



**Desarrollo de aplicación web para el análisis de datos metabolómicos para el grupo de neurociencias UdeA**

Juan Camilo Morales Vanegas

Informe de práctica presentado para optar al título de Ingeniero de Sistemas

Asesor

Julián David Arias Londoño, PhD

Universidad de Antioquia  
Facultad de Ingeniería  
Ingeniería de Sistemas  
Medellín, Antioquia, Colombia  
2023

Cita	Morales Vanegas [1]
<b>Referencia</b> Estilo IEEE (2020)	[1] J. Morales Vanegas, “Desarrollo de aplicación web para el análisis de datos metabolómicos para el grupo de neurociencias UdeA”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2023.



Centro de Documentación Ingeniería (CENDOI)

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

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

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 .....	6
ABSTRACT .....	7
I. INTRODUCCIÓN .....	8
II. OBJETIVOS .....	10
A. Objetivo general .....	10
B. Objetivos específicos .....	10
III. METODOLOGÍA .....	11
A. Primera fase (Revisión, planeación y diseño): .....	11
B. Segunda Fase(Desarrollo del proyecto).....	13
1) Operaciones CRUD .....	14
2) Nueva interfaz de usuario .....	15
3) Análisis de datos .....	15
4) Visualizaciones de datos.....	16
IV. RESULTADOS.....	18
V. CONCLUSIONES.....	23
VI. RECOMENDACIONES.....	26
REFERENCIAS .....	27

## LISTA DE FIGURAS

Fig. 1. Cronograma de actividades del desarrollo del proyecto .....	13
Fig. 2. Arquitectura cliente-servidor propuesta para el proyecto .....	13
Fig. 3. Pantallazo de la app después de ejecutar un análisis multivariado .....	21
Fig. 4. Vista dónde se listan todos los proyectos creados por el usuario .....	22
Fig. 5. Vista para crear un nuevo proyecto .....	22
Fig. 6. Vista del prototipo actual .....	25
Fig. 7. Vista del nueva versión web .....	25

---

## SIGLAS, ACRÓNIMOS Y ABREVIATURAS

<b>App</b>	Aplicación
<b>UX</b>	User Experience
<b>UI</b>	User Interface
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>UdeA</b>	Universidad de Antioquia
<b>GNA</b>	Grupo de neurociencias de la Universidad de Antioquia
<b>CSV</b>	Comma-separated Values
<b>DRF</b>	Django REST Framework
<b>JWT</b>	JSON Web Tokens
<b>SVG</b>	Scalable Vector Graphics
<b>PCA</b>	Parcical Component Analysis
<b>CRUD</b>	Create Read Update Delete
<b>ORM</b>	Object Relational Mapping

---

## RESUMEN

Este trabajo aborda la migración de la aplicación usada actualmente por el grupo de neurociencias de la Universidad de Antioquia para el análisis de datos metabólicos. Este prototipo realiza análisis estadístico de tipo multivariado y univariado, cuyos resultados se muestran a través de diferentes tipos de gráficas. A pesar de que la app se encuentra funcionando correctamente su uso es limitado, está implementada bajo la plataforma de MATLAB, lo que requiere la instalación de software adicional y una licencia para su uso. Por este motivo se construye una aplicación web funcional con la mayoría de características ya existentes y algunas nuevas que mejoran la experiencia del usuario. Para lograrlo se ha implementado una aplicación que usa la arquitectura cliente-servidor, donde el procesamiento de datos está implementado con *Python* a través de una API REST y la interfaz de usuario usando tecnologías web en conjunto con *D3* para las visualizaciones de datos.

***Palabras clave* — Metabolómica, lipodómica, análisis de datos, visualización de datos, aplicación web, software de investigación, neurociencia.**

---

## ABSTRACT

This work deals with the migration of the application currently used by the neurosciences group of the Universidad de Antioquia for the analysis of metabolomic data. This prototype performs multivariate and univariate statistical analysis, whose results are displayed through different types of charts. Although the app is working properly its use is limited, since it is implemented under the MATLAB platform, which requires the installation of additional software and a license for its use. To solve this problem a web application was built with most of the existing features and some new ones that improve user experience. To achieve this, an application using client-server architecture has been used; here, data processing is implemented with *Python* through a REST API. On the other hand, the user interface is crafted along with web technologies in conjunction with *D3* for data visualizations.

***Keywords*** — **Metabolomics, Lipidomics, data analysis, data visualization, web application, research software, neuroscience.**

## I. INTRODUCCIÓN

Actualmente el grupo de neurociencias de la Universidad de Antioquia como parte de sus actividades analiza datos metabólicos para identificar patrones asociados a enfermedades neurodegenerativas. En colaboración con el grupo *Intelligent Information Systems Lab* de la Facultad de Ingeniería, ha desarrollado un prototipo de aplicación que les permite a través de diferentes técnicas de análisis de datos, observar el comportamiento de especies lipídicas en muestras biológicas. Sin embargo, esta herramienta no está disponible directamente para los investigadores, para utilizarla el encargado de la plataforma debe primero recibir los datos<sup>1</sup>, convertirlos en el formato requerido y finalmente enviar de vuelta los resultados. El problema principal es que los análisis anteriormente mencionados tienen varios parámetros que permiten filtrar o agrupar los datos con el fin de probar diferentes escenarios, esto sin dejar de lado, que a partir de un análisis se pueden obtener múltiples gráficas con las que se puede interactuar, cambiando el tipo de técnica de análisis, criterios de ordenamiento, cantidad de datos visibles, etc. Cuando hay un requerimiento específico, se debe hacer nuevamente el proceso anteriormente mencionado, incluso llegando a modificar el código fuente para atender nuevos requerimientos.

Llevar este prototipo a una versión de uso generalizado tiene varios retos y con éstos múltiples ventajas. Actualmente el prototipo está implementado bajo la plataforma de MATLAB<sup>2</sup>, y aunque tiene una gran variedad de funciones y herramientas para realizar simulaciones, requiere la instalación de software adicional y requerimientos puntuales del sistema para poder funcionar, además de una licencia para su uso. Se propone un cambio de tecnología que junto a las características actuales, permita fácilmente distribuir nuevas versiones, y darle todas las capacidades del prototipo actual directamente a los investigadores.

El objetivo principal de este trabajo es migrar el prototipo actual a una aplicación web. Antes de empezar este nuevo proyecto se hace una revisión del estado del arte y se encuentran algunas aplicaciones web como *LipidSig* [1] y *LION/web* [2], las cuales tienen disponibles las

---

<sup>1</sup> Los datos suelen venir de diferentes fuentes, como por ejemplo otras universidades y no se exige en ningún formato especial.

<sup>2</sup> MATLAB es una plataforma de programación y cálculo numérico utilizada por millones de ingenieros y científicos para analizar datos, desarrollar algoritmos y crear modelos.

técnicas estadísticas fundamentales para el análisis e identificación de lípidos relevantes en un análisis de grupos, pero no soportan el análisis por insaturaciones o longitud de carbonos de las moléculas. En pocas palabras, no hay herramientas que cumplan exactamente las necesidades del grupo de neurociencias y se justifica la creación de este proyecto. La nueva aplicación estará compuesta de dos grandes partes, por un lado el frontend o interfaz de usuario utilizando el framework *Next.js* en conjunto con *D3*<sup>3</sup>, entre otras librerías para la experiencia de usuario, y por el otro lado el backend, implementado con *Django*, un framework de *Python* para la creación de aplicaciones web que se encargará principalmente de procesar los datos y entregar los parámetros necesarios para mostrar las gráficas en el frontend. Además, se ha utilizado *Firebase* para la autenticación de usuarios y alojar archivos estáticos<sup>4</sup>. Finalmente, se hace el despliegue de la app. en los servidores del grupo de investigación en colaboración con su equipo de infraestructura.

---

<sup>3</sup> *D3.js* es una biblioteca de JavaScript ampliamente utilizada para la manipulación y visualización de datos en la web.

<sup>4</sup> Los datos a ser analizados son archivos CSV que los usuarios cargan dentro de la aplicación.

## II. OBJETIVOS

### *A. Objetivo general*

Desarrollar la versión web de la herramienta actualmente utilizada por los investigadores del GNA para el análisis de datos metabolómicos, ofreciendo una interfaz renovada y nuevas características que mejoren la experiencia de los usuarios.

### *B. Objetivos específicos*

- Revisar el estado del arte para encontrar aplicaciones similares a la que se quiere construir, tomando ideas que ayuden a construir un producto más completo.
- Identificar requisitos funcionales y no funcionales en el prototipo actual, que permitan comprender los retos que trae hacer la migración a una plataforma web, con esto se busca entender cómo son aplicadas las técnicas de análisis y cómo las gráficas son generadas a partir de estos resultados.
- Escoger e implementar una arquitectura que permita desacoplar la vista y el procesamiento de datos para poder utilizar las herramientas más adecuadas para cada problema.
- Diseñar una nueva interfaz que mejore la organización visual respecto al prototipo actual, evitando la saturación de elementos y que discrimine entre los tipos de análisis soportados.
- Desarrollar una aplicación web funcional con las características actualmente soportadas y algunas nuevas, donde los usuarios puedan tener sus propios proyectos y compartirlos con otros, analizar datos sin intermediarios y descargar en alta calidad los gráficos generados.
- Desplegar la aplicación en los servidores del GNA.
- y documentar la herramienta desarrollada. Redactar una guía de uso y una documentación técnica que permita entender los componentes de la aplicación y cómo añadir y desplegar nuevos cambios.
- Redactar una guía de uso para los investigadores y una documentación para los desarrolladores que explique el funcionamiento técnico de los componentes principales.

### III. METODOLOGÍA

Inicialmente está dividido en dos fases: Diseño & Planeación y Desarrollo. Para comprender el desarrollo del proyecto se divide en tres partes, la primera que habla sobre la parte transaccional que son las operaciones CRUD, segundo, el análisis de datos y por último la visualización de datos.

La realización del proyecto se divide en dos fases: la primera dedicada a la revisión, planeación y diseño, y posteriormente una segunda dedicada al desarrollo del proyecto.

#### *A. Primera fase (Revisión, planeación y diseño):*

1. El GNA está interesado en un grupo específico de datos metabolómicos, entonces se empieza buscando trabajos que busquen resolver el análisis para este tipo particular de datos, que preferiblemente estuvieran desarrollados para la web. Se encuentran un par de herramientas que se acercan a los requerimientos, dejando de lado el análisis por insaturaciones o longitud de carbonos de las moléculas. Además, la manera en cómo funciona internamente no es transparente y no se tiene una idea clara de cómo es realizado el análisis [1]. Las opciones encontradas son *LipidSig* [2] y *LION/web* [3] que para descartarlas se exploraron minuciosamente cada una de sus funcionalidades, ajustando parámetros y probando los diferentes tipos de análisis ofrecidos, para finalmente cotejar los resultados con los del prototipo.
2. Se empieza definiendo una metodología de trabajo junto al asesor, que permitiera iterar entre las diferentes actividades a medida que se recibía la retroalimentación en cada avance. Por lo tanto, se define un cronograma tentativo, que ayudara a visualizar si había algún desfase y empezar a cambiar el ritmo o a tomar decisiones que desbloquearan el avance del proyecto. Más adelante se discuten algunas limitaciones que se encontraron y cómo fue cambiando el enfoque y el alcance del proyecto para poder llegar a cumplir de la mejor manera los objetivos. En la (**Fig. 1**) se puede observar el cronograma planeado para una ejecución en un periodo de 6 meses, donde se resumen las actividades a realizar y los sus objetivos asociados. Para hacer el seguimiento de las actividades se establecieron reuniones semanales junto al asesor, dónde

en algunas de ellas participaron miembros del GNA para ayudar a definir algunas convenciones, como el formato del archivo de datos para los proyectos.

3. Uno de los requerimientos de este proyecto es que toda la parte de analítica fuera implementada con Python, ya que es un lenguaje usado ampliamente en el campo científico y de ingeniería de datos, además de ofrecer un amplio conjunto de librerías de *Machine Learning* y análisis estadístico. Otra de las razones es que haciéndolo de esta manera, desde el GNA les iba a ser más fácil agregar nuevas técnicas o hacer cambios en el código base. Sin embargo, este lenguaje no es muy conveniente a la hora de hacer páginas web que aprovechen las ventajas del *frontend* moderno, se decide hacer una separación en dos proyectos, uno dedicado al análisis de datos y el otro a la interfaz de usuario. La arquitectura cliente-servidor es la que mejor se adapta a este escenario, dónde el servidor a través de una API REST va recibir las peticiones del cliente (por ejemplo: los filtros, los parámetros y demás información para el análisis) y de regreso entrega los datos necesarios a partir de los resultados, para construir los gráficos en el cliente o interfaz de usuario (Error! Reference source not found.Error! Reference source not found.).
4. Además de las visualizaciones de datos (gráficas de barras, diagramas de dispersión, etc.), la aplicación tiene una interfaz para autenticarse, crear proyectos, seleccionar los filtros, cambiar entre gráficos, etc. Se hacen los diseños de las vistas, pensando en darle una mejor estética respecto al prototipo anterior, donde los elementos están visualmente mejor organizados, se puede cambiar entre tipos de análisis y solamente hay una gráfica visible para que la pantalla no se vea saturada de elementos, el usuario puede navegar entre las gráficas a través *tabs* y selectores, que se escogieron estratégicamente. Los diseños realizados están disponibles en el siguiente enlace <https://bit.ly/45tNY6K>.



Fig. 1. Cronograma de actividades del desarrollo del proyecto

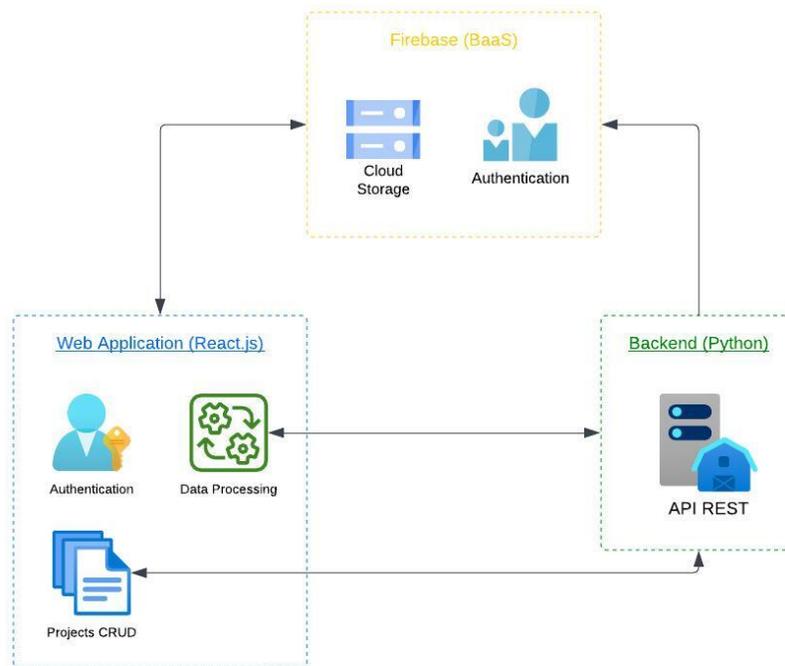


Fig. 2. Arquitectura cliente-servidor propuesta para el proyecto

### B. Segunda Fase(Desarrollo del proyecto)

El proceso de construcción de la aplicación se puede dividir en cuatro partes esenciales: la parte transaccional, que hace referencia a las partes de la aplicación que se resumen en un CRUD (*Create, Read, Update, Delete*) de datos; la interfaz de usuario; la implementación de las técnicas de análisis; y finalmente las visualizaciones de datos.

### 1) Operaciones CRUD

Cómo se menciona la sección anterior, se va a construir una API REST usando el lenguaje Python y para lograrlo existen varios *frameworks* y librerías que resuelven el problema. Dentro de éstas se encuentran dos muy populares *Flask* y *Django* que sirven para crear aplicaciones web, pero se escoge la segunda opción debido a su popularidad y completa documentación. Aunque *Django* tiene todas las herramientas necesarias para construir la API, en adición se usa una librería llamada DRF (*Django Rest Framework*) que hace que la creación de nuevos *endpoints* sea mucho más fácil y rápida, viene con un conjunto de clases que abstraen lógica común en este tipo de aplicaciones, evitando escribir código repetitivo.

Aparte de los servicios para la analítica de datos, también son necesarios otros que permitan gestionar proyectos y sus elementos asociados. Un proyecto es en esencia un archivo de datos provenientes de un experimento particular, pero que a su vez le hemos definidos otros atributos, algunos para describirlo cómo un título y una descripción, y otros para agregarle valor como un autor y filtros personalizados. Es importante aclarar que es obligatorio que un usuario esté autenticado dentro de la aplicación, al crear un nuevo proyecto el usuario actual será asociado cómo su autor y solamente éste tendrá permisos para visualizarlo y/o editarlo. Otro de los requerimientos de la aplicación es que los investigadores puedan ingresar con el correo institucional mediante su cuenta de Google, este tipo de integraciones, los requerimientos de seguridad y calidad requieren un esfuerzo considerable si se quiere construir un sistema de autenticación desde cero, por lo tanto se utiliza un software de terceros llamado *Firebase* que además tiene un servicio para alojar archivos estáticos, que en este caso son los archivos de los proyectos, cómo se menciona anteriormente. Los filtros personalizados nacen de un problema que se identificó durante el desarrollo y es que configurar los filtros para ejecutar un análisis puede tomar un tiempo considerable, cuando el investigador quiera probar otro escenario la configuración previamente hecha no se iba a poder recuperar, a esta configuración se le ha nombrado “Filtros personalizados”, que introducen la posibilidad de crear, guardar, editar y eliminar filtros pre-configurados.

## 2) Nueva interfaz de usuario

El prototipo actual presenta todos sus funcionalidades en una misma página, todos los elementos con los que puede interactuar el usuario están visibles al mismo tiempo. Como usuario es difícil identificar cada una de las herramientas disponibles y dónde ubicarlas, por ejemplo, si lo que se quiere ver es la gráfica de volcano, al mismo tiempo estábamos viendo el gráfico de componentes principales y las casillas de verificación para seleccionar la técnica de análisis, cuando el volcano no necesita de este parámetro. Para evitar esto se construye una nueva interfaz que además de solucionar estos problemas le dé una identidad visual a toda la aplicación. Esta tarea se lleva a cabo utilizando la tecnología de *React*, ya que es una librería que tiene una documentación muy completa y tiene actualmente una gran adopción entre los desarrolladores. La aplicación se construye con un *framework* de *React* llamado *Next.js* ya que provee la solución a muchos casos comunes, como el manejo de rutas, rendimiento, entre otros. Cuando se hizo el diseño de la aplicación no se hizo énfasis en cómo iban a lucir cada uno de los componentes, como los botones, los selectores, las pestañas, etc. sino que en su lugar se utilizaron *wireframes*, que a diferencia de pensar en los estilos, se preocupa por la ubicación de los elementos dentro de la página. Para manejar los estilos se escoge una librería llamada Material UI, que es una implementación de Material Design<sup>5</sup> para *React*, provee un conjunto de componentes ya hechos, estilizados y configurables. Es importante mencionar que los *wireframes* se hicieron pensando en que los componentes que se iban a utilizar luego iban a estar disponibles. No hay que dejar de lado, que hay otras librerías que también se utilizan que no hace falta necesario mencionar, pero que ayudan resolver problemas específicos del desarrollo.

## 3) Análisis de datos

Para poder entender cómo se realiza el análisis de datos en el prototipo actual, se revisa el código fuente para ver las funciones que están utilizando y los conceptos que están detrás. MATLAB por supuesto tiene muchas funciones disponibles dentro de su entorno, pero cómo se debe hacer en *Python*, hay que buscar que herramientas usar para extrapolar las mismas ideas al nuevo entorno. Aunque algunos métodos tienen su implementación propia los resultados debían ser mínimamente

---

<sup>5</sup> Material Design es un lenguaje de diseño creado por Google en el año 2014.

equivalentes a los del prototipo. Al final lo que el servidor de datos debe retornar al cliente son los parámetros necesarios para crear cada uno de los gráficos, con esta idea en mente, se sigue la estrategia de hacer ingeniería inversa que busca entender desde el gráfico hasta los datos (en esta dirección) la lógica que está en el medio. Por ejemplo, para la gráfica de componentes principales se debe mostrar las elipses de control, y cada elipse requiere sus coordenadas, su radio en el eje  $x$  y en el eje  $y$ , una vez se comprende como lo hace el prototipo, se busca la manera de hacer lo propio en Python. Para poder verificar que los datos eran equivalentes, se hicieron pruebas en un ambiente aislado utilizando pequeñas regiones de datos, probando diferentes herramientas y estrategias, para de a poco, poder ir llegando a cada una de las partes del gráfico. Éste fue el conocimiento que en resumen se fue replicando para cada una de las gráficas implementadas (TABLA II).

#### 4) Visualizaciones de datos

D3 (Data-Driven Documents) es una biblioteca JavaScript ampliamente utilizada para la visualización de datos en la web. Permite crear gráficos interactivos y dinámicos en una página web utilizando HTML, SVG y CSS, facilitando la manipulación de datos y la creación de visualizaciones personalizadas [4]. En el mercado se pueden encontrar muchas librerías que están construidas sobre D3 que proveen una interfaz más simple, a diferencia de D3 éstas tienen un enfoque declarativo, por ejemplo, si se quiere crear el gráfico de insaturaciones<sup>6</sup>, usando alguna de estas librerías sólo bastaría con pasarle los datos, alguna configuración para personalizar los estilos y con ya tendríamos un primer acercamiento bastante aceptable. Sin embargo, los requerimientos de la aplicación que se está construyendo son particulares, volviendo al ejemplo anterior, la gráfica de insaturaciones además necesita mostrar la desviación estándar, por citar otro ejemplo, el gráfico de dispersión de componentes principales tiene que mostrar las elipses de control de los grupos de análisis. Se necesita entonces hacerlo de una manera que permita tener un control minucioso de los elementos que se muestran en cada gráfica, por eso se decide usar D3 directamente, ya que ofrece un conjunto de funciones y herramientas que permite realizar visualizaciones personalizadas. Para entender mejor su funcionamiento supóngase que se quiere

---

<sup>6</sup> El gráfico de insaturaciones es un gráfico de barras que muestra la media de las insaturaciones de cada grupo de análisis por cada especie lipídica.

hacer un gráfico de dispersión y se tiene unos data en forma de lista y en cada posición de la lista tenemos el radio y la posición en  $x$  y en  $y$ , se itera el conjunto de datos y por cada ítem se crea un círculo y se le indica el radio y las coordenadas (atributos del ítem), D3 ya provee las funciones que mapean estos valores  $(x, y)$  al gráfico con base en su tamaño y así mismo las funciones para crear los ejes coordenados. Si extendemos este conocimiento al gráfico de componentes principales sólo faltan las elipses, siguiendo el mismo mecanismo, podemos definir una lista, dónde en cada posición se tienen los radios de la elipse  $(rx, ry)$  y su posición  $(x, y)$ , de la misma forma se itera la lista y se crea una elipse por cada ítem, durante esta iteración podemos agregar estilos entre otros atributos como rotación, translación, etc. De la misma forma es construyen todos y cada uno de los gráficos necesarios.

## IV. RESULTADOS

En esta sección, se presentan los resultados obtenidos en cada una de las etapas del trabajo. Se presentan los *endpoints* de la API desarrollada en Python, incluyendo los servicios para análisis de datos y cómo son implementadas cada una de las técnicas en relación con los datos necesarios para las gráficas, seguidamente se muestran los gráficos implementados dentro de la aplicación y finalmente se habla de la nueva interfaz.

Se crearon los *endpoints* para crear, listar, leer detalle y eliminar proyectos, no se contempla la opción de editar proyecto, ya que los únicos campos que tendría sentido editar sería la descripción y el título, ya que si se actualiza el archivo se debe regenerar todos los metadatos de nuevo en la base de datos, por simplicidad se remueve esta funcionalidad. Además se hace el CRUD completo para los filtros personalizados y un par de *endpoints* más para el análisis multivariado y univariado (**TABLA I**). Estos últimos reciben dentro de los datos de la petición los filtros y los parámetros para realizar el análisis.

TABLA I  
*Endpoints desarrollados*

Módulo		Ruta	Acción
<b>Proyectos</b>	POST	/api/projects	Crear
	GET	/api/projects	Listar
	GET	/api/projects/:projectId	Detalle
	DELETE	/api/projects/:projectId	Borrar
<b>Filtros personalizados</b>	POST	/api/custom_filters	Crear
	GET	/api/custom_filters	Listar
	GET	/api/custom_filters/:filterId	Detalle
	PUT	/api/custom_filters/:filterId	Actualizar
	DELETE	/api/custom_filters/:filterID	Borrar
<b>Análisis de datos</b>	POST	/api/analytics/multivariate	Análisis multivariado
	POST	/api/analytics/univariate	Análisis univariado

Se desarrollaron en la nueva versión el análisis multivariado y univariado. Para el análisis multivariado, se soportan las técnicas PCA, PLS-DA y OPLS-DA; por otro lado en el univariado se hacen los cálculos para mostrar sus gráficas correspondientes, la mayoría son operaciones matemáticas de matrices como la suma, media, desviación estándar, etc. Sin embargo, hay otra que sí que usan librerías que ya implementan muchos de los métodos necesarios y a continuación se muestra (**TABLA II**) cómo y dónde fueron utilizadas dentro del trabajo.

TABLA II  
LIBRERÍAS UTILIZADAS PARA EL ANÁLISIS DE DATOS

Librería	Descripción
Numpy	Proporciona soporte para matrices multidimensionales y funciones matemáticas para operaciones eficientes en estos arreglos. Se ha utilizado para hacer operaciones básicas como transponer una matriz, concatenar vectores, sumar matrices, etc.
Pandas	Librería de análisis de datos que proporciona estructuras de datos flexibles y herramientas de manipulación y análisis de datos. Se utiliza comúnmente para cargar, limpiar, transformar y analizar datos tabulares, como hojas de cálculo o conjuntos de datos CSV. Los archivos asociados a los proyectos se leen usando esta herramienta y además facilita el filtrado de datos para seleccionar únicamente la región de interés.
Scikit-learn	Librería de aprendizaje automático ( <i>machine learning</i> ) que proporciona una amplia gama de algoritmos y herramientas para tareas de clasificación, regresión, <i>clustering</i> y más. Se ha utilizado para la codificación de muestras, normalización de datos y para transformar datos usando la técnica de PCA.
SciPy	Librería que se basa en Numpy y ofrece funcionalidad adicional para realizar cálculos científicos y técnicos. Incluye módulos para optimización, integración numérica, interpolación, álgebra lineal, entre otros. Dentro del proyecto se ha usado para realizar <i>one-way ANOVA</i> , que en pocas palabras son los datos que necesita el gráfico de Volcano.
OPLS-MD	Regresores OPLS y PLS con cierta utilidad para ayudar en el análisis de simulaciones de dinámica molecular (MD). Básicamente se usa para soportar las técnicas de análisis PLS_DA y OPLS_DA.

La mayoría de las gráficas existentes en el prototipo fueron migradas (**TABLA III**), sin embargo, las versiones en 3D no fueron desarrolladas por problemas de implementación, limitación que se discute más adelante. Todos los gráficos son descargables, solamente en formato SVG. Las imágenes o gráficos en este formato tienen la característica de que no degradan su calidad cuando son ampliados ya que no utilizan píxeles sino vectores en su lugar.

TABLA III  
GRÁFICAS IMPLEMENTADAS EN LA NUEVA VERSIÓN

<b>Tipo</b>	<b>Nombre</b>
<b>Gráfico de barras</b>	Barras de insaturaciones
	Barras de longitud de carbono
<b>Diagrama de dispersión</b>	Diagrama de componentes principales
	Diagrama de pesos
	Volcano
<b>Mapa de calor</b>	Matriz de correlación
<b>Gráfico de contorno</b>	Gráfico de contorno

Con la versión web también llega una nueva interfaz (**Fig. 3**), que tiene una sección dedicada a los filtros, donde se puede seleccionar las familias lipídicas, material biológico y, a través de un componente nuevo, agrupar grupos experimentales, en la parte derecha de esta sección se encuentra el selector de filtros personalizados y las posibles acciones que se pueden hacer sobre éstos. En la parte de abajo se encuentra la sección donde el usuario va a poder cambiar entre tipos de análisis y seleccionar los parámetros para las técnicas de análisis, además de navegar entre cada una de las gráficas. Algunas de éstas tienen la característica de ser dinámicas, esto significa que tienen configuraciones propias que permiten modificar el contenido.

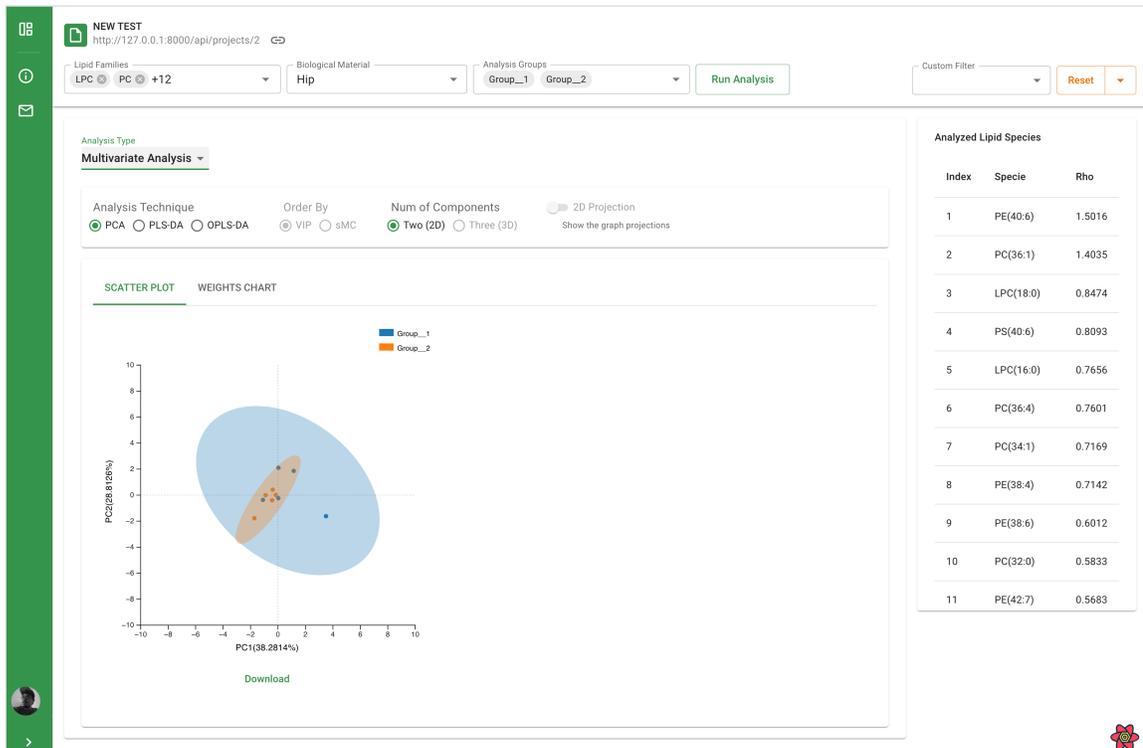


Fig. 3. Pantallazo de la app después de ejecutar un análisis multivariado

Como además se puede observar en la anterior figura, hay una barra lateral que permite cerrar la sesión de usuario, pero además entrar a la página de proyectos (**Fig. 4**), desde allí se listan en una tabla los proyectos creados por el usuario y se puede ingresar directamente en alguno de los tipos de análisis ofrecidos o eliminarlo. El botón en la parte superior derecha lleva al usuario a la página de creación de proyectos (**Fig. 5**) y una vez creado es re-direccionado de vuelta mostrando el proyecto nuevo en la lista.

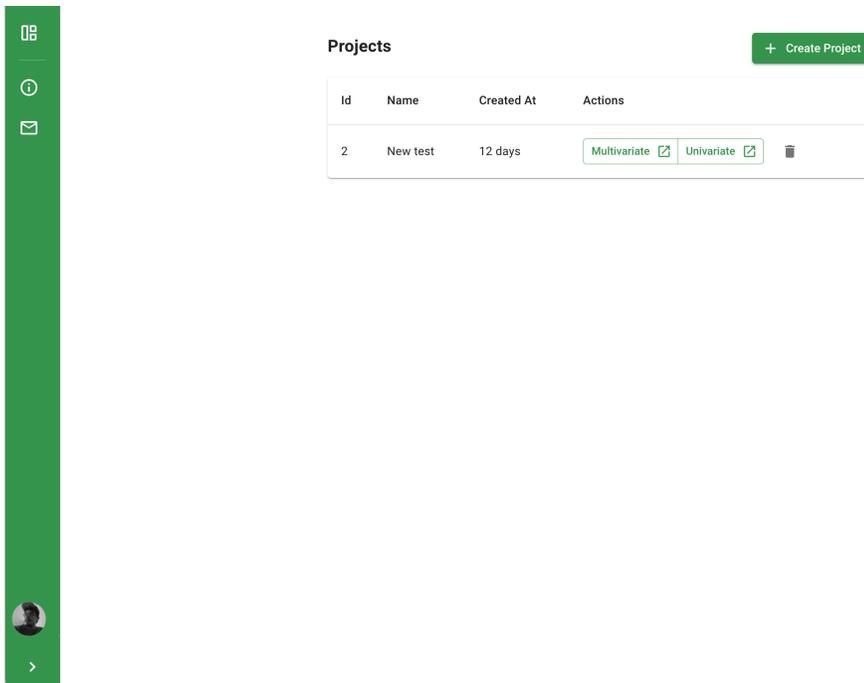


Fig. 4. Vista dónde se listan todos los proyectos creados por el usuario

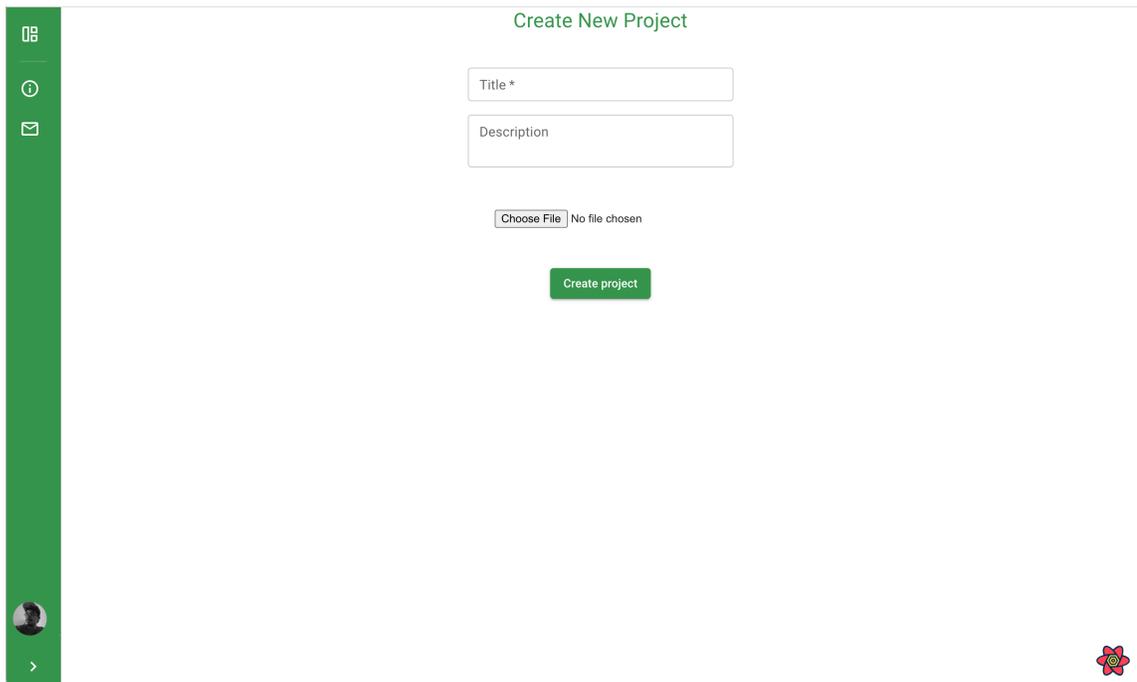


Fig. 5. Vista para crear un nuevo proyecto

## V. CONCLUSIONES

La aplicación que se construyó cuenta con la mayoría de las características del prototipo, algunas de ellas no se lograron implementar principalmente debido a que el tiempo disponible para el desarrollo es corto, y se decidió darle prioridad a otras tareas que ayudaran de igual forma a garantizar un primer producto mínimo viable, que sentara unas bases sólidas para esta plataforma. Este proyecto debe verse cómo un punto de partida para construir todo un entorno de trabajo que apoye la investigación del grupo de neurociencias de la Universidad de Antioquia, que a su vez trabaja de la mano con otras universidades.

El eje central de la plataforma es el procesamiento de datos, consta en tomar los datos provenientes de experimentos realizados por investigadores del área de neurobiología, para darles valor a través de las técnicas de análisis de datos adecuadas. La arquitectura propuesta permite que este motor de procesamiento sea utilizado desde otros clientes, como aplicaciones móviles, páginas web, servidores, funciones en la nube o cualquier otro recurso que pueda comunicarse a través a internet. Se abren muchas posibilidades cuando se construye un producto que no tiene requerimientos especiales para ser usado, ya que se puede adaptar para trabajar junto con otras herramientas, para seguir apoyando la investigación en este campo.

Los servicios realizados para el análisis de datos en un comienzo no eran equivalentes cuando se contrastaban con los del prototipo, la primera fue la técnica de PCA que durante su desarrollo se tuvo que iterar más de una vez, porque se obtenían valores para las elipses de control que ni siquiera se acercaban a los esperados. Desde este punto se empieza a retrasar el trabajo, pero igual se sigue el cronograma establecido. Luego de terminar la primera versión estable del análisis multivariado, se empieza la construcción de las gráficas en 3 dimensiones con poco éxito. La librería que es utilizada para crear las visualizaciones de datos no tiene soporte oficial para gráficos en 3D, no obstante las pocas librerías de terceros que para extender esta funcionalidad para D3 se limitaban a gráficos básicos, eran ineficientes y estaban desactualizadas. En el contexto de las aplicaciones web existen otras herramientas que ayudan a la creación de visualizaciones en 3D, pero están enfocadas en experiencias visuales y no en gráficos a partir de datos. Aunque es posible hacer éste tipo de gráficos utilizando las herramientas adecuadas, iba a llevar mucho tiempo, ya

que muchas cosas se iban a tener que hacer desde cero. En ese momento, se decide dejar el análisis 3D a un costado y continuar con el resto de actividades. Si bien no se lograron todas las características del prototipo actual, si se llega a una versión completa, aunque más acotada.

Crear una aplicación desde cero permite seleccionar tecnologías actuales, aquellas herramientas más idóneas para cada requerimiento. Además, es una oportunidad para re-pensar y cambiar los aspectos mejorables dentro del prototipo, que por las circunstancias se dieron de esa manera. Como se puede observar en la (**Fig. 3**) anteriormente los filtros, las gráficas y la selección de parámetros para las técnicas de análisis convivían en una misma página, que resultaba en experiencia de usuario abrumadora debido a la saturación de elementos visuales. La nueva interfaz tiene una organización que permite identificar a que dominio pertenece cada elemento, ahora hay una separación de conceptos e ideas, que a su vez facilitan la navegación visual a través de la aplicación. En términos de la experiencia de usuario, anteriormente al cerrar la aplicación se perdían las configuraciones que se habían hecho, cada vez que el usuario ingresaba a la herramienta debía volver a seleccionar todos los filtros y volver a ejecutar el análisis, ahora se hace una persistencia en el navegador del último estado de la aplicación, para evitar justamente volver este efecto reconfigurar todo el ambiente de trabajo, esto no debe confundirse con los filtros personalizados, que son respaldados en la base de datos y que están para guardar filtros que pueden usarse luego dentro de cada proyecto particular. Finalmente, hay que rescatar el hecho de que ahora los investigadores no necesita tener que cargar un nuevo archivo modificando el código base para poder utilizar un nuevo set de datos, sino que justamente el módulo de proyectos está pensado para que cada usuario pueda gestionar sus propios trabajos, que se crean a través de la interface. Inicialmente se quiso que los proyectos se pudieran compartir con otros usuarios, para habilitar el trabajo cooperativo sobre el mismo set de datos, lo que una vez por falta de tiempo no se logró hacer.

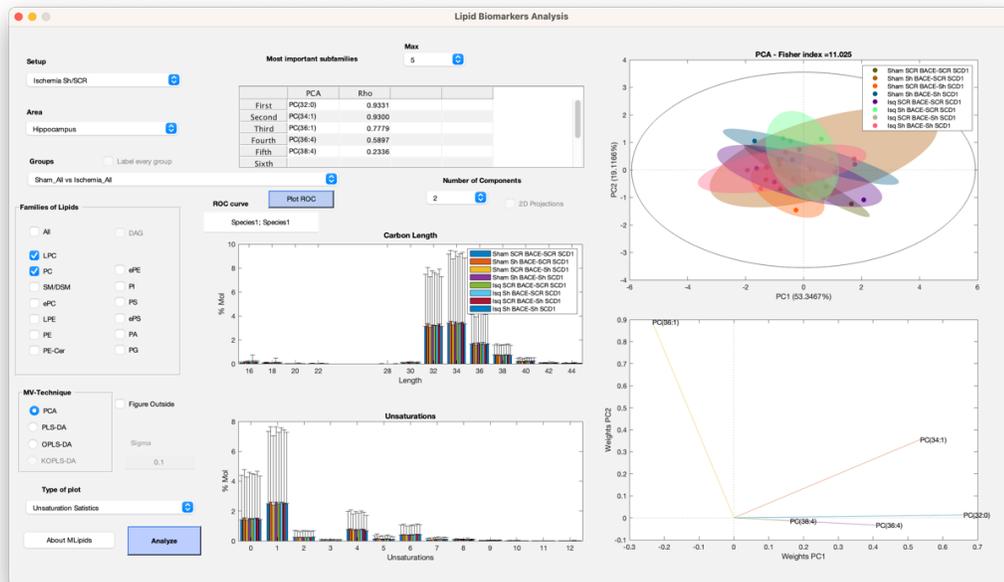


Fig. 6. Vista del prototipo actual

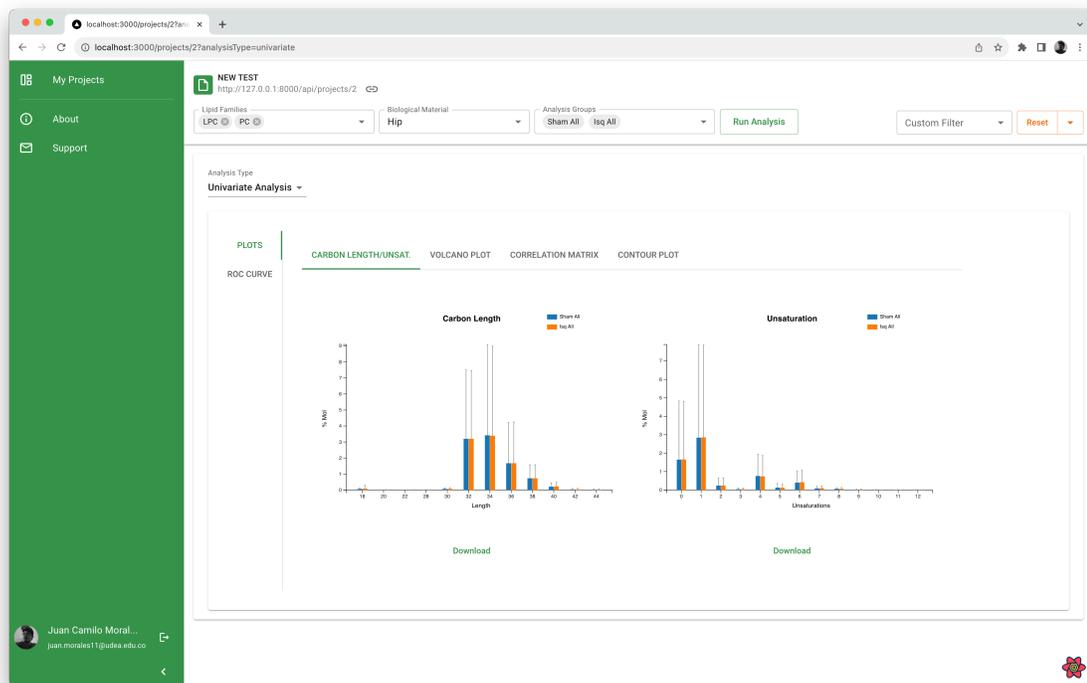


Fig. 7. Vista del nueva versión web

## VI. RECOMENDACIONES

Este proyecto es un comienzo, un primera versión alfa que recoge la mayor cantidad de características del prototipo posibles y en la que se han tomado decisiones de diseño que hacen la aplicación simple. Así cómo las características que se quedan pendientes también hay otras que sería bueno explorar en trabajos futuros con el fin de mejorar la experiencia del usuario y se listan a continuación:

- **Gráficos en 3D:** Tal y cómo se menciona en la metodología el tiempo se quedó corto para implementar gráficos en 3D. Esto se debe principalmente a dos cosas: la primera es que D3 no tiene soporte oficial para gráficos en tercera dimensión y si bien hay algunas librerías de terceros, no están bien documentas y/o actualizadas. En segunda instancia, los gráficos a partir de datos son un tema que aún no está muy desarrollado en la web. Existen muy buenas librerías y técnicas para hacerlo, sin embargo, éstas están más enfocadas a crear experiencias visuales que a gráficos de carácter científico. En este orden de ideas, implementar gráficos en 3D claro que es posible, pero es muy demandante y se prefiere dejar esta característica para un trabajo futuro.
- **Gráficos interactivos:** Actualmente D3 ofrece la posibilidad de agregar interacción a los gráficos, por ejemplo hacer zoom, que al pasar el mouse sobre cierto elemento te muestre ciertos detalles o que cuando la gráfica aparezca en pantalla tenga una animación particular, entre muchas otras cosas que a algunas gráficas le quedarían bastante bien.
- **Convertidor de archivos CSV:** Aunque ya se ha definido un formato estándar para el archivo fuente de cada proyecto, estos resultados vienen de diferentes fuentes y en diferentes formatos, entonces cuando un investigador quiera usar la app, debe convertir sus datos en el formato requerido y normalmente los archivos son bastante grandes. Esto es algo que la misma app. puede llegar a resolver, creando algunos transformadores para las fuentes más comunes y que incluso permita descartar el archivo convertido. A esta misma idea se puede sumar la habilidad de usar otros formatos similares a CSV cómo Excel u otros en caso de que en verdad ayude a los usuarios.

## REFERENCIAS

- [1] R. Spicer, R. M. Salek, P. Moreno, D. Cañueto, and C. Steinbeck, "Navigating freely-available software tools for metabolomics analysis," *Metabolomics*, vol. 13, no. 9, 2017.
- [2] W.-J. Lin et al., "LipidSig: A web-based tool for Lipidomic Data Analysis," *Nucleic Acids Research*, vol. 49, no. W1, 2021. doi:10.1093/nar/gkab419
- [3] M. R. Molenaar et al., "Lion/Web: A web-based ontology enrichment tool for lipidomic data analysis," *GigaScience*, vol. 8, no. 6, 2019. doi:10.1093/gigascience/giz061
- [4] M. Bostock, V. Ogievetsky and J. Heer, "D<sup>3</sup> Data-Driven Documents," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301-2309, Dec. 2011, doi: 10.1109/TVCG.2011.185.