

Autor

María Elisa Honorato Lira

Profesor guía

Alejandro Durán Vargas

Agradecimientos

A mi papá y mamá, por toda la ayuda y motivación con este proyecto y por apoyarme desde el primer día con la carrera (y con todo en la vida). A mi mamá en especial, por ser mi fiel ayudante, incluyendo varias traspasadas, muchísimas ideas y sobre todo mucha paciencia.

Al resto de mi familia, amigas y amigos en general, por estar siempre dispuestos a escuchar con entusiasmo mis ideas, así como también aportar desde su lado con su visión y opiniones.
A mi profesor Alejandro, por los buenos consejos y por ser un gran guía, siempre presente y con la mejor disposición a ayudar desde el día uno.

A todas mis amigas universitarias, por acompañarnos y vivir juntas no sólo ésta última, sino todas las experiencias de estos 5 años que pasaron. Mejor compañía no pude haber tenido.

Agradecimientos	2	
1. Resumen Ejecutivo	4	
Resumen del proyecto	5	
Principales logros	5	
Conclusiones clave	5	
2. Introducción	6	
Nuevos Desafíos del Método Científico	7	
Actual Estado del Conocimiento	8	
Caso de Estudio	9	
3. Descripción del Problema	10	
Identificación del Problema	11	
Análisis detallado del Problema	13	
La Interactividad como Objetivo Proyectual		13
4. Marco Teórico	15	
Epistemología	16	
de la Visualización	16	
Reconocimiento de Patrones Visuales	18	
Visualización de Datos	19	
Teoría de la Gestalt	24	
Visualización 3D	25	
8 Estado del Arte	26	
Antecedentes	27	
Referentes	28	
Patrón de Valor	31	
9. Contexto y Usuario	32	
10. Formulación/Objetivos	34	
11. Proceso de desarrollo	36	
Requisitos del Software	37	
Arquitectura del Software	40	
Api y Servicios	43	
Desarrollo de Software	44	
12. Implementación	61	
Business Model Canvas	63	

1. Resumen Ejecutivo

Resumen del proyecto

La proliferación de mecanismos automatizados de obtención de datos y el consecuente aumento en el tipo y magnitud de estas bases de datos (BBDDs) ha desencadenado un cambio en el paradigma epistemológico sobre la obtención del conocimiento científico. El método tradicional de teoría basada en hipótesis, experimentos y simulaciones avanza hacia una ciencia exploratoria intensiva en información y basada en datos. La cantidad de investigaciones publicadas y su diversidad disciplinar van más allá de lo que los expertos de un campo determinado pueden razonar. Es así como surge la oportunidad de avanzar en nuevas formas de integrar datasets desde múltiples fuentes de información de gran magnitud y distancia disciplinar.

Debido a nuestra capacidad evolutiva de descubrir patrones visuales, una de las mejores formas de explorar y entender los datos, es con su visualización. En este sentido, el diseño visual de la información es importante no solo para aumentar la perceptibilidad, sino también para revelar los patrones que se presentan dentro de la información compleja.

De esta manera se presenta LiHO, una plataforma open-source de visualización tridimensional multivariable de bases de datos científicos, una herramienta que permita a la comunidad científica, desarrollar nuevas formas de conocimiento desde la complejidad de los datos con el fin de poder analizar, manipular y visualizar los niveles de información actuales desde la concatenación de campos homologables.

Principales logros

Este proyecto logra los objetivos propuestos, ayudando a la comunidad científica a visualizar datos de mejor manera, sirviendo para entender mejor los datos, y de apoyo en una investigación científica

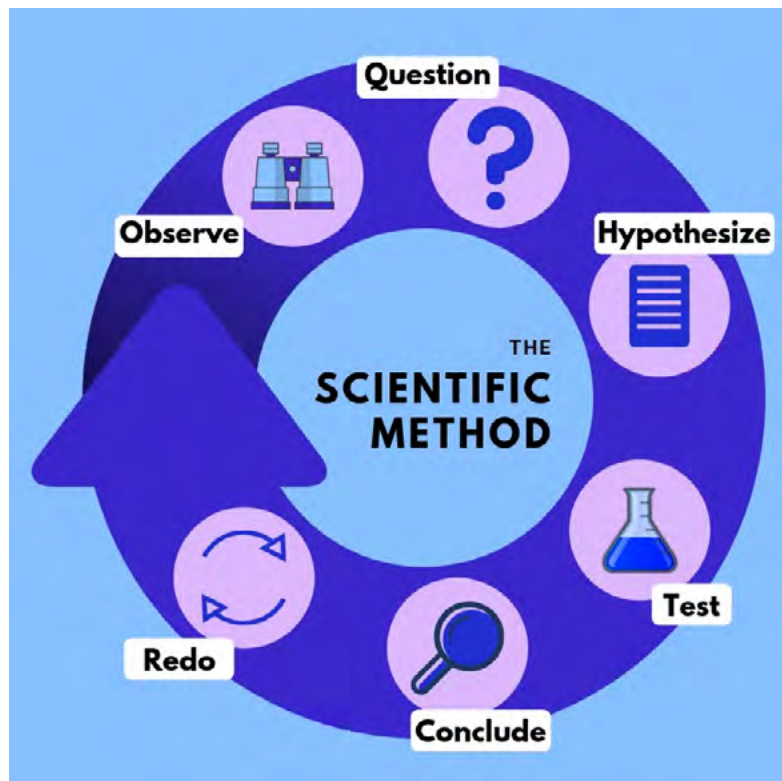
Conclusiones clave

Tras analizar los resultados obtenidos, se concluye que, si bien la solución implementada permite representar los datos, existe la necesidad de avanzar en la visualización de datos científicos. Si bien el enfoque actual es adecuado para algunos casos, se requiere un mayor desarrollo en la creación de gráficas y representaciones visuales que faciliten la comprensión y el análisis de los datos.

2. Introducción

Mientras los otros animales sólo están en el mundo, el hombre trata de entenderlo.

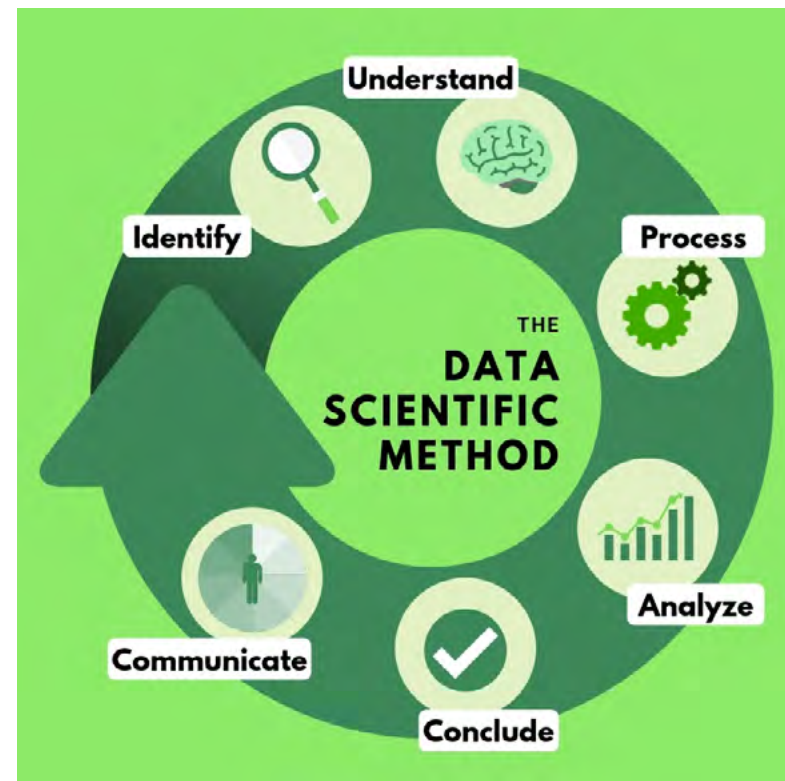
En este proceso, el hombre construye un mundo artificial que consiste en un creciente cuerpo de ideas llamado «ciencia», que puede caracterizarse como conocimiento racional, sistemático, exacto, verificable y por consiguiente falible. Por medio de la investigación científica el hombre ha alcanzado una reconstrucción conceptual del mundo que es cada vez más amplia, profunda y exacta (Bunge 2018).



(Fig 1)

CAMBIO DE

PARADIGMA



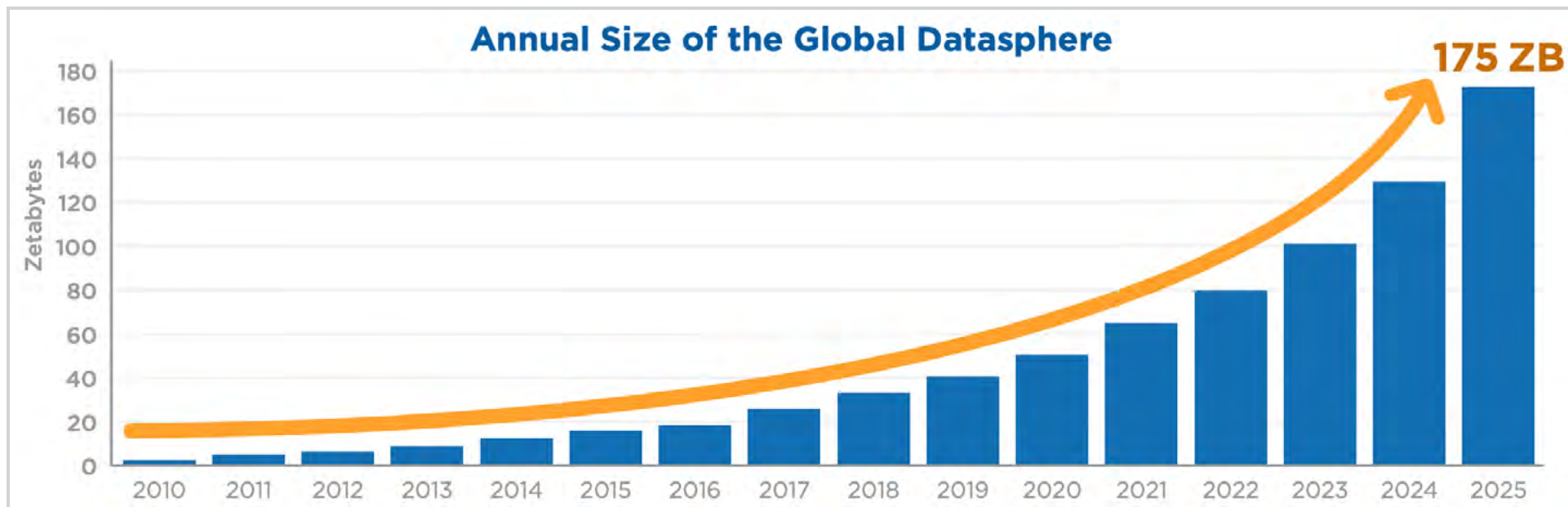
(Fig 2)

Nuevos Desafíos del Método Científico

Aquellos que estudiamos, ejercemos o investigamos el Diseño podemos consensuar que nuestra disciplina no se acota sólo a una serie de pasos que permiten llegar a un resultado (Buchanan, 1992). Si bien el Design Thinking ha favorecido una ampliación del valor de nuestro trabajo (Davis, 2010; Dorst, 2011; Glen et al., 2015; Plattner et al., 2018; Vinh, 2018), los procesos desde los cuales los diseñadores abordamos la acción de proyectar son mucho más complejos y variados que las fases enunciadas por el equipo de DSchool en 2008. (Jen, 2017; Vinsel, 2017). Cuando nos referimos a la generación de conocimiento científico, ocurre algo similar:

A menudo, el 'método científico' se presenta en los libros de texto y las páginas web educativas como un procedimiento fijo de cinco o seis pasos que comienza con la observación y cuestionamiento de un fenómeno, y continúa con la formulación de una hipótesis que explica el fenómeno, el diseño y la realización de experimentos para probar la hipótesis, analizando los resultados y finalizando con la elaboración de una conclusión (Fig 1). Dichas referencias a un método científico universal se pueden encontrar en material educativo en todos los niveles de educación científica (Blachowicz 2020) y numerosos estudios han demostrado que la idea de un método científico general y universal a menudo forma parte de la concepción de la ciencia tanto de estudiantes como de profesores (ver, por ejemplo, Aikenhead 1987; Osborne et al. 2003).

Sin embargo, en la era actual, el método tradicional de teoría basada en hipótesis, experimentos y simulaciones ha desencadenado un cambio de paradigma (Fig 2) hacia una ciencia exploratoria intensiva en información que es más bien colaborativa, en red y basada en datos (Bollier & Firestone, 2010; Kitchen, 2014; Trabucchi y Buganza, 2019; Masnick & Morris, 2022). La ciencia basada en Big Data renueva la primacía del razonamiento inductivo en forma de empirismo basado en tecnología y ha inspirado una visión del futuro en la que la extracción de datos automatizada conducirá directamente a nuevos descubrimientos. (Mazzocchi, 2015). El método estándar simplemente no puede manejar suficientemente el gran volumen de documentos o el crecimiento exponencial de los conjuntos de datos.



(Fig 3)

Big Data

El crecimiento exponencial de los datos ha planteado un desafío de gran envergadura en diversos ámbitos, como el tráfico de Internet. En este contexto, se genera una cantidad masiva de información que resulta difícil de almacenar sin asignar recursos significativos a este fin. Como consecuencia de esta creciente dependencia de los datos, se espera una expansión interminable en el tamaño de la Esfera Global de Datos. Según las estimaciones de IDC, en 2018 la Esfera Global de Datos se situaba en 33 ZB, pero se pronostica que crecerá hasta alcanzar los 175 ZB para el año 2025 (**Fig 3**). Esta rápida expansión plantea desafíos tanto en términos de capacidad de almacenamiento como de gestión y análisis de estos datos a gran escala.

Este fenómeno ha impulsado un cambio de paradigma en la ciencia, donde se ha intensificado la exploración de información basada en datos, fomentando la ciencia colaborativa

en red (Bollier & Firestone, 2010; Kitchin, 2014; Trabucchi y Buganza, 2019; Masnick & Morris, 2022). A su vez, la ciencia basada en Big Data ha revolucionado el enfoque en el razonamiento inductivo y el empirismo respaldado por la tecnología, abriendo nuevas perspectivas en las que la extracción automatizada de datos conduce directamente a descubrimientos innovadores (Mazzocchi, 2015).

Procesamientos que sobrepasan la capacidad humana

El método científico convencional se enfrenta al desafío de manejar el volumen de documentos y el crecimiento exponencial de los conjuntos de datos. Además, la acumulación de conocimiento en un dominio específico se vuelve cada vez más difícil de consumir y retener en la memoria humana, planteando una amenaza significativa para la adquisición y retención de conocimientos científicos (Kaisler et al., 2013)

Ante el crecimiento exponencial de los datos, es necesario explorar enfoques y tecnologías que vayan más allá de las capacidades humanas. El estudio metagenómico, el análisis de big data y otros campos similares requieren el uso de herramientas automatizadas, algoritmos avanzados y técnicas de aprendizaje automático para extraer conocimiento y obtener resultados significativos a partir de los grandes volúmenes de información disponibles.

Actual Estado del Conocimiento

Desafíos en la Era de la Información

En la era de la información, la cantidad de investigación publicada abarca un amplio espectro disciplinar, más allá de lo que los expertos de un campo determinado del conocimiento pueden razonar humanamente (Ding, 2016). El método para generar una hipótesis basada en la literatura, comienza a ser un proceso fortuito de gran incertidumbre, incluso más ahora por la gran cantidad de artículos publicados diariamente. El número medio de autores por publicación de PubMed ha aumentado de 1,94 a 5,61 desde 1965. Esto sugiere que ya no es posible que los expertos de un área, puedan manejar completamente el conocimiento de su propia disciplina. La especialización en las ciencias exactas también ha crecido con la expansión de la base de conocimiento científico, aumentando la necesidad de colaboración técnica y la interdisciplinariedad.

Las comunidades de investigación en muchas disciplinas finalmente están reconociendo que con los avances en la tecnología de la información, es necesario que haya nuevas formas de extraer entidades de publicaciones cada vez más intensivas en data e integrar y analizar conjuntos de datos a gran escala. (Ding et al., 2013).

Las bases de datos como herramientas de justificación de hipótesis

En el ámbito de la microbiología marina, los avances en modelos epistemológicos y tecnológicos han revolucionado la forma en que se llevan a cabo las investigaciones, permitiendo un mayor grado de precisión y exhaustividad (Handelsman et al., 1998; Gilbert et al., 2014; Wang et al., 2012). Sin embargo, es crucial abordar de manera crítica el uso de los datos y reconocer las limitaciones inherentes a los algoritmos, así como el papel esencial de los científicos en la interpretación y aplicación de los resultados obtenidos.

Es importante afirmar que, este nuevo enfoque no es simplemente jugar con datos para generar una hipótesis (van Helden, 2013), ya que de ser así, los científicos serían reemplazados rápidamente por algoritmos (Prensky, 2009; Dyche, 2012; Reichstein et al., 2019). Esta idea plantea un riesgo significativo (Lazer et al., 2014; Boulamwini y Gebru, 2018), como indican Balazka y Rodighiero (2020), al sostener suposiciones problemáticas sobre la neutralidad de los datos, lo cual puede generar una excesiva confianza en su precisión y subestimar los riesgos y las cuestiones epistemológicas asociadas.

En este contexto, es importante destacar la reflexión planteada por Anderson (2008), Kelling et al. (2009), Prensky (2009), Dyche (2012) y Torrecilla y Romo (2018), quienes señalan que los usuarios de big data a menudo se niegan a considerarse como profesionales reflexivos que interactúan activamente con los datos. En cambio, promueven una imagen de científicos como meros oyentes neutrales del concierto producido por el mundo en movimiento. Sin embargo, es esencial reconocer las limitaciones de esta perspectiva, ya que implica una confianza excesiva en la exactitud de los algoritmos y una subestimación de los riesgos y cuestiones epistemológicas asociadas (Balazka y Rodighiero, 2020).

En ámbitos como el de la metagenómica, donde la diversidad de microorganismos en las comunidades microbianas es abrumadora, el análisis manual de los datos se vuelve prácticamente imposible debido a la abundancia y variedad de especies presentes en una investigación (Handelsman et al., 1998; Gilbert et al., 2014; Wang et al., 2012). Por lo tanto, se hace necesario recurrir a técnicas de secuenciación de alto rendimiento y herramientas de bioinformática avanzadas. En vista de la complejidad inherente a este tipo de análisis de datos, se han desarrollado herramientas de visualización que permiten representar de manera clara y comprensible la diversidad de especies, las relaciones filogenéticas, la abundancia relativa de cada microorganismo y las interacciones entre los genes y las funciones dentro de las comunidades microbianas. Estas visualizaciones juegan un papel crucial en la interpretación y el análisis de los datos, permitiendo a los investigadores obtener una visión más completa y detallada de los fenómenos estudiados.

Caso de Estudio

En este contexto, el Laboratorio de Microbiología Marina UC (LabMicMar) emerge como un destacado caso de estudio en el análisis de las interacciones microbiológicas que trascienden las capacidades individuales. Consciente del desafío planteado por el crecimiento exponencial de los datos, el laboratorio reconoce la necesidad de emplear enfoques y tecnologías avanzadas.

El LabMicMar (**Fig 4 y Fig 5**) se centra en el análisis de comunidades microbianas marinas y en cómo responden frente a perturbaciones ambientales. Para llevar a cabo su investigación, se basa en el análisis del ADN comunitario o metagenómico, utilizando técnicas de secuenciación de alto rendimiento y herramientas avanzadas de bioinformática.

Además, se propone una colaboración con el Taller Lowtech & Ciencia UC, que ha establecido una experiencia de cooperación transdisciplinaria desde el año 2019 con el curso Microbiología Marina UC - BIO277M, dirigido por el Dr. Rodrigo De la Iglesia, director de pregrado de Ciencias Biológicas y académico a cargo del laboratorio. Esta colaboración ofrece oportunidades para la implementación de enfoques de visualización experimental en el análisis de datos microbiológicos.

Para llevar a cabo la investigación se hizo un levantamiento de información a través de entrevistas al Dr. Rodrigo De la Iglesia, para indagar en las herramientas de análisis y visualización de datos utilizadas por sus investigadores, entrevistas con el profesor guía del proyecto, quien se encuentra desarrollando su programa doctoral en asociación con este laboratorio y el departamento de Ecología, así como la observación etnográfica en el laboratorio de Microbiología Marina y en cátedras



(Fig 4)

del Taller Lowtech, dictadas por De la Iglesia y Durán, donde se tuvo la oportunidad, previo a la investigación, de participar como estudiante.

La cantidad y complejidad de datos que se encuentran disponibles tanto en el laboratorio como en bases de datos públicas, presentan una oportunidad en el campo de su visualización que no está siendo potenciada actualmente. Sobre esto, el Doctor Rodrigo de la Iglesia afirma: “Tenemos muchas bases de datos disponibles, y muy pocas personas hacen uso de ellas. Es necesario comenzar a darle a los datos la importancia que merecen. En ese sentido, hay algunas plataformas con las que nosotros trabajamos todo el tiempo, pero que su interfaz es muy general. Sería muy interesante ver una propuesta más específica, o algún sistema para visualizar datos complejos”.



(Fig 5)

“Tenemos muchas bases de datos disponibles, y muy pocas personas hacen uso de ellas. Es necesario comenzar a darle a los datos la importancia que merecen. En ese sentido, hay algunas plataformas con las que nosotros trabajamos todo el tiempo, pero que su interfaz es muy general. Sería muy interesante ver una propuesta más específica, o algún sistema para visualizar datos complejos”.

3. Descripción del Problema



(Fig 6)

Identificación del Problema

Durante esta etapa, se realizó un análisis exhaustivo de las necesidades de la comunidad científica en cuanto a la visualización de datos. Esto implicó la realización de entrevistas a varios investigadores y la consulta de diversas fuentes para comprender los desafíos que enfrentan al trabajar con grandes conjuntos de datos. En colaboración con el Profesor Rodrigo de la Iglesia C, se identificaron interacciones críticas de interés que resultaron fundamentales para definir el contexto de la problemática.

Manejo de Aplicaciones Especializadas: Actualmente la visualización de datos científicos se realiza comúnmente mediante aplicaciones especializadas. Se han propuesto diferentes

técnicas y herramientas de software para llevar a cabo el análisis de mapas científicos. Desde el Laboratorio de Microbiología Marina UC, se utilizan día a día estas herramientas. Sin embargo, existe una falta de manejo de estas aplicaciones, lo que impide un proceso fluido y óptimo para el usuario en cuestión. El uso de programación, junto con la cantidad de manejadores, pueden dificultar el aprendizaje por parte de quienes lo utilizan.

Formas de representación actuales sólo grafican magnitudes: Las formas de representación usadas por la comunidad científica actual acostumbran a graficar magnitudes; estos pueden ser gráficos de barra, punto o línea. Esto se contradice con nuestra habilidad para reconocer patrones, ya que se está desperdiciando una oportunidad de generar conocimiento a través de ellos. Véase por ejemplo la cantidad de juegos

gráficos con fines de seriación o percibir diferencias que hay entre los objetos. Surge de este fenómeno, el desafío de reproducir geometrías tridimensionales a partir de datos, capaces de generar patrones. En este sentido, John Maeda redefinió el uso de los medios electrónicos como instrumento de expresión al yuxtaponer técnicas artísticas con programas informáticos avanzados, generando formas artísticas fuera de lo tradicional a partir de datos.

Basándose en este marco proyectual y complementándolo con una investigación bibliográfica (Fig 6), se procedió al desarrollo de un primer prototipo, con el desafío de representar la cantidad de clorofila en una muestra de agua.

1	Muestra	a	b	Bco
2	CL.L.T 2	76973,1	51198,5	1268,9
3	CL.Q.T 1	70992,8	40577,8	1268,9
4	CL.Q.<3 2	24995,5	15556,7	1268,9
5	CL.L.T 1	85815,6	57209,9	1268,9
6	CL.Q.T 2	85310,3	48971,6	1268,9
7	CL.Q.<3 1	39609,4	23480,7	1268,9
8	CL.L.<3 3	18469,3	12712,4	1268,9
9	CL.L.<3 4	18947,8	13042,5	1268,9
10	CL.Q.T 4	65303,9	38675,5	1268,9
11	CL.L.T 4	59945,3	39720,3	1268,9
12	CL.Q.T 3	383810,3	214293,7	1131,7
13	CL.Q.<3 3	93037,8	52976,7	1131,7
14	CL.L.<3 2	28280,7	18826,4	1131,7
15	CL.L.<3 1	31530,8	20248,6	1131,7
16	CL.L.T 3	70832,0	47538,9	1131,7

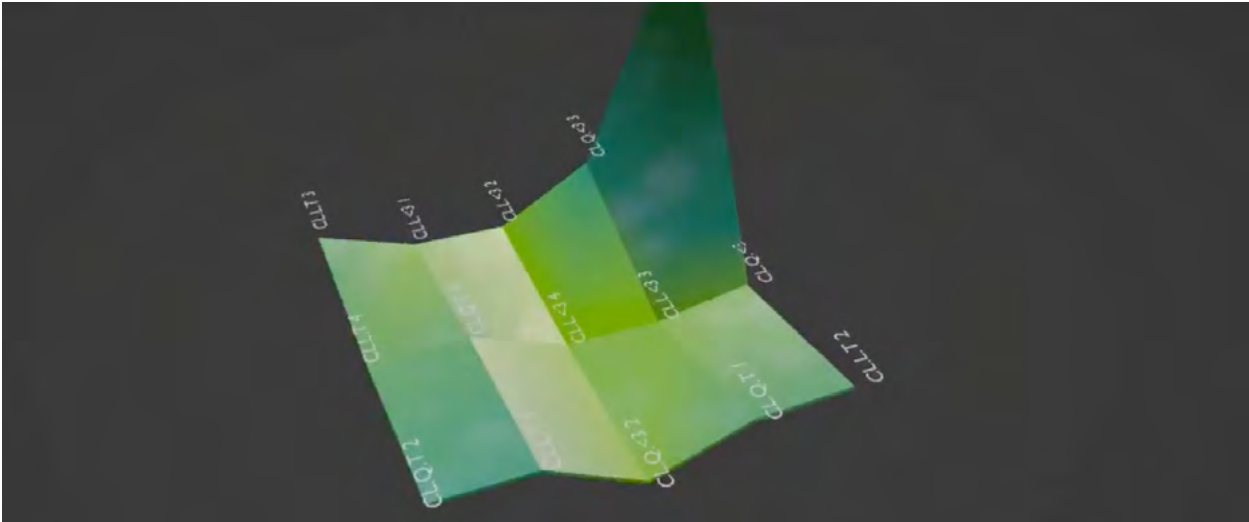
(Fig 7)

PROTOTIPO

Datos Utilizados: Cantidad de Clorofila en una muestra.
Requerimientos: Representar de manera gráfica, una base de datos entregada por el laboratorio de Microbiología Marina, enfocándose en dos requerimientos en los que el sistema debe visualizar la información:

- 1. Sin la necesidad de leer o intervenir el archivo.
- 2. De manera no convencional, es decir, interviniendo en las decisiones de diseño, como la morfología, colores o cualquier otra variable que implique una representación visual.

Uno de los principales desafíos consiste en desmarcarse del enfoque tradicional de las infografías, donde los diseñadores invierten tiempo en el filtrado, muestreo, agrupación y resumen manual de los datos. Esto se debe a que las herramientas utilizadas no permiten exponer grandes cantidades de información de manera automatizada. Es necesario superar esta limitación y adoptar un enfoque de visualización de datos automatizada. Como señala Tamara Munzner, una científica y experta reconocida en representación de datos, la visualización automatizada es esencial para manejar conjuntos de datos masivos, ya que permite que los usuarios se centren en explorar y analizar los datos en lugar de dedicar tiempo al procesamiento y la visualización manual (Munzner, 2014).



(Fig 8)

Por otro lado, los softwares existentes no permiten la variedad que un se requiere al visualizar datos (Cairo, 2012). Es por esto, que surge la necesidad de implementar una solución que considere el segundo aspecto como eje central.

Este ejercicio comienza con el procesamiento de datos (Fig 7), para la elaboración de un esquema relacional de los datos con sus nodos en Blender y Python API y luego su realización en un Render 3D. El gráfico representa la distribución de clorofila, según quince muestras del Laboratorio de Microbiología Marina UC. El eje Y (altura) de la figura, representa el nivel de clorofila, los textos en blanco, el nombre de la muestra (Fig 8).

Este primer prototipo demostró la posibilidad de crear una visualización de datos a partir de una cadena de caracteres, en este caso, la ruta a la base de datos. Sin embargo, se identificó un problema central: el usuario del proyecto no cuenta con las habilidades y herramientas necesarias para desarrollar una visualización como esta, ya que carece de conocimientos de programación y el software Blender está alejado de su campo de trabajo. Aunque es técnicamente factible, resulta poco probable que un científico realice este tipo de ejercicio. Esta situación planteó la siguiente pregunta:

¿Es posible que una persona sin conocimientos de programación y sin haber descargado Blender pueda realizar este proceso?

Tras realizar una investigación al respecto, se determinó que la respuesta es negativa. Ante este desafío, el proyecto se vio en la necesidad de elegir una dirección a seguir:

- 1. Proyecto de visualización de datos **no interactivo**: Consistiría en una presentación estática de la información, donde el usuario no tendría la capacidad de interactuar con los datos ni realizar cambios en la visualización.
- 2. Proyecto de visualización de datos **interactivo**: Permitiría al usuario manipular los datos, cambiar los parámetros de visualización, aplicar filtros y explorar diferentes aspectos de la información de manera interactiva.

Luego de un análisis exhaustivo de las etapas anteriores, se llegó a la determinación de optar por la segunda alternativa planteada. Esta elección implica emprender el desarrollo de una solución que permita al usuario interactuar y manipular los datos en la visualización de manera efectiva. En este contexto, se reconoce que la implementación de un software personalizado constituye la única opción viable para alcanzar una experiencia interactiva, de calidad y cumplir con los requisitos del proyecto de manera óptima.

Análisis detallado del Problema

La decisión de desarrollar un software surge del entendimiento de que las herramientas existentes y las alternativas no tecnológicas carecen de la capacidad necesaria para proporcionar la interactividad y la flexibilidad requeridas en la exploración y análisis de los datos. Por consiguiente, se concibe la creación de una interfaz intuitiva y amigable que permita al usuario interactuar de manera significativa con los datos, efectuar ajustes personalizados en la visualización y explorar diversos aspectos de la información con total libertad.

Con el propósito de alcanzar los estándares de calidad y competencia requeridos en el desarrollo del software, se ha tomado la decisión de dedicar un semestre de estudio y formación específica en el ámbito de la programación. Con un enfoque particular en asignaturas como Programación Avanzada e Ingeniería de Software, se busca adquirir las habilidades técnicas necesarias para abordar con éxito la implementación del software y garantizar un resultado final que satisfaga las expectativas del usuario.

Como resultado adicional de esta iteración, se reconoce el potencial de Blender en comparación con otras herramientas de trabajo para la representación de datos. Blender ofrece la capacidad de crear modelos complejos con texturas, materiales y efectos de iluminación avanzados, lo que permitiría una representación visual más completa y atractiva de los datos.

La Interactividad como Objetivo Proyectual

En este trabajo hemos identificado un problema central, que consiste en la deficiente capacidad de los softwares utilizados por la comunidad científica actualmente, para representar grandes cantidades de datos, de manera que se potencie el aprendizaje.

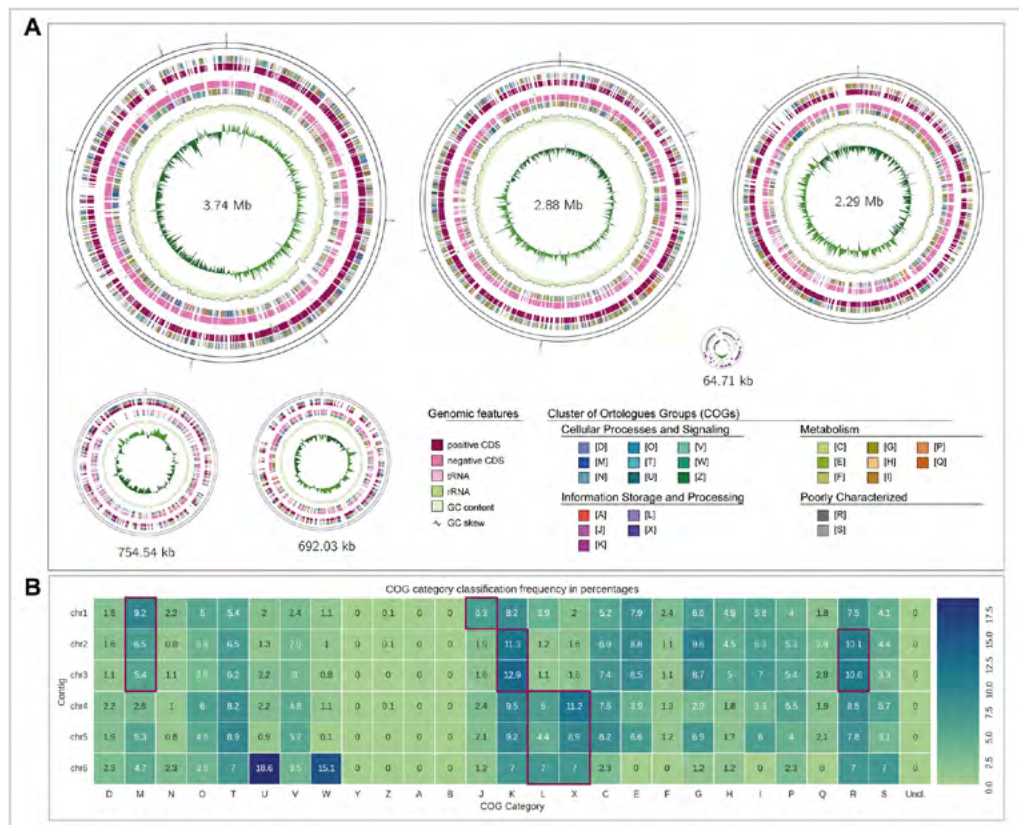
En 2011, Hullman y Diakopoulos propusieron un enfoque para analizar la visualización de datos que consistía en identificar cuatro capas editoriales: datos, representación visual, anotaciones e interactividad. La capa de datos se refiere a la selección y procesamiento de los datos que se utilizarán en la visualización. Cada una de estas capas desempeña un papel fundamental en la comunicación efectiva de la información.

La interactividad de la visualización puede ser el lugar donde se toman decisiones que limitan la interacción del usuario de manera que lo guían a explorar ciertos subconjuntos de datos. Estas decisiones de diseño interactivo pueden influir en cómo se presenta la información y en las rutas que el usuario puede seguir para explorar y comprender los datos. Al controlar las opciones disponibles y las formas en que se revela la información, se puede guiar al usuario hacia ciertos aspectos o perspectivas de los datos, enfatizando ciertos patrones o relaciones. Esto puede ayudar a transmitir de manera efectiva el mensaje deseado y facilitar la comprensión de los datos presentados, sobre todo si es que son bases de datos con grandes cantidades de información.

Los diseñadores deben decidir qué variables son relevantes y proporcionan la información necesaria para respaldar el objetivo de la visualización. Al elegir qué incluir y qué omitir,

es posible que se pierda cierta información, pero esto es necesario para simplificar y enfocar la visualización. La capa de representación visual lleva consigo huellas de las decisiones tomadas sobre cómo se mapearán los datos al dominio visual. Por ejemplo, mapear una variable continua a una escala de grises conlleva una “pérdida” de información debido a la sensibilidad y capacidad de percepción humana para distinguir diferentes niveles de intensidad (por ejemplo, “diferencias apenas perceptibles”).

Sin embargo, filtrar, entender y analizar los datos para una visualización de datos complejos, resulta una tarea compleja y fuera del marco profesional de un diseñador. Esto abre una brecha transdisciplinar que implica un desafío vez mayor para las personas que no están directamente relacionadas con la ciencia.



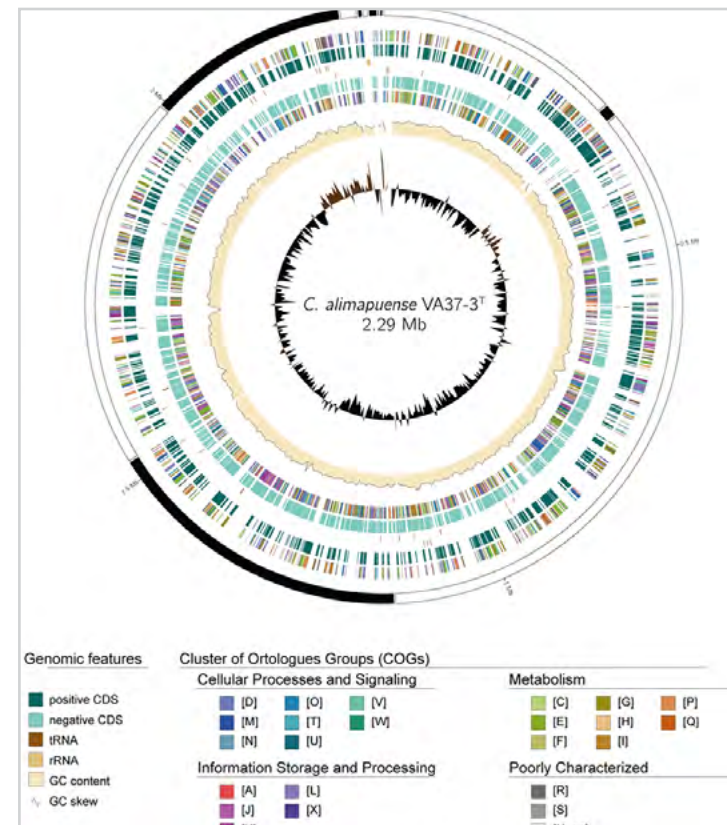
(Fig 9)

Debido a los desafíos, cada vez mayores, de entender, representar y filtrar los datos científicos, surge la necesidad de una propuesta que permita explorar los datasets, sin tener los conocimientos normalmente requeridos para producir gráficos. En el contexto del proyecto, el usuario no tiene que depurar los datos a utilizar, si no que la solución implementada, realiza este trabajo mediante algoritmos internos, lo que disminuye la pérdida de información y permite visualizar cantidades masivas de datasets. Esto permite manipular y optimizar la información a través de una interfaz, adaptando la intención del visualizador, de manera interactiva.

La interacción en la visualización de datos ha sido ampliamente estudiada y se ha demostrado que promueve el compromiso con los datos, fomenta la curiosidad y facilita la comprensión y el análisis (Heer & Shneiderman, 2012). Al

diseñar la interacción de manera experimental, es decir, que permita lo inesperado, se abre espacio para descubrimientos y nuevas perspectivas que enriquecen la comprensión de los datos (Dix, 2007), permitiendo a los usuarios interactuar con los gráficos y explorar diferentes aspectos de los datos de manera más profunda (Dix, 2007; Kosara, 2013).

GenoVi (Fig 9 y Fig 10) es un ejemplo destacado de visualización de datos complejos que utiliza recursos interactivos. Permite crear representaciones personalizadas de genomas circulares y visualizar la diversidad de especies, relaciones filogenéticas y abundancia relativa de microorganismos. También ofrece análisis de categorías COG, histogramas y mapas de calor. Su interfaz intuitiva y amigable la hace accesible a investigadores de diferentes niveles de experiencia. Los colores seleccionados cuidadosamente facilitan la identificación



(Fig 10)

rápida de especies y agrupaciones filogenéticas, así como la representación de la abundancia de genes. GenoVi es especialmente útil en publicaciones científicas, recursos educativos y divulgación científica, gracias a su capacidad para generar visualizaciones de alta calidad que comunican información de manera clara y precisa.

4. Marco Teórico

Epistemología de la Visualización

La visualización de datos ha demostrado ser una poderosa herramienta para la generación de conocimiento y comprensión de nuestro entorno (Keim et al., 2008; Weber & Rall, 2012; Bello-Orgaz et al., 2016). Su capacidad para humanizar y simplificar la complejidad de nuestro mundo es innegable. Como Florence Nightingale, pionera en la visualización de datos, afirmó: “La visualización nos permite capturar la esencia de los datos de manera más rápida y efectiva que cualquier otra forma de comunicación” (Nightingale, 1861).

En la actualidad, la visualización se considera fundamental en la investigación y en la toma de decisiones en diversos campos. Steven Kosslyn señala que “la visualización de datos nos permite detectar patrones, identificar tendencias y comprender relaciones complejas que de otra manera serían difíciles de apreciar” (Kosslyn, 2006).

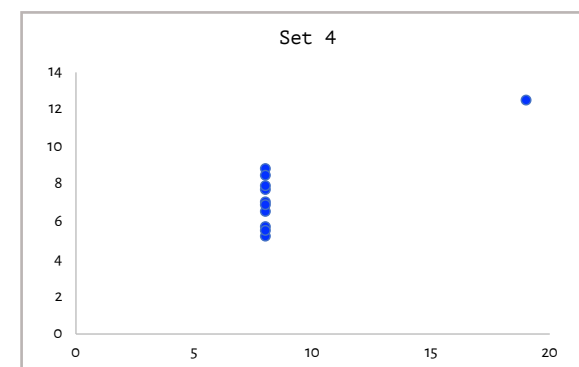
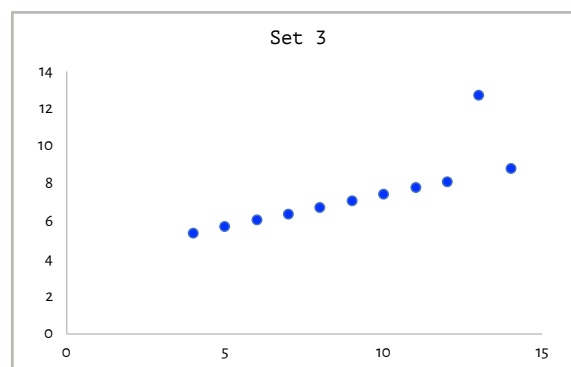
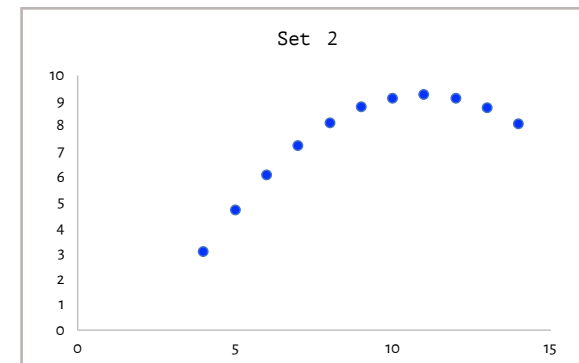
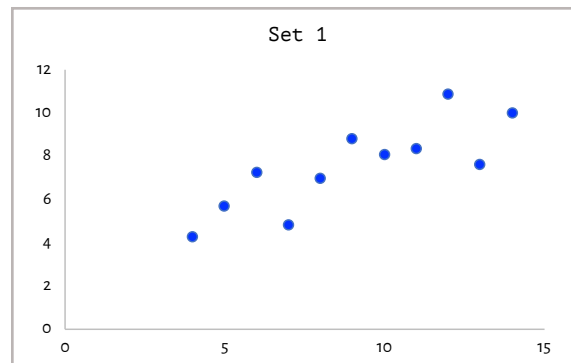
La visualización de datos es considerada la base de la epistemología científica, ya que permite una comprensión más rápida, efectiva y profunda de la información. Su papel en la generación de conocimiento y toma de decisiones informada ha sido reconocido por investigadores y profesionales en diversas disciplinas. Numerosos estudios argumentan que la comunicación visual es esencial para el proceso de adquisición de conocimiento, ya que la representación visual de información nos permite interpretar y comprender el mundo que nos rodea (Arnheim, 1954; Dondis, 1974; Ware, 2008).

**«La visualización nos permite
capturar la esencia de los datos
de manera más rápida y efectiva
que cualquier otra forma de
comunicación»**

	Set 1		Set 2		Set 3		Set 4	
	x	y	x	y	x	y	x	y
	10	8,04	10	9,14	10	7,46	8	6,58
	8	6,95	8	8,14	8	6,77	8	5,76
	13	7,58	13	8,74	13	12,74	8	7,71
	9	8,81	9	8,77	9	7,11	8	8,84
	11	8,33	11	9,26	11	7,81	8	8,47
	14	9,96	14	8,1	14	8,84	8	7,04
	6	7,24	6	6,13	6	6,08	8	5,25
	4	4,26	4	3,1	4	5,39	19	12,5
	12	10,84	12	9,13	12	8,15	8	5,56
	7	4,82	7	7,26	7	6,42	8	7,91
	5	5,68	5	4,74	5	5,73	8	6,89
Media	9,00	7,50	9,00	7,50	9,00	7,50	9,00	7,50
Varianza	11,00	4,13	11,00	4,13	11,00	4,12	11,00	4,12
PMCC	0,82		0,82		0,82		0,82	



(Fig 11)



Cooperación Humano-Computador

Las herramientas de visualización desempeñan un papel crucial al mostrar la estructura detallada de los conjuntos de datos. Esto es especialmente importante en situaciones en las que es necesario identificar patrones. Además, estas herramientas son útiles al evaluar la validez de un modelo estadístico, permitiendo determinar si realmente se ajusta a los datos.

En áreas como la investigación científica, donde el objetivo es generar y verificar hipótesis, las herramientas de visualización pueden acelerar y mejorar la capacidad de los usuarios para realizar análisis exploratorios. En lugar de depender únicamente de enfoques computacionales puramente automatizados, las herramientas de visualización permiten

a los científicos visualizar y comprender mejor conjuntos de datos complejos. Al proporcionar representaciones visuales interactivas y contextuales, estas herramientas ayudan a los investigadores a identificar patrones, tendencias y relaciones que podrían no ser evidentes en una vista numérica o textual (Munzer, 2014).

Un ejemplo destacado que ilustra la importancia de la participación humana en el análisis de datos, es el Cuarteto de Anscombe (Fig 11), que consiste en cuatro conjuntos de datos pequeños diseñados por un estadístico. A simple vista, estos conjuntos de datos comparten estadísticas descriptivas idénticas, como la media, la varianza, la correlación y las líneas de regresión lineal. Sin embargo, al representar gráficamente estos conjuntos de datos, se revela que tienen estructuras muy diferentes entre sí.

La inclusión de la computación en el procesamiento de datos permite superar las limitaciones de las representaciones manuales de conjuntos de datos. Mientras que las visualizaciones creadas a mano están restringidas a conjuntos de datos pequeños y estáticos debido a la capacidad de atención limitada de las personas, la computadora puede generar automáticamente representaciones visuales de datasets grandes y dinámicos. Esto ahorra un esfuerzo humano considerable, ya que organizar incluso datos pequeños podría llevar mucho tiempo. Al utilizar herramientas computarizadas, es posible explorar y presentar conjuntos de datos masivos de manera eficiente, abriendo nuevas posibilidades y descubrimientos.

	1																	18						
1	H																	He						
2	Li	Be																	B	C	N	O	F	Ne
3	Na	Mg																	Al	Si	P	S	Cl	Ar
4	K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr						
5	Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe						
6	Cs	Ba	La-Lu	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn						
7	Fr	Ra	Ac-Lr	Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Cn	Nh	Fl	Mc	Lv	Ts	Og						
LANTHANIDES			La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu							
ACTINIDES			Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr							
			NON METAL	ALKALI METAL	ALKALINE EARTH	TRANSITION METAL	BASIC METAL	METALLOID	HALOGEN	NOBLE GAS	LANTHANOID	ACTINOID												

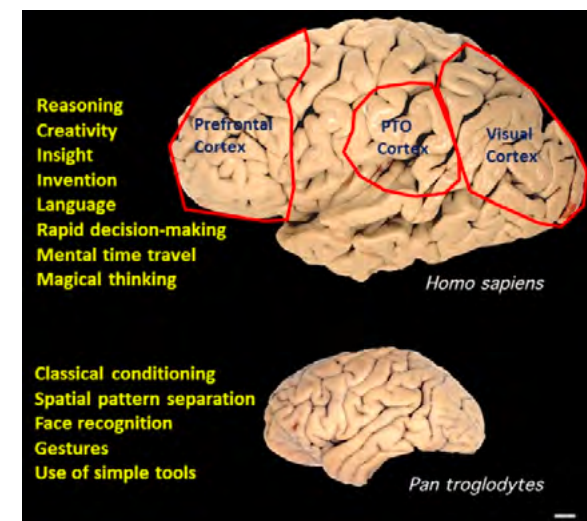
(Fig 12)

Reconocimiento de Patrones Visuales

Como se mencionó anteriormente, la información visual parece a menudo más fácil de asimilar que la verbal - piénsese, por ejemplo, en la distribución de los elementos químicos en la Tabla Periódica de Elementos producida por Mendeléyev (Fig 12). Los psicólogos plantean la hipótesis, de que esto se debe a nuestra historia evolutiva, dado que las habilidades verbales han aparecido mucho más tarde que las visuales (Evans, 2003, Sternberg y Leighton, 2004). Según Landau y Lakusta (2009): "Los mapas son el sistema simbólico por excelencia para codificar y retener permanentemente la información espacial. Los mapas son particularmente valiosos porque incluyen grandes cantidades de información sobre las relaciones

espaciales que no se pueden transmitir fácilmente a través del lenguaje (Mattson, 2014; Gonzalez-Redin et al., 2016).

Reconocer patrones visuales en el entorno es uno de los signos fundamentales del comportamiento inteligente. Un niño de 3 años, por ejemplo, puede discernir los alfabetos casi sin esfuerzo, un águila puede detectar a su presa desde 1000 pies sobre el suelo y un pez carnívoro puede capturar un pez más pequeño camuflado contra la arena. Cada uno de estos ejemplos ilustra la capacidad de percibir el orden desde el desorden, que no es solo una cuestión de habilidad visual sino también un signo de habilidad intelectual que es esencial para la supervivencia (Mattson, 2014; Rahman et al., 2019).

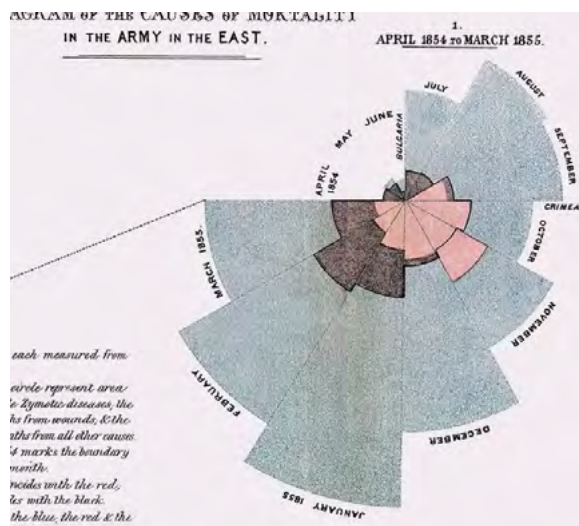


(Fig 13)

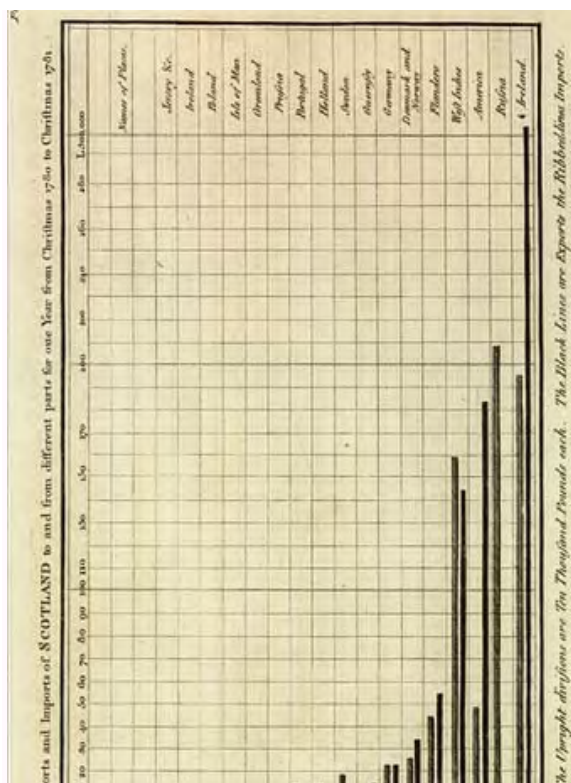
En el artículo de Mattson, se presenta una comparación de la anatomía macroscópica de los cerebros humanos y chimpancés (Pan troglodytes) revelando una expansión significativa de tres regiones en los cerebros humanos: la corteza prefrontal, la corteza visual y la unión parietal-temporal-occipital (PTO). Estas regiones expandidas son consideradas fundamentales para las capacidades avanzadas de procesamiento de patrones en los seres humanos (Fig 13).

Durante la evolución humana, las capacidades de procesamiento de patrones se volvieron cada vez más sofisticadas como resultado de la expansión de la corteza cerebral, particularmente la corteza prefrontal y las regiones involucradas en el procesamiento de imágenes (Mattson, 2014).

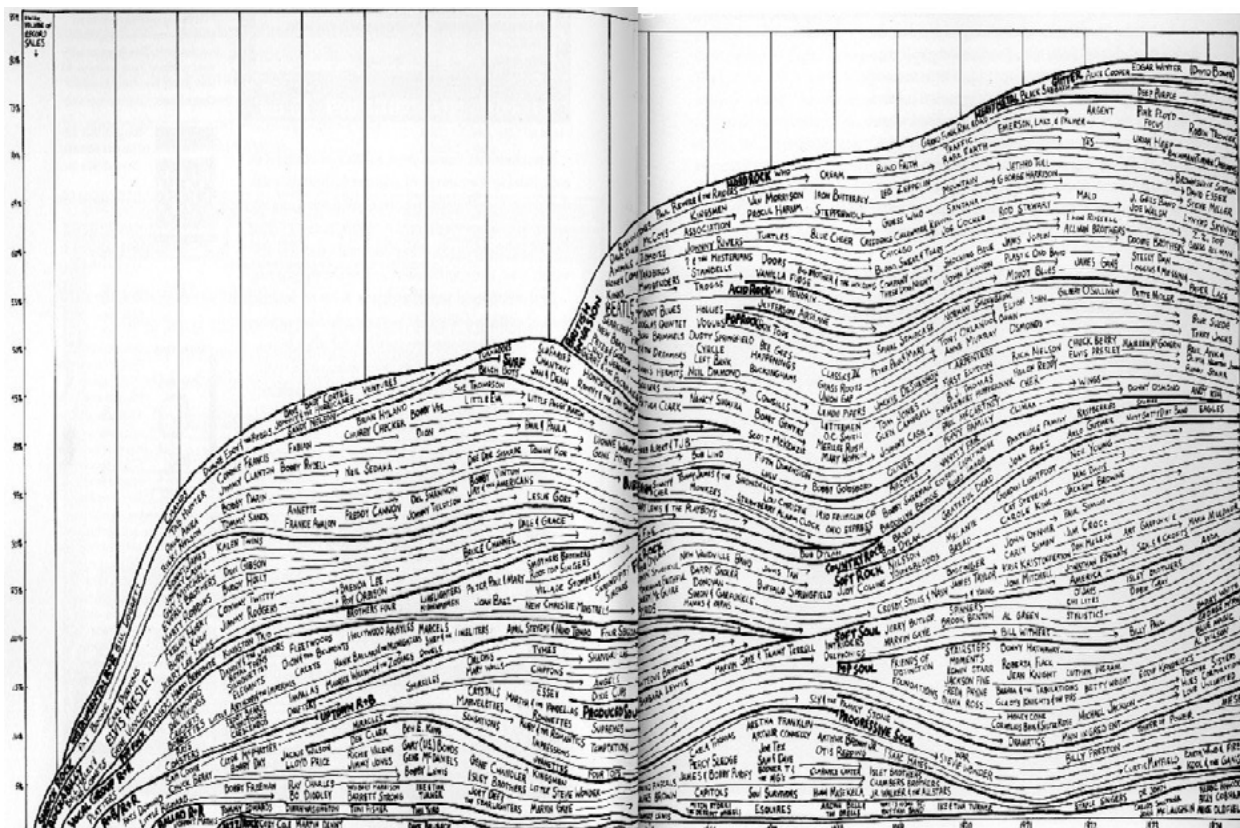
En este sentido, desde una perspectiva epistemológica, hemos desaprovechado una adaptación evolutiva, el reconocimiento visual de patrones, como medio para la identificación de relaciones de interés en los datos que generamos. Y si bien la premisa "correlación no implica causalidad" nos interpela a avanzar con precaución en la búsqueda de patrones entre fuentes distantes de información, se percibe de gran valor avanzar en plataformas y herramientas que nos faciliten transformar los datos en formas y colores con potencial como canales del conocimiento.



(Fig 14)



(Fig 15)



(Fig 16)

Visualización de Datos

Para algunos, las bases de datos pueden evocar ansiedad o dolores de cabeza. Para otros, equivalen a un comienzo para la creación artística. La visualización de información en el contexto gráfico ha sido desarrollada con mayor interés desde el aumento de los datos cuantitativos, como una herramienta para entenderlos. Playfair, W. (1759-1823) fue un referente en la invención de los gráficos de barra, línea y círculo. Según él, los gráficos permiten visualizar relaciones aparentemente inexistentes entre variables que suelen quedar ocultas entre los datos y cifras, difíciles de comparar de otra manera (Playfair, 1786). Nightingale, F. (1853-1856), en el contexto de la guerra de Crimea, inventó el gráfico Rosa para demostrar las deplorables condiciones sanitarias en los hospitales y el impacto en la mortalidad de los soldados. Según Nightingale, el gráfico

Rosa lograba transmitir visualmente lo que las palabras no podían comunicar a las mentes insensibles a través del oído (Nightingale, 1858).

El uso de gráficos para visualizar datos cuantitativos se hicieron más comunes con nuevos avances en el análisis estadístico (Ferster, 2013). Uno de los pioneros en este campo, fue un estadista norteamericano, Edward Tufte, más conocido como el Da Vinci de la Data (Times, 1998; Bamforth, 2011; Bentley, 2013; King, 2016). El mismo dice: "El uso de imágenes abstractas para mostrar números es una invención sorprendentemente reciente" y plantea que: "para razonar, explorar y resumir datos cuantitativos, la herramienta más sencilla y a la vez más poderosa es el diseño de visualización de datos" (Tufte, 1983).

En el ámbito de la Ciencia

La representación visual de datos es también la base del conocimiento científico colectivo (Weissgerber et al., 2015). Es por esto que muchos papers científicos incluyen recursos gráficos para facilitar la comprensión de lo que se está proponiendo. De hecho, algunas imágenes científicas han comenzado, en los últimos años, a ser apreciadas por su mérito estético, así como su relevancia para la investigación, y en muchas sucursales de la ciencia, ahora se otorgan premios a las imágenes más impactantes o bellas obtenidas a partir de datos experimentales, y se ha fundado una revista académica: the Art and Science Journal – para celebrar las imágenes que cierran la brecha entre estas dos esferas” (Roche et al., 2018).

Sin embargo, los estudios del Journal of the American Medical

Association y el British Medical Journal proporcionan pruebas convincentes de que se necesitan cambios fundamentales en los tipos de cifras que utilizan los científicos. Los autores generalmente usan cifras para presentar estadísticas resumidas, en lugar de proporcionar información detallada sobre la distribución de los datos o mostrar los datos completos (Cooper et al., 2002; Schriger et al., 2006).

Para el desarrollo de esta tesis, se utilizan los datos de el fondart bla bla



[Fig Challenger]

Problemas de la Representación de Datos

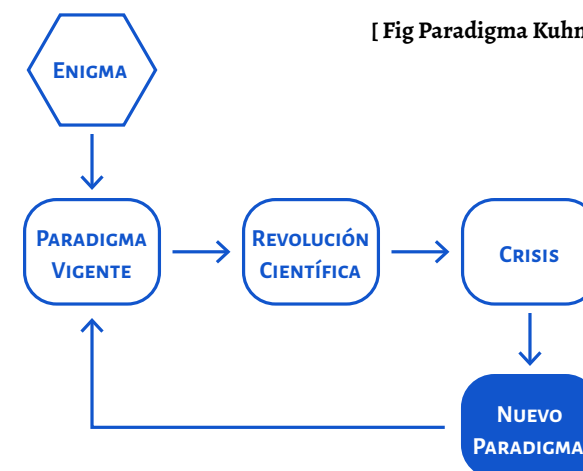
La representación de datos es un tema crítico en las ciencias exactas, ya que la comprensión y comunicación efectiva de los datos es esencial para la investigación y el avance científico. Sin embargo, la mayoría de los diseños de visualizaciones son ineficientes, debido a las complejidades y anomalías que pueden surgir en la presentación de los mismos, además de la amplia gama de posibilidades disponibles y la dificultad de encontrar soluciones adecuadas para contextos específicos. Cuando razonamos sobre evidencia cuantitativa, ciertos métodos para mostrar y analizar datos son mejores que otros. La diferencia entre un análisis excelente y uno defectuoso a veces puede tener consecuencias trascendentales. En lugar de optimizar, el objetivo del diseño de visualizaciones es satisfacer, es decir, encontrar una de las muchas soluciones viables. Es esencial considerar múltiples alternativas y explorar un amplio espacio de posibilidades. Además, el diseño de visualizaciones implica manejar compromisos, ya que un diseño puede tener fortalezas en ciertos aspectos pero debilidades en otros (Mattson, 2014).

Uno de los casos históricos más conocidos que ejemplifica un problema en la representación de datos es el desastre del transbordador espacial Challenger en 1986 (**Fig Challenger**). En este ejemplo, la mala visualización de datos fue uno de los factores que contribuyó al desastre, ya que los ingenieros de la NASA no lograron comunicar claramente el riesgo asociado a la temperatura baja mediante una representación gráfica insuficiente. El resultado fue la trágica pérdida de vidas y de la nave espacial. La investigación de Tufte sobre el desastre del Challenger ha sido ampliamente citada en la literatura sobre visualización de datos y ha llevado a una mayor conciencia de cómo la falta de atención a la representación de datos puede tener consecuencias graves y subraya la importancia de garantizar que los datos se presenten de manera clara y precisa en cualquier contexto.

Debido a las anomalías que pueden surgir en la representación de datos, es importante avanzar hacia una solución más interdisciplinaria en la que se aborde la representación de datos desde una perspectiva más amplia. Como señala Kuhn (1962) en su teoría de los paradigmas científicos, un nuevo paradigma puede proporcionar una perspectiva diferente y más amplia para la resolución de los problemas. En este sentido, es necesario estar abiertos a nuevas formas de representar y visualizar los datos, y a nuevas herramientas y tecnologías que puedan ayudarnos a superar los desafíos

actuales en la representación de datos en las ciencias. En este contexto, un diseñador que no es un científico de datos podría aportar un nuevo paradigma en términos de cómo representar y visualizar los datos. El enfoque del diseñador podría ser diferente al de un científico de datos, lo que podría llevar a nuevas formas de pensar sobre cómo presentar los datos de manera clara y efectiva. Además, la experiencia del diseñador en presentación visual y experiencia de usuario, podría ayudar a crear representaciones estéticamente enriquecidas de los datos que podrían mejorar la capacidad de las personas para comprenderlos.

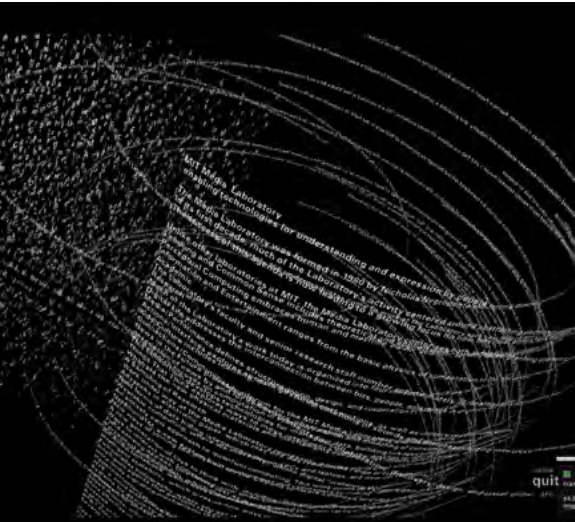
[Fig Paradigma Kuhn]



Nuevos Paradigmas

Con los avances de la tecnología, el diseñador ha tomado un rol activo en aprender, acercarse y usar las herramientas digitales para aprovechar sus singulares capacidades y características, y ampliar los límites de la expresión visual (Richardson, 2016). En el diseño gráfico, el computador creó un impacto rápido y masivo, artistas como Lorraine Wild y Emigre, se atrevieron a explorar este nuevo medio a pesar de las críticas del modernismo “hecho a mano” (Maeda & Antonelli, 2001). En medio de esta batalla, surge John Maeda, quien pudo demostrar que el computador es una herramienta poderosa, combinando el código informático con la estética del diseño como medio de expresión visual poderoso.

“Para lograr el objetivo de representar información de manera visualmente efectiva, es esencial comprender los diversos tipos de datos a representar”.



LI
HO

Tutorial

Ejemplo de Tabla

Esta es una tabla de ejemplo con datos generados aleatoriamente. Cada fila representa una muestra, y las variables son valores numéricos aleatorios entre 0 y 1 con una precisión de 3 decimales. Puedes personalizar esta tabla y sus datos según tus necesidades.

Nombre Muestra	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6
Muestra 1	0.438	0.165	0.743	0.990	0.895	0.943
Muestra 2	0.016	0.598	0.708	0.844	0.678	0.800
Muestra 3	0.878	0.095	0.092	0.756	0.530	0.260
Muestra 4	0.331	0.701	0.338	0.475	0.306	0.572
Muestra 5	0.849	0.513	0.027	0.887	0.371	0.058

GENERAR GRÁFICO

El aporte más importante de la obra de John Maeda, no es el resultado final, sino que su proceso. La idea fundamental de Maeda es que para diseñar con un computador, debemos crear, o por lo menos, entender el código que usamos (Maeda & Antonelli, 2001). Bajo esta misma línea, surgen referentes como Nathan Yau, quien realiza visualizaciones no sólo estéticamente interesantes, sino también incluye datos en tiempo real o interactividad con sus usuarios. En su libro “Flowing Data” dice: “Gana solo un poco de habilidades de programación, y puedes hacer mucho más con datos que si te quedaras solo con un software”. Las habilidades de programación nos dan la capacidad de ser más flexibles y capaces de adaptarse a diferentes tipos de datos (Yau, 2015).

Es crucial reconocer que los especialistas en visualización de datos concuerdan en que, para lograr el objetivo de represen-

tar información de manera visualmente efectiva, es esencial comprender los diversos tipos de datos a representar. Asimismo, surge un desafío importante al procesar bases de datos debido a sus variados formatos y extensiones. Por lo tanto, se destaca la relevancia de garantizar la compatibilidad entre las bases de datos, ya que esto facilita el acceso y la manipulación de los datos, permitiendo su integración en herramientas de visualización de manera más eficiente y efectiva.

Es importante señalar que los usuarios del presente proyecto, utilizan formatos generales como .csv o .xlsx. Para abordar este desafío, se ha implementado en el proyecto, una guía interactiva, que muestra un ejemplo de la base de datos a utilizar, junto con los diferentes tipos de datos requeridos como entrada. Esto ayuda a comprender cómo organizar los datos de manera adecuada para su visualización.

Tipos de Datos

BASE DE DATOS RELACIONAL:

Explicación: Una base de datos relacional es un sistema de gestión de datos que se basa en el modelo relacional. Este modelo organiza los datos en tablas, donde cada tabla representa una entidad o concepto particular. Las relaciones entre las entidades se establecen mediante claves primarias y claves externas.

Aplicación en el proyecto: En el proyecto, se utilizan bases de datos relacionales para almacenar y gestionar la información científica. Los datos se organizan en tablas con columnas que representan atributos específicos. Las relaciones entre las tablas permiten establecer conexiones y obtener información relacionada. Por ejemplo, cada fila representa una instancia del muestreo, y cada columna, una instancia de las variables a considerar.

DICCIONARIOS:

Explicación: En el contexto del proyecto, un diccionario es una estructura de datos que permite organizar y acceder a información de manera eficiente. Consiste en una colección de pares clave-valor, donde cada clave se asocia con un valor específico.

Aplicación en el proyecto: En el proyecto, los diccionarios se utilizan, entre otras cosas, para almacenar y enviar archivos GLB en forma de bytes. El diccionario organiza los archivos GLB, permitiendo acceder a ellos mediante una clave única. De esta manera, se facilita la gestión y envío de los archivos desde el servidor al cliente, generados por el proyecto.

BYTES

Explicación: Un byte es una unidad de medida que representa 8 bits de datos en sistemas informáticos. Cada byte puede contener información en forma binaria, como números, texto, imágenes u otros tipos de datos digitales.

Aplicación en el proyecto: Los bytes son utilizados en el proyecto de visualización de bases de datos científicas para representar y transferir información. Por ejemplo, los archivos GLB generados por el proyecto se envían en forma de bytes para su almacenamiento, transmisión y posterior visualización tridimensional. Los bytes permiten la representación compacta y eficiente de los datos digitales utilizados en el proyecto.

STRINGS (CADENAS DE TEXTO):

Explicación: Los strings son secuencias de caracteres que representan información en forma de texto descriptivo.

Aplicación en el proyecto: En el proyecto, los strings se utilizan para representar etiquetas, descripciones, nombres de objetos, títulos u otra información textual relevante. Por ejemplo, se utilizan para etiquetar los elementos de visualización como las variables u objetos, y así identificarlos por su nombre, lo cual es necesario por ejemplo para distinguir entre distintas variables o cuando el usuario quiera filtrar muestras.

NÚMEROS:

Explicación: Los números son valores numéricos utilizados en el análisis y cálculo científico.

Aplicación en el proyecto: En el proyecto, los números se utilizan para representar mediciones, magnitudes u otro tipo de datos cuantitativos relevantes. Por ejemplo, se pueden utilizar para representar valores de cantidad, que sean necesarios para el estudio o análisis científico.

DATOS GEOMÉTRICOS:

Explicación: Los datos geométricos se refieren a información relacionada con la geometría de un objeto, sus atributos espaciales y su posición en un sistema de coordenadas.

Aplicación en el proyecto: En el proyecto, los datos geométricos se utilizan para representar la forma, tamaño, posición y otros atributos espaciales de los objetos tridimensionales. Además, se utilizan los nodos para identificar los objetos padres e hijos que componen el árbol del archivo final. Estos datos geométricos se pueden almacenar y manipular en formatos específicos, como archivos GLB, para su posterior visualización y análisis en el proyecto.

Teoría de la Gestalt

La teoría de la Gestalt, basada en esta perspectiva más activa y creativa de la mente humana, revolucionó la psicología al postular que la percepción era el proceso inicial de la actividad mental y no un derivado cerebral de estados sensoriales (Köhler, 1929). Según Koffka (1935), la percepción era un proceso de extracción y selección de información relevante encargado de generar un estado de claridad y lucidez consciente que permitía el desempeño dentro del mayor grado de racionalidad y coherencia posibles con el mundo circundante. En este orden de ideas, el hecho de organizar los objetos a través de su forma equivale a tener de ellos una versión racional o en términos más exactos geométrica, que ofrece una idea clara, precisa e inconfundible de lo que la caracteriza.

Para demostrar la relación entre percepción y conceptualización de manera experimental, Koffka llevó a cabo un trabajo en el laboratorio que permitió descubrir las leyes de la percepción. Estas leyes confirmaron y reforzaron la teoría existente, lo que llevó a su amplia aceptación y a ser considerada uno de los enfoques más influyentes en la psicología de la percepción, teniendo una gran influencia en la teoría visual y el diseño gráfico (Albers, 1963; Rand, 1985; Tufte, 1990; Lupton, 2004; Norman, 2013). En relación con el proyecto de Título, algunas de las leyes de percepción relevantes son:

La ley de la semejanza: los elementos que comparten características similares, como el color, la forma o el tamaño, tienden a ser agrupados juntos.

La ley de la continuidad: los elementos dispuestos en una línea o curva continua tienden a ser percibidos como una unidad.

La ley de la simetría: los elementos que son simétricos tienden a ser percibidos como una unidad coherente.

La ley del cierre: nuestro cerebro tiende a completar patrones y formas incompletas para formar una imagen completa y coherente.

Al aplicar estos principios en visualización de datos, podemos crear visualizaciones que sean fáciles de entender y que permitan a los usuarios identificar patrones y tendencias de manera eficiente (Yau, 2014).

Graficar la misma data de antes, pero haciendo que las columnas de especies que pertenecen a mismos taxa, tengan colores similares. Eso y otras cosas que se te ocurran.

La ley de la proximidad: los elementos que están cerca entre sí tienden a ser agrupados juntos y percibidos como una unidad.



Visualización 3D

La visualización en 3D ofrece una serie de ventajas significativas en comparación con las representaciones en 2D. En primer lugar, la naturaleza inmersiva e intuitiva de las visualizaciones en 3D mejora la percepción y la comprensión de datos complejos al proporcionar una experiencia más envolvente (Chen, 2015). Al permitir que los usuarios interactúen con los datos desde diferentes ángulos y perspectivas, la visualización en 3D facilita la identificación de patrones espaciales y la comprensión de la distribución espacial de los datos (Wittek et al., 2014).

Otra ventaja clave de la visualización en 3D es la capacidad de exploración interactiva. Los usuarios pueden manipular y explorar los datos en tiempo real, lo que les permite descubrir

información relevante de manera más eficiente (Ljubic et al., 2017). Mediante acciones como el zoom, la rotación y la manipulación directa de los elementos visuales, los usuarios pueden examinar los datos en detalle y obtener una comprensión más profunda de las relaciones entre los puntos de datos.

Además, la visualización en 3D es especialmente adecuada para representar datos que requieren una representación realista. En campos como la arquitectura y las simulaciones científicas, la tercera dimensión permite una representación más precisa y detallada de la estructura y las características físicas de los objetos (Frohlich et al., 2019). Esto proporciona una experiencia visual más fiel y realista, lo que resulta en una comprensión más profunda de los datos.

Por último, la visualización en 3D es altamente efectiva cuando se trata de datos multidimensionales. Al asignar variables a diferentes dimensiones espaciales, se pueden representar y analizar simultáneamente múltiples atributos de los datos (Chen et al., 2019). Esto permite una comprensión más completa de las relaciones complejas entre las variables y facilita la detección de patrones y tendencias que podrían pasar desapercibidos en representaciones 2D.

8 Estado del Arte

Se requieren conocimientos en Programación?



Si



No

Antecedentes

Durante el análisis de los antecedentes recopilados, se han identificado diversas soluciones que son ampliamente utilizadas por la comunidad científica. Sin embargo, es importante tener en cuenta que cada una de estas soluciones presenta ciertas limitaciones y que su adopción por parte de los usuarios puede variar.



RAWGraphs:

Es un software de visualización de datos de código abierto que permite crear visualizaciones personalizadas a partir de datos tabulares. Ofrece una amplia variedad de opciones de visualización.

No se requieren conocimientos en programación.

SOFTWARE DE VISUALIZACIÓN DE DATOS INTERACTIVOS Y WEB:



Plotly:

Plotly es una poderosa herramienta interactiva de visualización de datos que facilita la creación de gráficos y visualizaciones personalizadas con calidad de publicación. Su enfoque en la interactividad y la capacidad de personalización lo convierten en una opción popular para aquellos con conocimientos de programación y análisis de datos. Es importante tener en cuenta que Plotly ofrece tanto una versión gratuita como una versión de pago con características adicionales, lo que implica que algunas funcionalidades pueden estar limitadas a usuarios que opten por suscripciones de pago. Se requieren conocimientos en programación en Python.



D3.js:

Es una biblioteca de JavaScript que permite crear visualizaciones personalizadas y dinámicas en la web. Una de sus principales fortalezas radica en su capacidad de brindar un control completo sobre los elementos visuales y las interacciones.

Se requieren conocimientos avanzados en programación en Javascript

SOFTWARE DE VISUALIZACIÓN Y ANÁLISIS DE DATOS CIENTÍFICOS:



Origin:

Es un software especializado en la visualización y análisis de datos científicos. Ofrece herramientas para crear gráficos 2D y 3D, realizar ajustes de curvas y realizar análisis estadísticos. No se requieren conocimientos en programación.



Matplotlib:

Es una biblioteca de Python para la creación de gráficos y visualizaciones científicas. Ofrece una amplia gama de opciones de visualización y personalización. Se requieren conocimientos en programación en Python.

SOFTWARE DE VISUALIZACIÓN DE DATOS ESPECÍFICOS:



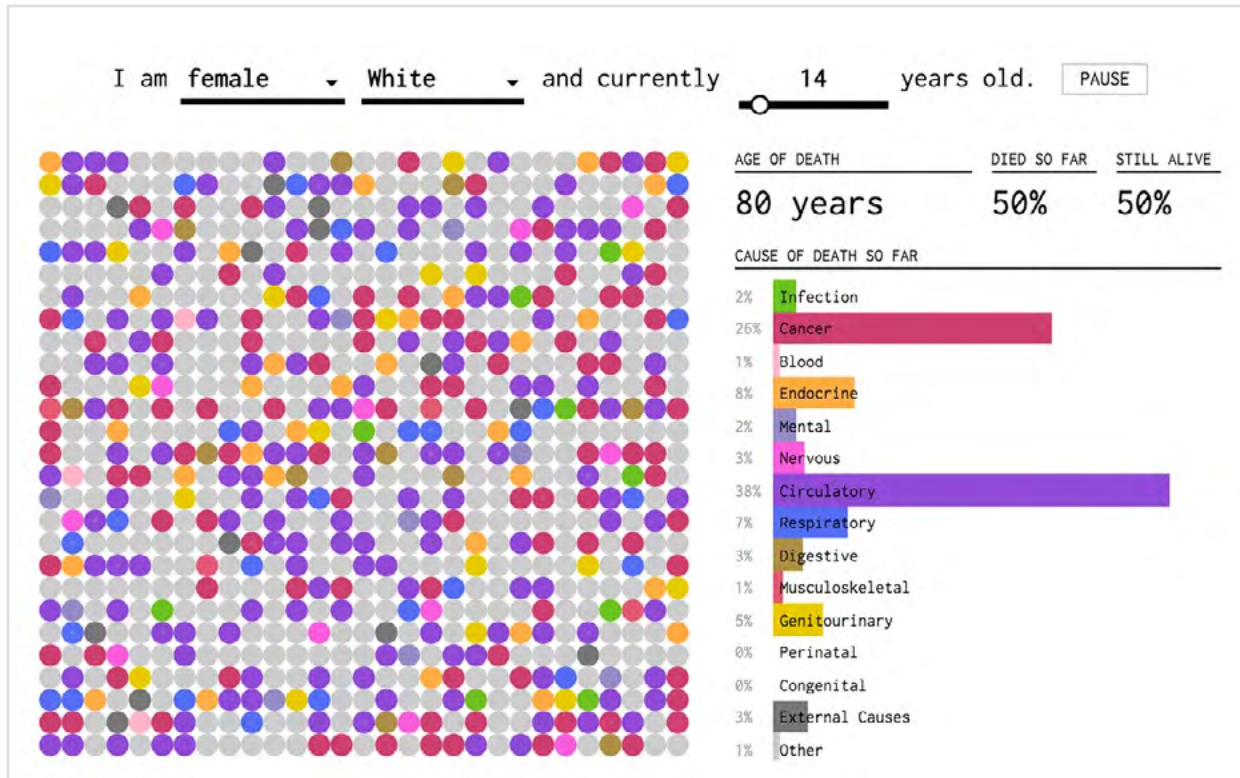
Logly:

Es un software especializado en la visualización de datos de registro (logs) y análisis de registros en tiempo real. Permite monitorear y analizar grandes volúmenes de datos de registro. Se requieren conocimientos básicos en programación.

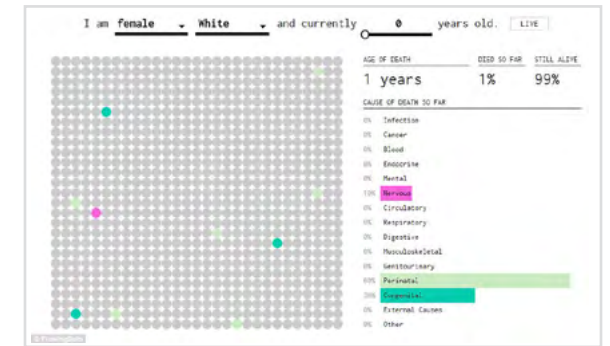


FlowJo:

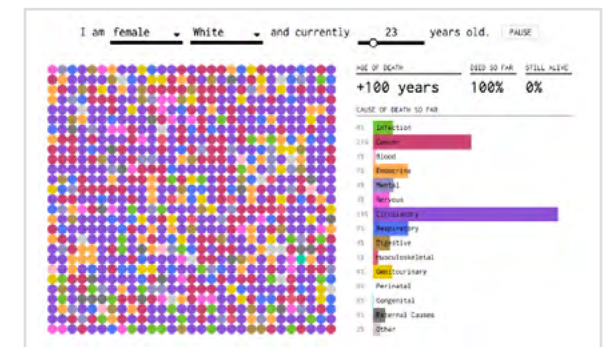
Es un software utilizado en el campo de la citometría de flujo para el análisis y visualización de datos. Permite la exploración y el análisis de datos de citometría de flujo de forma interactiva. Los archivos producidos por los citómetros de flujo modernos se escriben en el formato estándar de citometría de flujo con una extensión de archivo .fcs. No se requieren conocimientos en programación.



[Fig yau]



[Fig yau]



[Fig yau]

Referentes

Nathan Yau

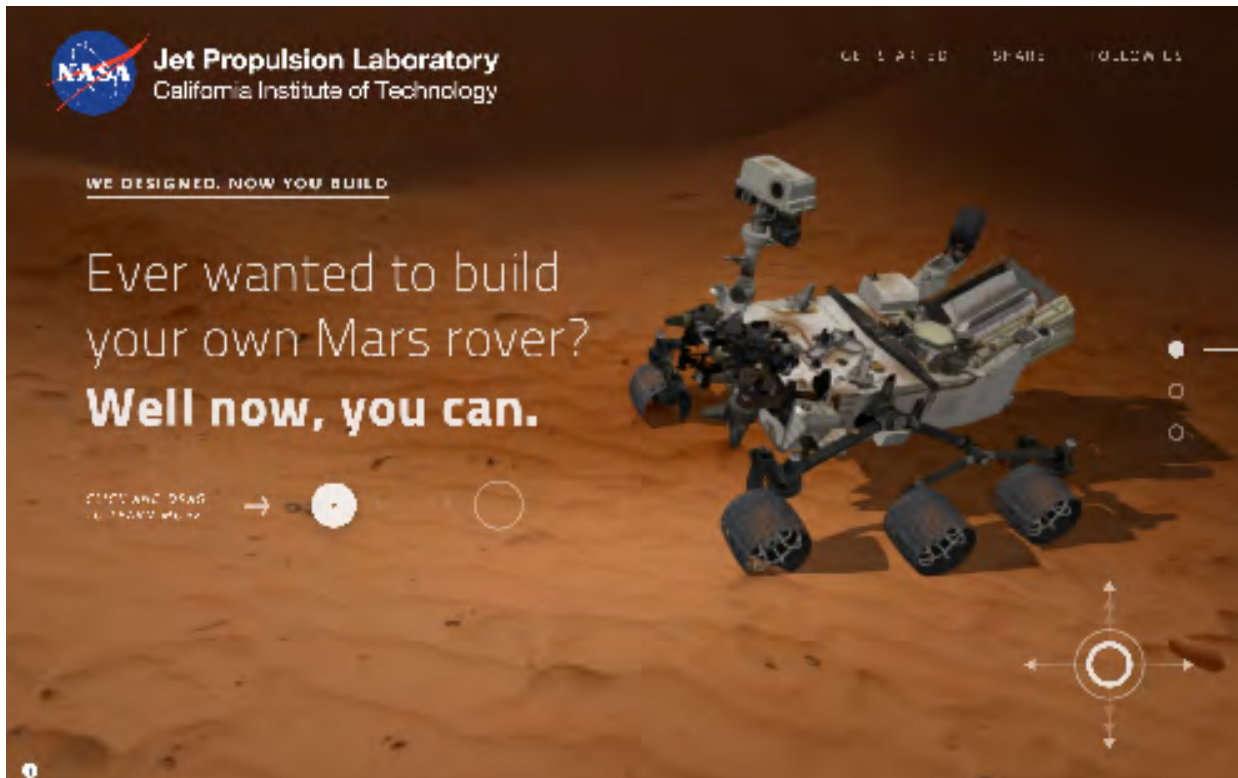
Nathan Yau es un reconocido estadístico, autor y visualizador de datos estadounidense. Ha escrito varios libros sobre visualización de datos, como "Data Points: Visualization That Means Something" y "Visualize This: The FlowingData Guide to Design, Visualization, and Statistics". Es conocido por ser el creador del popular sitio web FlowingData.com, donde comparte sus proyectos y trabajos.

Uno de sus proyectos destacados es "How You Will Die" (figura 1), una visualización de datos que muestra la cantidad de personas fallecidas en Estados Unidos entre 1999 y 2014, junto con las causas de muerte. Lo interesante de este proyecto es que utiliza la interactividad como una forma de resumir visualmente la información. Utiliza dropdown menús que permiten filtrar opciones y explorar los datos de manera más específica y personalizada.

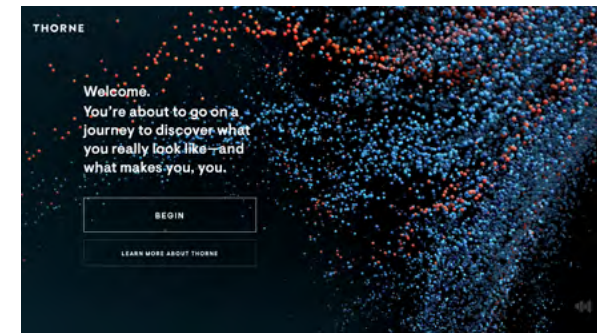
A través de su trabajo, Yau demuestra su habilidad para utilizar la visualización de datos como una herramienta efectiva para comunicar información compleja de manera clara y accesible. Su enfoque en la interactividad y la personalización permite a los usuarios interactuar con los datos y descubrir patrones y tendencias de una manera más significativa.

Para interactuar con la plataforma, se puede acceder mediante este [enlace](#).

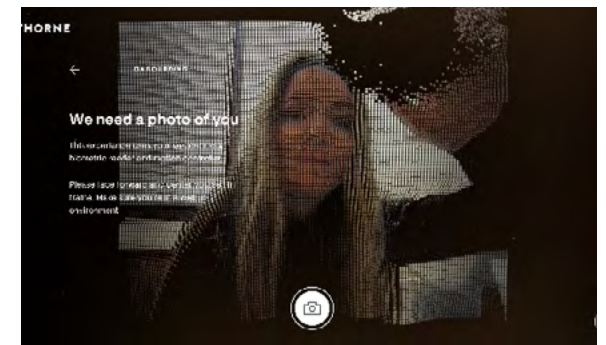
En el presente proyecto, se ha tomado como fuente de inspiración la aplicación desarrollada por Nathan Yau en relación al uso del color y la interactividad. Su enfoque ha sido una referencia clave para crear una experiencia similar, poniendo énfasis en la utilización efectiva del color y la implementación de interactividad con el objetivo de mejorar la exploración y comprensión de los datos. La visión y el trabajo de Nathan Yau han sido una influencia significativa en el enfoque de diseño y funcionalidad adoptado en este proyecto.



[Fig yau]



[Fig yau]



[Fig yau]

Active Theory

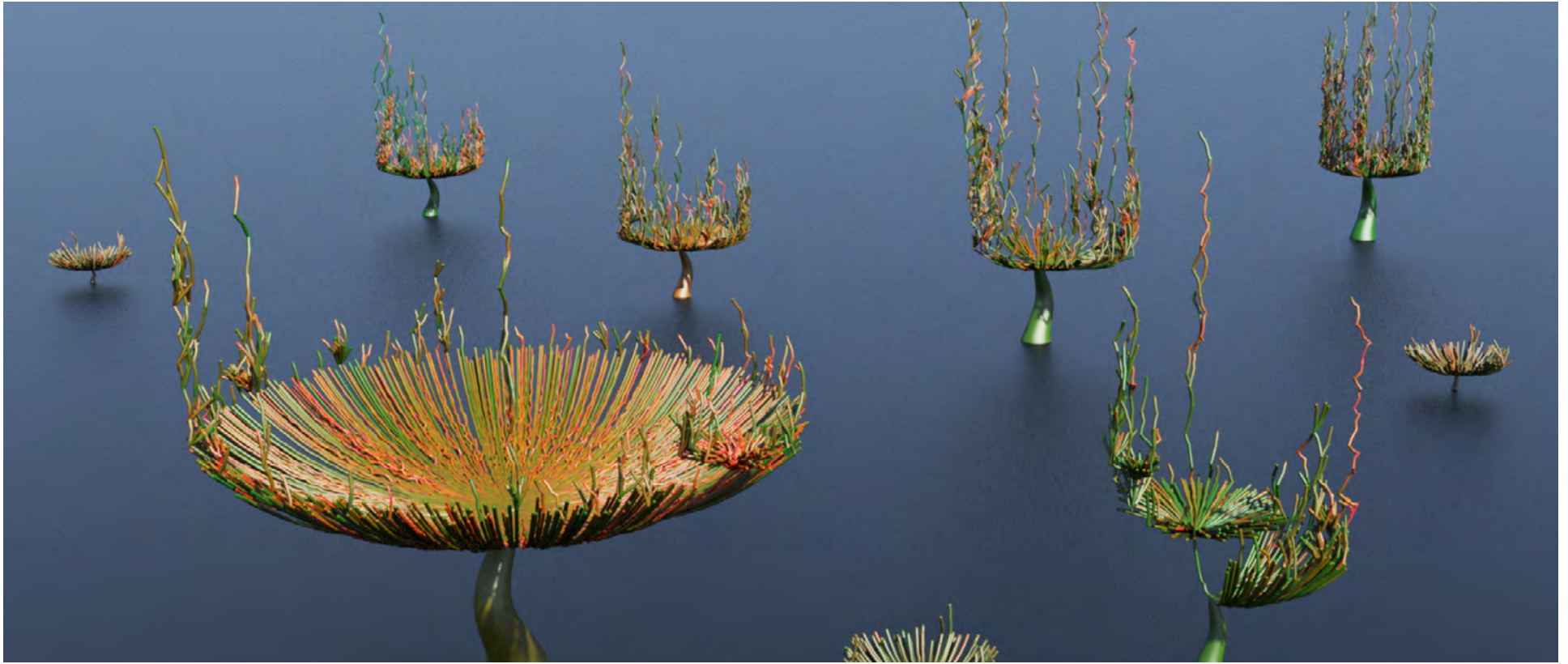
Active Theory es una agencia de diseño y desarrollo digital, especializada en la creación de experiencias interactivas, interfaces y contenido visualmente enriquecido. Su enfoque se centra en la combinación de tecnología, diseño y creatividad para proporcionar experiencias digitales inmersivas y memorables.

Una de las fortalezas distintivas de Active Theory radica en su expertise en el manejo de modelos 3D, lo que les ha permitido crear interfaces innovadoras que establecen conexiones significativas con los usuarios. Su trabajo abarca una amplia gama de proyectos, desde sitios web y aplicaciones móviles hasta instalaciones interactivas y experiencias de realidad virtual y aumentada. Su enfoque innovador los ha posicionado como

líderes en el campo del diseño de experiencias inmersivas.

El trabajo de Active Theory abarca una amplia gama de proyectos, desde sitios web y aplicaciones móviles hasta instalaciones interactivas. Su enfoque se basa en la combinación de narrativas cautivadoras, animaciones fluidas y usabilidad intuitiva para lograr experiencias de usuario envolventes.

Entre los clientes destacados de Active Theory se encuentran marcas como Google, Nike, Facebook, Airbnb, Red Bull, Sony, Disney y BMW, entre otros. Su enfoque innovador y su capacidad para crear experiencias interactivas y visualmente impresionantes los han convertido en un socio confiable y buscado para proyectos digitales de alto nivel. Se puede conocer más sobre el proyecto mediante este [enlace](#).



Peter Beshai

Peter Beshai es un investigador y desarrollador, creador de un proyecto enfocado en la visualización de la toxicidad en las conversaciones de Twitter. Se destacan sus técnicas de manejo de texturas y nodos para representar de manera visual la presencia de comportamientos tóxicos en los intercambios de mensajes en esta plataforma.

Beshai ha desarrollado métodos innovadores para analizar y visualizar grandes conjuntos de datos provenientes de conversaciones en Twitter. Utilizando técnicas de procesamiento de lenguaje natural y algoritmos de aprendizaje automático, es capaz de identificar patrones de toxicidad y comportamiento inapropiado en los mensajes.

Su manejo de texturas, consiste en asignar características visuales a los diferentes elementos de las conversaciones con el

fin de resaltar la toxicidad. Esto puede incluir el uso de colores intensos, tamaños de texto o efectos visuales específicos para resaltar palabras o frases ofensivas.

Además, Beshai utiliza nodos y enlaces para representar las interacciones entre los usuarios y las conversaciones en Twitter. Mediante la creación de visualizaciones de redes, es capaz de mostrar de manera intuitiva cómo se propagan los mensajes tóxicos y cómo afectan a la comunidad en general.

Patrón de Valor

En el desarrollo del curso BIO346D, Desafíos de las Ciencias Biológicas y el Diseño de mi profesor guía, se constata el amplio interés por parte de las y los estudiantes en profundizar sus habilidades en herramientas de dibujo vectorial (e.g. Adobe Illustrator) para mejorar las visualizaciones que generan desde herramientas de análisis de Datos (e.g. R-Studio).
Durán-Vargas, A. (2023) Comunicación personal.

Esta interacción crítica, denota la necesidad del usuario, de herramientas para desarrollar gráficos de mejor calidad que los antecedentes propuestos.

El caso de estudio serán los seres vivos que habitan en el agua, ya que presenta un marco interesante, debido a que sus escalas no son comprensibles meramente a través del ojo humano. En este sentido, el proyecto facilita el entendimiento de fenómenos que están fuera de nuestro campo visual, sin dejar de lado nuestra capacidad evolutiva de descubrir patrones a partir de imágenes.

Liho es una herramienta de visualización de datos que permite visualizar segmentos desde la tridimensionalidad poligonal, permitiendo desplegar y aplicar herramientas de geometría analítica y así descubrir patrones que no son evidentes. Algunas de estas herramientas son Substracción, Intersección

Alcances de liho es la parte importante, visualizar y comunicar. Pero lo que se logra es que la información cuantitativa sea repr de manera geometrica. Desplegar, comunicar, aplicar a este, las operaciones geométricas que conocemos del mundo tridimensional.

Alcances de liho es la parte importante, visualizar y comunicar. Pero lo que se logra es que la información cuantitativa sea repr de manera geometrica. Desplegar, comunicar, aplicar a este, las operaciones geométricas que conocemos del mundo tridimensional.

9. Contexto y Usuario

Investigador Científico de Universidades Complejas.

El usuario corresponde a un científico que investiga en una universidad compleja, la cual corresponde a un tipo de universidad que se distingue de las casas de estudios superiores, ya que además de formar profesionales y de las actividades en aulas, generan conocimiento científico a través de la investigación.

Son personas altamente capacitadas y dedicadas a la investigación en una determinada área del conocimiento, que trabajan en una universidad reconocida por su excelencia académica y científica.

Este investigador puede tener diferentes roles y responsabilidades dentro de la universidad, dependiendo de su nivel de experiencia y de la naturaleza de su investigación. Por lo general, los científicos investigadores en universidades complejas suelen tener un título de doctorado y haber llevado a cabo investigaciones de alto nivel en su campo.

Entre las responsabilidades típicas de un científico investigador en una universidad compleja se encuentran la realización de investigaciones originales, la publicación de sus resultados en revistas científicas de prestigio, la obtención de fondos para su investigación, la colaboración con otros científicos e instituciones, y la supervisión de estudiantes y postdoctorados que trabajan en su laboratorio.

Además, los científicos investigadores en universidades complejas suelen estar involucrados en actividades de enseñanza y divulgación científica, como la impartición de clases, la organización de seminarios y la participación en conferencias y otros eventos científicos. En general, se espera que los científicos investigadores en universidades complejas sean líderes en

su campo y contribuyan de manera significativa al avance del conocimiento científico.

La investigación es uno de los pilares de las universidades complejas como la UC. Ha habido una importante evolución de la política de investigación de la universidad y de su institucionalidad. Este progreso se ha expresado en una consolidación y expansión sistemática de la actividad investigativa, de innovación y transferencia que ha respondido adecuadamente al creciente aumento de su complejidad, y al posicionamiento de la Universidad Católica como un líder nacional y regional en la investigación de frontera. (UC, 2018)

10. Formulación/Objetivos

Open Web Tool

Qué

Plataforma open-source online de visualización tridimensional multivariable de bases de datos científicos con potencial de correlación por medio de la detección de patrones o representación de sus descubrimientos desde la manipulación, discriminación y análisis de la información.

Por qué

La complejidad y abundancia de fuentes de información y bases de datos provenientes de investigaciones científicas, supone una oportunidad disciplinar que emerge desde una integración exploratoria utilizando estos datos como materia prima.

Para qué

Vincular visualmente fuentes de información divergentes o complementarias desde herramientas digitales de código abierto que promueva una forma de indagación basada en las capacidades visuales de las personas y la comunidad científica.

Objetivos

Objetivo general

Elaborar una plataforma online, que a través de la visualización tridimensional y multivariable de datos científicos, se promueva una forma de indagación por medio de la detección de patrones.

Objetivos específicos

Permitir la adaptación de Bases de Datos provenientes múltiples fuentes y formatos en un estándar con potencialidad de uso.

I.O.V: Comparación de estos campos y formatos para su uso en las distintas dependencias o librerías que el proyecto utilizaría.

Implementar un módulo digital capaz de concatenar estos datos para luego generar formas tridimensionales a partir de ellos.

I.O.V: Ejecutar software multiplataforma y poder visualizar datos.
(todo sobre el software)

I.O.V: Carga de múltiples bases de datos en módulo digital.

Permitir la variabilidad en la representación y configuración de las formas al procesar datos.

I.O.V: Integrar una librería de clases que permita cambios morfológicos.

Ejecutar desde el navegador con todos sus componentes.

I.O.V: Integrar una librería de clases que permita cambios morfológicos.

Permitir la descarga del resultado final del procesamiento del sistema, tanto como geometría así como imagen final

I.O.V:

Propósito Proyectual: Permitir la generación de conocimiento original por medio de la utilización citada de estas visual-

izaciones.

* Motivación final del Proyecto, difícil de verificar.

Permitir la adaptación de Bases de Datos provenientes múltiples fuentes y formatos en un estándar con potencialidad de uso.

11. Proceso de desarrollo

FRONTEND

BACKEND

Requisitos del Software

Requisitos Generales

Durante la fase de investigación de soluciones existentes, se analizaron diversas tecnologías y herramientas para determinar las opciones más adecuadas para el proyecto. Como ya se mencionó anteriormente, el proyecto reconoce la potencialidad de Blender, Uno de los desafíos identificados fue la limitación de los navegadores web para ejecutar código Python de forma nativa, lo que impedía utilizar la biblioteca Blender Python Api (bpy) directamente en la web. Para superar esta limitación, se llegó a la conclusión de que se requerían agentes adicionales, específicamente un **frontend** y un **backend**, para gestionar la ejecución de bpy y ofrecer una experiencia interactiva al usuario.

Para una mejor comprensión del proyecto, es crucial tener una breve referencia de los **cuatro frameworks** principales que lo componen, ya que se mencionan constantemente a lo largo de su desarrollo. A continuación, se presenta una explicación breve pero precisa de cada uno de ellos, lo cual contribuirá a una comprensión más profunda.

El frontend se refiere a la parte de la aplicación que interactúa directamente con el usuario. Es la interfaz a través de la cual el usuario puede visualizar y manipular los datos. En el caso de este proyecto, el frontend sería la interfaz web que permite al usuario interactuar con la visualización de datos 3D basada en bpy. Esto implica el desarrollo de componentes como la interfaz gráfica, los controles interactivos y la presentación visual de los datos.

REACT

React es una biblioteca de JavaScript utilizada para construir interfaces de usuario interactivas. Se centra en la creación de componentes reutilizables y actualiza eficientemente la interfaz de usuario cuando el estado de la aplicación cambia. Es ampliamente utilizado en el desarrollo de aplicaciones web modernas.

THREE.JS

Three.js es una biblioteca de JavaScript que permite crear y renderizar gráficos en 3D en el navegador web. Permite agregar objetos, aplicar materiales y texturas, animar objetos y trabajar con luces y sombras. Es una herramienta popular para desarrollar experiencias visuales 3D interactivas en la web.

Por otro lado, el backend se encarga de gestionar y procesar los datos, así como de interactuar con la biblioteca bpy. Es la parte de la aplicación que se ejecuta en el servidor y se encarga de recibir las solicitudes del frontend, procesar los datos correspondientes y devolver los resultados al frontend para su visualización. El backend también se encarga de la lógica del negocio, la autenticación y la persistencia de datos.

DJANGO

Django es un framework de desarrollo web de alto nivel y de código abierto escrito en Python. Proporciona una estructura organizada para construir aplicaciones web, incluyendo manejo de bases de datos, enrutamiento de URL y creación de Modelos.

BLENDER PYTHON API (BPY)

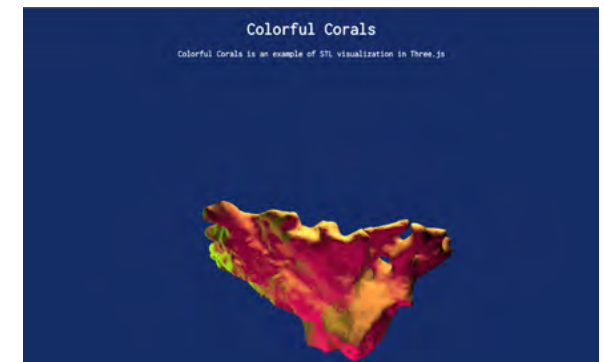
Blender es un software de creación de gráficos 3D de código abierto. Aunque no es un framework en sí mismo, se menciona porque se utiliza junto con la API de Python (bpy) para manipular y controlar Blender programáticamente. bpy permite automatizar tareas, crear complementos personalizados y extender las capacidades de Blender.



[Fig color 1]



[Fig color 2]



[Fig color 3]

Requisitos Frontend

Para solventar la incompatibilidad de bpy con la Web, se optó por exportar el modelo 3D, lo que posibilita su carga y visualización en tiempo real mediante el empleo de un renderizador adecuado.

Para mostrar esto en un navegador web, se evaluó, el renderizador Three.js, el cual fue sometido a un ejercicio de prueba de viabilidad.

Es por esto, que se desarrolla la plataforma Colorful Corals con el objetivo de comprobar, investigar y aprender sobre la inserción de modelos 3D en la web, así como su iluminación, escala y renderizado.

La figura actualmente mostrada representa un modelo tridi-

mensional de la base de una especie de alga parda llamada *Macrocystis pyrifera*. Este modelo fue creado mediante fotogrametría por el profesor Alejandro Durán.

Durante la evaluación de opciones para el desarrollo del frontend de la plataforma, se consideraron distintas tecnologías como Angular, Vue y React. Para esta plataforma se utiliza Vue. Sin embargo, se decidió utilizar React como framework principal.

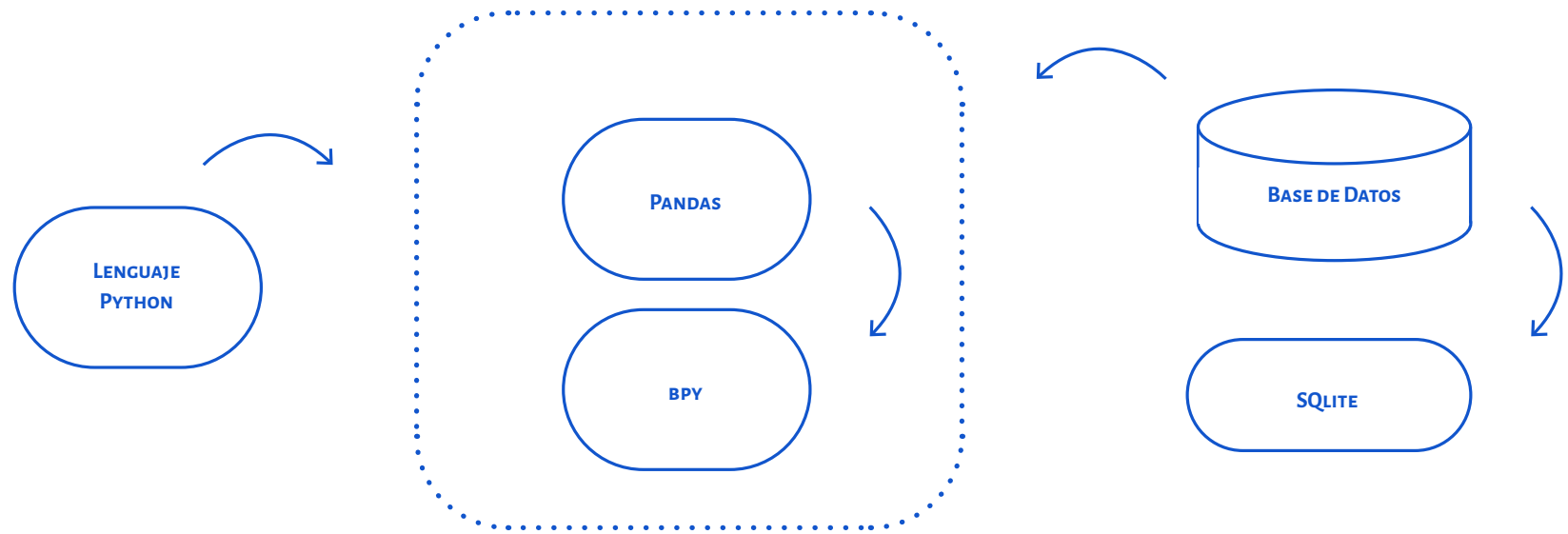
ELECCIÓN DE REACT

La elección de React por sobre Vue se basó en su capacidad para manejar grandes volúmenes de datos de manera eficiente. Al integrarse con Three.js, se logra una experiencia de usuario fluida que permite la interacción dinámica con los gráficos y los datos. Esto resulta especialmente beneficioso

para crear una experiencia inmersiva y altamente interactiva en la visualización de los gráficos tridimensionales.

Las imágenes utilizadas en las Figuras color 1, color 2 y color 3 fueron obtenidas de Internet, específicamente de la plataforma de elaboración propia recién mencionada. Para interactuar con el modelo 3d, visualizar estas imágenes y explorar más sobre el proyecto, puedes acceder a través de este enlace: https://elisahonorato.github.io/three.js_experimentacion/.

Los resultados obtenidos con este prototipo fueron positivos, ya que se logró la inserción exitosa de un modelo 3D propio, en formato glTF, sin perder las propiedades morfológicas del objeto. Esto demuestra la viabilidad de integrar modelos 3D personalizados en una interfaz.



Requisitos Backend

Para desarrollar un software completo, no basta con tener conocimientos en los lenguajes de programación utilizados, como Python. También es crucial contar con un backend que sea capaz de ejecutar código en ese lenguaje. El backend actúa como un servidor que procesa las solicitudes y respuestas del cliente, permitiendo la ejecución de operaciones y lógica en Python, junto con otros frameworks que puedan ser necesarios.

La presencia de este backend es esencial para que el software funcione correctamente. Permite la interacción entre el cliente y el servidor, procesando y respondiendo a las peticiones realizadas desde el frontend. De esta manera, se establece una comunicación bidireccional que permite realizar tareas como el procesamiento de datos, la gestión de bases de datos y la implementación de la lógica de negocio.

Se realizó una investigación exhaustiva con el fin de encontrar soluciones adecuadas para abordar los desafíos identificados en el proyecto. Durante este proceso, se descubrió Poetry, una herramienta de administración de paquetes y entorno virtual que simplifica la gestión de dependencias en Python. En cuanto a los principales frameworks y bibliotecas utilizados, se tomaron las siguientes decisiones:

ELECCIÓN DE DJANGO

La elección de Django como framework backend proporcionó una base sólida para el desarrollo de aplicaciones web. Su estructura organizada y sus herramientas integradas facilitaron la implementación de la lógica del negocio y permitieron una gestión eficiente de las solicitudes y respuestas del cliente.

SELECCIÓN DE PYTHON

Al utilizar Python como lenguaje de programación principal, se obtuvo una integración fluida y coherente en todo el software. Python, con su amplia biblioteca estándar y su comunidad activa, ofreció un conjunto de herramientas versátiles y confiables para el proyecto.

INTEGRACIÓN DE BPY

La integración de bpy, la API de Python para Blender, permitió manipular y crear modelos 3D de forma programática, asegurando una experiencia de usuario fluida y una integración perfecta entre el software y los modelos tridimensionales.

INCORPORACIÓN DE PANDAS

A medida que avanzaba el desarrollo, se reconoció la necesidad de leer y procesar conjuntos de datos tabulares en

formato CSV. Para esto, se incorporó Pandas, una biblioteca de Python que ofrece las capacidades necesarias para manejar eficientemente la carga, manipulación y análisis de datos tabulares, permitiendo trabajar con los datos científicos requeridos.

SQLITE

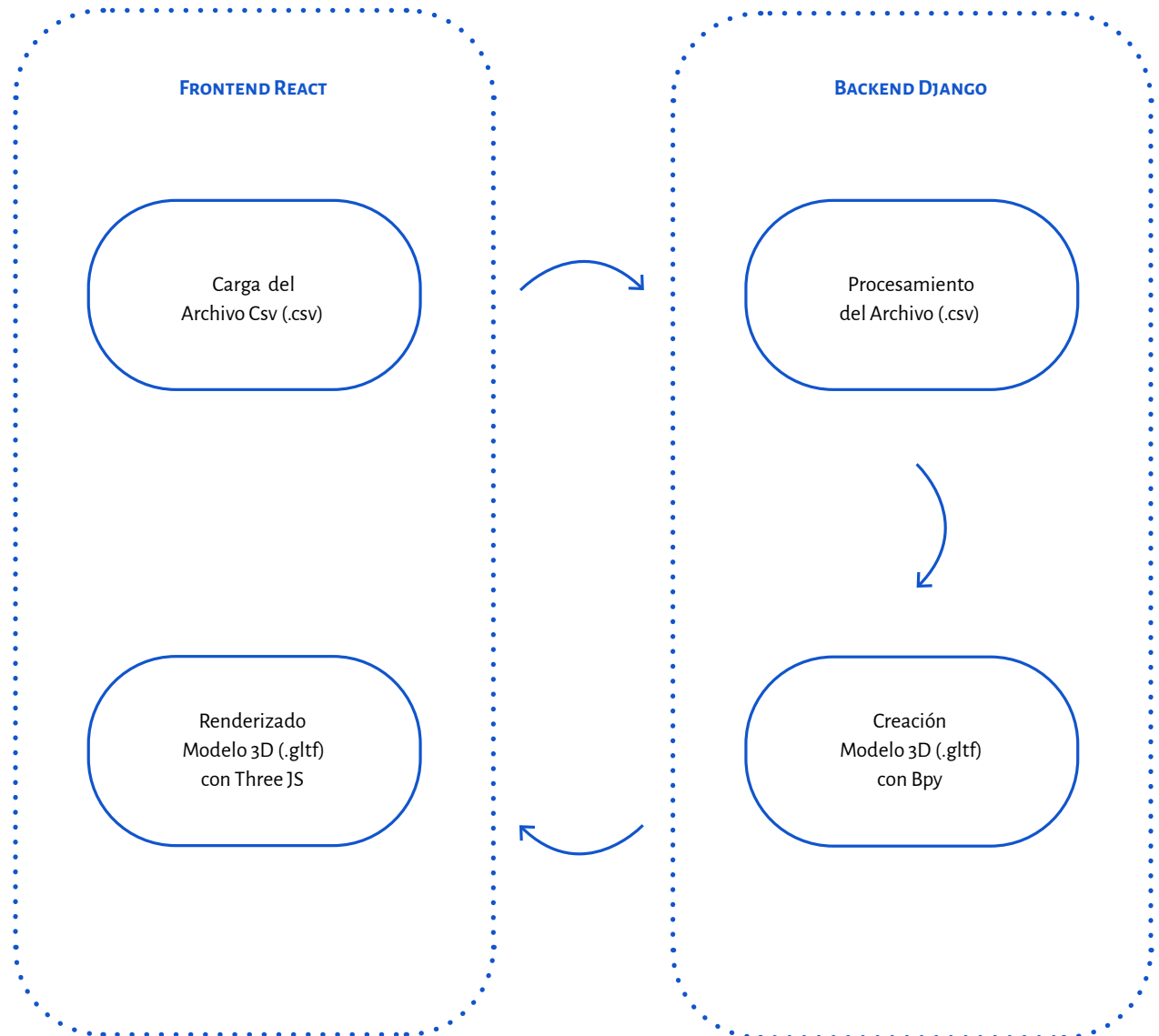
SQLite es un sistema de gestión de bases de datos relacional de código abierto, ligero y sin servidor. No se requiere configuración adicional para utilizar SQLite como base de datos principal en Django. La integración permite aprovechar el ORM de Django y realizar operaciones de base de datos de manera eficiente. SQLite es compatible con las migraciones de esquemas en Django, facilitando la administración de cambios en la estructura de la base de datos.

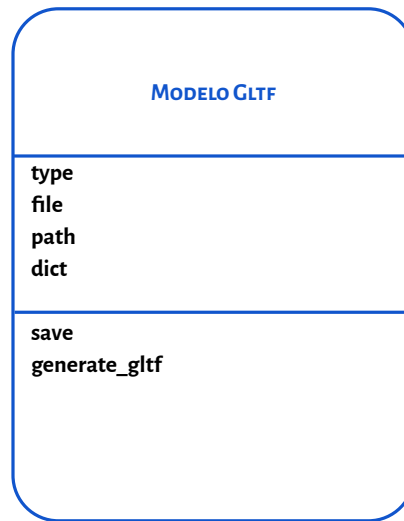
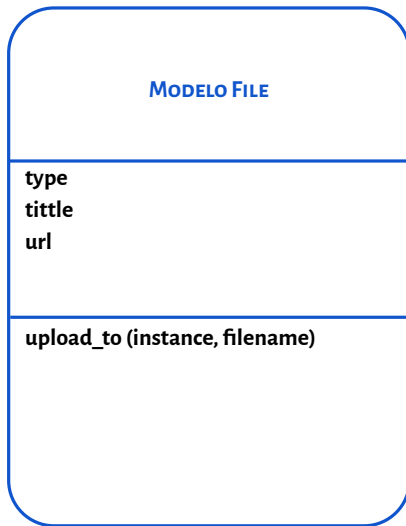
Estas elecciones y adiciones estratégicas de frameworks y bibliotecas permitieron enfrentar los desafíos específicos del proyecto y garantizar un desarrollo eficiente y exitoso.

Arquitectura del Software

Visión General

Una vez seleccionadas las tecnologías y herramientas adecuadas, se procedió a diseñar la arquitectura de la plataforma, la cual comienza con un componente en React que permite la carga de un archivo en formato .csv, el cual es enviado al backend en formato .json para su procesamiento. El backend, a su vez, utiliza la API de Python de Blender para crear un modelo en formato .gltf, junto con un archivo de ruta que especifica la ubicación del modelo en la escena 3D. El frontend, que utiliza Three.js para la visualización y manipulación de objetos en 3D, recibe estos archivos en formato .json y los carga en la escena. De esta manera, se puede interactuar con el modelo generado a través de la plataforma web, lo que permite una experiencia de usuario inmersiva y altamente interactiva.





En resumen, el código proporcionado incluye el modelo de datos **File** para representar archivos en el sistema, así como el modelo **GLTFFile** para representar archivos GLTF específicamente. La generación de archivos GLTF se realiza utilizando bibliotecas como **pandas** y **bpy**, y los objetos **File** y **GLTFFile** se utilizan para gestionar los archivos y almacenar información relacionada con ellos.

Módulos Principales

En el código proporcionado, se incluye el modelo de datos **File**, el cual representa un archivo en el sistema. A continuación, se proporciona una descripción más detallada de este modelo y su relación con el resto del código:

Modelo de datos “**File**”:

El modelo “**File**” representa un archivo en el sistema y tiene los siguientes campos:

“**type**”: Este campo es una cadena de texto que indica el tipo de archivo. En el código proporcionado, se establece el valor “csv” para representar archivos CSV. Este campo es utilizado en la función “**upload_to**” para especificar la extensión del archivo en el nombre de archivo generado.

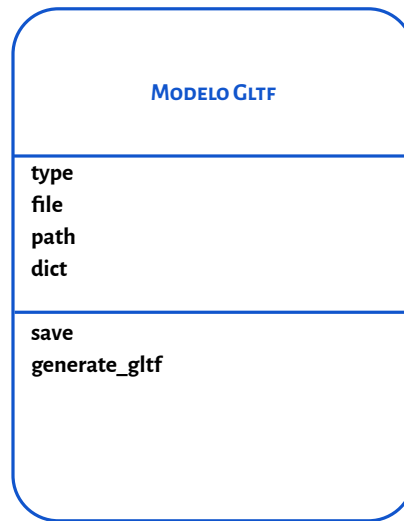
“**tittle**”: Este campo es de tipo “**CharField**” y representa el título o nombre del archivo. Tiene una longitud máxima de

80 caracteres y no se permite que esté en blanco.

“**url**”: Este campo es de tipo “**FileField**” y representa el campo donde se guarda el archivo. Utiliza la función “**upload_to**” para especificar la ruta y el nombre del archivo.

Función “**upload_to**”:

La función **upload_to** es utilizada como callback en el campo **url** del modelo **File** para determinar la ubicación y el nombre del archivo al subirlo. Toma dos argumentos: **instance**, que representa la instancia actual del modelo, y **filename**, que es el nombre original del archivo.



En resumen, el código proporcionado incluye el modelo de datos **File** para representar archivos en el sistema, así como el modelo **GLTFFile** para representar archivos GLTF específicamente. La generación de archivos GLTF se realiza utilizando bibliotecas como **pandas** y **bpy**, y los objetos **File** y **GLTFFile** se utilizan para gestionar los archivos y almacenar información relacionada con ellos.

En el código proporcionado, se define una lógica para generar la ruta y el nombre del archivo basado en el tipo y el título del archivo. Por ejemplo, la ruta podría ser "csv/nombre_archivo.csv", donde "csv" es el tipo y "nombre_archivo" es el título. El modelo **GLTFFile** está relacionado con el formato GLTF y contiene una referencia al modelo **File**: El modelo **GLTFFile** representa un archivo GLTF y contiene los siguientes campos:

type: Este campo es una cadena de texto que indica el tipo de archivo GLTF. En el código proporcionado, se establece el valor "gltf" para representar archivos GLTF.

file: Este campo es de tipo **OneToOneField** y establece una relación uno a uno con el modelo **File**. Representa el archivo asociado al objeto **GLTFFile**.

path: Este campo es **None** por defecto y se utiliza para almacenar la ruta del archivo GLTF generado.

dict: Este campo es **None** por defecto y se utiliza para almacenar un diccionario con información adicional relacionada con el archivo GLTF generado.

Función **save**:

La función **save** se encarga de guardar el objeto **GLTFFile** y establece los campos **dict** y **path** en **None**.

Función **generate_gltf**:

La función **generate_gltf** se utiliza para generar un archivo GLTF a partir de los datos CSV proporcionados. Utiliza las bibliotecas **pandas** para leer y manipular los datos CSV, y **bpy** (Blender Python) para generar la representación 3D en Blender.

La generación del archivo GLTF implica la creación de objetos 3D en Blender, como esferas y modelos volumétricos. Los datos CSV se utilizan para determinar la ubicación y las propiedades de estos objetos.

El archivo GLTF generado se guarda en una ruta local utilizando la función **bpy.ops.export_scene.gltf**. Luego, se realiza una verificación del tamaño del archivo utilizando la función **os.path.getsize**, que devuelve el tamaño del archivo en bytes, y se asegura de que el tamaño del archivo no sea demasiado grande (más de 100000000 bytes) antes de realizar cualquier

acción adicional.

Si es que el archivo cumple con las características antes mencionadas, en el diccionario **dict** del modelo **GLTFFile**, se almacena información relacionada con el volumen relativo, el volumen total y la ruta del archivo GLTF generado.

Api y Servicios

Documentación detallada de los endpoints, métodos y parámetros

La API del software ofrece los siguientes puntos finales (endpoints) y funcionalidades:

- **/files/**: Permite obtener una lista de todos los archivos disponibles y subir nuevos archivos al sistema. Se utilizan serializadores para convertir los datos en formato JSON y viceversa.
- **/files/{id}/**: Proporciona detalles de un archivo específico, incluyendo su tipo, título y URL. Permite actualizar y eliminar archivos existentes.
- **/gltf-files/**: Permite generar archivos GLTF a partir de datos CSV y almacenarlos en el sistema. Se utilizan serializadores para convertir los datos en formato JSON y viceversa.
- **/gltf-files/{id}/**: Proporciona detalles de un archivo GLTF específico, incluyendo su tipo, archivo asociado, ruta del archivo GLTF generado y detalles adicionales. Permite actualizar y eliminar archivos GLTF existentes.

```
class FilesSerializer(serializers.ModelSerializer):
    csv_url = serializers.FileField(required=False)
    gltf_url = serializers.FileField(required=False)
    class Meta:
        model = File
        fields = ["id", "csv"]
```

La vista API “PruebaView” se utiliza para subir archivos CSV y generar archivos GLTF en respuesta a las solicitudes POST. A continuación se detallan los pasos principales del proceso:

1. La solicitud POST debe incluir un archivo CSV y puede tener parámetros opcionales como “n_sam-

ples” y “n_columns”.

2. El archivo CSV se verifica para asegurarse de que cumpla con el formato correcto.
3. Se comprueba si se proporciona un nombre de archivo personalizado. Si no se proporciona, se asigna automáticamente un nombre basado en el contador de clientes.
4. Se crea un objeto “File” en la base de datos para almacenar el archivo CSV subido.
5. Se crea un objeto “GLTFFile” en la base de datos, que está asociado al archivo CSV.
6. Si se proporcionaron los parámetros “n_samples” y/o “n_columns”, se utilizan para limitar el número de muestras y columnas en el proceso de generación del archivo GLTF.
7. Se inicia un proceso en segundo plano utilizando un “ThreadPoolExecutor” para generar el archivo GLTF. Durante este proceso, se realiza el procesamiento necesario para convertir los datos del archivo CSV en la representación 3D del archivo GLTF.
8. Una vez completado el proceso de generación del archivo GLTF, se crea una respuesta JSON que incluye información sobre el archivo generado, como su ruta en el sistema de archivos.
9. Si el archivo GLTF contiene datos adicionales, se lee el contenido del archivo, se codifica en formato Base64 y se agrega a la respuesta JSON.
10. La respuesta JSON se devuelve al cliente como respuesta a la solicitud POST.

Desarrollo de Software

Desarrollo del Frontend

En el código proporcionado, se utiliza React como framework principal para el desarrollo del frontend. Algunas de las tecnologías y frameworks utilizados en el frontend incluyen:

- **React:** Es una biblioteca de JavaScript para construir interfaces de usuario. Permite crear componentes reutilizables y gestionar el estado de la aplicación de manera eficiente.
- **Redux:** Es una biblioteca de manejo de estado que se utiliza para mantener el estado global de la aplicación y facilitar la gestión de datos compartidos entre componentes.
- **Material-UI:** Es un conjunto de componentes y estilos predefinidos basados en Material Design. Se utiliza para construir interfaces de usuario atractivas y coherentes.
- **HTML2Canvas:** Es una biblioteca JavaScript que permite capturar una representación en forma de imagen de elementos HTML. Se utiliza en este caso para tomar capturas de pantalla del gráfico generado.

Estructura de carpetas y componentes principales:

- **src/**
- **components/:** Contiene componentes reutilizables utilizados en varias partes de la aplicación.
- **three/:** Contiene el componente ThreeScene que representa la escena 3D generada a partir de los datos GLTF.
- **Upload/:** Contiene el componente FileUpload que permite subir archivos CSV.
- **Theme/:** Contiene los archivos relacionados con el tema y la apariencia de la aplicación.
- **pages/:** Contiene las páginas principales de la aplicación.
- **Home.js:** Es el componente principal de la página de inicio que integra todos los componentes necesarios.

Ejemplos de código relevante y buenas prácticas de desarrollo:

- Se utiliza el estado (useState) para gestionar los datos del archivo GLTF y actualizarlos cuando se realiza una carga exitosa.
 - Se utiliza la referencia (useRef) para obtener una referencia al componente ThreeScene, lo que permite acceder a sus métodos y propiedades.
 - Se utiliza el patrón de renderizado condicional para mostrar o ocultar los componentes y la sección de descarga dependiendo de si hay datos GLTF disponibles.
 - Se utiliza el componente ThemeProvider de Material-UI para aplicar el tema y los estilos personalizados en la aplicación.
 - Se utiliza el componente Grid de Material-UI para establecer una estructura de cuadrícula responsiva en la página.
 - Se utiliza el componente Paper de Material-UI para envolver secciones específicas de la interfaz con una apariencia de papel.
 - Se utiliza el componente Button de Material-UI para crear un botón de descarga y se asocia un controlador de eventos (onClick) para capturar la imagen y descargarla.
 - Se sigue el enfoque de componentes funcionales en React, aprovechando los hooks (useState, useRef) para manejar el estado y las referencias.
 - Se utiliza una buena estructura de carpetas y nombres de componentes descriptivos para una fácil comprensión y mantenimiento del código.
- Estas prácticas de desarrollo promueven la modularidad, la reutilización de componentes y una estructura clara y organizada del proyecto.

Visualización 3D con Blender

La función `generate_gltf` es el núcleo del código y se encarga de generar un archivo GLTF (Graphics Library Transmission Format) a partir de los datos de un archivo CSV. Esta función utiliza la biblioteca Blender (mediante la importación del módulo `bpy`) para crear una visualización 3D basada en los datos proporcionados.

Aquí hay una explicación más detallada de la función `generate_gltf` y su funcionamiento:

1. Comienza importando los módulos necesarios, incluyendo `bpy` y `logging`, así como la definición del contexto de Blender y el objeto activo.
2. Luego, se verifica si hay un objeto activo en Blender. Si hay uno, se selecciona y se eliminan todos los objetos en la escena. Esto asegura que cada vez que se ejecute la función, se inicie con una escena limpia.
3. Se establece un valor para la variable `volume`, que representa el tamaño de las esferas que se crearán. Este valor puede ajustarse según las necesidades de la visualización.
4. Se inicializa un diccionario vacío para almacenar información sobre las muestras y variables encontradas en el archivo CSV.
5. Si se detectan encabezados en el archivo CSV, se establece la variable `header` en 1; de lo contrario, se deja como `None`.
6. Se utiliza la biblioteca `pandas` para leer el archivo CSV y cargar los datos en un `DataFrame`. Se especifica el separador de columnas, el separador decimal y se manejan los valores nulos.
7. Se obtienen los nombres de las columnas del `DataFrame` y se limita el número de columnas a utilizar según el valor de `n_columns`. Esto permite controlar la cantidad de variables que se mostrarán en la visualización.
8. Se establece un rango de valores para las posiciones

de las esferas, utilizando la variable `volume` para determinar el límite superior e inferior.

9. A continuación, se itera sobre las filas del `DataFrame` y se crea una esfera principal para cada muestra. Se extrae el nombre de la muestra y se agrega al diccionario de muestras. Luego se crea una esfera secundaria para cada columna adicional en la fila, representando las variables correspondientes. Se asigna un tamaño determinado a cada esfera según el valor de la variable en la fila y se establecen las posiciones relativas a lo largo del eje `x`.
10. Se crea una esfera adicional que representa el volumen total de los datos.
11. Después de generar todas las esferas y establecer sus relaciones jerárquicas, se guarda la visualización en un archivo GLTF. El archivo se guarda en una ubicación específica y se proporciona un nombre de archivo personalizado si se proporciona.
12. Durante el proceso de guardado, se redirige la salida de la consola de Blender a un archivo de texto para registrar cualquier error o mensaje de salida.
13. Una vez que se ha guardado el archivo GLTF, se verifica su tamaño. Si es demasiado grande, se devuelve un mensaje de error indicando que el archivo es demasiado grande. De lo contrario, se almacena la información del archivo GLTF en el diccionario y en el atributo `path` del objeto `GLTFFile`.
14. Por último, se devuelve el diccionario que contiene información sobre el archivo GLTF generado, como la ruta de acceso, el nombre del archivo

15. La función `has_headers` se utiliza para verificar si el archivo CSV tiene encabezados. Esto se realiza leyendo la primera línea del archivo y comprobando si cada valor es un nombre de columna válido. Si todos los valores son nombres de columna válidos, se asume que el archivo tiene encabezados y se devuelve `True`. De lo contrario, se devuelve `False`.

16. La función `round_number` se utiliza para redondear y truncar un número a un valor entero. Primero, se realiza una conversión de formato si el número es una cadena, reemplazando las comas por puntos. Luego, el número se convierte a un tipo `float` y se redondea a 5 decimales. Finalmente, se multiplica por 1000 y se convierte en un entero truncado. Este proceso garantiza que los tamaños de las esferas se ajusten correctamente.

La función `generate_gltf` utiliza los datos de un archivo CSV para crear una visualización 3D utilizando la biblioteca Blender. Cada fila del archivo CSV representa una muestra, y cada columna adicional representa una variable. Se generan esferas para cada muestra y variable, donde el tamaño de las esferas se basa en los valores de las variables. Al final, se guarda la visualización en un archivo GLTF y se devuelve un diccionario con información sobre el archivo generado.

Este código es parte de un proyecto que permite visualizar bases de datos científicas en un formato 3D interactivo, facilitando la comprensión y análisis de los datos desde una perspectiva visual.

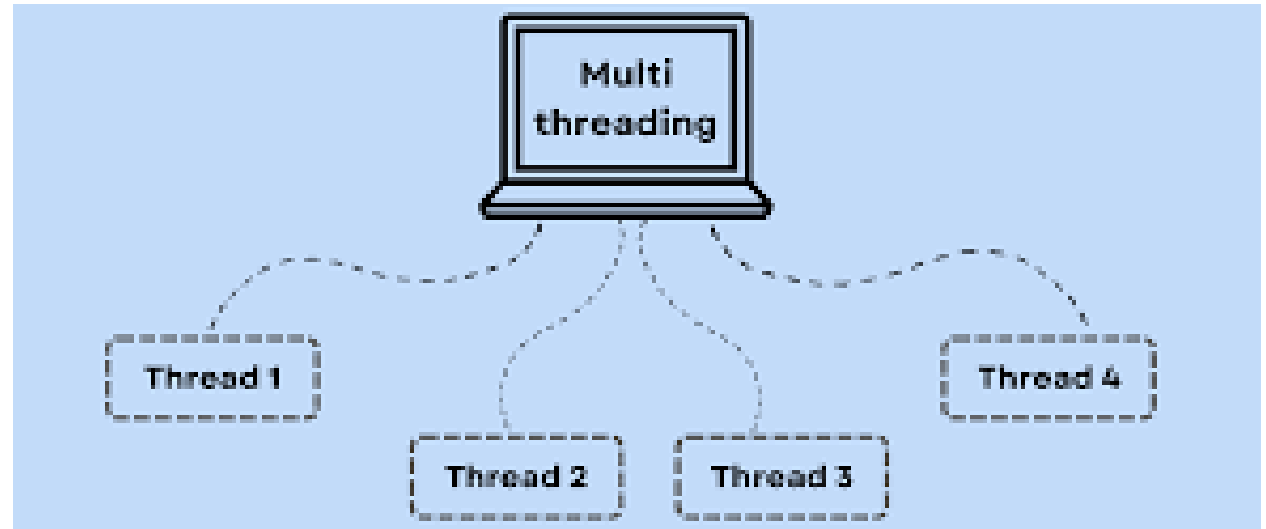
Visualización de Datos en Navegador Web

El código proporcionado es un componente de React que utiliza la biblioteca Three.js para renderizar una escena 3D en un navegador web. Aquí hay una explicación más detallada y técnica del código:

1. Importación de módulos:
 - Se importan los módulos necesarios de Three.js, como THREE, OrbitControls, GLTFLoader y GUI.
 - También se importan algunos componentes de la biblioteca Material-UI, como Box, Paper y Typography.
 - Se importa un objeto theme desde un archivo específico de temas.
2. Declaración del componente ThreeScene:
 - Se define una función de flecha con el nombre ThreeScene que recibe una prop llamada apiData.
 - Dentro de la función, se utilizan los hooks de React, como useState y useEffect, para manejar el estado y los efectos del componente.
3. Declaración de variables de estado:
 - Se utiliza el hook useState para declarar dos variables de estado: divRef y colorLegendData.
 - divRef se utiliza para almacenar una referencia a un elemento del DOM donde se renderizará la escena.
 - colorLegendData se utiliza para almacenar los datos de la leyenda de colores que se mostrarán en la interfaz de usuario.
4. Manejo de cambios de referencia:
 - Se define una función llamada refChangeHandler que se ejecuta cuando cambia la referencia del elemento DOM.
 - Si la referencia es nula, se sale de la función.
 - Si la referencia no es nula, se guarda en el estado divRef utilizando la función setDivRef.

5. Creación de la escena:
 - Se define una función asíncrona llamada createScene que toma como argumentos una referencia a la escena y los datos de la API.
 - Dentro de la función, se crea una nueva instancia de la clase THREE.Scene y se configuran las opciones de la escena, como el color de fondo y la cámara.
 - Se crea un renderizador WebGL utilizando la clase THREE.WebGLRenderer y se adjunta al elemento DOM referenciado por sceneRef.
 - Se configuran varias opciones del renderizador, como el tamaño, la relación de aspecto, las sombras y la codificación de salida.
 - Se crea una luz puntual y una luz ambiental, que se agregan a la escena.
 - Se crea una instancia de OrbitControls para permitir la interacción del usuario con la escena mediante el movimiento y la rotación de la cámara.
 - Se carga el modelo GLTF utilizando la clase GLTFLoader. El contenido del modelo se obtiene de los datos de la API y se decodifica de base64 a formato binario.
 - Una vez cargado el modelo, se ajusta su escala, se establecen los colores por defecto y se agregan a la escena.
 - Se definen varias funciones auxiliares, como showVolumen_relativo, showVolumen_total, defaultColors, distribuir, guiStyle y createGui, que se utilizan para mostrar/ocultar partes del modelo, cambiar colores, distribuir las muestras, crear la interfaz gráfica de usuario, etc.
 - Se añade un listener de eventos para capturar el movimiento del mouse y actualizar los controles de órbita.
6. Actualización de la escena:
 - Se utiliza el hook useEffect para llamar a la función createScene cuando cambian los datos de la API o la referencia del elemento DOM.
7. Renderizado del componente:

- En el renderizado, se muestra un componente Box de Material-UI con una sombra y un fondo de papel (Paper) que contiene una leyenda de colores.
 - También se renderiza un elemento div con un id de "canvas", que se utilizará como referencia para la escena Three.js.
- En resumen, el código proporcionado utiliza React y Three.js para renderizar una escena 3D en un navegador web. Utiliza varios hooks y funciones auxiliares para gestionar el estado, cargar y manipular modelos 3D, y controlar la interacción del usuario con la escena. Además, utiliza componentes de Material-UI para mostrar una interfaz de usuario con una leyenda de colores.



Manejo de Threads

En el contexto de este proyecto de código abierto, es importante reconocer los posibles desafíos económicos asociados con la capacidad limitada de los servidores para almacenar y procesar datos. Muchas soluciones comerciales ofrecen opciones de pago o limitan el número de usuarios a través de un proceso de autenticación.

Para evitar estos riesgos económicos y permitir un acceso ilimitado a los clientes, se implementa el uso de threads de Python en este código. Un thread es una secuencia de instrucciones que puede ejecutarse de forma independiente y simultánea con otras secuencias de instrucciones dentro de un programa.

Al utilizar threads, se evita que el servidor colapse al procesar múltiples solicitudes de manera concurrente. Esto significa que, en lugar de manejar cada solicitud secuencialmente y esperar a que una solicitud se complete antes de pasar a la siguiente, se pueden procesar varias solicitudes al mismo tiempo en hilos separados. Esto mejora la capacidad de respuesta del sistema y evita bloqueos.

El uso de threads permite que el servidor pueda seguir aten-

diendo otras solicitudes mientras se procesan las solicitudes actuales en segundo plano. Esto ayuda a optimizar los recursos del servidor y evitar costos negativos asociados con la expansión o sobrecarga del servidor.

En este caso, la clase `PruebaView` hereda de la clase `APIView` de Django y se utiliza para manejar las solicitudes POST recibidas en la vista. Al recibir una solicitud POST, se inicia un nuevo "thread" para procesar la solicitud de manera asíncrona y permitir que el servidor pueda seguir respondiendo a otras solicitudes mientras tanto.

El manejo de los "threads" se realiza utilizando la clase `ThreadPoolExecutor` del módulo `concurrent.futures`. Esta clase proporciona un conjunto de métodos para manejar ejecuciones concurrentes y paralelas de funciones o métodos en hilos de ejecución separados.

En el método `post`, se crea un `ThreadPoolExecutor` y se envía la función `self.process_request` para su ejecución en un "thread" separado. Esta función toma como argumentos la solicitud recibida, el objeto `gltf`, el número de muestras y columnas deseadas, y el nombre de archivo proporcionado por el usuario.

La función `process_request` a su vez invoca el método gener-

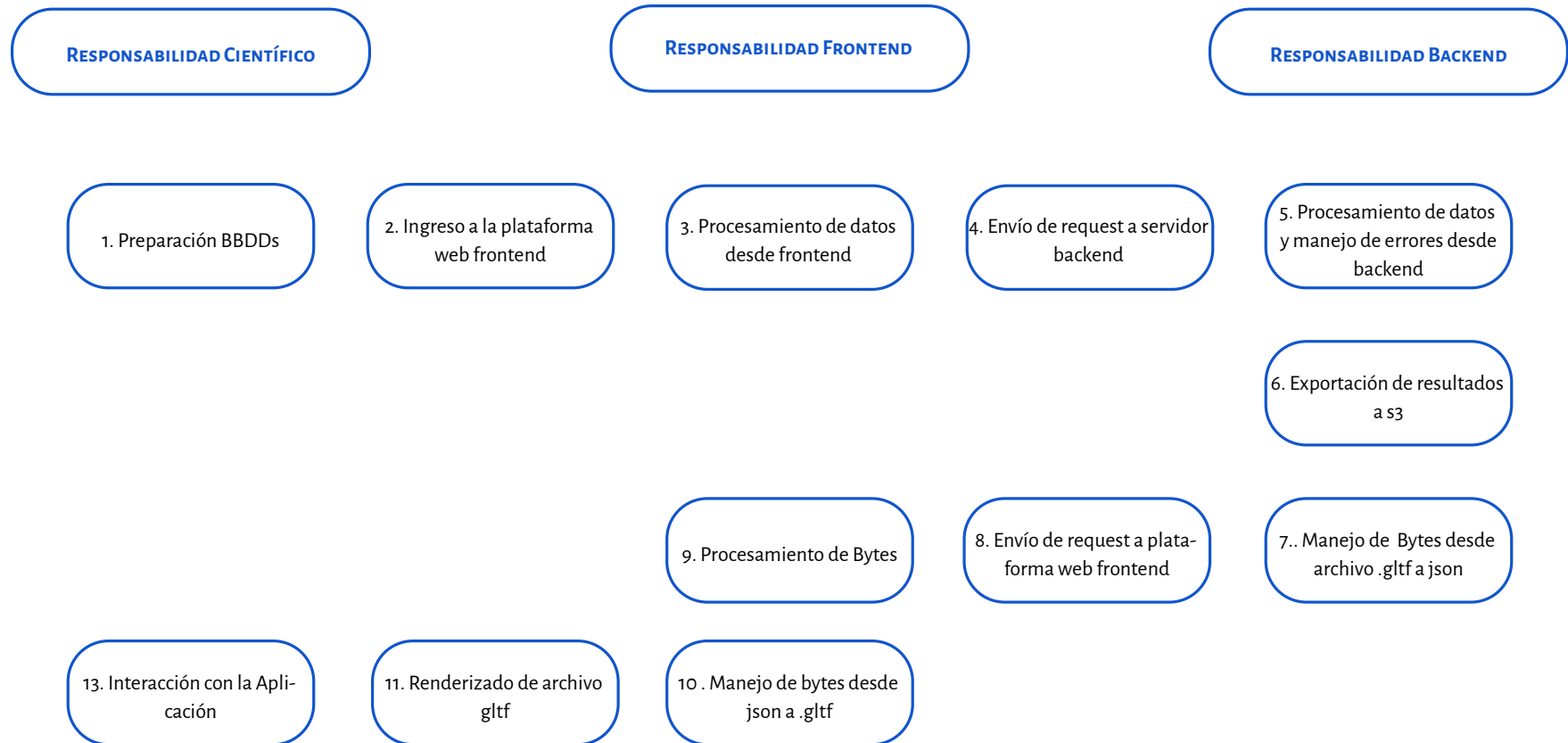
`ate_gltf` del objeto `gltf` para generar el archivo GLTF en otro "thread" separado. Una vez que la generación del archivo GLTF se completa, el contenido del archivo se lee y se convierte en una cadena Base64.

Al finalizar la ejecución del "thread", se devuelve la respuesta, que incluye los datos generados y el contenido del archivo GLTF en formato Base64.

El uso de "threads" en este contexto permite que el servidor pueda seguir procesando otras solicitudes de manera concurrente, evitando bloqueos y mejorando la eficiencia y capacidad de respuesta del sistema.

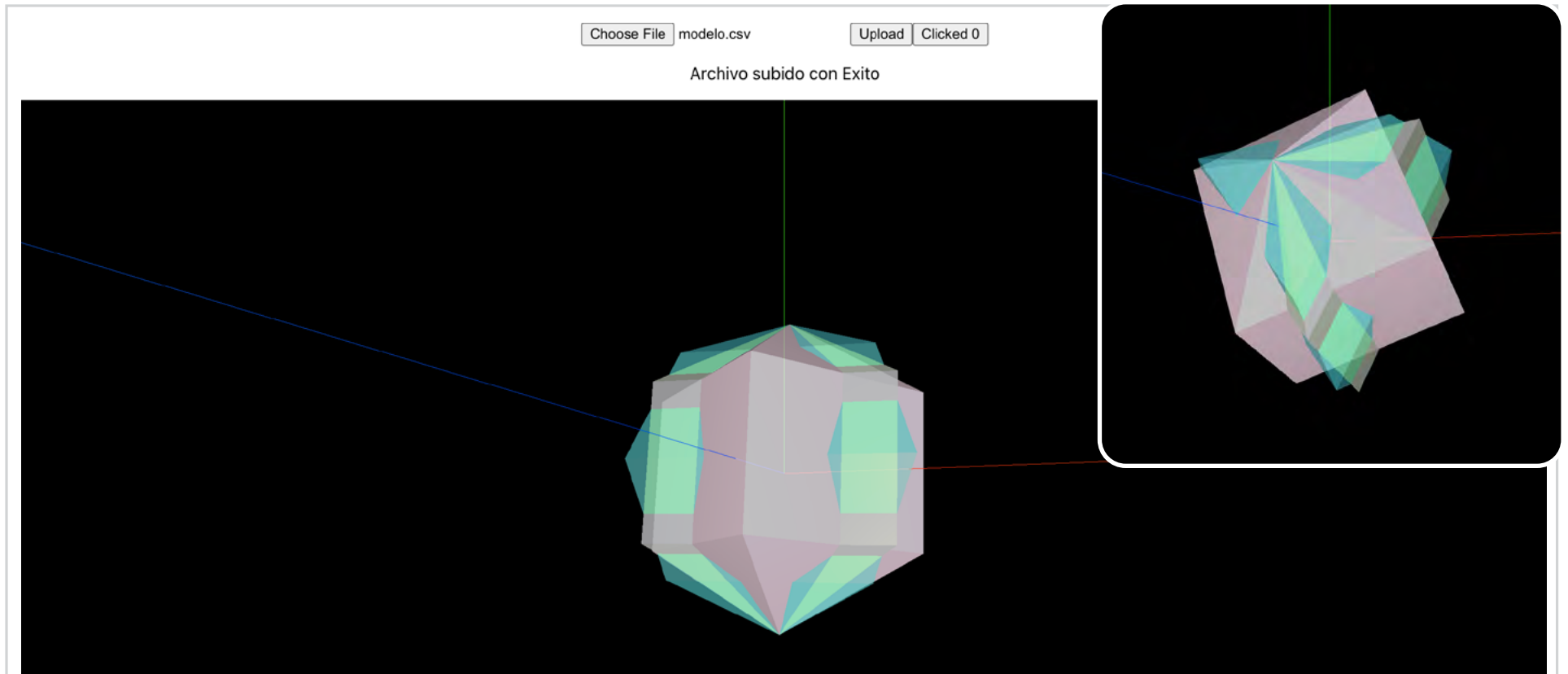
Diseño Visual

Flujo de Trabajo



Pruebas y Validación

Es importante destacar que,



[Fig Prot 1]

PROTOTIPO 1.

Datos Utilizados: Cantidad de Clorofila en una muestra.

Objetivo: Creación de una plataforma de visualización de datos con un nivel gráfico equiparable al prototipo 0.

Requerimientos:

1. La plataforma debe permitir a los usuarios interactuar con los datos en tiempo real, para que puedan ajustar la perspectiva, el zoom y la rotación, y obtener información detallada.
2. La plataforma debe contar con la capacidad de concatenar datos y exportarlos en formato .gltf.

La integración de bpy y Three.js puede ser una solución efectiva para aprovechar las fortalezas de ambas bibliotecas creando visualizaciones detalladas y complejas mientras se mantiene una experiencia de usuario interactiva y en tiempo

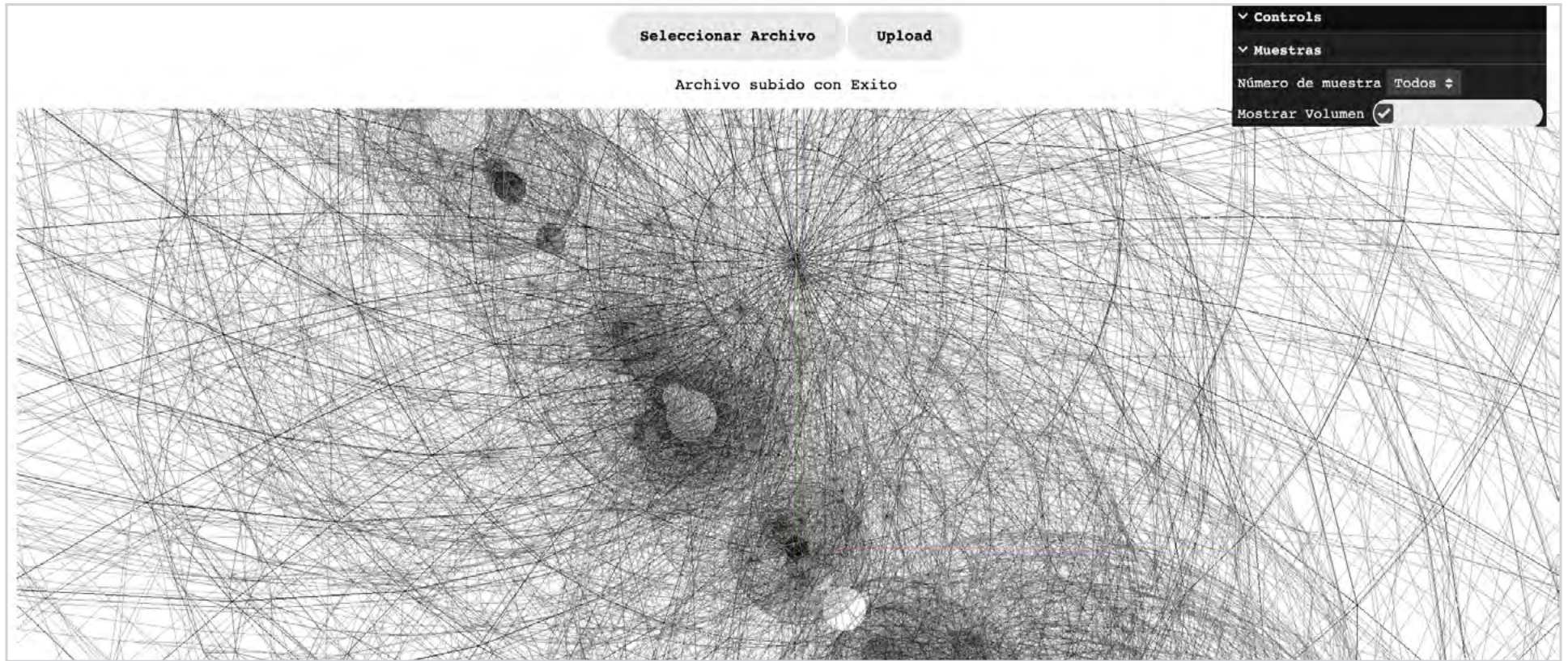
real. De esta manera, se desarrolla el prototipo inicial de la plataforma (Fig Prot 1) que integra las dos librerías mencionadas. El prototipo consta de una interfaz de entrada donde el frontend recibe los datos mediante un archivo .csv, el cual luego se envía al servidor backend. El backend se encarga de concatenar los datos del archivo csv y generar un archivo glb, el cual es devuelto a la plataforma.

Todos los requerimientos establecidos se han cumplido exitosamente. Se puede verificar en las figuras que es posible realizar la manipulación del objeto generado. La plataforma ha logrado integrar de manera eficiente las librerías necesarias, permitiendo la concatenación de datos y la generación de archivos .gltf. Los usuarios tienen la capacidad de interactuar con el objeto resultante, aplicando transformaciones y visualizando los cambios en tiempo real.

Testeo:

El testeo resultó ser positivo para el avance del proyecto, ya que se comprueba la factibilidad de las herramientas utilizadas y también un primer testeo. Este consistió en mostrar la plataforma a distintos usuarios con el objetivo de validar solamente la usabilidad, ya que para un avance en otros aspectos, se requería de conocimiento técnico y feedback especializado en el usuario.

Una gran parte de los participantes mostró interés por el aspecto visual del prototipo, pero, como era de esperar, no mostraba un mejor entendimiento de los datos. Este prototipo cumplió con los requerimientos, y ayudó a la creación del siguiente prototipo.



[Fig Prot 2]

PROTOTIPO 2

Datos Utilizados: Cantidad de Clorofila en una muestra.

Objetivo: Mejorar el aspecto epistemológico de la plataforma en comparación con el prototipo inicial. Se busca una mejor comprensión de los datos.

Requerimientos:

1. La plataforma debe poder mostrar características que no se encuentren explícitamente en los datos, pero que permitan aprender sobre ellos.
2. La plataforma debe contar con una funcionalidad de filtrado y agrupación de datos para facilitar la exploración y el análisis.
3. Los usuarios deben poder interactuar con los datos y las visualizaciones de manera efectiva, permitiendo explorar y descubrir.
4. La plataforma debe representar los datos con características morfológicas que optimicen el entendimiento de los datos.

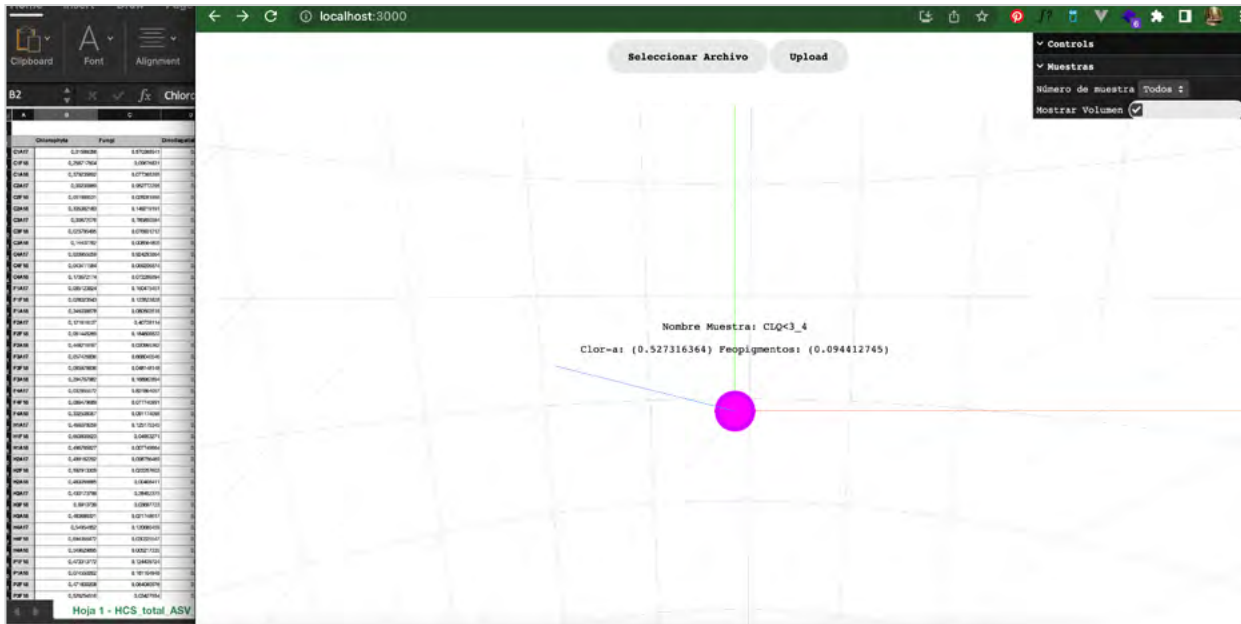
Para la elaboración de este prototipo, se utilizó el mismo csv que en la primera iteración. Se implementa una interfaz de usuario gráfica (GUI) que ofrece funcionalidades adicionales para mejorar el entendimiento de los datos. La GUI permite filtrar los datos según el número de muestra, lo que facilita el análisis y la visualización de conjuntos específicos de datos. Además, la interfaz proporciona una opción para visualizar todos los datos en conjunto, lo que permite obtener una visión general de la totalidad de los datos y su distribución. Esto es especialmente útil para identificar patrones o tendencias generales en los datos.

En este prototipo (Fig Prot 2) se ha utilizado una textura ligeramente transparente siguiendo el principio del cierre de la Gestalt. Esto permite visualizar todas las muestras en conjunto de manera clara y coherente, facilitando la comprensión global de los datos, sin obstruir las muestras.

Testeo:

El proceso de prueba fue exitoso y se obtuvieron resultados positivos con respecto a la GUI implementada. Sin embargo, se reconocieron áreas que requieren mejoras adicionales. A pesar de que los datos revelaban información en 3D, se determinó la necesidad de mejorar la representación gráfica de los resultados para lograr una visualización más efectiva y comprensible.

Para el próximo prototipo, se propone explorar diferentes enfoques y técnicas de visualización con el objetivo de encontrar la mejor manera de representar los datos en 3D. Se prestará atención a la presentación de información que sea fácil de entender y que proporcione una representación más intuitiva y accesible para los usuarios.



[Fig Prot 3]

PROTOTIPO 3.

Datos Utilizados: Cantidad de Clorofila en una muestra.

El prototipo 3 tiene como objetivo principal mejorar la representación gráfica de los datos, manteniendo los mismos objetivos y datos del prototipo anterior. Se presta especial atención a aspectos clave como la selección de colores apropiados, la manipulación de la escala para resaltar detalles relevantes y la inclusión de etiquetas o anotaciones para destacar aspectos significativos de los datos.

En el prototipo, se ha trabajado con datos de clorofila a y feopigmentos presentes en cada espécimen. Estos dos pigmentos son comunes en organismos fotosintéticos como plantas y algas. Para representar estos datos, se ha utilizado un mapa de color en forma de gradiente, asignando tonos azules para la clorofila a y tonos rojos para los feopigmentos.

Al posicionar el cursor sobre una muestra específica, se muestran los datos correspondientes como el nombre del espécimen,

la cantidad de clorofila a y la cantidad de feopigmentos. Esta visualización detallada permite a los usuarios obtener información precisa sobre cada muestra individual (Fig Prot 3)

La elección de un mapa de color en forma de gradiente facilita la comparación y análisis de los niveles de clorofila a y feopigmentos entre las muestras. Los tonos de color proporcionan una representación visual intuitiva, lo que permite identificar rápidamente las variaciones en los niveles de los pigmentos.

En el frontend, se utiliza una estructura de datos de diccionario para identificar y organizar los datos (Fig Dict). En esta estructura, los nombres de las muestras actúan como claves en el diccionario principal. Cada clave está asociada a otro diccionario que contiene las llaves "feopigmento" y "clorofila a", y sus respectivas cantidades se almacenan como valores.

Esta organización permite un acceso eficiente a los datos desde el frontend. Al tener el nombre de una muestra como

clave, es posible acceder rápidamente al diccionario asociado que contiene las cantidades de feopigmento y clorofila a para esa muestra en particular.

Esta estructura de diccionario facilita la manipulación y visualización de los datos en el frontend. Por ejemplo, se pueden realizar operaciones como filtrar muestras basadas en ciertos criterios o mostrar las cantidades de feopigmento y clorofila a de forma individual o en conjunto.

```
ejemplo = {
  'path': 'ruta/al/archivo.glb',
  'model': {
    'samples': [
      {
        'name': 'Muestra 1',
        'colors': [0.2, 0.5],
        # Valores de feopigmentos y clorofila a respectivamente
        'variables': ['Variable 1', 'Variable 2']
      },
      {
        'name': 'Muestra 2',
        'colors': [0.1, 0.3],
        'variables': ['Variable 1', 'Variable 2']
      },
      ...
    ],
    'vol': 10.5 # Valor del volumen total
  }
}
```

[Fig Dict]

Testeo:

Luego de testear, el usuario sugiere complejizar los datos que se reflejan en la plataforma, con el objetivo de potenciar su valor. Es por esto, que propone modificar el ejemplo utilizado, que consiste en la cantidad de clorofila en una muestra, al volumen de seres vivos en especímenes de agua.

Seleccionar Archivo

Upload

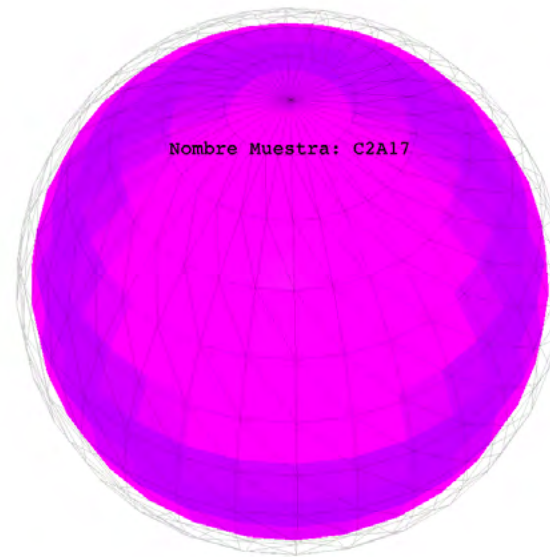
Archivo subido con Exito

Controls

Muestras

Número de muestra C2A17

Mostrar Volumen



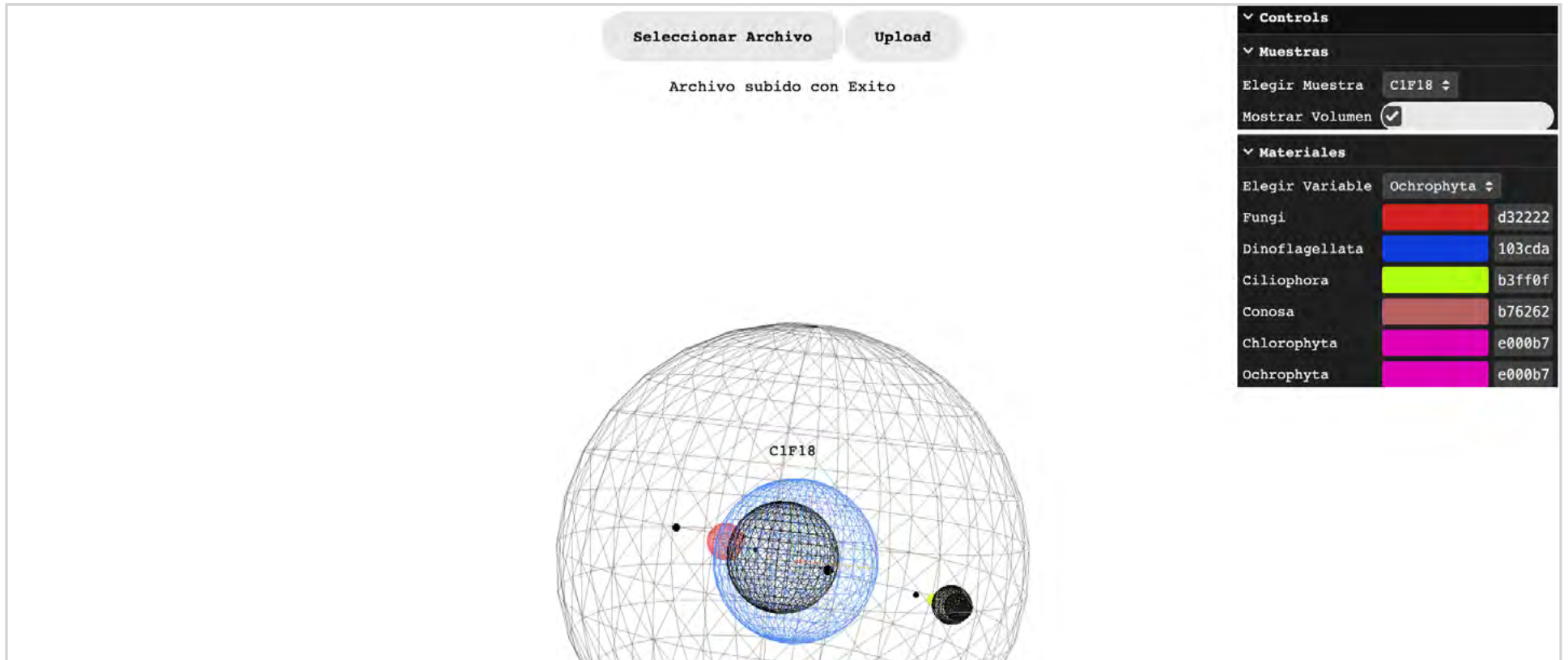
[Fig Prot]

La figura representa la muestra denominada C2A17. Al observarla, se puede notar una alta concentración de tonos azules, indicando la presencia predominante de clorofila a en la muestra. Sin embargo, también se aprecia la presencia de feopigmentos, representados por tonos morados en la gráfica. Esta combinación de colores intensos evidencia una mezcla significativa de ambos pigmentos en la muestra.

La muestra C2A17 exhibe un volumen total notable en comparación con su tamaño relativo, representado mediante un material transparente que envuelve la muestra. Esta característica sugiere una concentración significativa de clorofila en dicha muestra.

Es relevante resaltar que el prototipo actual se enfoca principalmente en el análisis individual de cada muestra, lo cual

funciona de manera efectiva. No obstante, su capacidad para representar y comparar todas las muestras en conjunto es limitada. Por lo tanto, para el próximo prototipo, se llevará a cabo una exploración exhaustiva de diferentes enfoques con el objetivo de presentar todas las muestras como un conjunto cohesivo.



[Fig Prot 4]

PROTOTIPO 4

Datos Utilizados: Cantidad de Seres Vivos en el agua.

El usuario proporciona un archivo .csv extenso que contiene alrededor de cuarenta variables representando seres vivos en el agua, junto con sesenta muestras. Este escenario implica un desafío grande en la eficiencia del procesamiento de datos, y también en su visualización.

Requerimientos:

1. La plataforma debe ser capaz de mostrar grandes cantidades de datos, y representarlos de manera distintiva entre sus categorías.

El software ha sido actualizado con un enfoque innovador centrado en la visualización de gráficos multivariantes. Esta nueva aproximación permite manejar conjuntos de datos complejos que contienen alrededor de 40 variables. La representación en 3D juega un papel central en este enfoque, aprovechando la

perspectiva espacial y la interactividad para mostrar de manera simultánea múltiples variables. Esta mejora en la visualización facilita el análisis y la comprensión de la información compleja al proporcionar una representación visual atractiva y comprensible, mejorando significativamente la comprensión de los patrones y relaciones en los datos.

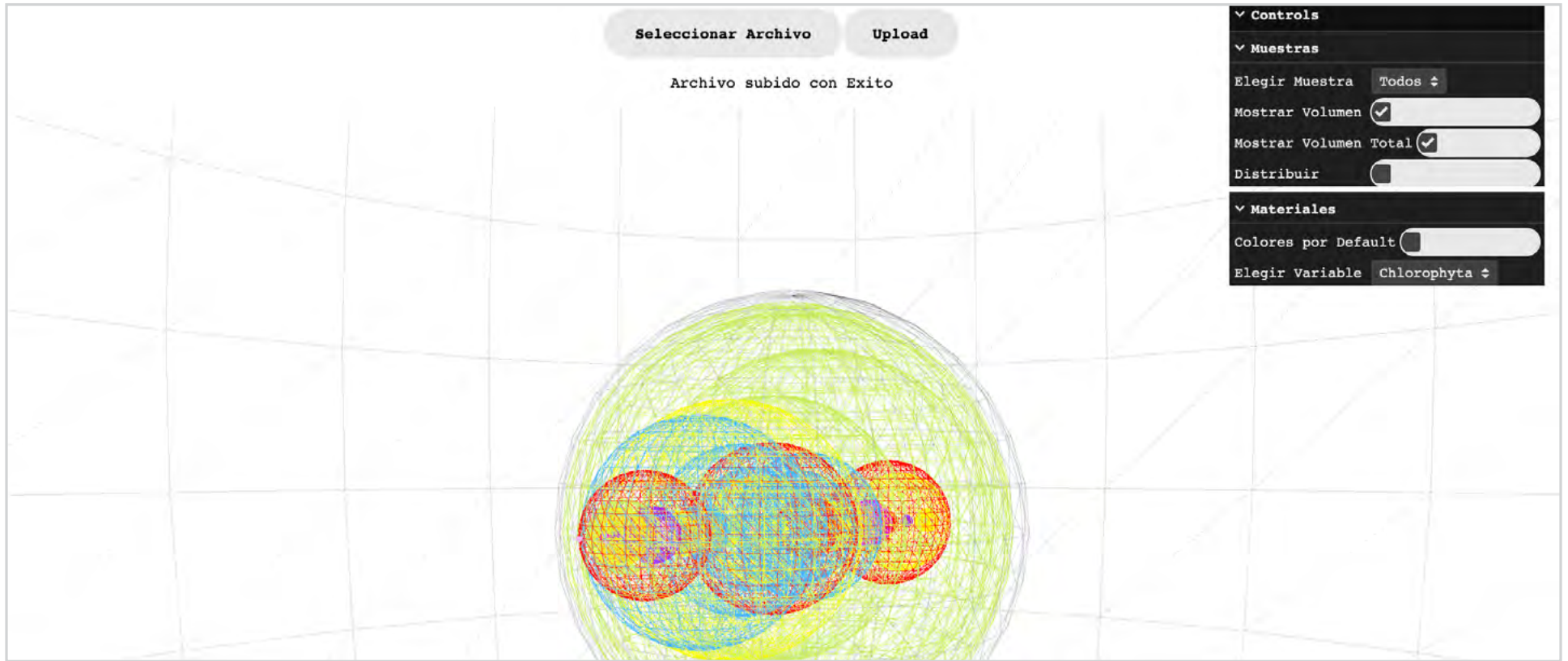
En el prototipo actual (Fig Prot 4), se muestra una composición de varias muestras, cada una con múltiples variables. Sin embargo, debido a consideraciones de tiempo y optimización, se han utilizado solo unas pocas variables en esta iteración. En entregas posteriores, se espera que todas las variables estén disponibles para su visualización.

Para ofrecer una experiencia de usuario más personalizada, se ha incorporado una opción que permite asignar un color específico a cada variable. En la imagen proporcionada, se puede

observar que la variable “fungi” se representa en color rojo, “dinoflagellata” en color azul y “ciliophora” en color verde. Aun no se han asignado colores a las demás variables presentes en la muestra, por lo que se pueden apreciar otros colores que no están relacionados con la muestra en sí.

Testeo:

Tras consultar con expertos en usabilidad, como el Profesor Felipe Cortéz, se ha tomado la decisión de implementar diferentes paletas de colores predefinidas que permitan a los usuarios seleccionar opciones dentro de un conjunto establecido. Esta medida, permitirá a los usuarios personalizar la apariencia visual de los gráficos y adaptarlos a sus preferencias individuales. Esto contribuirá a una mayor satisfacción y usabilidad del programa, brindando una experiencia más completa y agradable a los usuarios en su interacción con los datos.



[Fig Prot 5]

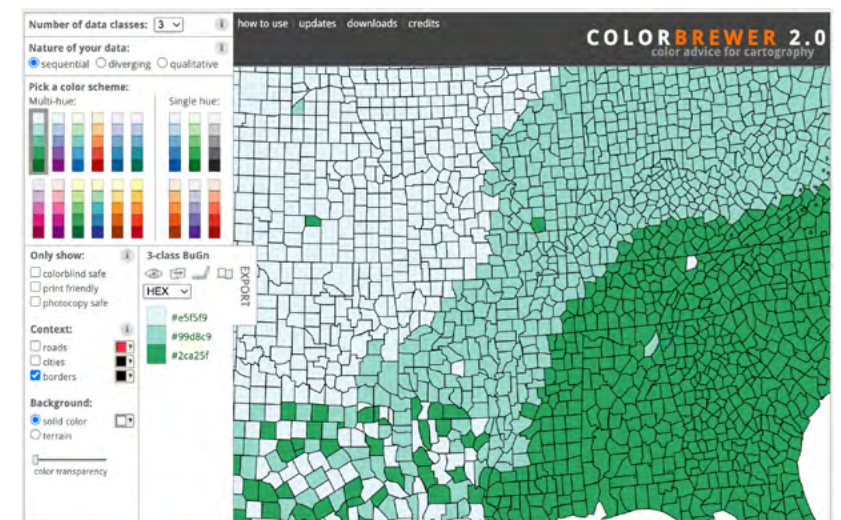
PROTOTIPO 5

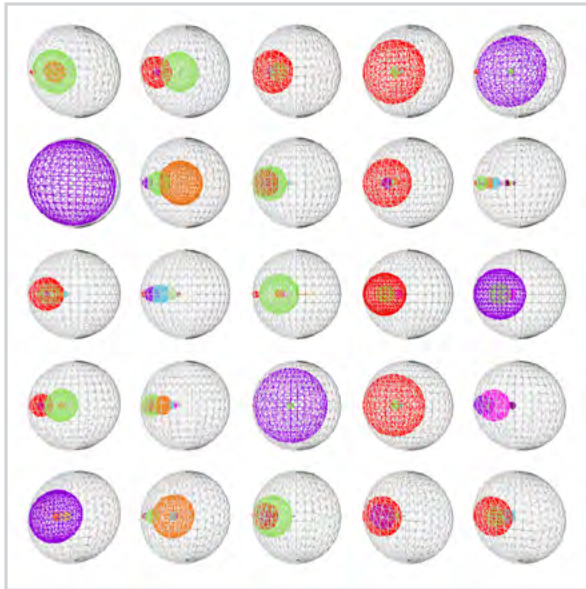
Datos Utilizados: Cantidad de Seres Vivos en el agua.

Requerimientos:

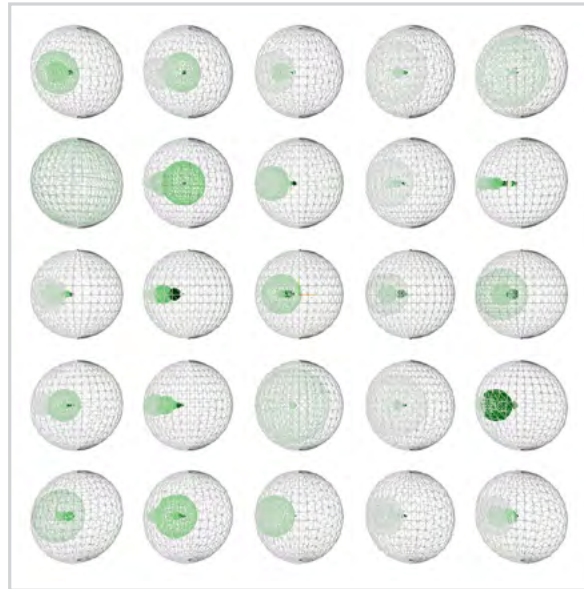
1. Proporcionar distintas opciones a los usuarios para seleccionar colores distintos.

En esta iteración (Fig Prot 5), se trabaja con el objetivo en proporcionar a los usuarios diversas opciones para seleccionar colores, incluyendo aquellas que consideren las necesidades de las personas con daltonismo. En este sentido, se exploraron referentes como Color Brewer (Fig Color Brewer), una plataforma reconocida por su enfoque en la generación de paletas de colores accesibles y visualmente armoniosas.

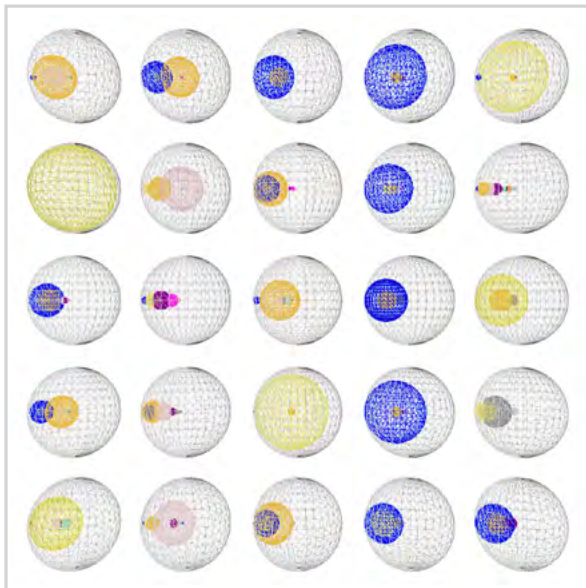




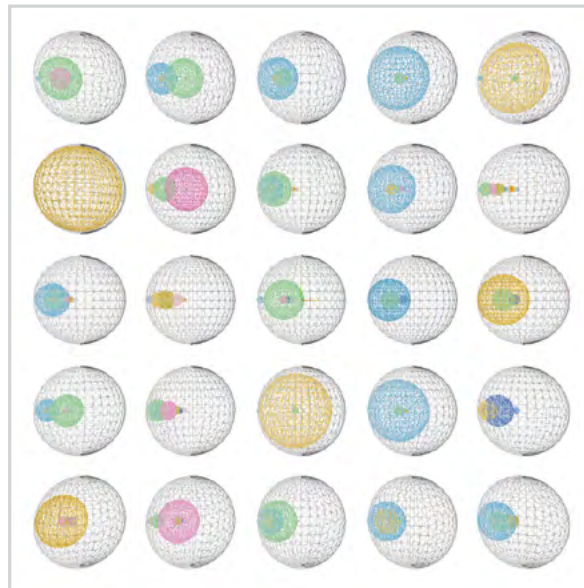
[Fig Col Default]



[Fig Col Secuencial]



[Fig Col Daltonico]



[Fig Col Divergente]

Se generan entonces, cuatro paletas de colores predeterminadas. Cada una de ellas, tiene su propia intención estética y funcionalidad, y su elección dependerá del contexto y los objetivos específicos del análisis de datos.

Paleta de colores default (Fig Col Default): Esta paleta ofrece una amplia gama de colores vibrantes y variados, desde tonos primarios intensos como el rojo y el azul, hasta tonos más suaves como el rosa claro y el lila. Incluye una mezcla de colores cálidos y fríos, lo que permite una representación visualmente atractiva y diversa de los datos.

Paleta de colores para daltónicos (Fig Col Daltonico): Esta paleta ha sido diseñada específicamente para ser accesible para personas con daltonismo. Los tonos cálidos como el amarillo y el naranja se combinan con tonos fríos como el azul y el verde para crear una representación legible para este tipo de usuario.

Paleta de colores secuenciales (Fig Col Secuencial): Esta paleta sigue una progresión gradual de colores, comenzando con tonos claros, avanzando hacia tonos más oscuros. Esta secuencia permite resaltar cambios o variaciones graduales en los datos y proporciona una representación visualmente coherente y ordenada.

Paleta de colores divergentes (Fig Col Divergente): Esta paleta está diseñada para resaltar diferencias significativas y contrastantes en los datos. Los colores se disponen en ambos extremos de la escala de valores, creando un efecto visual distintivo que destaca los puntos de inflexión o los contrastes notables en los datos.



PROTOTIPO 6

Datos Utilizados: Cantidad de Seres Vivos en el agua. El prototipo 6 presenta una visión más completa de la plataforma, incorporando elementos visuales como el logotipo y márgenes que brindan una apariencia más estética y organizada. Además, se ha añadido una leyenda que describe claramente las variables utilizadas en el gráfico, lo que facilita la comprensión de los datos presentados.

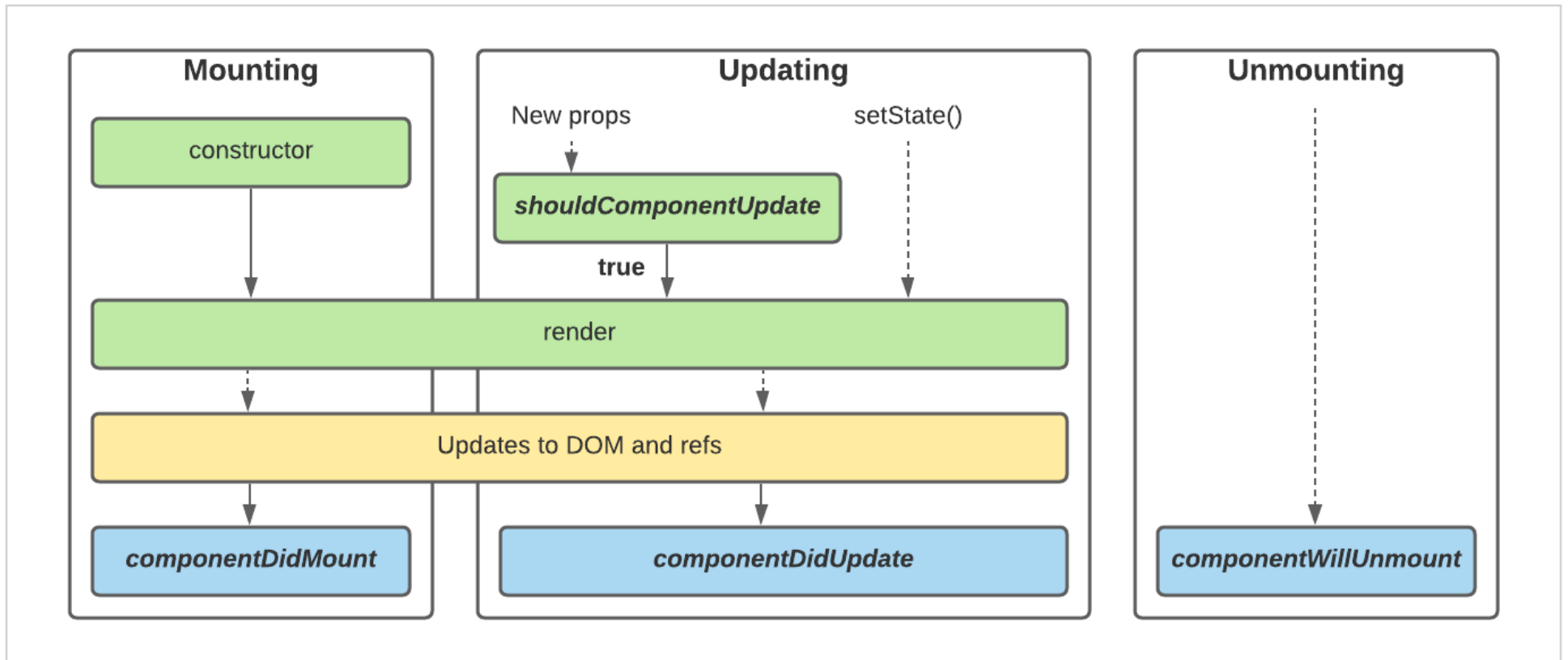
Una mejora destacada es la inclusión de una opción para descargar el gráfico generado, lo que brinda a los usuarios la posibilidad de guardar y compartir los resultados de manera conveniente. Esta funcionalidad añade versatilidad y permite un mayor aprovechamiento de los gráficos generados.

Testeo:

Después de realizar numerosas pruebas y recibir comentarios de los usuarios, surgió una pregunta recurrente: ¿Las formas en la plataforma están dispuestas de alguna manera específica o son aleatorias? Con el objetivo de abordar esta inquietud, se decidió realizar pruebas adicionales con un experto en visualización 3D, Leonel Merino.

Durante el testeo con el experto, se analizaron detalladamente las formas y variables presentes en la plataforma. Como resultado de estas evaluaciones, se llegó a la conclusión de que es necesario establecer un orden en la disposición de las figuras 3D en función de sus variables. Este enfoque no solo brindará una estructura más clara y coherente a la visualización, sino que también seguirá los principios gestálticos, que ayudan a los usuarios a percibir y comprender mejor la

información presentada, siguiendo la ley de simetría y semejanza específicamente.



[Fig React Hooks]

Durante el desarrollo de este prototipo, se identificaron ciertos inconvenientes que requerían atención inmediata. Uno de los problemas principales fue la incorrecta actualización de la leyenda, lo cual resultaba en la falta de asignación de colores a todas las variables. Este inconveniente era evidente en la figura correspondiente al prototipo 6, donde algunas variables carecían de colores asignados en la leyenda.

Para abordar esta situación, se dedicó un esfuerzo considerable a la reestructuración del código, con el objetivo de establecer una estructura más sólida y eficiente.

La mejora radica en varios aspectos clave, incluyendo el uso de hooks y la implementación de verificaciones adicionales, lo que contribuye a un código más consistente y robusto. A continuación, se detalla cómo estos elementos influyen en la mejora del código:

Utilización de Hooks (Fig React Hooks): La utilización de hooks en React ha revolucionado la forma en que se desarrollan las aplicaciones, proporcionando una alternativa más consistente y funcional a los componentes de clase. La función actualizada, utiliza Hooks para gestionar los datos de la leyenda de colores (**"setColorLegendData"**). Los hooks son una característica fundamental en React que permiten el manejo del estado y la lógica de componentes funcionales. Se incluye una figura que explica el proceso de los Hooks, ya que se utiliza muchísimo a lo largo de todo el programa.

Verificación de existencia de variables en los datos actualizados: se incluye una verificación para asegurarse de que no hayan datos duplicados. Esto se logra mediante el uso del método **"some()"** en la comparación de los identificadores de las variables.

Iteración estructurada: Se realiza una iteración estructurada tanto para las variables como para las muestras. Esto mejora la legibilidad del código y facilita su comprensión. Cada iteración se enfoca en una tarea específica, como agregar una variable a los datos actualizados o asignar colores a los elementos gráficos correspondientes. Esta estructura más clara y organizada contribuye a un código más consistente y mantenible.

Manejo adecuado de los objetos padre e hijos: En la función actualizada, se incluyen verificaciones para asegurarse de que los objetos padre e hijos existan antes de realizar operaciones en ellos. Esto evita errores y excepciones inesperadas al acceder a propiedades o métodos inexistentes. Al verificar la existencia de los objetos antes de interactuar con ellos, se promueve un código más seguro y consistente.

LI
HO

1. Sube tu archivo CSV

SELECCIONAR ARCHIVO

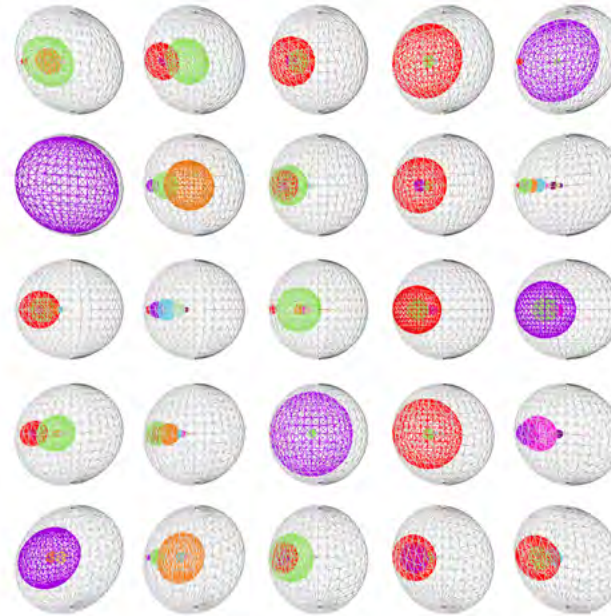
UPLOAD

Cargando Modelo...

2. Descarga tu gráfico

DESCARGAR

- Chlorophyta
- Fungi
- Dinoflagellata
- Ochrophyta
- Ciliophora
- Haptophyta
- Pseudofungi
- Conosa
- Metazoa
- Sagenista
- Cryptophyta
- Opalozoa
- Cercozoa
- Katablepharidophyta
- Telonemia
- Picozoa
- Prasinodermophyta
- Streptophyta
- Rhodophyta
- Stramenopiles_X
- Apicomplexa



Tutorial

Controls

Muestras

Elegir Muestra

Mostrar Volumen

Mostrar Volumen Total

Distribuir

Materiales

Colores por Default

```

r = round_number(row[column]) // 10
if r > 0:
    if -(r + abs(numero_x)) < intervalo[0]:
        posicion_x = numero_x + r
    else:
        posicion_x = numero_x
    numero_x += r

```

[Fig React Hooks]

[Fig React Hooks]

PROTOTIPO 7

Datos Utilizados: Cantidad de Seres Vivos en el agua.

Objetivo: El objetivo es representar las figuras geométricas de manera ordenada siguiendo los principios gestálticos mencionados en el prototipo anterior. Después de varias iteraciones de código, se llegó a una implementación final.

El backend de la aplicación utiliza un código en Python que utiliza el módulo bpy para organizar las muestras de figuras geométricas. En el código (Fig), se realiza un cálculo para obtener un valor entero "r" a partir de un número redondeado. Este valor se utiliza para determinar la posición en el eje x de las figuras geométricas. Si el valor "r" es mayor que cero, se aplica una lógica de ajuste de posición basada en intervalos en una línea recta.

En particular, se verifica si la posición está cerca del borde inferior de un intervalo. En ese caso, la posición se desplaza al borde inferior del siguiente intervalo. Si la posición no está cerca del borde inferior, se mantiene sin cambios. Luego, se actualiza la variable "numero_x" sumándole el valor "r".

La función "distribuir" se encarga de distribuir las figuras geométricas en una estructura espacial determinada, dependiendo del valor de la variable "visibility".

El proceso de distribución comienza definiendo un tamaño ("size") y una matriz vacía ("slicedArr") para almacenar las porciones de datos de las muestras.

A continuación, se establece un arreglo de ancho de cuadrícula ("grid_width") que define las posiciones en el eje x donde

se ubicarán las figuras. Este arreglo se utiliza para asignar las posiciones de las figuras en la distribución. Luego, se itera sobre las porciones de datos ("slicedArr") y se accede a cada elemento de cada porción utilizando un bucle anidado. Para cada elemento, se busca el objeto padre correspondiente en el modelo 3D mediante el método "getObjectByName".

Si se encuentra el objeto padre, se verifica el valor de "visibility". Si es "true", se establece la posición del objeto padre en base a las coordenadas definidas por el arreglo "grid_width", reinicia los controles de la escena y ajusta la posición de la cámara.

Por otro lado, si "visibility" es "false", se establece la posición del objeto padre en (0, 0, 0), es decir, en el origen de la escena.

12. Implementación

En el contexto de este proyecto en particular, los costos pueden verse afectados por la cantidad de espacio que ocupe en las instancias de Amazon EC2 y AWS S3. A medida que la utilización de recursos y almacenamiento aumente o disminuya, los costos asociados también se verán influenciados en consecuencia.

Business Model Canvas

Asociados Clave	Actividades Clave	Propuesta de Valor	Relación c/ Clientes	Segmento Clientes
	Recursos Clave		Canales	
Estructura de Costos			Vías de Ingreso	

- Albers, J. (1963). Interaction of color. Yale University Press.
- Amar, R., Eagan, J., & Stasko, J. (2005). Low-level components of analytic activity in information visualization. 111–117.
- Anderson, C. (2008). The end of theory: The data deluge makes the scientific method obsolete. *Wired Magazine*, 16(7), 16–07.
- Anscombe, F. J. (1973). Graphs in Statistical Analysis. *The American Statistician*, 27(1), 17–21. <https://doi.org/10.1080/00031305.1973.10478966>
- Arnheim, R. (2007). *Arte y percepción visual*. Alianza editorial.
- Balazka, D., & Rodighiero, D. (2020). Big Data and the Little Big Bang: An Epistemological (R) evolution. *Frontiers in Big Data*, 3, 31. <https://doi.org/10.3389/fdata.2020.00031>
- Bamforth, I. (2011). Edward R Tufte—the 'Information Man': A profile of America's leading proponent of effective strategies for seeing and showing. *British Journal of General Practice*, 61(593), 737–740. <https://doi.org/10.3399/bjgp11X613179>
- Bello-Organ, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, 28, 45–59. <https://doi.org/10.1016/j.inffus.2015.08.005>
- Bentley, T. (2013). Megaliths by a polymath. *The American Scholar*, 82(3), 10. Gale Academic OneFile.
- Bergemann, D., & Morris, S. (2019). Information Design: A Unified Perspective. *Journal of Economic Literature*, 57(1), 44–95. <https://doi.org/10.1257/jel.20181489>
- Berisha, B., & Mëziu, E. (2021). Big Data Analytics in Cloud Computing: An overview. <https://doi.org/10.13140/RC.2.2.26606.95048>
- Blachowicz, J. (2020). How Science Textbooks Treat Scientific Method: A Philosopher's Perspective. *The British Journal for the Philosophy of Science*, 60. <https://doi.org/10.1093/bjps/axp011>
- Bollier, D., & Firestone, C. M. (2010). The promise and peril of big data. Aspen Institute, Communications and Society Program Washington, DC.
- Buchanan, R. (1992). Wicked Problems in Design Thinking. *Design Issues*, 8(2), 5. <https://doi.org/10.2307/1511637>
- Bunge, M. (2018). La ciencia: Su método y su filosofía (Vol. 1). Laetoli.
- Burdick, A. (Ed.). (2012). *Digital humanities*. MIT Press.
- Cairo, A. (2012). *The Functional Art: An introduction to information graphics and visualization*. New Riders.
- Ceccarini, C., Mirri, S., Salomoni, P., & Prandi, C. (2021). On exploiting Data Visualization and IoT for Increasing Sustainability and Safety in a Smart Campus. *Mobile Networks and Applications*, 26(5), 2066–2075. <https://doi.org/10.1007/s11036-021-01742-4>
- Chang, M. (2016). *Principles of Scientific Methods* (0 ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/b17167>
- Cooper, R. J., Schriger, D. L., & Close, R. J. H. (2002). Graphical literacy: The quality of graphs in a large-circulation journal. *Annals of Emergency Medicine*, 40(3), 317–322. <https://doi.org/10.1067/mem.2002.127327>
- Cumsille, A., Durán, R. E., Rodríguez-Delherbe, A., Saona-Urmeneta, V., Cámara, B., Seeger, M., Araya, M., Jara, N., & Buil-Aranda, C. (2023). GenoVi, an open-source automated circular genome visualizer for bacteria and archaea. *PLOS Computational Biology*, 19(4), 1–16. <https://doi.org/10.1371/journal.pcbi.1010998>
- Davis, B. M. (2010). Creativity & Innovation in Business 2010 Teaching the Application of Design Thinking to Business. *Procedia - Social and Behavioral Sciences*, 2(4), 6532–6538. <https://doi.org/10.1016/j.sbspro.2010.04.062>
- De la Iglesia, R. (2022). [Entrevista Personal].
- Decap, V. (2020). Microbiología Marina y Diseño: Un resultado fascinante. <https://www.youtube.com/watch?v=M5qhWSKVPGM>
- Ding, Y. (2016). Data-driven Discovery: A New Era of Exploiting the Literature and Data. *Journal of Data and Information Science*, 1, 1–9. <https://doi.org/10.20309/jdis.201622>
- Ding, Y., Liu, X., Guo, C., & Cronin, B. (2013). The distribution of references across texts: Some implications for citation analysis. *Journal of Informetrics*, 7(3), 583–592. <https://doi.org/10.1016/j.joi.2013.03.003>
- Dirección de Comunicaciones. (2018). ¿Cómo ser una mejor universidad?
- Dix, A. (2007). Designing for appropriation. 1–4.
- Dondis, D. A. (2010). La sintaxis de la imagen: Introducción al alfabeto visual.
- Dorst, K. (2011). The core of 'design thinking' and its application. *Design Studies*, 32(6), 521–532. <https://doi.org/10.1016/j.destud.2011.07.006>
- Durrant, J. D. (2019). BlendMol: Advanced macromolecular visualization in Blender. *Bioinformatics*, 35(13), 2323–2325. <https://doi.org/10.1093/bioinformatics/bty968>
- Dyche, J. (2012). Big data 'Eureka!' don't just happen. *Harvard Business Review Blog*. http://blogs.hbr.org/cs/2012/11/eureka_doesnt_just_happen.html
- Evans, J. St. B. T. (2003). In two minds: Dual-process accounts of reasoning. *Trends in Cognitive Sciences*, 7(10), 454–459. <https://doi.org/10.1016/j.tics.2003.08.012>
- Feinberg, E. L., & Cooper, J. A. (2021). *Art in the Science Dominated World: Science, Logic and Art* (1st ed.). Routledge. <https://doi.org/10.4324/9781315074795>
- Ferster, B. (2013). Interactive visualization: Insight through inquiry. MIT Press.
- Garretón, M. (2014). Information visualization as a tool for access data collections. The case of Mira Santiago. <https://repositorio.ucll.xmliu/bitstream/handle/11534/46726/La%20visualizacio%CC%81n%20de%20informacio%CC%81n%20como%20herramienta%20para%20acceder%20a%20colecciones%20de%20datos.%20El%20caso%20de%20Mira%20Santiago.pdf?sequence=1>
- Gilbert, J. A., Jansson, J. K., & Knight, R. (2014). The Earth Microbiome project: Successes and aspirations. *BMC Biology*, 12, 1–4.
- Glen, R., Suciu, C., Baughn, C. C., & Anson, R. (2015). Teaching design thinking in business schools. *The International Journal of Management Education*, 13(2), 182–192. <https://doi.org/10.1016/j.ijme.2015.05.001>
- Gonzalez-Redin, J., Luque, S., Poggio, L., Smith, R., & Gimona, A. (2016). Spatial Bayesian belief networks as a planning decision tool for mapping ecosystem services trade-offs on forested landscapes. *Environmental Research*, 144, 15–26. <https://doi.org/10.1016/j.envres.2015.11.009>
- Gower, B. (1997). *Scientific method: An historical and philosophical introduction*. Routledge.
- Handelsman, J., Rondon, M. R., Brady, S. F., Clardy, J., & Goodman, R. M. (1998). Molecular biological access to the chemistry of unknown soil microbes: A new frontier for natural products. *Chemistry & Biology*, 5(10), R245–R249.
- Heer, J., & Shneiderman, B. (2012). Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4), 45–54.
- Hullman, J., & Diakopoulos, N. (2011). Visualization rhetoric: Framing effects in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2231–2240.
- Johnson, C. T., & Hertig, S. (2014). A guide to the visual analysis and communication of biomolecular structural data. *Nature Reviews Molecular Cell Biology*, 15(10), 690–698. <https://doi.org/10.1038/nrm3874>
- Kaisler, S., Armour, F., Espinosa, J., & Money, W. (2013). Big Data: Issues and Challenges Moving Forward. In *Proceedings of the Annual Hawaii International Conference on System Sciences* (p. 1004). <https://doi.org/10.1109/HICSS.2013.645>
- Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., & Ziegler, H. (2008). Visual Analytics: Scope and Challenges. In S. J. Simoff, M. H. Böhlen, & A. Mazaika (Eds.), *Visual Data Mining* (Vol. 4404, pp. 76–90). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-71080-6_6
- Kelling, S., Hochachka, W. M., Fink, D., Riedewald, M., Caruana, R., Ballard, G., & Hooker, G. (2009). Data-intensive Science: A New Paradigm for Biodiversity Studies. *BioScience*, 59(7), 613–620. <https://doi.org/10.1525/bio.2009.59.7.12>
- King, D. (2016). The Godfather of Data Visualisation. Medium. <https://everybodyknows.com.au/the-godfather-of-data-visualisation-9c74e8aa539c>
- Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, 1(1), 205395171452848. <https://doi.org/10.1177/2053951714528481>
- Koffka, K. (1962). *Principles of Gestalt psychology* (Vol. 44). routledge.
- Köhler, W. (1967). Gestalt psychology. *Psychologische Forschung*, 31(1), XVIII–XXX.
- Kosslyn, S. M. (2006). *Graph design for the eye and mind*. OUP USA.
- Kuhn, T. S. (2019). La estructura de las revoluciones científicas. Fondo de cultura económica.
- Landau, B., & Lakusta, L. (2009). Spatial representation across species: Geometry, language, and maps. *Current Opinion in Neurobiology*, 19(1), 12–19. <https://doi.org/10.1016/j.conb.2009.02.001>
- Lazer, D., Kennedy, R., King, G., & Vespignani, A. (2014). Google Flu Trends Still Appears Sick: An Evaluation of the 2013-2014 Flu Season. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2408560>
- Lee, J., Kang, Y., Shin, S. C., Park, H., & Lee, H. (2014). Combined Analysis of the Chloroplast Genome and Transcriptome of the Antarctic Vascular Plant *Deschampsia antarctica* Desv. *PLoS ONE*, 9(3), e92501. <https://doi.org/10.1371/journal.pone.0092501>
- Leighton, J. P., & Sternberg, R. J. (2004). *The nature of reasoning*. Cambridge University Press.
- Lupton, E. (2004). *Thinking with type: A critical guide for designers, writers, editors, & students*. Chronicle Books.
- Maeda, J., & Antonelli, P. (2001). *Design by numbers* (1. paperback ed). MIT Press.
- Malone, T., Laubacher, R., & Johns, T. (2011). The age of hyperspecialization. *Harvard Business Review*, 89, 56–65.
- Masnack, A. M., & Morris, B. J. (2022). A Model of Scientific Data Reasoning. *Education Sciences*, 12(2), 71. <https://doi.org/10.3390/educsci12020071>
- Mattson, M. P. (2014). Superior pattern processing is the essence of the evolved human brain. *Frontiers in Neuroscience*, 8. <https://doi.org/10.3389/fnins.2014.00265>
- Mazzocchi, F. (2015). Could Big Data be the end of theory in science?: A few remarks on the epistemology of data-driven science. *EMBO Reports*, 16(10), 1250–1255. <https://doi.org/10.15252/embr.201541001>
- Meinel, C., & Leifer, L. J. (2011). *Design thinking: Understand - improve - apply*. Springer.
- Munzner, T. (2014). *Visualization analysis and design*. CRC press.
- Nightingale, F. (1858). Diagram of the causes of mortality in the army in the east. Public Domain. Retrieved February, 21, 2015.
- Nightingale, F. (1987). Notes on matters affecting the health, efficiency, and hospital administration of the British army: Founded chiefly on the experience of the late war. Harrison and Sons, St. Martin's Lane, WC.
- Norman, D. A. (2002). *The Design of Everyday Things*. Basic Books, Inc.

- Ondov, B. D., Bergman, N. H., & Phillippy, A. M. (2011). Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics*, 12(1), 1–10.
- Osborne, J., Collins, S., Ratcliffe, M., Millar, R., & Duschl, R. (2003). What “ideas-about-science” should be taught in school science? A Delphi study of the expert community. *Journal of Research in Science Teaching*, 40(7), 692–720. <https://doi.org/10.1002/tea.10105>
- Oviedo, G. L. (2004). La definición del concepto de percepción en psicología con base en la teoría Gestalt. *Revista de Estudios Sociales*, 18, 89–96.
- Pavlopoulos, G. A., Malliarakis, D., Papanikolaou, N., Theodosiou, T., Enright, A. J., & Iliopoulos, I. (2015). Visualizing genome and systems biology: Technologies, tools, implementation techniques and trends, past, present and future. *GigaScience*, 4(1), 38. <https://doi.org/10.1186/s13742-015-0077-2>
- Pawłowski, J., Majchrowska, S., & Golan, T. (2022). Generation of microbial colonies dataset with deep learning style transfer. *Scientific Reports*, 12(1), 5212. <https://doi.org/10.1038/s41598-022-09264-z>
- Philipsen, L., & Kjærgaard, R. S. (2018). The aesthetics of scientific data representation: More than pretty pictures.
- Picotti, P., Bodenmiller, B., & Aebersold, R. (2013). Proteomics meets the scientific method. *Nature Methods*, 10(1), 24–27. <https://doi.org/10.1038/nmeth.2291>
- Plattner, H., Meinel, C., & Leifer, L. (Eds.). (2018). *Design Thinking Research: Making Distinctions: Collaboration versus Cooperation*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-60967-6>
- Playfair, W. (1786). *The Commercial and Political Atlas*.
- Playfair, W. (1805). *An Inquiry Into the Permanent Causes of the Decline and Fall of Powerful and Wealthy Nations*. Greenland & Norris. <https://books.google.cl/books?id=MLvIAAAAMAAJ>
- Ponsero, A. J., Bomhoff, M., Blumberg, K., Youens-Clark, K., Herz, N. M., Wood-Charlson, E. M., Delong, E. F., & Hurwitz, B. L. (2021). Planet Microbe: A platform for marine microbiology to discover and analyze interconnected omics and environmental data. *Nucleic Acids Research*, 49(D1), D792–D802. <https://doi.org/10.1093/nar/gkaa637>
- Prensky, M. (2009). *H. Sapiens Digital: From Digital Immigrants and Digital Natives to Digital Wisdom*. <https://www.learntechlib.org/p/104264/>.
- Rahman, S. M. M., Howlader, T., & Hatzinakos, D. (2019). Orthogonal Image Moments for Human-Centric Visual Pattern Recognition. In S. M. M. Rahman, T. Howlader, & D. Hatzinakos, *Orthogonal Image Moments for Human-Centric Visual Pattern Recognition* (pp. 1–17). Springer Singapore. https://doi.org/10.1007/978-981-32-9945-0_1
- Rand, P. (1985). *Paul Rand: A designer's art*. Yale University Press.
- Real Academia Española. (2023). *Diccionario de la lengua española*.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743), 195–204.
- Richardson, A. (2016). *Data-driven graphic design: Creative coding for visual communication*. Fairchild Books, an imprint of Bloombury Publishing Plc.
- Roche, R. A., Farina, F. R., & Commins, S. (2018). *Why Science Needs Art: From Historical to Modern Day Perspectives* (1st ed.). Routledge. <https://doi.org/10.4324/9781315660745>
- Rodríguez, A. (2019). La revolución de los datos: Académicos e investigadores se abren camino ante un cambio de paradigma. *Foro Chile - Italia 2019*. <https://www.uchile.cl/noticias/154084/la-revolucion-de-los-datos-un-cambio-de-paradigma#>
- Sardet, C. (2015). Plankton: Wonders of the Drifting World. *Science*, 348(6237), 870–870. <https://doi.org/10.1126/science.aac4683>
- Savage, M., & Burrows, R. (2007). The Coming Crisis of Empirical Sociology. *Sociology*, 41(5), 885–899. <https://doi.org/10.1177/0038038507080443>
- Schriger, D. L., Sinha, R., Schroter, S., Liu, P. Y., & Altman, D. G. (2006). From Submission to Publication: A Retrospective Review of the Tables and Figures in a Cohort of Randomized Controlled Trials Submitted to the British Medical Journal. *Annals of Emergency Medicine*, 48(6), 750–756.e21. <https://doi.org/10.1016/j.annemergmed.2006.06.017>
- Schwab, K. (2016). La cuarta revolución industrial. *Debate*.
- Segel, E., & Heer, J. (2010). Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1139–1148.
- Stefaner, M. (2013). New challenges for Data Design [Interview]. <https://truth-and-beauty.net/appearances/in-the-media/new-challenges-for-data-design>
- Sunagawa, S., Coelho, L. P., Chaffron, S., Kultima, J. R., Labadie, K., Salazar, G., Djahanschiri, B., Zeller, G., Mende, D. R., Alberti, A., Cornejo-Castillo, F. M., Costea, P. I., Cruaud, C., d'Ovidio, F., Engelen, S., Ferrera, I., Gasol, J. M., Guidi, L., Hildebrand, F., ... Velayoudon, D. (2015). Structure and function of the global ocean microbiome. *Science*, 348(6237), 1261359. <https://doi.org/10.1126/science.1261359>
- Times, T. N. Y. (1998). The da Vinci of Data. *The New York Times*. <https://www.nytimes.com/1998/03/30/business/the-da-vinci-of-data.html>
- Torrecilla, J. L., & Romo, J. (2018). Data learning from big data. *Statistics & Probability Letters*, 136, 15–19. <https://doi.org/10.1016/j.spl.2018.02.038>
- Trabucchi, D., & Buganza, T. (2019). Data-driven innovation: Switching the perspective on Big Data. *European Journal of Innovation Management*, 22(1), 23–40. <https://doi.org/10.1108/EJIM-01-2018-0017>
- Tufte, E. R. (1990). Envisioning information. *Optometry and Vision Science*, 68(4), 322–324.
- Tufte, E. R. (1997). *Visual and statistical thinking: Displays of evidence for making decisions* (Vol. 12). Graphics Press Cheshire, CT.
- Tufte, E. R. (2001). *The visual display of quantitative information* (2nd ed). Graphics Press.
- van Helden, P. (2013). Data-driven hypotheses. *EMBO Reports*, 14(2), 104–104. <https://doi.org/10.1038/embor.2012.207>
- Wang, Y., Sheng, H.-F., He, Y., Wu, J.-Y., Jiang, Y.-X., Tam, N. F.-Y., & Zhou, H.-W. (2012). Comparison of the levels of bacterial diversity in freshwater, intertidal wetland, and marine sediments by using millions of illumina tags. *Applied and Environmental Microbiology*, 78(23), 8264–8271.
- Ware, C. (2000). *Information visualization: Perception for design*. Morgan Kaufmann.
- Ware, C. (2010). *Visual thinking for design*. Elsevier.
- Weber, W., & Rall, H. (2012). Data Visualization in Online Journalism and Its Implications for the Production Process. 2012 16th International Conference on Information Visualisation, 349–356. <https://doi.org/10.1109/IV.2012.65>
- Weinreb, C., Wolock, S., & Klein, A. M. (2018). SPRING: A kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics*, 34(7), 1246–1248. <https://doi.org/10.1093/bioinformatics/btx792>
- Weissgerber, T. L., Milic, N. M., Winham, S. J., & Garovic, V. D. (2015). Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm. *PLOS Biology*, 13(4), e1002128. <https://doi.org/10.1371/journal.pbio.1002128>
- Wen, X., Zhu, X., Yu, R., Xiong, J., Gao, D., Jiang, Y., & Yang, G. (2019). Visualization of Chlorophyll Content Distribution in Apple Leaves Based on Hyperspectral Imaging Technology.

Agricultural Sciences, 10(06), 783–795. <https://doi.org/10.4236/as.2019.106060>

Wertheimer, M. (1912). Experimentelle studien uber das sehen von bewegung. *Zeitschrift Fur Psychologie*, 61, 161–165.

Yau, N. (2015). *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Wiley Publishing, Inc. <https://doi.org/10.1002/9781118722213>

[Fig 1] Turner, P (2019). The Phases of the Scientific Method, [Imagen]. Recuperado de [<https://towardsdatascience.com/a-data-scientific-method-80caa190dbd4>].

[Fig 2] Turner, P (2019). The Gaussian Data Scientific Method, [Imagen]. Recuperado de [<https://towardsdatascience.com/a-data-scientific-method-80caa190dbd4>].

[Fig 3] Seagate & IDC. (2018). Data Age 2025, [Imagen]. Patrocinado por Seagate con datos de IDC Global DataSphere. Recuperado de [<https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf>].

[Fig 4 y Fig 5] Facultad de Ciencias Biológicas UC. (2020). Estudiantes retoman trabajo práctico y se espera apertura de laboratorios en San Joaquín, [Imagen]. Recuperado de [<https://biologia.uc.cl/estudiantes-retoman-trabajo-practico-y-se-espera-apertura-de-laboratorios-en-san-joaquin>].

[Fig 6] Interacciones Críticas, elaboración propia (2023).

[Fig 7 y Fig 8] Prototipo O, elaboración propia (2023).

[Fig 9] Cumsille, A., Durán, R. E., Rodríguez-Delherbe, A., Saona-Urmeneta, V., Cámara, B., Seeger, M., Araya, M., Jara, N., & Buil-Aranda, C. (2023). Paraburkholderia terrae KU-15 visualization and analysis by GenoVi allows its replicon classification, [Imagen]. Recuperado de [<https://doi.org/10.1371/journal.pcbi.1010998.g006>].

[Fig 10] Cumsille, A., Durán, R. E., Rodríguez-Delherbe, A., Saona-Urmeneta, V., Cámara, B., Seeger, M., Araya, M., Jara, N., & Buil-Aranda, C. (2023). Application of GenoVi using a draft or complete bacterial genome., [Imagen]. Recuperado de [<https://doi.org/10.1371/journal.pcbi.1010998.g002>].

[Fig 11] Cuarteto de Anscombe, elaboración propia (2023).

[Fig 12] Sin autor. (2023). Blue Periodic Table of Elements, [Imagen]. Recuperado de [<https://www.pxfuel.com/en/desktop-wallpaper-vqwgo>]

[Fig 13] Mattson, M. P. (2014). Superior pattern processing is the essence of the evolved human brain, [Imagen] Recuperado de [<https://doi.org/10.3389/fnins.2014.00265>]