where the record converted to Wav sound will be stored (Directory to store the record). The areas where the path will be displayed for each action can be seen. Records may have more than one trace. If so, the corresponding trace for the record to be converted must be selected.

3.1.3.- Commands Block (4).

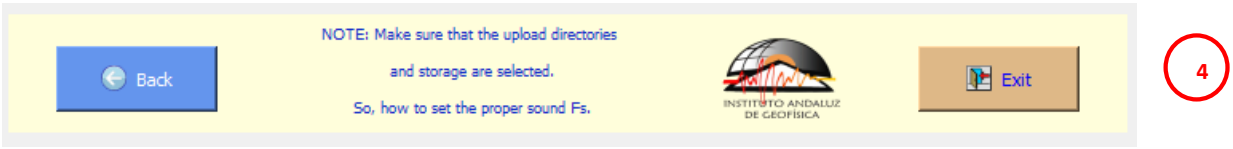
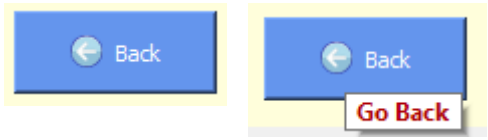


Fig. 12 Command Block, consisting of: "Go Back" and "Exit" buttons.

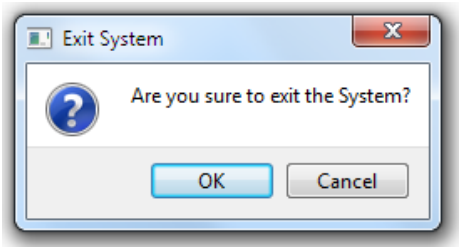
The image shows the "Go Back" button, which performs the search and loading of seismic records through various formats. The "Exit" button allows the user to exit the system.

1) **"Go Back" Button:** Allows the user to return to the initial presentation screen of the system (Index). When hovering the mouse pointer over it, a message appears indicating its function.



2) **"Exit" Button:** Allows for a complete exit from the system (after presenting a screen asking if the user wants to leave the system). When hovering the mouse pointer over it, a message appears indicating its function.

Just like in the home screen, if the "***Exit***" button is clicked, a window will appear asking the user if they are sure they want to exit the system.



Clicking "***OK***" closes the screen and completes the system exit. Clicking "***Cancel***" continues in the analysis screen.

4.- Examples of file path and directory loading.

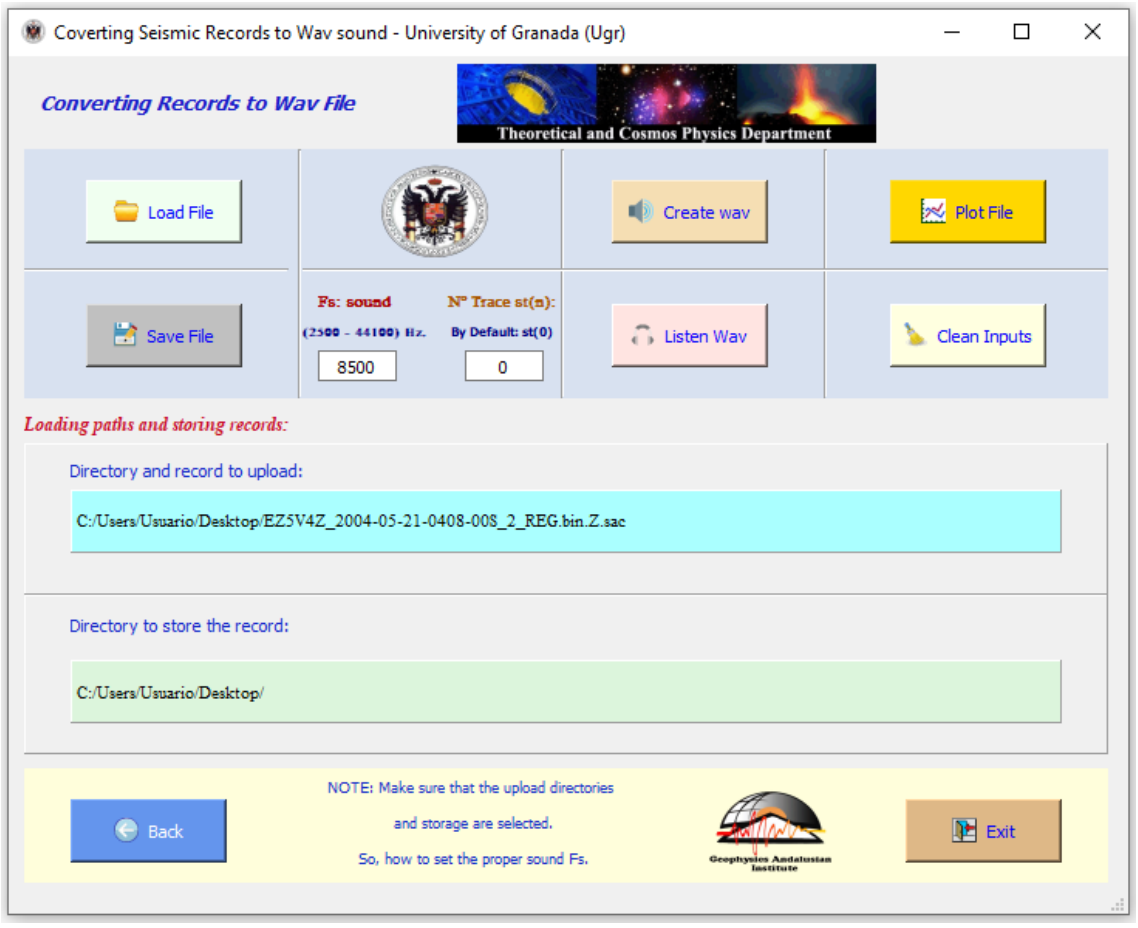
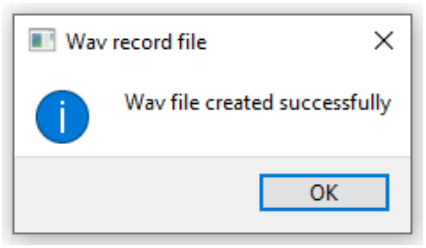


Fig. 13 Example of loading records and selecting the directory where the converted Wav sound file will be stored.

In the previous image of the interface, the SAC-type record that has been selected for conversion to Wav sound and the directory where it will be stored are shown. The sampling frequency (***samplerate***) has been set to **8500**. Once these paths have been designated, the "Create Wav" button is clicked to create and store the resulting record. If everything goes smoothly, a screen will appear indicating that the record has been created and stored successfully.



Otherwise, a series of validations will be presented that need to be resolved. Among them are the following.

5.- Error Validation in Records or Entries.

When clicking the "***Create Wav***" button, if an error occurs (invalid input, nonexistent record, or out-of-range format), a validation will be presented through several dialog boxes, displaying an alert message for this situation. This allows the user to modify the inputs or, alternatively, choose a valid record without causing the system to crash or stop. The screens that will be displayed are as follows:

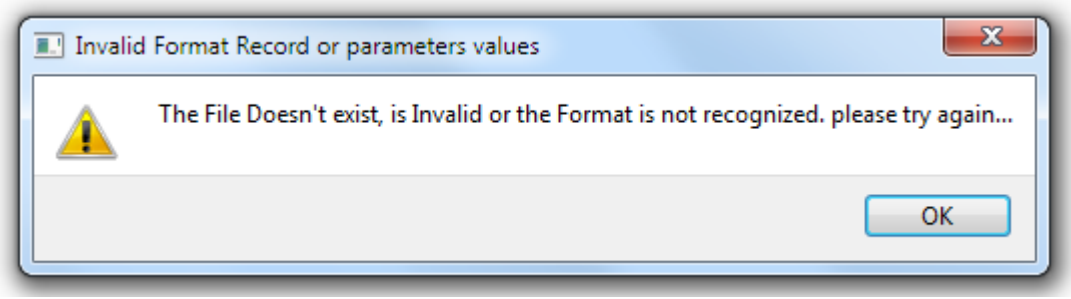


Fig. Validation of invalid inputs, formats, or incorrect records.

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 conversion cannot be performed because the record is invalid. Additionally, the parameters or inputs may also be outside the permitted range according to the signal being analyzed. By clicking the "***OK***" button, the user returns to the system to choose a valid file or correct the erroneous inputs. In this way, the program continues to execute without issues. The input validations are as follows:

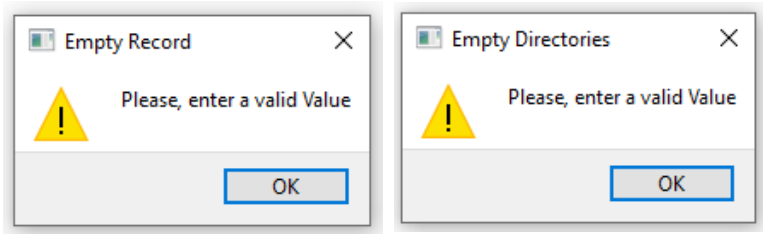


Fig. 15 Validation of empty inputs in the record and the directory where the result will be stored. A valid format must be entered in each of them.

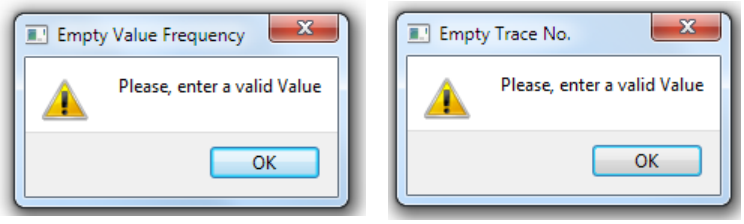


Fig. 16 Validation of empty sampling frequency input and validation of empty trace number.

6.- Results of file sections and conversión.

Next, examples of the final results from the process of selecting and converting the seismic record to a Wav file will be presented.

6.1.- Example of results from record selection and conversion.

According to the process described above, the steps for selecting and creating a Wav sound file from a seismic record are very simple and consist of the following:

a) Click the "Load File" button to select the seismic record you want to convert to sound. (The file path of the record is displayed: *Directory and record to Upload*). By default, the initial path is located in the root directory "C" of the PC, whether using Windows or Linux systems.

b) Open or select the folder or directory where the converted signal will be saved as a Wav file. (The folder path is displayed: *Directory to store the record*).

c) Enter the "Sampling Frequency (Fs)" and trace number (or leave the default trace as 0).

d) Click the "Create Wav" button to convert the record into a Wav sound file and store it in the chosen directory.

A dialog window will appear indicating that the Wav sound file has been successfully created in the folder or directory where it will be stored.

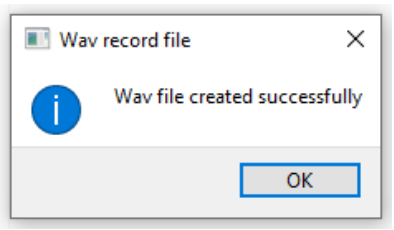


Fig. 17 Dialog box indicating that the Wav sound file has been created successfully.

Additionally, the signal or record that has been converted to sound can be visualized by clicking the "Plot File" button. The created sound can also be listened to by clicking the "Listen Wav" button. It is important to note that the duration of the record to be converted will depend on its size, so the resulting Wav sound may have a similarly longer duration. Therefore, it is necessary to consider this time when listening to the Wav sound file.

Here is an example of the graph of a record to be converted, as shown in the following figure.

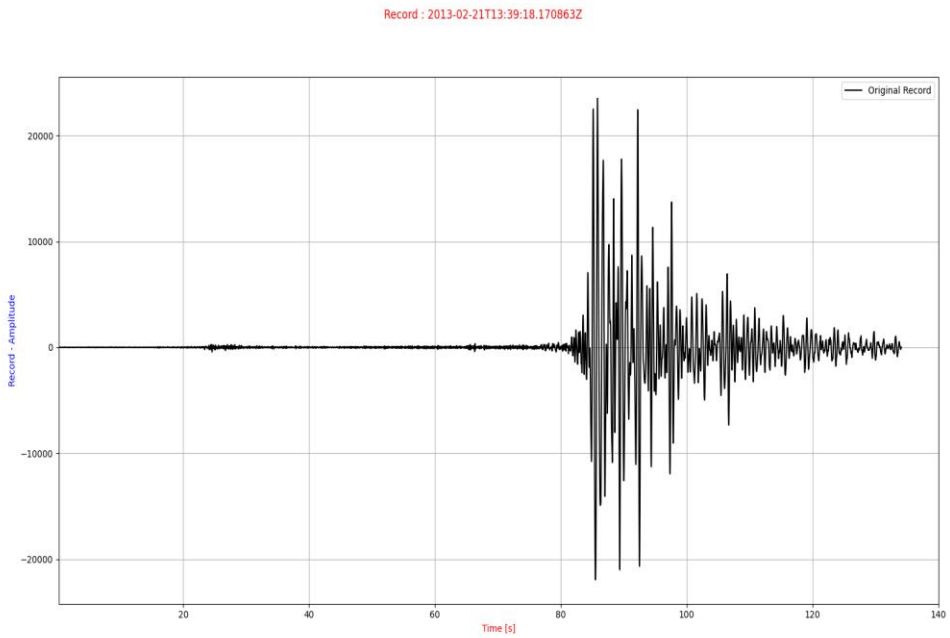


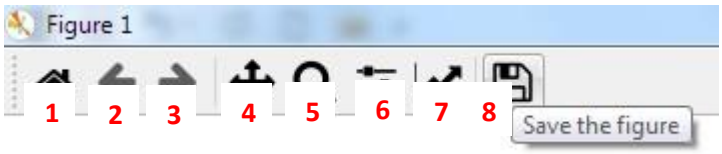
Fig. 18 An example of the graph of a SAC record to be converted into a Wav sound file.

Whether before or after saving the resulting Wav record, the resulting graph can be saved, including a zoom-in (*using the Zoom tool [Magnifier]*) applied to the original graph (*see Matplotlib Tools, pp. 17-22*).

IMPORTANT NOTE: The stored records are presented according to the original signal, exactly as it appears in the readable format. In this way, the event or record converted to Wav format will be similar (unfiltered) when saved. The user can apply filters through the analysis module (Analysis of Spectral Characteristics System of Seismic Records). Additionally, the user can use the program to cut the event (System to Select, Cut, and Save Events from Seismic Records) to carry out any further analyses deemed necessary.

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:

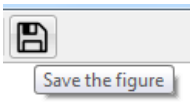
1. Reset original view: Restores all graphs to the initial state.
2. Back to previous view: Preview of the selected graph.
3. Forward to next view: Forward view of the next image.
4. Left button pans, Right button zooms, x/y fixes axis, CTRL fixes aspect. Moves the graph and axes left or right, zooms 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.

7.1.- *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.

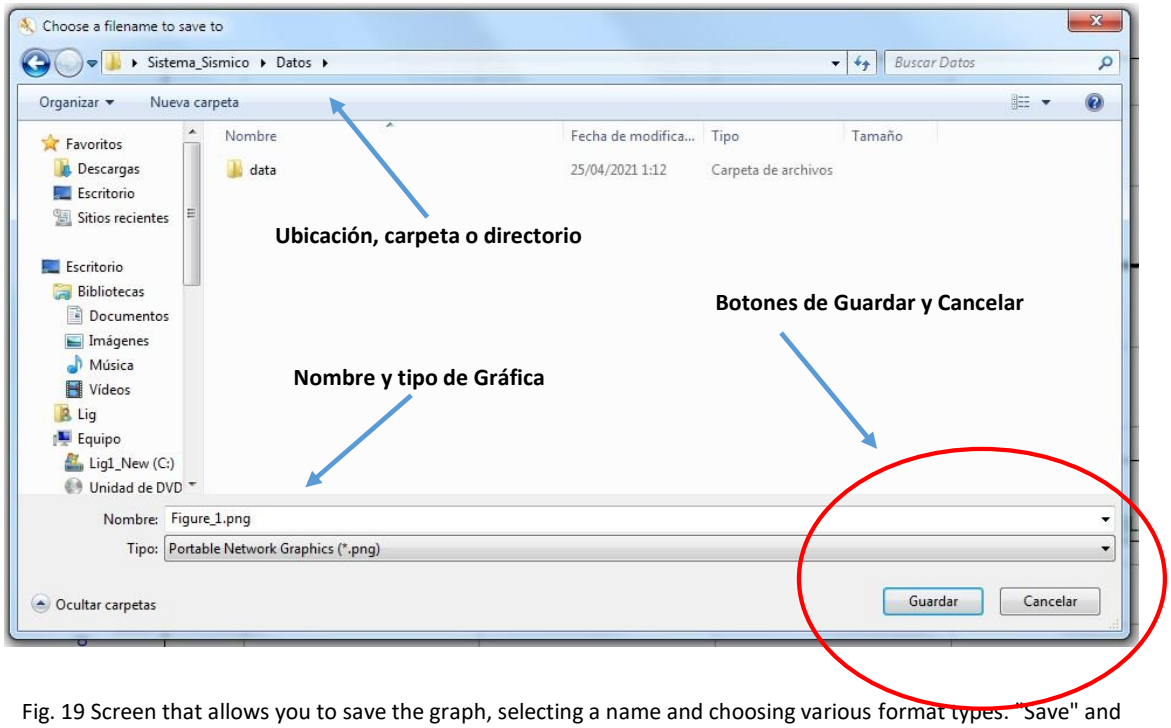


Fig. 19 Screen that allows you to save the graph, selecting a name and choosing various format types. "Save" and "Cancel" buttons are provided to complete or cancel the process.

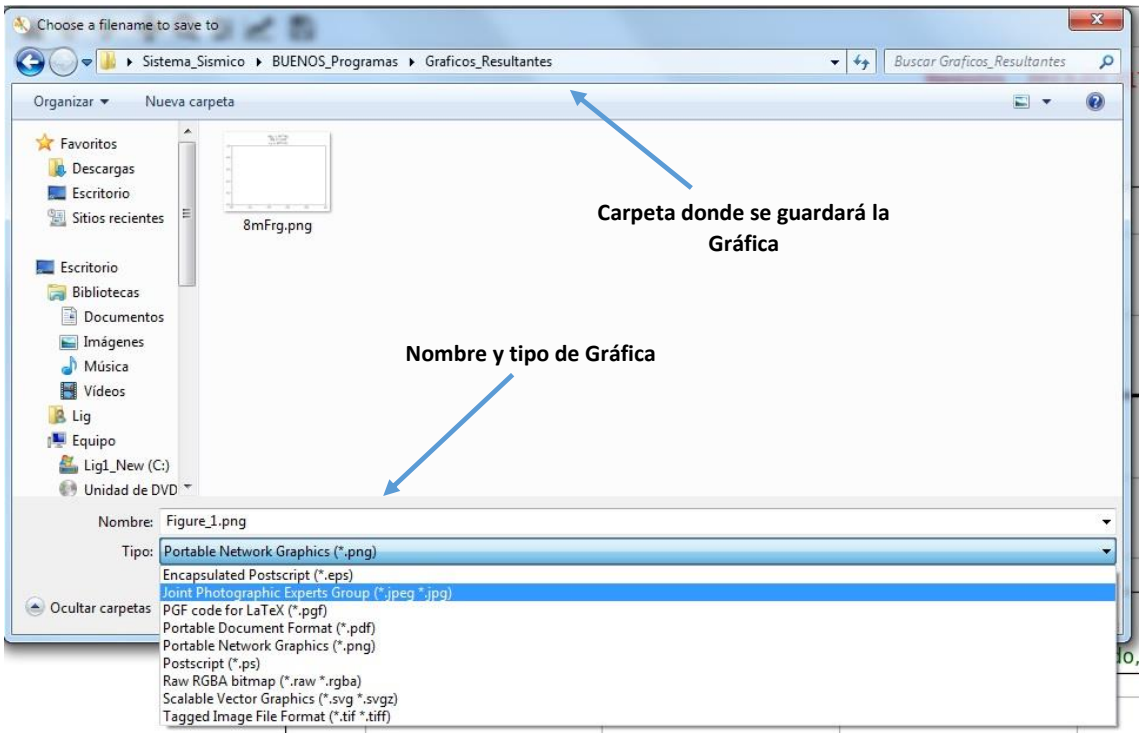


Fig. 20 Screen where you can see the types of formats available to save the graph.

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

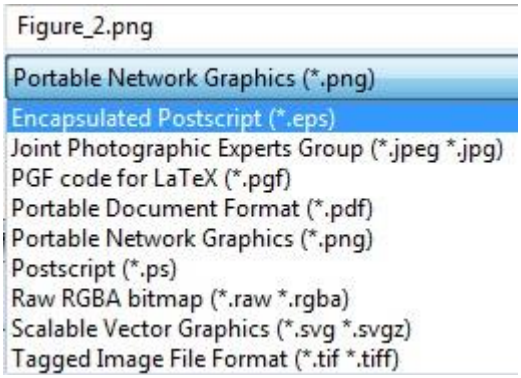
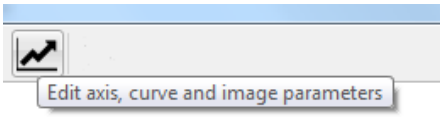


Fig. 21 List of available formats for saving the graph generated by the analysis.

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. 19*), 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 66*) 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.

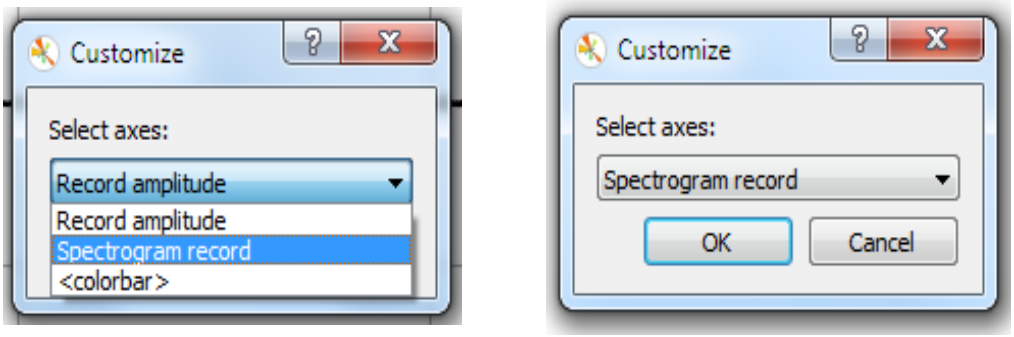


Fig. 22 Customize dialog box, the spectrogram axis has been selected.

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.

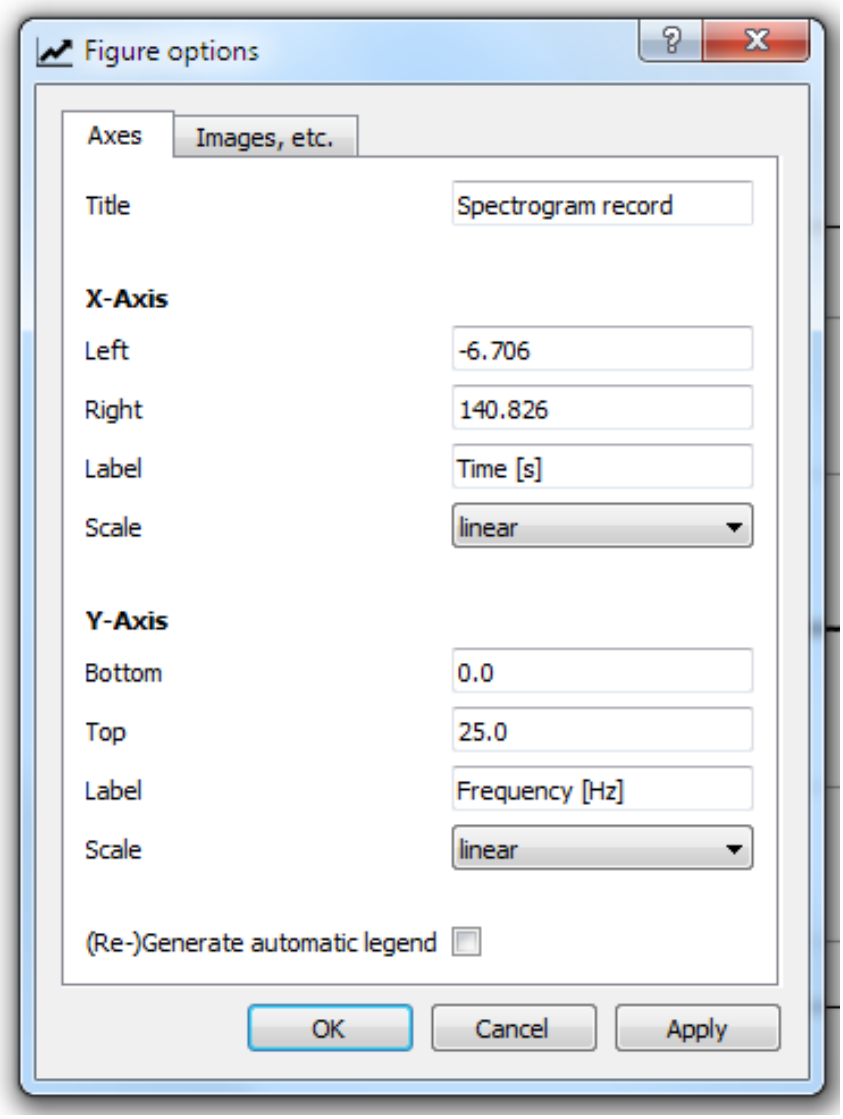


Fig. 23 Dialog box for editing options for the title and axes of the figure.

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.

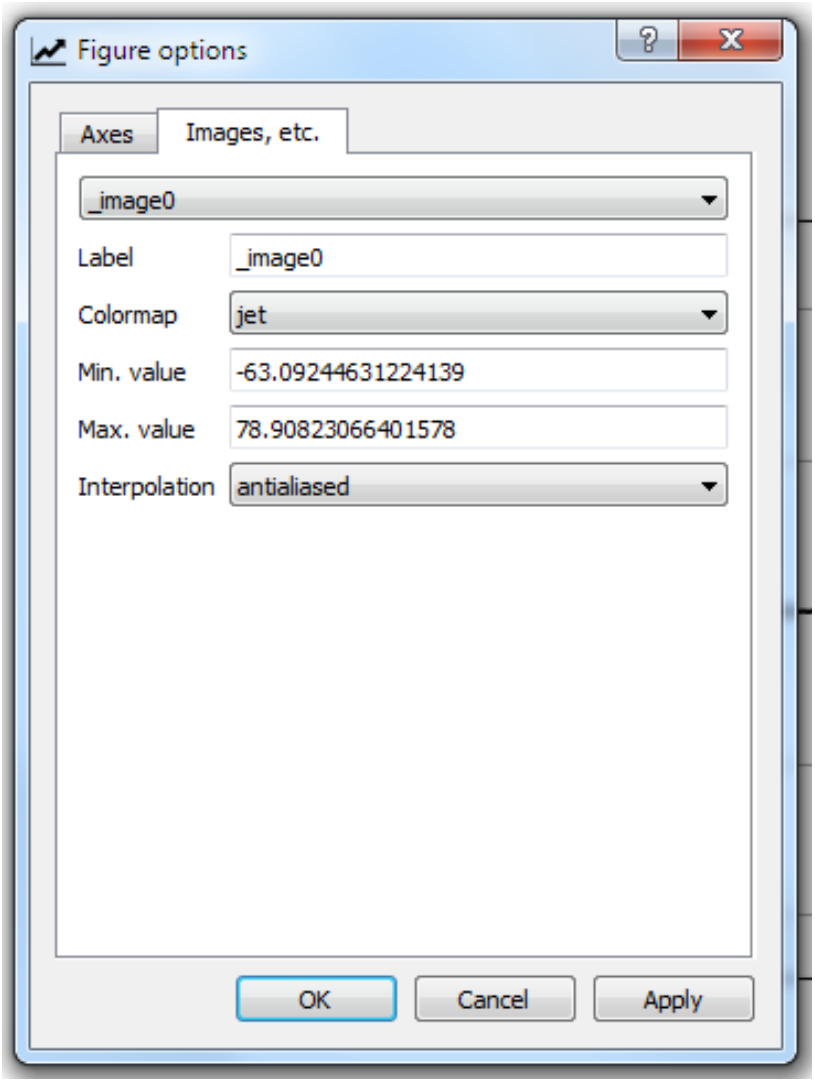


Fig. 24 Dialog box for editing options for image parameters.

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.

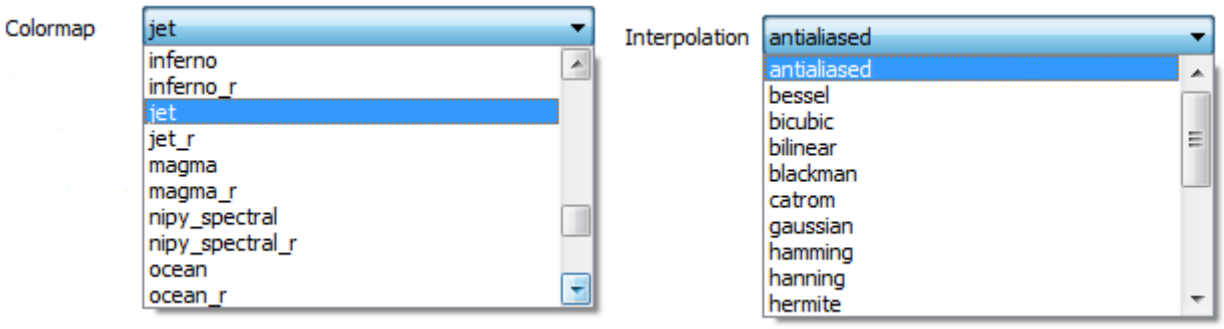


Fig. 25 Dialog boxes for editing some of the parameters of "*Colormap*" and "*Interpolation*" to select in the graph.

CONCLUSION: The module, like the rest of the system, is designed to be a tool that is easy to use, access, and understand. It features a user-friendly interface that offers reliable technological assistance to the human operator in the analysis of both tectonic and volcanic seismic records, and allows for the conversion process of seismic events into Wav sound files, to help build a segmented database through sound for future applications, among other uses. The simplicity of this first version lies in the fact that it consists of a single module, which includes the necessary development to select and convert a seismic record into Wav format. The need for this arises because each event (LP, VT, Tremor, etc.) exhibits a characteristic sound in each volcano, and using sound to identify the corresponding event sound may be helpful. In future versions, additional modules may be included to offer various types of analysis for the advancement of study and research within the scientific community.

Acknowledgments:

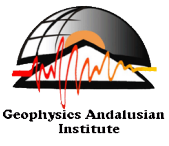
This software and its documentation are the result of research from Spanish projects:

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 – 2023



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: "***Sound_1.py***". Main analysis program to select and transform seismic records into sounds in 'Wav' format.

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: "***4_Manual_Seismic_signals_converting_Wav_file_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.

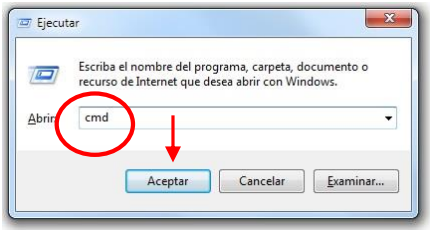


Fig. A1 Screen run in Windows. In the red circle, type "cmd" and click "OK."

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).

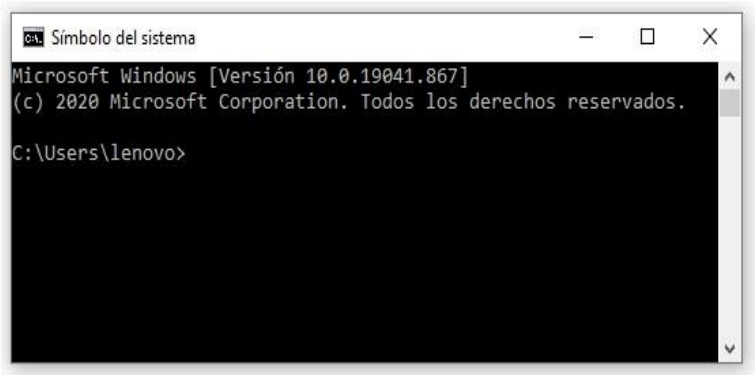


Fig. A4 Command Prompt (CMD) Screen in Windows 10.

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.

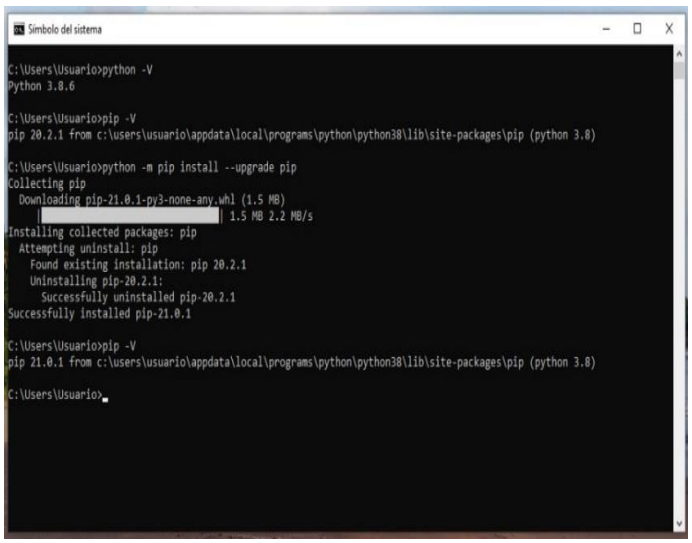


Fig. A6 Screen showing the update and verification of the new version of pip in Windows.

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 PROPER 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