

EEG Weaver Reporter. A freeware software for neurophysiologic reports

EEG Weaver Reporter. Programa gratuito para laudos de neurofisiologia

EEG Weaver Reporter. Un software Freeware para informes neurofisiológicos

Felipe Guimarães Kanda¹, Luma Hussein Salem²,
Rafael Guimarães Kanda³, Paulo Afonso Medeiros Kanda⁴

1. Medicine graduation student, Universidade de Taubaté. Taubaté-SP, Brazil. Orcid: <https://orcid.org/0009-0008-7018-3830>

2. Medicine graduation student, Universidade de Taubaté. Taubaté-SP, Brazil. Orcid: <https://orcid.org/0009-0006-8978-8317>

3. Neurology resident, Universidade Estadual de Campinas (UNICAMP). Campinas-SP, Brazil. Orcid: <https://orcid.org/0000-0002-4663-5787>

4. Neurologist (UNIFESP, H. São Paulo), Neurophysiologist, PhD (HCFMUSP), Laboratório de neurofisiologia/Neurovale. Taubaté-SP, Brazil. <https://orcid.org/0000-0002-0959-0540>

Resumo

EEG Weaver Reporter (EWR) é um aplicativo do Windows (app) cujo objetivo é disponibilizar aos neurofisiologistas um aplicativo simples que permita a implementação de relatórios de EEG e armazenamento em banco de dados, no modelo de software aberto, uma vez que alguns aplicativos disponíveis não oferecem essa funcionalidade ou produzem relatórios com aparência pouco profissional. O EWR pode ser reconfigurado para criar relatórios de outros procedimentos além do EEG. Ele é gratuito, simples de usar e seu tamanho instalado ocupa espaço mínimo no disco rígido (HD). EWR cria relatórios de EEG exportáveis para PDF e MS Word. Os usuários podem gerar quantos bancos de dados forem necessários para armazenar seus relatórios. Cada um desses bancos de dados pode armazenar modelos de relatórios de EEG reutilizáveis, classificados pelo nome do paciente. Bancos de dados e relatórios individuais podem ser compartilhados com pacientes e colegas.

Unitermos. Eletrencefalograma; laudo; EEG; neurofisiologia; *python*; *windows*

Abstract

EEG Weaver Reporter (EWR) is a Windows application (app) designed to provide neurophysiologists with a simple tool for generating EEG reports and storing them in a database using an open-source software model. This is particularly valuable as some currently available applications lack this functionality or produce reports with an unprofessional appearance. The EWR can also be configured to create reports for other procedures in addition to EEG. It is free, easy to use, and has a minimal footprint when installed on the hard disk (HD). The EWR generates exportable EEG reports in PDF and MS Word formats. Users can generate as many databases as needed to store their reports. Each of these databases can store reusable EEG report templates, categorized by the patient's name. Furthermore, both the databases and individual reports can be shared with patients and colleagues.

Keywords. Electroencephalogram; report; EEG; neurophysiology; *python*; *windows*

Resumen

El *EEG Weaver Reporter* (EWR) es una aplicación para Windows cuyo objetivo es proporcionar a los neurofisiólogos una herramienta sencilla que les permita generar informes de EEG y almacenarlos en una base de datos utilizando un modelo de software de código abierto. Esto es especialmente importante dado que algunos de las aplicaciones disponibles actualmente no ofrecen esta funcionalidad o generan informes con una apariencia poco profesional. El EWR

también se puede configurar para crear informes de otros procedimientos además del EEG. Es gratuito, fácil de usar y ocupa un espacio mínimo en el disco duro (HD) cuando se instala. El EWR permite crear informes de EEG exportables en formato PDF y MS Word. Los usuarios pueden generar tantas bases de datos como sean necesarias para almacenar sus informes. Cada una de estas bases de datos puede almacenar modelos de informes de EEG reutilizables, clasificados por el nombre del paciente. Además, las bases de datos y los informes individuales se pueden compartir con pacientes y colegas.

Palabras clave. Electroencefalograma; informe; EEG; neurofisiología; python; Windows

Research developed at Universidade de Taubaté. Taubaté-SP, Brazil.

Conflict of interest: no

Received in: 04/03/2023

Accepted in: 08/02/2023

Corresponding address: Paulo Afonso Medeiros Kanda. R. Portugal 131. CEP 12030-230. Jardim das Nações. Taubaté-SP, Brazil. Email: pkanda@alumni.usp.br

INTRODUCTION

EEG Weaver Reporter (EWR) purpose

EEG studies consist of three basic steps: recording, reviewing data, discussing, or explaining results to clinical staff. The latter is “the report”, the stage where the neurophysiologist conveys a written result of EEG visual analysis, along with an interpretation of its clinical significance. Here a clinical correlation is offered, directed at the specific diagnostic implications. The report is characterized by an outline of the study parameters, a description of the record, an interpretation or summary that includes an impression of whether the study is normal or abnormal, the degree of abnormality, and the correlation of the EEG with the clinical picture¹. Thus, the application responsible for creating the report should provide functionality for performing all the described steps.

EEG Weaver Reporter (EWR) is a Windows application (app) designed to provide neurophysiologists with a simple tool for generating EEG reports and storing them in a

database using an open-source software model. This is particularly valuable as some currently available applications lack this functionality or produce reports with an unprofessional appearance. The EWR can also be configured to create reports for other procedures in addition to EEG. EWR is a package that provides a free, an automatic and easy alternative to EEG report creation, although, it can be used to report almost anything. EWR was created because it is difficult to find a similar free alternative to generate EEG reports and because there are scenarios where some EEG softwares have limited report capabilities. Additionally, there are some excellent options for generating reports in free apps. For example, the reporter module of MNE² is a choice, although it requires proficiency in programming and produces results in HTML format. On the other hand, EWR is a Windows 11 application designed for end users. Its most impressive advantage is the ability to generate sqlite3 databanks³ allowing users to create as many report databanks as they need. Reports can be easily duplicated, edited, deleted, stored, and searched within these databanks. For instance, one can create separate databanks for 'normal EEG,' 'child EEG,' and 'temporal lobe epilepsy EEG,' each with a list of editable reports. However, it's important to note that EWR is not a comprehensive text editor with a wide range of options. It primarily focuses on generating EEG reports in two formats: Portable Document Format (PDF), standardized as ISO 32000⁴ or Microsoft word docx.

EWR is distributed freely on GNU General Public License. Everyone is permitted to copy and distribute verbatim copies of this software, but changing or use in a commercial product it is not allowed.

METHOD

EWR, developed and validated on a Windows 11 machine, is expected to be compatible with Windows 8 and 10 operating systems. However, due to variations in Python code between platforms, EWR is not designed to function on Linux or macOS systems. The software is packaged as a Windows executable file, limiting its compatibility to Windows environments.

For optimal usage, EWR requires a screen resolution of 1920x1080 pixels, offering full HD or 4K display capabilities. The implementation of EWR leverages the Python programming language, specifically Python 3.9⁵, and relies on the tkinter and ReportLab libraries. Python is currently recognized as one of the most widely used programming languages, enjoying high popularity among developers. Python is a versatile language that supports multiple programming paradigms, including object-oriented and structured programming. Additionally, Python provides support for other programming styles such as functional programming, aspect-oriented programming, and the utilization of magic methods⁶. For the development of EWR, we adopted an object-oriented approach, leveraging the

power and flexibility of Python's object-oriented features. By utilizing this approach, we were able to design the software in a modular and encapsulated manner, facilitating code organization, reusability, and maintainability.

The user interface of EWR was constructed utilizing Python Tkinter (Tk), a robust scripting and graphics library that seamlessly integrates with both Win32 and Win64 systems. Tkinter provides a reliable and efficient framework for developing graphical user interfaces (GUIs) in Python, offering a wide range of customizable widgets and powerful tools for creating interactive applications. With Tkinter's compatibility and adaptability to Windows systems, we were able to ensure a smooth and seamless user experience for EWR users on Win32 and Win64 platforms⁷.

ReportLab (RL) was the tool used to format the final EWR output. RL is a free open-source document creation engine for generating PDF documents and custom vector graphics⁸.

The codebase for EWR was developed using PyCharm Community, a dedicated Integrated Development Environment (IDE) specifically designed for Python programming and data science development, offering a comprehensive set of features and tools to enhance productivity⁹.

The primary sources of reference for coding EWR included Meier's¹⁰, Summerfield's¹¹, Roseman's¹² books, which provided valuable insights and best practices in Python development. Additionally, the YouTube channels Codemy¹³

and RfZorzi¹⁴ offered informative tutorials and instructional videos that aided in the coding process. To address complex coding challenges, the community-driven platform Stack Overflow (<https://stackoverflow.com>) served as a valuable resource. Stack Overflow facilitates collaboration among programmers, where individuals can seek assistance, exchange knowledge, and contribute to the programming community as a whole.

The icons used for buttons in EWR were sourced from reputable platforms such as <https://icons8.com/icon/> and <https://www.flaticon.com/br/>, ensuring a visually appealing and intuitive user interface.

RESULTS

The development of the EWR application resulted in the integration of three core modules: EEG_weaver_multiple_sqlite.py, responsible for building the databanks; EEG_weaver_Reporter_funcs.py, which stores the main functions; and EEG_weaver_Reporter.py, the main application initiator. All code and necessary files are stored on the GitHub repository ([https://github.com/paulokanda/EEG W Reporter](https://github.com/paulokanda/EEG_W_Reporter)). These modules are seamlessly integrated on the backend, while the frontend is represented by the user interface, activated by the file executable.

The graphical user interface (GUI) of EWR displays icons for various actions, text objects for user interaction, and a canvas to write the actual report. EWR's GUI consists of four

tabs. Upon opening EWR, Tab one (Figure 1), titled "Edit Report and create PDF," is displayed. This tab serves as the editing interface, containing four fields: "HEADER," "EEG REPORT BODY," "DOCTOR," and "END." In the "HEADER" field, the user enters the name and address of the clinic, utilizing a maximum of four or five lines. The "EEG REPORT BODY" section supports an unlimited number of lines to accommodate comprehensive reports. The "DOCTOR" field is dedicated to the reporter's name and credentials, using three lines. The "END" field allows for one final line of information or commentary. To save a new report, the user clicks "Save as New Report," and if they wish to create a copy of an existing report, they click "Save Changes Before Closing."

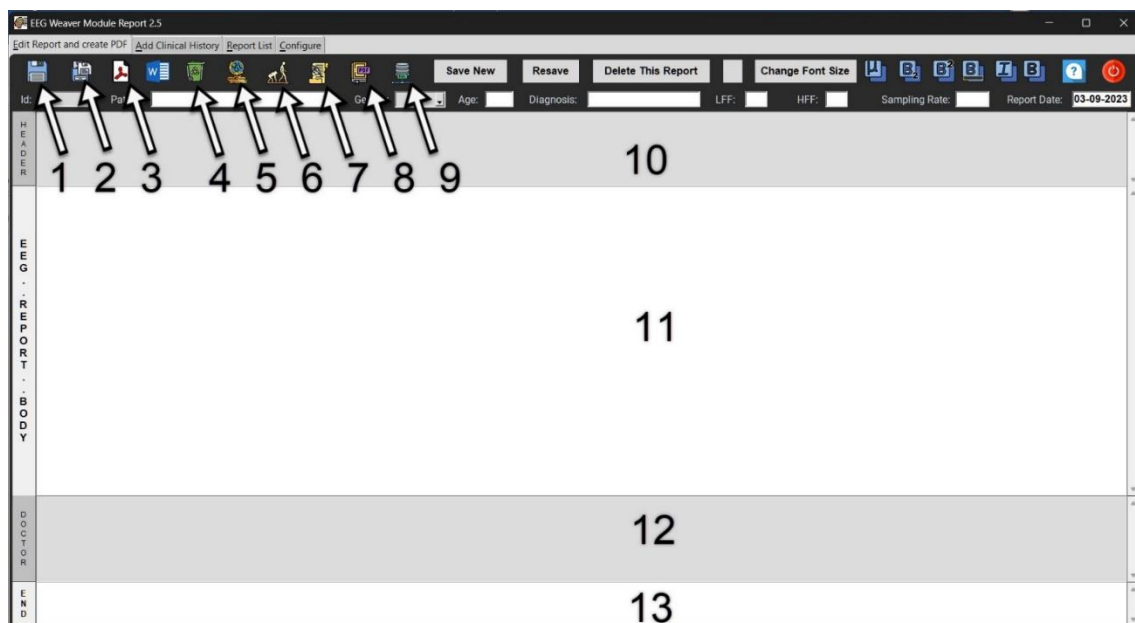
It's important to note that EWR is not a What You See Is What You Get (WYSIWYG) application. After entering each line, the user must press 'enter' to create a new paragraph, and the final result is visible only after the PDF creation process. Furthermore, EWR does not provide direct printing functionality. Instead, the user must create the PDF, store it, and then print it separately.

For a comprehensive understanding of EEG Weaver Report (EWR), a series of eight short video tutorials have been created. These tutorials can be accessed at "EEG tube" (youtube channel) playlist "EEG Weaver Reporter".

The EWR application can be downloaded from (<https://sinapsy.com.br/>) after appropriate identification. It is important to note that EWR should be downloaded only

from the author's official site, as it is an open-source software that may be subject to modifications.

Figure 1. Tab one "Edit Report and create PDF". Each number corresponds to a specific task: (1) save the report in the databank, (2) after rewriting the text it is updated here, (3) generates final PDF, (4) delete everything typed, (5) delete report but don't modify clinical story in next page, (6) save clinical story next page, (7) delete history only, (8) generate PDF only for clinical story, (9) complete remove this report and History from databank, (10) header with clinic name and address, (11) multipage body of report, (12) footer where reporter types his name and credentials, (13) free 1 line space for final message.



PDF pages will be automatically added to the report, each displaying a header and page number. The last page includes the name of the report author. An image can be added to the header, and a signature image can be placed at the bottom alongside the author's name.

Moving to Tab two, titled "Add Clinical History" (Figure 2), this page allows the neurophysiologist to register and store additional information about the patient. This

information is not printed in the main PDF but can be accessed separately if needed.

Figure 2. Tab 2 "Add Clinical History" has two new text options alongside with buttons: (1) "Delete history only" which erases only this page content and (2) "Save History" to update patient's clinical log before closing.



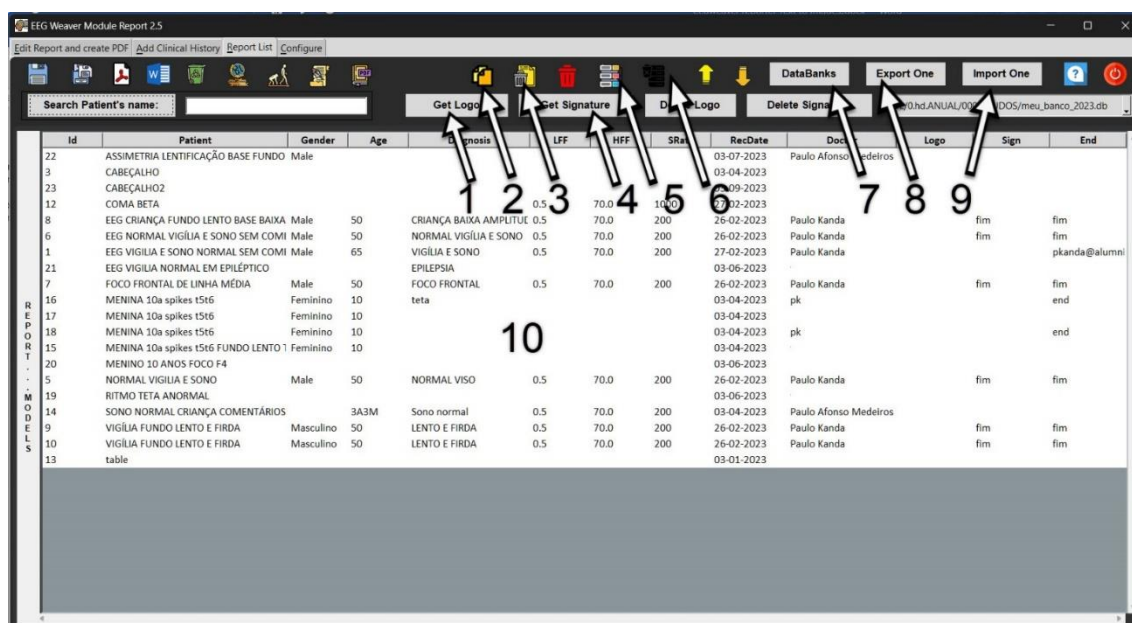
Tab three, "Report List" (Figure 3), provides an overview of the active databank, displaying a comprehensive list of individual reports. The columns in the list have self-explanatory names, facilitating easy navigation and reference.

DISCUSSION

The primary feature of EEG Weaver Reporter (EWR) is the creation of databanks that store EEG reports, allowing for easy duplication, editing, and sharing with other users. This functionality facilitates the management of clinical and

research information, in addition to generating reports on behalf of the patients.

Figure 3. Tab 3 "Report List". Other functionalities (1) get image (jpg or png) from folder to decorate the header, (2) Duplicate Report, (3) 'Delete Selected", (4) get image of a signature to decorate footer, (5) show all reports, (6) delete all reports from databank, "Delete Many" with [Ctrl-left mouse button] (7) open new window for databank creation, (8) export one report to backup and (9) import backup report. Reports inside a databank are shown in (10).



EWR was developed as a Windows application to address the limitations of existing EEG report generation tools. It provides a simple and user-friendly tool for neurophysiologists to generate EEG reports and store them in a database using an open-source software model. The ability to create multiple sqlite³ databanks within EWR is particularly impressive, enabling users to organize their reports efficiently. Each databank can be customized, providing a flexible and organized system for managing reports.

Compatibility with Windows systems is a key advantage of EWR, as it has been developed and validated on Windows 11, with expected compatibility for Windows 8 and 10 operating systems. The use of Python 3.9 as the programming language provides a solid foundation, given its popularity and versatility among developers.

To format the final output of the EEG reports, EWR utilizes the ReportLab (RL) library, an open-source document creation engine. RL specializes in generating PDF documents and custom vector graphics, providing the capability to generate reports in standardized formats such as PDF or Microsoft Word docx. This flexibility ensures compatibility and ease of use for report generation.

In addition to report generation, EWR offers additional functionalities such as the ability to add clinical history information, manage and manipulate multiple reports within databanks, and generate PDFs with automatically added headers and page numbers. These features enhance the efficiency and productivity of neurophysiological practices, allowing neurophysiologists to focus more on data analysis and clinical interpretation.

EWR is distributed freely on GNU General Public License¹⁵. Everyone is permitted to copy and distribute verbatim copies of this software, but changing or use in a commercial product it is not allowed.

CONCLUSION

EEG Weaver Reporter (EWR) provides a comprehensive solution for neurophysiologists by simplifying the process of generating EEG reports and managing clinical and research information. The creation of multiple databanks with editable reports, along with the ease of duplication, editing, and sharing, enhances the efficiency and productivity of neurophysiological practices. The user-friendly interface, compatibility with Windows systems, and utilization of popular programming tools contribute to the usability and effectiveness of EWR in clinical and research settings.

ACKNOWLEDGEMENTS

This research was funded by Neurophysiology and EEG Laboratory. All datasets and software used for supporting the conclusions of this article are available from <https://github>.

REFERENCES

- 1.Kanda RG, Cury IJ, Berto AT, Lovatel PAMK. Sugestões práticas para a descrição do EEG de rotina. Rev Neurocienc 2014;22:464-8. <https://doi.org/10.34024/rnc.2014.v22.8083>
- 2.Gramfort A, Luessi M, Larson E, Engemann DA, Strohmeier D, Brodbeck C, *et al.* MNE software for processing MEG and EEG data. NeuroImage 2014;86:446-60. <https://doi.org/10.1016/j.neuroimage.2013.10.027>
- 3.Srinath K. Python–the fastest growing programming language. J Inter Res J Engin Technol 2017;4:354-7. <https://www.irjet.net/archives/V4/i12/IRJET-V4I1266.pdf>
- 4.Bienz T, Cohn R, Systems A. Portable document format reference manual. Mountain View: Addison-Wesley; 1993.
- 5.Inden M. Short Introduction to Python 3.10. Python Challenges: 100 Proven Programming Tasks Designed to Prepare You for Anything. Berkeley: Apress; 2022; p.635-43.
- 6.Ramalho L. Fluent python. Sebastopol: O'Reilly Media, Inc.; 2022.
- 7.Grayson JE. Python and Tkinter programming. New York: Manning Publications Co. Greenwich; 2000.

- 8.ReportLab. ReportLab PDF Toolkit (open source) Wimbletech (Zone 2), Wimbledon Reference Library, Wimbledon Hill Rd, Wimbledon, London SW19 7NB, United Kingdom2022. Available from: https://www.reportlab.com/how_we_work/products/
- 9.Nguyen Q. Hands-on application development with PyCharm: Accelerate your python applications using practical coding techniques in PyCharm. Birmingham: Packt Publishing Ltd; 2019.
- 10.Meier B. Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5. Birmingham: Packt Publishing Ltd; 2019.
- 11.Summerfield M. Python in practice: create better programs using concurrency, libraries, and patterns. Boston: Addison-Wesley; 2013.
- 12.Roseman M. Modern Tkinter for Busy Python Developers: Quickly Learn to Create Great Looking User Interfaces for Windows, Mac and Linux Using Python's Standard GUI Toolkit. Victoria: Late Afternoon Press; 2020.
- 13.Elder J. Create Graphical User Interfaces With Python And TKinter. Codemy.com youtube: Codemy.com; 2022. Available from: <https://www.youtube.com/channel/UCFB0dxMudkws1q8w5NJEAww>
- 14.Serafim R. Tkinter course.RfZorzi 2022. Available from: <https://www.youtube.com/watch?v=RtrZcoVD1WM&t=41s>
- 15.Foundation TFS. Creative Commons Attribution-No Derivative Works 3.0 license 51 Franklin Street, Fifth Floor, Boston, MA 02110 The Free Software Foundation 2004 [Version 3, 29 June 2007]. Available from: <https://www.fsf.org/>