



Diseño de Metodología de BI para visualización de datos de neurociencia computacional.

Jhorman Stiven Ramírez Giraldo

Trabajo de grado presentado para optar al título de Bioingeniero

Asesor

John Fredy Ochoa Gómez, Doctor (PhD) en Ingeniería Electrónica

Línea de investigación:

Neuroingeniería

Grupo de investigación:

Grupo de Neuropsicología y Conducta

Universidad de Antioquia

Facultad de Ingeniería

Bioingeniería

Medellín

2025

Cita	Ramírez Giraldo [1]
Referencia	[1] J. S. Ramírez Giraldo, “Diseño de Metodología de BI para visualización de datos de neurociencia computacional”, Trabajo de grado profesional, Bioingeniería, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2025.
Estilo IEEE (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina.

Jefe departamento: John Fredy Ochoa Gómez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

RESUMEN.....	7
ABSTRACT	8
I. INTRODUCCIÓN	9
II. OBJETIVOS	11
A. Objetivo general	11
B. Objetivos específicos	11
III. MARCO TEÓRICO	12
IV. METODOLOGÍA	14
A. Identificación y Selección de Herramientas de BI Gratuitas	14
B. Transformación de BIDS a CSV	17
C. Diseño de base de datos relacional.....	17
V. RESULTADOS Y DISCUSIÓN	18
A. Transformación de BIDS a CSV	18
B. Montaje de la Base de Datos.....	19
C. Creación del Dashboard en Power BI.....	22
D. Encuesta de Percepción.....	36
CONCLUSIONES.....	38
REFERENCIAS	39
ANEXOS	40

LISTA DE TABLAS

TABLA I COMPARATIVO ENTRE GRAFANA Y POWER BI.....	16
TABLA II RESULTADOS DE LA ENCUESTA DE PERCEPCIÓN.....	36

LISTA DE FIGURAS

Fig. 1. Icono de New Schema.....	20
Fig. 2. Ventana emergente para crear un nuevo esquema.....	21
Fig. 3. Tabla Portables creada dentro de MySQL.....	22
Fig. 4. Botón de Inicio para crear un nuevo dashboard.	23
Fig. 5. Botón para seleccionar la fuente de datos.	23
Fig. 6. Vista previa del modelo de datos.	24
Fig. 7. Botones para crear nuevas medidas y columnas en el modelo de datos.	25
Fig. 8. Línea de código en DAX.	25
Fig. 9. Nueva tabla despivoteada.	26
Fig. 10. Página #1 del Dashboard.	28
Fig. 11. Página #2 del Dashboard.	29
Fig. 12. Sección de objetos visuales en Power BI.	30
Fig. 13. Marketplace de objetos visuales.	31
Fig. 14. Gráfico de Cajas y Bigotes.	32
Fig. 15. Editor de la visualización de cajas y bigotes.	32
Fig. 16. Sección de filtros.	34
Fig. 17. Botón para publicar el dashboard.	34

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

BIDS	Brain Imaging Data Structure
CSV	Comma-Separated-Values
BI	Business Intelligent
DB	Data Base

RESUMEN

Este proyecto tiene como objetivo principal desarrollar una metodología de Business Intelligence (BI) diseñada específicamente para la visualización de datos en el ámbito de la neurociencia computacional. El enfoque se centra en facilitar la interpretación y análisis de grandes volúmenes de datos generados en este campo, proporcionando a los investigadores herramientas accesibles y efectivas para extraer información relevante. Para ello, se transforma el formato estándar de almacenamiento de datos BIDS (Brain Imaging Data Structure) a formatos más manejables, como CSV, compatibles con diversas plataformas de BI. Esta transformación no solo simplifica el manejo de los datos, sino que también permite la integración con herramientas gratuitas de visualización interactiva. Como resultado, el grupo de investigación Gruneco podrá explorar sus datos de manera dinámica y detectar patrones que respalden sus hipótesis científicas. La metodología propuesta se presenta como una solución escalable y adaptable, ideal para otras áreas que enfrentan desafíos similares en el manejo de datos complejos.

Palabras clave — BIDS, CSV, Base de Datos, Power BI, Análisis

ABSTRACT

The main objective of this project is to develop a Business Intelligence (BI) methodology specifically designed for data visualization in the field of computational neuroscience. The focus is on facilitating the interpretation and analysis of large volumes of data generated in this field, providing researchers with accessible and effective tools to extract relevant information. To achieve this, the standard BIDS (Brain Imaging Data Structure) data storage format is transformed into more manageable formats, such as CSV, which are compatible with various BI platforms. This transformation not only simplifies data management but also enables integration with free interactive visualization tools. As a result, the Gruneco research group will be able to explore their data dynamically, detect patterns, and generate insights that support their scientific hypotheses. This methodology is presented as a scalable and adaptable solution, ideal for other areas facing similar challenges in managing complex datasets.

***Keywords* — BIDS, CSV, Base de Datos, Power BI, Analytics.**

I. INTRODUCCIÓN

La neurociencia computacional es un campo interdisciplinario que emplea modelos matemáticos y análisis teóricos para entender los principios que gobiernan la fisiología, la estructura, el desarrollo del sistema nervioso central y las habilidades cognitivas relacionadas [1]. Con el avance de las tecnologías de adquisición de datos, como la neuroimagen y la electrofisiología, se ha generado una enorme cantidad de datos que requieren soluciones avanzadas para su almacenamiento, análisis, visualización e interpretación [2]. El motivo de esta necesidad es debido a que estos están en formatos tales como `fif`, `feather`, etc. Los cuales usualmente son procesados utilizando lenguajes de programación como Python para poder ser visualizados y analizados, y se almacenan normalmente en discos duros o almacenamientos en la nube, pues una base de datos convencional no soporta este tipo de formatos; si bien estos métodos son funcionales pues al final del día resuelven una necesidad, no son para nada cómodos puesto que por ejemplo se requiere un nivel técnico medio-alto para poder utilizar Python de forma adecuada para manipular estos datos y es necesario correr estos scripts cada vez que se necesite actualizar la visualización.

Adicional, a medida que los conjuntos de datos se vuelven más complejos, la visualización se convierte en una herramienta crucial para elucidar relaciones que no serían accesibles a través de tablas o estadísticas resumidas, permitiendo una mejor comprensión de variables y estimaciones abstractas [3]. Por lo que adicional a lo mencionado anteriormente, estos datos requieren el uso de gráficas más completas, que entreguen varias medidas estadísticas al tiempo, como lo podrían ser gráficas de cajas y bigotes o gráficas de violín.

En respuesta a esta necesidad, este proyecto propone el uso de una base de datos relacional y una herramienta de Business Intelligence (BI) como solución para Gruneco, ofreciendo una forma confiable y sostenible en el tiempo para almacenar y visualizar sus datos neurocientíficos; el motivo de esta propuesta, es debido a que las herramientas de BI resuelven dos problemáticas presentadas anteriormente, la necesidad de un nivel técnico medio-alto en un lenguaje de programación para poder generar estos análisis y visualizaciones y la necesidad de utilizar gráficas que entreguen información estadística más completa de los datos. Esto es debido a que en general la curva de aprendizaje de una herramienta de BI es mucho mas corta que la de un lenguaje de programación,

y estas herramientas traen una gran variedad de objetos visuales, desde gráficos de barras y gráficos de línea hasta gráficos de dispersión y de cajas y bigotes.

El grupo de investigación maneja sus datos en el formato BIDS (Brain Imaging Data Structure), un estándar que organiza los datos de imágenes cerebrales en una estructura de directorios clara y accesible, permitiendo así tener una estructuración que facilite el compartimiento de forma más organizada los volúmenes de datos [4]. Sin embargo, debido a las limitaciones inherentes de las bases de datos relacionales, que solo permiten el almacenamiento de datos en formato tabular, es necesario transformar los datos BIDS a un formato compatible, como CSV, para integrarlos efectivamente en la solución propuesta. Sin embargo, existe una limitante y es que existen herramientas de BI comerciales muy útiles en el mercado, pero con un costo muy elevado, es por esto que se explorará las versiones gratuitas de estas herramientas.

En síntesis, este documento se encuentra organizado de la siguiente manera: 1) el marco teórico que ilustran los principios fundamentales en los que se basó el desarrollo de este trabajo, 2) el apartado de la metodología, en donde se describe y se detalla cada uno de los procesos necesarios para dar cumplimiento a cada uno de los objetivos propuestos, 3) la sección de resultados y discusión, en donde se presentan cada uno de los resultados obtenidos, de acuerdo al esquema presentado en la metodología y se realiza el análisis de los resultados y 4) donde finalmente se muestran las conclusiones sobre el proyecto realizado.

II. OBJETIVOS

A. Objetivo general

Diseñar una metodología de Business Intelligence (BI) para la visualización de datos de neurociencia computacional que facilite la interpretación y análisis de grandes volúmenes de información para la toma de decisiones.

B. Objetivos específicos

Seleccionar una herramienta de BI gratuita adecuada para la visualización de datos de neurociencia computacional.

Diseñar una base de datos relacional para almacenar los datos obtenidos por el grupo de investigación.

Implementar técnicas de análisis y visualización de datos que permitan explorar patrones y tendencias en los datos.

Validar la propuesta en una base de datos de registros de EEG y una encuesta que recoja la percepción de los usuarios finales.

III. MARCO TEÓRICO

El proyecto planteado se puede dividir en 3 etapas: transformación de la data, montaje de la base de datos relacional y montaje de la herramienta de BI.

La primera parte representa un reto pues los archivos BIDS tienen una estructura de directorios, lo que dificulta la transformación al conocido formato tabular. Actualmente no existen librerías que por sí solas hagan esta transformación, por lo que será necesario crear nuestras propias funciones en Python para este objetivo. Aun así, no será necesario transformar todo el archivo BIDS, solo es necesario hacerlo con las características, pues es lo que más valor tiene para el grupo de investigación y lo más usado por ellos.

Para la segunda etapa, a parte de una base de datos relacional en local, se podría plantear el uso de un servicio de Data Warehouse, pero ¿Que es Data Warehouse? Un data warehouse (almacén de datos) es un tipo de sistema de gestión de datos diseñado para habilitar y dar soporte a las tareas de inteligencia empresarial (BI), especialmente las analíticas. Los data warehouses solo se han diseñado para realizar consultas y tareas de análisis, y suelen contener grandes cantidades de datos históricos. A menudo, la información dentro de un data warehouse proviene de una amplia gama de fuentes, como los archivos de registro de aplicaciones o las aplicaciones de transacción. Un almacén de datos centraliza y fusiona grandes cantidades de datos de múltiples fuentes. Gracias a sus capacidades analíticas, las organizaciones pueden obtener información empresarial valiosa a partir de los datos y mejorar las decisiones. Con el tiempo, se construye un registro histórico de gran valor para los expertos en datos y los analistas de negocio. Gracias a estas funciones, un almacén de datos puede considerarse la "fuente única de datos" de una organización [5]. Nótese que un data warehouse es una base de datos relacional con una increíble capacidad de procesamiento gracias a los servidores que poseen aquellas empresas que ofrecen este servicio; sin embargo, es debido a la potencia que estos servicios ofrecen, representan un gran costo que usualmente solo las compañías dedicadas al manejo de datos se pueden permitir. Por este motivo y porque adicional los datos que posee Gruneco no son tan grandes, se puede optar por trabajar con una base de datos relacional en local.

Por último, la etapa del BI; ¿Que es BI? El Business Intelligence, BI o la inteligencia de negocios combina análisis de negocios, minería, visualización, herramientas e infraestructura de

datos, además de prácticas recomendadas para ayudar a las empresas a tomar decisiones basadas en los datos [6].

Para la etapa de la herramienta BI ocurre algo similar con la base de datos, Las herramientas que hay en el mercado como los son Power BI, Tableau, Looker, Qlik, Grafana, etc. Han sido pensadas para las compañías, por lo que suelen ser costosas, sin embargo, existen versiones gratuitas de estas mismas (con menos funciones cabe aclarar) como, por ejemplo:

- Power BI: Power BI es una colección de servicios de software, aplicaciones y conectores que funcionan conjuntamente para convertir orígenes de datos sin relación entre sí en información coherente, interactiva y atractiva visualmente. Sus datos podrían ser una hoja de cálculo de Excel o una colección de almacenes de datos híbridos locales y basados en la nube. Power BI permite conectarse con facilidad a los orígenes de datos, visualizar y descubrir qué es importante y compartirlo con cualquiera o con todos los usuarios que desee [7].
- Grafana: Es un stack (pila) completamente administrada en la nube, ofrecido como un servicio. Grafana Cloud es la forma más rápida de adoptar e incluye un Backend administrado y escalable para generar métricas, registros y seguimientos. Muy útil para aquellos que no quieren tomar la carga de alojar la solución en las instalaciones y no quieren preocuparse por administrar toda la infraestructura de implementación. Se ejecuta en clústeres de Kubernetes. El backend es compatible con Prometheus y Graphite. Por lo tanto, podemos elegir la instancia en la nube de Grafana o ambas. Gestionado y administrado por Grafana Labs con opciones gratuitas y de pago para individuos, equipos y grandes empresas. Incluye un sólido nivel gratuito con acceso a métricas de 10.000, registros de 50GB, rastreos de 50GB, retención de datos de 2 semanas y 3 usuarios [8].

- Para el desarrollo de este proyecto se revisará a fondo estas opciones.

IV. METODOLOGÍA

Para dar cumplimiento a los objetivos establecidos en el presente trabajo se llevó a cabo la siguiente metodología la cual consta de tres etapas principales: Identificación y Selección de Herramientas de BI Gratuitas, Transformación de BIDS a CSV y Diseño de base de datos relacional. La correcta elaboración de esta fase es fundamental para el correcto desarrollo de los objetivos.

A. *Identificación y Selección de Herramientas de BI Gratuitas*

Lo primero en definirse fue cual herramienta de BI utilizar para este proyecto, las características indispensables que se estaban buscando en para este proyecto eran dos: que tuviera una versión gratuita, esto debido a la naturaleza del proyecto y del grupo de investigación el cual no se podría permitir costear un herramienta costosa y segundo que fuera una herramienta con un catálogo de visualizaciones que permitiera presentar datos tan complejos de analizar como lo son los datos de EEG, esto debido a que requiere gráficos que permitan hacer análisis estadísticos. Luego de mucho revisar se redujo los candidatos a dos opciones: Power BI y Grafana.

Power BI es quizá la herramienta más conocida internacionalmente para trabajar en BI, esto debido a la facilidad que presenta para conectarse a distintos motores de bases de datos, la simplicidad y flexibilidad de su capa semántica que permite crear relaciones entre tablas y crear nuevos campos a partir de las tablas conectadas, sin mencionar su gran variedad en visualizaciones, teniendo unas tan simples como simples tarjetas que presentan un dato en concreto, hasta otras mas complejas como gráficos de dispersión, gráficos de cajas y bigotes, etc.

Por su parte Grafana es reconocido principalmente por ser una herramienta completamente Open Source (con versión paga para su versión cloud), esto permite que tenga un alto nivel de personalización en sus dashboards, pues permite crear tus propias visualizaciones si necesitas algo mas personalizado, por lo que no sería un problema el tipo de visualización requerido para el análisis de los datos de EEG.

Al final después de mucho analizarlo, se decidió utilizar la herramienta de Power BI por los siguientes motivos:

- Debido a sus cualidades, ambas herramientas cuentan con una gran comunidad, lo que puede facilitar el aprendizaje de la herramienta. Sin embargo, Power BI tiene una curva de aprendizaje muchísimo más corta debido a que está diseñada para ser usada por usuarios no técnicos como gerentes o líderes corporativos, mientras que Grafana al ser Open Source es necesario un poco más de conocimiento técnico para sacarle todo el provecho; este punto es importante debido a que en el grupo de investigación no solo hay Ingenieros trabajando sino que también hay profesionales de la salud como médicos, esta facilidad de uso que presenta Power BI permitiría que no solo los ingenieros estén a cargo del análisis estadístico de los datos.
- El precio de su versión Pro. Si bien uno de los puntos fundamentales de este proyecto es que la herramienta sea gratuita, mirando más a futuro es posible que se requiera funcionalidades extra que solo trae las versiones pagas como por ejemplo el trabajo colaborativo en nube o poder embeber estos dashboards en páginas web. Para este caso sucede que el precio de la versión de paga más económica de Power BI es de 9,40 €/mes y la versión premium son 18,70 €/mes por usuario, mientras que la versión de paga más económica de Grafana es de 17,94 €/mes y la más costosa de 282,31 €/mes. Estas cifras tan diferentes son lo que ayudaron a determinar la herramienta a utilizar.

A continuación, se presenta un cuadro comparativo más detallado:

TABLA I
COMPARATIVO ENTRE GRAFANA Y POWER BI

Características	Grafana	Power BI
Enfoque principal	Monitoreo en tiempo real, series temporales y datos técnicos.	Análisis de datos empresariales, creación de informes y dashboards interactivos.
Orígenes de datos	Muy fuerte en datos técnicos (Prometheus, InfluxDB, Elasticsearch) y algunos SQL/NoSQL.	Gran soporte para datos empresariales, bases relacionales, archivos y servicios en la nube.
Personalización	Altamente personalizable, especialmente con plugins y desarrollos específicos.	Personalización media, con acceso a visualizaciones adicionales desde su marketplace.
Facilidad de uso	Requiere más conocimientos técnicos, especialmente para la configuración inicial.	Amigable para usuarios no técnicos, ideal para principiantes en análisis de datos.
Alertas	Alertas avanzadas integradas, muy útiles para monitoreo proactivo.	Alertas limitadas en Power BI Service; no es su enfoque principal.
Interactividad	Paneles dinámicos, pero menos sofisticados en exploración ad hoc.	Muy fuerte en interactividad para explorar datos y aplicar filtros en tiempo real.
Escalabilidad	Altamente escalable para entornos técnicos y grandes infraestructuras.	Escalabilidad empresarial con opciones Premium para grandes volúmenes de datos.
Código abierto	Sí, completamente open source (aunque tiene opciones empresariales como Grafana Cloud).	No, es software propietario con licencias gratuitas y de pago.
Facilidad de integración	Excelente integración con herramientas DevOps y de monitoreo.	Excelente integración con el ecosistema de Microsoft y plataformas empresariales.
Costo	Gratuito (open source), costos adicionales para Grafana Cloud.	Versión gratuita, pero las versiones Pro y Premium tienen costos de suscripción.

B. Transformación de BIDS a CSV

Los datos al estar presentados en formatos BIDS fue necesario hacer una transformación para poder usar esos datos. Se hizo uso de Python para poder transformar los datos a CSV, ya que BIDS es un formato tipo directorio que guarda los archivos en carpetas, el código debía encargarse de recorrer las carpetas buscando los archivos de interés. Para este proyecto se utilizó unos datos previamente recolectados por el grupo de investigación el cual contiene archivos de tipo tsv, fif, json, etc. Este primer base de datos contiene la información de 10 sujetos, los datos de estos sujetos fueron sometidos a 4 etapas de procesamiento: Wavelet, WICA, Reject y Muscle. Adicional se usaron las métricas de otra base de datos definido como PORTABLES, la cual contiene la potencia relativa por cada uno de los electrodos de interés junto con información demográfica de los sujetos, este archivo de métricas está en un formato tipo Feather.

C. Diseño de base de datos relacional

Para elegir el motor de base de datos apropiado también fue necesario basarse en ciertas características indispensables como lo eran: que posea versión gratuita y se pueda montar en local, esto debido a que en el grupo de investigación ya se está adelantando un proyecto para tener sus propios servidores y sería ideal poder montar en un futuro la base de datos en estos; adicional el motor debe permitir conexión con Python, esto es porque lo ideal es que el mismo código que transforma los datos, almacene estos en la base de datos. Debido a estas características se decidió utilizar MySQL.

Luego de concluir estos pasos, se pudo dar inicio a las siguientes etapas del proyecto.

V. RESULTADOS Y DISCUSIÓN

A. Transformación de BIDS a CSV.

Para esta parte del proyecto se decidió utilizar Python como lenguaje de programación. El código se puede dividir en 3 etapas principales: obtención de los archivos, transformación de los archivos y almacenamiento de los archivos transformados.

Para la etapa de la obtención de los archivos se utilizó las librerías OS y Pandas, OS para poder leer los archivos dentro de las carpetas del BIDS y Pandas porque tiene una función llamada *read_feather* que permite leer este tipo de archivos, esta función recibe como parámetro la ruta del archivo.

Para la segunda etapa del código se hace uso de la función de Pandas *to_csv*, Esta función se utiliza para transformar el archivo FEATHER a CSV, recibe como parámetro el nombre del archivo que quieres ponerle al nuevo archivo. En la fig. se puede ver la parte del código que hace estas primeras dos etapas.

Para la tercera etapa se hicieron dos cosas, la primera fue guardar el archivo CSV en local, mas adelante se explicará el motivo, y lo segundo que se hizo fue guardar el archivo transformado en la base de datos directamente. Para esto se hizo creó una conexión a la base de datos en local y se utilizaron las siguientes líneas de código para enviar el archivo a la base de datos y guardarla como una tabla:

- Comando de bash para instalar el conector de Python a Mysql:

```
pip install mysql-connector-python
```

- Código Python para generar la conexión:

```
import mysql.connector
import csv

# Configurar conexión a MySQL
conexion = mysql.connector.connect(
    host="localhost",
    user="root",
    password="tu_contraseña",
    database="mi_base_de_datos"
)

cursor = conexion.cursor()

# Ruta al archivo CSV
archivo_csv = "empleados.csv"

# Leer el archivo CSV y cargarlo en la tabla
with open(archivo_csv, mode='r', encoding='utf-8') as file:
    reader = csv.reader(file)
    next(reader) # Saltar el encabezado
    for fila in reader:
        cursor.execute(
            "INSERT INTO empleados (id, nombre, puesto, salario) VALUES (%s, %s, %s, %s)",
            fila
        )

# Confirmar cambios
conexion.commit()
print("Datos cargados exitosamente desde el archivo CSV.")

# Cerrar conexión
cursor.close()
conexion.close()
```

De acuerdo con los resultados obtenidos de esta transformación de datos, es posible convertir los archivos con formato propio de los datos de EEG a formato tabular, lo cual permite almacenarlas en bases de datos relacionales y utilizar herramientas de BI para analizar los datos de manera más simple y dinámica.

B. Montaje de la Base de Datos.

Para instalar MySQL en local se deben seguir los siguientes pasos:

- Ir al sitio oficial de MySQL y descargar el instalador, se recomienda MySQL Installer for Windows (web community), debido a que es más liviano.

- Ejecutar como administrador. Elige la opción Developer Default (incluye el servidor MySQL, Workbench, y herramientas adicionales).
- En la configuración de la base de datos seleccionar: Tipo de configuración: **Standalone MySQL Server**; Tipo de puerto: Por defecto es **3306**. Se recomienda no cambiarlo; modo de autenticación recomendado: **Use Strong Password Encryption for Authentication**. Crea una contraseña para el usuario root.
- La instalación finalizará y también se instalará MySQL Workbench y MySQL Shell.

Ahora que MySQL está instalado, es necesario primero crear un esquema dentro de esta donde estarán las tablas, para eso se debe seguir el siguiente paso a paso:

- o En la página principal de MySQL Workbench, seleccionar “Create a new Schema” que se encuentra en la parte superior de la pantalla.

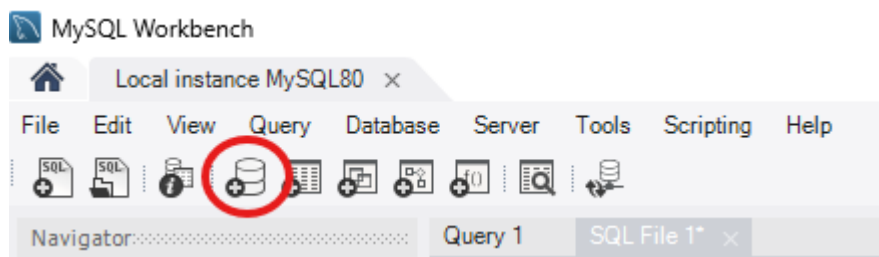


Fig. 1. Icono de New Schema.

- o Se nombra el esquema como se desee y le da “apply”.

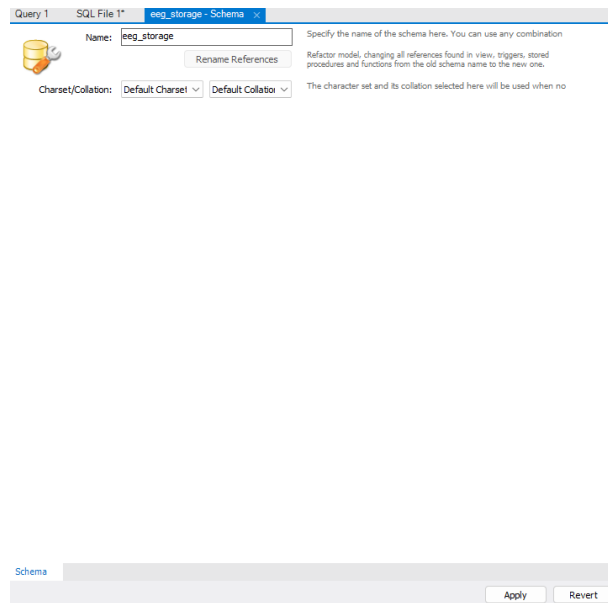


Fig. 2. Ventana emergente para crear un nuevo esquema.

Luego procedemos a crear una tabla vacía, este paso es importante ya que, si la tabla no existe previamente en la base de datos, no podremos cargar la información desde el código Python. Aquí tenemos un ejemplo en SQL de cómo crear una tabla vacía:

```
CREATE TABLE empleados (  
id INT AUTO_INCREMENT PRIMARY KEY,  
nombre VARCHAR(100),  
puesto VARCHAR(50),  
salario DECIMAL(10, 2)  
);
```

Nota: En la sección de anexos se agrega los queries utilizados para crear las tablas, no se agregan aquí por lo extensas que son.

Como se puede observar, para crear una tabla vacía es necesario definir el nombre de la tabla, el nombre de cada campo y que tipo de dato es.

- Desde la página oficial de Power BI: ingresar al siguiente link, seleccionar el idioma y dar clic en descargar: <https://www.microsoft.com/es-es/download/details.aspx?id=58494>. Seguir el paso a paso que da el instalador.

El motivo por el que es necesario descargar Power BI desktop y no se utiliza la versión online gratuita es debido a sus grandes limitantes a la hora de crear nuevas dimensiones y métricas. Una vez que se tiene instalado Power BI, el primer paso para crear un dashboard es cargar el modelo de datos, para esto se siguen los siguientes pasos:

- En la página inicial dar clic en “Informe en Blanco”.

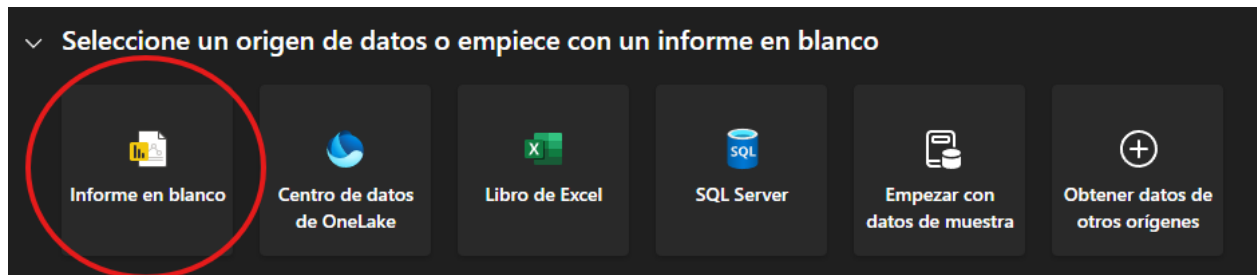


Fig. 4. Botón de Inicio para crear un nuevo dashboard.

- Una vez dentro del dashboard en blanco, en la barra de opciones buscar “Obtener Datos” y dar clic. Esta opción permite importar datos de varios orígenes de datos, como puede ser bases de datos, archivos, modelos semánticos de Power BI importados de otros dashboards. En este caso se importó los archivos CSV que se habían guardado en local en el paso del código, ya que al tener la base de datos en local no fue posible hacer la conexión con Power BI.

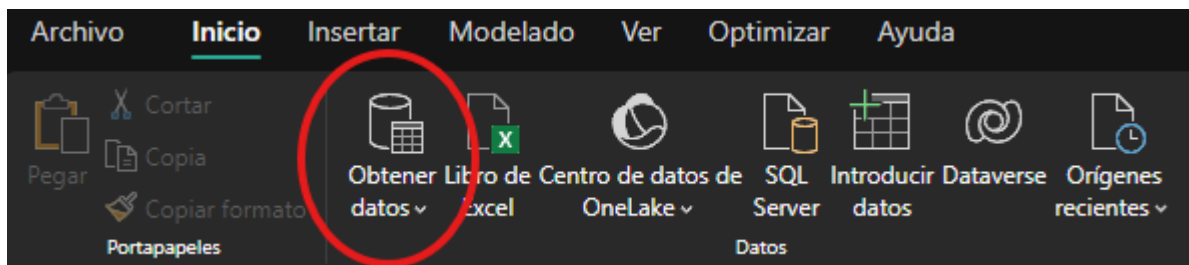


Fig. 5. Botón para seleccionar la fuente de datos.

- Seleccionar la opción “Texto o CSV”, Seleccionar el CSV deseado.
- Aparecerá una ventana emergente con una vista previa del modelo de datos, con opciones de editar el origen del archivo, delimitador de columnas y una detección de tipo de datos. Dar clic en “cargar”.

portables.csv

Origen de archivo: 1252: Europeo occidental (Windows) | Delimitador: Punto y coma | Detección del tipo de datos: Basado en las primeras 200 filas

subject	Task	group	O2_Delta	O2_Theta	O2_Alpha-1	O2_Alpha-2	O2_Beta1	O2_Beta2	O2_Beta3	O2_Gamma
sub-GG003	CE	GG	2,67162E+14	6,89095E+14	5,93775E+16	5,47911E+16	1,51892E+16	4,39388E+14	1,59319E+14	1,94609E+16
sub-GG004	CE	GG	1,45268E+15	3,564E+16	3,08277E+16	2,94323E+16	7,3511E+14	4,07177E+15	3,71332E+16	2,73271E+16
sub-GG009	CE	GG	1,76961E+15	6,36325E+15	1,22244E+16	1,67996E+16	1,66861E+16	2,80827E+16	1,21917E+16	1,52306E+16
sub-GG010	CE	GG0	2,33064E+16	8,97226E+13	7,04288E+15	9,3182E+15	1,90149E+16	4,82343E+15	1,47922E+16	1,27297E+15
sub-GG017	CE	GG	1,13833E+16	1,35303E+16	2,75579E+16	2,57683E+15	1,5094E+16	1,85155E+16	3,19651E+15	1,61816E+16
sub-GG018	CE	GG	2,09867E+16	2,28712E+16	2,35503E+16	1,26884E+16	1,10983E+16	2,35871E+16	3,83739E+15	2,60901E+16
sub-GG019	CE	GG	2,32468E+16	6,73889E+15	6,06597E+14	6,58629E+15	1,52224E+16	5,19666E+15	1,96691E+16	1,72739E+15
sub-GG023	CE	GG	2,022E+15	1,44951E+16	2,55319E+16	1,57087E+16	8,67932E+15	2,71296E+15	5,9739E+15	6,67813E+15
sub-GG029	CE	GG	2,26616E+16	1,13039E+16	2,34635E+16	2,23272E+15	1,27223E+16	1,74729E+15	3,39413E+15	2,38019E+15
sub-GG031	CE	GG	2,06448E+16	4,17895E+16	8,58425E+15	2,59235E+16	2,65164E+15	2,71411E+16	7,70621E+15	3,73173E+15
sub-GG032	CE	GG	3,33508E+16	4,38312E+16	4,73852E+15	5,67887E+15	1,37791E+15	4,07E+15	1,60232E+15	1,79765E+15
sub-GG033	CE	GG	1,93138E+16	1,6071E+16	2,14449E+15	1,27122E+16	1,32181E+15	2,99508E+16	6,72192E+15	7,52298E+15
sub-GG035	CE	GG	2,87994E+15	6,86248E+15	2,19506E+16	2,07882E+15	1,17689E+16	2,13376E+16	4,49106E+14	3,20552E+15
sub-GG038	CE	GG	3,02499E+15	7,76659E+15	1,19907E+16	1,79936E+15	1,96826E+16	2,03018E+16	6,42814E+15	3,85833E+14
sub-GG041	CE	GG	2,15467E+16	1,14236E+15	2,10595E+15	1,75899E+16	1,30822E+15	2,56676E+16	6,68635E+15	6,04492E+15
sub-GG042	CE	GG	3,79732E+15	1,19954E+16	8,06851E+15	5,41436E+16	9,97144E+15	3,55154E+15	1,21434E+16	1,08823E+16
sub-GG043	CE	GG	2,69418E+16	7,81766E+15	7,40948E+15	6,50508E+15	1,11302E+16	3,76468E+15	1,52213E+16	2,12097E+16
sub-GG044	CE	GG	2,85452E+16	1,24153E+16	2,19354E+16	1,57863E+16	1,0489E+16	2,73988E+16	4,56888E+16	3,52006E+15
sub-GG045	CE	GG	3,72625E+15	1,10029E+16	1,17993E+16	1,33976E+16	1,61288E+16	2,31258E+15	4,88284E+15	3,2135E+15
sub-GG047	CE	GG	1,53457E+15	8,43083E+15	2,01058E+16	2,19248E+16	1,24601E+15	2,3857E+16	8,95179E+15	1,03953E+15

Extraer tabla mediante ejemplos | Cargar | Transformar datos | Cancelar

Fig. 6. Vista previa del modelo de datos.

Una vez terminado este paso, ahora se tiene la capa semántica definida para este dashboard. Desde ya se puede empezar a crear las visualizaciones que tendrá este dashboard a partir de las columnas cargadas en la capa semántica. Sin embargo, en ocasiones es necesario crear una nueva columna o medida, para crear estas nuevas columnas o medidas se pueden seguir los siguientes pasos:

- En la parte superior de la pantalla dirigirse a la pestaña “Modelado”, dentro verás varias opciones de creación de Cálculos.

- Para crear una nueva medida se puede seleccionar entre “Nueva medida” y “Medida rápida”. La diferencia entre ambas es que “Nueva medida” requiere utilizar el lenguaje DAX, propio de Power BI para crear la medida, mientras que “Medida rápida” trae por defecto cálculos que evitan la necesidad de usar el lenguaje DAX. Debido a esto “Medida rápida” trae ciertas limitantes a la hora de crear nuevas métricas, pues si el cálculo que se desea hacer no está dentro de las opciones, será necesario usar DAX para crearla.

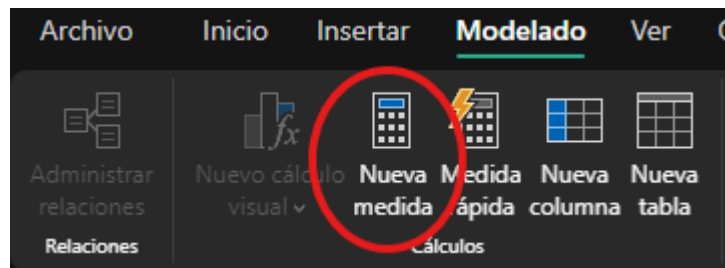


Fig. 7. Botones para crear nuevas medidas y columnas en el modelo de datos.

- Para el caso de este proyecto fue necesario crear una nueva métrica llamada “Total de Pacientes”. para esto se usó este comando escrito en lenguaje DAX:

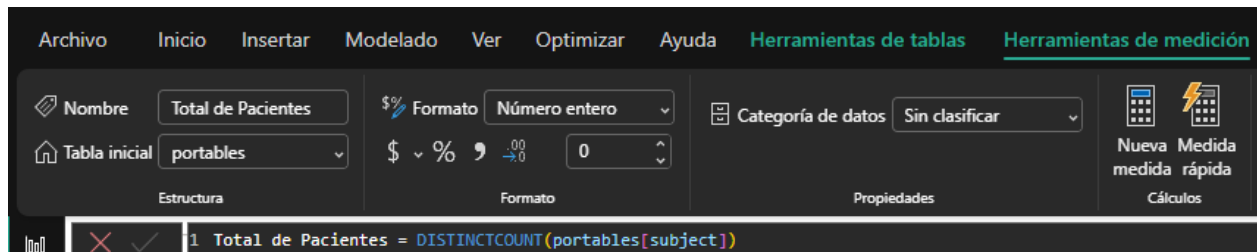


Fig. 8. Línea de código en DAX.

La función “DISTINCTCOUNT” regresa el conteo total de distintos valores dentro de una columna, “portables” es el nombre que tiene el modelo de datos dentro de Power BI y “subject” es el nombre de la columna a la que deseamos aplicar la función.

También se creó un nuevo archivo CSV, esto debido a que el archivo que el archivo original trae todas las medidas de los ritmos en pivote, lo que dificulta demasiado su uso en las visualizaciones. El nuevo archivo CSV posee tres columnas nuevas:

- Electrode, que contiene el nombre de todos los ritmos.
- Electrode-position, que posee las ubicaciones de los electrodos.
- Value, que contiene el valor de las potencias tomadas en cada uno de los electrodos por sujeto.

subject	Task	group	sex	age	Electrode-Position	Electrode	value
sub-GG023	CE	GG	F	25	O2	Delta	0,2
sub-GG023	CE	GG	F	25	O2	Theta	0,14
sub-GG023	CE	GG	F	25	O2	Alpha-1	0,26
sub-GG023	CE	GG	F	25	O2	Alpha-2	0,16
sub-GG023	CE	GG	F	25	O2	Beta1	0,09
sub-GG023	CE	GG	F	25	O2	Beta2	0,03
sub-GG023	CE	GG	F	25	O2	Beta3	0,06
sub-GG023	CE	GG	F	25	O2	Gamma	0,07
sub-GG023	CE	GG	F	25	C4	Delta	0,3
sub-GG023	CE	GG	F	25	C4	Theta	0,16
sub-GG023	CE	GG	F	25	C4	Alpha-1	0,19
sub-GG023	CE	GG	F	25	C4	Alpha-2	0,12
sub-GG023	CE	GG	F	25	C4	Beta1	0,1
sub-GG023	CE	GG	F	25	C4	Beta2	0,03
sub-GG023	CE	GG	F	25	C4	Beta3	0,06
sub-GG023	CE	GG	F	25	C4	Gamma	0,04
sub-GG023	CE	GG	F	25	O1	Delta	0,29
sub-GG023	CE	GG	F	25	O1	Theta	0,13
sub-GG023	CE	GG	F	25	O1	Alpha-1	0,17
sub-GG023	CE	GG	F	25	O1	Alpha-2	0,11
sub-GG023	CE	GG	F	25	O1	Beta1	0,11
sub-GG023	CE	GG	F	25	O1	Beta2	0,03

Fig. 9. Nueva tabla despivoteada.

Este archivo CSV fue creado por medio de Python, usando la función “melt” de la librería Pandas. A continuación, se muestra la línea de código encargada de despivotear las columnas:

```
[ ] # Despivotar todas las columnas excepto 'subject', 'Task', 'group'
df_melted = pd.melt(df, id_vars=['subject', 'Task', 'group', 'sex', 'age'], var_name='Electrode', value_name='value')

df_melted['Electrode_position'] = df['Electrode'].str[:2]
df_melted['Electrode'] = df['Electrode'].str.split("_").str[1]

df_melted.to_csv('portables_unpivot.csv', index=False)
files.download('portables_unpivot.csv')
```

- df: archivo csv previamente cargado.
- Id_vars: Las columnas que se mantendrán como en el archivo original, el despivotado se hará agrupando los valores en estas columnas.
- var_name: Esta nueva columna se llenará con los nombres de los campos de las antiguas columnas que queremos despivotar. El string “Electrode” es el nombre que recibirá este nuevo campo.
- Value_name: En esta nueva columna se almacenarán los valores que tenían las columnas que ahora hacen parte del campo “var_name”. El string “Value” es el nombre que recibirá este nuevo campo.
- La segunda línea de código crea una nueva columna llamada “Electrode_Position” usando los dos primeros caracteres de la columna Electrode.
- La tercer línea de Código elimina todo lo que haya antes de “_”.

Gracias a este nuevo archivo CSV sin pivote, se puede lograr las visualizaciones deseadas.

Ahora, se presenta el dashboard final el cual consta de 2 páginas.

La primera presenta un resumen de los datos, en este caso específicamente de la distribución de los pacientes por sexo, grupo y edad. Esta primera vista de los datos es fundamental para analizar e interpretar los datos, pues el valor de las potencias puede variar entre sujetos debido a estas distribuciones.

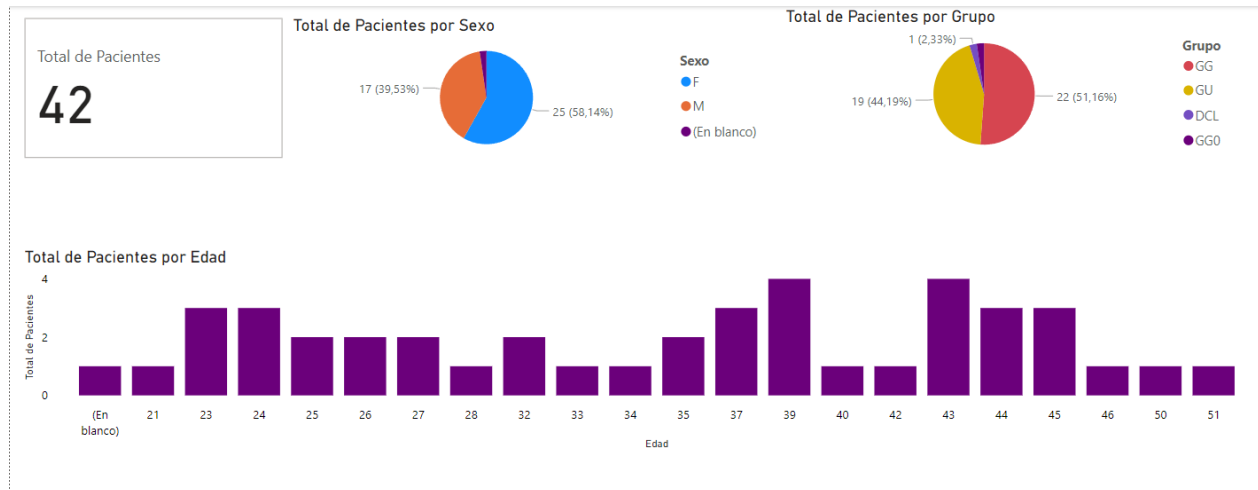


Fig. 10. Página #1 del Dashboard.

Para esta primera página se usó la medida creada previamente “Total de pacientes” en todas las visualizaciones.

Esta primer página no posee filtros, pues la idea de estas visualizaciones es que den un recuento total de los datos, que en todo momento muestren la cantidad de sujetos que se tienen.

La segunda página presenta unas gráficas de gatos y bigotes, en el que se puede ver las potencias por ritmos, este tipo de gráfica fue seleccionada debido a que permite ver en una sola gráfica varias medidas estadísticas, necesarias para el análisis, para esta segunda página se hizo uso del CSV despivoteado creado previamente.

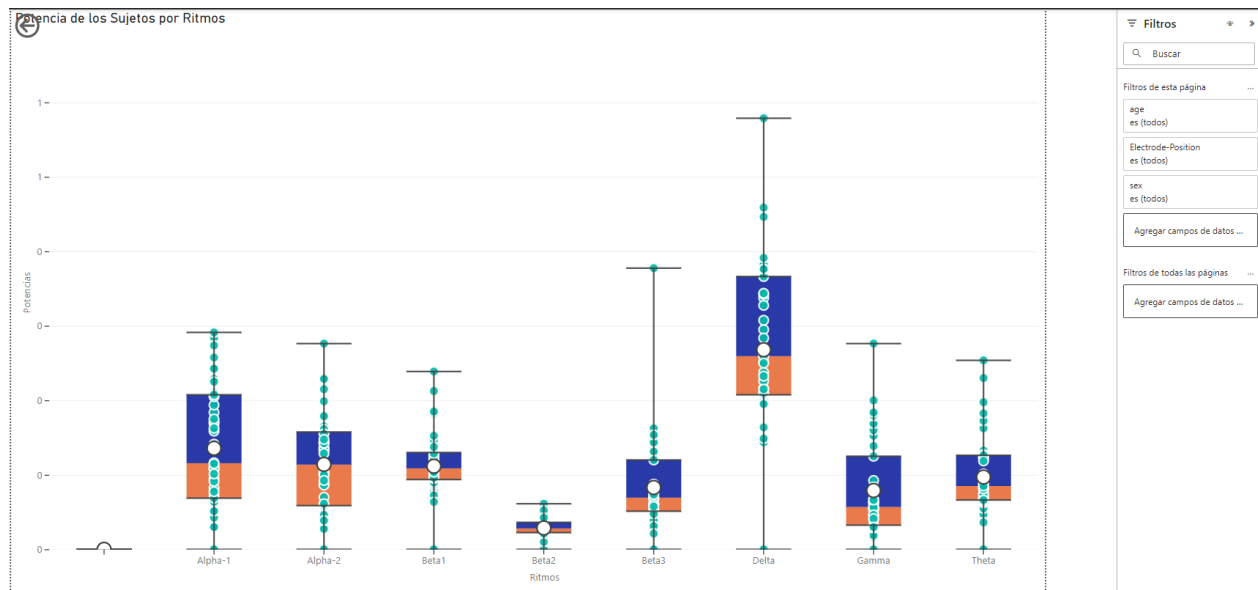


Fig. 11. Página #2 del Dashboard.

Por defecto Power BI no trae instalada la visualización de cajas y bigotes, sin embargo, es posible instalarla de forma gratuita desde el Marketplace de Power BI, siguiendo los siguientes pasos:

- En la sección de “Visualizaciones” buscar los tres puntos seguidos (...) esta se encuentra al final de todas las visualizaciones.

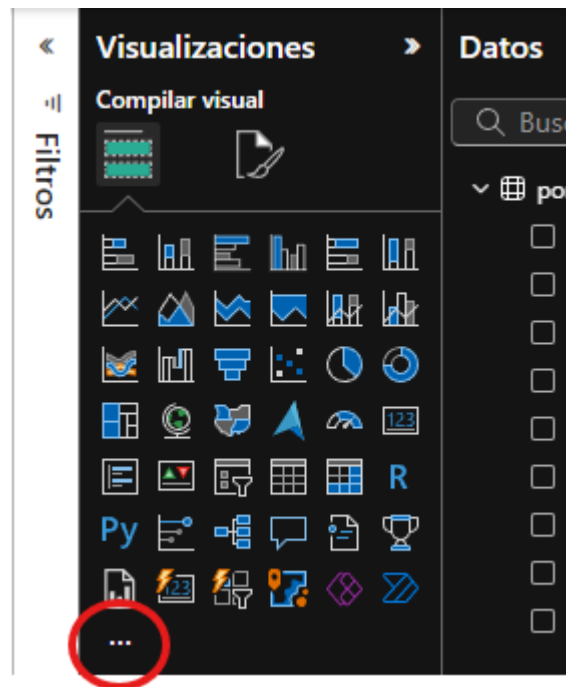


Fig. 12. Sección de objetos visuales en Power BI.

- Dar clic en (...) y seleccionar “Obtener más objetos visuales” se abrirá una ventana emergente con una gran cantidad de objetos visuales.

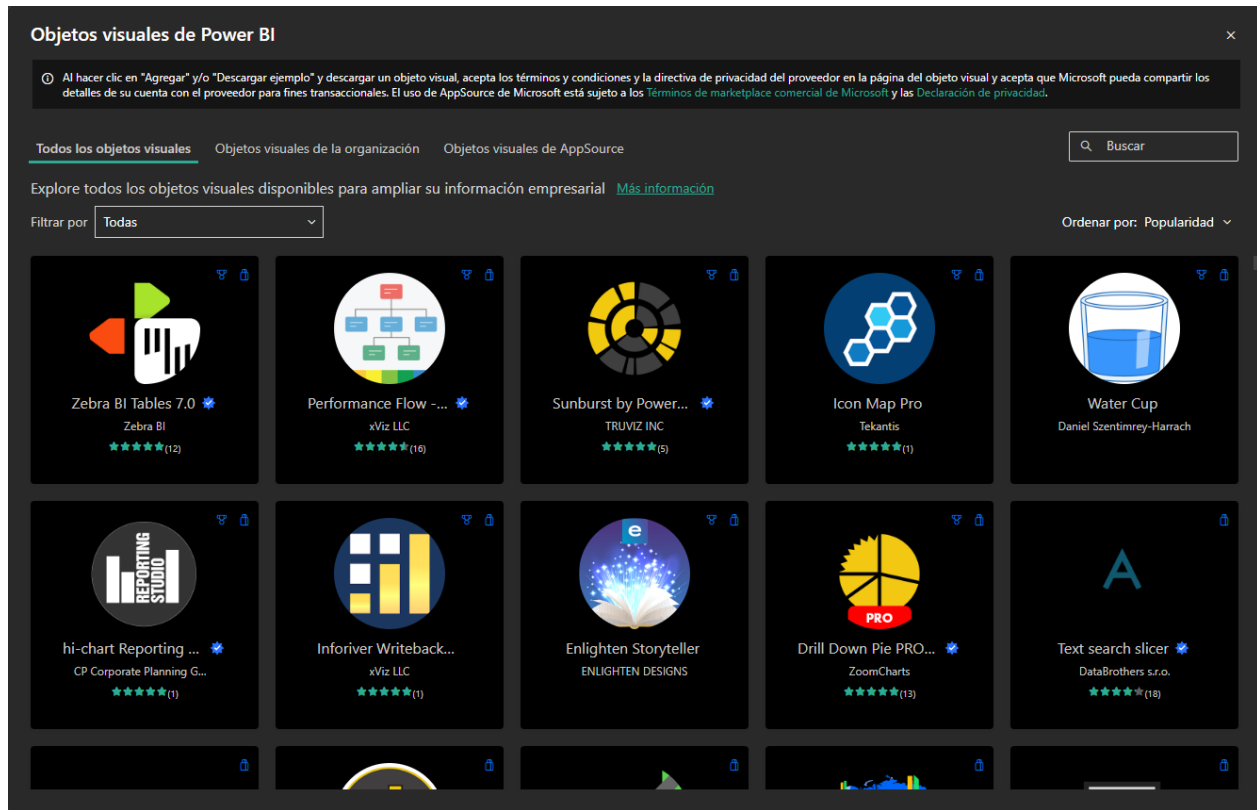


Fig. 13. Marketplace de objetos visuales.

- Buscar “Box and Whisker chart” y darle clic en agregar, una vez hecho esto ya se tiene disponible este objeto visual dentro de las opciones.



Fig. 14. Gráfico de Cajas y Bigotes.

Nota: También es posible instalar una visualización personalizada, cargando un archivo que se tenga previamente.

Para esta visualización lo que deseamos es ver los ritmos los valores de potencia de los sujetos por Ritmos. Para esto en la visualización se genera la siguiente configuración:

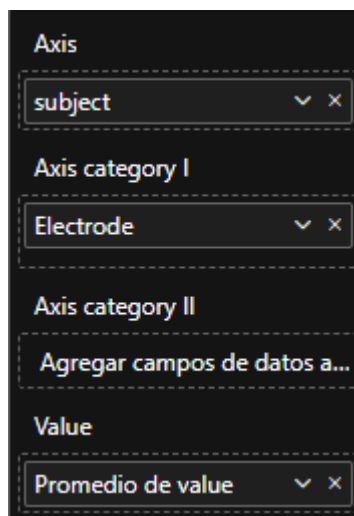


Fig. 15. Editor de la visualización de cajas y bigotes.

En el Axis se selecciona la columna sujeto, pues la intención es que las distintas variables estadísticas se calculen en base a los sujetos; en el Axis Category I se selecciona la columna que queremos que se use para separar los datos en distintas cajas y bigotes, en este caso se utiliza Electrode que son los ritmos; por último en Value se selecciona la medida que será usada para hacer los cálculos que hará internamente la visualización, aquí se puede seleccionar que tipo de agrupación tendrá la medida, puede ser suma, promedio, mediana, etc. En nuestro caso la agrupación que mas sentido tiene es el promedio, ya que cuando en los filtros se tienen seleccionados todas las posiciones de los electrodos, se agrupan varios ritmos, por lo que tiene mas sentido mostrar el promedio de todos estos ritmos.

Y esto nos lleva los filtros que tiene esta página, el primero y más importante es el de la posición de los electrodos, en este filtro se pueden seleccionar valores como C3, C4, FP1, etc. Este filtro es de suma importancia pues el análisis de estos datos tiene sentido al comparar los ritmos por segmentos, este filtro nos evita tener que crear la misma visualización por cada segmento existente. Los otros filtros que se tienen disponibles son el de la edad y el sexo de los sujetos, estos filtros nos permiten segmentar los sujetos y hacer análisis en base a esto.

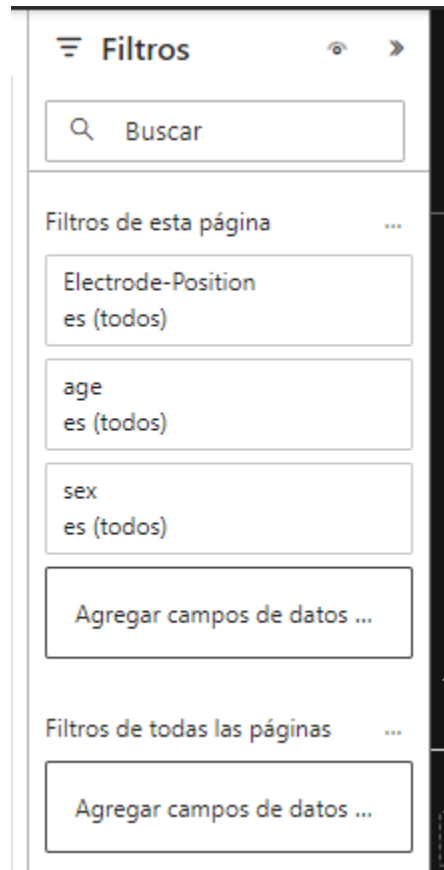


Fig. 16. Sección de filtros.

Luego de haber creado el dashboard por completo, el último paso que queda es publicarlo en nuestra cuenta de Power BI en la nube, este paso nos permite enviarle el dashboard a quien nosotros deseemos (solo en caso de tener la versión PRO), para esto se selecciona la opción “Publicar” en la parte superior de la pantalla.

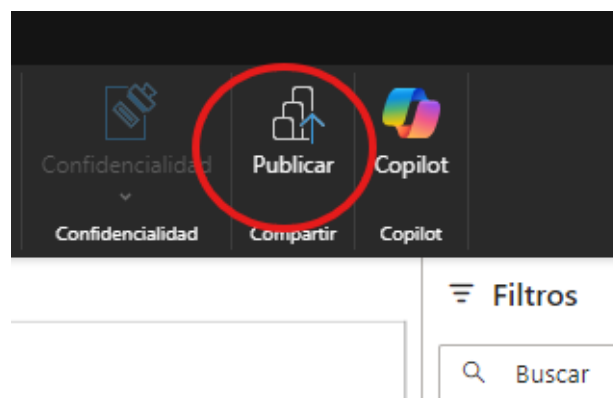


Fig. 17. Botón para publicar el dashboard.

El resultado de esta última etapa del proyecto define el uso de Power BI perfectamente viable, debido a sus muchas cualidades como lo son:

- Facilidad al crear el modelo de datos debido a la gran capacidad que tiene para cargar los datos ya sea desde archivos o desde bases de datos.
- Flexibilidad a la hora de crear nuevas medidas.
- Corta curva de aprendizaje, permitiendo que los usuarios nuevos y no técnicos puedan aprender la herramienta en un corto periodo de tiempo.
- Gran variedad de objetos visuales, permitiendo personalizar el dashboard a completo gusto y necesidad del usuario.

Todas estas cualidades hacen que Power BI sea una herramienta útil para el grupo de investigación, el cual puede ser usado tanto por usuarios técnicos como lo serían los ingenieros como los no técnicos como lo serían los médicos.

La única limitante que ha presentado hasta el momento es la conexión a una base de datos local, lo que hace que la etapa 2 de este proyecto pierda parcialmente su función, pues queda sirviendo únicamente para almacenar los datos en formato tabular que, si bien es útil, puesto que es necesario centralizar el almacenamiento de los datos en un solo lugar, esto es algo que se podría hacer perfectamente en un lugar como Google drive o similares. Es por este motivo que considero que esta limitante que presenta Power BI puede dejar medianamente obsoleta a la base de datos relacional; hay que tener en cuenta que esto cambiaría una vez se tenga la base de datos en un servidor al que se pueda tener acceso desde Power BI, pues en ese hipotético, si podría conectarse a la base de datos y extraer los datos directamente desde ahí.

Se revisó un proyecto de grados en el cual el autor (Zapata) propone como método de visualización de datos de EEG utilizar Python y mostrarlos por medio de un aplicativo web. Al comparar los resultados de este proyecto con los resultados de Zapata [9] se puede notar como principal diferencia la cantidad de gráficas que se generan para presentar los análisis ya sea de forma global, por banda de frecuencia, por electrodos, etc. Al comparar el análisis por electrodos, que es el equivalente al que se hizo en el dashboard para este proyecto, se puede ver que Zapata necesito alrededor de 8 gráficas para poder visualizar todas las opciones, mientras que en el caso del dashboard con solo una gráfica y un filtro por electrodos se puede obtener los mismos análisis. Esto demuestra que esta propuesta no solo permite generar gráficas de forma más rápida que

usando Python, sino que además permite realizar múltiples análisis con solo una gráfica, seleccionando los filtros correctamente. Esto nos permite concluir que el uso de Power BI es una muy buena opción para el análisis de datos de EEG mejorando incluso algunos pasos que se realizan de forma convencional usando Python.

D. Encuesta de Percepción.

Por último, se presentó el proyecto a los integrantes del grupo de investigación. Se hizo un tutorial mostrando el flujo de trabajo y la forma en la que se creaban los dashboards, también presentando los beneficios de utilizar una herramienta de BI para el análisis de los datos. Se les entregó una encuesta en Google forms con la intención de conocer sus opiniones frente a esta propuesta y estos fueron los resultados.

TABLA II
RESULTADOS DE LA ENCUESTA DE PERCEPCIÓN

Preguntas	1	2	3	4	5
El código en Python presentado para la transformación de los datos es fácil de entender y mantener	0%	0%	0%	0%	100%
La propuesta de guardar los archivos CSV en una base de datos relacional me parece coherente.	0%	0%	0%	25%	75%
Considero que la herramienta Power BI es fácil de aprender y utilizar.	0%	0%	0%	0%	100%
El flujo de trabajo es fácil de entender y replicar	0%	0%	0%	75%	25%
El uso de este flujo de trabajo puede facilitar el análisis de los datos.	0%	0%	0%	25%	75%
Este flujo de trabajo es pertinente para el tipo de trabajo que se realiza dentro del grupo de investigación.	0%	0%	0%	75%	25%
Me interesaría utilizar este flujo de trabajo en el futuro.	0%	0%	25%	0%	75%

Estos resultados reflejan la aceptación por parte de los integrantes del grupo de investigación, que consideran viable implementar esta metodología propuesta en el futuro, ya que en general puede ayudar a mejorar el análisis de los datos, gracias a los filtros que se pueden agregar al dashboard generan un reporte dinámico, pues pueden filtrar los datos por un segmento cerebral, por sexo o edad de los sujetos, etc. Esto también permite que el reporte sea visualmente mas agradable pues en lugar de tener que hacer 8 o 10 graficas iguales que permitan analizar los datos desde distintos puntos de vista, puede lograrse lo mismo con un solo gráfico.

XI. CONCLUSIONES

El proyecto logró cumplir los objetivos planteados al proporcionar una metodología eficiente y escalable para la visualización de datos en neurociencia computacional. Esto incluyó la transformación de formatos complejos como BIDS a tabulares, el diseño de una base de datos relacional en MySQL y la implementación de dashboards interactivos.

Power BI en este contexto resulta ser la herramienta de BI idónea debido a sus características tales como generación de dashboards dinámicos, su ambiente altamente amigable para usuarios no técnicos, posibilidad de agregar mas objetos visuales a parte de los básicos gracias a su Marketplace interno y su capacidad de filtrar los datos en tiempo real, permitiendo un análisis mas potente. Todas estas características le dieron su victoria frente a Grafana, que en este caso era su competencia directa para esta selección.

La integración de herramientas gratuitas y accesibles, como Power BI Desktop y Python, aseguró que la metodología pueda ser replicada y escalada por el grupo de investigación sin costos adicionales significativos.

Los dashboards desarrollados ofrecen una herramienta poderosa para explorar datos de EEG, permitiendo análisis profundos y decisiones basadas en evidencia utilizando una herramienta de fácil aprendizaje, lo que permite que personas no técnicas tomen más acción en dichos análisis y tomas de decisiones. Esto marca un avance significativo en cómo los datos complejos pueden traducirse en información utilizable para la neurociencia computacional.

Aunque la metodología es efectiva para el volumen actual de datos, el crecimiento futuro puede requerir la migración a soluciones más robustas, como data warehouses. Además, sería útil integrar el código desarrollado durante este proyecto al código que genera los datos BIDS, de forma que este proceso sea mucho más ágil.

El uso de Power BI facilita en gran medida el análisis de los datos, esto debido a sus grandes ventajas como fácil aprendizaje, corto tiempo de desarrollo de las gráficas y su filtrado en tiempo real, posibilitando hacer diferentes análisis con una sola gráfica.

REFERENCIAS

- [1] Gianluca, Susi. (s.f.). Neurociencia computacional. <https://meg.ucm.es/>.
- [2] Alexiou, A., Ashraf, G. M., Chellappan, D. K., Dua, K., Dureja, H., Gupta, G., Jha, N. K., Jha, S. K., Khan, F. A., Kesari, K. K., Kumar, N., Ojha, S., Prasanna, P., Roychoudhury, S., Singh, S. K., Upadhye, V., Usman, M. B., & Zaccani, F. (2022). Biological databases and tools for neurological disorders. *Journal of Integrative Neuroscience*, 21(1), 41. <https://doi.org/10.31083/j.jin2101041>
- [3] Elena A. Allen, Erik B. Erhardt, Vince D. Calhoun. (2012). Data Visualization in the Neurosciences: Overcoming the Curse of Dimensionality. <https://doi.org/10.1016/j.neuron.2012.05.001>
- [4] BIMCV. estructura de datos de imágenes médicas ampliada a múltiples modalidades y regiones anatómicas. [https://bimcv.cipf.es/bimcv-projects/mids/#:~:text=Brain%20Imaging%20Data%20Structure%20\(BIDS,un%20potente%20est%C3%A1ndar%20de%20gesti%C3%B3n](https://bimcv.cipf.es/bimcv-projects/mids/#:~:text=Brain%20Imaging%20Data%20Structure%20(BIDS,un%20potente%20est%C3%A1ndar%20de%20gesti%C3%B3n)
- [5] Oracle Colombia. [s.f.]. ¿Qué es un almacén de datos?. <https://www.oracle.com/autonomous-database/modern-data-warehouse/>.
- [6] Tableau from salesforce. (s.f.). ¿Qué es Business Intelligence o inteligencia de negocios?. <https://www.tableau.com/data-insights/business-intelligence>.
- [7] Microsoft Learn. (22/03/2024). ¿Qué es Power BI?. <https://learn.microsoft.com/en-us/training/powerplatform/power-bi>.
- [8] Jerson Martínez. (27-12-2021). ¿Qué es Grafana y primeros pasos?. <https://openwebinars.net/cursos/monitorizacion-grafana-prometheus/>.
- [9] Luisa Z. (2022). Desarrollo de aplicación de servicios web basado en estándares de informática médica para el preprocesamiento y visualización de registros EEG. <https://hdl.handle.net/10495/30073>

ANEXOS

A. Código Python.

Convertir los archivos .feather a .csv:

```
import pandas as pd
from google.colab import files
from google.colab import drive
import os
drive.mount('/content/drive')

# Leer el archivo Feather
df = pd.read_feather('/content/power_PORTABLES_CE_columns.feather')

# Guardar como CSV
df = df.to_csv('portables.csv', index=False)
files.download('portables.csv')

# Despivotar todas las columnas excepto 'subject', 'Task', 'group'
df_melted = pd.melt(df, id_vars=['subject', 'Task', 'group', 'sex',
'age'], var_name='Electrode', value_name='value')

# Separar en dos columnas
df_melted['Electrode_position'] = df['Electrode'].str[:2]
df_melted['Electrode'] = df['Electrode'].str.split("_").str[1]

# Convertir a csv el nuevo archivo y descargar
df_melted.to_csv('portables_unpivot.csv', index=False)
files.download('portables_unpivot.csv')
```

Enviar los archivos a MySQL:

```
import mysql.connector
import csv

# Configurar conexión a MySQL
conexion = mysql.connector.connect(
    host="localhost",
    user="root",
```



```
password="1020488779.Jr",
database="mi_base_de_datos"
)

cursor = conexion.cursor()

# Ruta al archivo CSV
archivo_csv = "portables.csv"

# Leer el archivo CSV y cargarlo en la tabla
with open(archivo_csv, mode='r', encoding='utf-8') as file:
    reader = csv.reader(file)
    next(reader) # Saltar el encabezado
    for fila in reader:
        cursor.execute( # Se genera la query para insertar los nuevos
valores a la tabla creada en la DB
            "INSERT INTO portables (Task, group, O2_Delta, O2_Theta,
O2_Alpha-1, O2_Alpha-2, O2_Beta1, O2_Beta2, O2_Beta3, O2_Gamma, C4_Delta,
C4_Theta, C4_Alpha-1 C4_Alpha-
2 C4_Beta1 C4_Beta2 C4_Beta3 C4_Gamma O1_Delta O1_Theta O1_Alpha-
1 O1_Alpha-2 O1_Beta1 O1_Beta2 O1_Beta3 O1_Gamma FP1_Delta FP1_Theta
FP1_Alpha-1 FP1_Alpha-2 FP1_Beta1 FP1_Beta2 FP1_Beta3 FP1_Gamma
P8_Delta P8_Theta P8_Alpha-1 P8_Alpha-
2 P8_Beta1 P8_Beta2 P8_Beta3 P8_Gamma P7_Delta P7_Theta P7_Alpha-
1 P7_Alpha-2 P7_Beta1 P7_Beta2 P7_Beta3 P7_Gamma FP2_Delta FP2_Theta
FP2_Alpha-1 FP2_Alpha-2 FP2_Beta1 FP2_Beta2 FP2_Beta3 FP2_Gamma
C3_Delta C3_Theta C3_Alpha-1 C3_Alpha-
2 C3_Beta1 C3_Beta2 C3_Beta3 C3_Gamma center sex age) VALUES (%s,
%s, %s, %s)",
            fila
        )

# Confirmar cambios
conexion.commit()
print("Datos cargados exitosamente desde el archivo CSV.")

# Cerrar conexión
cursor.close()
conexion.close()
```

Buscar cualquier archivo dentro de la carpeta BIDS:

```
# Función para buscar en la carpeta seleccionada el archivo deseado, esto
con el fin de buscar dentro de las carpetas BIDS
def buscar_archivo(ruta_inicial, nombre_archivo):
    archivos_encontrados = []

    for ruta_actual, subcarpetas, archivos in os.walk(ruta_inicial):
        for archivo in archivos:
            if archivo.lower() == nombre_archivo.lower():
                archivos_encontrados.append(os.path.join(ruta_actual,
archivos))

    return archivos_encontrados
```

B. Código SQL.

Crear la tabla vacía “Portables”:

```
CREATE TABLE portables (
    subject VARCHAR(255),    -- Cadena de texto
    Task VARCHAR(255),      -- Cadena de texto
    `group` VARCHAR(255),   -- "group" es una palabra reservada en MySQL, por eso se
usan comillas invertidas
    O2_Delta FLOAT,        -- Número con decimales
    O2_Theta FLOAT,        -- Número con decimales
    O2_Alpha_1 FLOAT,      -- Número con decimales
    O2_Alpha_2 FLOAT,      -- Número con decimales
    O2_Beta1 FLOAT,        -- Número con decimales
    O2_Beta2 FLOAT,        -- Número con decimales
    O2_Beta3 FLOAT,        -- Número con decimales
    O2_Gamma FLOAT,        -- Número con decimales
    C4_Delta FLOAT,
    C4_Theta FLOAT,
    C4_Alpha_1 FLOAT,
    C4_Alpha_2 FLOAT,
```

C4_Beta1 FLOAT,
C4_Beta2 FLOAT,
C4_Beta3 FLOAT,
C4_Gamma FLOAT,
O1_Delta FLOAT,
O1_Theta FLOAT,
O1_Alpha_1 FLOAT,
O1_Alpha_2 FLOAT,
O1_Beta1 FLOAT,
O1_Beta2 FLOAT,
O1_Beta3 FLOAT,
O1_Gamma FLOAT,
FP1_Delta FLOAT,
FP1_Theta FLOAT,
FP1_Alpha_1 FLOAT,
FP1_Alpha_2 FLOAT,
FP1_Beta1 FLOAT,
FP1_Beta2 FLOAT,
FP1_Beta3 FLOAT,
FP1_Gamma FLOAT,
P8_Delta FLOAT,
P8_Theta FLOAT,
P8_Alpha_1 FLOAT,
P8_Alpha_2 FLOAT,
P8_Beta1 FLOAT,
P8_Beta2 FLOAT,
P8_Beta3 FLOAT,
P8_Gamma FLOAT,
P7_Delta FLOAT,
P7_Theta FLOAT,
P7_Alpha_1 FLOAT,

```
P7_Alpha_2 FLOAT,  
P7_Beta1 FLOAT,  
P7_Beta2 FLOAT,  
P7_Beta3 FLOAT,  
P7_Gamma FLOAT,  
FP2_Delta FLOAT,  
FP2_Theta FLOAT,  
FP2_Alpha_1 FLOAT,  
FP2_Alpha_2 FLOAT,  
FP2_Beta1 FLOAT,  
FP2_Beta2 FLOAT,  
FP2_Beta3 FLOAT,  
FP2_Gamma FLOAT,  
C3_Delta FLOAT,  
C3_Theta FLOAT,  
C3_Alpha_1 FLOAT,  
C3_Alpha_2 FLOAT,  
C3_Beta1 FLOAT,  
C3_Beta2 FLOAT,  
C3_Beta3 FLOAT,  
C3_Gamma FLOAT,  
center VARCHAR(255),    -- Cadena de texto  
sex VARCHAR(255),      -- Cadena de texto  
age INT                 -- Número entero  
);
```

Crear la tabla vacía “portable_unpivot”:

```
CREATE TABLE portables (  
    subject VARCHAR(255),    -- Cadena de texto  
    Task VARCHAR(255),      -- Cadena de texto
```

```
`group` VARCHAR(255),    -- "group" es una palabra reservada en MySQL, por eso se  
usan comillas invertidas
```

```
sex VARCHAR(255),      -- Cadena de texto
```

```
age INT,               -- Número entero
```

```
Electrode-Position VARCHAR(255),
```

```
Electrode VARCHAR(255),
```

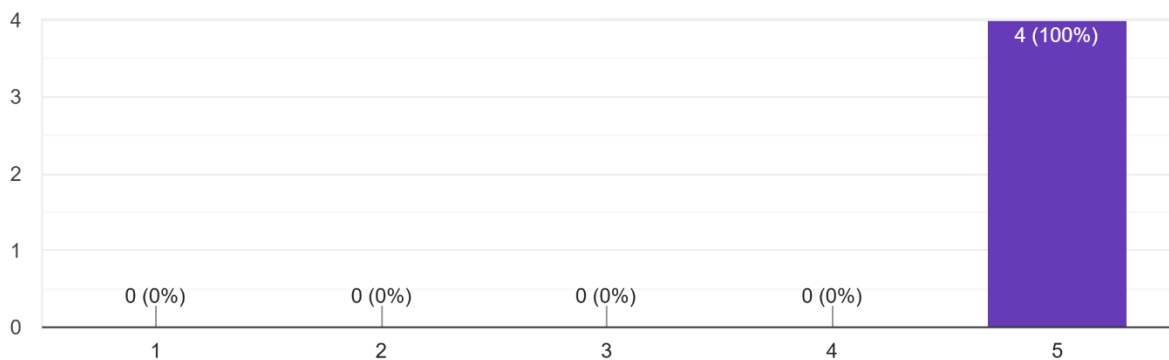
```
Value FLOAT
```

```
);
```

C. Resultados encuesta: “Percepción del flujo de trabajo en Power BI”

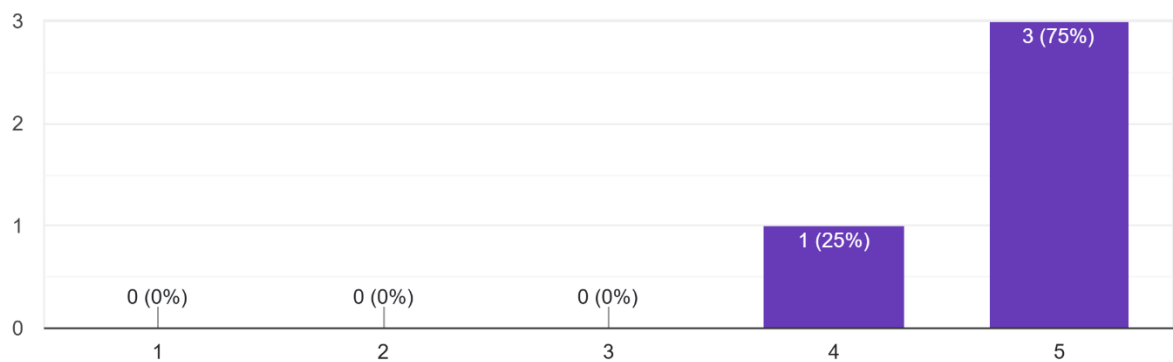
El código en Python presentado para la transformación de los datos es fácil de entender y mantener

4 respuestas



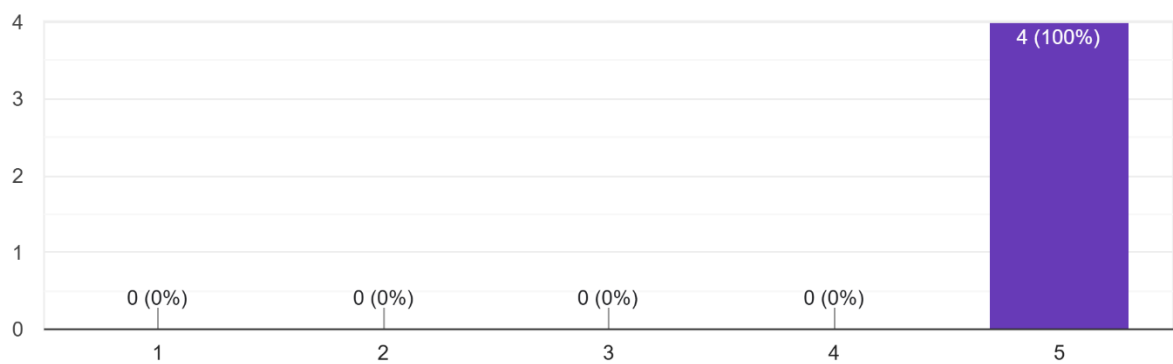
La propuesta de guardar los archivos CSV en una base de datos relacional me parece coherente.

4 respuestas



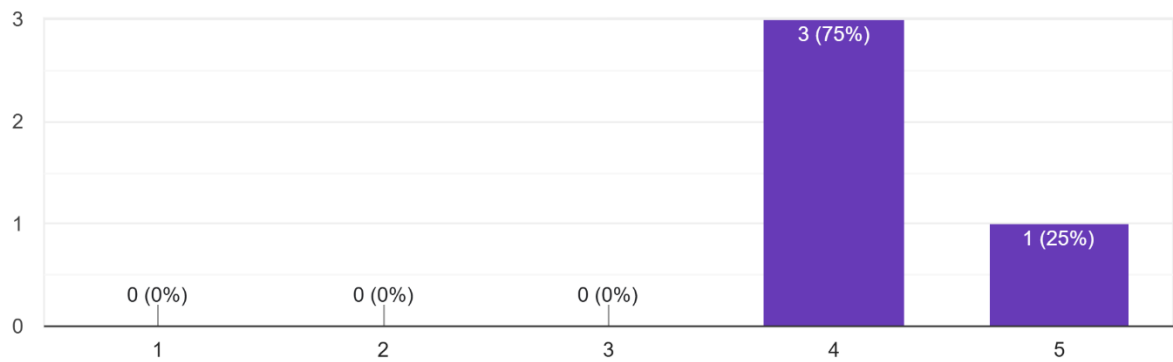
Considero que la herramienta Power BI es fácil de aprender y utilizar.

4 respuestas



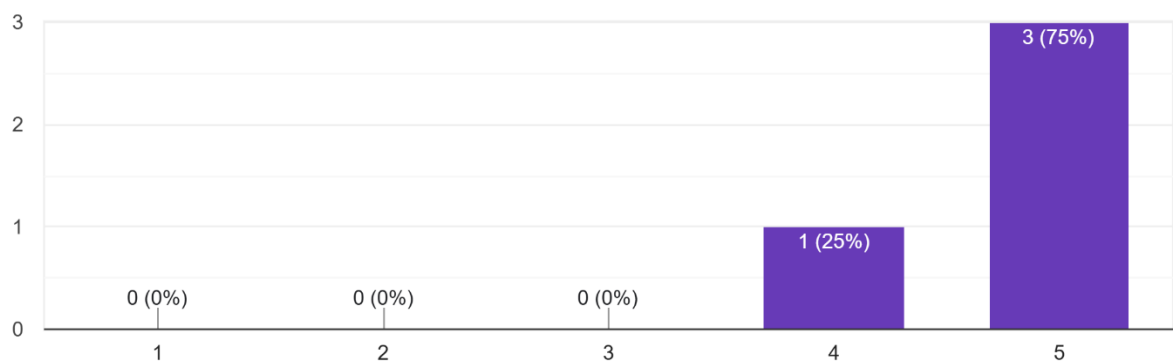
El flujo de trabajo es fácil de entender y replicar

4 respuestas



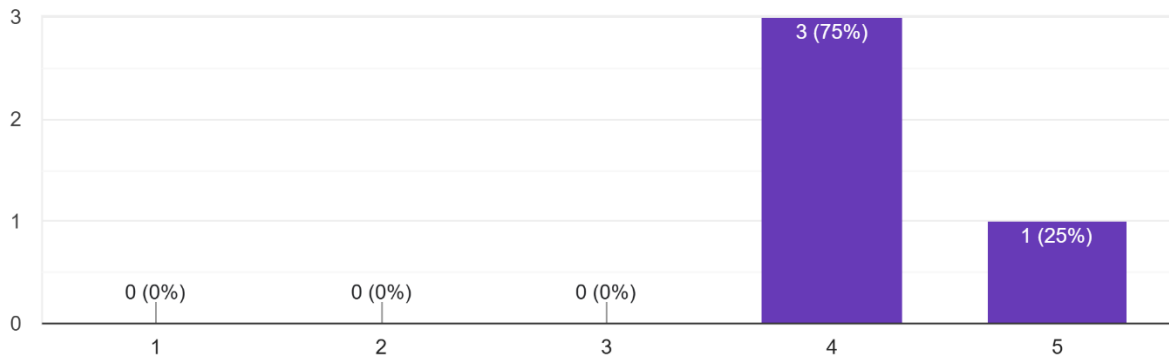
El uso de este flujo de trabajo puede facilitar el análisis de los datos.

4 respuestas



Este flujo de trabajo es pertinente para el tipo de trabajo que se realiza dentro del grupo de investigación.

4 respuestas



Me interesaría utilizar este flujo de trabajo en el futuro.

4 respuestas

