

Diseño e implementación frontend de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas.

Rubén Danilo García Bautista, Angie Thalía Ríos Sandoval

Trabajo de grado para optar por el título de Ingeniero de Telecomunicaciones

Director

Ricardo Andrés Medina Puentes

Magister en Redes y Sistemas de Telecomunicaciones

Universidad Santo Tomás, Bucaramanga

División de Ingenierías y Arquitectura

Ingeniería de Telecomunicaciones

2022

Dedicatoria

Dedicamos nuestra tesis a Dios y a nuestros padres que nos dieron el privilegio de poder cumplir una meta más en nuestra vida profesional, tenemos una inmensa gratitud con nuestras familias porque siempre estuvieron en cada momento apoyándonos; también damos gracias a Sergio Zabala, decano de nuestra facultad de ingeniería de telecomunicaciones y todo el equipo de docentes que aportaron demasiados conocimientos en nuestra formación académica. A la ingeniera Yuli Pizarro y al ingeniero Ricardo Puentes un agradecimiento especial por el tiempo compartido, y las enseñanzas que nos brindaron en este proceso son personas extraordinarias que siempre dieron lo mejor en cada disposición. ¡Gracias a todos por su acompañamiento!

Contenido

Introducción	11
1. Diseño e implementación de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas	12
1.1 Planteamiento del problema.....	12
1.2 Justificación.....	13
1.3 Objetivos	14
1.3.1 Objetivo general	14
1.3.2 Objetivos específicos.....	14
2. Marco referencial.....	15
2.1 Marco teórico	15
2.2 Marco conceptual	17
3. Desarrollo del proyecto.....	19
3.1 Planificación del desarrollo.....	19
3.1.1 Investigación del lenguaje de diseño	19
3.1.2 Desarrollo de la estructura del proyecto	20
3.2 Desarrollo del aplicativo web.....	22
3.2.1 Desarrollo de componentes para la interfaz de usuario	22
3.2.2 Implementación de la mutación de estados	33
3.3 Desarrollo del sitio web.....	36
3.3.1 Desarrollo del sitio web del aplicativo	36
3.4 Puesta en operación del proyecto	38
3.4.1 Migración de la plataforma a una aplicación híbrida	38

3.4.2 Despliegue del software.....	40
3.4.3 Liberación del código fuente	40
3.4.4 Definición del manual de usuario	41
4. Conclusiones	43
5. Trabajo a futuro.....	44
Referencias.....	45

Lista de tablas

Tabla 1. <i>Comparativa de uso para frameworks de JavaScript</i>	16
---	----

Lista de figuras

Figura 1. <i>Comparativa de Frameworks UI.</i>	20
Figura 2. <i>Lógica de visualización y comunicación de interfaces</i>	20
Figura 3. <i>Categorización de componentes</i>	21
Figura 4. <i>Formatos de los gráficos</i>	21
Figura 5. <i>Ruta de navegación</i>	22
Figura 6. <i>Funcionalidades de perseverancia</i>	23
Figura 7. <i>Características de Administración</i>	24
Figura 8. <i>Monitoreo y visualización de proyectos</i>	25
Figura 9. <i>Administrador de proyectos de estadística</i>	25
Figura 10. <i>Creación de un nuevo dashboard de métricas</i>	26
Figura 11. <i>Creación de un nuevo dashboard de estadística</i>	26
Figura 12. <i>Administrador de base de datos</i>	27
Figura 13. <i>Funcionalidad "Agregar nueva Base de Datos"</i>	27
Figura 14. <i>Selección de tipo de gráfico</i>	28
Figura 15. <i>Consulta a la base de datos en el dashboard de métricas</i>	29
Figura 16. <i>Previsualización de la información consultada</i>	29
Figura 17. <i>Configuración de características del gráfico</i>	30
Figura 18. <i>Visualización del gráfico creado</i>	30
Figura 19. <i>Análisis estadísticos realizados</i>	31
Figura 20. <i>Selección de tipo de análisis estadístico</i>	32
Figura 21. <i>Consulta a la base de datos en el dashboard de analítica</i>	32
Figura 22. <i>Propiedades de la visualización</i>	33

Figura 23. <i>Stores de almacenamiento de estados globales</i>	34
Figura 24. <i>Estado global del componente</i>	34
Figura 25. <i>Componentes de la vista principal en VueJS</i>	36
Figura 26. <i>Página Web del proyecto</i>	37
Figura 27. <i>Manual de usuario (Guía de inicio)</i>	42
Figura 28. <i>Manual de usuario (Estructura de carpetas)</i>	42
Figura 29. <i>Manual de usuario (Instalación)</i>	42

Resumen

La presente tesis realiza un software libre para la administración y análisis de información de bases de datos relacionales y no relacionales en medianas y pequeñas empresas “MiPymes”. El software cuenta con la posibilidad de modificar y adecuar el código de acuerdo con las necesidades de cada empresa; así mismo, en la construcción del dashboard se da enfoque a la representación y visualización de datos mediante gráficos y diagramas para que brinden información de calidad a los usuarios con el propósito de generar una interfaz intuitiva mediante el framework de JavaScript VueJS. Cabe mencionar que, para el desarrollo del software libre, fue fundamental indagar acerca de conceptos y metodologías en diferentes fuentes con antecedentes en el tema para planificar y dar inicio con al flujo de desarrollo pertinente. Los resultados muestran la funcionalidad del dashboard configurado para que los usuarios puedan hacer consultas y visualicen graficas con la respectiva información en formato de vectores SVG, así mismo imágenes de alta calidad PNG y valores individuales CSV; por otro lado, cada componente grafico se le permite ser redimensionado y desplazado dentro del panel de visualización, con el fin de permitirle al usuario el total control de la información seleccionada.

Palabras clave: MiPymes, VueJs, dashboard, métricas

Abstract

This thesis develops an open-source software for the administration and analysis of information from relational and non-relational databases in medium and small companies "MiPymes". The software has the possibility of modifying and adapting the code according to the needs of each company; Likewise, in the construction of the dashboard, focus is given to the representation and visualization of data through graphs and diagrams so that they provide quality information to users with the purpose of generating an intuitive interface through the VueJS JavaScript framework. It is worth mentioning that, for the development of open source software, it was essential to inquire about concepts and methodologies in different sources with background on the subject to plan and start the relevant development flow. The results show the functionality of the dashboard configured so that users can make queries and view graphs with the respective information in SVG vector format, as well as high-quality PNG images and individual CSV values; on the other hand, each graphic component is allowed to be resized and moved within the display panel, in order to allow the user full control of the selected information.

Keywords: MiPymes, VueJs, dashboard

Glosario

Lista alfabética de términos relevantes y sus definiciones para la comprensión del documento.

MiPymes: pequeña o mediana empresa con un volumen de recursos limitado.

JavaScript: lenguaje de programación sin necesidad de compilación usado en navegadores web.

VueJS: marco de JavaScript de código abierto enfocado a crear interfaces de usuario y aplicaciones web de una sola página.

Dashboard: Herramienta que organiza y monitorea de manera visual la información.

Pinia Store: Conjunto de archivos de texto en formato JavaScript que conserva la lógica y comportamiento de un componente creado en VueJS.

Introducción

El dominio de las telecomunicaciones desde la década de los 80, ha incluido el inicio de nuevas tecnologías para la administración de base de datos en empresas. Muchas microempresas recolectan y administran datos en hojas de cálculo; siendo consecuencia de cargas laborales y muchas veces alto índice de inseguridad en el almacenamiento de información. La implementación de softwares libres es una alternativa para el progreso tecnológico de MiPymes, donde cada vez aumenta la credibilidad de códigos abiertos, los cuales resuelven desafíos comerciales y mejoran la eficiencia en las empresas. Uno de los impedimentos para que las medianas y pequeñas empresas utilicen estas alternativas tecnológicas es el desconocimiento de softwares libres.

En esta tesis se presenta un proyecto de código abierto enfocado al monitoreo de base de datos relacionales y no relacionales para MiPymes; El principal objetivo es ofrecer un sistema de uso y modificación libre para que se adapte a las necesidades de cada negocio y se pueda generar una conexión con la base de datos de la microempresa; llevándola a un entorno local, donde se puedan obtener consultas originadas por usuarios y de esta manera visualizar la información deseada en gráficos detallados. Se hace énfasis en la visualización de datos debido a que es un factor importante en el desarrollo comercial y competitivo de las MiPymes, ya que mejora la planeación de estrategias, incluyendo también, análisis estadístico y monitoreo de grandes cantidades de datos.

1. Diseño e implementación frontend de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas

1.1 Planteamiento del problema

La MiPymes en Colombia representan más del 90% del sector productivo de Colombia, generan alrededor del 35% del producto interno bruto (PIB) y cerca del 80% del empleo [1], actualmente una de cada dos MiPymes cuenta con una base de datos y dentro de sus ventajas destaca la facilidad para acceder a la información, la mejora en la toma de decisiones y el aumento en la reacción de imprevistos [2], sin embargo, observar y monitorear esta información suele ser una labor tediosa y que no es posible realizar con únicamente una base de datos.

Múltiples soluciones conocidas como “Dashboards para el monitoreo de bases de datos” han tomado gran relevancia en la última década, en las cuales, el manejo de la información se ha convertido en el factor principal de la planeación para el crecimiento de una empresa; Soluciones que habilitan al usuario para crear y organizar gráficos con la información de una base de datos permitiéndoles desarrollar una ventaja competitiva en el mercado.

Estos Dashboards actualmente presentan una gran ventaja a las empresas que lo implementan, sin embargo, cuentan con una brecha disruptiva que imposibilita a pequeñas y medianas empresas equiparar a la competencia, sus altos costos monetarios, disminuida posibilidad de modificación y complicada curva de aprendizaje.

1.2 Justificación

La visualización de los datos se ha vuelto relevante en las empresas en los últimos años debido a la forma en que el cerebro humano procesa la información, la implementación de diagramas o gráficas para visualizar grandes cantidades de datos resulta más práctico que registrarlos en hojas de cálculo para su posterior análisis. La visualización de datos permite a las empresas identificar áreas que necesitan atención o mejoras, así como establecer qué factores influyen el comportamiento de los clientes, mantenerse al día a los requerimientos y reaccionar con mayor velocidad ante incidentes que ocurran y que puedan ser monitoreados gracias a la información de la empresa visualizada mediante un Dashboard.

El desarrollo Frontend de un software libre para la visualización de la información en bases de datos relacionales y no relacionales de MiPymes tiene como objetivo generar un sistema de uso y modificación libre para el monitoreo de la información mediante gráficos y diagramas generados a través de la interfaz de usuario, con un control intuitivo y, un flujo de uso simplificado orientado a brindar a las pequeñas y medianas empresas una funcionalidad adaptable a sus necesidades y, a los desarrolladores, una base que cumpla con todos los estándares para su modificación libre.

1.3 Objetivos

1.3.1 *Objetivo general*

Desarrollar un software de uso y modificación libre para el monitoreo y análisis de información obtenida en bases de datos relacionales y no relacionales, mediante gráficos y estadísticas generados a través de la interfaz de usuario con el fin de facilitar la interacción y la curva de aprendizaje para su uso en pequeñas y medianas empresas orientado a mejorar su crecimiento y planificación estratégica.

1.3.2 *Objetivos específicos*

1. Desarrollar componentes de interfaz y experiencia de usuario mediante el framework JavaScript VueJS con el fin de segmentar y presentar las funcionalidades del aplicativo con un control intuitivo y completo de la plataforma a los usuarios.

2. Generar una aplicación híbrida usando Electron PHP mediante la migración del software obteniendo una aplicación nativa de escritorio y de esta forma el usuario cuente con su uso sin la necesidad de una conexión a internet.

3. Registrar el aplicativo en el repositorio de GitHub y la página web oficial bajo licencia Copyleft ofreciendo un código que sea accesible al público para su uso.

4. Definir el manual de usuarios exponiendo de manera intuitiva las funcionalidades del software que facilite el uso del aplicativo a las empresas.

2. Marco referencial

2.1 Marco teórico

Los paneles de administración [3] son herramientas de visualización que se utilizan para el análisis de datos dentro de varios negocios. Estos paneles ofrecen información esencial sobre informes y métricas. Los tableros muestran datos críticos en tiempo real e indicaciones de rendimiento, de forma similar al panel de instrumentos de un vehículo. Esto ayuda a los usuarios a tomar decisiones más informadas y navegar por su entorno de manera más efectiva. Los métodos de visualización se han sincronizado y, a través de una sola interfaz, se representa aproximadamente la misma cantidad de datos que antes. La tecnología de visualización de datos permite a las empresas brindar un mejor servicio a sus clientes, además de desempeñar un papel en la mejora de las operaciones y el desarrollo de nuevos bienes. La presentación de datos visualmente es un método de transmitir información que es a la vez clara y sucinta. Los paneles de administración son un método eficiente para representar visualmente los datos, lo que ayuda en el proceso de toma de decisiones dentro de las organizaciones.

Se implementa el framework VueJS de JavaScript [4] para la creación de la aplicación web. VueJS es una alternativa interesante a considerar al seleccionar un marco para el desarrollo de aplicaciones web porque se implementa mediante JavaScript [5]. De la misma forma que el framework VueJS se basa en el modelo de componentes, que permite crear aplicaciones web de forma modular reutilizando código, ofrece muchos beneficios en términos de rendimiento y facilidad de uso, además de contar con una gran comunidad, debido a esto, el desarrollo de aplicaciones web puede hacerse de manera más eficiente, ya que se pueden construir nuevos componentes reutilizando código de componentes anteriores.

VueJS también ofrece una gran cantidad de herramientas y librerías para facilitar el desarrollo de aplicaciones web. Algunas de estas herramientas son:

- Vue CLI: permite crear proyectos de frontend de forma rápida y sencilla.
- Ant Design: Librería que proporciona componentes de UI para aplicaciones web.
- Vue Router: librería para crear enrutamiento de Single Page Applications (SPA).
- Pinia: librería para el manejo estados del store en aplicaciones.

VueJS uno de los frameworks de frontend más utilizados en la actualidad como se presenta en el siguiente grafico comparativo basado en las estrellas (medio de puntuación) que han adquirido por parte de los usuarios en sus respectivos repositorios oficiales de GitHub:

Tabla 1. *Comparativa de uso para frameworks de JavaScript*

#	Organización	Framework	GitHub Stars [K]
1	VueJS	Vue	194
2	Facebook	React	185
3	Angular	Angular	80.4

El proceso de desarrollo de software ha evolucionado como resultado de la proliferación de sistemas de control de versiones distribuidos como git y plataformas construidas sobre ellos como GitHub [6]. El proceso de desarrollo de software se ha simplificado sustancialmente gracias a sitios web como GitHub y GitLab, que hacen posible que los ingenieros trabajen juntos en un entorno centralizado.

Además, estas plataformas incluyen una variedad de funciones adicionales, como soporte para integración e implementación continuas, herramientas de revisión de código y compatibilidad entre plataformas. Debido a que tienen un impacto directo en la calidad general del software, estas

funciones generan una serie de beneficios durante las fases del ciclo de vida del software conocidas como desarrollo y mantenimiento. Uno de estos beneficios es la reducción de las dificultades de integración. La empresa 720Tec [7] desarrolló un programa que es capaz de leer, analizar y visualizar los datos que se recopilan de las diversas bases de datos que mantiene.

2.2 Marco conceptual

Cuando se desarrolla una aplicación para software de código abierto, se hace con el entendimiento de que sus creadores deben respetar la libertad de sus usuarios y de la comunidad en su conjunto [8]. Esto significa que los usuarios tienen la libertad de usar, copiar, alterar y mejorar el software. Se ha concedido una licencia copyleft al proyecto para que el software pueda ser utilizado y actualizado por un gran número de usuarios con la condición de que siga teniendo la naturaleza de software libre. Esto puede entenderse como una indicación de que el software debe seguir distribuyéndose sin cargo. Dicho de otro modo, es posible que el software en cuestión no se venda. Se dice que los usuarios tienen ciertos principios de libertad si el software que usan se considera libre. Por un lado, la libertad de redistribuir copias para ayudar a otros, y por otro, la libertad de redistribuir copias de las versiones modificadas del proyecto a terceros. Además, el usuario tendrá la libertad de ejecutar el programa bajo su responsabilidad y por el motivo que sea necesario. También tendrán la libertad de estudiar cómo funciona el programa y cambiarlo para que puedan satisfacer sus requerimientos teniéndole como base.

Business Intelligence (BI) es una colección de técnicas y procesos analíticos que permiten a las empresas comprender y analizar de manera efectiva las operaciones comerciales, la información y las relaciones entre ellos y, por lo tanto, poder tomar las mejores decisiones posibles. Los paneles de visualización de métricas internas forman parte de la "Gestión del rendimiento

empresarial" [9]. La inteligencia empresarial (BI) tiene un impacto perceptible en la planificación estratégica y la competitividad de las empresas, y su aplicación es cada vez más global. Una forma de definir las operaciones de inteligencia empresarial es como un conjunto de técnicas y herramientas analíticas que permiten a las empresas comprender y analizar de manera efectiva los negocios, la información y las relaciones entre ellos y, por lo tanto, estar en condiciones de tomar las mejores decisiones para la empresa. Aunque el término "operaciones de inteligencia de negocios" es muy amplio y su significado varía según la perspectiva de cada organización, esta es una forma de definir las operaciones de inteligencia de negocios.

Los paneles de visualización de datos internos [10] son las herramientas que consisten en construir un mapa visual a partir de datos para resaltar patrones, descubrir enlaces o mostrar tendencias. Estos paneles también se pueden utilizar para mostrar las tendencias de toda la empresa. Es importante que los mapas visuales sean atractivos y legibles. El usuario típico no es un estadístico ni un analista, sino una persona que participa activamente en su trabajo. Es esencial tener en cuenta la percepción humana al desarrollar un mapa visual eficiente. ¿Las personas requieren acceso a todos los datos para hacer una copia? ¿Es posible disminuir el número de componentes gráficos manteniendo el mismo nivel de información? Cuando se intenta representar de manera efectiva varios tipos de datos, se recomienda utilizar varios formatos de visualización alternativos. Las personas no son objetivas cuando se trata de un mapa visual; la forma en que lo perciben depende de qué tan conveniente sea para ellos verlo, su nivel de cansancio y la forma en que se diseñó el mapa visual. No se debe descartar el lenguaje de colores, el tipo de letra y el tamaño de la tipografía. Para el usuario, cualquiera de estos componentes puede funcionar como una representación del propósito previsto del mapa visual. Cuando los datos se visualizan en un mapa que es fácil de leer y comprender, aumenta la utilidad del mapa como herramienta de análisis.

El desarrollo de aplicaciones web integra dos flujos de trabajo independientes que interactúan entre sí para generar las funcionalidades que se integran, estos son el Backend y el Frontend, es en este último al que se encuentra enfocado el desarrollo del proyecto. El frontend define la estructura visual e interactividad con el aplicativo, lo que el usuario ve y experimenta, trabajando con lenguajes de programación tales como JavaScript y lenguajes de marcado y estilo como lo son HTML y CSS encargados de estructurar, formatear y organizar el contenido que se está generando en el aplicativo.

3. Desarrollo del proyecto

3.1 Planificación del desarrollo

3.1.1 Investigación del lenguaje de diseño

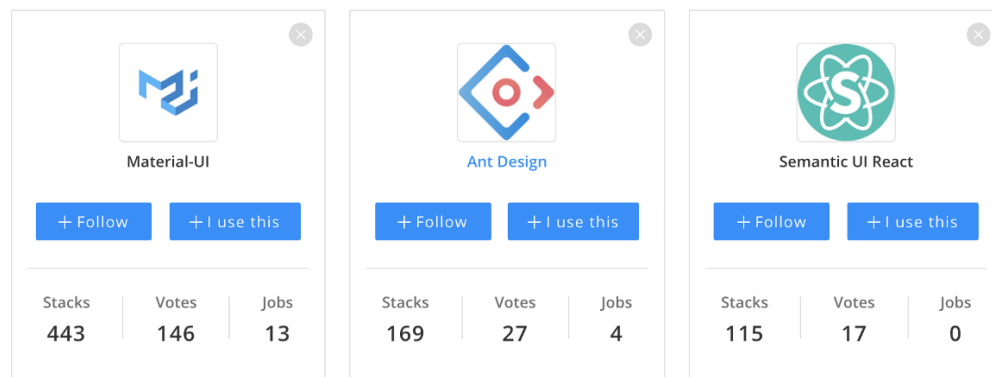
Una de las estructuraciones más importantes que se tienen en cuenta al momento del desarrollo web, enfocado a Frontend, es la del lenguaje de diseño, también conocido como políticas o vocabulario de diseño, es la esquematización o estilo que guiará todo el flujo creativo del desarrollo.

VueJS [11] cuenta con paquetes que implementan diferentes lenguajes de diseño los cuales pueden ser adaptados por el usuario de manera libre permitiendo crear una identidad única para cada desarrollo de software.

Teniendo en cuenta las necesidades del proyecto se optó por el uso de Ant Design [12], una librería de patrones y principios enfocados al sector corporativo. Incluye componentes de

interfaz y experiencia de usuario desarrollados mediante principios de abstracción de código permitiendo la reutilización y optimización del flujo de ejecución en el proyecto.

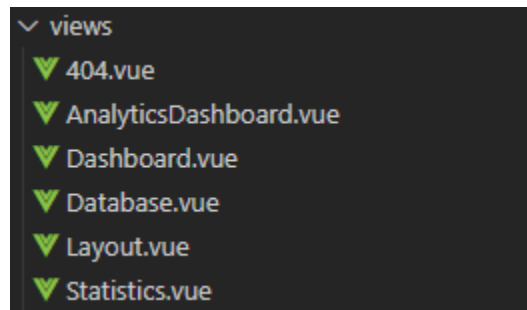
Figura 1. *Comparativa de Frameworks UI.*



3.1.2 Desarrollo de la estructura del proyecto

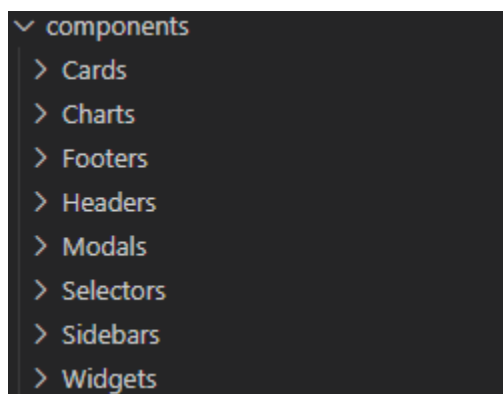
Teniendo como base los principios de abstracción de código provistos por el framework de UI seleccionado y, las funcionalidades planeadas para el aplicativo, se estructuran los componentes utilizando un sistema de jerarquía teniendo como elemento principal las vistas, archivos que contienen la lógica de visualización y comunicación de interfaces tal como se presenta a continuación:

Figura 2. *Lógica de visualización y comunicación de interfaces*



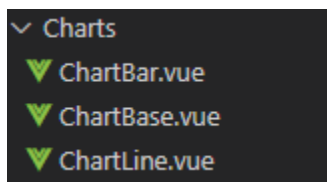
Posteriormente en la jerarquía, se integran los Componentes, archivos de código que contienen elementos visuales y funcionales que podrán ser utilizados múltiples veces dependiendo del requerimiento de la vista. Los componentes se encuentran categorizados por el tipo de visualización a la que corresponden de la siguiente manera:

Figura 3. *Categorización de componentes*



El nombre de los componentes individuales corresponde a su vista destino o su tipo y a su funcionalidad principal como lo es el caso de los gráficos donde se especifica que es de tipo chart y que cuenta con la funcionalidad de graficar en formato de línea, base o barras.

Figura 4. *Formatos de los gráficos*



La lógica funcional engloba la parte final de la estructura del proyecto, debido a que se hace necesario mantener una comunicación y formato con el proyecto de Backend, se establece un concepto denominado “Store”, debido a la complejidad del proyecto la interacción entre

componentes dispersos puede resultar afectada de forma en que su dificultad de configuración aumenta de forma lineal con la escala del proyecto. Una Store permite independizar estados y funcionalidades de componentes para posteriormente distribuir estas mutaciones.

Con el auge de VueJS en su versión 3.0 y su Composition API, nuevas librerías fueron creadas por la comunidad aprovechando las funcionalidades implementadas, una de ellas es Pinia [13], una librería enfocada a la administración de estados y mutaciones en el proyecto, brindando reactividad en los estados y permitiendo una comunicación constante entre componentes.

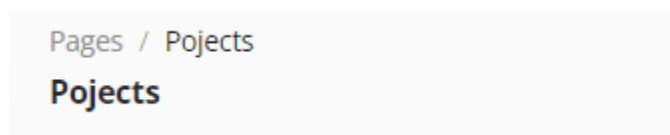
3.2 Desarrollo del aplicativo web

3.2.1 Desarrollo de componentes para la interfaz de usuario

Teniendo como base la estructuración de componentes previa, se inicia el desarrollo con las vistas principales y componentes en paralelo, puesto que estos últimos son desarrollados según los requerimientos de la vista. Inicialmente se desarrolla el esquema global de las vistas del cual hacen parte el encabezado y el panel lateral.

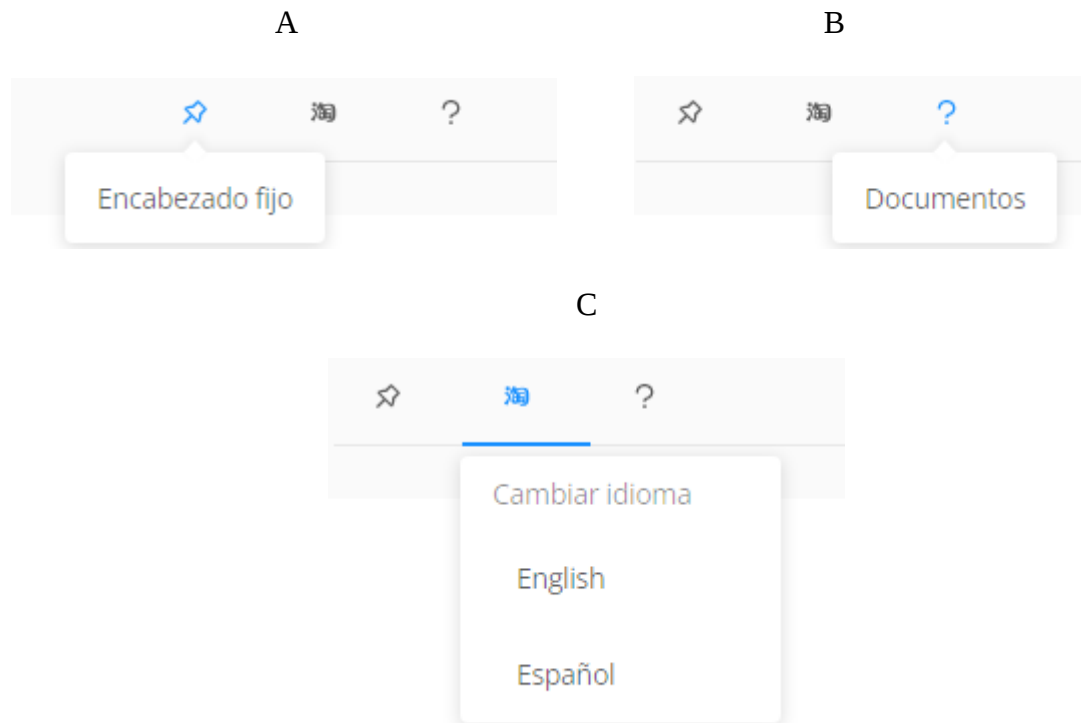
El encabezado tiene como propósito principal brindar funcionalidades de accesibilidad mediante las cuales tiene la capacidad de observar y alterar factores importantes del aplicativo, el primero la presentación de la ruta en la cual se encuentra ubicado y la capacidad de permitirle navegar por las mismas.

Figura 5. Ruta de navegación



El segundo contempla las funcionalidades de perseverancia de visualización del encabezado (A), acceso directo a la documentación (B) y selección de idioma (C).

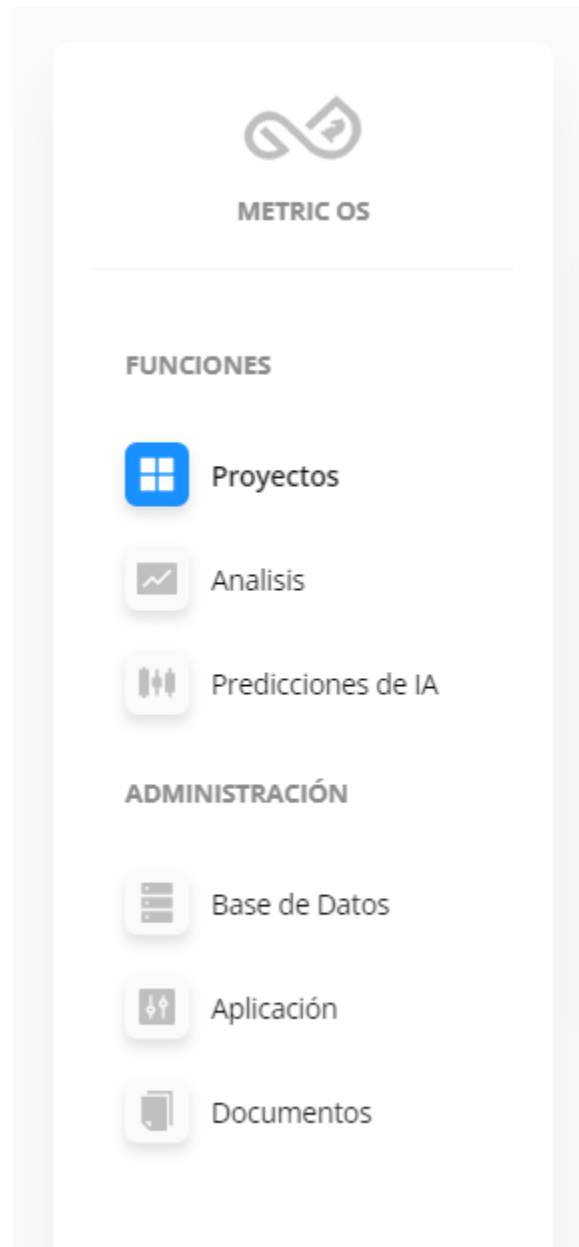
Figura 6. *Funcionalidades de perseverancia*



El panel lateral incluye dos características principales con el objetivo de facilitar el acceso del usuario a las funciones principales del aplicativo; El primero es el acceso directo a las vistas principales, estas incluyen los paneles de visualización de métricas denominado “Proyectos”, el panel de visualización de analíticas denominado “Análisis” y un campo previsto para un trabajo futuro que englobará funcionalidades de inteligencia artificial denominado “Predicciones de IA”.

El segundo incluye características de administración donde se habilita al usuario para que configure y administre sus bases de datos, modifique factores de diseño del aplicativo y, finalmente, acceda a la documentación.

Figura 7. *Características de Administración*



Dentro de los dos paneles de visualización principales encontramos el dashboard de métricas, donde se ubicará la información obtenida de las bases de datos del usuario y se permite interactuar con ella y, el dashboard de análisis, donde el usuario podrá generar gráficos los cuales contienen información estadística según los requerimientos.

Los paneles de métricas y análisis permiten la visualización de los proyectos respectivos creados por el usuario incluyendo una breve descripción de estos y, el botón para acceder a su contenido. En su sección final, se incluye la funcionalidad que habilita la creación de un nuevo panel de métricas.

Figura 8. *Monitoreo y visualización de proyectos*

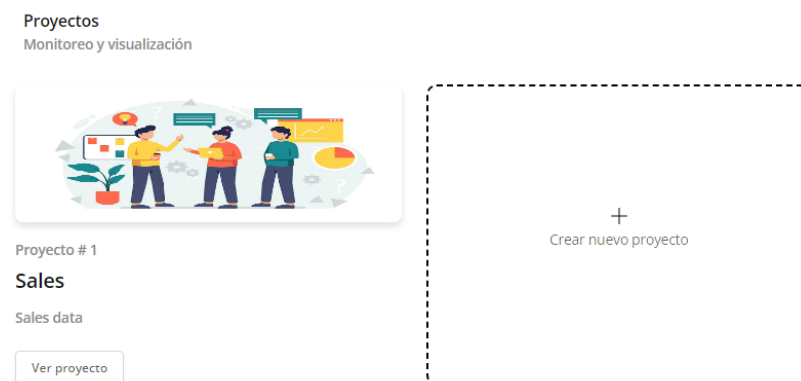
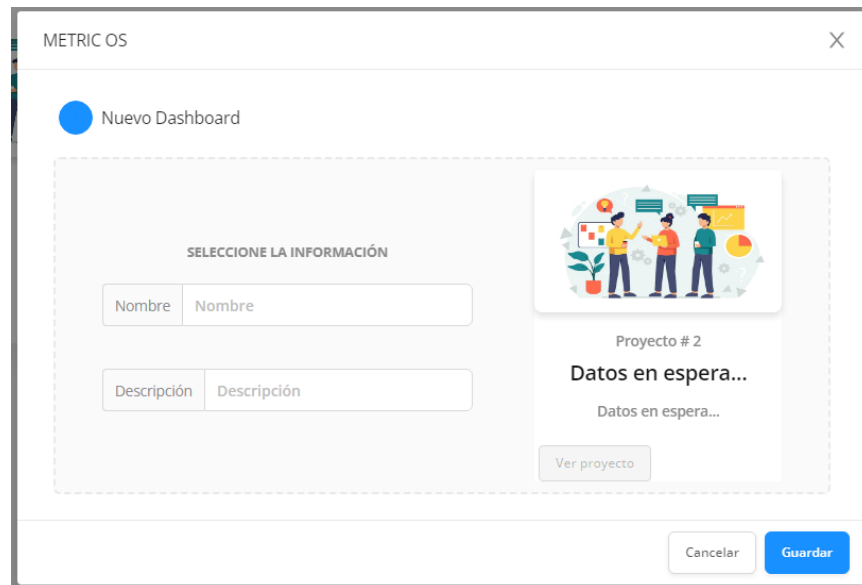


Figura 9. *Administrador de proyectos de estadística*



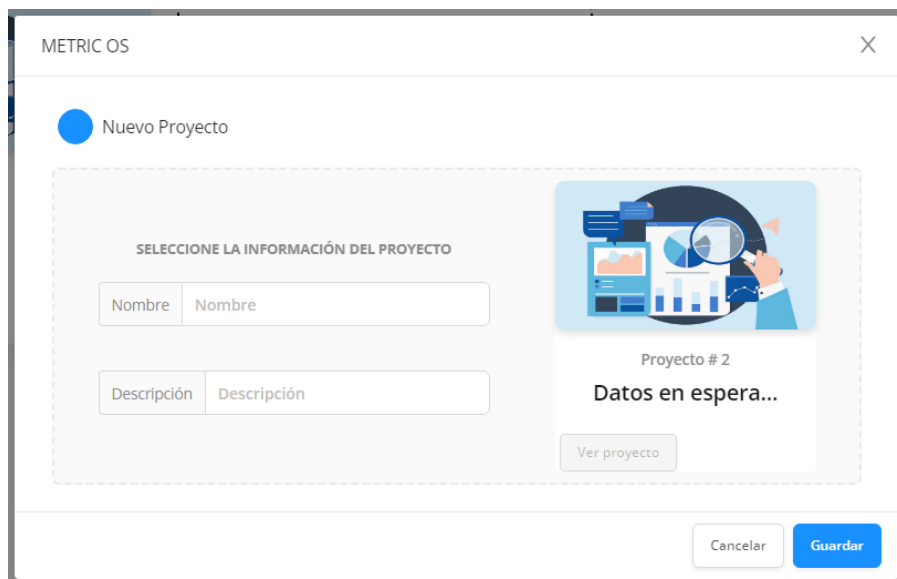
La creación de un nuevo proyecto se lleva a cabo mediante un formulario que le solicita al usuario el nombre y una descripción respectiva. Se habilita la previsualización del nuevo proyecto antes de finalizar su creación.

Figura 10. Creación de un nuevo dashboard de métricas



The screenshot shows a web application window titled 'METRIC OS'. Inside, there's a section titled 'Nuevo Dashboard' with a blue circular icon. Below this, a dashed box contains the heading 'SELECCIONE LA INFORMACIÓN'. There are two input fields: 'Nombre' (Name) and 'Descripción' (Description), each with a placeholder text 'Nombre' and 'Descripción' respectively. To the right of the input fields is a preview card for 'Proyecto # 2' with the text 'Datos en espera...' (Data pending...) and a 'Ver proyecto' (View project) button. At the bottom right of the form are 'Cancelar' (Cancel) and 'Guardar' (Save) buttons.

Figura 11. Creación de un nuevo dashboard de estadística



The screenshot shows a web application window titled 'METRIC OS'. Inside, there's a section titled 'Nuevo Proyecto' with a blue circular icon. Below this, a dashed box contains the heading 'SELECCIONE LA INFORMACIÓN DEL PROYECTO'. There are two input fields: 'Nombre' (Name) and 'Descripción' (Description), each with a placeholder text 'Nombre' and 'Descripción' respectively. To the right of the input fields is a preview card for 'Proyecto # 2' with the text 'Datos en espera...' (Data pending...) and a 'Ver proyecto' (View project) button. At the bottom right of the form are 'Cancelar' (Cancel) and 'Guardar' (Save) buttons.

La administración de la base de datos del usuario se habilita desde la característica *Administración – Base de datos* ubicada en el panel lateral del aplicativo, aquí el usuario podrá intercambiar entre las diferentes bases de datos se encuentren registradas, así como la funcionalidad de agregar una nueva si lo requiere mediante un formulario sencillo e intuitivo.

Figura 12. *Administrador de base de datos*

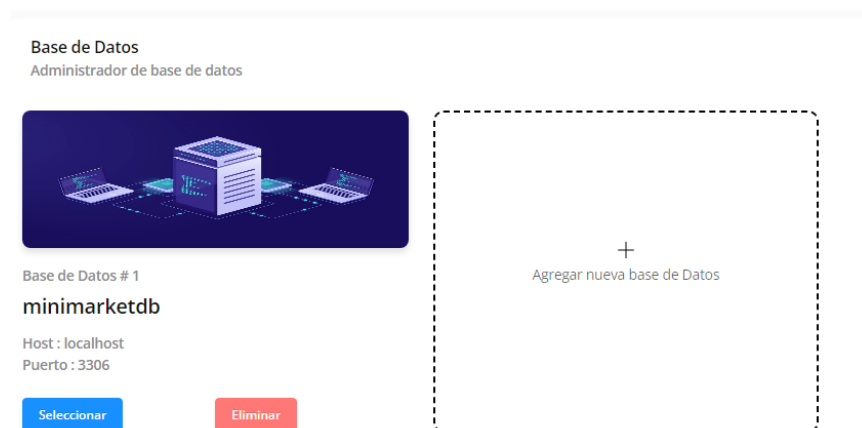
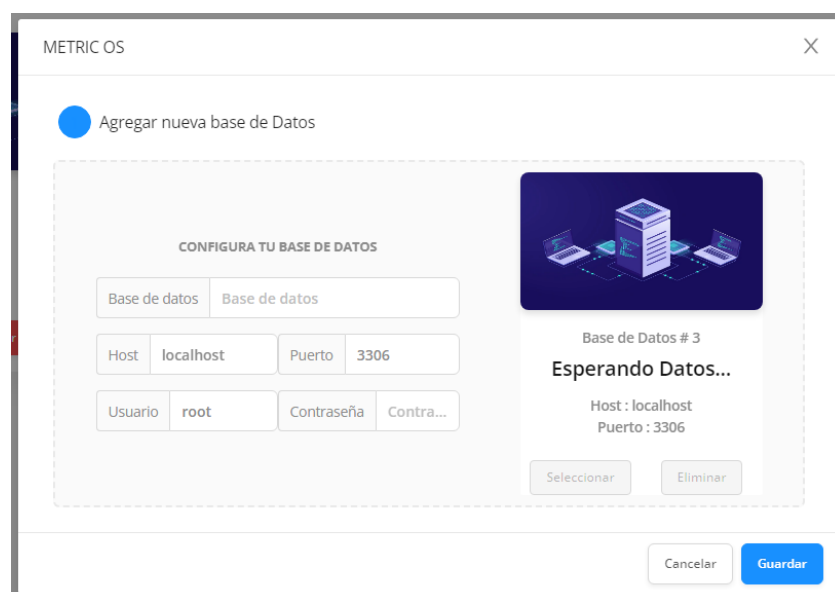


Figura 13. *Funcionalidad "Agregar nueva Base de Datos"*



Los proyectos dentro de la sección de *monitoreo y visualización* permiten acceder a sus dashboards de métricas respectivamente, en esta sección se encuentran las gráficas que el usuario haya configurado. Las gráficas cuentan con funcionalidades relevantes para el usuario tales como lo son el aumento y desplazamiento en áreas específicas y, la descarga de la gráfica y su información en formato de vectores SVG, imagen de alta calidad PNG o valores individuales CSV. Cada componente grafico cuenta con la funcionalidad de ser redimensionado y desplazado dentro del panel de visualización permitiendo al usuario un control completo sobre la presentación de la información.

La creación de un gráfico se hace desde el botón flotante  el cual despliega el proceso que está compuesto por tres secciones, la primera permite seleccionar el tipo de grafico que se desea crear, junto con la propiedad de habilitar los controles de aumento interno.

Figura 14. Selección de tipo de gráfico



La sección numero dos permite realizar la consulta a la base de datos y establecer propiedades necesarias para el filtrado y ordenamiento de la información. Mediante el botón *TEST* se permite observar una previsualización en bruto de la información que la consulta ha retornado y así evitar errores a nivel de usuario.

Figura 15. Consulta a la base de datos en el dashboard de métricas

METRIC OS

✓ Tipo de Gráfico — Selección de datos — 3 Propiedades del componente

INSERTAR LA CONSULTA DE DATOS

Seleccione	*	Desde	TablaNoExistente	Donde	Condición
Tiene	Condición	Agrupar por	Condición	Ordenar por	Condición

Prueba

ASÍ ES COMO SE VEN SUS DATOS:

Tabla o vista no encontrada en la base de datos

Anterior Siguiente

Figura 16. Previsualización de la información consultada

METRIC OS

✓ Tipo de Gráfico — Selección de datos — 3 Propiedades del componente

INSERTAR LA CONSULTA DE DATOS

Seleccione	*	Desde	ventas	Donde	Condición
Tiene	Condición	Agrupar por	Condición	Ordenar por	fecha

Prueba

ASÍ ES COMO SE VEN SUS DATOS:

```
[{"id":1,"cantidad":200,"fecha":"2022-01-19 20:23:59"}, {"id":2,"cantidad":250,"fecha":"2022-02-11 20:25:23"}, {"id":3,"cantidad":120,"fecha":"2022-03-09 20:25:49"}, {"id":4,"cantidad":320,"fecha":"2022-04-22 20:26:09"}, {"id":5,"cantidad":219,"fecha":"2022-05-11 20:26:27"}]
```

Anterior Siguiente

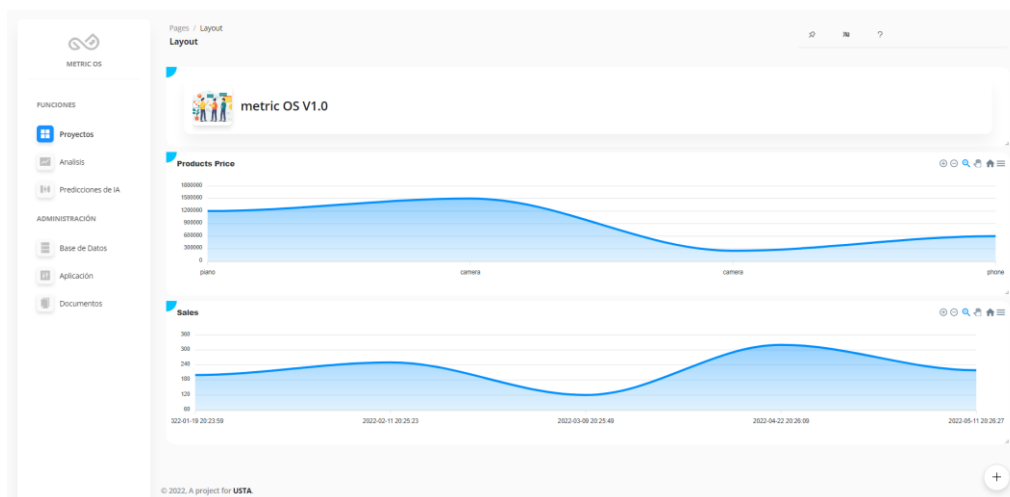
La sección final cuenta con la funcionalidad de adición de título para el grafico y la selección de etiquetas horizontales y verticales con la información obtenida en la consulta previa.

Figura 17. Configuración de características del gráfico

The screenshot shows a configuration window titled "METRIC OS" with a close button (X) in the top right corner. Below the title bar, there are three steps in a sequence: "Tipo de Gráfico" (checked), "Selección de datos" (checked), and "Propiedades del componente" (active). The main area is titled "SELECCIONE LAS PROPIEDADES DEL GRÁFICO". It contains a form with the following fields: "Gráfico del Gráfico" (set to "Ventas mensuales"), "Etiquetas horizontales" (set to "fecha"), "Valores verticales" (set to "cantidad"), and a dropdown menu for "cantidad". At the bottom right, there are two buttons: "Anterior" and "Guardar".

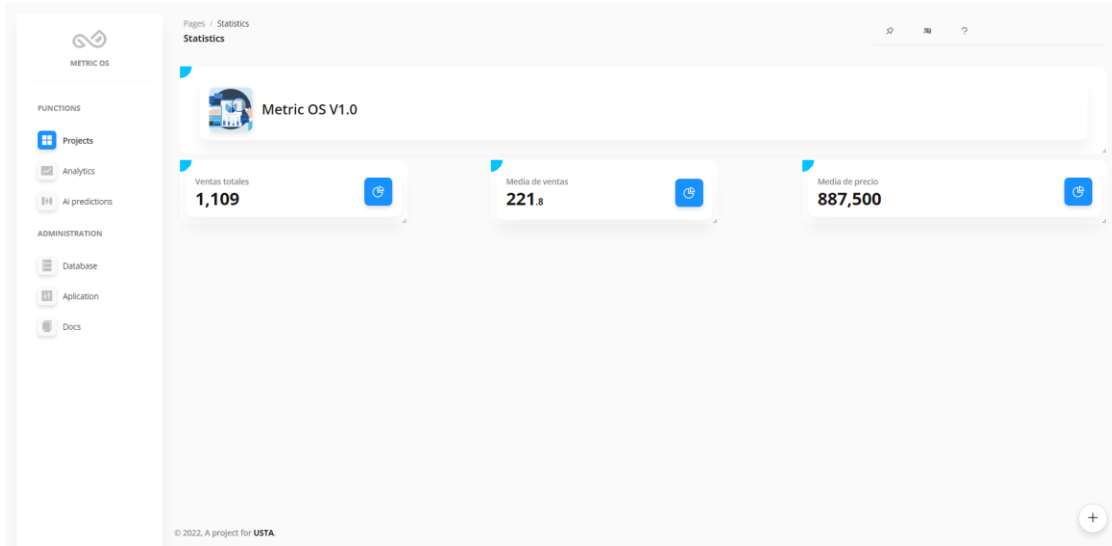
Una vez seleccionada la opción *Guardar* el panel de visualización se actualizará incluyendo el nuevo grafico creado.

Figura 18. Visualización del gráfico creado



Los proyectos dentro de la sección de *análisis* permiten acceder a sus paneles de visualización respectivamente donde se encontrarán los análisis estadísticos realizados por el usuario.

Figura 19. *Análisis estadísticos realizados*



La creación de un nuevo análisis se realiza mediante el botón  el cual desplegará el formulario de creación para el usuario de forma similar a la creación de un gráfico en el panel de visualización de métricas.

El usuario inicialmente seleccionará el tipo de análisis que desea realizar, en esta sección tendrá habilitados ocho tipos dentro de los cuales se encuentran operaciones básicas como la suma, producto, máximo y mínimo, así como operaciones para hallar la media, mediana, varianza y desviación estándar.

Figura 20. Selección de tipo de análisis estadístico

METRIC OS

1 Statistic type 2 Data selection 3 Component properties

SELECCIONAR TIPO DE ANÁLISIS

Mínimo	Máximo	Suma	Producto
Seleccionar	Seleccionar	Seleccionar	Seleccionar
Media	Mediana	Varianza	Desviación estandar
Seleccionar	Seleccionar	Seleccionar	Seleccionar

Siguiente

La creación de la consulta que obtendrá la información de la base de datos tiene el mismo procedimiento que en la creación de gráficos permitiendo que el usuario relacione estas funcionalidades con patrones específicos para permitir que el aplicativo sea intuitivo en todo momento.

Figura 21. Consulta a la base de datos en el dashboard de analítica

METRIC OS

1 Statistic type 2 Data selection 3 Component properties

INSERTAR LA CONSULTA DE DATOS

Seleccionar	*	Desde	Ventas	Donde	Condición
Tiene	Condición	Agrupar por	Condición	Ordenar por	Condición

Prueba

ASÍ ES COMO SE VEN SUS DATOS:

```
[{"id":1,"cantidad":200,"fecha":"2022-01-19 20:23:59"}, {"id":2,"cantidad":250,"fecha":"2022-02-11 20:25:23"}, {"id":3,"cantidad":120,"fecha":"2022-03-09 20:25:49"}, {"id":4,"cantidad":320,"fecha":"2022-04-22 20:26:09"}, {"id":5,"cantidad":219,"fecha":"2022-05-11 20:26:27"}]
```

Anterior Siguiente

La sección final contempla el título del análisis, así como las propiedades de visualización. Cada análisis cuenta con la funcionalidad de contener un texto que preceda al valor (prefijo), el campo de la información que corresponde al valor y, un campo de texto posterior (sufijo), facilitando la lectura de la información obtenida de la base de datos.

Figura 22. *Propiedades de la visualización*

METRIC OS

✓ Statistic type — ✓ Data selection — ● Component properties

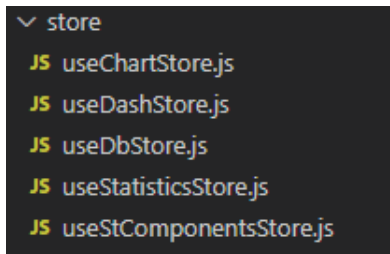
SELECCIONAR LAS PROPIEDADES DEL COMPONENTE

Component title	Ventas totales					
Prefijo	\$	Datos	cantidad	cantidad ▾	Sufijo	USD

Anterior Guardar

3.2.2 Implementación de la mutación de estados

Teniendo en cuenta la necesidad de mantener una comunicación persistente entre vistas y componentes, se implementó Pinia como librería de VueJS para la administración de estados y mutaciones entre componentes. Las *Stores* creadas corresponden a cada una de las vistas principales del aplicativo (Panel de visualización de métricas, analíticas y gestor de bases de datos) y a sus componentes (gráficos) como se presenta a continuación:

Figura 23. Stores de almacenamiento de estados globales

Internamente cada *Store* cuenta con dos secciones fundamentales para la administración de la comunicación; La primera se conoce como *Estado*, donde se almacenarán variables que afectan a cada componente de su tipo, tomando como ejemplo la *Store* de los gráficos, también denominada *ChartStore* (El termino *Use* se agrega por convención y no es relevante) observamos como en su sección de estado cuenta con información relacionada a su tipo, propiedades y la consulta del usuario que obtiene los valores a mostrar, de la misma forma, contiene información sobre su ubicación y dimensiones dentro de la cuadrícula lógica que se encarga de organizar cada uno de los gráficos del panel de visualización.

Figura 24. Estado global del componente

```
state: () => ({
  isModalOpen: false,
  currentStep: 0,
  chartData: {
    type: "line",
    zoom: false,
    title: "",
    xaxis: "",
    yaxis: "",
    dataQuery: "",
  },
  newLayout: {
    x: null,
    y: null,
    w: null,
    h: null,
    i: null,
    parent: null,
  },
  layout: [],
})
```

La segunda sección de la *Store* se denomina *Acciones* y, como su nombre lo indica, contendrá toda la lógica del componente para su interacción con el proyecto backend [14] el cual es el encargado de administrar y preservar en memoria las diferentes configuraciones que se realicen sobre el aplicativo y sus componentes. Siguiendo con el ejemplo anterior para la *ChartStore* encontramos en su sección de acciones funciones relacionadas a la adquisición y formato de la información.

Figura 25. Sección de "acciones" del store

```
async readComponents(parentId) {  
    var formdata = new FormData();  
    formdata.append("dashboardId", parentId);  
  
    var requestOptions = {  
        method: "POST",  
        body: formdata,  
    };  
  
    await fetch(  
        READ_COMPONENTS_ENDPOINT,  
        requestOptions  
    )  
    .then((response) => response.text())  
    .then((result) => (this.layout = JSON.parse(result)))  
    .then(() => {  
        let length = this.layout.length;  
        this.newLayout = {  
            x: 0,  
            y: Math.max(...this.layout.map((component) => component.y)),  
            w: 7,  
            h: 7,  
            i: this.layout[length - 1].i + 1,  
            parent: parentId,  
        };  
    })  
    .then(() => this.readComponentsData())  
    .catch((error) => console.log("error", error));  
}
```

3.3 Desarrollo del sitio web

3.3.1 Desarrollo del sitio web del aplicativo

La visibilidad del aplicativo al público es un factor relevante teniendo en cuenta que su continua mejora está basada en la creación de una comunidad de desarrolladores y usuarios que aporten tanto ideas como módulos y mejoras al proyecto base.

Se desarrolló un sitio web que integrara la descripción del aplicativo, su propósito y los accesos directos para acceder a cada una de las partes de código fuente, así como de su documentación pertinente.

Teniendo en cuenta que VueJS es empleado tanto en proyectos de gran escala como en proyectos sencillos fue seleccionado para el desarrollo de esta página web de presentación mediante la integración de una vista principal conformada de los diferentes componentes que en este caso no tienen como objetivo ser reutilizados sino permitir la lectura y mejorar la organización.

Figura 25. Componentes de la vista principal en VueJS

```
<div>
  <!-- Start Features Component -->
  <features-area />
  <!-- End Features Component -->

  <!-- Start About Component -->
  <about-area />
  <!-- End About Component -->

  <!-- Purpose Component Start -->
  <purpose-area />
  <!-- Purpose Component End -->

  <!-- Product Component Start -->
  <product-area />
  <!-- Product Component End -->
</div>
```

Figura 26. *Página Web del proyecto*

3.4 Puesta en operación del proyecto

3.4.1 Migración de la plataforma a una aplicación híbrida

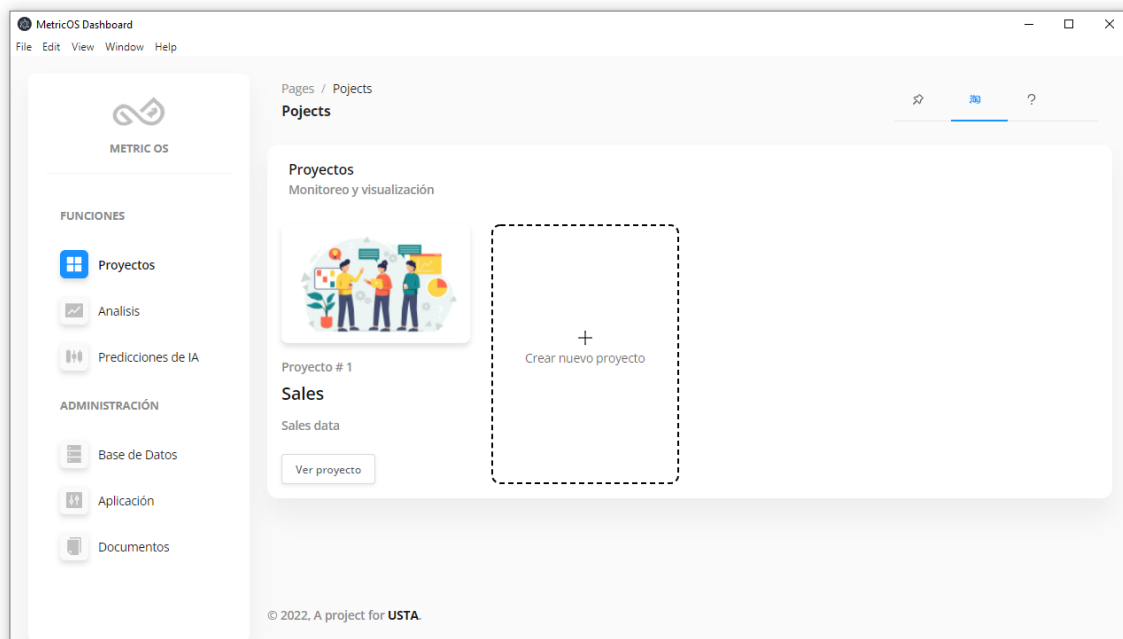
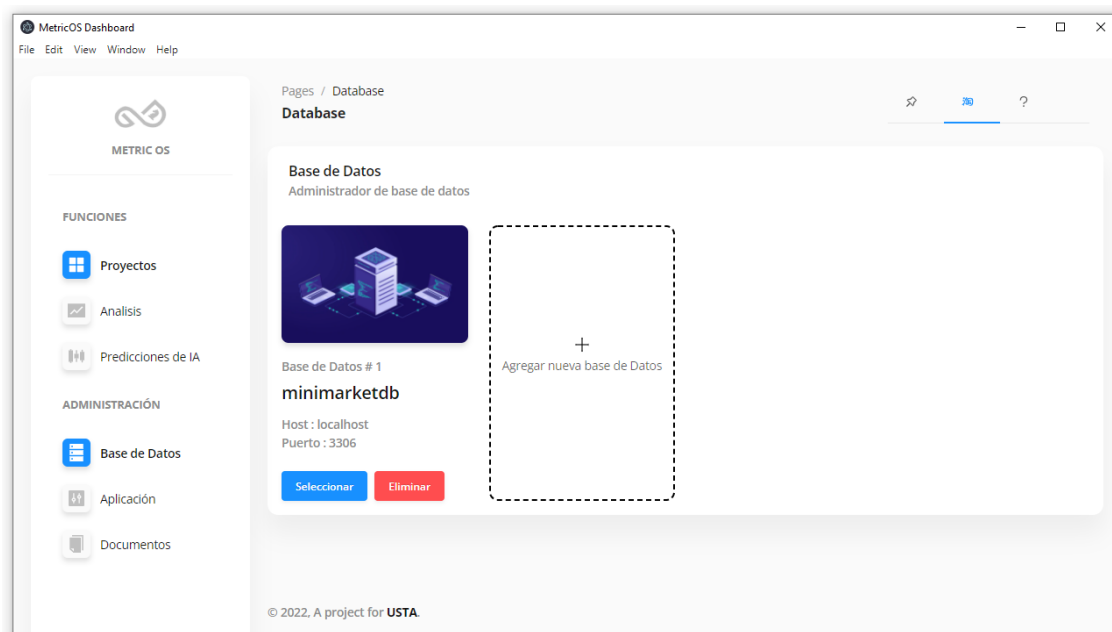
La meta del aplicativo frontend es poder ser ejecutado de manera local por los usuarios sin necesidad de depender de recursos externos como lo pueden ser servicios de alojamiento (físicos o en la nube) y plataformas de hosting y dominio aumentando los costos de integración e implementación.

Electron [15] es un framework de JavaScript de código abierto desarrollado para migrar o construir aplicaciones nativas de escritorio (Windows, Mac y Linux) usando tecnologías de desarrollo web como lo son VueJS. Electron funciona mediante la implementación de Chromium y NodeJS para compilar el código como un software independiente de escritorio.

VueJS CLI implementa la configuración base de Electron para migrar un proyecto existente mediante su paquete “electron-vue” el cual es instalado mediante NPM. Configuraciones relativas al aplicativo son realizadas dentro el archivo “background” de JavaScript dentro del cual se establecen las directivas de visualización y funcionalidades de la ventana de escritorio.

Figura 27. Configuración ventana de escritorio (Electron)

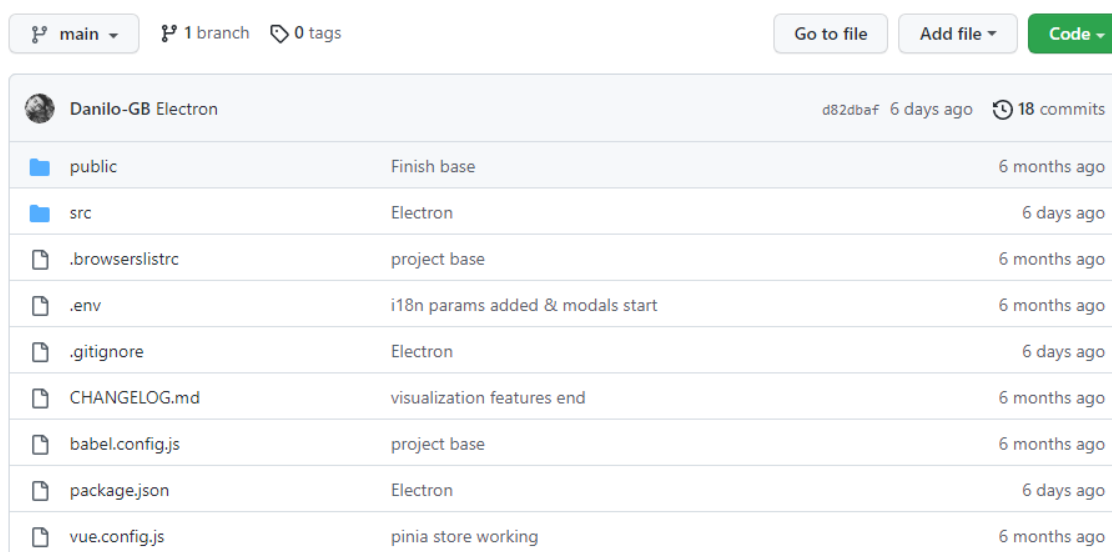
```
const win = new BrowserWindow({  
  width: 1280,  
  height: 720,  
  webPreferences: {  
    nodeIntegration: process.env.ELECTRON_NODE_INTEGRATION,  
    contextIsolation: !process.env.ELECTRON_NODE_INTEGRATION,  
  },  
});
```

Figura 28. *Vista de Proyectos (Versión de Escritorio)***Figura 29.** *Vista de Base de Datos (Versión de Escritorio)*

3.4.2 Despliegue del software

El repositorio del proyecto fue creado en la plataforma de GitHub mediante el versionado de código Git, de esta forma, los desarrolladores accederán directamente al código fuente del aplicativo y los usuarios finales tendrán acceso a la carpeta con el ejecutable de escritorio si lo que desean utilizar es el proyecto base, ambos accesos pueden ser encontrados en la página oficial del aplicativo dispuesta al público mediante la dirección web: metric-os.netlify.app.

Figura 30. Repositorio GitHub



File/Folder	Description	Last Commit
public	Finish base	6 months ago
src	Electron	6 days ago
.browserslistrc	project base	6 months ago
.env	i18n params added & modals start	6 months ago
.gitignore	Electron	6 days ago
CHANGELOG.md	visualization features end	6 months ago
babel.config.js	project base	6 months ago
package.json	Electron	6 days ago
vue.config.js	pinia store working	6 months ago

3.4.3 Liberación del código fuente

Establecido el código fuente en su respectivo repositorio y asignado como de acceso público, se hace necesaria la asignación de una respectiva licencia de uso que mantenga la misión y visión intrínseca del proyecto la cual lo puede resumir como un proyecto de uso libre y gratuito que sirva como base a desarrolladores y usuarios. Se opta por la licencia “Mozilla Public License 2.0” [16] la cual se encuentra englobada en el tipo de licencias Copyleft cuyo propósito es mantener gratuito y modificable el uso del proyecto. La licencia empleada también permite que

los usuarios que generen grandes cambios al proyecto, como lo podrían ser la adición de módulos específicos de análisis y la creación de gráficos personalizados, con el fin de que el proyecto base se siga manteniendo como público pero que los usuarios puedan reclamar una versión editable como propia, permitiendo a las compañías mantener su identidad y adaptar el proyecto como un producto interno.

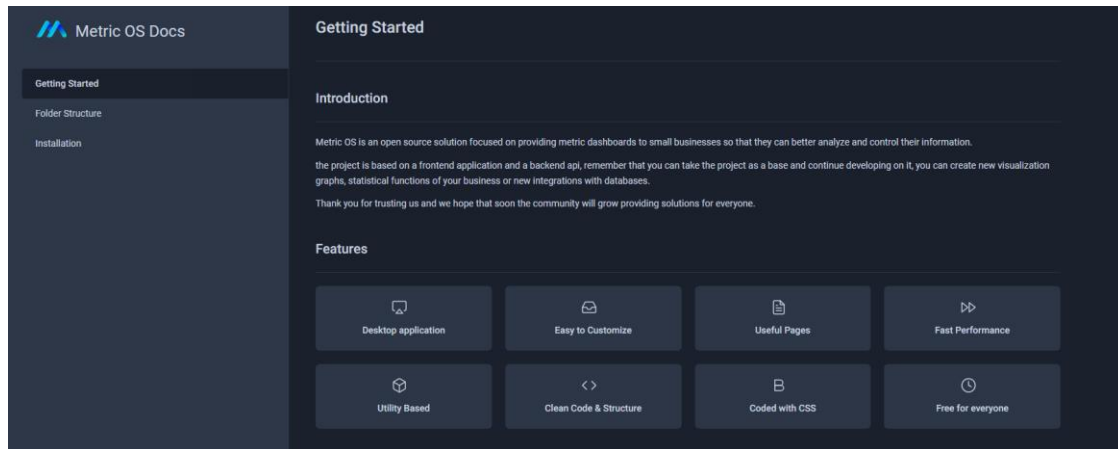
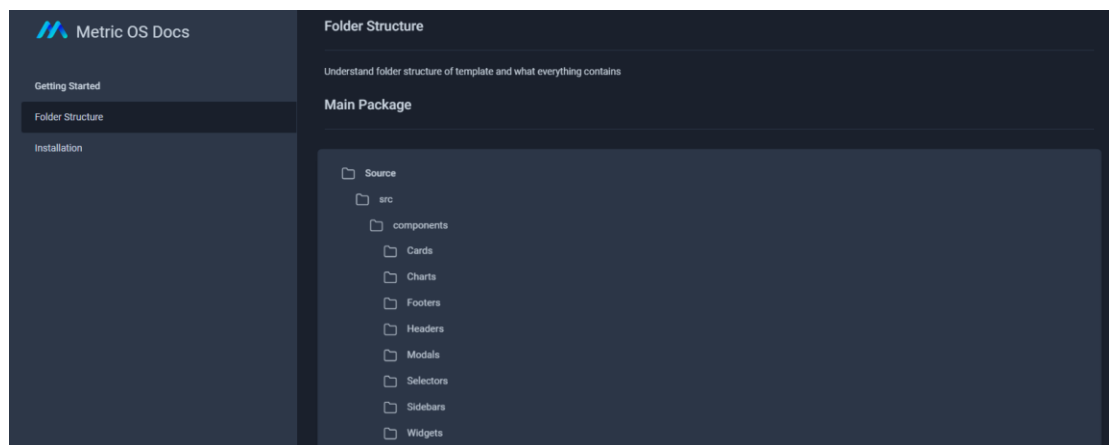
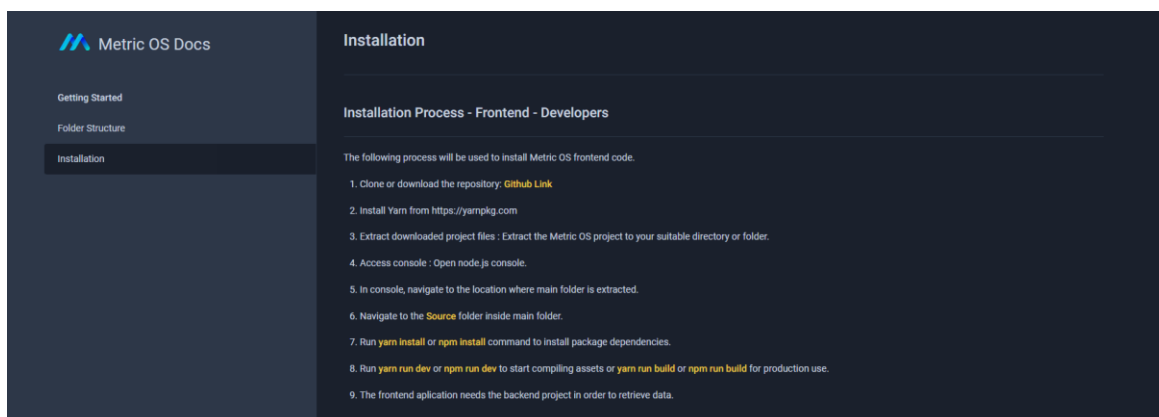
Figura 31. *Mozilla Public License 2.0*

 Danilo-GB/MetricOS is licensed under the Mozilla Public License 2.0 Permissions of this weak copyleft license are conditioned on making available source code of licensed files and modifications of those files under the same license (or in certain cases, one of the GNU licenses). Copyright and license notices must be preserved. Contributors provide an express grant of patent rights. However, a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.	Permissions ✓ Commercial use ✓ Modification ✓ Distribution ✓ Patent use ✓ Private use	Limitations ✗ Liability ✗ Trademark use ✗ Warranty	Conditions ⓘ Disclose source ⓘ License and copyright notice ⓘ Same license (file)
This is not legal advice. Learn more about repository licenses.			

3.4.4 Definición del manual de usuario

Debido a que el proyecto se encuentra enfocado tanto a desarrolladores como a usuarios finales sin experiencia en programación, se genera en VueJS un sitio web que contenga las funcionalidades principales, tecnologías utilizadas, distribución y jerarquía de carpetas y las instrucciones necesarias para la descarga e instalación tanto del proyecto frontend como de su aplicativo de escritorio. El manual de usuario se encuentra servido al público a través del sitio web:

metric-os-docs.netlify.app

Figura 27. *Manual de usuario (Guía de inicio)***Figura 28.** *Manual de usuario (Estructura de carpetas)***Figura 29.** *Manual de usuario (Instalación)*

4. Conclusiones

El aplicativo desarrollado cumple con las políticas de diseño planteadas inicialmente, manteniendo una gama de patrones y estructuras a su vez, que permite su intuitiva manipulación reduciendo las interacciones de los usuarios para una misma funcionalidad. Por lo tanto, el usuario al realizar una consulta obtiene la información requerida de la base de datos y esta información se genera en relación con la creación de los gráficos, permitiendo en todo momento que el aplicativo sea intuitivo.

Es importante mencionar que el aplicativo se creó con el propósito, a largo plazo, de poder ser ejecutado por usuarios de manera local sin necesidad de depender de alojamientos físicos o en la nube.

La estructura del aplicativo fue diseñada con el fin de que cada componente gráfico tuviera la posibilidad de ser redimensionado y desplazado a su vez, dentro del panel de visualización, esto para que el usuario tenga mejor compresión visual y se apropie totalmente de la información presentada.

La experiencia de trabajar mediante el framework de JavaScript VueJS fue enriquecedora y muy importante para el desarrollo del software, ya que ofrece variedad de herramientas y librerías, las cuales facilitaron la creación del aplicativo, donde VueJS dio la posibilidad de elegir el lenguaje de diseño y, en consecuencia, se puede adaptar el código de manera libre, permitiendo crear una identidad única para cada desarrollo de software.

5. Trabajo a futuro

El proyecto Frontend desarrollado presenta una base sólida tanto para el usuario final como para desarrolladores, como trabajo a futuro se plantea la estandarización de las funcionalidades desarrolladas en la mutación de estados encargadas de su interacción con el proyecto Backend [14] para así permitir a los usuarios, de manera visual, establecer las configuraciones y parámetros para establecer la conexión con diferentes APIs puesto que, actualmente, se hace necesario del proyecto de Backend para el uso completo del aplicativo Frontend

Referencias

- [1] Franco Ángel, M., & Urbano, D. (2019). Caracterización de las pymes colombianas y de sus fundadores: Un Análisis Desde dos Regiones del País. *Estudios Gerenciales*, 81–91. <https://doi.org/10.18046/j.estger.2019.150.2968>
- [2] M. E. L. P. Rosario de Fátima Suárez Améndola, «Bases de Datos en las Pequeñas y Medianas Empresas,» 2012.
- [3] “Dashboard Empresarial. Qué Es y Cómo funciona,” *tudashboard*, 10-Jun-2021. [Online]. Available: <https://tudashboard.com/dashboard-empresarial/>. [Accessed: 27-Mar-2022].
- [4] “Vue.js - the progressive JavaScript framework: Vue.js,” Vue.js - The Progressive JavaScript Framework | Vue.js. [Online]. Available: <https://vuejs.org/>. [Accessed: 27-Mar-2022].
- [5] T. Ambler and N. Cloud, “JavaScript frameworks for modern web dev,” 2015.
- [6] A. Fiechter, R. Minelli, C. Nagy and M. Lanza, "Visualizing GitHub Issues," 2021 Working Conference on Software Visualization (VISOFT), 2021, pp. 155-159, doi: 10.1109/VISOFT52517.2021.00030.
- [7] Oussama Alouat, Herramienta de modelado y procesamiento gráfico de datos. (2018).
- [8] F. S. Foundation, «El sistema operativo GNU,» [En línea]. Available <https://www.gnu.org/philosophy/free-sw.es.html>. K. Elissa, “Title of paper if known,” unpublished.
- [9] “How do business intelligence, business analytics and enterprise performance management fit together?,” *Strategic Business Management*, pp. 127–130, 2017.
- [10] “Fundamental design principles for dashboards: Idashboards software,” *iDashboards*, 10Mar2022. [Online]. Available:

<https://www.idashboards.com/?guides=fundamentaldesignprinciples-for-dashboards%2F>.
[Accessed: 27-Mar-2022].

- [11] «Vue.js,» 2022. [En línea]. Available: <https://vuejs.org/>.
- [12] «Ant Design Vue,» 2022. [En línea]. Available: <https://antdv.com/components/overview>.
- [13] «Pinia,» 2022. [En línea]. Available: <https://pinia.vuejs.org/>.
- [14] J.D. Arias Estupiñán, “Diseño e implementación de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y mediana empresas”, Tesis Pregrado, Universidad Santo Tomás, Bucaramanga, Colombia, 2022.
- [15] O. J. Foundation, «Electron,» 2022. [En línea]. Available: <https://www.electronjs.org/es/docs/latest>.
- [16] Mozilla, «Mozilla Public License,» 2022. [En línea]. Available: <https://www.mozilla.org/en-US/MPL/2.0/>.