REPORTE TÉCNICO. Automatización de anonimización de datos de medición EEG.

Por: Alan Oswaldo Huerta Becerra

PROGRAMA DELFÍN 2023.

Dra. Laura Ivoone Garay Jimenez

Dra. Blanca Tovar Corona

Contenido

[**1.** **ANONIMIZACIÓN DE DATOS EXTRAÍDOS** 2](#_Toc141870367)

[**1.1.** **Introducción** 2](#_Toc141870368)

[**1.2.** **Desarrollo de código** 3](#_Toc141870369)

[**REFERENCIAS** 7](#_Toc141870370)

[**ANEXOS** 7](#_Toc141870371)

## **ANONIMIZACIÓN DE DATOS EXTRAÍDOS**

## **Introducción**

En el presente reporte técnico se muestra a detalle la codificación del programa de automatización de datos obtenidos a través del dispositivo MyDAQ a través del programa LabView ‘EEGStress.ash’. Los resultados de la extracción de datos obtenidos en señales electroencefalográficas cuentan con un sistema de adquisición de señales de 7 canales.

Las señales analizadas forman parte de la base de datos del Laboratorio de Instrumentación y Procesamiento de Señales de la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas (UPIITA-IPN). Cabe destacar que los archivos de adquisición contienen 8 variables, la primera del tiempo de adquisición (medida en segundos desde que inicia el proceso de guardado) y las 7 restantes por cada electrodo o canal de adquisición.

El presente documento toma en cuenta que el lector ya ha utilizado el programa ‘EEGStress.ash’ y ha realizado la extracción de cada una de las señales en su carpeta deseada. Tome en cuenta que, la dirección de guardado de estos archivos a lo largo del documento es ‘C:\Users\Usuario77\Documents\P.DELFIN\RAW\SUJETO #X’, donde X representa el orden en que el paciente realizó su registro para la prueba.

## **Desarrollo de código**

Con el motivo de que el lector desee realizar o replicar las pruebas de estrés y desee anonimizar estos datos, pero se cuente con un mayor número de voluntarios y el proceso de anonimización no se vuelva largo y tedioso, se propone la automatización de este proceso con base en un código Python que sea capaz de modificar las carpetas y el orden de los archivos marcados, a continuación, se realizará la explicación del funcionamiento, por etapas, del programa propuesto.

* Importación de módulos.



Los módulos ‘os’ y ‘shutil’ son utilizados para realizar operaciones relacionadas con el sistema operativo, como la manipulación de archivos y directorios.

* Definición de la función.



Esta función toma tres argumentos:

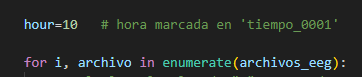
* direccion\_base: Es la ruta de la carpeta donde se encuentran los archivos EEG y la carpeta "Tiempo".
* carpeta\_destino: Es el nombre de la carpeta donde se colocarán los archivos EEG y sus archivos de tiempo.
* cantidad\_archivos: Es el número de archivos EEG que se desean organizar y mover.
* Obtención de la lista de archivos EEG en la ‘direccion\_base’.

Se obtiene una lista de nombres de archivos que comienzan con "EEG\_00" en la ‘direccion\_base’.



* Procesamiento y organización de los archivos EEG.

Se establece dentro de la variable “hour”, la hora en que se realiza el primer registro y se implementa un ciclo ‘for’ que itera sobre la lista de archivos EEG.



* Calculo de valores para la lista de archivos EEG.

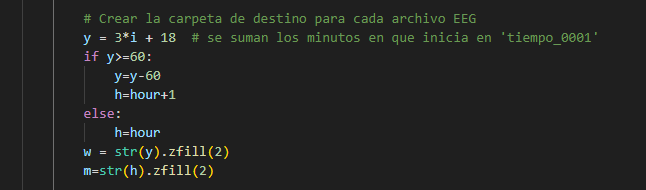
Se obtiene el valor de "x" para cada archivo EEG, el cual se utiliza para nombrar las carpetas de destino, tomando en cuenta que cualquier valor de ‘x’ se rellenará de tal manera que cuente con 3 cifras, por ejemplo, si x tiene un valor de 3, entonces se representará como “003”.



* Calculo de hora en que se realizan los registros posteriores.

Después, ‘y’ representa la variable numérica utilizada para el calculo de los minutos posteriores tomando como base ‘i’ , donde ‘y’ aumenta 3 veces ‘i’ debido a que cada registro se realiza cada 3 minutos mientras que el sumando posterior representa el minuto en que se realizó la primera medición. Además, se agrega un condicional “if”, el cual se encargará de reducir la variable ‘y’ y aumenta la variable “hour” si ésta supera o equivale el valor de 60, el cual es el valor máximo de minutos en una hora.

Al finalizar la condicional, convierte el valor numérico de “y” y “hour” a una cadena de caracteres “w” y “m”, respectivamente, utilizando “. zfill” para que esta cadena se componga de un número de 2 cifras y se acomode al nombre del documento.



* Creación de carpetas.

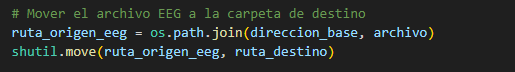
Se crea el nombre de la carpeta destino de cada archivo, utilizando las variables calculadas anteriormente y con base en el método de anonimización que se desee, en este caso, se utilizó la anonimización planteada en [Ruán J., 2023].



Enseguida, se determina la ruta donde se almacena cada una de las carpetas generadas utilizando el comando “os.path.join” y se verifica que esta ruta sí exista.

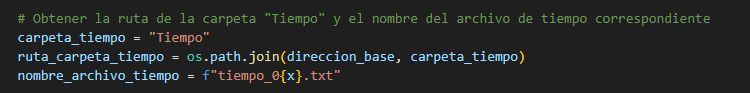


* Movimiento de los archivos “EEG\_000x.txt” desde la ‘direccion\_base’ a la ‘ruta\_destino’.



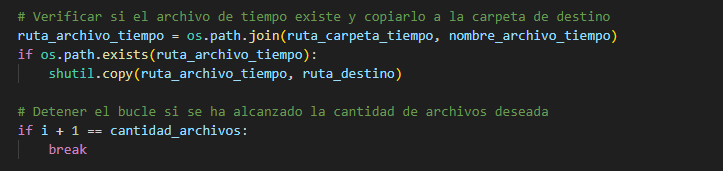
* Archivos ‘tiempo\_000x.txt’.

De igual manera, se determina la dirección en que se almacenan los archivos “tiempo\_000x.txt” utilizando el comando “os.path.join” y se selecciona también el respectivo archivo “tiempo\_000x.txt” almacenándolo en la variable “nombre\_archivo\_tiempo”.



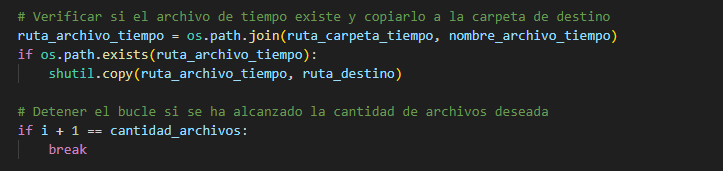
* Creación de copia para cada respectivo archivo de tiempo.

Utilizando el comando “os.path.join” se establece la ruta y el archivo que se busca, que dependerá del ciclo ‘for’ actual, además, se utiliza la condicional “if”, donde se establece que, si existe la ruta marcada, realizará una copia del archivo en la misma carpeta donde se almacenó el archivo “EEG\_000x.txt”.



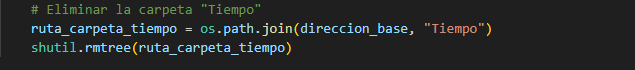
* Finalizando el ciclo “for”.

Tomando en cuenta que “i” es la variable relacionada al ciclo “for”, si este valor alcanza el valor asignado a la cantidad de archivos creados “EEG\_000x.txt”, entonces el ciclo “for” se terminará.



* Eliminación de la carpeta remanente “Tiempo”.

Se vuelve a redirigir a la carpeta “Tiempo”, tomando “dirección\_base” como la ruta especificada y, utilizando el comando “sutil.rmtree” se elimina toda la carpeta “Tiempo”, luego de haber extraído copias de todo su contenido.

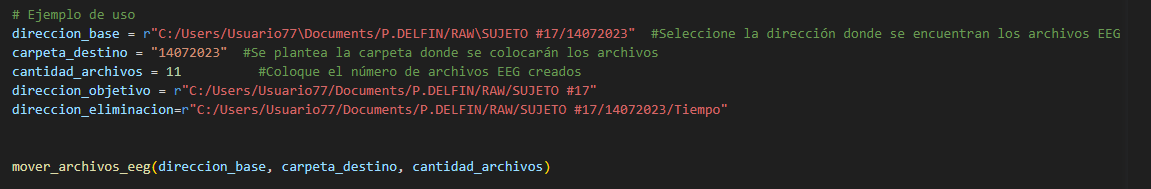


* Variables de Usuario.

Al final, fuera de la función, el usuario deberá modificar (aparte de la variable “hours”) las siguientes variables:

* dirección\_base. Corresponde a la dirección en que se ubican los archivos “EEG\_000x.txt”.
* carpeta\_destino. Corresponde a la carpeta donde deseas colocar las carpetas anonimizadas, debe encontrarse en la “dirección\_objetivo”.
* cantidad\_archivos. Corresponde a la cantidad de archivos “EEG\_000x.txt” generados por el registro.
* dirección\_objetivo. Corresponde a la ruta donde se desea generar las carpetas generadas.
* direccion\_eliminacion. Corresponde a la dirección donde se encuentra la carpeta “Tiempo” que se desea eliminar al terminar la anonimización.

Para finalizar, se ejecuta la función “mover\_archivos\_eeg” definida anteriormente.



## **REFERENCIAS**

[Ruán J., 2023]

J. Ruán-Limas, "Anonimización para Participantes", 28 de junio de 2023, Anonimización para participantes-DELFIN 2023, Ciudad de México, México, Carpeta "Registros de Atención".

## **ANEXOS**

El código empleado para la automatización de la anonimización del set de datos es el siguiente:

**import** **os**

**import** **shutil**

**def** **mover\_archivos\_eeg**(direccion\_base, carpeta\_destino, cantidad\_archivos):

# Obtener la lista de archivos EEG en la dirección base

archivos\_eeg = [archivo **for** archivo **in** os.listdir(direccion\_base) **if** archivo.startswith("EEG\_00")]

hour=**10** # hora marcada en 'tiempo\_0001'

minutes=**18**

**for** i, archivo **in** enumerate(archivos\_eeg):

# Calcular el valor de "x" para cada archivo EEG

x = str(i + **1**).zfill(**3**)

# Crear la carpeta de destino para cada archivo EEG

y = **3**\*i + minutes # se suman los minutos en que inicia en

'tiempo\_0001'

**if** y>=**60**:

y=y-**60**

h=hour+**1**

**else**:

h=hour

w = str(y).zfill(**2**)

m=str(h).zfill(**2**)

carpeta\_eeg = f"Estrés EEG\_21\_1\_0{x}\_H\_25\_140723\_{m}{w}"

ruta\_destino = os.path.join(direccion\_objetivo, carpeta\_destino,

carpeta\_eeg)

os.makedirs(ruta\_destino, exist\_ok=**True**)

# Mover el archivo EEG a la carpeta de destino

ruta\_origen\_eeg = os.path.join(direccion\_base, archivo)

shutil.move(ruta\_origen\_eeg, ruta\_destino)

# Obtener la ruta de la carpeta "Tiempo" y el nombre

del archivo de tiempo correspondiente

carpeta\_tiempo = "Tiempo"

ruta\_carpeta\_tiempo = os.path.join(direccion\_base, carpeta\_tiempo)

nombre\_archivo\_tiempo = f"tiempo\_0{x}.txt"

# Verificar si el archivo de tiempo existe y copiarlo a la

carpeta de destino

ruta\_archivo\_tiempo = os.path.join(ruta\_carpeta\_tiempo,

nombre\_archivo\_tiempo)

**if** os.path.exists(ruta\_archivo\_tiempo):

shutil.copy(ruta\_archivo\_tiempo, ruta\_destino)

# Detener el bucle si se ha alcanzado la cantidad de archivos

deseada

**if** i + **1** == cantidad\_archivos:

**break**

# Eliminar la carpeta "Tiempo"

ruta\_carpeta\_tiempo = os.path.join(direccion\_base, "Tiempo")

shutil.rmtree(ruta\_carpeta\_tiempo)

# Ejemplo de uso

direccion\_base = r"C:/Users/Usuario77\Documents/P.DELFIN/RAW\SUJETO #17/14072023" #Seleccione la dirección donde se encuentran los archivos EEG

carpeta\_destino = "14072023" #Se plantea la carpeta donde se colocarán los archivos

cantidad\_archivos = **11** #Coloque el número de archivos EEG creados

direccion\_objetivo = r"C:/Users/Usuario77/Documents/P.DELFIN/RAW/SUJETO #17"

direccion\_eliminacion=r"C:/Users/Usuario77/Documents/P.DELFIN/RAW/SUJETO #17/14072023/Tiempo"

mover\_archivos\_eeg(direccion\_base, carpeta\_destino, cantidad\_archivos)