



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Visualización Web de Datos para Usuarios Finales en Entornos Educativos

AUTORES: Bértora Sebastian y Spaltro Franco

DIRECTOR: Lliteras Alejandra Beatriz y Grigera Julián

CODIRECTOR:

ASESOR PROFESIONAL:

CARRERA: Licenciatura en Sistemas

Resumen

Al encontrarnos en la era de la información resulta necesario que los datos se muestren en un formato accesible para la mayoría de las personas. Para representar dichos datos de forma clara y sencilla nacieron los gráficos estadísticos. Actualmente existen muchas herramientas de visualización en el mercado. La presente tesina se enfoca en estudiar y comparar estas herramientas y proponer un prototipo de visualizador de datos para usuarios finales en entornos educativos, específicamente en escuelas secundarias.

Palabras Clave

Visualizadores de datos, Modelo de datos, Modelo de bases de datos, SQL, NoSQL, Bases de datos orientadas a objetos, gráficos, grafos sociales, funcionalidad, accesibilidad

Conclusiones

Entre las herramientas de visualización estudiadas no existe una herramienta de visualización gratuita destinada a usuarios finales y que además incorporara el concepto de cursos, actividades y corrección para estudiantes y docentes.

Se espera que este prototipo de visualizador facilite la enseñanza en los establecimientos educativos como así también concientice a los estudiantes de lo poderoso que puede ser construir un gráfico de una forma sencilla para afrontar la gran cantidad de información que se puede obtener hoy en día.

Trabajos Realizados

Se realizó una comparación entre los diferentes tipos de modelos de bases de datos. También se cotejó un conjunto de herramientas de visualización de datos disponibles en el mercado.

Luego de dichas comparaciones y en consecuencia de ellas se creó un prototipo de visualizador de datos para usuarios finales en entornos educativos que permite la carga de datasets y la visualización de ellos a través de gráficos. Además permite a usuarios profesores la creación de cursos y actividades y la corrección de las entregas que realicen los usuarios alumnos de las mencionadas actividades.

Se realizó un estudio inicial de funcionalidad y accesibilidad de la plataforma propuesta.

Trabajos Futuros

Permitir la carga de otros formatos de archivos.

Añadir versionado de archivos.

Permitir vincular más de un dataset a una actividad.

Añadir mayor especificación a los cursos.

Crear persistencia inteligente de datos.

Implementar pruebas de usuario a mayor escala.

Abordar los problemas de accesibilidad detectados.

Diseñar una interfaz amigable

Sumar otros tipos de actividades.

Añadir un login integrado.

Crear nuevas estrategias de adición a cursos.

Proponer un producto disponible masivamente.

Posibilidad de trabajar en comunidad.

Sumar un foro de preguntas frecuentes.

Permitir creación colaborativa de actividades.

Fecha de la presentación: FEBRERO 2022

Índice

Índice	1
Capítulo 1: Introducción	5
Motivación	5
Objetivo	6
Estructura de la tesina	6
Capítulo 2: Estado del Arte	8
Análisis de herramientas de visualización	10
Herramientas para usuarios finales	10
Tableau	10
CODAP	10
DataWrapper	10
Herramientas para programadores	11
Saiku Analytics (Pentaho)	11
Bokeh	11
Kibana	12
Charts	12
Conclusión	12
Capítulo 3: Marco Teórico	13
Conceptos relacionados con las bases de datos	13
Datos e Información	13
Información en un contexto	13
Modelo de datos	14
Modelos de Bases de Datos	15
SQL - Bases de Datos Relacionales	15
NoSQL o Bases de datos No Relacionales	17
Orientadas a Clave-valor	18
Orientadas a documentos	19
Orientadas a columnas	20
Orientadas a Grafos	21
NewSQL	22
Bases de datos orientadas a objetos	23
Proceso ETL (extraer, transformar y cargar)	24
ELK: Elasticsearch, Logstash y Kibana	25
ElasticSearch	25
Logstash	26
Inteligencia de negocios	26
Big data	26

Visualización de datos	26
Estética de las visualización y tipos de datos	27
Habilidades de percepción de los lectores	27
Elegir el gráfico adecuado	27
La importancia de elegir un correcto diseño de gráfico	28
Decisiones de diseño en las visualizaciones de datos	28
Título, subtítulos y etiquetas	28
Marcas de graduación	29
Leyendas	29
Color	29
Tamaño y fuente	29
Forma	29
Grafos sociales	30
Capítulo 4: Comparaciones de herramientas y Experiencia de Uso	32
Comparaciones entre bases de datos SQL y NOSQL	32
Facilidad de entendimiento y uso	32
Escalabilidad horizontal y habilidad de distribución	32
Teorema CAP	32
Comparación entre ACID (SQL) y BASE (NoSQL)	33
Estandarización	34
Popularidad	34
Tabla de capacidades	36
Conclusión de la comparación entre bases de datos SQL y NOSQL	37
Definición y aplicación del proceso ETL	37
Apache Spark	37
RDD	38
Data frame	38
PySpark	38
Aplicación del proceso ETL en SQL	38
Extracción (del proceso ETL)	38
Transformación (del proceso ETL):	39
Carga (del proceso ETL):	40
Aplicación del proceso ETL para NoSQL	40
Extracción y Transformación del proceso ETL	40
Carga del proceso ETL	41
Carga en la base de datos Elasticsearch	41
Carga en MongoDB	42
Comparación de las Herramientas de visualización	42
Razón de la elección de las herramientas de visualización	42
Tableau	43
Saiku Analytics (Pentaho)	43
Bokeh	43
Kibana	43
Charts	43

CODAP y DataWrapper	43
Comparación de conexiones	44
Saiku Analytics (Pentaho)	44
Tableau	44
Bokeh	44
CODAP	44
DataWrapper	44
Comparación de costos	45
Conclusiones	45
Saiku Analytics (Pentaho)	46
Tableau	46
Bokeh	47
Kibana	47
Charts	47
CODAP	48
DataWrapper	48
APIs analizadas para construir grafos sociales	48
API de Facebook - Graph API	48
Experiencia de uso	49
API de Twitter V2	49
Experiencia de uso	50
Capítulo 5: Prototipo del Visualizador Web	51
Arquitectura del visualizador prototípico	51
Una posible implementación de la arquitectura propuesta	51
Clientes en la Plataforma	51
Servidor de la Plataforma	52
Patrón Model View Controller	52
Elección de los motores de bases de datos	53
Elección del graficador a utilizar	53
Elección del framework	53
Una implementación del prototipo de Visualizador	53
Capítulo 6: Funcionamiento de CeoDatum	61
Roles	61
Funcionalidades	61
Log in y registro en la aplicación	61
Carga de datasets a la aplicación	63
Generar gráficos a partir de los datasets	66
Inspección de datos	68
Vistas de los datasets	68
Buscador de recurrencias de palabras en tweets	71
Vista y Creación de cursos	73
Vista y Creación de actividades	75
Realización de entregas de actividades	79

Corrección de actividades	80
Perfil	82
Configuración	83
Capítulo 7: Caso de estudio y análisis de usabilidad	84
Problemática a resolver	84
Resolución de la problemática	84
Pruebas de usuario	94
Población	94
Metodología	94
Resultados de la primera pasada	95
Análisis de la primer pasada	96
Resultados de la segunda pasada	96
Análisis de la segunda pasada	97
Análisis de usabilidad	98
Capítulo 8: Análisis de accesibilidad de CeoDatum	99
Accesibilidad en aplicaciones web	99
Accesibilidad en CeoDatum	99
Accesibilidad en la creación de actividades	100
Accesibilidad en la resolución de actividades	101
Discusión del análisis de accesibilidad	102
Capítulo 9: Conclusiones y Trabajos Futuros	103
Conclusiones	103
Trabajos Futuros	104
Referencias	106
Anexo 1: Elección de la base de datos SQL	109
Anexo 2: Comparación entre los Frameworks Django y Flask de Python	112
Anexo 3: Gráfico completo de la implementación del prototipo de visualizador	113
Anexo 4: Documento de tareas a realizar en las pruebas de usuarios	114
Anexo 5: Encuesta SUS post prueba de usuario	115
Anexo 6: Resultados de las encuestas SUS	119

Capítulo 1: Introducción

Nuestra cultura es visual. El ser humano se ve bombardeado de imágenes e información a diario. Dicha información debe ser presentada de manera amigable y entendible para que sea recibida de forma correcta. El ojo humano se siente atraído por los colores y los patrones, fácilmente podemos diferenciar los colores claros de los oscuros y los círculos de los cuadrados. “La cultura visual se ocupa de los eventos visuales en los que el consumidor busca información, significado o placer en una interfaz con la tecnología visual. Por tecnología visual, me refiero a cualquier aparato diseñado para ser observado, desde las pinturas al óleo hasta la televisión y el Internet.” [Mirzoeff, 1999].

Motivación

Desde las publicaciones científicas hasta los medios de comunicación se hace uso de las visualizaciones de datos para comunicar información. Formalmente, la visualización de datos es la representación gráfica de información y datos mediante el uso de elementos visuales como cuadros, gráficos y mapas. Desde un punto de vista social, es otra forma de arte visual que capta nuestro interés y mantiene los ojos enfocados en el mensaje, es contar historias con un fin específico. Cuando se ve un gráfico, se ven rápidamente los valores llamativos. Si se puede observar y entender algo, entonces se comprende con mayor facilidad. La visualización de datos ayuda a la comunicación a través de la selección de datos en una forma más comprensible, destacando las tendencias y los valores atípicos [Unwin, 2020]. Una buena visualización cuenta una historia, elimina los datos poco entendibles y resalta la información útil. Sin embargo, esto no es tan fácil como simplemente modificar el gráfico para que se vea mejor o hacer aparecer el nombre de los datos en la tabla. La visualización de datos eficaz es un delicado equilibrio entre función y forma [Tableau, 2021]. El gráfico más simple puede ser demasiado aburrido para llamar la atención, o puede ilustrar un punto poderoso; la visualización más sorprendente puede no transmitir la información correcta en absoluto, o puede hablar por sí sola. Los datos y las visualizaciones deben cooperar entre sí y combinar un gran análisis con una gran narración, haciendo de esto un gran arte.

Las herramientas y técnicas de visualización son esenciales para analizar grandes cantidades de datos y tomar decisiones en base a ellos. Actualmente, con el incremento de la información disponible y el auge del Big Data, se vuelve necesario que sea muestreada en un formato sencillo de leer para así, ahorrar tiempo y ayudar en la toma de decisiones de las empresas, el Estado o cualquier organismo que lo requiera. Además, los datos se presentan en múltiples formatos y prescindir de esto cómo posible inconveniente al momento de presentar visualmente la información es de gran importancia.

Desde la perspectiva educativa, la visualización de datos se usa en diferentes áreas, por ejemplo en arte visual [Joy et al., 2021], matemática [Clark-Wilson et al., 2021] [Phillips et al., 2010] y ciencias sociales, donde algunas veces el uso de datos en forma de tablas no son adecuados y es necesario analizarlos en formato de grafos [Yu & Muñoz-Justicia, 2020].

Si bien existen diversas herramientas para realizar visualización de datos para usuarios sin conocimiento de programación, por ejemplo de propósito general para datos relacionales Tableau¹ y DataWrapper [Andre, 2020], y en particular para educación CODAP², y para datos en forma de grafo (estilo Twitter) SocioViz³, las mismas continúan siendo complejas o no permiten en la misma herramienta de visualización el uso de datos relacionales y en formato de grafos.

En particular, esta tesina es parte del tema de investigación desarrollado en [Lliteras, 2020] donde se propone trabajar en temas de ciencias sociales con visualización de datos en escuelas secundarias. Como parte del trabajo propuesto en [Lliteras, 2020] se dictaron diferentes seminarios y cursos para y con docentes. A partir de estos, en conjunto con el estudio bibliográfico correspondiente, surge la necesidad del prototipo aquí propuesto, donde los diferentes docentes que atravesaron algunas de las formaciones mencionadas participaron en la generación de requerimientos acordes a sus disciplinas. Los aspectos propios del marco de trabajo de [Lliteras, 2020], queda fuera del informe de esta tesina. El foco de la misma está en los aspectos relacionados a las Ciencias de la Computación.

Objetivo

Se presentará un prototipo web de visualización de datos para usuarios finales. Dicho prototipo, permite a los docentes crear actividades que involucran conjuntos de datos y diferentes tipos de visualizaciones de los mismos. Los estudiantes al resolver una actividad generan y analizan diferentes representaciones de los datos propuestos considerando una consigna dada por el docente.

Se realizará un análisis de las herramientas ofrecidas por el mercado actual para la visualización de datos y de artefactos de software desde la perspectiva del desarrollador y se optará por la más adecuada para utilizar en el desarrollo de la plataforma prototípica propuesta.

Estructura de la tesina

La presente tesina se estructura de la siguiente manera: el Capítulo 1 brinda una introducción de la problemática abordada. En el Capítulo 2 se describe el Estado de Arte. En el Capítulo 3 se aborda el Marco de Trabajo. Durante el Capítulo 4 se realiza una comparación de herramientas analizadas y se comparte la experiencia de uso de ellas. El Capítulo 5 presenta el prototipo propuesto en esta tesina. El Capítulo 6 muestra el funcionamiento del prototipo para presentar en el Capítulo 6 un caso de estudio usando el prototipo. El Capítulo 7 muestra un caso de estudio realizado en dos etapas mientras que en el Capítulo 8 se realiza un análisis de accesibilidad sobre el prototipo. Finalmente el Capítulo 9, presenta las Conclusiones y Trabajos Futuros. Al final de la tesina se presentan seis anexos. Anexo 1 describiendo la elección de la base de datos, Anexo 2 presenta una comparación en los frameworks Django y Flask. Anexo 3 muestra el gráfico completo de la implementación del prototipo de la implementación. El Anexo 4 presenta el documento

¹ <https://www.tableau.com/>

² <https://codap.concord.org/about/>

³ <https://socioviz.net/>

usado durante las pruebas de usuario. En el Anexo 5 se muestra el formato del formulario SUS utilizado al finalizar las pruebas con usuarios. Por último, el Anexo 6, muestra los resultados del formulario SUS para cada una de las dos etapas de las pruebas con usuarios-

Capítulo 2: Estado del Arte

Hoy en día las personas acceden a grandes volúmenes de datos, generalmente, a través de números o tablas que resultan poco útiles para lo que se está analizando. Uno de los retos más difíciles es interpretar estos datos, ya que, por la enorme cantidad que se presentan y las múltiples formas en que se exponen, y la persona que intenta interpretarlos tiende a agobiarse. Debido a ésta dificultad en la interpretación, muchos datos interesantes no llegan a mostrar todo potencial para obtener información.

Lograr hacer que los datos se presenten de una forma entendible tiene una gran trascendencia en la industria profesional. Aunque parece obvio, la ciencia, tecnología, ingeniería y matemática, así como el sector público, las finanzas, el marketing y hasta la historia se benefician profundamente de la correcta comprensión de los datos. Dado que el mundo de las visualizaciones es tan extenso, la aplicación y comprensión de las mismas resultan unas de las habilidades profesionales más útiles a desarrollar. Mientras mejor se logre llegar al punto visualmente a través de los gráficos, la información será más rica y aprovechable. La ciencia de datos ha tomado mayor relevancia en el campo económico y social en la última década. Las habilidades de los profesionales están cambiando para adaptarse al nuevo mundo basado en datos [Tableau, 2021]. Los datos pasaron a ser uno de los mayores activos de las organizaciones. A partir de ellos se toman decisiones y se responden preguntas de quién, qué, cuándo, dónde y cómo. Es por esto que la visualización de datos se encuentra justo en el medio entre el análisis técnico y la narración visual.

La visualización de datos es en parte arte y en parte ciencia. El reto consiste en hacer que la parte artística se muestre lo mejor posible sin que la parte científica se vea afectada y vice-versa. Lo principal que tiene que cumplir una visualización de datos es transmitir con precisión los datos sin que estos sean engañosos o se encuentren distorsionados. Al mismo tiempo una visualización de datos debe ser estéticamente agradable. Una buena presentación visual tiende a mejorar el mensaje que se está intentando transmitir con la visualización [Wilke, 2019].

Con el fin de mejorar la interpretación, se comenzó a utilizar los gráficos creados a partir de datos (visualización de datos), resultando una representación más amigable para el ojo humano. La visualización de datos es la presentación de datos abstractos que hacen que la información sea más fácil de comprender. Visualizar datos es muy beneficioso para los grandes volúmenes de datos ya que se puede concentrar mucha información en un simple gráfico y, gracias a esto, hacer que el entendimiento de esa información sea más rápido y efectivo [Faronius Hofmann & Håkansson, 2021].

Por un largo tiempo se ha pensado que las visualizaciones de datos son algo que puede ser construido usando solo el sentido común, en otras palabras, que cualquiera puede hacerlo. Pero esto ha demostrado ser falso, ya que los gráficos a menudo se usan incorrectamente y por lo tanto se malinterpreta la información [Faronius Hofmann & Håkansson, 2021].

Algunas de las razones por las que se utilizan las visualizaciones de datos son:

- Comparar datos.
- Mostrar tendencias.
- Mostrar cambios dentro de un periodo de tiempo.
- Mostrar información de grandes cantidades de datos.

Para realizar las visualizaciones, se extraen los datos crudos y se transforman para llegar a la información que será mostrada en el gráfico. Quien esté construyendo el gráfico podrá mostrar en él la información obtenida. Como por ejemplo la correlación entre los datos o las tendencias a través del tiempo [Faronius Hofmann & Håkansson, 2021].

Por otro lado un tipo de gráfico no sirve para todo tipo de información que se quiera mostrar, la tarea de seleccionar el tipo de gráfico debe llevarse a cabo con mucho cuidado ya que ciertos gráficos son más o menos útiles para distintos tipos de información.

Cuando se habla de visualización de datos, se puede agrupar a los gráficos en tipos principales de gráficos [Unwin, 2020]:

- *Gráficos de presentación*: Como su nombre lo indica son usados para presentar en forma resumida la información conseguida a partir de los datos. En ellos, las variables tienen que estar correctamente explicadas.
- *Gráficos exploratorios*: Son utilizados por los analistas para realizar investigaciones en los datos. Resulta de gran importancia que se presenten de una forma interactiva para el usuario que los manipule.

Los gráficos revelan información que las estadísticas y modelos tienden a perder. Como por ejemplo:

- Distribución inusual de los datos.
- Patrones locales.
- Brechas.
- Valores perdidos.
- Evidencias de redondeo o aproximación.
- Límites implícitos.
- Valores atípicos.

Así pueden llevar a las personas a plantear preguntas que estimulen la investigación como también a sugerir ideas. Aunque suena fácil, al interpretar gráficos se necesita experiencia para poder identificar qué información es realmente de ayuda para lo que se está buscando. Ya que un problema de los gráficos es la sobre interpretación de la información conseguida [Unwin, 2020].

A continuación se detalla el análisis realizado previamente a la elección de las herramientas de visualización utilizadas durante el desarrollo de la herramienta de visualización web prototípica. Por un lado se presentan herramientas de visualización para:

- Usuarios finales sin conocimiento de programación
- Usuarios con noción de programación
- Programadores.

Análisis de herramientas de visualización

A continuación se presentan algunas de las herramientas que están disponibles actualmente en el mercado, describiendo sus características de forma general y especificando su forma de uso. En este capítulo se dividirá a las herramientas entre herramientas para usuarios finales y herramientas para programadores.

Herramientas para usuarios finales

En esta sección se presentan los análisis de herramientas que se encuentran actualmente en el mercado y son para usuarios finales:

Tableau

Tableau es una herramienta que se utiliza para generar visualizaciones de datos, la cual comúnmente se emplea para realizar inteligencia de negocios que simplifica los datos en bruto en un formato muy fácil de entender [Balaji, 2021] [Murray, 2014].

La esencia de Tableau es simple y a la vez muy relevante: ayudar a las personas y empresas a ver y comprender todos sus datos. Y esto lo consigue ofreciendo a los usuarios toda una selección de herramientas útiles e intuitivas de inteligencia de negocios.

A través de funciones simples como la de arrastrar y soltar, se puede acceder y analizar de forma sencilla datos, e incluso, crear informes y compartir esta información con otros usuarios. Tableau se focaliza en ser una herramienta para usuarios finales por lo que no se necesitan conocimientos técnicos de sistemas de información para lograr generar la visualización que se necesita. De acuerdo a lo mencionado en [Balaji, 2021] el conocimiento sobre la utilización de Tableau se encuentra entre los más requeridos por las empresas enfocadas a la inteligencia de negocios y análisis de datos [Murray, 2014].

CODAP

CODAP (Common Online Data Analysis Platform) es una aplicación web gratuita de código abierto (open source) la cual está diseñada para el análisis de datos, en ella se puede generar visualizaciones de los datos previamente cargados a la aplicación.

Esta herramienta está pensada para el uso educativo, por esta razón, no posee demasiadas funciones ya que busca ser muy sencilla de utilizar. Debido a que es una aplicación web no necesita de una instalación y con solo conexión a internet ya se puede utilizar esta herramienta. En Codap se puede agregar la información en el momento en que se la está utilizando y luego exportarla con los nuevos registros generados [Joachim, 2018].

DataWrapper

DataWrapper es una herramienta de visualización de datos desarrollada por DataWrapper GmbH, para ser usado por periodistas, desarrolladores y diseñadores por lo que no es dificultoso de usar. En esta herramienta los usuarios pueden crear tanto gráficos como también mapas, ambos interactivos y que pueden ser usados desde la web de DataWrapper. Esta es una herramienta pensada para usuarios finales por lo que no se

necesita tener conocimientos de programación para utilizarla. Los gráficos y mapas generados a partir de esta herramienta pueden ser exportados como PNG o PDF (Al hacer esto pierden la función de ser interactivos) [Andre, 2020].

Herramientas para programadores

A continuación se detalla el análisis de herramientas de visualización que actualmente se encuentran en el mercado para las cuales se debe tener conocimiento de programación para poder utilizarlas

Saiku Analytics (Pentaho)

*Saiku Analytics*⁴ es una herramienta desarrollada por Metorite BI específicamente para el uso de Pentaho Community Edition, esta herramienta permite la creación y visualización de consultas de manera estructural (tabla) o gráfica mediante la implementación de gráficos.

Saiku Analytics es un visor OLAP (OLAP es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing)) que proporciona al usuario final una magnífica herramienta para realizar análisis de datos de forma fácil e intuitiva.

Puedes embeberlo en un servidor Pentaho como un plugin de forma fácil y sencilla. Es un plugin de Pentaho. Las operaciones se traducen a consultas MDX (MultiDimensional eXpression) que trabajan sobre la estructura dimensional [Díaz, Osorio & Amadeo, 2019] [Bermeo Torres & Simbaña Velastegui, 2020].

Para la utilización de este visor de datos se debe especificar un archivo Mondrian que requiere un esquema donde se explicita los cubos con sus respectivas dimensiones y medidas. Este esquema no es más que un archivo XML, que relaciona las tablas y columnas del motor relacional a las dimensiones y medidas correspondientes del modelo dimensional [Díaz, Osorio & Amadeo, 2019].

Bokeh

*Bokeh*⁵ es una librería de visualización interactiva de datos de Python que se dirige a los navegadores web modernos para la presentación de los mismos. Su objetivo es proporcionar una construcción elegante y concisa de gráficos versátiles, y extender esta capacidad con interactividad de alto rendimiento en conjuntos de datos muy grandes o de transmisión por secuencias. Bokeh puede ayudar a cualquier persona que quiera crear parcelas interactivas, paneles y aplicaciones de datos de forma rápida y sencilla [Jolly, 2018].

La implementación de esta herramienta, al ser una librería de Python, se hace mediante escritura de código. Para obtener los datos desde las bases de datos se pueden utilizar librerías como lo son Pandas y PySpark⁶ y también métodos del propio

⁴ <https://saiku-documentation.readthedocs.io/en/latest/>

⁵ <https://bokeh.org/>

⁶ <https://spark.apache.org/docs/latest/api/python/index.html>

lenguaje de Python. Por otra parte, para generar las distintas visualizaciones se utilizan los métodos de la propia librería de Bokeh [10].

Kibana

*Kibana*⁷ es una herramienta de visualización y gestión de datos la cual busca proveer visualizaciones mediante una interfaz flexible combinada con la búsqueda de *ElasticSearch*⁸. Lo que se requiere para la utilización de esta es un servidor web (el cual está incluido en el paquete Kibana si se quiere correr este de manera local) y cualquier navegador web que soporte las visualizaciones. Esta herramienta se conecta a ElasticSearch usando una API REST.

Kibana brinda histogramas en tiempo real, gráficos circulares y mapas. Kibana también incluye aplicaciones avanzadas, como Canvas, que permite a los usuarios crear infografías dinámicas personalizadas con base en sus datos, y Elastic Maps para visualizar los datos geoespaciales. Es libre y gratuita para su uso local [Grupta, 2015].

Charts

*MongoDB Charts*⁹ es una herramienta para crear representaciones visuales de datos alojados en MongoDB. Esta herramienta hace que la comunicación de sus datos sea un proceso sencillo al proporcionar herramientas integradas para compartir y colaborar fácilmente en visualizaciones. Proporciona una amplia variedad de tipos de gráficos para visualizar sus datos, incluidos gráficos de barras, diagramas de dispersión, gráficos geoespaciales y más [Giamas, 2019].

Conclusión

Debido a que no existe una herramienta para usuarios finales que sea gratuita y soporte una gran cantidad de datos, hay herramientas que cumplen con al menos dos de estas tres características pero no las tres, se propone realizar una herramienta de visualización que sí lo cumpla.

⁷ <https://www.elastic.co/es/kibana/>

⁸ <https://www.elastic.co/es/what-is/elasticsearch>

⁹ <https://docs.mongodb.com/charts/saas/>

Capítulo 3: Marco Teórico

En este capítulo se presentan conceptos necesarios para posteriormente abordar la arquitectura y la implementación del visualizador de datos web prototípico propuesto. Para mayor legibilidad, los conceptos se agrupan en los relacionados con las bases de datos, tratamiento de información y visualización de datos.

Conceptos relacionados con las bases de datos

Datos e Información

Los términos datos e información tienen significados muy distintos. La palabra dato se deriva del latín datum, el cual significa hecho, que puede ser un número, una afirmación o una imagen. Por otro lado, la información son los hechos o las conclusiones que tienen un significado dentro de un contexto determinado. Para la producción de la información los datos actúan como materia prima en este proceso [Oz, 2008].

Los datos básicos rara vez son significativos o útiles como información. Para convertirse en información, los datos se manipulan mediante la formación de tablas, la suma, la resta, la división o cualquier otra operación que permita comprender mejor una situación.

Un dato es una representación formalizada de entidades o hechos adecuada para la comunicación, interpretación y procesamiento por medios humanos o automáticos. El dato es un material de valor escaso o nulo para un individuo en una situación concreta, es una representación simbólica que por sí misma no reduce la dosis de ignorancia o el grado de incertidumbre de quien tiene que tomar una decisión.

La información es el aumento del conocimiento proporcionado por los datos, que puede usarse para mejorar o posibilitar una decisión. Por lo tanto, que un dato informe de algo o no lo haga, depende del conocimiento e interpretación del receptor. En la Fig. 1 se puede observar la relación y dependencia entre datos e información.



Fig. 1: Procesamiento de datos para información

Fuente:[http://www3.uacj.mx/CGTI/CDTE/JPM/Documents/IIT/Introduccion_TI/3_Modelos_sistemas/datos-vs.-informaci%C3%B3n.html]

Información en un contexto

En ciertos casos se puede dar que los datos en un contexto se consideren información en otro contexto. Por ejemplo, si una organización necesita saber la edad de

todas las personas que asisten a un partido de fútbol, una lista de esos datos es en realidad información. Pero si la misma organización quiere conocer el precio promedio de los boletos que compran las personas por edad, la lista de edades constituye simplemente datos que la organización debe procesar para generar la información.

La información es un recurso muy importante para las personas y las organizaciones. Pero no toda la información es útil, para serlo debe ser relevante, completa, precisa y actual.

Modelo de datos

Bajo la estructura de las bases de datos se encuentra el modelo de datos, el cual se compone de una colección de herramientas conceptuales para describir los datos, sus relaciones, su semántica y las restricciones de consistencia. Los modelos de datos ofrecen un modo de describir el diseño de las bases de datos en los niveles físico, lógico y conceptual. [Fundamentos de bases de datos (5a. ed.) ,2019]. Pueden clasificarse en cuatro categorías diferentes (aunque puede haber variaciones de los mismos):

- **El modelo entidad-relación.** El modelo de datos entidad-relación (E-R) se basa en una percepción del mundo real que consiste en una colección de objetos básicos, denominados entidades, y de las relaciones entre ellos.
- **Modelo de datos orientado a objetos.** El modelo orientado a objetos se puede considerar como una extensión del modelo E-R con los conceptos de clase, objeto, encapsulamiento, herencia, métodos y la identidad de los objetos.
- **Modelo relacional.** El modelo relacional usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas, y cada columna tiene un nombre único. Cabe destacar dentro de este modelo al **modelo de datos de red** y el **modelo de datos jerárquico** que precedieron cronológicamente al relacional. Estos modelos se dejaron de utilizar ya que estuvieron íntimamente ligados a la implementación subyacente y complicaba la tarea del modelado de datos. En consecuencia, se usan muy poco hoy en día, excepto en el código de bases de datos antiguas que sigue estando en servicio en algunos lugares.
- **Modelo de datos semiestructurados.** El modelo de datos semiestructurados permite la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto lo diferencia de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos.

Las bases de datos se diseñan para, en su mayoría, gestionar grandes cantidades de información. El diseño de estas bases de datos implica principalmente el diseño del modelo de las mismas. Durante el proceso de diseño de la base de datos, los modelos de datos de alto nivel resultan útiles a los diseñadores al ofrecerles un marco conceptual en el que especificar, de manera sistemática, los requisitos de datos de los usuarios y la manera en que se estructurará la base de datos para satisfacer esos requisitos. En base a toda esta información, el diseñador escoge un modelo de datos y, mediante la aplicación de los conceptos del modelo elegido, traduce esos requisitos en un esquema conceptual de la base de datos. Durante la etapa de diseño de datos el esquema generado ofrece una visión general detallada de cómo será la base de datos, abstrayendo la información de la

tecnología que será utilizada. En esta etapa, la atención del diseñador se centra en describir los datos y sus relaciones, más que en especificar los detalles del almacenamiento físico de los datos.

Para pasar de un modelo de datos abstracto a la implementación de la base de datos se continúa el desarrollo a través de dos etapas de diseño. En la etapa de diseño lógico el diseñador relaciona el esquema conceptual de alto nivel con el modelo de implementación de datos del sistema de bases de datos que se va a usar. El diseñador usa el esquema de bases de datos específico para el sistema resultante en la etapa de diseño físico posterior, en la que se especifican las características físicas de la base de datos.

Modelos de Bases de Datos

Una vez que el diseñador concluye las etapas de diseño conceptual y lógico, comienza el diseño físico de la base de datos. Durante esta etapa se deben tener en cuenta las herramientas tecnológicas que permitan almacenar, acceder y manipular los datos con la eficiencia esperada. Estas tareas de monitorización de los datos se realizan a través del Sistema Gestor de Bases de Datos o SGBD. Dentro de la monitorización se puede definir el acceso, creación, modificación, mantenimiento y eliminación de datos, así como la seguridad de los mismos.

Se pueden diferenciar cuatro distintos modelos de bases de datos para el diseño físico, estos son SQL, NoSQL, NewSQL y Orientado a Objetos [Roy-Hubara & Sturm, 2020]. Los modelos SQL, también llamados relacionales por el modelo de datos en el que se basan (modelo relacional o entidad-relación), dominan el campo entre las posibles soluciones que existen actualmente ya que ofrecen una opción segura al usuario por las características que este ofrece. A pesar de esto, durante los últimos años, las bases de datos NoSQL o también denominadas bases de datos no relacionales, han ganado terreno ofreciendo una alternativa a la rigidez que maneja su contraparte. Por su parte, las bases de datos NewSQL aparecen como un punto medio entre las opciones mencionadas anteriormente, para obtener una posible solución que brinde seguridad al usuario pero a su vez permita un manejo flexible [Pavlo & Aslett, 2016]. Y por último, recientemente ha habido un incremento en el interés del paradigma orientado a objetos para la gestión de bases de datos ya que ofrecen una opción viable a aquellos sistemas programados en lenguajes Orientados a Objetos que tienen una gran brecha de modelado de datos con las bases de datos relacionales.

SQL - Bases de Datos Relacionales

Aunque SQL es mayormente conocido como un lenguaje de consultas, también es el nombre que popularmente en algunos casos se le asigna a las bases de datos relacionales (basadas en el modelo de datos relacional o entidad-relación), es decir que se utiliza para definir una estructura de datos. Las bases de datos relacionales son una forma de modelar una base de datos definiendo la forma en la que se guardarán los mismos y las restricciones al momento de almacenarlos y manipularlos.

El lenguaje de consultas SQL está basado en el álgebra relacional. Este último se trata de un conjunto de operaciones que toman una o dos relaciones como entrada y generan una nueva relación como resultado. Las operaciones fundamentales que el álgebra

relacional define son selección, proyección, unión, diferencia de conjuntos, producto cartesiano y renombramiento. A estas operaciones se suman otras no fundamentales como el producto natural, división o asignación, pero son definidas en términos de las operaciones fundamentales. SQL agrega nuevas operaciones necesarias para la manipulación de datos como lo son la inserción, borrado y modificación y operaciones aritméticas como la suma, resta o promedio. Toda operación de álgebra relacional tiene una equivalente en SQL, pero no ocurre lo mismo en la dirección contraria. En la figura 2.1 se puede observar la operación de selección tanto en SQL como álgebra relacional.

Álgebra	SQL
(Pacientes)	Select * From Pacientes
$\sigma_{idade > 18}$ (Pacientes)	Select * From Pacientes Where idade > 18
$\pi_{CPF, nome}$ (Pacientes)	Select CPF, nome From Pacientes
$\pi_{CPF, nome}(\sigma_{idade > 18}(\text{Pacientes}))$	Select CPF, nome From Pacientes Where idade > 18

Fig. 2.1: Ejemplos de operación equivalente de selección en SQL y algebra relacional

Fuente:[<https://sites.google.com/site/syllabustics2carloscabezas/unidad-i-diseno-de-bases-de-datos-relacionales/1-4-algebra-relacional-y-los-calculos-relacionales---conocer-los-fundamentos-del-algebra-relacional>]

Las bases de datos relacionales almacenan y manipulan los datos como un conjunto de tablas con datos definidos en columnas. Se trata de un esquema rígido que define las estructuras de las tablas, tipos de datos y restricciones de los mismos. En la Fig. 2.2 se puede observar un ejemplo de modelo de base de datos relacional, en donde se percibe a los objetos del mundo real como tablas y a las características de los mismos como columnas. Cada columna está relacionada con un tipo de datos y cada tabla se puede relacionar con otra tabla a través de las relaciones. Los usuarios pueden acceder a los datos de la base de datos sin saber la estructura de la mencionada tabla.

Las bases de datos relacionales se apoyan en ACID para la manipulación de datos, es decir que las transacciones poseen las siguientes características:

- **Atomicidad (Atomicity):** Todas las transacciones relacionadas se consideran conjuntamente, si alguna falla al ejecutarse, todas se cancelan.
- **Consistencia (Consistency):** La ejecución de una transacción aislada conserva la consistencia de la base de datos.
- **Aislamiento (Isolation):** Cada transacción ignora al resto de las transacciones que se ejecutan concurrentemente en el sistema
- **Durabilidad (Durability):** Al finalizar una transacción, los cambios perduran en el tiempo aunque haya un posible fallo posterior en el sistema.

Esto último refuerza la seguridad de la base de datos, brindando al usuario un sistema confiable en donde salvaguardar su información. El punto débil de las bases de datos SQL se encuentra en la rigidez que presentan. A partir de las herramientas que utilizaron para crear un sistema certero, de cierta forma impidieron la flexibilidad del mismo.

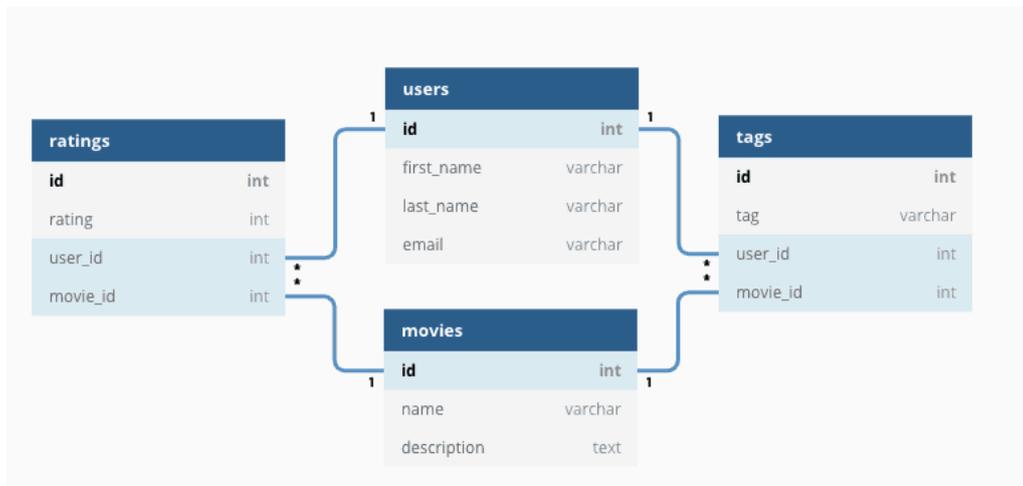


Fig. 2.2: Ejemplo de modelo de base de datos relacional.
Fuente: <https://www.omnisci.com/technical-glossary/relational-database>

NoSQL o Bases de datos No Relacionales

Debido a la necesidad de gestionar volúmenes masivos de información surgen las bases de datos NoSQL, término acuñado a finales de los 90 y que engloba todas las tecnologías de almacenamiento estructurado que no cumplen el esquema relacional.

Este término incluye a cualquier base de datos que no siga el modelo tradicional denominado sistema de gestión de bases de datos relacionales (RDBMS), lo que se busca con este concepto es tener un diseño más simplificado, un procesamiento rápido y una especialización para tratar los datos, alcanzando un mayor rendimiento, puesto que en un RDBMS los datos tienen que adaptarse a las tablas y cumplir una serie de reglas de diseño. Por lo tanto, NoSQL se puede definir como “un conjunto de conceptos que permite el procesamiento rápido y eficiente de conjuntos de datos con un enfoque en el rendimiento, la confiabilidad y la agilidad” [Moniruzzaman & Hossain, 2013] [Venkatraman et al., 2016].

Las bases de datos no relacionales están basadas en el concepto de BASE:

- **Basically available:** (Básicamente Disponible) Este término hace referencia a que las operaciones de lectura y escritura están disponibles para usarse tanto como sea posible, pero sin una eventual consistencia por lo que el dato podría no persistir en la base de datos (ya sea porque se generó un conflicto en una escritura o que el dato que se leyó no sea realmente el más actualizado)
- **Soft-state:** (Estado suave) Este término hace referencia a que debido a que no se puede garantizar la consistencia de los datos en este tipo de bases de datos solo se puede decir que el dato esté probablemente en el estado que está reflejado en la base de datos.
- **Eventual consistency:** (Eventualmente consistente) Este término hace referencia a que en algunos momentos la base de datos puede tener un estado inconsistente.

Gracias a estas características las bases de datos NOSQL son fácilmente escalables, tienen un carácter descentralizado y son flexibles, ya que permiten adaptarse a diversas necesidades [Özsu & Valduriez, 2020].

NoSQL (Not Only SQL) realmente es una categoría muy amplia para un grupo de soluciones de persistencia que no siguen el modelo de datos relacional, y que no utilizan SQL como lenguaje de consulta; pero en resumen, las bases de datos NoSQL pueden clasificarse en función de su modelo de datos en las siguientes cuatro categorías [Roy-Hubara & Sturm, 2020][Chen & Lee, 2019]:

- Orientadas a clave-valor (Key-Value stores).
- Orientadas a columnas (Wide Column stores).
- Orientadas a documentos (Document stores).
- Orientadas a grafos (Graph databases).

Orientadas a Clave-valor

Es uno de los modelos más populares dentro de las bases de datos NoSQL y también el más sencillo a comparación del resto. Este tipo funciona de la siguiente manera: cada elemento dentro de la base de datos está identificado por una clave única, lo que hace que las operaciones de escritura y lectura sean mucho más rápidas. Dado que solo se almacena el dato y una clave para identificarlo, la información se suele almacenar como un objeto binario. En la Fig. 3.1 se puede observar un ejemplo de base de datos clave-valor que representa vehículos.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year: 2005 Transmission: Auto

Fig. 3.1: Bases de datos de Clave-valor

Fuente: [https://www.researchgate.net/figure/Key-value-store-NoSQL-database-27_fig1_336686999]

Los sistemas de clave-valor prometen un rendimiento excelente para volúmenes de datos muy grandes, a cambio de ser muy simples y renunciar a funcionalidades que se tienen en otros sistemas. Una característica de este tipo es que las validaciones de los datos se delegan completamente en la aplicación cliente, siendo la base de datos, simplemente el lugar donde se guardan los datos. No se verifican integridades, no se comprueban referencias cruzadas, todo esto se ha de implementar a nivel de aplicación.

En un sistema clave-valor se pueden agrupar los datos en contenedores (Fig. 3.2), también llamados cabinets, en cada contenedor se puede almacenar tantas parejas de

clave-valor como se necesite. Hay sistemas que permiten tener claves duplicadas y hay otros que no (dependiendo de para que se utilicen los datos guardados podría convenir uno o el otro), o que se puede indicar que no se quiere que se dupliquen. En cada contenedor es posible tener datos de la misma naturaleza (por ejemplo productos, pedidos, clientes, etc.) o totalmente diferentes.

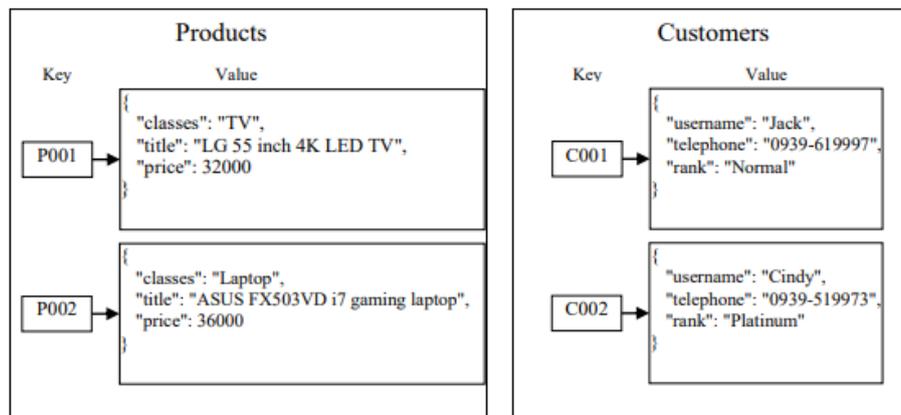


Fig. 3.2: Ejemplo de dos contenedores en una base de datos Clave-Valor

Fuente: [Chen & Lee, 2019]

Las características esenciales de las bases de datos clave-valor son:

- **Simplicidad:** Las bases de datos clave-valor utilizan una estructura de datos muy sencilla.
- **Velocidad:** Esta es una de sus principales características, debido a su simple estructura de datos pueden ofrecer operaciones de alto rendimiento en poco tiempo.
- **Escalabilidad:** Dado que su manejo es muy sencillo, mientras se posea suficiente espacio la base de datos puede escalar tanto como sea necesario.

Orientadas a documentos

Una base de datos de documentos es un tipo de base de datos no relacional que está diseñada para manipular información orientada a documentos o datos semi-estructurados. Este tipo de bases de datos constituye una de las principales categorías de las llamadas bases de datos NoSQL. A diferencia de lo que se conoce como bases de datos relacionales con su definición de lo que sería una "tabla", los sistemas documentales están diseñados en torno a la definición abstracta de un "documento". Este tipo de bases de datos almacena la información mediante documentos (como se observa en la Fig. 4). Cabe aclarar que se habla de documentos no se refiere a documentos de texto, sino que se refiere a archivos XML¹⁰ o JSON¹¹ generalmente, aunque también se pueden utilizar otros tipos de archivos como YAML.

Una característica importante es que los documentos pueden tener embebidos documentos dentro de ellos. Esto brinda la posibilidad a los usuarios de guardar datos

¹⁰ Archivo de lenguaje marcado extensible

¹¹ JavaScript Object Notation - Formato ligero de intercambio de datos

relacionados a los documentos dentro del mismo documento y gracias a esto lograr una mejor performance de la base de datos.

A excepción de algunas, la mayor parte de estas bases de datos almacenan los datos como documentos con la notación de objetos de JavaScript¹² (JSON) y ofrecen diferentes API para varios lenguajes. Las razones por las que se utilizan son la velocidad y escalabilidad de este tipo de bases de datos esto se logra principalmente debido a la posibilidad de tener documentos embebidos en otros documentos. Entre los sistemas de bases de datos más utilizados se encuentran: MongoDB, CouchDB y RavenDB

```
Customer Document
"customer" =
{
  "id": "Customer:1",
  "firstName": "John",
  "lastName": "Wick",
  "age": 25,
  "address": {
    "country": "US",
    "city": "New York",
    "state": "NY",
    "street": "21 2nd Street",
  },
  "hobbies": [ Football, Hiking ],
  "phoneNumbers": [
    {
      "type": "Home",
      "number": "212 555-1234"
    },
    {
      "type": "Office",
      "number": "616 565-6789"
    }
  ]
}
```

Fig. 4: Ejemplo de un documento que contiene otros embebidos
Fuente:[<https://www.alachisoft.com/nosdb/document-databases.html>]

La popularidad del término "base de datos orientada a documentos" ha crecido a la par del término NoSQL en sí ya que se puede considerar como la opción de bases de datos no relacionales más utilizada. Las bases de datos de documentales son consideradas como un escalón superior ante las bases de datos de clave-valor, puesto que permiten encapsular pares de clave-valor en estructuras más complejas.

Orientadas a columnas

Las bases de datos orientadas a columnas son probablemente conocidas por la aplicación de estas en *BigTable*¹³ de Google o por la implementación en Cassandra¹⁴ de Apache. A primera vista tienen muchas similitudes con las bases de datos relacionales, pero en realidad son muy diferentes. Una de las principales diferencias entre estas radica en que en el sistema relacional el almacenamiento de datos es por filas mientras que en una base de datos orientada a columnas, es por columnas. La otra diferencia es que las orientadas a columna tienen optimización en las consultas para mejorar los tiempos de respuesta en comparación con los sistemas relacionales.

¹² <https://www.javascript.com/>

¹³ <https://cloud.google.com/bigtable/docs/overview>

¹⁴ https://cassandra.apache.org/_/index.html

Las bases de datos orientadas a columnas son tablas de datos donde las columnas de valores de datos representan el almacenamiento estructural. En estas los datos son almacenados como secciones de las columnas de datos en lugar de filas de datos, como en la mayoría de los gestores relacionales. En la Fig. 5 se puede observar una comparación de almacenamiento de datos entre una base de datos relacional y una orientada a columnas sobre un mismo conjunto de datos.

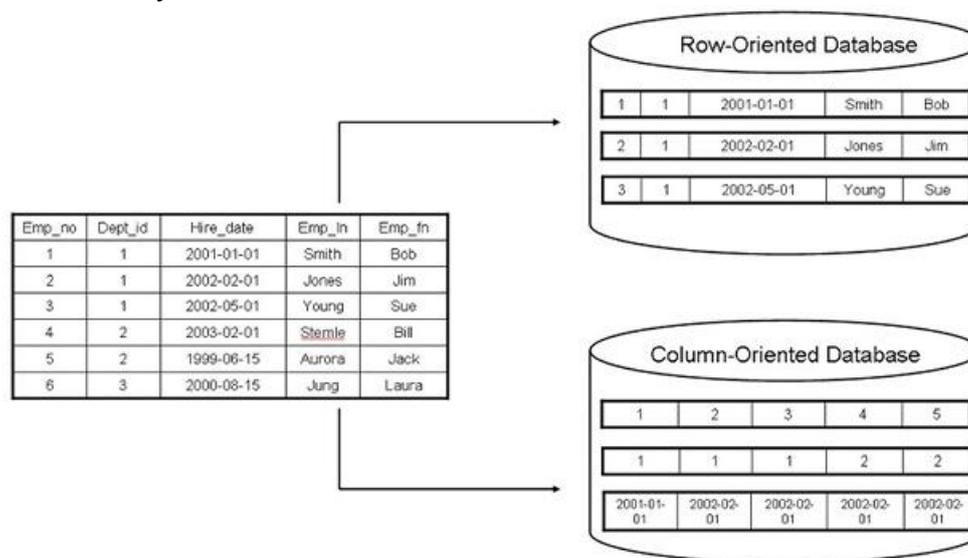


Fig. 5: Ejemplo de cómo se almacenaría los datos en una base de datos relacional y en una orientada a columnas

Fuente: <https://www.dbbest.com/blog/column-oriented-database-technologies/>

Una de las razones por las que se usa el almacenamiento basado en columnas es para reducir el coste de operaciones E/S del disco. Y esta es la razón por la que estas bases de datos son utilizadas aplicaciones analíticas, ya que permiten guardar un gran volumen de datos a gran velocidad y facilidad de recuperar estos datos.

Es muy importante entender que las columnas pueden variar entre filas, es decir, puede haber columnas que tienen un valor para un registro mientras que para otro tiene otro valor o incluso puede ser nula y no existir.

Los principales beneficios que ofrecen las bases de datos orientadas a columnas son:

- Alto rendimiento en las consultas de agregación (como COUNT, SUM, AVG, MIN, MAX)
- escalabilidad y carga rápida de datos para grandes volúmenes de datos.

Debido a su alto rendimiento en consultas de agregación, las bases de datos orientadas a columnas ofrecen ventajas clave para la inteligencia de negocios (Business intelligence) y el análisis de grandes cantidades de datos .

Orientadas a Grafos

Para entender este tipo de bases de datos es importante comprender que un grafo es un conjunto de objetos llamados nodos y que están unidos entre ellos mediante lo que se llaman arcos o aristas, gracias a esto se consigue representar relaciones entre estos nodos. Por lo tanto, este tipo de base de datos recibe este nombre ya que la información que de la

misma se representa como nodos de un grafo que serían las entidades y las relaciones mediante aristas, de esta forma como la base de datos es un grafo como tal, se puede utilizar la teoría de grafos para recorrerla.

Por definición, una base de datos orientada a grafos es cualquier sistema de almacenamiento que permite la adyacencia libre de índice. Esto quiere decir que cada elemento (ya sea un nodo o una relación) contiene un puntero directo a sus elementos “adyacentes” por lo que no sería necesario realizar consultas por sus índices.

Este tipo de base de datos está diseñada para los datos cuyas relaciones están bien representadas en forma de grafo, o sea, los datos son elementos interconectados con un número no determinado de relaciones entre ellos. La Fig. 6 ilustra un ejemplo de representación de una base de datos en un grafo.

No existe un consenso general sobre la terminología existente en el área de grafos pues hay muchos tipos diferentes de modelos de grafos. Sin embargo, se están realizando algunos esfuerzos para crear el Modelo de Grafo de Propiedad, que unifica la mayoría de las diferentes implementaciones de grafos. De acuerdo con este Modelo, la información en un grafo de propiedad se modela utilizando tres elementos básicos:

- El nodo (vértice)
- La relación (arista) con dirección y tipo (etiquetado y dirigido)
- La propiedad (atributo) en los nodos y en las relaciones

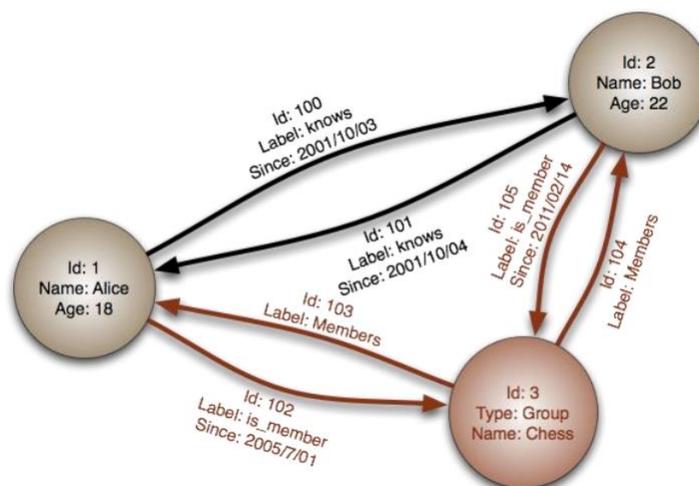


Fig. 6: Representación de la estructura de una base de datos orientada a grafos
Fuente: [Bases de datos NoSQL, Telemática]

Las consultas que se realizan en este tipo de bases de datos son más rápidas que en las bases de datos relacionales, esto es debido a que no se realizan uniones para cruzar información de las tablas, sino que se siguen las aristas (las relaciones) de nodo a nodo, que es una operación mucho más sencilla y menos costosa.

NewSQL

Como mencionamos anteriormente, una base de datos que utiliza herramientas de diseño NewSQL busca combinar la escalabilidad, disponibilidad y flexibilidad de NoSQL con la fuerte consistencia y usabilidad de las bases de datos relacionales. Surgió con el objetivo

de tratar información de compañías que son resguardadas con la seguridad de una base de datos SQL sumado a la capacidad de que esta información sea escalable y flexible. Por otro lado, estas nuevas bases de datos suelen pasar por alto propiedades consideradas importantes como la consistencia e integridad. El término NewSQL es reciente, fue utilizado por primera vez en 2011 para comenzar una nueva era de bases de datos diferenciada de las ya establecidas por grandes compañías (Oracle, IBM, Microsoft y otros). Debido a esto es que aún está en discusión cuáles tecnologías aplican al término y cuáles no, ya que ha sido referenciado incorrectamente en algunos casos. Esto se suma a la escasa información disponible de las mismas en relación con las demás, por lo que la comparación realizada en el informe con respecto a NewSQL será en base a definiciones y descripciones otorgadas en los informes referenciados en la bibliografía.

Aunque es un término reciente, las bases de datos en las bases de datos NewSQL se pueden identificar características en común:

- Modelo de datos relacional y estándar SQL.
- Transacciones ACID.
- Escalabilidad con el uso de partición de datos y nodos independientes.
- Disponibilidad con el uso de replicación de datos.

Las bases de datos NewSQL por lo general utilizan una arquitectura de aplicación HTAP (Transacción Híbrida/Proceso Analítico). Es un término al cual se asocian aquellas bases de datos cuyo mayor objetivo es ejecutar OLAP(sistema de procesamiento analítico en línea) y OLTP (sistema de procesamiento de transacciones en línea) sobre el mismo conjunto de datos. OLAP incluye aquellas operaciones orientadas al procesamiento analítico, implicando la lectura de grandes cantidades de datos para obtener información útil para el usuario. OLTP por su parte se orienta en el procesamiento atómico, más típico en las bases de datos operacionales, implicando operaciones de inserción, modificación o borrado. Las bases de datos HTAP pueden realizar un análisis en tiempo real de los datos, evitando la distinción entre las bases de datos operativas y de almacenamiento de datos [Schreiner et al., 2019].

Actualmente algunos sistemas híbridos que pueden considerarse NewSQL son Google F1, LeanXcale, Apache Ignite, CockroachDB, Esgyn, GridGain, MemSQL, Nuodb, Splice Machine, VoltDB y SAP HANA.

Bases de datos orientadas a objetos

Una base de datos orientada a objetos se basa en que toda su información esté representada con objetos. Estas bases de datos ofrecen la posibilidad de tener un acceso directo a los objetos definidos dentro del lenguaje de programación en cuestión y guardarlos dentro de la base de datos como un objeto sin tener que hacer una conversión para esto y de la misma forma poder recuperarlos desde la base de datos. Esto es una gran diferencia con los sistemas de bases de datos relacionales donde todo debe ser estructurado en tablas. Tanto objetos simples como complejos pueden ser guardados en estos tipos de bases de datos. Una característica de las mismas es la capacidad de poder definir tipos de datos abstractos que permiten al usuario definir características a los datos al momento de que estos sean guardados en la base de datos. Es muy importante entender la capacidad de este tipo de bases de datos ya que las bases de datos relacionales solo pueden guardar

datos en columnas de tablas mientras que las bases de datos orientadas a objetos los guardan directamente como un objeto que puede tener muchos atributos distintos como también instrucciones (métodos encapsulados) [Thearle, 2019].

Principales características de las bases de datos orientadas a objetos:

- Identidad de objeto: En una base de datos orientada a objetos, los objetos tendrán una identidad propia que será un valor diferente para cada objeto y con el que se podrá diferenciarlo de cualquier otro.
- Encapsulación: En una base de datos orientada a objetos, se tiene la capacidad de ocultar los detalles de implementación de los objetos, haciendo que un objeto sea accesible mediante su interfaz pública.
- Jerarquía de clases: Permite organizar los objetos en clases, y definir otras nuevas clases especializando o generalizando en otras ya existentes.
- Polimorfismo: Distintas clases podrán proporcionar distintos métodos para una misma operación; el sistema determinará dinámicamente qué método se debe ejecutar.
- Extensibilidad: Cada sistema de bases de datos orientado a objetos tiene un conjunto de tipos predefinidos por el sistema. Todos los tipos de datos definidos por el sistema y los tipos definidos por el usuario puede ser extendido.
- Tipos y clases: Un sistema de bases de datos orientado a objetos permite definir tanto tipos de datos como tipos de objetos.

Desventajas de las bases de datos orientadas a objetos:

- Carencia de un modelo de datos universal: No hay ningún modelo de datos universalmente aceptado para los SGBDOO (sistema gestor de base de datos orientado a objetos) y la mayoría de los modelos carecen de una base teórica.
- Carencia de estándares: Existe una carencia de estándares generales para los SGBDOO.
- La optimización de consultas compromete la encapsulación: La optimización de consultas requiere una comprensión de la implementación de los objetos, para poder acceder a la base de datos de manera eficiente. Sin embargo, esto compromete el concepto de encapsulación.
- El modelo de objetos aún no tiene una teoría matemática coherente que le sirva de base:

Proceso ETL (extraer, transformar y cargar)

ETL es el proceso responsable de la extracción e integración (transformar y cargar) de los datos tomados desde una o múltiples fuentes o aplicaciones. Este proceso se refiere a el almacenamiento de datos extraídos desde una o más fuentes de datos, los cuales se transforman a las necesidades operativas de como se los va a necesitar, esto puede incluir controles de calidad, limpieza de los datos, personalización, cambio de formato o integración para por último ser cargados en la base de datos de destino final [Díaz, Osorio & Amadeo, 2019]. El proceso ETL constituye uno de los procesos más importantes de los Data Warehouse [Vlahu-Gjorgievska & Trajkovik, 2019].

La extracción es el primer paso a realizar dentro de este proceso de cargar los datos en el entorno del almacenamiento de datos. Luego, se comienza con el siguiente paso, el cual es manipular los datos (Transformar) a través de diferentes técnicas y metodologías. Por último, la etapa final del proceso de ETL (carga) es la estructuración física y la carga de los datos según el modelo dimensional definido en el área de representación. Esta etapa es fundamental porque es donde los datos se cargan a la base de datos y estos deberían ser correctos para evitar las fallas o errores en el posterior tratamiento de los mismos.

ELK: ElasticSearch, Logstash y Kibana

ELK¹⁵ se trata de un stack formado por tres herramientas independientes: ElasticSearch, Logstash¹⁶ y Kibana. Estos proyectos combinados forman una poderosa herramienta para la búsqueda, análisis y visualización de datos en tiempo real. Elasticsearch es un motor de búsqueda y analítica. Logstash es un pipeline de procesamiento de datos del lado del servidor que ingesta datos de una multitud de fuentes simultáneamente, los transforma y luego los envía a un "escondite", como Elasticsearch. Kibana permite a los usuarios visualizar los datos en cuadros y gráficos con Elasticsearch. En la Fig. 7 se observa la relación entre las herramientas que componen el stack.



Fig 7: Stack ELK

Fuente: <https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/>

ElasticSearch

Se trata de un motor de analítica y análisis distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados. Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad, Elasticsearch es el componente principal de ELK.

Los datos sin procesar fluyen hacia Elasticsearch desde una variedad de fuentes. La ingesta de datos es el proceso mediante el cual estos datos son parseados, normalizados y enriquecidos antes de su indexación en Elasticsearch. Una vez indexados en Elasticsearch, los usuarios pueden ejecutar consultas complejas sobre sus datos y usar agregaciones para recuperar resúmenes complejos de sus datos. Se dice que es un motor de búsqueda documental ya que almacena los datos como documentos JSON. Cada documento correlaciona un conjunto de claves con sus valores correspondientes. A su vez, los llamados índices relacionan documentos creando una lista de cada palabra única que aparece en cualquiera de ellos e identifica todos aquellos en que ocurre cada palabra para poder buscar datos casi en tiempo real.

¹⁵ <https://www.elastic.co/es/what-is/elk-stack>

¹⁶ <https://www.elastic.co/es/logstash/>

Logstash

Logstash, uno de los productos principales del Elastic Stack, se usa para agregar y procesar datos y enviarlos a Elasticsearch. Logstash es una pipeline de procesamiento de datos open source y del lado del servidor que te permite ingresar datos de múltiples fuentes simultáneamente y enriquecerlos y transformarlos antes de que se indexen en Elasticsearch.

Inteligencia de negocios

La inteligencia de negocios (Business Intelligence. BI) se define como un proceso de toma de decisiones respaldado por la integración y análisis de los datos de una organización. De hecho, la inteligencia de negocios juega un papel cada vez más crítico en la mayoría de las compañías porque, con el correr del tiempo se ha identificado a la información como el activo más valioso de una empresa, es considerado como un recurso fundamental para el desarrollo de la misma.

Por otro lado, la inteligencia de negocios es un término que incluye las aplicaciones, la infraestructura, las herramientas y las mejores prácticas que permitan el acceso y análisis de la información para mejorar y optimizar las tomas de decisiones y el rendimiento de una organización.

Big data

Uno de los términos comprendidos dentro de la inteligencia de negocios es Big Data que son grandes volúmenes de datos, que tienen una gran diversidad y complejidad los cuales para ser manejado de manera eficiente y económica escapan a las soluciones convencionales.

Cuando se habla de Big Data se suele referir a un conjunto de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, dentro del tiempo necesario para que estos sean útiles.

Visualización de datos

La visualización de datos es el cuidado y manipulación de datos recolectados que se muestran e informan al lector. Dado que existen varias técnicas de visualización diferentes para comunicar eficazmente los datos al lector, el formato de la información mostrada varía. Algunos ejemplos de técnicas de visualización de datos pueden ser gráficos de línea, gráficos de barra, gráficos de torta o gráficos de dispersión [Faronius Hofmann & Håkansson, 2021].

Estas técnicas de visualización utilizan gráficos contruidos a partir de datos que gracias a que las computadoras transforman patrones, tendencias, relaciones y composiciones de los datos en gráficos que son más fáciles de entender y procesar por el ojo humano. Pero estas técnicas de visualización no solo comprenden los diferentes tipos de gráficos o tablas, sino que también se encargan de la elección de colores, animaciones,

fuentes de texto. Entre otras. Ya que estas pueden usarse para aclarar el contenido de los datos.

Para mostrar gráficos de presentación, el uso del tipo correcto de gráficos es de gran importancia. El término gráfico se refiere a una presentación visual de datos mediante el uso de curvas o líneas o puntos presentados, por lo general, en un plano bidimensional. Un gráfico puede dividir fácilmente datos brutos y números exactos en categorías o subconjuntos de datos. Los gráficos se utilizan para presentar resultados [Chen et al., 2008].

Estética de las visualización y tipos de datos

La estética describe todos los aspectos de un elemento de una visualización de datos. Un componente crítico de cada elemento en una visualización es su posición, que describe dónde se encuentra el elemento.

Una visualización de datos siempre contiene por lo menos las siguientes características: forma, color y tamaño. Incluso si estamos preparando un dibujo en blanco y negro, Los elementos deben tener un color para ser visibles: por ejemplo, negro si el fondo es blanco o blanco si el fondo es negro.

Hay muchas otras características estéticas que se pueden encontrar en una visualización de datos. Por ejemplo, si se quiere mostrar texto, es posible que se tenga que especificar la fuente y tamaño del texto, por otro lado, si los objetos a mostrar en la visualización se superponen, es posible que se deba especificar que estos sean parcialmente transparentes para así enfatizar las zonas en las que se superponen mostrando el color en un tono más fuerte.

Habilidades de percepción de los lectores

Las visualizaciones de datos recaen en la percepción visual del ser humano para comunicar lo que muestran. La percepción visual es especialmente adecuada para transmitir la información que se muestra, ya que proporciona un gran canal de comunicación con el cerebro de las personas. Las capacidades visuales del ser humano tienen sus limitaciones las cuales deben tenerse en cuenta al realizar las visualizaciones. Estas limitaciones son la principal dificultad de las visualizaciones de datos.

Elegir el gráfico adecuado

Una tarea muy importante de la visualización de datos es la correcta elección tipo de visualización a mostrar, ya que si esta visualización es correcta transmitirá efectivamente el mensaje, mientras que una visualización incorrecta confundirá al lector o transmitirá un mensaje incorrecto

Para todos los conjuntos de datos, hay más de un gráfico que puede ser usado para ayudar al lector a interpretar los datos. Sin embargo no todos los gráficos son adecuados para todos los tipos de datos, dependiendo de los datos que se tienen y lo que se quiere

mostrar los distintos tipos de gráficos son mejores que otros. El tipo de gráfico apropiado para un determinado tipo de datos depende, entre otras cosas, de si los datos son cualitativos o cuantitativos.

La importancia de elegir un correcto diseño de gráfico

Es muy importante elegir un gráfico y un diseño adecuados para que el lector comprenda y confíe en los datos visualizados. La importancia de las decisiones de diseño recae ya que la credibilidad de la visualización se verá afectada por las decisiones de diseño tomadas. El formato visual, la interactividad (como por ejemplo el zoom), la calidad del diseño (por ejemplo color, fuente) y la comprensión de los datos afecta a la credibilidad de un gráfico.

La interactividad da al usuario la posibilidad de explorar la visualización de datos e interactuar con ella, lo cual provee al lector detalles específicos de los datos mostrados en la visualización. Esta es la gran ventaja de una visualización de datos vista desde una computadora contra verla desde un papel.

Las decisiones de diseño de una visualización de datos también afectarán su credibilidad. En este caso, la calidad del diseño puede referirse a una visión objetiva y subjetiva de un individuo sobre lo que es un buen diseño y lo que no lo es. Las reacciones de las personas frente a una visualización de datos a menudo provienen de la claridad de los datos y la complejidad del mismo.

Es de suma importancia en las visualizaciones de datos la comprensión de los datos que se están mostrando. El lector debe comprender fácilmente lo que muestra la visualización de datos y donde encontrar un valor de datos específico

Decisiones de diseño en las visualizaciones de datos

Las decisiones de diseño deben ser tomadas cuando se está construyendo el gráfico. Atributos como marcas de graduación, color, tamaño, título, subtítulo, etiquetas, fuente, forma, leyendas y ancho y tipo de línea pueden beneficiar y aumentar la percepción de datos y la credibilidad de una visualización de datos. Además, el orden y la categorización de los datos también afectan a la visualización de datos. Estos atributos se describen en las siguientes secciones:

Título, subtítulos y etiquetas

Aunque una imagen vale más que mil palabras, en la visualización de datos existe la necesidad de agregar títulos, subtítulos o etiquetas que hagan que la visualización sea más sencilla de interpretar por parte del lector.

Al agregar palabras como complemento a una visualización de datos, debe hacerse con cuidado. Ya que si se agregan demasiadas etiquetas y subtítulos estas pueden distraer al lector de los datos reales y dar una impresión de desorden. Aunque por el otro lado la

carencia de etiquetas y subtítulos puede hacer que el entendimiento de la visualización de datos se dificulte.

Marcas de graduación

Un atributo muy importante de todos los gráficos son las marcas de graduación y las marcas de valores en los ejes. Las marcas de graduación son pequeñas líneas en el eje x o en el eje y que marcan un valor determinado. Los valores y las marcas deben resaltar los puntos interesantes sin hacer que estas etiquetas se vean borrosas. Las marcas de graduación deben ser colocadas cuidadosamente en sus correspondientes posiciones.

Leyendas

Las leyendas describen qué colores y/o símbolos corresponden a un cierto tipo de datos. Estas ayudan a las personas a leer las visualizaciones de datos haciéndolas más explicativas. Las leyendas deben mostrarse directamente en la visualización en lugar de separada de él para que el lector se pueda concentrar en la visualización de datos.

Color

Al personalizar la visualización de datos los colores y las sombras son una de las formas más efectivas de personalizar la visualización. Los colores deben usarse cuando se muestran diferentes datos en una visualización de datos. El uso de colores en las visualizaciones hace que estas sean mucho más efectivas al transmitir la información, ya que una visualización colorida es más memorable.

Tamaño y fuente

Las visualizaciones de datos deben ser lo suficientemente grandes para que el lector vea la información en ellos claramente, pero no mucho más grande que lo necesario para cumplir esta condición. Los marcos y bordes de la visualización sólo deben usarse con el propósito de separarla, (de por ejemplo, separar la visualización de otra visualización o texto) estos marcos y bordes deben ser lo más delgados posible, pero aún visible para cumplir su propósito.

Forma

Los datos de la visualización pueden tener diferentes formas, por ejemplo, círculos o barras. Las formas se pueden utilizar de varias formas diferentes para manipular los datos durante la visualización. Las formas pueden ser como lo son en la vida real, como por ejemplo países en un mapa. También se puede utilizar para diferenciar entre series de datos o ilustrar un punto en un eje.

Grafos sociales

Boyd y Ellison definen, y de forma aceptada, los grafos de redes sociales (grafos sociales): Una red social es un grupo de personas, por ejemplo, amigos, conocidos o compañeros de trabajo. Los cuales están relacionados entre sí por relaciones interpersonales. Este concepto se remonta al principio de la historia de la humanidad, pero en la última década, el uso de servicios en línea y el uso generalizado de computadoras, tabletas y celulares provocó un crecimiento exponencial de las redes sociales tanto en su escala como en sus propósitos [Boyd & Ellison, 2007].

Se diferencian de otras estructuras de red (biológicas, de transporte y de telecomunicaciones, por citar algunas) debido a la presencia de correlaciones de grados denominadas asortatividad (La asortatividad es la preferencia de los nodos de una red por unirse a otros que le son similares en alguna característica) [Ianni et al., 2020]. De hecho, en las redes sociales el comportamiento similar de agruparse se ha definido como homofilia, lo que implica, desde un punto de vista topológico, que hay muchas aristas dentro de un grupo de personas similares en la red como se muestra en la Fig. 8. Esto tiende a darse en redes sociales como Facebook¹⁷ o Instagram¹⁸.

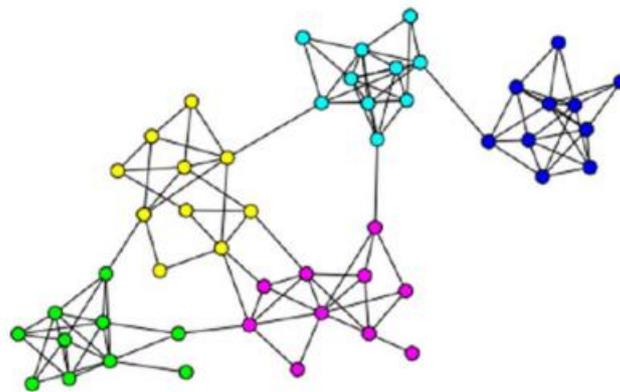


Fig 8.1: Grafo que presenta asortatividad (homofilia).
Fuente: [Ianni et al., 2020]

Sin embargo, también existen redes desasortativas cuya topología se caracteriza por bordes entre grupos como se informa en la Fig 8.2. En esta se puede citar como ejemplo a Tinder¹⁹, donde en general las personas tienden a establecer un vínculo con personas de un grupo diferente.

¹⁷ <https://www.facebook.com/>

¹⁸ <https://www.instagram.com/>

¹⁹ <https://tinder.com/>

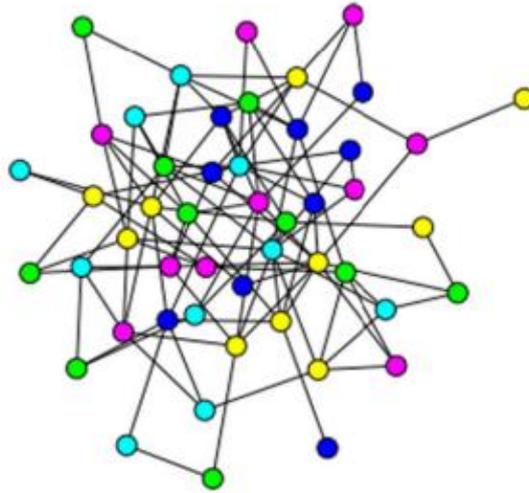


Fig 8.2: Grafo que presenta desasortividad.
Fuente: [Ianni et al., 2020]

Capítulo 4: Comparaciones de herramientas y Experiencia de Uso

Comparaciones entre bases de datos SQL y NOSQL

Facilidad de entendimiento y uso

SQL debido a su forma de ser estructurado y estar basado en conceptos matemáticos como lo son el álgebra relacional y teoría de conjuntos, se hace mucho más fácil el aprendizaje de la utilización del mismo al momento de saber cómo escribir una consulta en una base de datos SQL. Esto representa una clara ventaja, aunque no muy importante, de las bases de datos SQL sobre las bases de datos NoSQL para los desarrolladores que sean nuevos en el mundo de las bases de datos. Esta brecha de facilidad de entendimiento y uso se ha estado achicando en el último tiempo debido a que por su gran difusión se han hecho algunos proyectos para encapsular sus funcionalidades y hacerlas más amigables a desarrolladores.

Escalabilidad horizontal y habilidad de distribución

En las bases de datos NoSQL existe una gran facilidad para añadir, eliminar o realizar operaciones con sus elementos sin afectar a su rendimiento. Gracias a esta escalabilidad se puede realizar réplicas y distribuir datos de manera más sencilla sobre servidores. Este almacenamiento distribuido aumenta la fiabilidad y escalabilidad, pero en contraposición hace que los costos aumenten.

Teorema CAP

Eric Brewer en el año 2000 propuso una percepción sobre los entornos distribuidos en la que comenta que estos no pueden garantizar de manera simultánea una consistencia, disponibilidad y tolerancia a particiones. La Fig. 9 ilustra una representación gráfica del teorema.

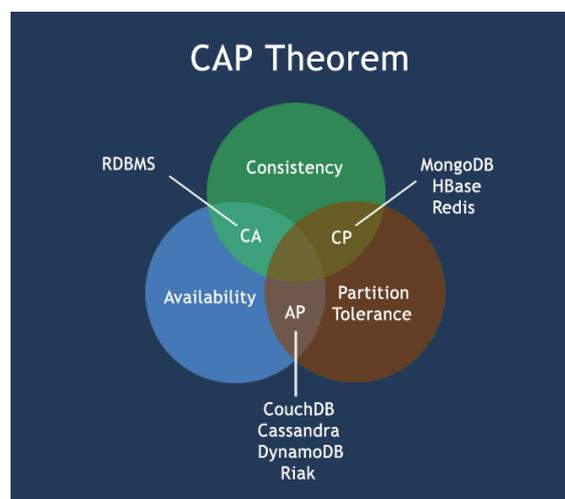


Fig. 9: Representación del teorema CAP y que Sistemas de bases de datos están en cada uno
Fuente:[<http://satasuarez.blogspot.com/2013/08/teorema-de-cap.html>]

- Consistencia: Cuando se realiza una consulta o una inserción en la base de datos, siempre se tiene que recibir la misma información independientemente del nodo o del servidor desde que se realiza. Es decir, los usuarios tienen la posibilidad de acceder al mismo registro de manera simultánea.
- Disponibilidad: Es la garantía de que cada solicitud a un nodo reciba una respuesta de si ha sido o no resuelta satisfactoriamente.
- Tolerancia a particiones: Como los sistemas distribuidos están divididos en particiones, el sistema debe continuar funcionando en el caso de que se produzca una caída o un fallo parcial en el sistema.

Para ser distribuidas y escalables, las bases de datos no pueden cumplir las tres características simultáneamente.

Condiciones que se cumplen en el teorema de CAP :

- AP: Garantiza la disponibilidad y la tolerancia a particiones, pero no la consistencia de manera total. Esto quiere decir que no se puede garantizar que los datos sean iguales en todos los nodos todo el tiempo.
- CP: Garantiza la consistencia y la tolerancia a particiones, pero para ello hay que sacrificar la disponibilidad. Aunque se puede alcanzar un buen nivel del mismo.
- CA: Garantiza la consistencia y la disponibilidad, debido a esto surgen problemas con la tolerancia a particiones ya que se garantiza que los datos son iguales siempre; y que el sistema estará disponible respondiendo a todas las peticiones.

Comparación entre ACID (SQL) y BASE (NoSQL)

Las bases de datos relacionales se apoyan en ACID para la manipulación de datos, es decir que las transacciones poseen las siguientes características:

- Atomicidad (**Atomicity**): Todas las transacciones relacionadas se consideran conjuntamente, si alguna falla al ejecutarse, todas se cancelan.
- Consistencia (**Consistency**): La ejecución de una transacción aislada conserva la consistencia de la base de datos.
- Aislamiento (**Isolation**): Cada transacción ignora al resto de las transacciones que se ejecutan concurrentemente en el sistema
- Durabilidad (**Durability**): Al finalizar una transacción, los cambios perduran en el tiempo aunque haya un posible fallo posterior en el sistema.

Las bases de datos no relacionales están basadas en BASE para la manipulación de datos, es decir que las transacciones poseen las siguientes características:

- Básicamente Disponible: (**Basically available**) Este término hace referencia a que las operaciones de lectura y escritura están disponibles para usarse tanto como sea posible, pero sin una eventual consistencia por lo que el dato podría no persistir en la base de datos (ya sea porque se genero un conflicto en una escritura o que el dato que se leyó no sea realmente el mas actualizado)
- Estado suave: (**Soft-state**) Este término hace referencia a que debido a que no se puede garantizar la consistencia de los datos en este tipo de bases de datos solo se

puede decir que el dato esté probablemente en el estado que está reflejado en la base de datos.

- Eventualmente consistente: (*Eventual consistency*) Este término hace referencia a que en algunos momentos la base de datos puede tener un estado inconsistente.

Estandarización

Al ser términos recientes, NoSQL y NewSQL no están estandarizados, es decir que no existe un patrón totalmente definido para comparar o valorar estos tipos de modelos. Entre todos los modelos, SQL lleva más tiempo en vigencia por lo que tiene más herramientas e información disponible en comparación con los otros modelos. Debido a la ventaja en tiempo de SQL, los programadores se encuentran familiarizados con este entorno.

Por otra parte y relacionado con la estandarización y el tiempo de vida, el soporte para el modelo SQL es mayor, por lo que los usuarios se encuentran más seguros al trabajar en este entorno.

Popularidad

La popularidad de las opciones en el mercado de las bases de datos es algo a tener en cuenta. Aunque no afecta al rendimiento directamente, el grado de popularidad influye en la comunidad de programadores en términos de obtener soporte entre los mismos.

En la Fig. 10.1 se puede observar un gráfico de popularidad de las bases de datos a la fecha (Noviembre 2020). En ella se destacan ampliamente las bases de datos SQL por sobre las demás, aunque las NoSQL están adquiriendo mayor popularidad en los últimos años. Mientras que en la Fig. 10.2 Se muestra una comparación entre las búsquedas de SQL y NoSQL en Google.

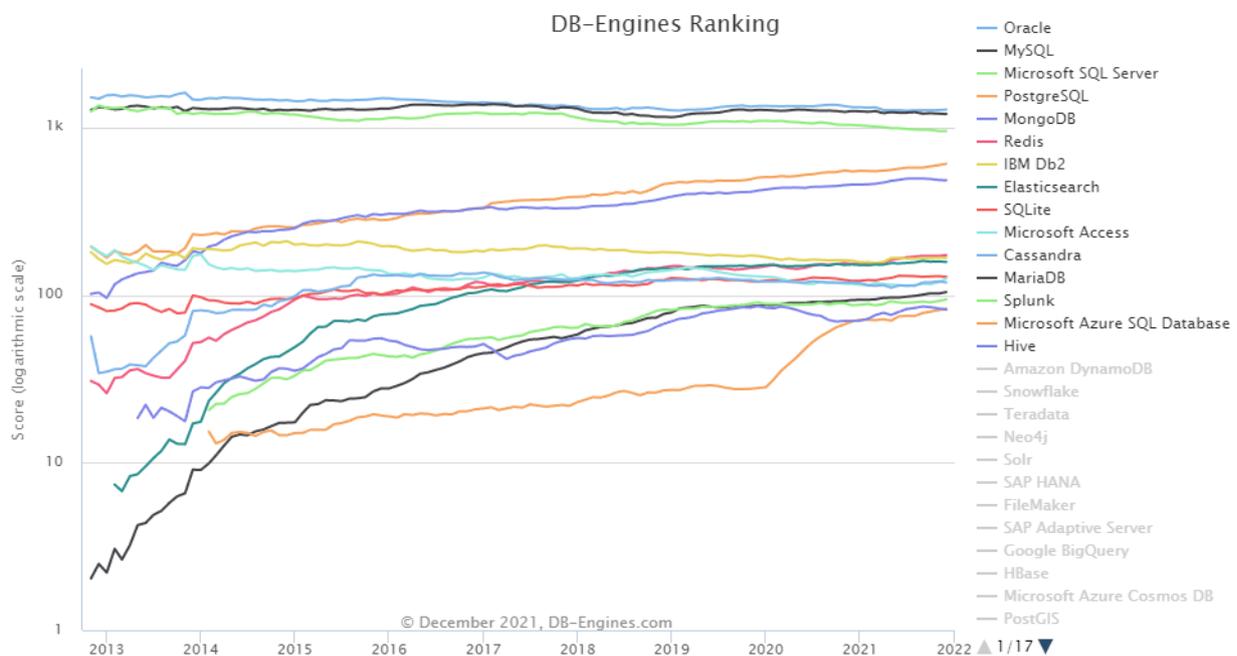


Fig. 10.1: Ranking de popularidad de DB-Engines
Fuente: https://db-engines.com/en/ranking_definition

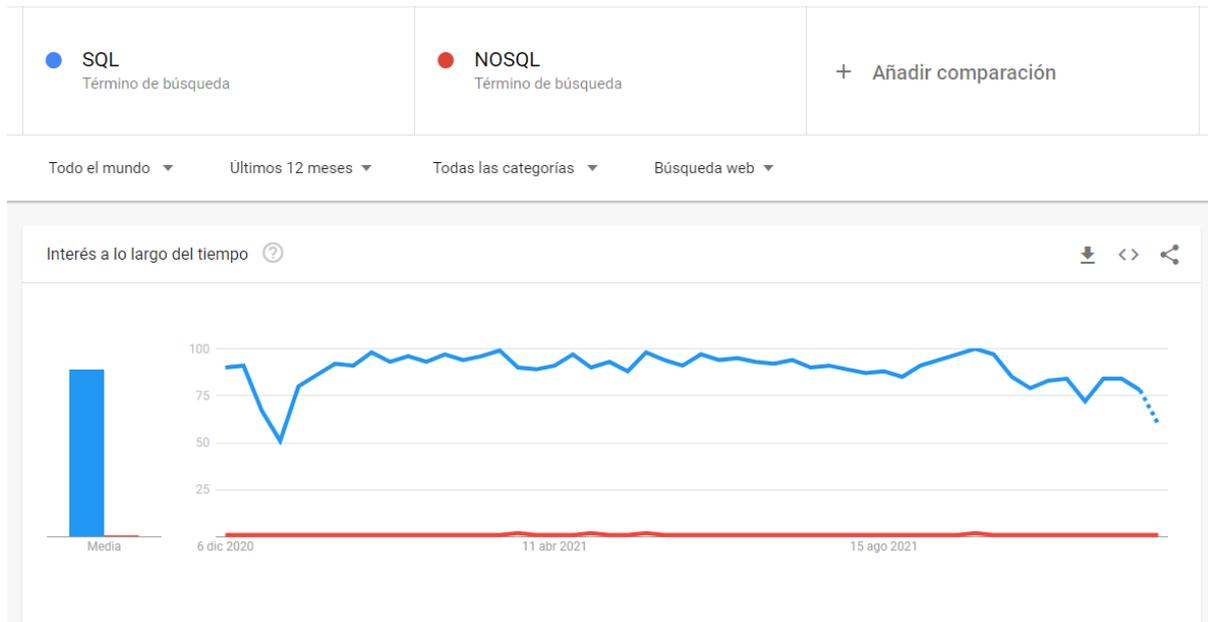


Fig. 10.2: Comparación de cantidad de búsquedas de los SQL y NOSQL en Google
Fuente: <https://trends.google.es/trends/explore?q=SQL,NOSQL>

Tabla de capacidades

	SQL	NoSQL Clave-valor	NoSQL Documentos	NoSQL Columnas	NoSQL Grafos	NewSQL	Orientada a objetos
Estandarización	Estandarizada	No-Estandarizada	No-Estandarizada	No-Estandarizada	No-Estandarizada	No-Estandarizada	Estandarizada
ACID/BASE	ACID	BASE	BASE	BASE	BASE	ACID	ACID
Teorema de CAP	CA	AP	CP	AP	AP	CA	CA
Bases de datos distribuidas	Permitido	Permitido	Permitido	Permitido	Permitido	Permitido	Permitido
Eficiencia para grandes volúmenes de datos	Media	Alta	Alta	Alta	Alta	Media	Media
Fundamentación matemática	Si	No	No	No	No	Si	No
Integridad referencial	Si	No	No	No	Si	Si	No
Flexibilidad y descentralización	Baja	Alta	Alta	Alta	Alta	Alta	Media
Escalabilidad	Alta pero costosa	Alta	Alta	Alta	Alta	Alta	Alta
Atomicidad de transacciones	Si	No	No	No	No	Si	Si
Schema-less	No	Si	Si	Si	Si	No	No
Dificultad de uso	Baja	Baja	Media	Media	Alta	Media	Media
Simplicidad	Media	Alta	Media	Media	Baja	Media	Media
Facilidad de aprendizaje	Muy Alta	Media	Alta	Media	Baja	Baja	Media
Soporte	Muy Alto	Medio	Alto	Bajo	Muy bajo	Bajo	Bajo

Tabla 1: Capacidades de las distintas bases de datos

Conclusión de la comparación entre bases de datos SQL y NOSQL

En base a la comparación realizada, se llegó a la conclusión de que es conveniente la utilización de una base de datos SQL por sobre una NOSQL. Las razones de la elección se listan a continuación:

- Las bases de datos SQL llevan mayor tiempo en el mercado, por lo que tienen un mayor soporte y una comunidad más consolidada.
- SQL posee atomicidad de operaciones en la base de datos. Esto es, que al realizar una operación, ésta se ejecuta en su totalidad o no lo hace, brindando una mayor seguridad en integridad de los datos. Esto lo logra utilizando la técnica del rollback.
- Los datos de las bases SQL deben cumplir requisitos de integridad y que el lenguaje de consultas SQL está basado en el álgebra relacional, se puede lograr un dinamismo en las consultas a partir de variables que ayudarán en gran medida a la construcción de la aplicación.
- La información a usar en la aplicación va a estar estructurada, por lo que es conveniente utilizar una base de datos SQL. Las bases de datos NoSQL carecen de esquema en los datos.
- Cabe marcar en contraposición la escalabilidad de las bases de datos SQL, que aunque probada en muchos entornos productivos suele, por norma, ser inferior a las bases de datos NoSQL.

Definición y aplicación del proceso ETL

Para probar las herramientas de visualización se inició un proceso ETL sobre un dataset del ministerio de la salud argentina "COVID-19. Casos registrados en la República Argentina"²⁰. Durante el proceso, se realiza dos transformaciones diferentes del archivo CSV²¹. Por un lado, se crean tablas de dimensiones y hechos para tener una base de datos normalizada y por el otro, se crea un archivo con datos en formato JSON para las bases de datos NoSQL. Para la aplicación del proceso ETL se usan las herramientas:

- Apache Spark
- PySpark

Apache Spark

*Apache Spark*²² es un framework de programación para procesamiento de datos distribuidos diseñado para ser rápido y de propósito general. Spark es la evolución natural de Hadoop²³, cuya funcionalidad es muy rígida y limitada en el sentido de que no aprovecha al máximo las capacidades del procesamiento distribuido. Algunas de las ventajas de Spark sobre Hadoop son el procesamiento en memoria que disminuye las operaciones de lectura/escritura, la posibilidad de análisis interactivo con SQL.

²⁰ <http://datos.salud.gob.ar/dataset/covid-19-casos-registrados-en-la-republica-argentina>

²¹ Comma Separated Values - Valores separados por coma

²² <https://spark.apache.org/>

²³ <https://hadoop.apache.org/>

Además, Spark permite trabajar con datos más o menos estructurados como lo son las RDDs, los data frames y los datasets, dependiendo de las necesidades y preferencias del usuario. Spark incluye procesamiento batch, algoritmos iterativos, queries interactivas, procesamiento streaming [Abuín et al., 2020].

RDD

Es una colección de elementos tolerantes a fallas que se pueden operar en paralelo. Hay dos formas de crearlos: paralelizar una colección existente en un controlador o hacer referencia a un conjunto de datos en un sistema de almacenamiento externo, como un sistema de archivos compartido, HDFS²⁴, HBase²⁵ o cualquier fuente de datos que ofrezca un formato de entrada Hadoop.

Data frame

Son estructuras heterogéneas de datos de dos dimensiones. Esto quiere decir que pueden contener datos de distintos tipos. Son las más utilizadas para realizar análisis de datos. Las filas representan casos, individuos u observaciones, mientras que las columnas representan atributos, rasgos o variables.

PySpark

Por último Pyspark es una API de Python para utilizarlo como Spark. Para poner un poco en contexto la ventaja de la utilización de esta herramienta, el dataset “COVID-19. Casos registrados en la República Argentina” tenía poco más de 2,5 gigabytes de datos en los que había alrededor de 11,2 millones de filas y para la aplicación del proceso ETL tardó aproximadamente 10 minutos²⁶, lo que resulta un tiempo considerablemente bajo ya que es una herramienta totalmente gratuita.

PySpark aparte de proveer métodos para tratar los RDDs y Data Frames, permite el uso del lenguaje SQL para realizar tareas de selección, filtrado, agrupación y joins.

Aplicación del proceso ETL en SQL

A continuación se detalla el uso de las herramientas seleccionadas para la aplicación del proceso ETL:

Extracción (del proceso ETL)

En la Fig. 11 se puede observar el código Python utilizado se puede observar el código implementado para la extracción de los datos del CSV del Ministerio de la Salud Argentina.

```
1 covid = sqlContext.read.format("csv").\  
2   options(header='true', inferSchema='true').\  
3   option("encoding", "UTF-8").\  
4   load("/content/drive/My Drive/ColabNotebooks/Tabi/Covid19Casos.csv")
```

Fig. 11: Código Python en el que se observa el proceso de extracción.

²⁴ Sistema de ficheros distribuido de Hadoop.

²⁵ <https://hbase.apache.org/>

²⁶ En un procesador Intel Core i5-6200 U 2.3 ghz. 8gb de RAM.

- En la 1er línea de código es importante remarcar que se especifica el formato del archivo a cargar
- En la 2da y 3er línea se definieron opciones, cómo la aparición de headers en el CSV (que la primera línea son los títulos de las columnas) y el formato de codificación de caracteres
- En la 4ta línea del código se define la ruta donde se encuentra el archivo dentro del Drive²⁷ que se utilizó.

Transformación (del proceso ETL):

En la Fig. 12.1 se observa cómo se construyó una tabla de dimensión.

```

1 age_dimension = covid.select('edad', 'edad_anios_meses').distinct()
2 w = Window.orderBy('id2')
3 age_dimension = age_dimension.withColumn("id2", F.monotonically_increasing_id())\
4   .withColumn("id", F.row_number().over(w))
5 age_dimension = age_dimension.select('id', 'edad', 'edad_anios_meses').\
6   withColumnRenamed('id', 'ageID').\
7   withColumnRenamed('edad', 'quantity').\
8   withColumnRenamed('edad_anios_meses', 'year_or_month')

```

Fig 12.1: Código Python de la transformación de los datos a una tabla de dimensión.

- En la 1ra línea se puede ver como se hizo un select distinct de las columnas con las que se quiso obtener del data frame “covid” y se lo guardó en una variable.
- En la 2da, 3er y 4ta línea se empleó una forma de asignarles “id” a las filas de la tabla, para esto se usó una de las funciones “Window” de Spark.
- Y luego en las líneas que siguen se hizo un select renombrando las columnas a como se quiere que se llamen.

Cómo se puede ver en la Fig. 12.2, lo que se hizo es construir la tabla del hecho para esto se usó pyspark SQL para escribir como si se estuviera haciendo una query SQL.

```

1 covid.registerTempTable("covid")
2 age_dimension.registerTempTable("age_dimension")
3 #fact_test
4 fact_test = sqlContext.sql(
5     "SELECT covid.id_evento_caso as caseID, genre.genreID AS genreID, age.ageID as ageID, loc.lo
6     "FROM covid " +
7     "INNER JOIN genre_dimension AS gend ON covid.sexo = gend.name " +
8     "INNER JOIN age_dimension AS age ON covid.edad = age.quantity AND covid.edad_anios_meses =
9     "INNER JOIN location_dimension AS loc ON covid.residencia_pais_nombre = loc.country AND cov
10    "LEFT JOIN date_dimension AS dsynto ON covid.fecha_inicio_sintomas = dsynto.date " +
11    "LEFT JOIN date_dimension AS dcs ON covid.fecha_apertura = dcs.date " +
12    "INNER JOIN cuidados_intensivos_dimension as cid ON covid.cuidado_intensivo = cid.name " +
13    "INNER JOIN died_dimension as dd ON covid.fallecido = dd.name " +
14    "LEFT JOIN date_dimension AS ddd ON covid.fecha_fallecimiento = ddd.date " +
15    "INNER JOIN respiratory_assistance_dimension AS rad ON covid.asistencia_respiratoria_mecani
16    "INNER JOIN financing_dimension as fd ON covid.origen_financiamiento = fd.name " +
17    "INNER JOIN classification_dimension as cd ON covid.clasificacion_resumen = cd.name " +
18    "LEFT JOIN date_dimension AS ddiag ON covid.fecha_diagnostico = ddiag.date " +
19    "ORDER BY covid.id_evento_caso ASC "
20 )

```

Fig. 12.2: Código Python en el que se observa la transformación a una tabla de hechos

²⁷ <https://drive.google.com/drive/my-drive>

- En la 1er y 2da línea de código se puede ver cómo registrar los data frames para poder usarlos con pyspark SQL, estas se estarían registrando como tablas temporales.
- Luego en las siguientes líneas se puede ver la query SQL.

Carga (del proceso ETL):

Para la carga de la tabla a Postgres se hizo mediante el método write de pyspark, como se puede ver en la siguiente porción de código (Fig. 13):

```

1  clasificacion.select("classificationID", "name")\
2  .write.format("jdbc")\
3  .option("truncate", "true") \
4  .option("url", "jdbc:postgresql://localhost:5432/covid") \
5  .option("dbtable", "classification_dimension") \
6  .option("user", "postgres") \
7  .option("password", "212121") \
8  .option("driver", "org.postgresql.Driver") \
9  .mode("append")\
10 .save()

```

Fig. 13: Código Python en el que se observa la carga de los datos a Postgres.

- En la 1er línea de código se puede ver cómo se seleccionan las columnas de la tabla a escribir en la base de datos
- En la 2da se ve el método write especificando que se utilizara JDBC (que es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java)
- en la 4ta se puede ver como se especifica el puerto y la base de datos
- en la 5 se especifica la tabla
- en la 6ta y 7ma se especifican usuario y contraseña respectivamente
- en la 8va el driver que se utiliza para la conexión a la base de datos (que es de JDBC)
- en la 9na el modo de escritura que es append, la cual especifica que los valores deben agregarse a la tabla
- y en la 10ma se especifica que se guarden los cambios hechos

Aplicación del proceso ETL para NoSQL

En el caso del proceso ETL para NoSQL se trata de un pequeño script que transforma el .csv en un archivo en formato JSON.

Extracción y Transformación del proceso ETL

El proceso de extracción no difiere al proceso realizado en SQL, como se puede observar en la primera línea de código de la Fig. 14. Mientras que el proceso de transformación, cómo se puede apreciar en la segunda y tercera línea de código de la misma Fig. Es más simple que en SQL. No se necesitó construir una tabla de dimensión ni

dividir la información en distintas tablas. En su lugar, al tratar los datos de forma menos rígida, para NoSQL el proceso consistió en cambiar el formato en que se presentan los datos y no su estructura en sí. Ya que la mayoría de los motores de bases de datos NoSQL manejan la información en formato JSON, se definió un código que reciba los datos de un .CSV y los devuelva en un TXT con formato JSON, completando así la etapa de Transformación del proceso ETL.

```

1 covid = sqlContext.read.format("csv")
2   .options(header='true', inferSchema='true')
3   .option("encoding", "UTF-8").load("/content/drive/My Drive/Covid/Covid19Casos.csv")
4 covid = covid.withColumnRenamed('edad años meses', 'edad años meses')
5 covid.coalesce(1).write.format("json").save("/content/drive/My Drive/Covid/resultado")_

```

Fig. 14 : Código del proceso de extracción y transformación de un CSV a TXT en formato JSON.

Carga del proceso ETL

Carga en la base de datos ElasticSearch

Una vez obtenido el archivo en formato JSON, se utilizó la herramienta Logstash para la carga de los datos al motor de base de datos elegido, en este caso, Elastic. Para el pasaje de datos de un archivo JSON al motor elegido se necesita un archivo .conf con tres configuraciones, una entrada (input), un filtro(filter) y una salida(output), que Logstash lo utilizará para realizar la carga de información en Elastic. Como se puede observar en la Fig. 15, en la sentencia de entrada de la líneas 1-7 se detallan los datos del archivo origen (cómo la posición de entrada en la línea 3 y la ruta en la 4). Por otra parte, en las líneas 9 a 16, se describe el filtro en donde se trabaja y selecciona los datos que consideramos que deben guardarse, por ejemplo en la línea 14 se eliminan los datos “message”, “host”, entre otros. Finalmente, en la sentencia de salida de las líneas 18-25, se configura dónde y cómo queremos enviar los datos, es decir, el destino de la información. Se puede observar en la línea 22 el host (en nuestro caso el puerto 9200 de nuestro servidor local) y el nombre de nuestra base de datos en la línea 23 (bbdd2_datajson).

```

1 input {
2   file {
3     start_position => "beginning"
4     path => "/home/franco/Escritorio/Jsontdata/dataJSON.log"
5     sinedb_path => "/dev/null"
6   }
7 }
8
9 filter {
10  json {
11    source => "message"
12  }
13  mutate {
14    remove_field => ["message", "@timestamp", "path", "host", "@version"]
15  }
16 }
17
18 output {
19  stdout {
20  }
21  elasticsearch {
22    hosts => "http://localhost:9200"
23    index => "bbdd2_datajson"
24  }
25 }

```

Fig 15: Archivo en el que se define qué hará el logstash

Una vez efectuado el comando que ejecuta Logstash con el archivo .conf indicado, si se realiza satisfactoriamente, se tendrán los datos cargados en la base de datos indicada. El comando consta del path donde se encuentra el ejecutable, el flag -f indica que la configuración será cargada de un archivo o directorio específico que se indica a continuación en la línea de comando. En la Fig. 16 se muestra el comando utilizado para cargar según lo especificado en el archivo “bbdd2_carga.conf”.

```
franco@franco-MS-7A15:~/Escritorio$ /home/franco/Escritorio/Elastic-Kibana/logstash-7.12.1-linux-x86_64/logstash -f bbdd2_carga.conf
```

Fig. 16: Ejecución del Logstash

Finalmente los datos se encontrarán disponibles en Elastic y, por su integración con Logstash y Kibana, podremos visualizarlos a través del último.

Carga en MongoDB

El proceso en esta base de datos fue más sencillo. MongoDB ofrece un potente GUI²⁸ llamado MongoDB Compass, que permite la carga y manipulación de datos de forma rápida y totalmente intuitiva. Como puede verse en la Fig. 17, una vez seleccionada la base de datos donde se guardará la información, se accede a la opción “ADD DATA” donde aparecerá un menú para importar archivos. En dicho menú se permite elegir el tipo de archivo que se desea cargar (JSON o CSV). Como el archivo resultante del proceso de transformación fue un JSON, fácilmente se puede buscarlo e importarlo a nuestra base de datos. Luego de este proceso, los datos ya están disponibles para su futura visualización.

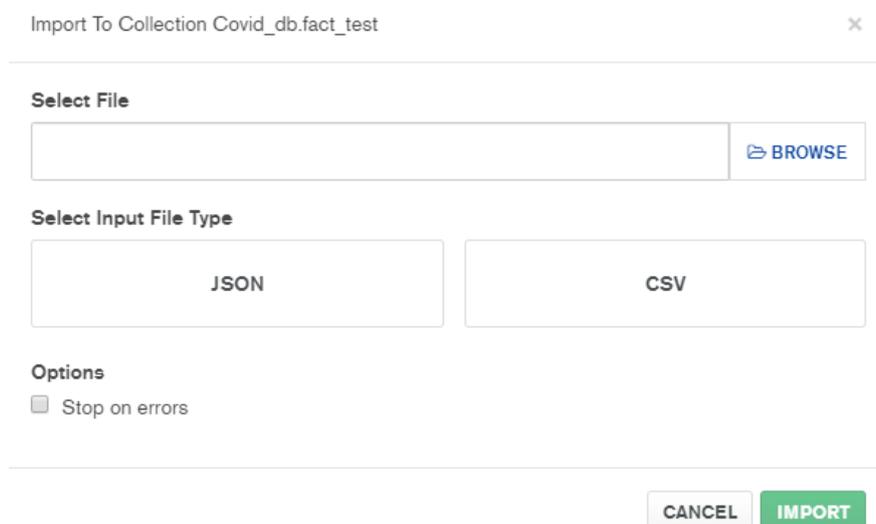


Fig. 17: Menú de importación de archivos en MongoDB Compass.

Comparación de las Herramientas de visualización

Razón de la elección de las herramientas de visualización

En la siguiente sección se explicará brevemente qué herramientas de visualización se utilizaron en el estudio realizado y la razón de la elección de cada una de ellas.

²⁸ Interfaz de usuario gráfica

Tableau

La elección de la herramienta Tableau es debido a que es una de las herramientas más potentes del mercado, aunque frente a las demás herramientas mencionadas en este capítulo tiene una clara desventaja que es que para su utilización es necesario pagar una suscripción. A pesar de esto, se incluyó esta herramienta debido a que se consiguió una licencia de estudiante que permite su utilización para uso estudiantil.

Saiku Analytics (Pentaho)

Se consideró la herramienta Saiku Analytics ya que Pentaho es una plataforma de código abierto que cuenta con herramientas para la integración y análisis de los datos. Pentaho posee una "Community edition" la cual es de uso gratuita y está incluye la utilización de Saiku Analytics.

Bokeh

La elección de la herramienta fue debido a que para el proceso ETL sobre Postgres se usó PySpark que es una librería de Python, se eligió esta herramienta ya que mediante Pyspark y Python se puede convertir la información a como Bokeh la necesita y generar visualizaciones fácilmente. Por otro lado esta librería es de uso gratuito así como también se puede exportar la visualización a un HTML que pesa muy poco y debido a esto se puede compartir muy fácilmente.

Kibana

Se tomó la herramienta Kibana para ser analizada ya que está ya posee una integración con el motor de bases de datos Elasticsearch. Ambos forman parte del stack ELK formado por tres herramientas independientes: Elasticsearch, Logstash y Kibana. Además, una característica de gran importancia para la elección de esta herramienta fue debido a que es gratuita siempre que se use de forma local.

Charts

La herramienta Charts fue seleccionada debido a que es el visualizador por defecto que posee MongoDB ya que pertenecen al mismo desarrollador y forman parte de una potente integración de herramientas. Es fácilmente configurable y su conexión con la base de datos de MongoDB es sencilla. Además, se trata de una herramienta gratuita a nivel local. Debido a todas estas razones se consideró a Charts como uno de los visualizadores de datos a utilizar.

CODAP y DataWrapper

Tanto la elección de la herramienta CODAP como la elección de la herramienta DataWrapper, fue debido a que presentaban similitudes con el objetivo final de la tesina. Ambas son herramientas que poseen una versión gratuita (CODAP es únicamente gratuita). CODAP es de código abierto y Datawrapper provee una API para trabajar desde otros sitios. Además, ambas funcionan como una aplicación web.

Comparación de conexiones

En esta sección se comparan las conexiones de las herramientas de visualización seleccionadas previamente. Con conexiones se hace referencia a las formas de comunicación con otras tecnologías, ya sea base de datos o una aplicación en sí. Éste es un aspecto a destacar tanto en Tableau como en Bokeh y Saiku Analytics, los cuales pueden leer datos desde distintas bases de datos, a diferencia de Kibana que su conexión más potente es con Elasticsearch y Charts que solo puede leer datos de MongoDB.

Saiku Analytics (Pentaho)

Saiku Analytics puede tomar datos desde las siguientes bases de datos: Spark SQL, Cloudera Impala, Impala, Hadoop Hive 2, Generic Database, H2, MonetDB, Hypersonic, MySQL, PostgreSQL, Pentaho Data Service y Snowflake.

Tableau

Tableau, al día del presente (27/05/2021) documento, posee 87 conectores a distintas fuentes de datos entre los que se destacan gran cantidad de bases de SQL como PostgreSQL, MySQL, Oracle o MariaDB y también distintas bases de datos NoSQL como lo son Google BigQuery, MongoDB, Elasticsearch. Aparte de estos, en Tableau se puede utilizar la API de conectores Java Database Connectivity (JDBC) para conectarse mediante java y también la API de “Open DataBase Connectivity” para conectarse a las bases de datos. Por otro lado Tableau también puede tomar datos desde archivos de texto en formato JSON o CSV como también hasta archivos EXCEL, PDF o de texto plano.

Bokeh

Bokeh gracias a la utilización de PySpark puede tomar datos desde una gran cantidad de bases de datos ya que este utiliza la API de conectores Java Database Connectivity (JDBC) para conectarse mediante java a las bases de datos. Como así también tomar datos mediante Pyspark desde un archivo que se encuentre en archivos de texto en formato JSON o CSV. Como también obtener respuestas de una API Rest siempre que la respuesta sea traducible a una estructura de datos (como podrían serlo si devuelven un texto en formato JSON o CSV).

CODAP

En CODAP se puede exportar todo lo que se haya generado en el documento a un archivo de texto (que se exporta como un archivo texto en formato JSON) y luego se puede importar estos archivos a la aplicación y continuar utilizándolos. También se pueden cargar archivos CSV que generan una nueva tabla en el documento.

DataWrapper

En DataWrapper se puede importar archivos CSV (Comma Separated Values) y hojas de cálculo, como también ingresar manualmente un texto con un formato CSV. Por otro lado, se puede conectar a un conjunto de datos externo, así ese gráfico se actualizará automáticamente cuando los datos cambian. Los datos deben ser accesibles públicamente en la web en un formato CSV apto para la lectura en Datawrapper.

Comparación de costos

En la siguiente tabla (Tabla 2) se pueden ver las distintas versiones que ofrece cada Visualizador como también los costos y/o cuanto tiempo se puede utilizar esa versión (ya sea gratuita o a partir del pago).

Herramienta/ Versión	Versión gratuita	Versión de prueba	Version para Estudiantes	Versión paga de menor costo
Saiku Analytics	Si	No	No	No
Kibana	Si	Si (30 días)	No	\$16 USD /mes Ofrece todo el sistema de ElasticSearch (Incluye base de datos) para utilizarlo de forma remota (Cloud) (*)
Bokeh	Si	No	No	No
Tableau	No	Si (15 días)	Si (1 año)	\$70 USD/mes Para la utilización de la herramienta (*)
Charts	Si	Si (Hasta 512MB de almacenamiento)	No	\$57 USD/mes Ofrece todo el sistema de MongoDB (Incluye la base de datos)para utilizarlo de forma remota (Cloud) (*)
CODAP	Si	No	No	No
DataWrapper	Si	No	No	\$599 USD/mes Para la utilización de la herramienta. Esta versión posee 10 licencias (*)

Tabla 2: Relación entre versiones y las herramientas de visualización (*) al 2021-06-17

Conclusiones

Existe una gran cantidad de herramientas de visualización de datos actualmente en el mercado. Para tomar la decisión de cuál de ellas utilizar se deben tomar en cuenta distintos puntos. En primer lugar, y como el más importante, es necesario basar la decisión

en el tipo de base de datos en la que se encuentran los datos, ya que a partir del tipo elegido se ofrecen herramientas para cada uno de ellos. Además del tipo, es necesario saber qué base de datos es utilizada ya que, por ejemplo cómo vimos con kibana, existen potentes integraciones de herramientas que facilitan el proceso.

Para comparar las herramientas se buscó generar gráficos que tengan como objetivo responder las siguientes preguntas que consiguen resolverse a partir de la información del dataset trabajado:

1. **¿Cuál es la cantidad de casos confirmados por fecha?**
Para responder a esta pregunta se buscó generar un gráfico de barras en donde cada barra representa la cantidad de casos en un día.
2. **¿Cuál es la cantidad de muertes totales por fecha?**
Para responder a esta pregunta se decidió generar un gráfico de línea en donde la cantidad de muertes se acumule a medida que pasa el tiempo que permita visualizar los cambios a lo largo de un rango continuo, como lo es la fecha.
3. **¿Cuál es la cantidad de muertes por fecha?**
Para responder a esta pregunta se eligió generar un gráfico de barras en donde cada barra representa la cantidad de casos en un día.
4. **¿Cuál es el porcentaje de casos confirmados por financiamiento?**
Para responder a esta pregunta se decidió generar un gráfico de torta o gráfico circular en el que cada porción representa cada tipo de financiación.

A continuación se detallan las conclusiones obtenidas de la utilización de cada una de estas herramientas al generar las visualizaciones que responden a estas preguntas:

Saiku Analytics (Pentaho)

Luego de la utilización de Saiku Analytics, la conclusión que se obtuvo fue que si bien la herramienta logró cumplir con la generación de visualizaciones, presenta un claro problema. Para llegar al punto de generar las visualizaciones, se debe cargar un archivo Mondrian (formato XML), el cual resultó muy complejo lograr entender cómo escribirlo. Aunque podían generarse las visualizaciones a través de consultas MDX, también resultaron difíciles de comprender debido a que se encontró muy poca información y documentación sobre estas. Con diferencia, entre las herramientas utilizadas, Saiku Analytics pareció ser la más engorrosa de utilizar

Tableau

Finalizada la utilización de esta herramienta, se concluyó que fue la más simple de utilizar. Además, presenta un gran abanico de visualizaciones disponibles. Una clara ventaja sobre las demás, es que no se necesita un usuario con gran conocimiento en programación para lograr generar las visualizaciones. También presenta una gran variedad de opciones para cargar los datos, ya sea desde una base de datos hasta una planilla de Excel, un archivo JSON o un archivo CSV. Debido a todo lo anterior se tomó en cuenta esta opción como la mejor opción entre las estudiadas. De todas formas, al ser una aplicación paga con licencia gratuita temporal únicamente de prueba o para estudiantes, no se podrá considerar para el desarrollo de la plataforma.

Bokeh

Esta herramienta parte con la desventaja de que hay que integrarlo con otras librerías de Python para lograr generar las visualizaciones a partir de la base de datos, lo que presenta cierto grado de dificultad. Sin embargo, pese a esto, presenta un gran potencial. Bokeh presenta una gran variedad de visualizaciones con una gran cantidad de variaciones en cada una de ellas, dependiendo de cómo se quiera mostrar la información. Un punto fuerte es la posibilidad que brinda de integrar fácilmente los elementos generados a páginas web, como así también la posibilidad de exportar la visualización generada mediante un archivo HTML. Este archivo pesa muy poco y se puede ver desde cualquier navegador.

Se considera que si bien esta herramienta presenta un fuerte potencial, no logra alcanzar a su contraparte previamente mencionada, Tableau. Esto debido principalmente a que no es tan fácil de utilizar por un usuario no experimentado en programación.

Kibana

Kibana presenta una fuerte herramienta para la visualización de datos por su integración con Elasticsearch y Logstash. El proceso ETL del dataset estudiado para la utilización de esta herramienta resultó sencillo ya que la documentación disponible de ELK es clara y detallada. Es por esto que la conversión del dataset y su carga a la base de datos no resulta dificultosa.

Por otra parte y a diferencia de Tableau, por ejemplo, la creación de los gráficos no resultó tan sencilla e intuitiva. Para hacerlo, se debe arrastrar los campos que se desean visualizar (tanto horizontal como vertical), elegir el tipo de gráfico y filtrar/seleccionar la información (rangos de edad, campos con datos específicos, etc.). En este sentido, la información disponible no es extensa ya que se trata de una tarea particular y depende de las necesidades del usuario. Finalmente, y como último comentario, para comenzar a utilizar las herramientas de visualización se debe crear un índice sobre los datos, agregándole un paso más a todo el proceso.

Charts

Charts es la mejor opción para graficar datos guardados en una base de datos MongoDB ya que, como mencionamos anteriormente, pertenecen al mismo desarrollador. El proceso ETL, al igual que en Elastic, no resultó dificultoso ya que se trata de una simple conversión de datos a formato JSON y su posterior carga a MongoDB. La carga de datos resultó ser la más sencilla de realizar, ya que su GUI de Windows ofrece múltiples opciones para hacerlo.

En cuanto al visualizador en sí, si bien la integración de las herramientas logró que no sea dificultosa su configuración, se consideró que tiene ciertos puntos dónde queda en desventaja comparado con sus contrapartes. En primer lugar, los datos de tipo fecha no son reconocidos como tal y su formato no se muestra de una forma simple. Por otra parte, al no utilizar índices, el tiempo que tarda en realizar los gráficos cuándo se selecciona la información que se desea, es alto comparado con el resto de los visualizadores utilizados en el transcurso del informe. El último punto a mencionar está en el hecho de que no posee

variedad de funciones de agregación y, por ejemplo, al no presentar la función suma imposibilita la realización del gráfico de la totalidad de fallecidos.

CODAP

Al utilizar CODAP se presentaron grandes limitaciones. El dataset que utilizamos para probar los visualizadores tenía aproximadamente 12 millones de filas y CODAP recomienda trabajar con no más de 5000 registros. Aunque la herramienta reconoció el archivo al cargarlo y presentaba una función para tomar 5000 registros al azar para generar las visualizaciones, no se logró realizar visualizaciones en las que los resultados conseguidos sean precisos. Por otro lado CODAP no posee una amplia carta de gráficos disponibles, lo que se considera una enorme falencia en comparación con sus competidores.

DataWrapper

Luego de la utilización de DataWrapper, se concluye que si bien la herramienta logra cumplir con la generación de visualizaciones, presenta grandes limitaciones. En primer lugar no se pueden cargar archivos de más de 2 MB de tamaño. Por lo tanto, deben ser a partir de archivos que no sean de mayor tamaño. Por otro lado, esta herramienta no se puede agrupar a los registros, lo que presenta una gran desventaja contra sus competidores. Por último, no posee una versión en castellano, lo que sería una dificultad a la hora de ser presentado en contextos escolares. Cabe destacar que la mayor ventaja que posee DataWrapper es que está destinada para usuarios finales, quitando la necesidad de conocimiento en programación para utilizar la aplicación.

APIs analizadas para construir grafos sociales

En el siguiente apartado se detallarán las Apis analizadas para la construcción de grafos sociales como así también las experiencias de uso de los mismos.

API de Facebook - Graph API

La *API Graph*²⁹ de Facebook es la principal herramienta para ingresar datos en la plataforma de Facebook, como así también es la mejor para extraer datos de esta plataforma.

Esta API se trata de una API basada en HTTP (Protocolo de Transferencia de Hipertexto) que puede ser usado por las aplicaciones de una manera programática para consultar datos de la plataforma, publicar nuevas historias, administrar anuncios, subir fotos, entre otros.

El nombre de la Graph API proviene del concepto de “grafo social”, ya que así se puede representar la información en la plataforma de Facebook. Este gráfico se compone de nodos, aristas y campos. Los nodos se utilizan para representar un objeto en específico; las aristas se usan para obtener colecciones de objetos (nodos) a partir de un único objeto

²⁹ <https://developers.facebook.com/docs>

(nodo); y los campos que contienen los datos sobre un objeto único o algún dato sobre la colección de objetos.

En esta API se necesita generar un token el cual permite a la aplicación acceder a la API Graph. Todos los puntos de conexión de la API Graph requieren algún tipo de token de acceso, por lo que todas las solicitudes de acceso pueden requerir uno. Normalmente, realizan dos funciones:

- Permite que la app acceda a la información de un usuario sin necesidad de proporcionar su contraseña.
- También permite identificar la app, el usuario que la utiliza y los tipos de datos a los que este le permitió acceder.

Experiencia de uso

Para acceder a esta API se debió crear una cuenta de desarrollador en la plataforma de Facebook para así poder registrar a la aplicación en la API de Facebook. Mediante esta API se pueden acceder a los datos propios de los usuarios que hayan permitido a la aplicación creada la utilización de información. Debido a que Facebook no brinda de forma pública ningún dato a través de su API se dificulta mucho recolectar información desde esta que sea interesante de analizar. En esta API hay muchas restricciones de privacidad por ejemplo, si se quiere acceder a las lista de amigos de un usuario que permitió esto a la aplicación solo devuelve la cantidad de amigos que dicho usuario posee, si se quiere acceder a los datos de estos amigos (como el nombre) cada amigo debe permitirle acceder a esa información a la aplicación que se esté desarrollando.

API de Twitter V2

La *API de Twitter*³⁰ es una API Rest la cual permite a los programadores el acceso a los datos de Twitter como son los tweets y los usuarios de la misma. Y todo lo que conlleva con ellos (en el caso de los usuarios, por ejemplo, obtener las personas que los siguen).

Twitter comparte su información de la forma más amplia posible, Proporciona a las empresas, los desarrolladores y los usuarios un acceso programático a los datos de Twitter mediante el uso de su API.

Twitter permite acceder a los datos de su servicio mediante su API lo que permite a las personas crear software que se integren con Twitter y así brindar una solución que ayuda a las empresas a responder a la opinión del cliente en Twitter.

La API de Twitter ofrece todos los datos que se encuentren de forma pública en su aplicación. Por otro lado, la API de Twitter también permite a los usuarios administrar su propia información que no es pública (como lo son los mensajes directos) y brindar esta información a los desarrolladores autorizados para administrarla.

Las API de Twitter incluye una gran variedad de Objetos que pueden ser consultados, los cuales se dividen en cinco grupos principales:

- Cuentas y usuarios

³⁰ <https://developer.twitter.com/en/docs/twitter-api>

- Tweets y respuestas
- Mensajes Directos
- Anuncios
- Herramientas y SDK de editor

Experiencia de uso

Para acceder a la API se debió hacer una cuenta de desarrollador en Twitter para luego registrar a la aplicación en la API de Twitter. Esta API de forma predeterminada permite a la aplicación que solo pueda acceder a la información pública de Twitter. Por defecto la API de Twitter da acceso a su versión Essential (que es gratuita) la cual brinda la posibilidad de consultar hasta quinientos mil Tweets por mes. Debido a que esta cantidad resultó baja se solicitó la mejora a la versión Elevated la cual es gratuita y permite consultar hasta 2 millones de tweets.

Para su utilización solo se necesita autenticarse en la API de Twitter desde el código de la plataforma prototípica antes de comenzar a hacer los requerimientos.

Capítulo 5: Prototipo del Visualizador Web

En este capítulo se presenta el modelo de arquitectura elegido para el visualizador propuesto y una posible implementación de la misma. Se definen las tecnologías utilizadas para construir el visualizador, así como los motivos por los cuáles se llegaron a elegir algunas de ellas. Se consideró como principales elecciones la tecnología para generar los gráficos y el motor de base de datos para almacenamiento.

Arquitectura del visualizador prototípico

La arquitectura seleccionada para construir el visualizador será de tipo Cliente-Servidor [Oluwatosin, 2014]. Es por esto que se diferenciará qué recursos y actores participarán como clientes y que herramientas harán la tarea de servidor.

Una posible implementación de la arquitectura propuesta

A continuación se presentará una posible implementación para los clientes y el servidor del Visualizador Web al cual denominamos “CeoDatum”. Este nombre consiste en una palabra conformada por otras dos. Por una lado “Ceo” es el titán de la inteligencia en la mitología griega y por otro “Datum” es dato en latín. Debido a esto se considera que el nombre hace referencia a la inteligencia de datos. La Fig. 18 representa gráficamente la posible implementación del modelo elegido. En esta el cliente hace peticiones HTTP al servidor que contiene la aplicación y la BBDD. La aplicación recibe dicha petición, la procesa, consulta la BBDD si es necesario y envía una respuesta al cliente.

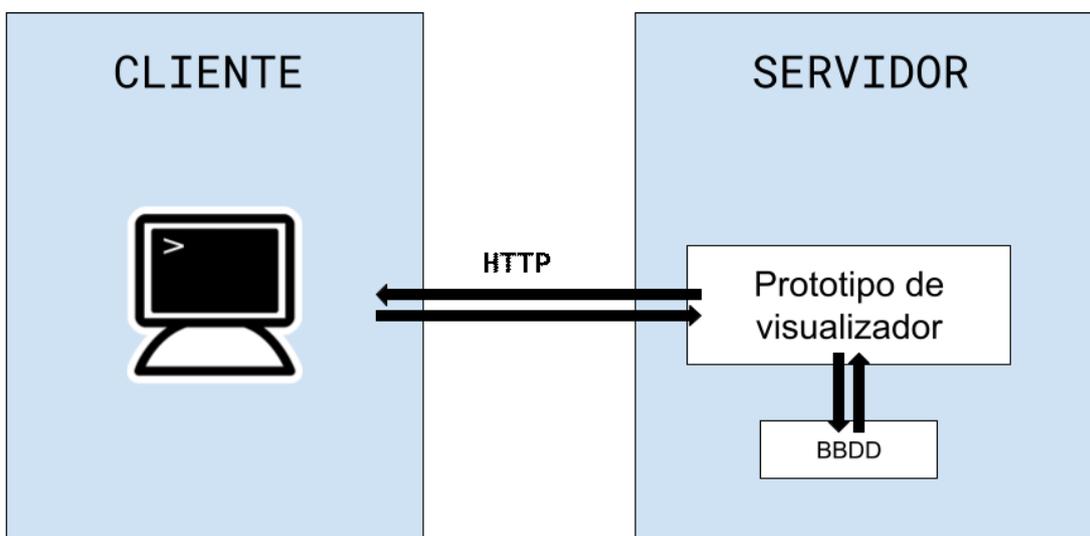


Fig. 18: Representación del modelo de arquitectura de la plataforma CeoDatum

Clientes en la Plataforma

Los clientes en el visualizador son los usuarios finales a través de un navegador web. Como se menciona en la definición de la arquitectura, las tareas se encuentran distribuidas entre el cliente y el servidor. En este caso, algunas tareas sencillas como comprobaciones de campos, creación de interfaces interactivas u conexiones asincrónicas a

la base de datos se realizan del lado cliente a través de JavaScript, su librería JQuery³¹ y la técnica AJAX³². Estas tecnologías se detallarán más adelante.

Servidor de la Plataforma

El servidor aloja la lógica de la plataforma y el motor de base de datos. Recibe peticiones de uno o más clientes, las gestiona y responde a través de actualizaciones de la página HTML. El servidor WEB para alojar la visualizador prototípico será brindado por el laboratorio LIFIA de la Facultad de Informática de la Universidad Nacional de La Plata (lugar de trabajo de los directores del presente informe).

Patrón Model View Controller

La lógica e interfaz de la plataforma sigue una implementación del patrón MVC, se encuentra dividida en tres áreas: el modelo, la vista y el controlador. En primer lugar, en el modelo se encuentran las clases que manejan los datos y la lógica de negocios. En él se describen y llevan a cabo las consultas a la base de datos, por lo que el modelo define los datos que mostrará el visualizador prototípico. Por otra parte, la vista define cómo van a mostrarse esos datos al usuario. Su principal función es el diseño y la presentación. Por último el controlador actúa como puente entre el modelo, la vista y el cliente. Define las rutas por las que el cliente podrá conectarse a la plataforma y las funciones que se realizarán al hacerlo. El controlador gestiona peticiones (a través del protocolo HTTP) del usuario, consulta y recibe los datos del modelo, arma una vista a partir de ellos y la devuelve al cliente para ser mostrada. En la Fig. 19 se puede observar un esquema que refleja el funcionamiento de la implementación de la arquitectura del prototipo de visualizador, siguiendo el patrón MVC.

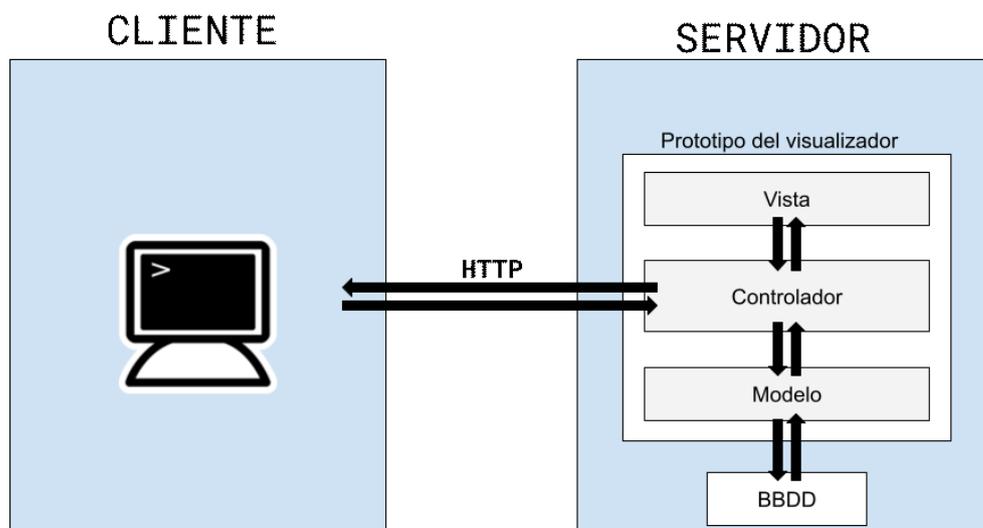


Fig. 19: Representación del funcionamiento de la plataforma CeoDatum siguiendo una implementación del patrón MVC

³¹ <https://jquery.com/>

³² Técnica de desarrollo web para aplicaciones asincrónicas.

Elección de los motores de bases de datos

Para la elección del motor de base de datos, se realizó una comparación entre SQL y NOSQL con el fin de definir cuál es el que usará en la visualizador prototípico. El resultado permitió escoger la mejor opción para almacenar los datos cargados a partir de los archivos subidos por los usuarios, como así también los datos propios de CeoDatum. En el Anexo 1 se detalla el análisis realizado para escoger el motor de base de datos SQL.

Elección del graficador a utilizar

Se decidió utilizar el motor de gráficos Bokeh (Librería de Python) para realizar las visualizaciones de datos. Las razones de la elección se listan a continuación:

- Tiene un gran potencial. Sin embargo, para lograr un mayor potencial hay que integrarlo con otras librerías de Python. Su utilización es mediante código de programación, por lo que no es sencillo para un usuario no experimentado.
- Presenta una gran variedad de visualizaciones a generar, con cantidad de variaciones en cada una de ellas. El programador así, consigue un amplio abanico de opciones y personalizaciones para la presentación de los gráficos.
- Un punto muy fuerte es la posibilidad de integrarlo fácilmente a páginas web, como así también la posibilidad de exportar la visualización generada mediante un archivo HTML. Este archivo es considerablemente pequeño y se puede ver desde cualquier navegador.
- Posee una licencia gratuita y de código abierto.

Bokeh posee una dócil integración con la construcción de páginas web, lo que coopera con la construcción de la plataforma. Debido a que la utilización del graficador es a través de código de programación, la principal tarea fue conseguir a través de una interfaz sencilla para el usuario, una conexión entre lo que se demanda y lo que se solicita a la herramienta.

Elección del framework

Debido a que se trata de un prototipo de visualizador web y que Bokeh es una librería del lenguaje Python, se decidió utilizar éste como principal lenguaje de programación para la realización del proyecto. Entre los frameworks más comunes de Python se encuentran Flask y Django [Devndra, 2020] [Idris et al., 2020]. En el Anexo 2 se detalla la comparación entre ellos para escoger el más adecuado para el desarrollo del visualizador prototípico en cuestión, la cual resultó en la elección de Flask como el Framework a utilizar en la construcción del mismo.

Una implementación del prototipo de Visualizador

En esta sección se realizará una explicación progresiva de la implementación de la arquitectura del aplicación. Se llevará a cabo a través de gráficos acumulativos que irán sumando información a medida que la explicación avance. Se dividirá en tres figuras que aumentan el nivel de detalle en cada una de ellas.

Se comenzará con el gráfico de la Fig. 20 como puntapié inicial de la explicación. En ella se muestran diversas herramientas que fueron utilizadas durante la construcción de la plataforma.

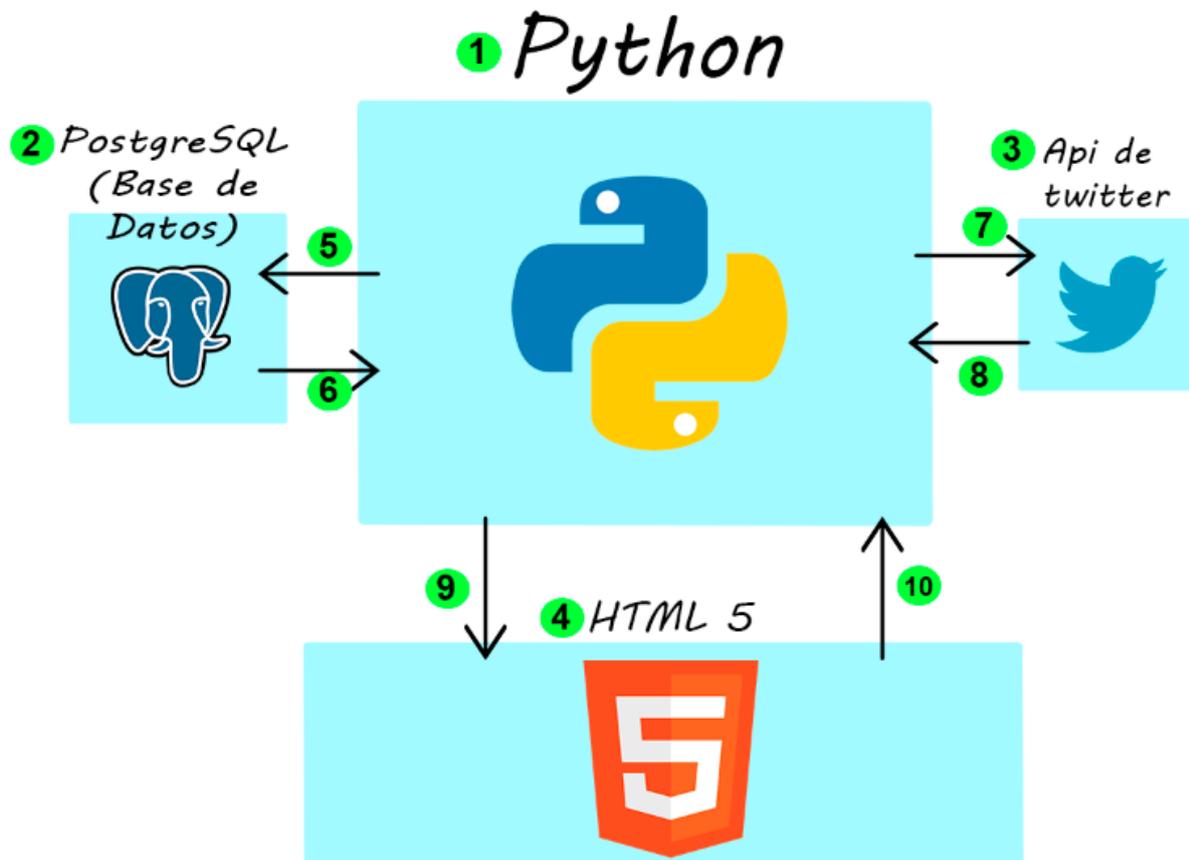


Fig. 20: Representación general de la implementación de la arquitectura de CeoDatum

A continuación se realizará una explicación de las referencias visualizadas en la Fig. 20.

1. **Python:** Es un lenguaje de programación de alto nivel utilizado para el desarrollo de aplicaciones de todo tipo. Este lenguaje de programación se ha utilizado para desarrollar la mayor parte del visualizador prototípico, específicamente la lógica de la misma.
2. **PostgreSQL:** Es una sistema de bases de datos relacional de código abierto que lleva más de 30 años activo. Ha generado una sólida reputación por su confiabilidad, solidez de funciones y rendimiento. En la plataforma se ha utilizado PostgreSQL como base de datos de la misma, tanto para guardar los datos de la aplicación como también los dataset que se carguen de la misma.
3. **Api de Twitter:** Es una API REST³³ que permite a los programadores el acceso a los datos de Twitter, como son los tweets y los usuarios de la misma y todo lo que conlleva con ellos (en el caso de los usuarios, por ejemplo, obtener las personas que los siguen).
4. **HTML5:** HTML5 es la última versión del estándar HTML que se utiliza para crear páginas web.

³³ Interfaz para conectar varios sistemas basados en el protocolo HTTP.

5. **Flecha desde Python hacia PostgreSQL:** Python hace requerimientos a la base de datos PostgreSQL a través de consultas que envía posteriormente a haber generado un conexión con ésta.
6. **Flecha desde PostgreSQL hacia Python:** PostgreSQL devuelve los datos requeridos como respuestas a la consulta recibida.
7. **Flecha desde Python al api de Twitter:** Python realiza requerimientos en los que envía distintos parámetros al api de Twitter. Debe estar autenticado para poder realizarlos.
8. **Flecha desde api de Twitter a Python:** La api de Twitter devuelve una lista de los objetos que se haya solicitado (usuarios, tweets, etc.).
9. **Flecha desde Python a HTML5:** Python renderiza archivos HTML. Para esto utiliza los estándares de lenguaje HTML para hacer una correcta interpretación de la información a mostrar.
10. **Flecha desde HTML5 a Python:** HTML5 devuelve cómo renderizar el archivo HTML.

En la Fig. 20 se presentó la vista general de los componentes y herramientas del prototipo de Visualizador. A continuación, en la Fig. 21 se procederá a explicar en mayor detalle el módulo de Python, especificando frameworks y librerías utilizadas.

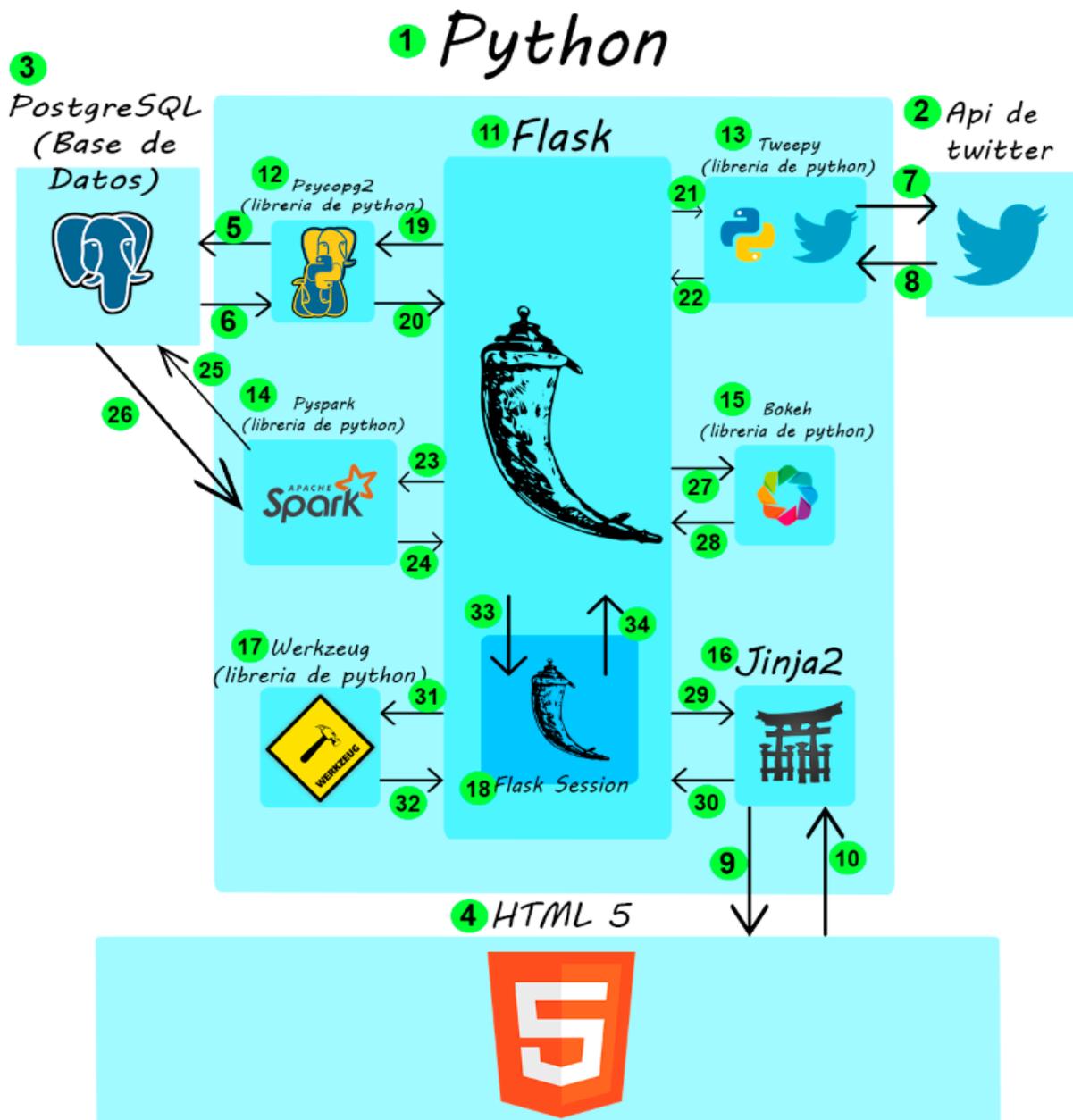


Fig. 21: Representación de la implementación de la arquitectura con mayor detalle en el módulo Python

A continuación se realizará una explicación de las referencias añadidas en la Fig. 21 (en relación a la Fig. 20), incluyendo una descripción de los cambios en las conexiones ya existentes.

- 11. Flask**: Flask es un micro-framework de python que permite el desarrollo de aplicaciones web. Y es el que se ha utilizado para el desarrollo del prototipo de visualizador construido.
- 12. Psycopg2**: Psycopg es el adaptador de base de datos PostgreSQL más popular para el lenguaje de programación Python. Una de sus principales características es la seguridad de subprocessos (varios subprocessos pueden compartir la misma conexión). Fue diseñado para aplicaciones con muchos subprocessos múltiples que crean y destruyen muchos cursores y hacen una gran cantidad de "INSERT" o "UPDATE" simultáneos.

13. **Tweepy**: Tweepy es una librería de Python que permite realizar las peticiones a la api de Twitter de una forma mucho más rápida y sencilla. Es de código abierto y se encuentra mantenido por su comunidad.
14. **PySpark**: Pyspark es una librería de Python la cual funciona como una interfaz de Apache Spark en Python. PySpark es compatible con la mayoría de las funciones de Spark, como Spark SQL, DataFrame, Streaming, MLlib (aprendizaje automático) y Spark Core. Esta librería contiene también a la librería de Pandas la cual es muy potente para transformar los datos.
15. **Bokeh**: Bokeh es una librería de Python para crear visualizaciones interactivas para navegadores web modernos. Con esta librería se puede crear desde los gráficos más simples hasta otros con un grado de complejidad considerable. Con Bokeh se pueden crear visualizaciones impulsadas por JavaScript sin escribir ninguna línea de JavaScript.
16. **Jinja2**: Jinja2 es un lenguaje de plantillas moderno y fácil de diseñar para Python, modelado a partir de las plantillas de Django. Es rápido, ampliamente utilizado y seguro. Los marcadores de posición especiales en la plantilla permiten escribir código similar a la sintaxis de Python. Luego, a la plantilla se le pasan datos para representar el documento final.
17. **Werkzeug**: Werkzeug es una biblioteca completa de aplicaciones web WSGI (Web Server Gateway Interface). Comenzó como una colección simple de varias utilidades para aplicaciones WSGI y se ha convertido en una de las bibliotecas de utilidades WSGI más avanzadas. Se ha utilizado para comprobar si los archivos son seguros y para encriptar las contraseñas.
18. **Flask Session**: Flask-Session es una extensión para Flask que agrega soporte para las sesiones del lado del servidor para poder usarla dentro del prototipo de visualizador.
19. **Flecha desde Flask hacia Psycopg2**: Flask solicita la conexión con la base de datos PostgreSQL a través de Psycopg2, para esto le envía los datos requeridos para realizar la conexión.
20. **Flecha desde Psycopg2 hacia Flask**: Psycopg envía a Flask la conexión a la base de datos solicitada.
21. **Flecha desde Flask hacia Tweepy**: Flask utiliza los métodos de tweepy para realizar los requerimientos al api de Twitter y así utilizar esta de una forma más sencilla.
22. **Flecha desde Tweepy hacia Flask**: Tweepy devuelve hacia Flask los objetos obtenidos desde la api de Twitter solicitados por el método que ha especificado Flask.
23. **Flecha desde Flask hacia PySpark**: Flask utiliza los métodos de PySpark para tratar los datos obtenidos desde los archivos como así también para cargar estos datos en la base de datos de PostgreSQL.
24. **Flecha desde PySpark hacia Flask**: PySpark devuelve RDDs y Data frames a Flask a medida que se tratan los datos para poder continuar la transformación.
25. **Flecha desde Pyspark hacia PostgreSQL**: PySpark envía los datos a ser cargados en las tablas de la base de datos de PostgreSQL.
26. **Flecha desde PostgreSQL hacia Pyspark**: PostgreSQL devuelve a PySpark cualquier tipo de error que pudiera ocurrir durante la carga de los datos recibidos
27. **Flecha desde Flask hacia Bokeh**: Flask envía hacia Bokeh todos los datos y parámetros requeridos para construir las visualizaciones.

28. **Flecha desde Bokeh hacia Flask:** Bokeh construye las visualizaciones y las devuelve en un formato el cual puede ser renderizado en las páginas de la visualizador prototípico.
29. **Flecha desde Flask hacia Jinja2:** Flask envía los datos que van a ser mostrados en las páginas web del visualizador prototípico.
30. **Flecha desde Jinja2 hacia Flask:** Jinja2 construye el HTML a ser mostrado y lo devuelve a Flask.
31. **Flecha desde Flask hacia Werkzeug:** Flask envía las contraseñas a ser encriptadas o desencriptadas como así también envía los archivos de los datasets a ser chequeados a Werkzeug.
32. **Flecha desde Werkzeug hacia Flask:** Werkzeug devuelve a Flask las contraseñas recibidas ya sea encriptadas o desencriptadas, dependiendo de lo solicitado. Y también devuelve a Flask si el archivo es un archivo seguro.
33. **Flecha desde Flask hacia Flask Session:** Flask crea una sesión de Flask Session con los datos que necesite utilizar para la navegación en el visualizador prototípico.
34. **Flecha desde Flask Session hacia Flask:** Flask Session devuelve hacia Flask los datos que se hayan guardado en la sesión, cuando se lo solicite Flask.

A continuación se describen las conexiones que cambiaron en el gráfico de la Fig. 21 en relación al de la Fig. 20:

5. **Flecha desde Psycopg2 hacia PostgreSQL:** Las consultas a la base de datos PostgreSQL se realizan desde la conexión de Psycopg2
6. **Flecha desde PostgreSQL hacia Psycopg2:** PostgreSQL devuelve los registros solicitados a través de la conexión realizada a través de Psycopg2.
7. **Flecha desde Tweepy hacia la Api de Twitter:** Se utiliza los métodos de Tweepy para hacer los requerimientos a la Api de Twitter
8. **Flecha desde la Api de Twitter hacia Tweepy:** La Api de Twitter devuelve a Tweepy los objetos solicitados por el requerimiento.
9. **Flecha desde Jinja2 hacia HTML5:** Jinja2 construye los archivos HTML basándose en los estándares de html5 para renderizar correctamente la información.
10. **Flecha desde HTML5 hacia Jinja2:** HTML5 devuelve a Jinja2 Como debe renderizar las páginas de la aplicación.

Por último en el gráfico de la Fig. 22 se profundizará en la implementación de HTML 5 para realizar las páginas web del prototipo de Visualizador. En el gráfico de la Fig. 22 se observa que tecnologías se usaron y las conexiones que estas tienen entre sí y con HTML5

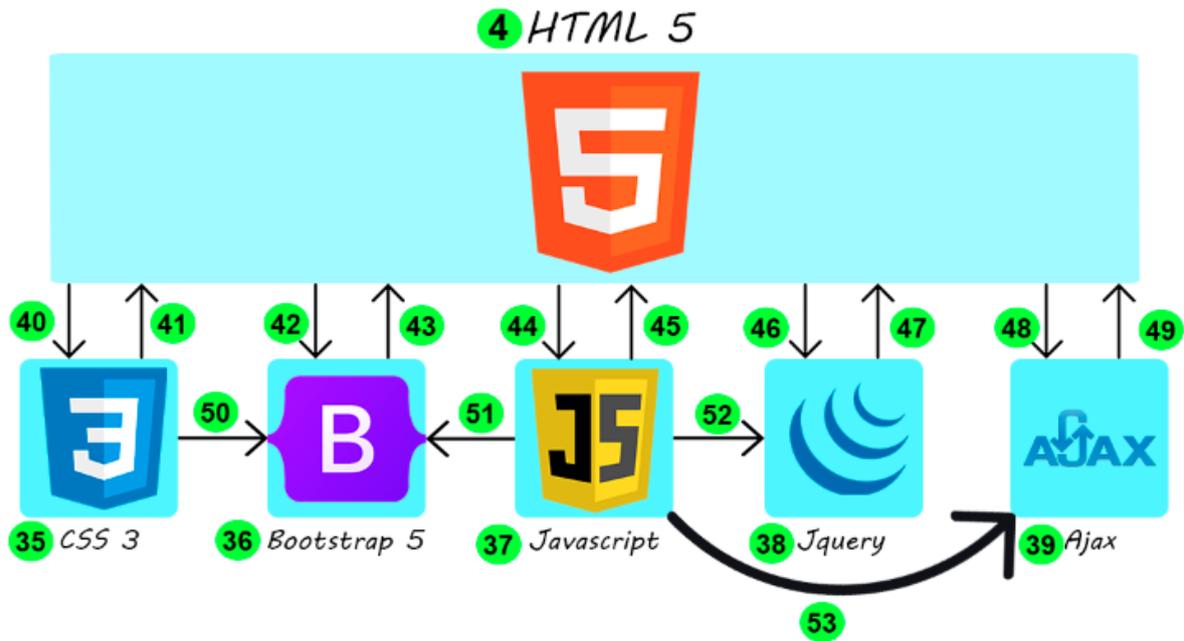


Fig. 22: Vista detallada del módulo HTML5 de la implementación de la arquitectura.

A continuación se realizará una explicación de las referencias de la Fig. 22:

- 35. **CSS 3**: es el lenguaje de estilos utilizado para describir la presentación de documentos HTML. En CSS3 se describe cómo debe ser renderizado cada elemento estructurado en la pantalla
- 36. **Bootstrap 5**: Bootstrap es una librería gratuita y de código abierto constituida por código CSS y JavaScript / jQuery el cual se utiliza para crear sitios web y aplicaciones web con un diseño dinámico.
- 37. **Javascript**: JavaScript (JS) es el lenguaje de programación usado para añadir características interactivas a las páginas web. Este es un lenguaje de programación ligero, interpretado y compilado justo-a-tiempo (Se compila cuando es utilizado, just-in-time).
- 38. **Jquery**: jQuery es una librería de JavaScript que simplifica enormemente la programación de JavaScript. JQuery toma muchas tareas comunes que requieren muchas líneas de código en JavaScript para ser llevadas a cabo y las envuelve en métodos que pueden ser llamados con una sola línea de código.
- 39. **Ajax**: Ajax (Asynchronous JavaScript And XML, Javascript y XML asíncrono) no es un lenguaje de programación en sí mismo, sino que es una combinación de:
 - Un objeto XMLHttpRequest integrado en el navegador (para solicitar datos de un servidor web).
 - JavaScript y HTML (para mostrar o usar los datos).

AJAX permite que las páginas web se actualicen de forma asíncrona mediante el intercambio de datos con un servidor web. Esto significa que es posible actualizar partes de una página web o su completitud, sin tener que recargar toda la página.

- 40. **Flecha desde HTML5 hacia CSS3**: en HTML se especifican los estilos de sus etiquetas los cuales definen como se muestra el contenido de la página web.
- 41. **Flecha desde CSS3 hacia HTML5**: CSS3 es el lenguaje que especifica los estilos utilizados por HTML5
- 42. **Flecha desde HTML5 hacia Bootstrap 5**: Para definir fácilmente un conjunto de

estilos a determinadas etiquetas HTML5 utiliza la librería de Bootstrap 5 para mediante definir clases a las etiquetas se logran dar de una forma muy sencilla estilo y funcionalidad a las etiquetas

- 43. Flecha desde Bootstrap 5 hacia HTML5:** Bootstrap 5 brinda a HTML5 estas clases que pueden ser agregadas a las etiquetas del HTML
- 44. Flecha desde HTML5 hacia Javascript:** En las etiquetas HTML se pueden definir eventos los cuales deben ser resueltos mediante código de JavaScript (Por ejemplo cuando se cliquea en algún lugar de la pantalla)
- 45. Flecha desde Javascript hacia HTML5:** mediante el lenguaje de Javascript se definen comportamientos de las etiquetas HTML ante determinados eventos que ocurran sobre la página web.
- 46. Flecha desde HTML5 hacia JQuery:** Ya que JQuery es una librería de Javascript, los eventos que se pueden definir en las etiquetas de HTML5 pueden ser resueltas usando esta librería.
- 47. Flecha desde JQuery hacia HTML5:** Con esta librería (Jquery) se pueden definir comportamientos de las etiquetas HTML ante los distintos eventos que pueden ocurrir en una página web.
- 48. Flecha desde HTML5 hacia Ajax:** Mediante eventos que ocurren en HTML se inician métodos Javascript los cuales pueden ser Ajax.
- 49. Flecha desde Ajax hacia HTML5:** los métodos que contienen Ajax son utilizados para hacer consultas al servidor y con los datos que obtienen resolver una determinada lógica o simplemente renderizarlos mediante JavaScript.
- 50. Flecha desde CSS3 hacia Bootstrap 5:** Bootstrap 5 genera sus clases (las cuales son utilizadas por las etiquetas de HTML) utilizando CSS3 y JavaScript
- 51. Flecha desde Javascript hacia Bootstrap 5:** Bootstrap 5 genera sus clases (las cuales son utilizadas por las etiquetas de HTML) utilizando CSS3 y JavaScript
- 52. Flecha desde Javascript hacia JQuery:** JQuery es una librería de Javascript que cumple su misma función pero de un modo más simple.
- 53. Flecha desde Javascript hacia Ajax:** Los métodos Asíncronicos de Ajax son utilizados mediante codificación Javascript.

En el Anexo 3 se puede observar el gráfico resultante al unir las Figuras 20, 21 y 22. El código del prototipo propuesto se encuentra disponible en:

<https://github.com/sebaber123/CeoDatum>

Capítulo 6: Funcionamiento de CeoDatum

En este capítulo se detalla el funcionamiento del prototipo de visualizador desarrollado. Por un lado se mencionan los roles de usuario posibles y por otro las funcionalidades permitidas, listadas a continuación:

- Login y registro de usuarios en la aplicación.
- Carga de datasets a la aplicación.
- Generación de gráficos y visualizaciones a partir de los datasets.
- Vista de los datasets.
- Buscador de recurrencias de palabras en tweets (Grafos sociales).
- Vista y creación de cursos.
- Vista y creación de actividades.
- Realización de entregas de actividades.
- Corrección de actividades.
- Acceso al perfil de usuario.
- Configuración del sistema.

Cabe aclarar que, salvo para el login y registro, la utilización de las funciones necesitan la autenticación previa del usuario.

Roles

Se definieron 4 roles posibles: super-admin, admin, profesor y estudiante. Los usuarios pueden tener más de un rol. Un usuario sin roles se considerará un usuario inactivo. A continuación se detallan los permisos de cada uno de ellos:

- **Superadmin:** Tiene permitido el acceso a la configuración. Es el único habilitado para asignar/quitar el rol de administrador a otros usuarios.
- **Admin:** Tiene permitido el acceso a la configuración. Puede cargar el dataset de instituciones. Puede asignar o quitar roles de profesor y estudiantes a los usuarios.
- **Profesor:** Tiene acceso y permiso de carga de datasets. Tiene acceso a la pestaña de cursos. Puede crear cursos. Puede crear actividades. Puede calificar entregas de actividades incluyendo comentarios para los estudiantes. Tiene acceso a la pestaña de datasets, incluyendo los propios cargados. Tiene acceso a la pestaña de grafos sociales. Tiene acceso a la pestaña perfil.
- **Estudiante:** Tiene acceso a la pestaña de actividades. Puede realizar entregas de actividades. Tiene acceso a la vista de datasets protegidos y públicos. Puede acceder a la pestaña de grafos sociales. Tiene acceso a la pestaña perfil.

Funcionalidades

En la siguiente sección se realiza una descripción detallada de las funcionalidades cubiertas en el prototipo de visualizador web.

Log in y registro en la aplicación

El login y registro de la aplicación se presentan a través de dos formularios que se pueden observar en la Fig. 23.1 y la Fig. 23.2 respectivamente. El acceso al formulario de login es a través de la barra navegadora y en él, se encuentra el acceso al formulario de

registro. Un usuario registrado comienza con el rol de estudiante. Para poder registrarse, no debe existir un usuario con el mismo nombre de usuario o el mismo email asociado. Los campos del login son el nombre de usuario o email y la contraseña. Los campos del formulario de registro son:

1. **Correo electrónico:** cadena de caracteres obligatoria única.
2. **Nombre de usuario:** cadena de caracteres obligatoria única.
3. **Contraseña:** cadena de mínimo 8 caracteres obligatoria.
4. **Confirmación de contraseña:** cadena de caracteres que debe ser equivalente a contraseña
5. **Nombre:** cadena de caracteres obligatoria.
6. **Apellido:** cadena de caracteres obligatoria.
7. **Fecha de nacimiento:** campo de tipo fecha obligatorio.
8. **Provincia:** Selección de valores obligatoria
9. **Ciudad:** Selección de valores obligatoria, según la provincia elegida
10. **Institución:** Selección de valores, según la ciudad elegida.

A continuación se muestra una imagen con la vista del Login (Fig. 23.1):



Fig. 23.1: Formulario de login de usuarios de CeoDatum

Una vez presentada la Fig. 23.1 se listan los elementos de la pantalla de login:

1. **Campo para ingresar el nombre de usuario:** input de nombre de usuario.
2. **Campo para ingresar la contraseña:** input de contraseña.
3. **Botón para autenticarse:** botón de submit de formulario de login
4. **Botón para ingresar al formulario de registro:** link a página de registro de usuario.
5. **Botón para contraseña olvidada:** link a formulario de recuperación de contraseña.

Por otra parte, a continuación se observa una imagen del formulario de registro de usuarios (Fig. 23.2):

Formulario de registro

Correo electrónico (*)

1

Nombre de usuario (*)

2

Contraseña (*)

3

Repetir contraseña (*)

4

Nombre (*)

5

Apellido (*)

6

Fecha de nacimiento (*)

7

Provincia (*)

8

Ciudad (*)

9

Establecimiento educativo (*)

10

11 **12**

(*) Campos obligatorios

Fig 23.2: Formulario de registro de usuarios de CeoDatum

Presentada la Fig. 23.2 se procede a listar los elementos enumerados en ella:

- 1-10.Inputs del formulario:** Inputs del formulario de registro. Fueron listados anteriormente.
- 11. Botón para realizar el registro:** Submit del formulario de registro.
- 12. Botón de retroceso:** Link a página de login.

Carga de datasets a la aplicación

En esta sección se explicará cómo funciona la carga de los datasets a la prototipo de visualizador web. Para esto se inicia en la pantalla mostrada en la Fig. 24.1.

CeoDatum Datasets ▾ Gráfos sociales ▾ Hola, **sebastian** [Cerrar sesión](#)

Nuevo dataset

Haga click en examinar o suelte un archivo (CSV o JSON) sobre el area marcada.

Browse...

Subir

Fig. 24.1: Pantalla de carga de datasets de Ceodatum

A continuación se listan los elementos enumerados en la Fig. 24.1:

1. **Campo para ingresar el archivo:** En este campo se debe ingresar el archivo a subir ya sea cliqueando en el campo y eligiendo el archivo o arrastrando y soltando el archivo en el campo
2. **Botón de “subir”:** al presionar este botón, si se ingresó un archivo de tipo CSV o JSON en el campo que se solicita el archivo se procederá a subir el archivo a la aplicación y continuar con el proceso de creación del dataset. Ya que dependiendo de si es un archivo CSV o un archivo JSON se solicitan al usuario distintos valores.

Luego de subir el archivo, si se ingresó un archivo CSV se procede a la pantalla visualizada en la Figura 24.2.

Fig 24.2: Formulario de ingreso de datos del dataset cargado en formato CSV a GeoDatum

En la Fig. 24.2 se observa una pantalla en la que se solicitan los datos asociados a un dataset. A continuación se listan los elementos enumerados:

3. **Campo del nombre del dataset:** En este campo se solicita al usuario que ingrese el nombre que tendrá el dataset, que es el nombre que llevará la base de datos en la que se guardan sus datos como así también por el cual se lo buscara en la aplicación.
4. **Campo del estado del dataset:** En este área se pide al usuario que elija entre las opciones público, protegido y privado. Público será que todos pueden acceder al dataset, protegido será que brinda acceso a todos los que se encuentre en su misma

escuela y privado será que solo el que lo subió tiene acceso a menos que lo comparta en un curso.

5. **Campo de formato de campos tipo fecha:** debido a que existe una gran variedad de campos tipo fecha y resultó imposible detectar cual se está usando dinámicamente se solicita al usuario que ingrese qué formato utiliza en su archivo (si no tiene campos de tipo fecha esto no afectará en nada.). Los formatos entre los que se pueden elegir son todas las combinaciones de posiciones entre día, mes y año, separadas ya sea por "/" o "-" y también si el año es de 2 o 4 números.
6. **Campo de títulos:** Dado que los CSV pueden contener los títulos de las columnas en su primera fila, se pregunta sobre esto para saber cómo tratar el archivo una vez subido.
7. **Sección de columnas:** A partir de este título se mostrará por cada columna del archivo un valor de ejemplo y que nombre se elegirá para dicha los datos de dicha columna
8. **Valor de ejemplo:** Durante la carga se extraen los valores de la primer fila del archivo para mostrarlos en esta sección así el usuario puede saber con más facilidad que nombre ponerle a dicha columna ya que tiene una referencia
9. **Nombre de la columna:** En este campo se le solicita al usuario que elija un nombre que se usará para tratar a los datos de dicha columna. Cabe aclarar que los nombres de estas columnas deben ser todos distintos (la aplicación verifica esto) para así poder guardarlos correctamente.
10. **Botón de finalizar:** Al presionar en este botón se chequea que todos los campos hayan sido completados y estén correctamente completados en el caso de los nombres de las columnas. Luego se continúa la carga del CSV la cual consta de dos procesos que son la creación de la base de datos que contendrá los datos y la subida de los datos a la misma.

En cambio, si el archivo que se subió es de tipo JSON se procede a la pantalla visualizada en la Fig. 24.3 en la que se solicita al usuario que ingrese la información asociada al dataset.



The screenshot shows a web interface for CeoDatum. At the top, there is a navigation bar with 'CeoDatum', 'Datasets', and 'Grafos sociales'. On the right, it says 'Hola, sebastian' and 'Cerrar sesión'. The main content area has a blue background and contains a form titled 'por favor ingrese los datos solicitados'. The form has three numbered input fields: 3 is a text input for 'ingrese el nombre de la base de datos:', 4 is a dropdown menu for 'Estado del dataset:' with 'Publico' selected, and 5 is a dropdown menu for 'Formato de campos de tipo fecha:' with 'año(2 numeros)/día/mes' selected. At the bottom of the form is a blue button labeled 'Finalizar' with the number 11 next to it.

Fig 24.3: Formulario de ingreso de datos del dataset cargado en formato JSON a CeoDatum

En la Fig. 24.3 presentada se enumeran los elementos de la pantalla de carga de datasets JSON. Los elementos 3, 4 y 5 son los mismos que en la pantalla de carga de datasets en formato CSV mostrada en la Fig. 24.2. Se agrega un nuevo elemento que es:

- 11. Botón de finalizar:** Al presionar en este botón se chequea que todos los campos hayan sido completados. Luego se continúa la carga del archivo JSON la cual consta de dos procesos que son la creación de la base de datos que contendrá los datos y la subida de los datos a la misma.

Generar gráficos a partir de los datasets

A continuación se explicará el funcionamiento del graficador que se ha construido en este prototipo de visualizador. En la Fig. 25.1 se muestra cómo es la pantalla final del graficador.

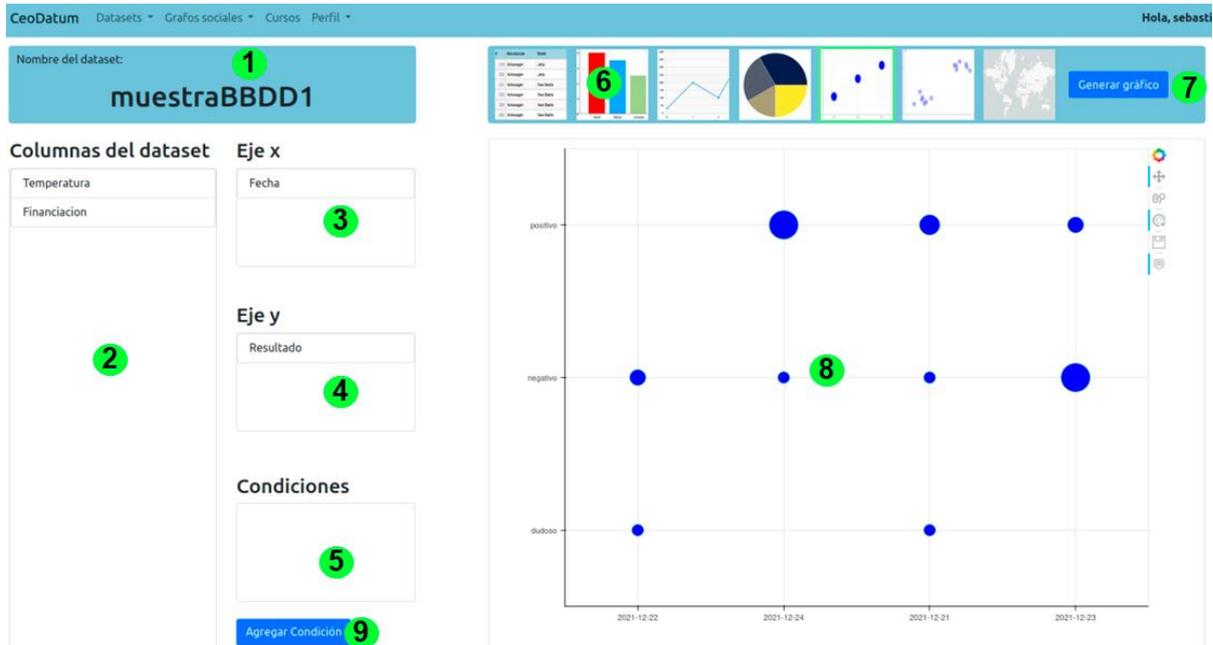


Fig. 25.1: Pantalla de generación de visualizaciones de un dataset específico de CeoDatum

Una vez presentada la Fig. 25.1 se procede a definir los elementos de la pantalla enumerados en ella:

- 1. Título del dataset:** En este espacio se puede ver el Título del dataset que es el mismo que le ha dado el usuario que lo sube.
- 2. Columnas del dataset:** En esta caja (Box) se pueden ver las columnas por las que se puede graficar en el dataset. Estas columnas son elementos que se arrastran y soltan en las cajas de “Eje x” y “Eje y” para generar el gráfico a partir de las columnas que se suelten en estas dos cajas.
- 3. Caja de Eje X/Latitud:** Este espacio (caja) “Eje X” es en el cual se debe soltar una de las columnas del dataset para así a partir de esto generar el gráfico. Cuando se selecciona el gráfico de mapa en este campo se debe soltar la columna que indica la latitud (Cambia el título de la caja a Latitud).
- 4. Caja de Eje X/Longitud:** En esta área (caja) “Eje y” se puede soltar una de las columnas del dataset para especificar sobre qué columna del dataset se debe graficar el Eje Y. Si se selecciona el gráfico mapa, en esta área se debe soltar una columna que indique la longitud (Cambia el título del área a longitud)
- 5. Caja de condiciones:** en este área se verán las condiciones que se hayan agregado a la visualización.

6. **Selección de gráfico:** en esta área se debe seleccionar el gráfico que se quiere realizar. En base al gráfico que se seleccione cambiarán los títulos de la “Caja de Eje X” y la “Caja de Eje Y”, como también cambiarán distintas opciones que se pueden generar agregar al gráfico
7. **Botón de “Generar gráfico”:** Al presionar este botón se valida que se haya elegido un gráfico en el seleccionador de gráfico y que se haya soltado las correspondientes columnas en la “caja de Eje X” y de ser necesario en la “caja de Eje Y”. Si se ha seleccionado un gráfico y soltado una columna en las correspondientes cajas se generará el gráfico.
8. **Área de gráfico:** en este área se devolverá el gráfico una vez que se haya presionado el botón de “Generar gráfico”. A la derecha del gráfico se pueden apreciar las opciones que se tiene en el mismo, las cuales son: Mover, Zoom, activar o desactivar función de presionar, guardar en formato PNG y activar o desactivar la información que se ve cuando se pasa el mouse por encima de los elementos del gráfico (en este caso son barras). La función de presionar hace que si se presiona uno de los elementos del gráfico (en este caso el elemento son las barras) se abre una nueva pestaña para poder inspeccionar los datos de los registros que se mostraron en ese elemento del gráfico.
9. **Botón de “Agregar condición”:** cuando se presiona en este botón se abre el panel en el cuál se pueden crear condiciones para el gráfico. Este panel se puede observar a continuación en la Fig. 25.2:

The image shows a modal window titled "Genere el filtro siguiendo los pasos" with a close button (X) in the top right corner. The window contains three numbered steps:

- 10. Seleccione la columna por la que quiere filtrar:**
 - Fecha
 - Temperatura
 - Financiacion
 - Resultado
- 11. Seleccione el criterio para filtrar la columna seleccionada:**
 - Incluir
 - Excluir
 - Similar a la palabra
 - Mayor que
 - Menor que
 - Dentro del rango
- 12. Ingrese/seleccione el valor o los valores para el criterio de filtrado:**

mm / dd / yyyy

At the bottom right of the window, there are two buttons: "Cerrar" (grey) and "Agregar" (blue). A green circle with the number "13" is positioned next to the "Agregar" button.

Fig. 25.2: Ventana emergente de ingreso de condiciones al visualizador de datasets de CeoDatum

A continuación se listan los elementos enumerados en el panel de condiciones de la Fig. 25.2 que emerge de la pantalla del visualizador observado en la Fig. 25.1:

- 10. Seleccionar la variable:** en esta sección se debe seleccionar la variable por la cual se quiere generar la condición. dependiendo del tipo de variable se podrán hacer distintas formas de cómo filtrar.
- 11. Seleccionar como filtrar:** en este área se puede elegir qué valor incluir/excluir sin importar el tipo de variable y dependiendo el tipo de variable se puede agregar una palabra que contenga si es de tipo cadena de caracteres. Y si es de tipo fecha o número se le puede agregar un valor al que debe ser mayor o menor o dentro de un rango en que debe estar.
- 12. Ingresar valor:** En este área se debe seleccionar los valor a incluir/excluir si se selecciona esta opción o sino ingresar el valor por el que se desee filtrar
- 13. Botón “Agregar”:** Cuando se presiona este botón se agrega la condición a la Caja de condiciones con un botón para borrarla en caso de ser necesario.

Inspección de datos

En esta sección se explicará qué se muestra cuando se inspecciona un elemento del graficador mostrado en la sección “Generar gráficos a partir de los datasets”. Cuando se presiona uno de los elementos del gráfico obtenido (ya sea por ejemplo una barra en el gráfico de barras o una porción de torta en el gráfico de torta), automáticamente se abre una nueva ventana en la que se detalla los datos que representa dicho elemento, como se puede observar en la siguiente imagen (Fig. 26):

Explore los datos de la base de datos Tests_de_covid

Fecha	Temperatura	Financiacion	Resultado
2021-12-23	36	publico	negativo
2021-12-23	36	publico	negativo
2021-12-23	37	publico	negativo
2021-12-23	37	privado	positivo
2021-12-23	38	publico	negativo
2021-12-23	37	publico	negativo
2021-12-23	37	privado	positivo

Fig. 26: Pantalla de exploración de datos de un dataset cargado en CeoDatum

Vistas de los datasets

En el siguiente apartado se explicará cómo se accede a buscar los distintos datasets que se encuentran en la plataforma prototípica web. Para esto, como se observa en la imagen representada por la Fig. 27.1, se emplea un Dropdown de “Datasets” situado en la barra de navegación.



Fig. 27.1: Dropdown de navegación de datasets de CeoDatum

A continuación se procede a listar los elementos enumerados en la Fig. 27.1:

1. **DropDown Datasets:** en este DropDown que se encuentra en la barra de navegación se puede encontrar los botones a las distintas formas que se pueden listar los datasets a los que tiene acceso el usuario.
2. **Botón de Datasets Públicos:** Cuando se presiona en este botón se muestran el listado de los Datasets que se encuentran en estado público.
3. **Botón de Datasets Protegidos:** Una vez que el usuario presiona en este botón se accede a la pantalla que muestran los datasets a los que el usuario posee acceso. Estos son, los que se le brindó por estar en un determinado establecimiento o los que se le dio acceso mediante los cursos en los que se encuentra
4. **Botón de mis Datasets:** Cuando el usuario selecciona este botón se le lleva a la pantalla donde se listan sus propios datasets. Solo accesible para usuarios profesores.

Por otra parte, en la Fig. 27.2, se observa la pantalla de vista de datasets. Dependiendo de las características del conjunto de datos que se está buscando (públicos, protegidos o los datasets propios del usuario), cambiará cuáles de ellos se mostrarán al usuario.

Datasets Públicos

Nombre del dataset 5	Subido por 6	Acciones 7
csvConHeaders	sebastian bertora	Inspeccionar Graficar
testPublic	sebastian bertora	Inspeccionar Graficar
starbucksprueba	sebastian bertora	Inspeccionar Graficar
test2.Json	sebastian bertora	Inspeccionar Graficar
pruebaValidacion	sebastian bertora	Inspeccionar Graficar

[Anterior](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Siguiete](#)

8

Fig. 27.2: Pantalla de listado de datasets públicos de CeoDatum

Se procede a listar los elementos de la Fig. 27.2 previamente presentada:

5. **Columna de nombre:** En esta columna se puede encontrar los nombres de los datasets que se están listando en la pantalla.
6. **Columna de Subido por:** Se puede ver en esta columna la persona que ha subido cada uno de los Datasets que se ven en el listado.

7. **Columna de acciones:** En esta columna se muestran los botones de “Graficar” el cual lleva al graficador para que el usuario genere los gráficos que quiera hacer y el botón de inspeccionar en el cual se pueden ver distintos datos del Dataset Como se muestra en Fig. 27.2.
8. **Paginación:** la sección que se muestra debajo del Dataset se utiliza para navegar entre las páginas del listado del Dataset. Cuando se presiona un número se va a la página de ese número y el botón de “anterior” y “siguiente” llevan a la página anterior y siguiente si es que esto es posible.

En la siguiente imagen (Fig 27.3) se muestra la información y funciones que pueden realizarse cuando se inspecciona un dataset.

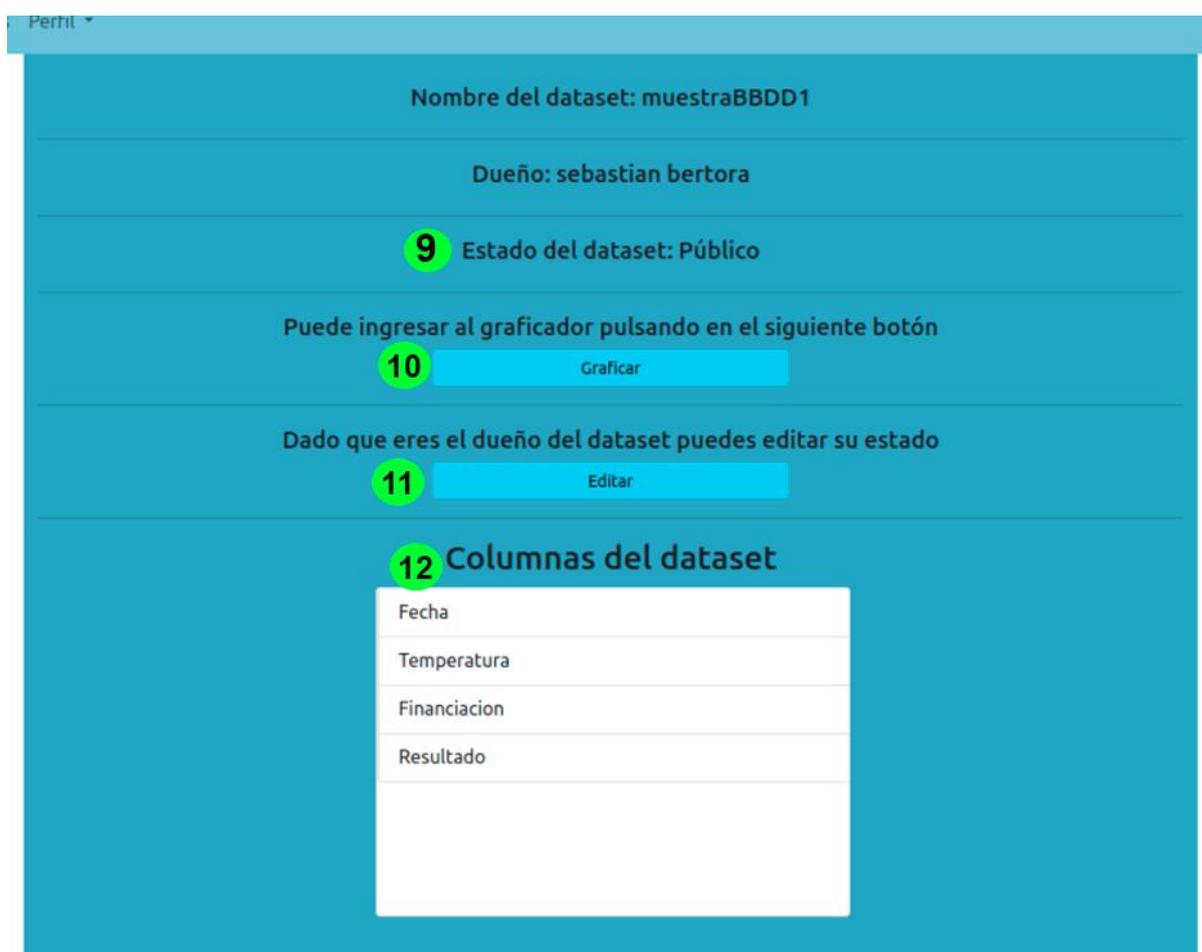


Fig. 27.3: Pantalla de listado de datos de un dataset seleccionado en CeoDatum

A continuación se listan los elementos enumerados en la Fig. 27.3:

9. **Área de “Estado”:** en este área se muestra cual es el estado del Dataset, si es que este es Público, protegido o privado.
10. **Botón de Graficar:** Cuando el usuario presiona este botón se lleva al usuario a la pantalla del graficador.
11. **Sección de Editar:** Si el usuario que está viendo esta página es el dueño del dataset, se le renderiza esta sección en la que puede modificar el estado del Dataset.

12. Sección de columnas: En esta sección se muestran las columnas que se tiene en el dataset y que son los datos por los que se va a poder graficar.

Buscador de recurrencias de palabras en tweets

En este apartado se muestra la página de buscador de recurrencias de palabras en Twitter. Esto se realiza en conjunto con la API de la red social, la cual devuelve los tweets que contienen una determinada palabra o poseen alguna relación con ella. En la Fig. 28.1, que muestra la pantalla del contador de recurrencias mencionado y el acceso a una nube de palabras generada a partir de él.

Fig. 28.1: Pantalla de exploración de grafos sociales de CeoDatum

A continuación se listan los elementos enumerados en la Fig. 28.1 previamente presentada:

- 1. DropDown de “Grafos sociales”:** En este DropDown se encuentra un botón para buscar las recurrencias de palabras en determinados tweets.
- 2. Campo de palabra o frase:** En este campo se debe escribir la palabra o frase por la cual se quiere buscar que contengan o hablen mismo tema, los tweets a pedir a la API de Twitter.
- 3. Cantidad de palabras:** una de las funcionalidades es que al buscar los tweets se construye un grafo en el que aparecen las palabras representadas por nodos, la cantidad de nodos que aparece depende del número que se elija en este campo.
- 4. Sección de excluir:** en esta sección se muestran las palabras que se pueden excluir del grafo y nube de palabras que se generará. Esto se hace debido a que si se excluyen estas palabras puede aparecer información más útil para el usuario
- 5. Botón de “Buscar”:** Al presionar este botón se genera el grafo y la nube de palabras a partir de los parámetros ingresados por el usuario (palabra/frase, cantidad de palabras y exclusiones).
- 6. Botón de grafo:** Cuando se presiona este botón (sólo aparece cuando se ha presionado el botón de buscar) se despliega una sección donde se puede observar un grafo. En él, los nodos representan las palabras, su tamaño la cantidad de veces que apareció la palabra (si el nodo es más grande es porque esta palabra tuvo más apariciones) y las aristas cuántas veces aparecieron junto con la palabra que se relaciona. Esto se puede apreciar en la Fig. 28.2.

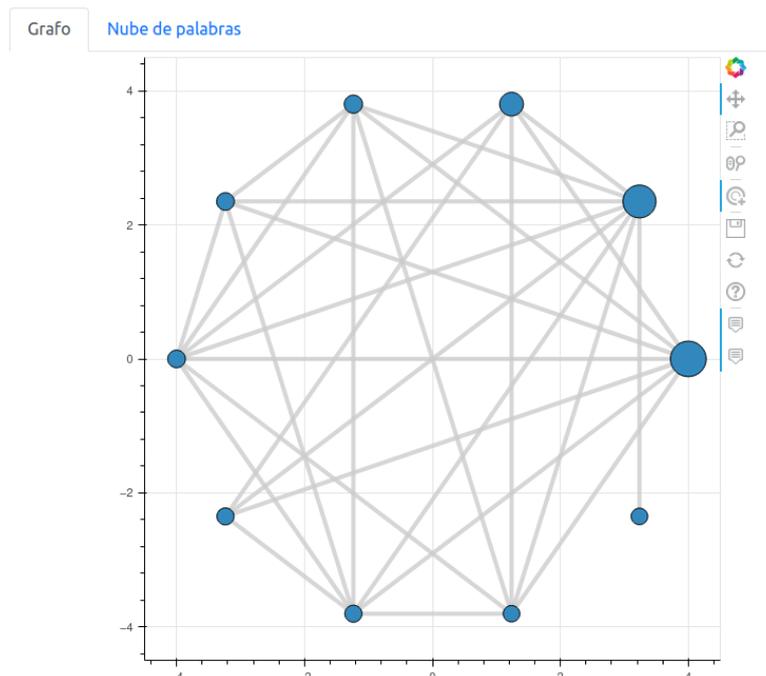


Fig. 28.2: Ejemplo de grafo social creado en CeoDatum

- Botón de Nube de palabras:** Sólo aparece una vez presionado el botón buscar. Al presionarlo se despliega una sección donde se observa una nube de palabras generada a partir de la información devuelta por la API de Twitter. El tamaño de las palabras es proporcional a la cantidad de veces que aparece dicha palabra. Esto puede observarse en Fig. 28.3:



Fig. 28.3: Ejemplo de nube de palabras creada en CeoDatum

Vista y Creación de cursos

A través de la pestaña cursos en la barra de navegación, un profesor puede acceder a los cursos creados por el usuario autenticado. La lista de cursos se muestra cómo en la Fig. 29.1. En esta vista se permite la creación de un nuevo curso. La creación de un curso se realiza a través de un formulario que se puede observar en la Fig. 29.2. El formulario tiene los siguientes campos:

- Nombre de curso: cadena de caracteres obligatoria.
- Fecha de inicio: dato tipo fecha.
- Fecha de fin: dato tipo fecha

Una vez creado un curso o para un curso ya creado, un profesor puede acceder a los datos del mismo a través de la pestaña cursos y haciendo clic en la lista sobre el que se desee. Cómo se observa en la Fig. 29.3, los datos del curso mostrados son: los datos propios del curso, los estudiantes inscriptos, las actividades asignadas al curso y los datasets ligados al mismo.



Nombre	Fecha de comienzo	Fecha de fin	Establecimiento educativo
Administración de Empresas - 2022	2022-02-21	2022-12-05	INSTITUTO PIO XII
Matemática quinto 2022	2022-03-07	2022-12-05	ESCUELA DE EDUCACIÓN SECUNDARIA N°36

Crear nuevo curso

Fig 29.1: Pantalla del listado de cursos de un usuario profesor autenticado en CeoDatum

Nuevo curso



Nombre
Ciencias Sociales - 2021

Fecha de inicio
15 / 03 / 2021

Fecha de fin
15 / 08 / 2022

Establecimiento educativo
ESCUELA DE EDUCACIÓN SECUNDARIA N°36

Ciclo
Ciclo básico

Año
Primero

Ámbito curricular
Ciencias Sociales

Crear curso

Fig. 29.2: Formulario de creación de nuevo curso en CeoDatum

Administración de Empresas - 2022

✓ **Estudiantes inscriptos** 1

Nombre 4	Email 5	Nombre de usuario 6
Amado, Delfina	delfina.amado@gmail.com	delfinaamado
Falzone, Lucia	luchafalzone@gmail.com 7	luciafalzone
Occhionero, Nicole	nicleocchionero@gmail.com	NicoleOcchio

Añadir usuarios al curso:
- Seleccione un estudiante de la lista -

Invitar 8

✓ **Actividades** 2

Título 9	Descripción 10	Estado 11
Actividad de reconocimiento	A partir del dataset de ejemplo, utilizar la herramienta para hallar el máximo de ventas	Finaliza el 2022-03-27

Nueva actividad 12

Nueva actividad 13

✓ **Datasets** 3

Nombre del dataset 14	Subido por 15	Acciones 16
Juegos 17	ernesto lopez	Inspeccionar Graficar

Agregar dataset 18

Fig. 29.3: Pantalla de datos ligados a un curso en CeoDatum

A continuación se listan los elementos enumerados en la pantalla de cursos de CeoDatum observados en la Fig. 29.1 ya introducida:

1. **Pestaña cursos de la barra de navegación:** al hacer clic sobre ella se redirige a esta pantalla.
2. **Nombre del curso:** Columna de la lista de cursos con el nombre de cada uno.
3. **Fecha de comienzo:** Columna de la lista de cursos con la fecha de comienzo de cada uno.
4. **Fecha de fin:** Columna de la lista de cursos con la fecha de fin de cada uno.
5. **Establecimiento educativo:** Columna de la lista de cursos con el nombre del establecimiento educativo de cada uno.
6. **Filas de cursos:** Cada fila representa un curso. Al hacer clic sobre ellas se redirige al usuario a la pantalla de información del curso indicado.
7. **Botón de creación de nuevo curso:** Link a formulario de creación de curso.

Por otra parte, se listan los elementos enumerados en la pantalla de creación de un nuevo curso, según se observa en la Fig. 29.2:

1. **Inputs del formulario de creación de cursos:** Campos a completar por el usuario según los datos del curso a crear. Descritos anteriormente.
2. **Botón de submit de formulario:** Botón para crear el curso. Si los campos son correctos, redirige a la pantalla de cursos.

En último lugar se listan los elementos enumerados en la página de información de un curso determinado mostrados en la Fig. 29.3:

1. **Dropdown de estudiantes inscriptos:** Haciendo clic sobre este dropdown se mostrará u ocultará la lista de estudiantes inscriptos al curso.
2. **Dropdown de actividades asociadas al curso:** Haciendo clic sobre este dropdown se mostrará u ocultará las actividades asociadas al curso.
3. **Dropdown de datasets asociados al curso:** Haciendo clic sobre este dropdown se mostrará u ocultará los datasets asociados al curso.
4. **Nombres de estudiantes:** Columna de los nombres de estudiantes ligados al curso.
5. **Emails de estudiantes:** Columna de emails de estudiantes ligados a un curso.
6. **Nombres de usuarios de estudiantes:** Columna de nombres de usuarios de estudiantes ligados al curso.
7. **Filas de estudiantes:** Cada fila representa un estudiante participante del curso.
8. **Input de invitación de estudiantes al curso:** En este input se puede seleccionar el estudiante para invitarlo al curso haciendo clic sobre el botón invitar. El estudiante debe estar ligado al instituto del curso.
9. **Títulos de actividades:** Columna de títulos de actividades ligadas al curso.
10. **Descripciones de actividades:** Columna de descripciones de actividades ligadas al curso.
11. **Estados de actividades:** Columna de estados de actividades ligadas al curso.
12. **Filas de actividades:** Cada fila representa una actividad ligada al curso. Al hacer clic sobre ella se redirigirá a la pantalla de información de la actividad seleccionada.
13. **Botón para creación de nueva actividad:** Link al formulario de creación de actividades.
14. **Nombres de datasets asociados al curso:** Columna de nombres de datasets asociados al curso.
15. **Usuarios propietarios de los datasets asociados al curso:** Columna de propietarios de datasets asociados al curso.
16. **Acciones posibles a realizar sobre los datasets asociados al curso:** Columna de acciones posibles a realizar sobre un dataset asociado al curso.
17. **Datasets asociados al curso:** Cada fila representa un dataset asociado al curso.
18. **Botón para asociar datasets al curso:** Este botón despliega una ventana emergente con datasets para asociar al curso.

Vista y Creación de actividades

En la vista de información de un curso se permite crear actividades para el mismo. La vista de creación de actividades se puede observar en la Fig. 30.1. Consta de un formulario con los siguientes campos:

- Título de actividad: cadena de máximo 64 caracteres obligatoria.
- Enunciado: sección de enunciado de actividad
 - Título de enunciado: cadena de máximo 64 caracteres obligatoria.
 - Consigna: cadena de máximo 264 caracteres obligatoria.
- Objetivo: cadena de máximo 264 caracteres obligatoria.
- Descripción: cadena de máximo 264 caracteres obligatoria.
- Dataset: selección de valores entre los datasets elegibles.
- Gráficos permitidos: conjunto de checkbox de visualizaciones habilitadas en la resolución de la actividad.
- Fecha de inicio: campo de tipo fecha obligatorio.

- Fecha de fin: campo de tipo fecha obligatorio.
- Estudiantes: selección entre todos los estudiantes o un conjunto de ellos:
 - Estudiantes ligados: conjunto de checkbox con los estudiantes designados a realizar la actividad
- Entrega fuera de término: checkbox que permite la entrega fuera de término de la actividad:
- Calificación: checkbox que determina si la actividad posee calificación numérica.

Por otra parte, para acceder a la información de una actividad, un profesor puede hacerlo a través del curso en el que fue creada la actividad. Esto lo realiza en la sección de actividades, haciendo clic sobre la actividad deseada. La vista de actividad para un profesor muestra los datos de la misma como se observa en la Fig. 30.2. Se muestran tres pestañas, en una el estado de la entrega de los estudiantes, en otra el enunciado de la actividad y en la última los datos de la misma. Por otro lado, un estudiante accede a sus actividades a través de la pestaña Actividades de la barra de navegación. En esta se lista las actividades, como muestra la Fig. 30.3, en cinco grupos:

- Pendientes: muestra aquellas actividades creadas para el estudiante que aún no ha superado la fecha de fin y que el estudiante no ha resuelto. Permite resolverlas.
- Atrasadas: muestra aquellas actividades creadas para el estudiante que aún no ha resuelto y que la fecha de fin ha pasado, pero que permite la entrega fuera de término. Permite resolverlas.
- Corregidas: muestra aquellas actividades que el estudiante ha resuelto y el profesor ha realizado la corrección. Permite acceder a la información de la entrega realizada.
- Entregas: muestra aquellas actividades que el estudiante ha resuelto pero el profesor aún no ha realizado la corrección. Permite acceder a la información de la entrega realizada.
- Vencidas: muestra aquellas actividades creadas para el estudiante que aún no ha resuelto y que la fecha de fin ha pasado y no se permite la entrega fuera de término.

Nueva actividad

Título de actividad Máximo 64 caracteres

Ingrese un título

Enunciado

Título del enunciado Máximo 64 caracteres

Consigna Máximo 256 caracteres

Objetivo Máximo 256 caracteres

Descripción Máximo 256 caracteres

Dataset

Gráficos permitidos

Tabla de datos Gráfico de barras Gráfico de líneas Gráfico de torta Gráfico de puntos Gráfico de dispersión Mapa

Grafos sociales

Fecha de publicación

dd / mm / aaaa

Fecha de entrega

dd / mm / aaaa

Crear actividad para:

Todos los estudiantes

Permitir entrega fuera de término

Actividad sin calificación

Crear actividad

Fig. 30.1: Formulario de creación de actividad en CeoDatum

Cursos Perfil ▾

1 Título de act

desc

2 Estado de entregas **3** Enunciado **4** Datos de actividad

5 Alumno	6 Estado	7 Fecha de entrega	8 Nota	9 Ver entrega
martin bertora	Calificado	2021-12-17 23:21:00	7	10 Ver entrega
se ba	No calificado	2021-12-19 18:29:55	No calificado	Ver entrega

Fig. 30.2: Pantalla de datos ligados a una actividad en CeoDatum

fos sociales ▾ Actividades **1** ▾

Mis actividades

2 Pendientes **3** Atrasadas **4** Corregidas **5** Entregadas **6** Vencidas

7 Título	8 Curso	9 Fecha de comienzo	10 Fecha de cierre	11
sin grafo	nombreqwe	31-12-23	31-12-99	Resolver
sin plotter	nombreqwe	31-12-31	23-12-99	Resolver
titulo act	nombreqwe	17-12-21	27-12-21	Resolver

Fig. 30.3: Pantalla de actividades ligados a un estudiante en CeoDatum

A continuación se detallarán los elementos enumerados en la pantalla de creación de actividades mostrada en la Fig. 30.1:

1. **Campos del formulario de creación de actividades:** los campos fueron descritos con anterioridad.
2. **Submit de formulario de creación de actividad:** si todos los campos son correctos, se procede a crear la actividad y asociarla a los estudiantes seleccionados. Se procede a la pantalla del curso asociado.

Por otro lado, se listan los elementos enumerados de la pantalla de información de una actividad determinada plasmados en la Fig. 30.2:

1. **Título y descripción de actividad:** Título y descripción de la actividad seleccionada.
2. **Pestaña de estados de entregas:** Al hacer clic sobre ella se muestran aquellas entregas realizadas para la actividad seleccionada.
3. **Pestaña de enunciado:** Al hacer clic sobre ella se muestran los datos asociados al enunciado de la actividad seleccionada.
4. **Pestaña de datos:** Al hacer clic sobre ella se muestran los datos asociados a la actividad seleccionada, sin incluir el enunciado.
5. **estudiantes que realizaron la entrega:** Columna que muestra el nombre de los estudiantes que realizaron la entrega de la actividad seleccionada.
6. **Estado de las entregas:** Columna que muestra el estado de la corrección de la entrega realizada por el estudiante en la actividad seleccionada. Puede ser calificado o no calificado.
7. **Fechas de entregas:** Columna que muestra la fecha de la entrega realizada por el estudiante de la actividad seleccionada.
8. **Notas de las entregas:** Nota asignada al estudiante que realizó la entrega de la actividad seleccionada luego de corregirla. Si no se ha corregido aparece "No calificado". Si la actividad no tiene nota numérica aparece "Sin calificación".
9. **Ver entrega:** Botón que permite ver la entrega del estudiante de la actividad seleccionada. Procede a la vista de la entrega y el formulario de corrección de la misma.

En última instancia, se listan los elementos enumerados que pueden observarse en la Fig. 30.3 previamente presentada:

1. **Pestaña de actividades de la barra de navegación:** Al hacer clic sobre ella se dirige a la pantalla de actividades observada.
2. **Pestaña de actividades pendientes:** Al hacer clic sobre ella se dirige a la lista de actividades pendientes.
3. **Pestaña de actividades atrasadas:** Al hacer clic sobre ella se dirige a la lista de actividades atrasadas.
4. **Pestaña de actividades corregidas:** Al hacer clic sobre ella se dirige a la lista de actividades corregidas.
5. **Pestaña de actividades entregadas:** Al hacer clic sobre ella se dirige a la lista de actividades entregadas.
6. **Pestaña de actividades vencidas:** Al hacer clic sobre ella se dirige a la lista de actividades vencidas.

7. **Títulos de actividades:** Columna del título de las actividades asociadas al estudiante autenticado.
8. **Cursos de las actividades:** Columna del título del curso asociado a la actividad representada.
9. **Fecha de comienzo de las actividades:** Columna de las fechas de comienzo de las actividades asociadas al estudiante autenticado.
10. **Fecha de cierre:** Columna de las fechas de fin de las actividades asociadas al estudiante autenticado.
11. **Botón de resolución de actividad:** Botón que redirige a la pantalla de resolución de la actividad seleccionada.

Realización de entregas de actividades

Los usuarios estudiantes pueden realizar entregas de actividades accediendo a través de la vista de actividades mencionada anteriormente, en la sección de “Actividades pendientes”, “Actividades atrasadas” y “Actividades entregadas”(Ya que el estudiante puede generar más de una entrega de la misma actividad). La vista de entrega de actividades se observa en la Fig. 31. En ella se visualizan tres pestañas entre las que se puede navegar para realizar la actividad.

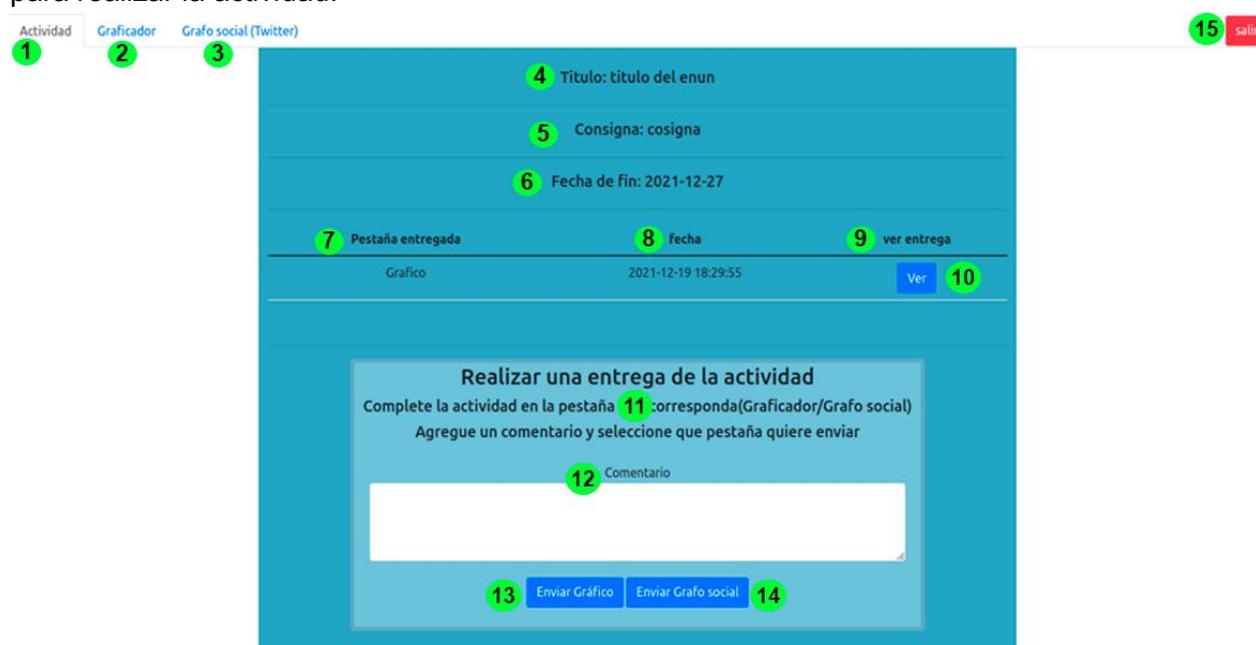


Fig. 31: Pantalla de resolución de una actividad en CeoDatum

Los elementos enumerados en la Fig. 31 se detallan a continuación:

1. **Botón de pestaña de “Actividad”:** Al presionar este botón se despliega la sección de actividad en la que se muestra la información de la actividad, las entregas que ha realizado el estudiante y una sección para realizar nuevas entregas de la actividad ya sea un gráfico o un grafo social.
2. **Botón de pestaña de “Graficador”:** Cuando el estudiante presiona en este botón se muestra el graficador (el mismo que se puede observar en la Fig. 25.1) para que el estudiante pueda resolver la actividad propuesta por el docente.

3. **Botón de pestaña de “Grafo social (Twitter)”**: Al presionar este botón se muestra la página del contador de recurrencias de palabras de Twitter (esto se puede ver en la Fig. 28.1) en la que el estudiante puede resolver la actividad propuesta por el docente.
4. **Sección de “Título”**: En esta sección se muestra el título de la consigna dada por el docente para la actividad.
5. **Sección de “Consigna”**: En esta área se muestra la consigna de la actividad propuesta por el docente, esta es muy importante ya que en él se muestra como debe realizarse la actividad.
6. **Sección de “Fecha de fin”**: En esta sección se muestra hasta cuando tiene tiempo el estudiante para entregar la actividad. Ya que luego pasará a estar vencida y no podrá entregarla.
7. **Columna de “Pestaña entregada”**: En esta columna se muestra cual es la pestaña que entregó el estudiante al realizar la entrega de la actividad.
8. **Columna de “Fecha”**: En esta columna de la tabla se muestra cuál fue la fecha y hora en la que se realizó dicha entrega de la actividad.
9. **Columna de “Ver entrega”**: En esta sección de la tabla se muestra un botón para ver la entrega realizada por el estudiante.
10. **Botón de “Ver”**: Al presionar este botón se muestra una “Ventana Modal” por encima de la pantalla en el que se puede ver cuál es la entrega realizada por el estudiante. En esta ventana se muestra el gráfico con las columnas elegidas y las condiciones agregadas o la información elegida para realizar el grafo social. Como así también el comentario que se realizó al momento de la entrega
11. **Sección de “realizar una entrega”**: En esta sección el estudiante puede realizar un comentario y dependiendo del botón que elija decidirá qué pestaña entregar.
12. **Campo de “comentario”**: En este campo el estudiante debe escribir el comentario que quiere adjuntar con la pestaña que entregará. Este comentario es importante para que el estudiante pueda escribir una justificación porque ha realizado de esa forma la entrega.
13. **Botón de “Enviar gráfico”**: Cuando el estudiante presiona este botón, si es que realizó un gráfico en la pestaña de “Graficador”. Se realizará una entrega de la actividad que estaba resolviendo el estudiante junto con el comentario que haya escrito en el campo de comentario.
14. **Botón de “Enviar grafo social”**: Al presionar este botón, si es que el estudiante realizó una búsqueda en la pestaña de “grafo social (Twitter)”, el usuario realizará una entrega de la actividad que estaba resolviendo junto con el comentario que haya escrito en el campo de comentario.

Corrección de actividades

Para acceder a la pantalla de corrección de una actividad, un profesor debe inspeccionar el estado de los estudiantes sobre una actividad particular. Esto se realiza a través de la vista de actividades mencionada en la sección anterior, cliqueando sobre la entrega deseada. La vista de corrección de actividades es un formulario como el de la Fig.

32. En ésta se observa las entregas realizadas por el estudiante como también una sección para realizar la corrección de la actividad.



Fig. 32: Pantalla de corrección de entrega de actividad de un estudiante en CeoDatum

A continuación se detalla el contenido de la página de corrección de una actividad particular. Para esto se listan los elementos enumerados en la Fig. 32:

1. **Sección de “Nombre”**: En esta sección se muestra el nombre del estudiante que realizó las entregas que se muestran y también a quien se le realizará la corrección.
2. **Sección de entregas**: En este área se muestran las entregas de la actividad realizadas por el estudiante.
3. **Columna de “Pestaña entregada”**: Explicada en realización de entregas de actividades.
4. **Columna de “Fecha”**: Explicada en realización de entregas de actividades.
5. **columna de “Ver entrega”**: Explicada en realización de entregas de actividades.
6. **Botón de ver**: Explicada en realización de entregas de actividades.
7. **Campo de “Nota”**: En este campo el docente debe ingresar la nota que recibirá el usuario dependiendo de cómo haya realizado la actividad y de si esta corresponde con la consigna dada por el docente.
8. **Campo de “Comentario”**: En este campo el docente puede adjuntar un comentario a la corrección para explicar al estudiante el porqué de la nota que ha conseguido.
9. **Botón de “Enviar corrección”**: Al presionar este botón el docente envía la corrección de la actividad al estudiante y vuelve a la página de la actividad sobre la que está realizando correcciones..
10. **Botón de “Volver”**: Al presionar este botón el docente vuelve a la página de la actividad sobre la que está realizando correcciones.

Perfil

Como se puede observar en la Fig. 33, en la pestaña Perfil de la barra de navegación se puede acceder al perfil del usuario autenticado y, si tiene más de un rol, cambiar el rol activo. En la ventana perfil se listan los datos del usuario. Además se permite adicionar instituciones ligadas al usuario.

Perfil ▾

Ver perfil **1**

Ver página como:

Estudiante **2**

Profesor

Delfina Amado **3**
delfinaamado

NECOCHEA, Buenos Aires, Argentina
delfina.amado@gmail.com

INSTITUTOS RELACIONADOS

ESCUELA PRIMARIA ARGENTINO DANESA ALTA MIRA
NECOCHEA, Buenos Aires

Dirección: 43 / 64 Y 66 2920
Email: epb@danesaaltamira.com.ar;
dpp0751051@abc.gov.ar
Teléfono: (02362) 42-7962 **4**

INSTITUTO PIO XII
NECOCHEA, Buenos Aires

Dirección: 60 3068
Email: direccionpioxxii@hotmail.com
Teléfono: (02362) 42-7850

[Añadir instituto](#) **5**

Provincia

Buenos Aires

Ciudad (*)

NECOCHEA

Establecimiento educativo (*)

INSTITUTO DE ESTUDIOS SUPERIORES RIO QUEQUEN

Añadir

Fig. 33: Pantalla de perfil del usuario autenticado en CeoDatum

A continuación se listan los elementos enumerados de la pantalla de perfil según muestra la Fig. 33:

- 1. Pestaña de ver perfil de la barra de navegación:** Al hacer clic sobre esta pestaña se redirigirá a la pantalla observada.
- 2. Vistas de rol posibles:** Muestra los roles del usuario, el rol actual y permite el intercambio de rol.
- 3. Datos del usuario:** Datos relacionados con el usuario autenticado.
- 4. Institutos relacionados:** Datos de los institutos relacionados con el usuario autenticado.

5. **Adición de nuevo instituto relacionado:** Al hacer clic sobre “Añadir instituto” se despliega la selección de provincias.
6. **Formulario de adición de instituto:** A través de este formulario se añaden institutos relacionados al usuario actual.

Configuración

Para acceder a la configuración del sistema el usuario debe ser admin o superadmin. Lo hará a través de la pestaña “Configuración” en la barra de navegación. En la pantalla de configuración observada en la Fig. 34 se pueden realizar dos tareas.

- Carga de dataset de instituciones nacionales: a partir de este dataset se extraen los datos de provincias, ciudades e instituciones nacionales para el registro de usuarios y la adición de instituciones en las que trabajan los profesores.
- Cambio de roles: se muestra una lista de usuarios a los que se les puede modificar los roles asignados. Como mencionamos anteriormente, se permite la aplicación de más de un rol.

The screenshot shows two main sections: 'Dataset de institutos' and 'Usuario del sistema'. The 'Dataset de institutos' section includes a text input for the current file name ('ARCHIVO ACTUAL: Dataset_colegios.csv'), a file selection button ('Browse...'), and a 'Cargar' button. The 'Usuario del sistema' section features a search bar, a role filter (radio buttons for 'Todo', 'Admin', 'Profesores', 'Estudiantes'), and a table of users with columns for 'Nombre', 'Email', and 'Rol'. The 'Rol' column contains checkboxes for 'Admin', 'Profesor', and 'Estudiante'. A pagination control at the bottom shows 'Anterior', '1', '2', '3', and 'Siguiete'.

Fig. 34: Pantalla de configuración del sistema en CeoDatum vista como superadmin.

A continuación se listan los elementos enumerados de la pantalla de configuración del sistema, según se muestra en la Figura X previamente presentada.

1. **Dataset de donde se toman los datos de los institutos educativos nacionales:** Nombre del archivo actual donde se toman los datos de los institutos educativos que se permite elegir en el prototipo de visualizador.
2. **Input del archivo de institutos educacionales:** Input que permite la carga del nuevo archivo de institutos.
3. **Botón de carga del archivo:** Al hacer clic se envía el archivo y se realiza la actualización de institutos educacionales.
4. **Buscador de usuarios registrados:** Barra de búsqueda de usuarios registrados en el sistema.
5. **Filtro de roles de usuarios:** Selección de roles de usuarios que se desea mostrar en la lista.
6. **Nombres de usuarios:** Columna de nombre de usuarios registrados en CeoDatum
7. **Email de usuarios:** Columna de email de usuarios registrados en CeoDatum
8. **Roles de usuarios:** Conjunto de checkbox con roles asignados a los usuarios registrados en el sistema. Permite asignar nuevos roles.
9. **Páginas:** Navegación a través de páginas de los usuarios registrados.

Capítulo 7: Caso de estudio y análisis de usabilidad

En este capítulo se describe el caso de estudio elegido para probar la usabilidad y el funcionamiento del prototipo de visualizador web. En primer lugar, se especifica una problemática a resolver mediante la utilización del prototipo, luego se propone como se resolvería la problemática utilizando el prototipo, y por último se finalizará con el detalle de las pruebas de usuario y sus resultados.

Problemática a resolver

Como problemática a resolver mediante la utilización del prototipo de visualizador web construido, a partir de la arquitectura y su implementación propuestas, se presenta la necesidad por parte de los docentes de mostrar la importancia de aprender a comprender, analizar y generar gráficos. Éstos ayudan a estudiar de una manera alternativa las problemáticas del mundo real, debido al gran volumen de datos al que se tiene acceso hoy en día. Para esto, el docente, propone particularmente mostrar el poder de la utilización de CeoDatum ya que con solo conseguir los datasets en un archivo que se encuentre en uno de los formatos aceptados puede cargar el archivo al visualizador web y generar cualquiera de los gráficos que éste brinda. Otra característica que posee el prototipo es la de permitir ciertas consultas sobre los tweets de la plataforma Twitter.

Por otro lado el docente con el objetivo de conocer el desempeño actual de sus estudiantes respecto a cómo generar gráficos para que éstos brinden información adecuada sobre lo que se está buscando en el dataset, enviará a sus estudiantes una actividad en la cual éstos deban generar un gráfico para resolver la consigna propuesta por el docente.

Con el fin de proponer una problemática que sea capaz de abarcar a docentes de la mayor cantidad de áreas curriculares posibles se propone analizar el dataset de determinaciones de Covid-19 de la Nación Argentina que abarca el período desde el primero de diciembre de 2022 hasta el 26 de diciembre de 2022

Resolución de la problemática

A continuación se presenta como el docente resolvería la importancia de aprender a comprender, analizar y generar gráficos a los estudiantes mediante el prototipo de visualizador web. Como también evaluar el desempeño actual de sus estudiantes respecto a cómo generar gráficos para que éstos, brinden información adecuada sobre lo que se está buscando en el dataset, mediante una actividad creada en el prototipo la cual los estudiantes deberán resolver:

Para utilizar el prototipo de visualizador web, el docente deberá ingresar a la página web de CeoDatum, el prototipo inicia en la pantalla de login y en ella se debe presionar el botón de registrarse como puede verse en la Fig. 35.1.

Login de usuario

Ingrese usuario y contraseña para acceder.

Nombre de usuario

Contraseña

[¿Olvidaste tu contraseña? haz click aquí](#)

Fig. 35.1: Pantalla de login donde se destaca el botón de registro.

Al presionar el botón “Registrarse” (destacado en la Fig. 35.1), el docente pasará a la pantalla del formulario de registro como puede verse en la Fig. 35.2 en la cual deberá ingresar todos los datos solicitados por los campos que aparecen en la pantalla y luego presionar el botón de “Registrarse”.

Formulario de registro

Correo electrónico (*)

Nombre de usuario (*)

Contraseña (*) Contraseña válida!

Repetir contraseña (*)

Nombre (*)

Apellido (*)

Fecha de nacimiento (*)

Provincia (*)

Ciudad (*)

Establecimiento educativo (*)

(*) Campos obligatorios

Fig. 35.2: Pantalla de creación de usuario con un ejemplo incorporado.

Una vez que el docente presiona el botón de “Registrarse”, el mismo pasa a la pantalla de login como se ve en la Fig. 35.3 en la cual deberá ingresar los datos con los que se haya registrado y presionar en el botón de “Entrar” para acceder a CeoDatum y poder usar el prototipo de visualizador web.

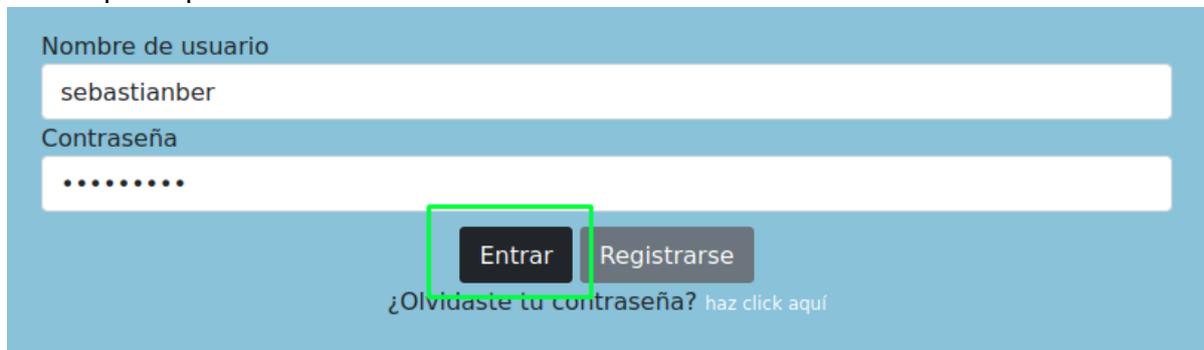


Fig. 35.3: Login de usuario completado con el botón de login destacado.

Ya que el docente desea acceder al graficador con los datos del dataset público de determinaciones de covid 19. El mismo accederá a la sección de dataset públicos presionando en el botón de dataset el cual es un dropdown que al presionarlo despliega opciones y seleccionar el botón que dice “Datasets Públicos” como se puede observar en la Fig. 35.4.

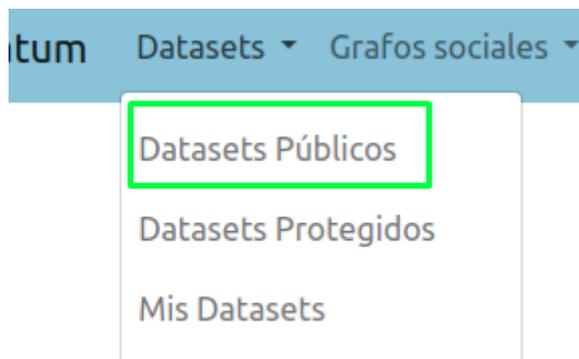


Fig. 35.4: Dropdown de datasets con la pestaña de Datasets Públicos destacada.

Una vez que el docente se encuentra en esta pantalla, deberá escribir en el buscador el dataset que quiere, en este caso escribirá “covid” y presionará en el botón “Buscar” como puede verse en la Fig. 35.5 para que se listen los datasets que contengan como nombre dicha palabra.

Datasets Públicos

Buscador

Nombre del dataset	Subido por	Acciones
starbucksprueba	sebastian bertora	<input type="button" value="Ver"/> <input type="button" value="Graficar"/>

Fig. 35.5: Vista de Datasets Públicos con el buscador destacado.

Al listarse los dataset públicos que contengan la palabra “covid” el docente presionará en el botón de “Graficar” en la columna de “Acciones” de la fila en la que se muestra el dataset que quiere utilizar como puede verse en la Fig. 35.6

Nombre del dataset	Subido por	Acciones
Tests de covid	sebastian bertora	Ver Graficar
Tests_de_covid	sebastian bertora	Ver Graficar

Fig. 35.6: Pantalla de Datasets Públicos con el botón de “Graficar” destacado

Ya en la pantalla del graficador el docente generará un gráfico de barras por la fecha de las personas que hayan tenido resultados positivos con una fecha mayor al 15 de diciembre de 2021. Y esto se realizará como se ven en pantalla, y luego deberá presionar en el botón de “Generar gráfico” como se observa en la Fig. 35.7.

The screenshot shows a user interface for creating a visualization. At the top left, there is a blue header with the text "_covid". To the right of the header is a toolbar containing several icons for different chart types: a bar chart, a line chart, a pie chart, a scatter plot, and a world map. A blue button labeled "Generar gráfico" is highlighted with a green border in the toolbar. Below the header, on the left side, there is a section titled "Eje x" with a text input field containing the word "Fecha". Below that is a section titled "Condiciones" which contains two filter conditions: "Resultado = 'positivo'" and "Fecha >= '2021-12-15'". Each condition has a red trash icon to its right. At the bottom left of the "Condiciones" section is a blue button labeled "Agregar Condición". The main area of the interface is a large, empty white space intended for the generated chart.

Fig. 35.7: Graficador del prototipo de visualizador con el botón de “Generar gráfico” destacado

Una vez generado el gráfico, el docente procederá a analizar el gráfico generado, el cual puede observarse en la Figura 35.8.

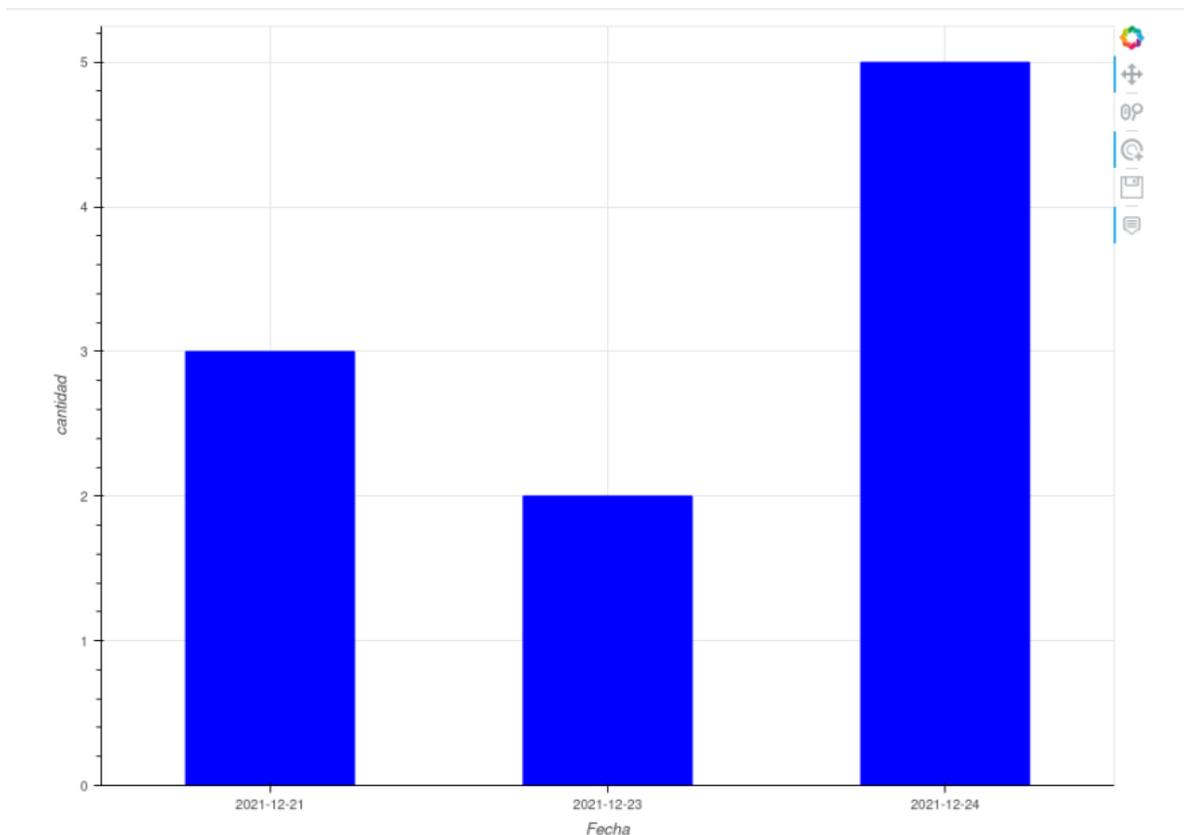


Fig. 35.8: Gráfico que debería generar el usuario.

Luego de analizar el gráfico observado y ver un aumento en las determinaciones positivas de covid el docente procede a ver cómo está impactando en la actualidad los casos de covid en la sociedad y para ello ingresa a la sección de recurrencias de palabras en tweets para ver que está escribiendo la gente en sus tweets acerca de lo observado, al buscador de recurrencias de palabras puede accederse mediante el dropdown de “Grafos sociales” y presionando en el botón de “Twitter (ocurrencias)” como puede verse en la Fig. 35.9.

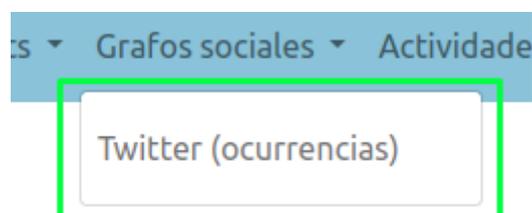


Fig. 35.9: Pestaña de Grafos Sociales con el de Twitter destacado.

Ya en la pantalla de buscador de recurrencias de palabras en tweets el docente escribirá en el campo de qué palabra o frase buscar, la palabra “covid” y activará todos los filtros como puede observarse en la Fig. 35.10 para que aparezca la información más relevante posible sobre lo que escribe la gente en sus tweets sobre el covid y luego presionar el botón de “Buscar”.

Palabra o frase que quiere buscar: covid

Cantidad de palabras a mostrar: 10

Excluir:

- Preposiciones
- Artículos
- Pronombres
- Conjunciones
- Adverbios
- Conjugaciones de ser, estar, haber y tener
- Links

Buscar

Fig. 35.10: Pantalla del buscador de ocurrencias con el ejemplo "Covid" y el botón "Buscar" destacado.

Una vez generado el gráfico, el docente procederá a analizar el grafo y la nube de palabras generadas, como se puede ver en la Fig. 35.11.

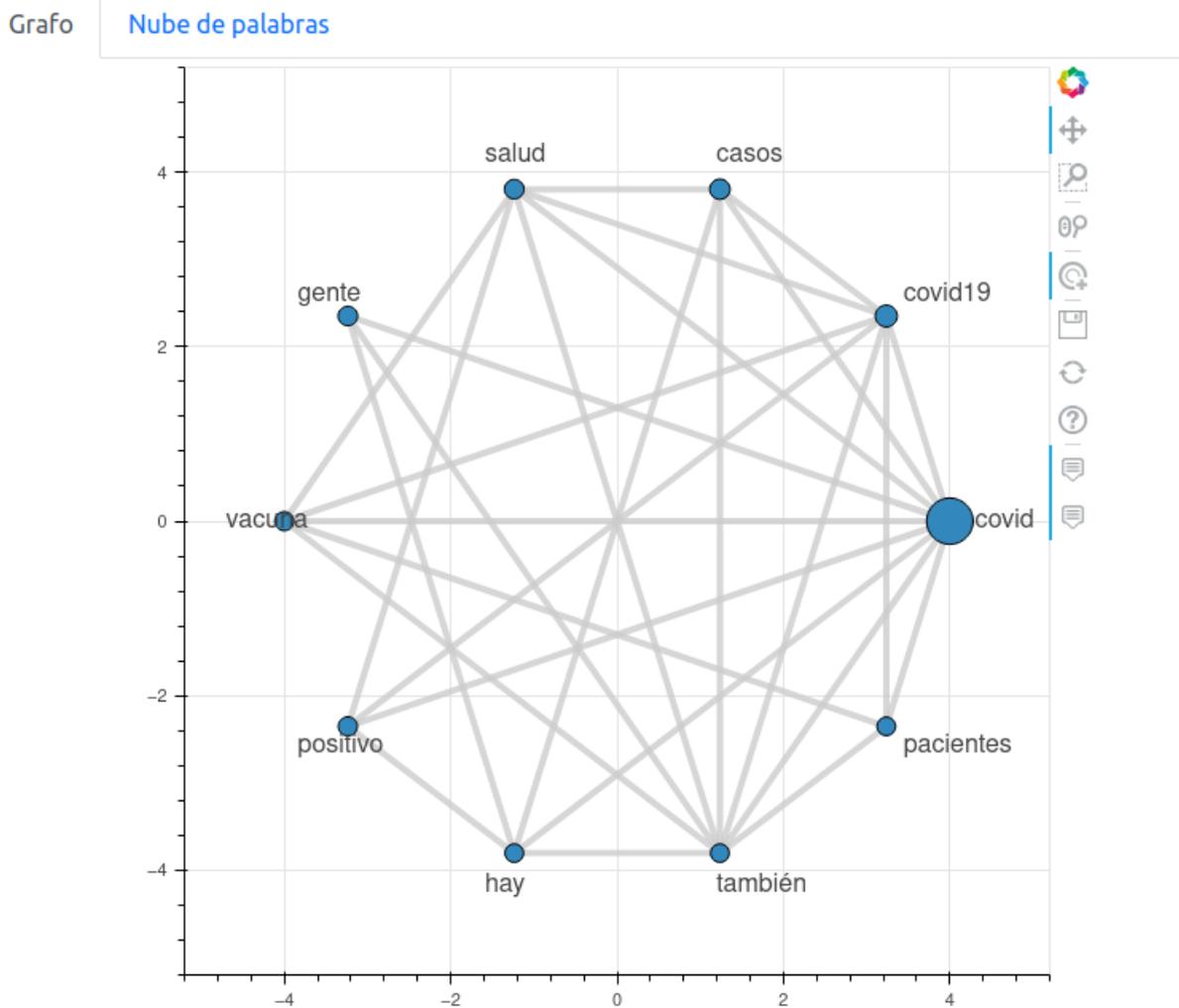


Fig. 35.11: Grafo que debería generar el usuario.

Luego de realizar estos análisis. El docente procederá a crear un curso para poder generar actividades a los estudiantes. Para esto el docente ingresa a la sección de cursos presionando en el botón de "Cursos" el cual se encuentra en la barra de navegación, como se observa en la Fig. 35.12.

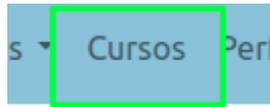


Fig. 35.12: Pestaña de cursos destacada

Ya en la pantalla de cursos el docente presionará el botón de “Crear nuevo curso” el cual puede verse en la Fig. 35.13.

Mis cursos

Nombre	Fecha de comienzo	Fecha de fin
CursoMuestra	2021-12-24	2021-12-28
Matematica 2022	2022-01-28	2022-12-15

[Crear nuevo curso](#)

Fig. 35.13: Pantalla de “Mis Cursos” con el botón “Crear nuevo curso” destacado

Al presionar dicho botón, el docente pasará a la pantalla del formulario para crear un curso en la cual deberá completar todos los campos que se le solicitan y luego presionar en el botón de “Crear curso”, esto puede observarse en la Fig. 35.14 y completar la creación del curso

Nuevo curso

Nombre

Fecha de inicio

Fecha de fin

Establecimiento educativo

Ciclo

Ámbito curricular

[Crear curso](#)

Fig. 35.14: Pantalla de creación de nuevo curso.

Una vez completada la creación del curso el docente pasará a la pantalla de cursos en la que aparecerá listado el curso que creó como puede verse en la Fig. 35.15 y al cual podrá acceder presionando en cualquier lugar de la fila en la que aparece el curso. El docente presionara en la fila para acceder al curso.

Mis cursos

Nombre	Fecha de comienzo	Fecha de fin
CursoMuestra	2021-12-24	2021-12-28
Matematica 2022	2022-01-28	2022-12-15

[Crear nuevo curso](#)

Fig. 35.15: Pantalla de cursos con el curso creado correctamente.

Luego de acceder al curso el docente procederá a agregar a los estudiantes que se hayan registrado previamente en CeoDatum. Para esto, el docente ampliará la sección de “estudiantes inscriptos” y escribirá el nombre de cada estudiante y presionará el botón de “añadir” para invitar a cada estudiante al curso, esto puede observarse en la Fig. 35.16.

Matemática 2022

▼ Estudiantes inscriptos

Nombre	Email	Nombre de Estudiante
bertora, martin	martin@martin	martin
bertora, ivan	ivan@gmail.com	ivan
kesselman, debora	debora@gmail.com	debora
melchiori, candela	candela@gmail.com	candela
Spaltro, Franco	fran@gmail.com	fran

Añadir estudiante al curso:

-- Seleccione un estudiante de la lista --
▼

Añadir

Fig. 35.16: Área de alumnos inscriptos del curso de Matemática 2022.

Por otro lado el docente agregará el dataset de covid al curso para poder agregarlo en las actividades, para esto presionará el botón de “Agregar dataset” el cual puede verse en la Fig. 35.17.

▼ Datasets

Nombre del dataset	Subido por	Acciones
Agregar dataset		

Fig. 35.17: Botón de “Agregar Dataset” en el curso Matemática 2022 en el área de Datasets.

Al presionar dicho botón se abrirá el modal de Agregar dataset, como puede observarse en la Fig. 35.18, en el cual el docente elegirá uno de los datasets que tiene acceso el docente y presionará en el botón de “Agregar” para agregarlo.

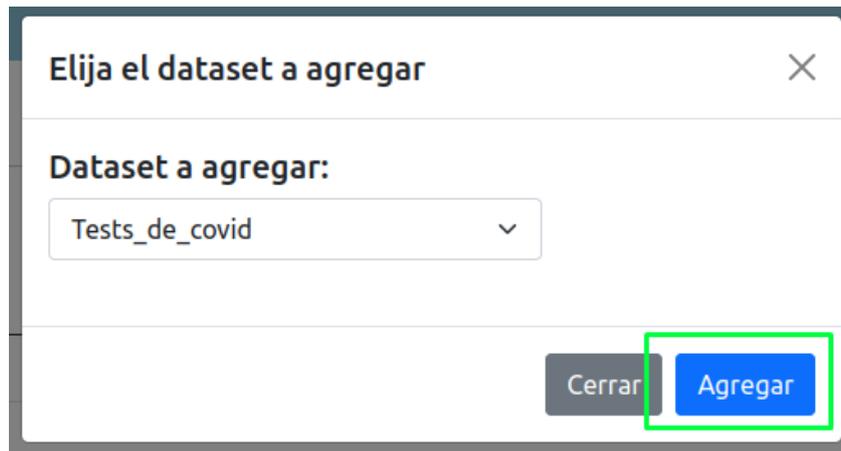


Fig. 35.18: Ventana de agregar dataset a un curso con el botón “Agregar” destacado.

Una vez agregados los estudiantes y asignado el dataset al curso el docente procederá a generar la actividad para esto presiona en el botón de “Nueva actividad” el cual puede verse en la Fig. 35.19.

▼ Actividades

Titulo	Descripción	Estado
Nueva actividad		

▼ Datasets

Fig. 35.19: Botón de creación de nueva actividad en un curso en el área “Actividades”

Al presionar dicho botón el docente procederá a la pantalla del formulario para crear la actividad en la cual deberá completar los campos solicitados entre los cuales debe especificar qué gráficos podrá utilizar el estudiante al momento de resolver la actividad y luego presionar el botón de “Crear actividad” como puede observarse en la Figura 35.20.

Nueva actividad

Título de actividad Quedan 38 caracteres.
Análisis de casos de covid

Enunciado

Título del enunciado Quedan 31 caracteres.
Analice la suba de casos de covid

Consigna Quedan 171 caracteres.
Cree un gráfico que muestre como han aumentado los casos de covid en el ultimo tiempo

Objetivo Quedan 210 caracteres.
mostrar aumento de casos de covid en argentina

Descripción Quedan 210 caracteres.
mostrar aumento de casos de covid en argentina

Dataset
Tests_de_covid

Gráficos permitidos
 tabla de datos barras línea torta punto dispersion mapa
 Grafos sociales

Fecha de inicio
01 / 28 / 2022

Fecha de fin
02 / 15 / 2022

Crear actividad para:
Todos los alumnos

Permitir entrega fuera de término
 Actividad sin calificación

Crear actividad

Fig. 35.20: Creación de la actividad solicitada al usuario con el botón "Crear actividad" destacado.

Luego de crear dicha actividad el docente estará en la pantalla del curso desde la cual el docente accederá a la actividad que ha creado presionando en cualquier lugar de la fila en la que aparece la actividad la cual puede verse en la Fig. 35.21.

Titulo	Descripción	Estado
Análisis de casos de covid	mostrar aumento de casos de covid en argentina	Finaliza el 2022-02-15

Fig. 35.21: Actividad añadida correctamente y mostrada en el área de actividades de un curso.

Ya en la pantalla de actividad el docente podrá ver la información de la actividad generada como así también ver el estado de las entregas de la actividad por parte de los estudiantes y corregir las mismas presionando en el botón de "Ver entrega", como puede verse en la Fig. 35.22

Análisis de casos de covid

mostrar aumento de casos de covid en argentina

Estado de entregasEnunciadoDatos de actividad

Alumno	Estado	Fecha de entrega	Nota	Ver entrega
martin bertora	No calificado	2021-12-27 13:38:56	No calificado	Ver entrega

Fig. 35.22: Vista de la actividad específica con el botón "Ver entrega" destacado.

Al presionar dicho botón el docente estará en la pantalla de ver entrega del estudiante en la cual podrá ver las entregas que realizó el estudiante (los gráficos entregados junto con su comentario) y calificar la actividad y junto con la calificación escribir un comentario sobre la corrección de la entrega realizada por el estudiante los cuales se enviarán cuando el docente presione el botón “Enviar corrección”. Esto puede observarse en la Fig. 35.23.

Actividad de martin bertora

Entregas realizadas por el alumno

Tipo de entrega	Fecha	Ver entrega
Gráfico	2021-12-28 18:05:30	Ver

Enunciado
Analice la suba de casos de covid
Genere un grafico que muestre la suba de casos positivos de covid en Argentina durante el mes de diciembre

Corrección
Nota
8
Comentario
Buen grafico realizado pero podrias haber utilizado un filtro

[Enviar corrección](#)
[Volver](#)

Fig. 35.23: Vista de corrección de la actividad de un alumno con el botón “Enviar corrección” destacado

Pruebas de usuario

A continuación se detalla cómo se realizarán las pruebas de usuario siguiendo la problemática anteriormente planteada. Las mismas serán llevadas adelante con un grupo reducido de personas para dejar como trabajo futuro, una prueba con más usuarios. Se especificará qué población será la que realice las pruebas de usuario, la metodología que se tomará para la realización de las mismas y el posterior análisis de los resultados obtenidos.

Población

Se tomará como población para la realización de las pruebas de usuarios a docentes del ciclo superior de los establecimientos educativos de la nación Argentina.

Metodología

Como metodología de pruebas de usuario se realizarán dos pasadas en las cuales se tomarán tres personas dentro de la población buscada por las pruebas de usuario. A dichas personas se les dará una computadora en la que se tenga acceso a CeoDatum y un documento el cual contendrá una lista de tareas a realizar dentro de la aplicación para

resolver la problemática planteada anteriormente. Por otro lado, se realizó a los mismos usuarios de las pruebas, una encuesta que incluye un SUS (Simple Usability Scale), “Un cuestionario simple de diez preguntas que provee una visión global de aspectos subjetivos de la usabilidad” [Brooke, 1996]. Dicha lista de tareas puede observarse en el Anexo 4 y la encuesta SUS realizada por los docentes puede observarse en el Anexo 5.

Resultados de la primera pasada

En la primera pasada de pruebas de usuario se dio la lista de tareas a 3 docentes para que las completaran utilizando CeoDatum. Los datos de los mismos son:

- Primer docente:
 - Edad: 48 años
 - Asignatura dictada: Química y Físicoquímica
- Segundo docente:
 - Edad: 52 años
 - Asignatura dictada: Matemática
- Tercer docente:
 - Edad: 50 años
 - Asignatura: Biología

Los tres docentes completaron la lista de tareas correctamente y se observó una navegación intuitiva dentro de CeoDatum. Sin embargo, con respecto al feedback que recibe el usuario al realizar ciertas actividades o que es lo que están viendo cuando cambian de página dentro de CeoDatum, se encontraron los siguientes problemas:

1. El primer problema detectado fue que al agregar una condición en el graficador, el Modal del mismo no se cierra. Esto llevó a la confusión de los docentes, ya que no identificaron que la condición se había agregado correctamente.
2. Por otro lado, al agregar un dataset a un curso, dos de los tres docentes dudaron si se había agregado correctamente al curso, ya que al realizar esta acción se recargaba la página y no muestra ningún mensaje de lo que ha ocurrido.
3. A su vez, dos de los tres docentes al crear una actividad no comprendía por qué se le pedían datos que para ellos eran lo mismo en el formulario de crear actividad, esto sucedió con los campos de:
 - Título de la actividad y título del enunciado.
 - Consigna y objetivo.
4. Al ingresar en "Twitter (ocurrencias)" en la sección de grafo social, se notó abrumados a los docentes por no saber lo que estaba sucediendo en pantalla. No comprendieron tanto para que se utilizará dicha página como también cómo utilizarla.
5. Durante el registro de usuario uno de los docentes tuvo problemas al buscar su colegio en la lista de establecimientos educativos que se presentan.

Respecto a las encuestas SUS realizadas a los usuarios de la primera pasada de pruebas de usuario, se obtuvo un puntaje general de 87.5. Lo que implica un nivel Excelente de usabilidad (a partir de los 85) [Bangor et al., 2009]. Los resultados de estas encuestas pueden verse en el Anexo 6.

Análisis de la primer pasada

Debido a los problemas detectados durante la primera pasada de las pruebas de usuario se decidió implementar las siguientes modificaciones a CeoDatum (la numeración utilizada a continuación hace referencia a la utilizada en los resultados de la primera pasada):

1. Debido al problema detectado al agregar condiciones se cambió la interacción. Cuando se presiona el botón de “Agregar” en el modal de “Agregar condición” el mismo se cierra para que el usuario (ya sea docente o estudiante) vea que su interacción agregó dicha condición correctamente.
2. Ya que al agregar un dataset en la página de un curso la misma se refresca, se agregó una alerta (en color verde) en la que se notifica que se agregó correctamente el dataset.
3. Al notar el problema de que los docentes encontraron muy parecidos los campos de “Título de la actividad” y “Título de enunciado” y los campos de “Consigna” y “Objetivo”, se decidió dejar explícito en la página que los campos de “Título del enunciado” y “Consigna” son los campos que verá el alumno.
4. Debido a que al ingresar a "Twitter (ocurrencias)" en la sección de grafo social se percibió a los docentes abrumados al no saber para qué utilizar dicha página y cómo utilizarla, se decidió agregar un modal el cual se abre al cargar la página. El mismo contiene una explicación de qué es lo que se puede hacer en la página y un tutorial de cómo utilizarla.
5. Ya que uno de los docentes tuvo dificultades para encontrar su establecimiento educativo en el formulario de registro (debido a la gran cantidad de establecimientos en la ciudad que eligió), se decidió agregar un campo en el que se pueda escribir para filtrar las opciones que aparecen en el campo de selección de establecimiento educativo.

Resultados de la segunda pasada

Luego de incorporar las mejoras necesarias que fueron detectadas durante la primera pasada, se realiza una segunda pasada de las pruebas de usuario descritas a docentes. A continuación se presentarán los resultados de la misma, comenzando con la descripción de los docentes participantes:

- Primer docente:
 - Edad: 30 años.
 - Asignatura: Educación Agraria.
- Segundo docente:
 - Edad: 55 años.
 - Asignatura: Matemática
- Tercer docente:
 - Edad: 50 años
 - Asignatura: Genética y Biología (Ciencias Naturales).

En esta prueba de usuario los mismos debieron realizar las tareas propuestas en la lista de tareas que se mencionó anteriormente. Durante el desarrollo de las pruebas, al igual

que en la primera pasada, se notó una intuitiva navegación. Pese a esto, se detectó un inconveniente en CeoDatum durante la misma:

1. Al realizar el paso de crear una actividad, uno de los docentes omitió el paso de agregar dataset y comenzó el paso de crear una actividad. Pero para crear una actividad, se debe tener al menos un estudiante en el curso y un dataset agregado al mismo. Es por esto que no pudo completar la creación de la actividad y tuvo que volver a la pantalla del curso para agregar un dataset.

Por otro lado se encontraron situaciones que, aunque no tienen la misma relevancia que el error anteriormente mencionado, se consideran en el informe.

2. Uno de los docentes nunca cerró una alerta que apareció en pantalla cuando él mismo intentó crear un gráfico de mapa con columnas que no contenían latitud y longitud.
3. Uno de los docentes escribió una contraseña más corta que lo requerido y al mostrarle dicho error, el mismo solo corrigió el error en el campo de "contraseña" pero no en el de "repetir contraseña". Esto se debe a que el error solo se marcaba en dicho campo y no en ambos.
4. Al estar completando el formulario de creación de actividad, a uno de los docentes le faltó ingresar uno de los campos y al presionar en el botón de "crear curso" se le notificó que debe ingresar todos los campos pero los mismos se borraron. Se notó la molestia del docente al habersele borrado todo lo que había escrito y no saber qué campo es el que no ingresó.
5. Uno de los docentes olvidó la consigna de la actividad al estar realizando la corrección y tuvo que volver a la pantalla de la actividad para verla.

Como resultado de las encuestas SUS realizadas a los usuarios de la segunda pasada de pruebas de usuario se obtuvo un puntaje general de 90.83. Lo que implica un nivel Excelente de usabilidad (recordamos a partir de los 85). Los resultados de estas encuestas pueden verse en el Anexo 6.

Análisis de la segunda pasada

Luego de realizar la segunda pasada, se procedió a realizar los cambios indicados a continuación para corregir el prototipo y mejorar la experiencia de usuario. Siguiendo la numeración anteriormente indicada, se procede a detallar los cambios incorporados:

1. Se añade una alerta al presionar el botón "Nueva actividad". Si no hay alumnos inscriptos en el curso o no existe un mínimo de un dataset asociado al mismo, no se permite ingresar a la creación de actividad y se le informa al usuario.
2. Se agrega un timeout en el alerta mencionado en este punto. Luego de 10 segundos, desaparece.
3. Si el usuario docente ingresa una contraseña que no cumple el mínimo de caracteres en ambos inputs, se añade una alerta en cada uno de ellos.
4. Luego de realizar las comprobaciones correspondientes, si existe un error en los campos de creación de nueva actividad, al volver a la pantalla de creación de la misma se envían los valores ingresados por el usuario previamente. Éstos se añaden como valores de los campos completados correctamente, ahorrando tiempo y posible molestia en el usuario.

5. Se añade el título y consigna de la actividad en la pantalla de corrección de las entregas de la misma para que el docente no tenga que navegar entre páginas para poder visualizarlos.

Análisis de usabilidad

Las pruebas preliminares con usuarios, basadas en dos pasadas, lograron implementar mejoras inmediatas que al momento de realizar futuras pruebas de usuarios masivas, agilizarán y refinarán los inconvenientes o necesidad de mejoras a ser detectados, con lo que se espera bajar el nivel de frustración de futuros usuarios y así lograr un mejor aprovechamiento de dichas pruebas.

En general se detectó, a partir de la observación de ambas pasadas que dos de los seis docentes participantes de las pruebas omitieron el tutorial propuesto, pero aún así, la interfaz implementada resultó de uso intuitivo.

Uno de los docentes asoció parte del funcionamiento de CeoDatum a la propuesta existente de Google Classroom³⁴ desde la posibilidad de envíos de resoluciones por parte de los estudiantes y la posibilidad de devolución por parte del docente, destacando que en CeoDatum, es posible además que el estudiante suba sucesivas entregas quedando un historial de las mismas para el docente.

Como se trata de una etapa inicial de análisis se define como trabajo futuro la implementación de un análisis exhaustivo de usabilidad para conseguir resultados más precisos y lograr una aplicación de mayor calidad para el usuario.

Aclaración: El capítulo 6 no contiene las implementaciones hechas como resultado de las pruebas de usuario.

³⁴ <https://classroom.google.com/>

Capítulo 8: Análisis de accesibilidad de CeoDatum

En este capítulo se establecen y realizan una serie de pruebas de accesibilidad sobre un conjunto de páginas web consideradas críticas en el prototipo desarrollado. Será realizada luego de aplicar las modificaciones resultantes a partir del caso de estudio y las modificaciones elementales detectadas durante el mismo. El análisis de accesibilidad se realiza a través de la herramienta Wave³⁵ que examina el código HTML de las páginas seleccionadas de CeoDatum. A partir de lo arrojado por dicha herramienta se reportará el análisis de accesibilidad en dos de las funcionalidades del prototipo: la creación de actividades por parte del docente y la resolución de las mismas por parte de los estudiantes.

Accesibilidad en aplicaciones web

La web es una herramienta creada para que todos y todas puedan utilizarla, sin importar el idioma, la ubicación o discapacidad. Ésta característica se define como accesibilidad de la web y es esencial para crear sitios y herramientas web de calidad, sin excluir a grupos de personas en su uso. La W3C³⁶ define la accesibilidad en la Web. Se define cómo accesible si todas las personas pueden percibir, comprender, navegar e interactuar con la Web y contribuir a ella. Abarca todas las discapacidades, desde auditivas, del habla o visuales hasta cognitivas, neurológicas o físicas. También comprende a personas sin discapacidad: personas mayores, personas con “discapacidades temporales” cómo alguien que perdió sus gafas, personas con conexión lenta a Internet o limitadas por su ubicación.

La accesibilidad web es requisito legal en algunas ocasiones. En Argentina se encuentra respaldada por la Ley N° 26.653³⁷. Ésta promueve la accesibilidad de la información para facilitar y asegurar el acceso a todas las personas con discapacidad, generando igualdad real de oportunidades y trato y evitando así todo tipo de discriminación.

Accesibilidad en CeoDatum

Se realizará una prueba de accesibilidad reducida del Prototipo de Visualizador Web desarrollado, sobre dos de las páginas web del mismo consideradas críticas: creación de actividades y resolución de las mismas. La prueba se realizará a través de la herramienta Wave y se establecerán los resultados arrojados para en un trabajo futuro realizar las modificaciones necesarias para garantizar la accesibilidad del sitio

³⁵ <https://wave.webaim.org/>

³⁶ <https://www.w3.org/WAI/fundamentals/accessibility-intro/es#context>

³⁷ <http://servicios.infoleg.gob.ar/infolegInternet/anexos/175000-179999/175694/norma.htm>

Accesibilidad en la creación de actividades

En la Fig. 36.1 se muestra el resultado arrojado por la herramienta de análisis de accesibilidad Web Wave sobre la página de creación de actividades de CeoDatum.

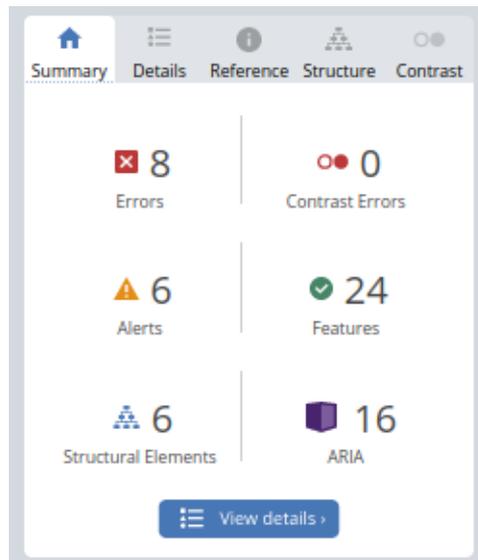


Fig. 36.1: Resumen del resultado arrojado en Wave sobre la página de creación de actividades

En la Fig. 36.1, se puede observar que la ejecución de la herramienta indica ocho errores y seis advertencias en el examen de accesibilidad. Siete de los errores arrojados son sobre labels: pérdida, duplicaciones o vacíos. El otro error se trata de la falta de especificación del lenguaje de la página Web. Por otro lado, las advertencias se tratan de un label de un select perdido, un label que tiene asociado un input inexistente, la falta de agrupación de checkboxes en un fieldset y que la página no tiene un heading de primer nivel. En la Fig. 36.2 se puede observar cada error arrojado por la herramienta en la página de creación de actividades.

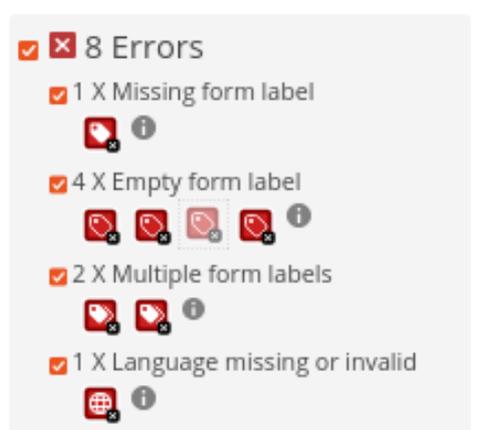


Fig. 36.2: Errores arrojados en Wave sobre la página de creación de actividades

A continuación se procede a explicar en detalle los errores observados en la Fig. 36.2:

1. **Missing form label:** El input no tiene el label correspondiente. Cuando esto pasa, es posible que la función o el propósito de este input no se presente a los usuarios de lectores de pantalla. Los labels también proporcionan descripciones visibles y objetivos más grandes sobre los que se puede clicar para el input.
2. **Empty form label:** El label se encuentra presente, pero no tiene contenido. Por lo tanto no presentará ninguna información sobre el input del formulario al usuario.
3. **Multiple form labels:** Un input tiene más de un label asociado. Un input debe tener como máximo un label asociado. Si hay más, es posible que la tecnología de asistencia no lea la etiqueta adecuada.
4. **Language missing or invalid:** El idioma del documento no se encuentra identificado o el valor del atributo lang no es válido. Es importante definir el idioma de la página porque permite a los lectores de pantalla leer el contenido en el idioma adecuado y facilita la traducción automática de contenidos.

Accesibilidad en la resolución de actividades

En la Fig. 37 se muestra el resultado arrojado por la herramienta de análisis de accesibilidad Web Wave sobre la página de resolución de actividades de CeoDatum por parte de un estudiante.

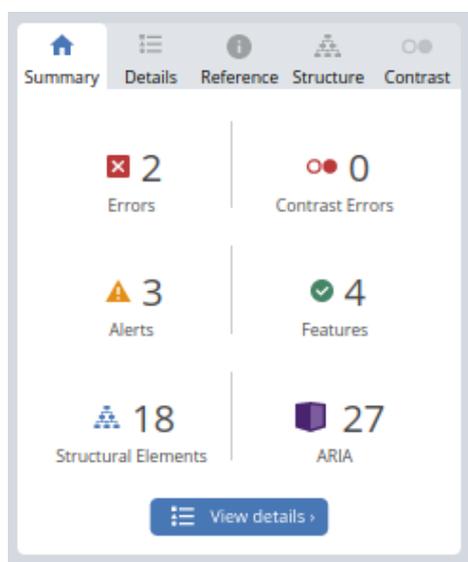


Fig. 37.1: Resumen del resultado arrojado en Wave sobre la página de resolución de actividades

En este caso, como se visualiza en la Fig. 37.1, la ejecución de la herramienta Wave, informa dos errores: por un lado al igual que en la prueba anterior la página Web no especifica el idioma empleado, por el otro la página no tiene un título asociado. Por su parte, las alertas son por la falta de asociación de radio buttons en un fieldset, la falta de un heading de primer nivel y la omisión de un nivel de heading. En la Figura 37.2 se puede observar en detalle los errores arrojados.

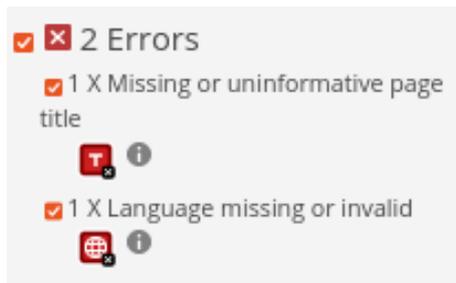


Fig. 37.2: Errores arrojados en Wave sobre la página de resolución de actividades

Se procede a explicar el error arrojado en la herramienta observado en la Fig. 37.2, que no fue explicado anteriormente en los errores de la Fig. 36.2:

- **Missing or uninformative page title:** El título de la página no se encuentra o no es descriptivo. Un título descriptivo ayuda a los usuarios a comprender el propósito o el contenido de una página. Sin un título adecuado, muchos usuarios, sobre todo aquellos que utilizan tecnologías de asistencia, pueden tener dificultades para orientarse en la página.

Discusión del análisis de accesibilidad

Se puede observar por los resultados arrojados que se deben corregir ciertos aspectos de las páginas Web que conforman el Prototipo de Visualizador Web para garantizar la accesibilidad de todas las personas. La mayoría de los errores o advertencias arrojadas tienen una solución considerablemente sencilla, cómo agregar un nuevo elemento HTML o modificar alguno de ellos. Consideramos que asegurar la accesibilidad a todas las personas es de suma importancia para crear una herramienta inclusiva y no excluyente. A futuro se planifica realizar un estudio más exhaustivo de la accesibilidad y abordar los resultados obtenidos.

Capítulo 9: Conclusiones y Trabajos Futuros

En este capítulo se presentan las conclusiones y los trabajos futuros sobre el prototipo web de visualización.

Conclusiones

En este trabajo, se investigó el estado del arte de distintas herramientas de visualización que existen en el mercado actualmente. Se analizó cómo es la utilización de cada una de ellas como así también que costo (económico) tiene su uso. Debido a esto se concluyó que, entre las herramientas investigadas, no existe una herramienta de visualización gratuita destinada a usuarios finales y que además incorporara el concepto de cursos, actividades y corrección para estudiantes y docentes. Por lo que debido a esto se decidió construir un prototipo de visualizador web capaz de cumplir con estas características. Gracias al estudio realizado se decidió que en la construcción de este prototipo se utilizaría Bokeh como herramienta para construir las visualizaciones

Por otro lado se analizó los distintos tipos de base de datos para definir cuál era el mejor para utilizar en la aplicación como así también que framework de Python utilizar ya que Bokeh es una librería de Python.

Según el objetivo, se definió una arquitectura cliente-servidor en la que se utilizó una implementación del patrón MVC (modelo, vista, controlador). Para el cual se presentó una posible implementación de la misma especificando cómo interactúan los componentes que la componen entre sí para lograr el funcionamiento del prototipo de visualizador web, detallando también cada componente por separado.

Como conclusión de la tesina se puede observar el prototipo de visualizador web construido en el que se pueden realizar las siguientes funciones principales:

- Con la ayuda de Pyspark se definió la forma en la que se cargan los dataset a la aplicación a partir de archivos JSON y CSV.
- A partir de la librería Bokeh se definió que el usuario pueda construir gráficos sólo eligiendo el tipo de gráfico (barras, línea, circular, dispersión, mapa o tabla) y por qué variable del dataset quiere graficar.
- Debido a la utilización de la API de Twitter combinado con Bokeh. Se logró analizar las palabras más recurrentes de los últimos 100 tweets que tengan relación con lo que haya buscado el usuario y a partir de estas palabras conseguidas se construye un grafo y una nube de palabras que muestran la información conseguida.
- Por otro lado se construyó una plataforma en la que profesores puedan crear cursos y actividades para que los estudiantes las resuelvan con el graficador, a partir de un dataset previamente cargado o con el contador de recurrencias de Twitter.

Se espera que este prototipo de visualizador facilite la enseñanza en los establecimientos educativos como así también concientice a los estudiantes de lo poderoso que puede ser construir un gráfico de una forma sencilla para afrontar la gran cantidad de información que se puede obtener hoy en día.

Trabajos Futuros

Definidas las conclusiones y el trabajo propuesto en esta tesina, se incluyen como posible trabajo futuro una serie de nuevas funciones o características para añadir al prototipo de visualizador web. A continuación se presenta un listado de las mismas:

- Permitir la carga de nuevos formatos de archivos a cargar. Actualmente, sólo permite la carga de archivos en formato CSV y JSON, por lo que puede considerarse restrictiva en este sentido. Se podrían considerar otros formatos como TAB, XLS, XML e incluso TXT o también archivos con datos geoespaciales en sus formatos SHP, SHX, TIF, etc.
- Actualmente la aplicación no hace uso de versionado de archivos. Con esto, se hace referencia a que si se desea subir una nueva versión de cualquier archivo que fue cargado, éste será reemplazado o considerado como otro diferente. Principalmente consideramos importante esta característica para los archivos que mantienen los datos propios del prototipo y no aquellos que se usan para realizar las visualizaciones. Esto se debe a que, por ejemplo, si el archivo de instituciones estudiantiles es reemplazado por una nueva versión podría generar inconsistencias en los datos que tomaban información del anterior.
- Permitir la elección de más de un dataset para una actividad. Actualmente, al crear una tarea, ésta considera un único dataset para realizar las visualizaciones. Se realizó de esta forma para poder presentar una idea a través del prototipo, pero consideramos que posibilitar el uso de más de un dataset enriquecería la realización de actividades.
- Añadir mayor especificación en la creación de un curso. Cómo el principal objetivo del presente informe no era en sí la creación de cursos, al momento de ser creados se requiere información mínima del mismo. Se podría agregar datos referidos a la materia, temas tratados, pautas de aprobación, aplicación de más de un área curricular y otros.
- Cómo trabajo futuro más ambicioso se propone realizar una persistencia inteligente de datos. Con esto se hace referencia a que los datasets que son cargados en CeoDatum, a través de metadatos, sean almacenados según convenga y sea más eficiente en distintos motores de bases de datos. Esto se haría dependiendo de lo analizado y comparado sobre cada motor durante la presente tesina.
- Realizar pruebas de usuario a mayor escala: Si bien las pruebas de dos pasadas permitieron abordar las necesidades más inmediatas, es necesario aún realizar pruebas con una mayor cantidad de docentes.
- Problemas de accesibilidad detectados en el análisis: se espera abordar la accesibilidad en una futura versión del prototipo así como continuar con las pruebas en la totalidad de los formularios.
- Diseño de la interfaz: Al tratarse de un prototipo no se hizo foco en el diseño gráfico de los formularios, sino que el foco está en la funcionalidad. A futuro, se espera trabajar con mejoras en este aspecto.
- Mayor variedad de actividades: Sumar la posibilidad de nuevas actividades que permitan trabajar con otros tipos de grafos
- Login Integrado: se espera trabajar con la posibilidad de login integrado con plataformas como Google, Facebook e Instagram entre otras.

- Sumarse a un curso o actividad como estudiante: implementar nuevas estrategias para sumarse a un curso, por ejemplo, mediante un código QR generado al momento de crear un curso o una actividad.
- CeoDatum como Plataforma: al evolucionar las versiones del prototipo se espera proponer un producto disponible masivamente, y posteriormente generar una API que permita la integración de CeoDatum con otras plataformas educativas ya existentes.
- Comunidad: sumar la posibilidad de trabajar en comunidad para resolver problemas y proponer nuevos desafíos.
- Foro: sumar un foro para trabajar y generar preguntas frecuentes
- Colaboración: Permitir la creación colaborativa de actividades

De este modo se concluye este capítulo, esperando ser fuente de inspiración para futuras tesinas.

Referencias

- [Abuín et al., 2020] Abuín, J. M., Lopes, N., Ferreira, L., Pena, T. F., & Schmidt, B. (2020). Big data in metagenomics: Apache spark vs MPI. Plos one, 15(10), e0239741.
- [Andre, 2020]. Andre Louie. 2020. Datawrapper Review. Finances on-line Reviews for Business. <https://reviews.financesonline.com/p/datawrapper/>
- [Balaji, 2021] Balaji, N (2021). Data Visualization in Splunk and Tableau: A Case Study Demonstration
- [Bangor et al., 2009] Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of usability studies, 4(3), 114-123.
- [Bermeo Torres & Simbaña Velastegui, 2020] Bermeo Torres, L. J., & Simbaña Velastegui, E. A. (2020). Creación de una solución de inteligencia de negocios para la generación de reportes con información relevante del proceso de titulación de los estudiantes de la Carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Sede Quito (Bachelor's thesis).
- [Boyd & Ellison, 2007] Boyd, D.M., & Ellison, N.B. (2007). Social network sites: definition, history, and scholarship. Journal of Computer-Mediated Communication
- [Brooke, 1996] Brooke, J. (1996). SUS: a "quick and dirty usability scale". Usability
- [Chen & Lee, 2019] Chen, J. K., & Lee, W. Z. (2019). An Introduction of NoSQL Databases based on their categories and application industries. Algorithms, 12(5), 106.
- [Chen et al., 2008] Chun-houh Chen, Wolfgang Hardle, and Antony Unwin. (2008). Handbook of Data Visualization. Berlin, Heidelberg: Springer Berlin Heidelberg
- [Clark-Wilson et al., 2021] Clark-Wilson, A., Donevska-Todorova, A., Faggiano, E., Trgalova, J., & Weigand, H. G. (Eds.). (2021). Mathematics Education in the Digital Age: Learning, Practice and Theory. Routledge.
- [Devndra, 2020] Devndra Ghimire. (2020). Comparative study on Python web frameworks: Flask and Django. Metropolia University of Applied Sciences
- [Díaz, Osorio & Amadeo, 2019] Díaz, F. J., Osorio, M. A., & Amadeo, A. P. (2019). Tecnologías para el análisis de datos basadas en software libre. Series: Libros de Cátedra.
- [Faronius Hofmann & Håkansson, 2021] Therese Faronius Hofmann and Linda Håkansson. (2021). Visualization Design Effects on Credibility and Data Perception, and the Importance of Digital Interaction. Uppsala universitet

[Giamas, 2019] Giamas, A. (2019). Mastering MongoDB 4. x: Expert techniques to run high-volume and fault-tolerant database solutions using MongoDB 4. x. Packt Publishing Ltd.

[Gupta, 2015] Gupta, Y. (2015). Kibana essentials. Packt Publishing Ltd.

[Ianni et al., 2020] Michele Ianni. Elio Masciari. Giancarlo Sperli. (2020). A survey of Big Data dimensions vs Social Networks analysis. Journal of Intelligent Information Systems.

[Idris et al., 2020] Idris, N., Foozy, C. F. M., & Shamala, P. (2020). A generic review of web technology: Django and flask. International Journal of Advanced Science Computing and Engineering, 2(1), 34-40.

[Joachim, 2018] Joachim Engel. 2018. Exploring civic statistics with codap. Ludwigsburg University of Education, Ludwigsburg, Germany

[Jolly, 2018] Jolly, K. (2018). Hands-on data visualization with Bokeh: Interactive web plotting for Python using Bokeh. Packt Publishing Ltd.

[Joy et al., 2021] Joy G. Bertling, Lynn Hodge & Shande King (2021) The Case for Data Visualization in the Art Classroom, Art Education, 74:2, 44-49, DOI: 10.1080/00043125.2020.1852381

[Kroc et al., 2019] Kroc, K., Kizun, O., & Skublewska-Paszowska, M. (2020). Performance analysis of relational databases MySQL, PostgreSQL, MariaDB and H2. Journal of Computer Sciences Institute, 14, 1-7.

[Lliteras, 2020] Lliteras Alejandra. Propuesta doctoral. Facultad de Informática, UNLP. Resolución HCD N.º 319/20, del 11 de diciembre de 2020.

[Mirzoeff, 1999] Mirzoeff, N (1999). An introduction to visual culture. Psychology Press.

[Moniruzzaman & Hossain, 2013] Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. arXiv preprint arXiv:1307.0191.

[Murray, 2014] Murray, D. (2014, 29th December). What Tableau IS and IS NOT. [Weblog]. Retrieved 12 October 2021, from <https://interworks.com/blog/dmurray/2014/12/29/what-tableau-and-not/>

[Oluwatosin, 2014] Oluwatosin, H. S. (2014). Client-server model. IOSR J Comput. Eng, 16(1), 2278-8727.

[Oz, 2008] Effy Oz. (2008). Administración de los sistemas de información. Quinta edición. The Pennsylvania State University, Great Valley

[Özsu & Valduriez, 2020] Özsu, M. T., & Valduriez, P. (2020). NoSQL, NewSQL, and Polystores. In Principles of Distributed Database Systems (pp. 519-557). Springer, Cham.

[Pavlo & Aslett, 2016] Pavlo, A., & Aslett, M. (2016). What's really new with NewSQL?. *ACM Sigmod Record*, 45(2), 45-55.

[Phillips et al., 2010] Phillips, L. M., Norris, S. P., & Macnab, J. S. (2010). *Visualization in mathematics, reading and science education* (Vol. 5). Springer Science & Business Media.

[Roy-Hubara & Sturm, 2020] Roy-Hubara, N., & Sturm, A. (2020). Design methods for the new database era: a systematic literature review. *Software and Systems Modeling*, 19(2), 297-312.

[Schreiner et al., 2019] Schreiner, G. A., Knob, R., Duarte, D., Vilain, P., & Mello, R. D. S. (2019, December). NewSQL Through the Looking Glass. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services* (pp. 361-369).

[Solarz & Szymczyk, 2020] Solarz, A., & Szymczyk, T. (2020). Oracle 19c, SQL Server 2019, Postgresql 12 and MySQL 8 database systems comparison. *Journal of Computer Sciences Institute*, 17, 373-378.

[Tableau, 2021] What Is Data Visualization? Definition, Examples, And Learning Resources. (n.d.). Tableau. URL: <https://www.tableau.com/learn/articles/data-visualization>. Fecha de último acceso 21 de Noviembre de 2021.

[Thearle, 2019] Thearle, R. W. (2019). A survey of object oriented database systems. In *Object Management* (pp. 23-30). Routledge.

[Unwin, 2020] Unwin, A. (2020). Why is Data Visualization Important? What is Important in Data Visualization?. *Harvard Data Science Review*, 2(1).

[Unwin, 2020] Unwin, A. (2020). Why is Data Visualization Important? What is Important in Data Visualization?. *Harvard Data Science Review*.

[Venkatraman et al., 2016] Venkatraman, S., Fahd, K., Kaspi, S., & Venkatraman, R. (2016). SQL versus NoSQL movement with big data analytics. *Int. J. Inform. Technol. Comput. Sci*, 8, 59-66.

[Vlahu-Gjorgievska & Trajkovik, 2019] Elena Vlahu-Gjorgievska, Vladimir Trajkovik. (2019). *Healthcare data warehouse system supporting cross-border interoperability*.

[Wilke, 2019] Claus O. Wilke. (2019) *Fundamentals of Data Visualization*

[Yu & Muñoz-Justicia, 2020] Yu, J., & Muñoz-Justicia, J. (2020). Free and low-cost twitter research software tools for social science. *Social Science Computer Review*, 0894439320904318.

Anexo 1: Elección de la base de datos SQL

Para la elección del motor de bases de datos SQL se tomó como referencia al ranking de DB-Engines³⁸ en el que se ordena a las distintas bases de datos por popularidad (El cálculo de este ranking se realiza en base a número de menciones en distintos sitios web, interés general en el motor sacado de google trends, relevancia en redes sociales, entre otros). El ranking en marzo de 2021 es el siguiente:

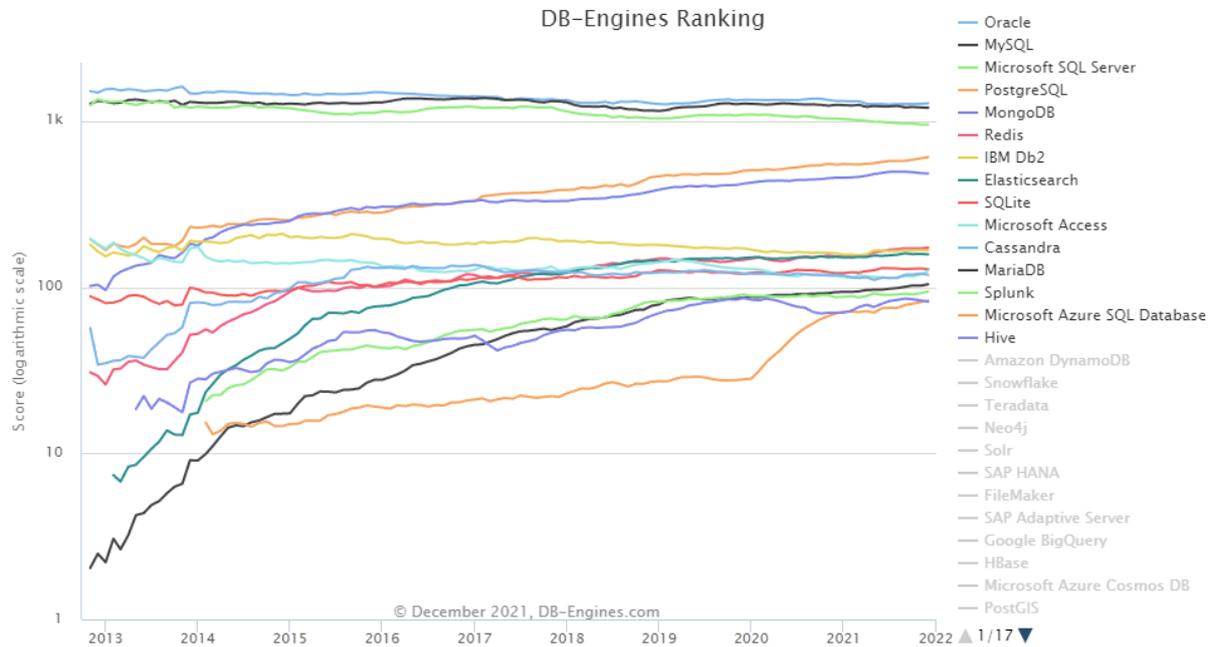


Fig. 38: ranking de popularidad de bases de datos de DB-Engines
Fuente: https://db-engines.com/en/ranking_definition

En la Figura 38 se puede ver que de las cinco más populares (Este ranking fue tomado de diciembre de 2021), cuatro son SQL y dos de estas son de licencia gratuita, MySQL y PostgreSQL. A continuación se comparan estos dos motores para demostrar cuál será la opción más adecuada a utilizar para hacer las consultas y realizar las visualizaciones de los datos. Para esta comparación se realizará una investigación sobre distintos papers que comparen estos motores.

Debido a que lo que se quiere realizar es una visualización, la operación a comparar debe ser una operación de consulta ya que es la que se realizará con mayor frecuencia. En el gráfico de la Figura 39 se puede observar que las bases de datos MariaDB, MySQL y PostgreSQL lograron tiempos similares para una operación determinada, sin embargo, PostgreSQL para 10,000 registros notó un aumento en el tiempo de ejecución de la operación en aproximadamente 38 milisegundos en comparación con MariaDB y MySQL.

³⁸ <https://db-engines.com/en/>

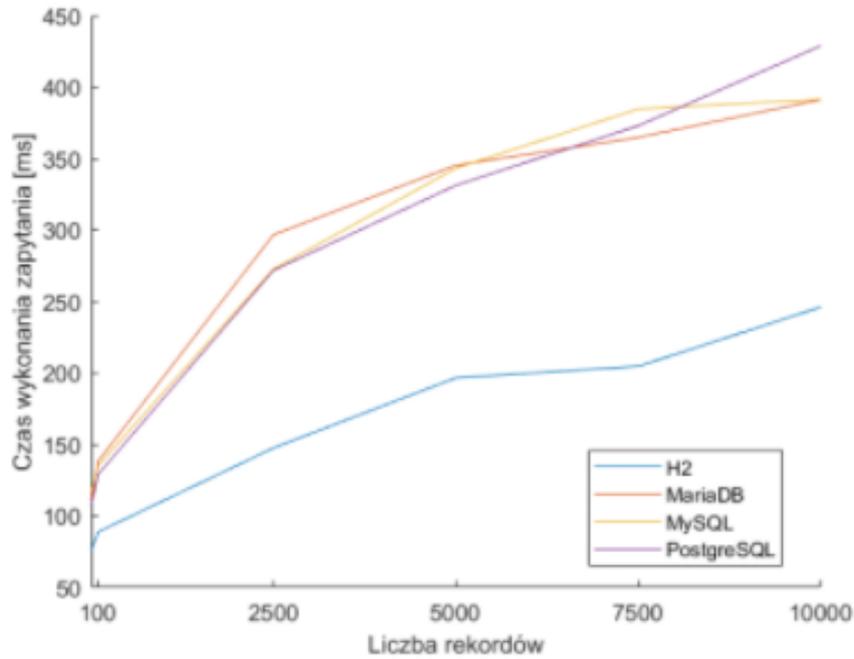


Fig 39: Tiempos promedio(en ms) de ejecución de la consulta para la operación de selección de datos
Fuente: [Kroc et al., 2019]

El siguiente estudio consiste en calcular el tiempo de ejecución de la operación de selección de datos con la cláusula WHERE. En la Figura 40 se puede observar que, al igual que en el estudio anterior, las bases de datos MariaDB, MySQL y PostgreSQL tienen tiempos similares.

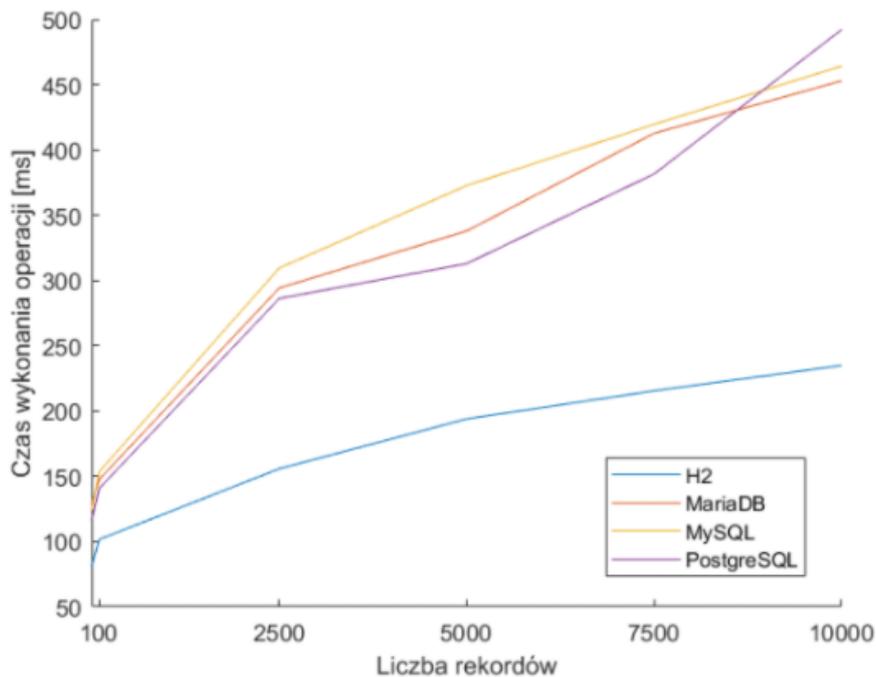


Fig 40: Tiempos promedio de ejecución de consultas para operaciones de selección de datos con la cláusula WHERE
Fuente: [Kroc et al., 2019]

Cambiando de estudio, en el informe [Oracle 19c, SQL Server 2019, Postgresql 12 and MySQL 8 database systems comparison] se puede observar en la Figura 41 que con una cantidad mucho más grande de datos, MySQL tiende a tener tiempos muchos peores que PostgreSQL llegando a obtener tiempos que duplican a los de PostgreSQL.

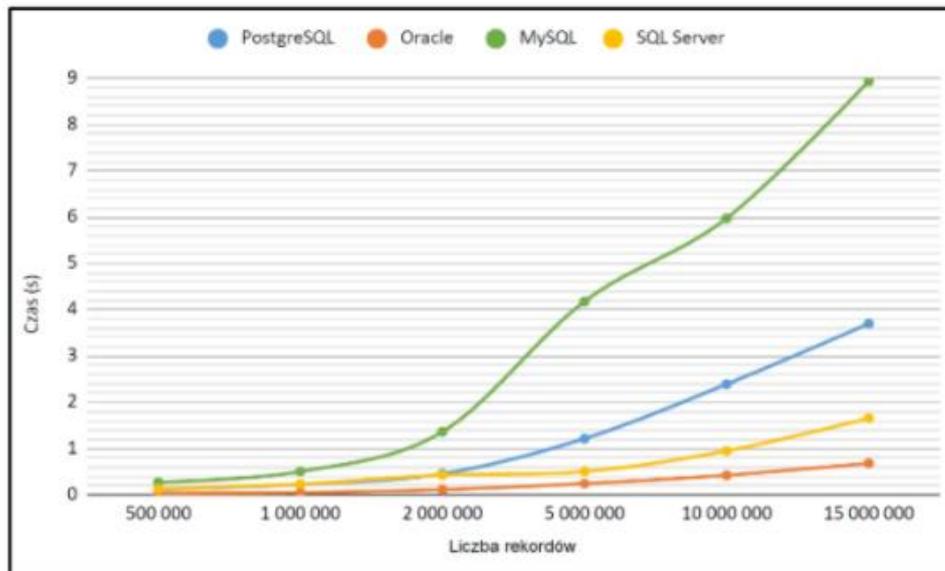


Fig 41: Gráfico de tiempo de búsqueda por número registros
Fuente: [Solarz & Szymczyk, 2020]

Es por esto que se decidió usar PostgreSQL como motor de bases de datos, ya que al estar tratando con un dataset de un gran número de datos los tiempos de PostgreSQL son más eficientes que los de MySQL.

Anexo 2: Comparación entre los Frameworks Django y Flask de Python

Django es un framework diseñado para trabajar utilizando el patrón modelo, vista y controlador (MVC). Django es lo que se conoce como un framework “full stack” para Python con el cual se pueden abordar todo tipo de proyectos en este lenguaje como el desarrollo web escalable y de alta calidad.

Por su parte, Flask es un micro-framework el cual permite el desarrollo de aplicaciones web de una forma mucho más sencilla que Django, ya que este está especialmente construido para un desarrollo web fácil y rápido a través de Python. Los que buscan un framework sencillo para desarrollar aplicaciones web utilizan Flask, es un micro framework ideal para principiantes. En Flask se utilizan las líneas de código necesarias para realizar cualquier acción por lo que es mucho más sencillo comprender la estructura de cualquier aplicación o script, y saber qué es lo que realiza.

Ambos frameworks permiten un backend limpio en manejo de usuarios, pudiendo limitar memoria y velocidad para evitar caídas en el servidor. Debido a ser un micro-framework, Flask posee un diseño minimalista de su estructura lo que le permite ser más rápido y con un mejor desempeño que Django. A su vez también resulta más fácil la integración con otras librerías de Python, lo que tiene una gran relevancia para lograr obtener variedad en las visualizaciones.

Anexo 3: Gráfico completo de la implementación del prototipo de visualizador

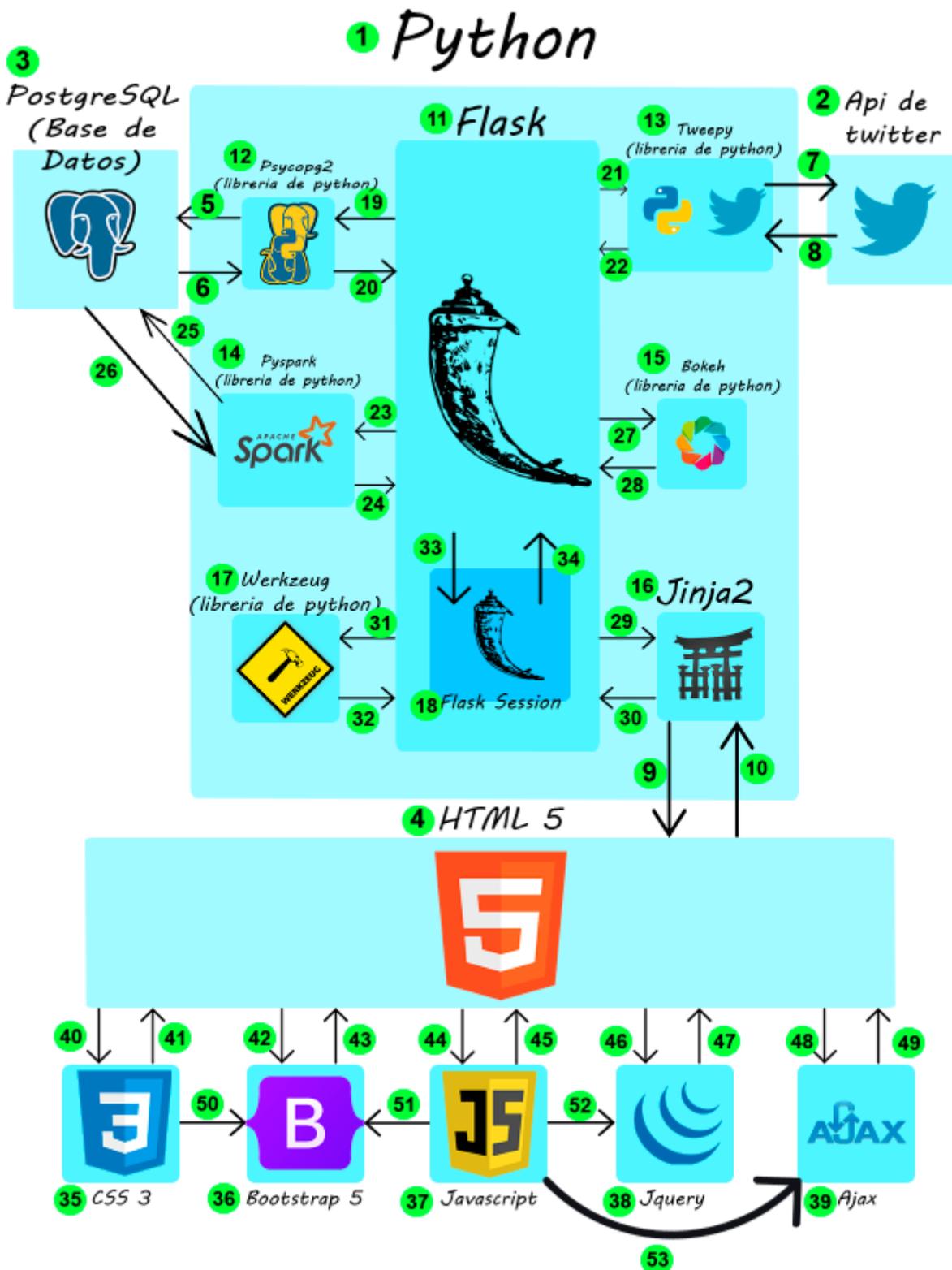


Fig 42: Gráfico completo de la implementación del prototipo de visualizador

Anexo 4: Documento de tareas a realizar en las pruebas de usuarios

Se realizarán las tareas agrupadas en dos pruebas de usuario distintas: por una lado las tareas de utilización del prototipo de visualizador web y por otro lado las tareas de gestión de Cursos y Actividades. A continuación se muestra la planilla presentada a los usuarios de la prueba:

Siga la siguiente lista de tareas en el orden numérico en que se encuentran:

Utilización del prototipo de visualizador web

1. Acceder a CeoDatum ingresando desde el navegador a la página: 127.0.0.1:5000/
2. Registrarse en CeoDatum.
3. Ingresar en la con el usuario y contraseña que se registró a CeoDatum.
4. Ingresar a los “dataset públicos”.
5. Buscar el dataset llamado “Determinaciones_de_covid_Argentina”.
6. Ingresar al graficador utilizando el dataset “Determinaciones_de_covid_Argentina”.
7. Generar un gráfico que muestre cómo han crecido las determinaciones de covid positivas en la nación Argentina desde el 15 de diciembre de 2022.
8. En la sección de “Grafos sociales” ingresar a “Twitter (recurrencias)”.
9. Generar una búsqueda de recurrencias de palabras relacionadas al covid aplicando los filtros que parezcan convenientes.
10. Cierre su sesión en CeoDatum.

Gestión de “Cursos” y “Actividades”

1. Ingrese nuevamente con el usuario “sebastian” y contraseña “franco” a CeoDatum.
2. Acceda a la sección de cursos.
3. Cree un curso.
4. Acceda al curso que ha creado.
5. Agregue un alumno al curso.
6. Agregue el dataset de “Determinaciones_de_covid_Argentina” al curso.
7. Cree una actividad.
8. Acceda nuevamente a la sección de cursos, ingrese al curso “Matemática 2022” e ingrese a la actividad “Análisis de casos de covid”.
9. Acceda a una de las entregas realizadas por los alumnos.
10. Realice la corrección de la actividad realizada por el alumno.

Anexo 5: Encuesta SUS post prueba de usuario

A continuación se mostraran en la Figura 43 las preguntas realizadas en el cuestionario:

1- Creo que usaría CeoDatum frecuentemente *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

2- Encuentro a CeoDatum innecesariamente compleja *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

3- Creo que CeoDatum fue fácil de usar *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

4- Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar CeoDatum *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

5- Las funciones de CeoDatum están bien integradas *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

6- Creo que CeoDatum es muy inconsistente *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

7- Imagino que la mayoría de gente aprendería a usar CeoDatum en forma muy rápida *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

8- Encuentro que CeoDatum es muy difícil de usar *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

9- Me siento con confianza al utilizar CeoDatum *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

10- Necesite aprender muchas cosas antes de ser capaz de usar CeoDatum *

- En completo desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Completamente de acuerdo

Fig 43: Encuestas SUS realizadas en las pruebas de usuario

Anexo 6: Resultados de las encuestas SUS

A continuación se mostrarán los resultados de las encuestas SUS completadas por los docentes luego de realizar las pruebas de usuario. El cálculo del resultado se hace sumando los puntos que dieron las preguntas y multiplicando en la suma por 2,5. Las mismas se dividirán como las pruebas de usuario en primera y segunda pasada:

Primera pasada

A continuación en la tabla 3 se pueden ver los resultados de las respuestas de los docentes de la primera pasada a las encuestas.

Participante/Pregunta	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Resultado
Participante 1	3	4	4	4	4	4	4	4	3	4	95
Participante 2	4	4	3	4	4	4	2	3	4	3	87.5
Participante 3	3	4	3	3	4	4	3	2	3	3	80

Tabla 3: Resultados de las respuestas de la primera pasada de las encuestas

Como promedio de los tres resultados se obtuvo 87.5 en la primera pasada de las encuestas SUS.

Segunda pasada

En esta sección, en la tabla 4 se pueden ver los resultados de las respuestas de los docentes de la segunda pasada a las encuestas.

Participante/Pregunta	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Resultado
Participante 1	3	4	4	3	4	4	3	4	3	3	87.5
Participante 2	3	4	3	4	4	4	3	4	4	4	92.5
Participante 3	4	4	4	4	3	4	3	4	3	4	92.5

Tabla 4: Resultados de las respuestas de la segunda pasada de las encuestas

Como promedio de los tres resultados se obtuvo 90.83 en la segunda pasada de las encuestas SUS.