

Componente 1. Entregable 1.6.8 Ministerio de Minas y Energía

Proyecto: CTO No. 116231-225-2024

Herramienta de generación de informes tipo dashboard.
Según necesidades definidas por el MME

Fecha: julio de 2025
Elaborado por, ADA S.A.S.
Versión: 7.0

Contenido

1. Introducción	4
2. Objetivo	4
3. Arquitectura Funcional y Tecnológica del Aplicativo web	4
3.1. Acceso y Consulta de Datos	4
3.1.1 Tecnologías empleadas en la interfaz (frontend)	5
3.1.2 Tecnologías del backend e infraestructura de datos.....	5
3.1.3 Detalle control de Versionamiento herramientas usadas para el aplicativo web	6
3.2 Aceptación y Acceso.....	6
4. Funcionalidades principales.....	7
4.1 Acceso y Consulta de Datos	7
4.1.1 Visualización tabular	7
4.1.2 Representación geográfica.....	7
4.1.3 Intercambio de vistas	7
4.1.4 Búsqueda avanzada con tecnología AJAX.....	7
4.1.5 Selección dinámica con menú desplegable tipo acordeón.....	8
4.1.6 Rendimiento optimizado	8
4.1.7 Escalabilidad e implementación	8
4.2 Representación gráfica	8
4.2.1 Visualización de mapas interactivos.....	9
4.3 Capas	9
4.3.1 Capa Lógica (Backend).....	9
4.3.2 Capa de Datos	9
4.3 Descarga y exportación de datos	9
4.4 Tecnologías recomendadas	10
4.5 Características destacadas	10
4.5.1 Tooltip Visualización contextual.....	10
4.5.2 Exportación flexible	10
4.5.3 Tecnologías que se proponen para esta funcionalidad.....	11
4.6 Generación de Informes y Exportación	11
4.6.1 Ejemplo de uso	12
4.6.2 Creación de informes dinámicos en PDF utilizando ReportLab	12
4.6.3 Ejemplo de uso	12
4.6.4 Metadatos incorporados en el informe (fecha, fuente, filtros)	13
4.6.5 Ejemplo de metadatos en un archivo PDF	13
4.6.6 Creación de gráficos en informes mediante Matplotlib	14
5. Requerimientos Técnicos.....	14

5.1 Infraestructura.....	14
5.1.1 Interfaz de Usuario.....	15
5.1.2 Backend (Sistema Interno).....	15
5.1.3 Base de Datos.....	15
5.1.4 Infraestructura en la Nube.....	16
5.2. Procesamiento de Datos.....	16
5.3 Código Fuente del Desarrollo.....	17
5.4 Perfil del profesional idóneo para realizar despliegues del aplicativo.....	17
6. Criterios de Aceptación.....	17
7. Aprobación.....	17

Índice de tablas

Tabla 1. Detalle Versionamiento de las herramientas usadas en el desarrollo del aplicativo.....	6
---	---

Índice de ilustraciones

Ilustración 1. Tooltip determinantes. Fuente: ADA.....	10
Ilustración 2. Metadatos incorporados en el informe (fecha, fuente, filtros). Fuente: ADA.....	14
Ilustración 3. Infraestructura. Fuente: ADA.....	16

1. Introducción

El documento que se presenta a continuación tiene como finalidad describir de forma detallada y comprensible, el procedimiento que guía la elaboración de informes para el Ministerio de Minas y Energía (MME). Esta labor no se limita únicamente a la generación de reportes; va más allá, ya que constituye el pilar fundamental para la creación de tableros de control interactivos (dashboards), herramientas que se convierten en el principal punto de acceso a la información clave del sector minero-energético colombiano.

Estos tableros permiten visualizar, de manera clara y estructurada, los principales indicadores y determinantes del sector, facilitando, el seguimiento de políticas públicas y la evaluación de resultados. En este sentido, el aplicativo web no solo cumple una función técnica, sino que se posiciona como un puente entre los datos y las personas usuarias, asegurando que la información esté siempre disponible, sea fácilmente accesible y conserve altos estándares de confiabilidad.

2. Objetivo

El propósito central de este proyecto es desarrollar un aplicativo web, avanzado y robusto, que permita consultar datos de manera intuitiva y rápida. Esta plataforma tiene la capacidad de generar informes en formatos comúnmente utilizados, como PDF y Excel, lo que facilita su lectura, archivo y distribución.

Además, se contempla la incorporación de mapas interactivos y tablas dinámicas que reflejen la información territorial, permitiendo una visualización geoespacial que enriquece la comprensión de los datos y aporta una capa adicional de análisis.

Desde su concepción, el sistema está pensado para cumplir con los principios fundamentales del desarrollo digital contemporáneo: la usabilidad, para que cualquier persona pueda navegarlo sin dificultad y la escalabilidad, para que crezca al ritmo de las necesidades del sector.

3. Arquitectura Funcional y Tecnológica del Aplicativo web

3.1. Acceso y Consulta de Datos

En este aplicativo web se encuentra una arquitectura tecnológica moderna, diseñada para responder a las exigencias actuales en términos de rendimiento, flexibilidad y experiencia de usuario. Esta sección expone los principales componentes técnicos que conforman el sistema, divididos en frontend e infraestructura de backend.

3.1.1 Tecnologías empleadas en la interfaz (frontend)

- **React.js:** Se utiliza esta potente librería de JavaScript para construir una interfaz dinámica basada en componentes reutilizables. Gracias a su enfoque en aplicaciones de una sola página (SPA), la navegación se percibe fluida, sin interrupciones, con tiempos de carga mínimos y una experiencia de usuario muy superior a las páginas web tradicionales.
- **Tailwind CSS:** Este framework de estilos permite crear interfaces limpias, modernas y adaptables. Facilitan una personalización eficiente del diseño, manteniendo la consistencia visual y una estética profesional que se adapta a diferentes resoluciones y dispositivos.
- **Axios:** Esta herramienta se encarga de realizar peticiones HTTP asincrónicas al backend, lo que permite obtener y enviar datos de forma rápida y sin necesidad de recargar la página.
- **React Router:** Gracias a este componente, la navegación entre diferentes secciones del aplicativo web público se realiza de forma natural, sin interrupciones visuales ni cargas completas de la página, lo que mejora la fluidez general del sistema.
- **Leaflet.js:** Esta librería ligera permite incorporar mapas interactivos directamente en la interfaz, ofreciendo al usuario la posibilidad de explorar la información territorial de forma visual, intuitiva y enriquecedora.

3.1.2 Tecnologías del backend e infraestructura de datos

- **Django:** Este robusto framework de desarrollo web basado en Python permite construir una API segura, escalable y mantenible. Su arquitectura modular acelera el desarrollo y garantiza la estabilidad del sistema.
- **Django REST Framework (DRF):** Extensión de Django que facilita la creación de APIs RESTful. Actúa como intermediario entre el frontend y la base de datos, asegurando una comunicación eficiente y segura.
- **PostgreSQL con extensión PostGIS:** La base de datos relacional elegida incluye funcionalidades espaciales que permiten almacenar y consultar datos geográficos con precisión. Esto resulta fundamental para la representación territorial de los indicadores del sector.
- **Redis + Django Cache Framework:** Se implementa un sistema de almacenamiento en memoria para agilizar las consultas más frecuentes, mejorando el rendimiento general del sistema y reduciendo los tiempos de respuesta.
- **Celery + Redis (opcional):** Este conjunto permite ejecutar tareas asíncronas, como actualizaciones masivas de datos, sin afectar el desempeño general del sistema, lo que resulta especialmente útil en procesos de alto volumen.

3.1.3 Detalle control de Versionamiento herramientas usadas para el aplicativo web

Nombre del software	Versión	Playwright	Cypress	Proveedor	Fecha de instalación	Entorno
Cliente: Docker Engine	28.1.1	N/A	N/A	Docker Inc.	18-abr-25	Pruebas MME
Servidor: Docker Engine	28.0.4	N/A	N/A	Docker Inc.	25-mar-25	Pruebas MME
Python	3.11 slim	1.43.0	N/A	Python software Foundation	15-may-25	Pruebas MME
React	18.3.1	N/A	N/A	Meta Platforms, Inc	15-may-25	Pruebas MME
PostgreSQL	16.8	N/A	N/A	PostgreSQL Global Development Group	15-may-25	Pruebas MME
PostGIS	15 - 3.3	N/A	N/A	Refractions Research	15-may-25	Pruebas MME

Tabla 1. Detalle Versionamiento de las herramientas usadas en el desarrollo del aplicativo.

3.2 Aceptación y Acceso

El aplicativo web está diseñado pensando en la diversidad de entornos digitales que utilizan hoy las personas. Por eso, se asegura de ofrecer una experiencia coherente y funcional sin importar el navegador web que se utilice. Es plenamente compatible con los principales navegadores modernos como Google Chrome, Mozilla Firefox, Safari y Microsoft Edge, garantizando así que el acceso al sistema no se vea limitado por la herramienta de navegación preferida del usuario. Esta compatibilidad se mantiene actualizada conforme evolucionan los navegadores, lo que permite que la plataforma continúe siendo accesible y confiable en el tiempo.

Además, el diseño de la interfaz se construye desde una perspectiva adaptable. Siguiendo principios de desarrollo responsivo, y aplicando estrategias como Mobile First junto con el uso de Media Queries, el aplicativo se ajusta de forma automática a cualquier tamaño de pantalla. Esto significa que su funcionalidad y presentación se conservan tanto en pantallas grandes de escritorio como en tablets y teléfonos móviles, ofreciendo siempre una experiencia clara, ágil y optimizada. La adaptabilidad no solo se limita al aspecto visual, sino que también contempla la usabilidad: los elementos interactivos y los flujos de navegación están diseñados para ser intuitivos en cada dispositivo, sin sacrificar rendimiento ni accesibilidad.

De esta manera, se garantiza que cualquier ciudadano o funcionario pueda acceder al aplicativo web desde el entorno digital que le resulte más cómodo, sin fricciones técnicas y con una experiencia uniforme y eficiente.

4. Funcionalidades principales

4.1 Acceso y Consulta de Datos

El aplicativo web ofrece al usuario un conjunto robusto de herramientas diseñadas para facilitar la exploración, el análisis y la comprensión de los datos. A través de una interfaz amigable e intuitiva, es posible acceder a información de manera eficiente, visualizarla desde distintos enfoques y ajustarla según necesidades específicas, todo sin perder fluidez en la experiencia de navegación.

4.1.1 Visualización tabular

Los datos se presentan inicialmente en tablas interactivas, organizadas de forma clara y ordenada. Estas tablas permiten ordenar la información por columnas, aplicar filtros personalizados y exportar los resultados a formato XLSX. Esta funcionalidad está pensada para quienes necesitan manipular los datos en herramientas externas o simplemente desean conservar un respaldo local de la información consultada.

4.1.2 Representación geográfica

Además de la vista tabular, el sistema incorpora mapas dinámicos basados en la biblioteca Leaflet. Sobre estos mapas se despliega la información de forma visual y georreferenciada, lo que permite al usuario explorar los datos en su dimensión espacial. El mapa es completamente interactivo: se puede hacer zoom, activar o desactivar capas temáticas, y visualizar marcadores que, al ser seleccionados, despliegan ventanas emergentes con información más detallada y contextual. Esta función es especialmente útil cuando se requiere identificar patrones geográficos o analizar impactos en el territorio.

4.1.3 Intercambio de vistas

El usuario puede alternar entre la visualización tabular y la representación geográfica de los datos en cualquier momento. Esta posibilidad de cambiar de vista según las necesidades del análisis o la preferencia personal del usuario aporta una gran flexibilidad y enriquece la experiencia de consulta, favoreciendo una comprensión más profunda de la información disponible.

4.1.4 Búsqueda avanzada con tecnología AJAX

El sistema incorpora un potente motor de búsqueda que permite realizar consultas sin necesidad de recargar la página, gracias a la integración de tecnologías asincrónicas (AJAX). A través de filtros por ubicación de zona y categorías como social, ambiental, sectorial o de infraestructura, el usuario puede acotar los resultados y encontrar con precisión los datos que necesita. Adicional, se implementan funciones de autocompletado

del panel de filtrado con tan solo dar un clic en el mapa, lo que enriquece aún más la experiencia de búsqueda.

4.1.5 Selección dinámica con menú desplegable tipo acordeón

Para facilitar la navegación entre categorías jerárquicas, se ofrece un menú desplegable tipo acordeón, ubicado en una barra lateral o en la parte superior de la interfaz. Este menú permite al usuario explorar y seleccionar filtros o capas de información de manera progresiva, construyendo así sus propias vistas personalizadas. El diseño se enfoca en la simplicidad y en guiar al usuario de forma natural en la construcción de sus análisis.

4.1.6 Rendimiento optimizado

Con el objetivo de ofrecer una experiencia fluida incluso ante grandes volúmenes de datos, el sistema incorpora mecanismos de mejora del rendimiento:

- **Caché de consultas:** Se almacena temporalmente la información de búsquedas frecuentes mediante la caché interna de Django, evitando así realizar consultas repetitivas a la base de datos. Esto reduce los tiempos de espera y mejora la capacidad de respuesta del sistema.
- **Carga diferida (lazy loading):** Se implementa una estrategia de carga progresiva de componentes y datos. Es decir, solo se cargan aquellos elementos que el usuario solicita o necesita ver, lo cual disminuye significativamente el tiempo de carga inicial del aplicativo web público.

4.1.7 Escalabilidad e implementación

El aplicativo web está construido sobre una arquitectura modular que permite escalar y mantener cada componente de forma independiente:

- **Separación entre frontend y backend:** El desarrollo del frontend (HTML, CSS, JavaScript) está claramente separado del backend (Django, PostgreSQL con extensiones como PostGIS). Esta división permite actualizar o escalar cada parte de forma aislada, sin afectar el funcionamiento general del sistema.
- **Uso de contenedores Docker (opcional):** Para facilitar la implementación del sistema en distintos entornos (desarrollo, pruebas, producción), se contempla el uso de contenedores Docker, que encapsulan todas las dependencias necesarias y garantizan consistencia entre entornos.

4.2 Representación gráfica

La dimensión visual toma un papel central en este aplicativo web, especialmente en lo que respecta a la representación geográfica de los datos. A través de mapas interactivos, el

sistema permite explorar el territorio de manera intuitiva, cruzando información de diversas fuentes y capas superpuestas.

4.2.1 Visualización de mapas interactivos

El usuario cuenta con un visor geográfico totalmente interactivo, donde puede desplazarse, hacer zoom o seleccionar zonas de interés. Este mapa actúa como una herramienta de análisis espacial que permite observar no solo la localización de infraestructuras o eventos, sino también las relaciones entre distintas variables geográficas.

4.3 Capas

4.3.1 Capa Lógica (Backend)

Este componente se encarga de procesar las reglas del negocio, administrar la lógica de la aplicación y coordinar las operaciones entre la base de datos y el frontend:

- **FastAPI (Python):** Framework ligero y altamente eficiente para el desarrollo de APIs RESTful.
- **Supabase:** Plataforma de desarrollo backend utilizada para ambientes de pruebas, que integra autenticación, almacenamiento y base de datos.
- **PostgreSQL + PostGIS:** Motor de base de datos relacional con extensiones geoespaciales para manejo avanzado de datos territoriales.

Nota: Para conocer el detalle, ver Anexo 2. Documentación técnica del código Back End.

4.3.2 Capa de Datos

Responsable del almacenamiento seguro y estructurado de toda la información territorial y socioeconómica del aplicativo web:

- **Tecnología empleada:** PostgreSQL con la extensión PostGIS, que permite consultas espaciales optimizadas y una representación detallada del territorio.
- **Funciones clave:**
 - Almacenamiento estructurado de datos temáticos y geográficos.
 - Gestión eficiente de entidades espaciales.
 - Ejecución de consultas geográficas avanzadas a través de filtros como la selección de zonas del país, listas recurrentes, departamentos y municipios, adicionalmente con la posibilidad de realizar comparaciones visualizando hasta dos mapas al mismo tiempo.

4.3 Descarga y exportación de datos

El usuario puede exportar mapas en formato PDF y las tablas en formato Excel.

4.4 Tecnologías recomendadas

- Para la visualización web: **Leaflet**.
- Para el manejo de datos espaciales: **PostGIS** sobre una base de datos **PostgreSQL**.
- Para la publicación de servicios geoespaciales: **GeoServer** o **ArcGIS Server**, mediante estándares WMS y WFS.
- Para la edición y revisión previa: **QGIS** como herramienta complementaria.

4.5 Características destacadas

4.5.1 Tooltip Visualización contextual

Al interactuar con el mapa, si el usuario da clic sobre uno de los polígonos de la consulta realizada, el sistema despliega automáticamente un cuadro emergente (tooltip) que muestra la información de la grilla, como las coordenadas X, Y, la presencia o no de determinantes, y el cálculo de los mismos en valor porcentual en escala de 0 a 100%, entre otros. Esta funcionalidad enriquece la comprensión del dato, ya que el usuario no solo ve un número, sino que accede a un pequeño contexto que le ayuda a interpretarlo mejor.

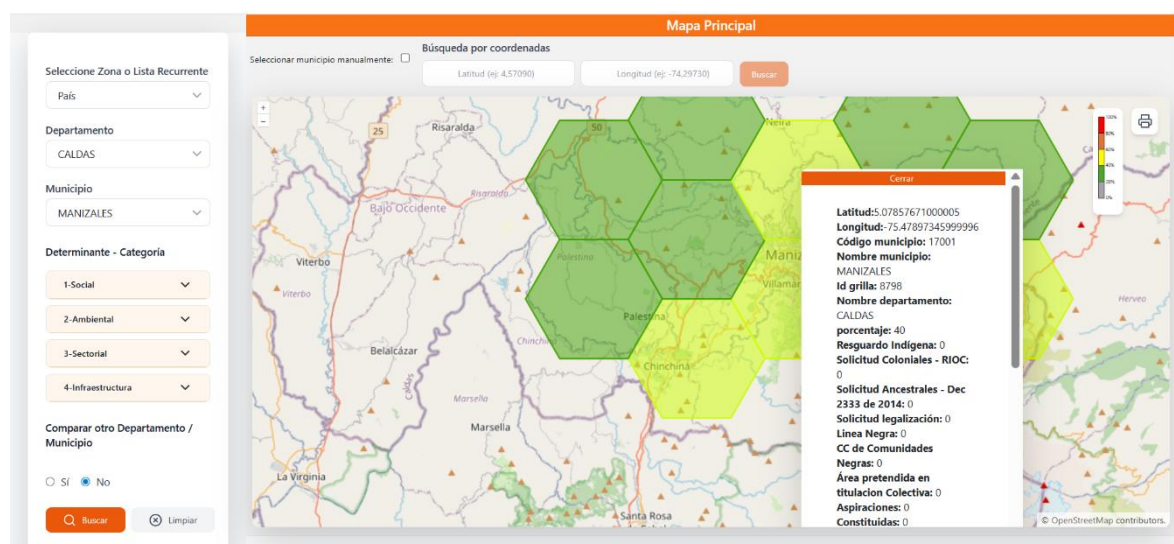


Ilustración 1. Tooltip determinantes. Fuente: ADA.

4.5.2 Exportación flexible

Para quienes necesitan utilizar los resultados del análisis fuera del entorno digital del aplicativo web, la herramienta permite exportar los gráficos generados en formato de imagen (PDF), y descargar los datos en hojas de cálculo (.xlsx), lo que facilita su inclusión en informes, presentaciones o estudios más amplios.

4.5.3 Tecnologías que se proponen para esta funcionalidad

Para la implementación de estos gráficos interactivos y responsivos, se contempla el uso de tecnologías especializadas en visualización de datos. Entre las opciones más robustas y versátiles se incluyen:

- **Chart.js**, que permiten crear gráficos web altamente personalizados y dinámicos, adaptándose fácilmente a los requerimientos específicos de visualización del usuario.
- **Plotly.js**, que ofrecen un abanico más amplio de funcionalidades interactivas, permitiendo personalizaciones avanzadas en cuanto a estilo, interactividad y presentación.
- **Highcharts**, en caso de requerirse una solución con acabados visuales de alta calidad y plantillas prediseñadas, ideal para entornos donde la estética y la experiencia visual del usuario sean prioridades (teniendo en cuenta que esta opción requiere una licencia comercial).

4.6 Generación de Informes y Exportación

Una de las funcionalidades más importantes dentro del aplicativo web público es la posibilidad de generar informes y exportar información en formatos accesibles y reutilizables, como archivos Excel. Esta capacidad permite a los usuarios obtener, con pocos pasos, representaciones claras y estructuradas de los datos que han sido procesados o filtrados previamente dentro del sistema, facilitando así su análisis, seguimiento y toma de decisiones.

Para llevar a cabo esta tarea, se hace uso extensivo de la librería **Pandas**, particularmente de su método `DataFrame.to_excel()`, que permite convertir estructuras de datos complejas en tablas comprensibles. Cada conjunto de datos, una vez almacenado en un `DataFrame`, puede ser exportado directamente a un archivo con extensión `.xlsx`, manteniendo su organización, encabezados y estructura interna.

Sin embargo, no se limita a una exportación básica. La aplicación se integra con herramientas adicionales como **openpyxl**, que permiten personalizar a fondo los documentos generados. Gracias a estas librerías, es posible modificar estilos de celdas, aplicar formatos numéricos o de fecha, agregar colores, bordes o incluso trabajar con múltiples hojas dentro del mismo archivo, lo cual enriquece significativamente la presentación visual del informe. Este nivel de personalización mejora la experiencia de usuario y da un aspecto profesional a los resultados exportados.

La lógica de exportación está profundamente integrada con el flujo de trabajo del aplicativo web público. Los datos que se exportan no son simplemente copias sin contexto, sino que responden al estado actual de la aplicación, a los filtros que ha aplicado el usuario, a los resultados de los cálculos internos o a las consultas realizadas. Esto asegura que cada informe tenga valor y relevancia inmediata, alineado con lo que el usuario necesita visualizar o compartir.

4.6.1 Ejemplo de uso

Un usuario realiza una consulta sobre el municipio de Envigado en el departamento de Antioquia, filtrando por país, departamento y municipio, luego selecciona los determinantes a consultar, el sistema procesa esta información y la presenta en el mapa principal. Luego, con un solo clic, el usuario puede exportar estos datos a un archivo Excel. En ese archivo, el encabezado aparece con los metadatos y debajo de esta, la información de los polígonos cargados con detalles numéricos de los determinantes, mencionado archivo podrá ser modificado de acuerdo a la necesidad del usuario.

4.6.2 Creación de informes dinámicos en PDF utilizando ReportLab

Una de las funcionalidades clave que incorpora el aplicativo web es la capacidad de generar informes en formato PDF de manera completamente dinámica. Esta característica se implementa a través de la biblioteca **ReportLab**, una herramienta robusta y ampliamente utilizada en entornos Python, que permite construir documentos PDF desde cero con un alto nivel de control y personalización.

Desde el inicio del proceso, se aprovecha el potencial de ReportLab para ir estructurando el documento paso a paso. Se define el contenido textual con precisión, se insertan imágenes en posiciones estratégicas, se diseñan tablas con datos obtenidos directamente de la base de datos y se crean encabezados jerárquicos que organizan la información de forma clara y comprensible. Todo esto se realiza en tiempo real, a partir de los parámetros que define el usuario mediante los filtros de selección disponibles en la interfaz del sistema.

El enfoque está centrado en ofrecer flexibilidad: el usuario no solo elige qué información desea visualizar, sino que también influye en cómo se presenta visualmente el informe. Es posible, por ejemplo, ajustar el formato de las tablas, definir la ubicación de los elementos gráficos, modificar el estilo tipográfico de los títulos y hasta seleccionar qué secciones incluir o excluir del documento final.

Este tipo de generación de informes resulta especialmente útil en el backend del sistema, donde se automatizan tareas repetitivas como la creación periódica de reportes o la exportación de resultados provenientes de consultas a bases de datos. De este modo, se asegura la consistencia de la presentación, al mismo tiempo que se minimiza el trabajo manual por parte del equipo técnico o de los usuarios finales.

4.6.3 Ejemplo de uso

Un usuario realiza una consulta sobre el municipio de Envigado en el departamento de Antioquia, filtrando por país, departamento y municipio, luego selecciona los determinantes a consultar, el sistema procesa esta información y la presenta en el mapa principal. Luego, con un solo clic, el usuario puede exportar estos datos a un archivo PDF. En ese archivo, el encabezado aparece con los metadatos, en la parte central el mapa con la información de los polígonos cargados y más abajo la tabla con el detalle de los determinantes seleccionados.

4.6.4 Metadatos incorporados en el informe (fecha, fuente, filtros)

El proceso de generar un informe robusto y detallado implica no solo presentar la información relevante, sino también asegurar que esta esté contextualizada de manera precisa. En este sentido, los metadatos juegan un rol crucial, proporcionando detalles adicionales que permiten al lector comprender el origen, la temporalidad y las condiciones bajo las cuales se obtuvo la información. Los metadatos son, por tanto, un componente esencial para garantizar la transparencia y la trazabilidad de los datos incluidos en el informe.

Los metadatos que deben ser incorporados de manera sistemática incluyen los siguientes elementos clave:

1. **Fecha de generación del informe:** Es fundamental que se especifique claramente el momento exacto en el que se genera el informe. Esto no solo proporciona una referencia temporal para evaluar la vigencia de la información, sino que también permite a los usuarios identificar la última actualización o versión de los datos presentados. El tiempo es un factor crítico en muchos análisis, por lo que tener esta referencia ayuda a comprender mejor las circunstancias bajo las cuales se recopiló la información.
2. **Fuente de los datos:** Todo dato tiene un origen. Indicar la fuente de los datos, ya sea una base de datos específica, un sistema de gestión o alguna otra fuente confiable, es esencial para garantizar la validez y la fiabilidad de la información. Esto permite a los usuarios rastrear los datos hasta su fuente original, lo que aporta una capa adicional de confianza en los resultados obtenidos. La especificación de la fuente también facilita la identificación de posibles sesgos o limitaciones inherentes a la misma.
3. **Filtros aplicados a los datos:** En muchas ocasiones, los datos se extraen bajo ciertas condiciones o restricciones que pueden afectar su alcance. Por ejemplo, pueden haberse aplicado filtros por región geográfica, periodo de tiempo, categoría específica de información o cualquier otro criterio relevante. Incluir estos filtros dentro de los metadatos permite al usuario entender cómo se ha delimitado el conjunto de datos, lo cual es fundamental para interpretar correctamente los resultados. El uso de filtros específicos puede influir en los patrones y tendencias que se observan, por lo que su inclusión en el informe garantiza una mayor comprensión sobre la precisión y el enfoque del análisis.

4.6.5 Ejemplo de metadatos en un archivo PDF

Al generar un informe en formato PDF, es posible que estos metadatos se incorporen de manera invisible para el lector, pero que jueguen un rol fundamental en la autenticidad del informe. Algunos ejemplos de metadatos que podrían encontrarse en un archivo PDF incluyen la fecha de creación del documento, el autor del archivo, la fuente de los datos y los parámetros de filtro utilizados para seleccionar la información presentada.

```
python

from datetime import datetime

fecha_actual = datetime.now().strftime("%Y-%m-%d %H:%M")
pdf.drawString(100, 700, f"Fecha de generación: {fecha_actual}")
pdf.drawString(100, 685, "Fuente: Sistema XYZ")
pdf.drawString(100, 670, "Filtros: Año=2024, Región=Andina")
```

Ilustración 2. Metadatos incorporados en el informe (fecha, fuente, filtros). Fuente: ADA

4.6.6 Creación de gráficos en informes mediante Matplotlib

La creación de gráficos en los informes es una parte fundamental para hacer los datos más comprensibles y visualmente atractivos. En este contexto, Matplotlib es una herramienta clave que se utiliza para generar gráficos diversos, como de barras, líneas o tortas, entre otros, a partir de los datos procesados previamente.

Este proceso de visualización no solo facilita la interpretación de la información, sino que también proporciona un enfoque más interactivo y accesible a los informes. Al emplear Matplotlib, se pueden crear representaciones gráficas detalladas y precisas, que brindan una visión clara de las tendencias, comparaciones o proporciones dentro de los datos, mejorando la comunicación de los resultados.

Una vez generados los gráficos, estos pueden ser visualizados en el aplicativo como imágenes en la sección de los indicadores, lo que permite integrarlos fácilmente en los informes finales. De esta forma, los usuarios pueden ver de manera inmediata sin necesidad de abrir archivos separados.

Así, se mejora la comprensión de los datos al complementar los informes con visualizaciones que son intuitivas y fáciles de interpretar. Los gráficos permiten que el usuario obtenga una visión más clara y completa de los resultados, lo que optimiza la presentación de la información de manera efectiva y profesional.

5. Requerimientos Técnicos

5.1 Infraestructura

La arquitectura tecnológica del aplicativo web público está basada en una infraestructura moderna, escalable y orientada a servicios. Este enfoque permite que el sistema crezca y se adapte a las necesidades futuras sin comprometer su rendimiento o disponibilidad. A continuación, se detallan los principales componentes de la infraestructura que soportan el funcionamiento y la seguridad del sistema:

5.1.1 Interfaz de Usuario

El front-end del aplicativo web público está construido utilizando **React.js**, una poderosa biblioteca de JavaScript que facilita la creación de interfaces de usuario dinámicas, modulares y altamente interactivas. React.js permite una actualización eficiente de la interfaz sin la necesidad de recargar la página completa, lo que mejora la experiencia del usuario.

Para gestionar el estado de la aplicación de manera centralizada y optimizar la comunicación entre los diferentes componentes de la interfaz, se utiliza **Redux**. Este patrón de gestión de estado facilita la coordinación de las interacciones entre los distintos módulos de la aplicación y controla el flujo de los datos en aplicaciones más complejas, permitiendo que cada componente reciba solo la información necesaria en el momento adecuado.

En términos de diseño y componentes visuales, se emplean bibliotecas como **Tailwind CSS**, para crear interfaces de usuario consistentes, atractivas y responsivas. Estas herramientas proporcionan una amplia gama de componentes predefinidos que agilizan el proceso de desarrollo y aseguran una apariencia y experiencia coherentes en diferentes dispositivos.

5.1.2 Backend (Sistema Interno)

El backend del sistema está desarrollado con **Django**, un framework web de alto nivel que facilita el desarrollo rápido de aplicaciones seguras y eficientes. Django es conocido por sus buenas prácticas, su enfoque en la seguridad y su capacidad para manejar grandes volúmenes de tráfico y datos.

Para permitir que el backend se comunique de manera eficiente con el frontend y otros sistemas externos, se implementa una **API RESTful** utilizando **Django REST Framework (DRF)**. DRF es una herramienta potente que permite exponer servicios web de manera flexible, robusta y fácil de consumir por aplicaciones cliente.

5.1.3 Base de Datos

La base de datos utilizada es **PostgreSQL**, un sistema de gestión de bases de datos relacional que se caracteriza por su fiabilidad, rendimiento y flexibilidad. PostgreSQL es una solución open source que ha sido ampliamente adoptada en aplicaciones empresariales debido a su robustez y capacidad para manejar grandes volúmenes de datos de manera eficiente.

Además, para manejar información geoespacial y territorial, se utiliza la extensión **PostGIS** de PostgreSQL. Esta extensión permite almacenar y consultar datos geográficos, lo que es esencial para aplicaciones que requieren el manejo de mapas, ubicaciones o cualquier tipo de análisis espacial.

5.1.4 Infraestructura en la Nube

La infraestructura del sistema está desplegada en la nube, utilizando **Docker** para la contenedorización. Docker permite encapsular la aplicación en contenedores que pueden ejecutarse de manera consistente en diferentes entornos, lo que mejora la portabilidad y facilita las tareas de desarrollo, pruebas y producción.

La orquestación de estos contenedores se gestiona mediante **Kubernetes**, que automatiza el despliegue, la escalabilidad y la administración de los contenedores. Kubernetes también garantiza que el sistema pueda recuperarse de fallos y escalar de manera automática según la demanda de usuarios.

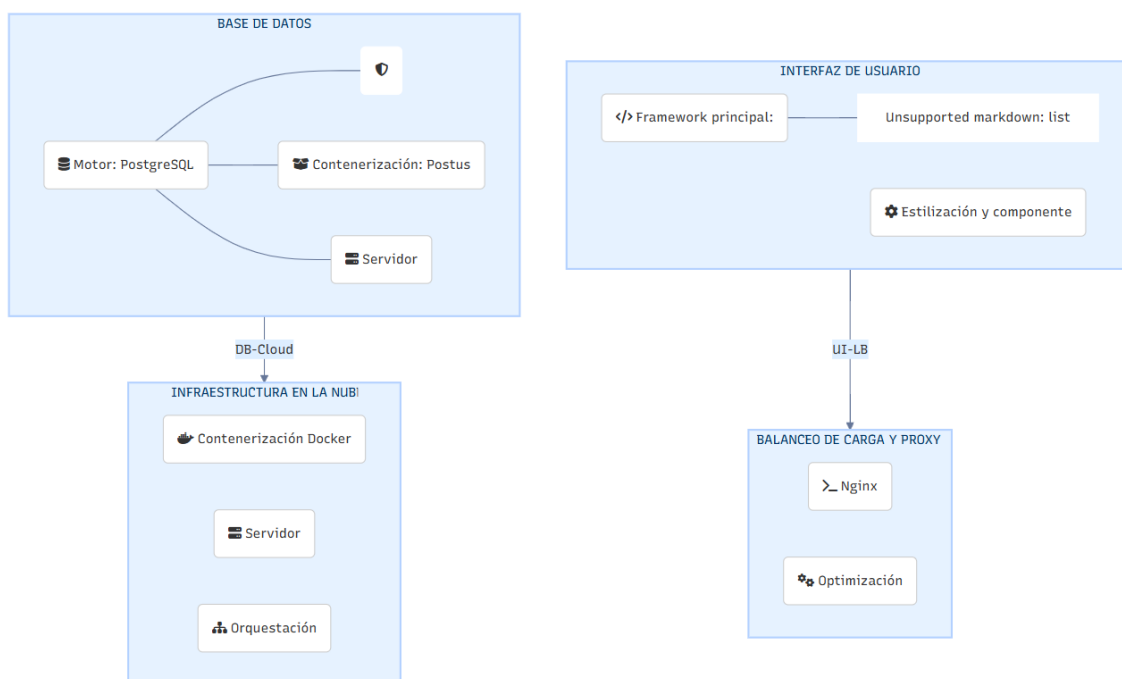


Ilustración 3. Infraestructura. Fuente: ADA

5.2. Procesamiento de Datos

El procesamiento de datos en este sistema se lleva a cabo a través de un proceso automatizado de extracción de información proveniente de archivos Excel y Shapefiles, usando procesos ETL (Extract, Transform, Load) que permiten integrar de manera eficiente grandes volúmenes de datos. Este proceso no solo se limita a la extracción, sino que también incluye etapas de validación y transformación de los datos, realizadas mediante Apache Airflow, una herramienta clave para la automatización de flujos de trabajo en la nube. Gracias a Apache Airflow, se asegura que los datos sean procesados de manera consistente y precisa, respetando las reglas de negocio y las necesidades del sistema.

Una vez que los datos han sido extraídos y transformados, se organiza la información de forma estructurada y optimizada, garantizando que sea fácilmente accesible y eficiente para

las consultas posteriores. Para esto, se utiliza Elasticsearch, un motor de búsqueda y análisis de datos en tiempo real que mejora considerablemente la rapidez y la precisión de las consultas, facilitando así el acceso a la información que los usuarios requieren para sus análisis.

5.3 Código Fuente del Desarrollo

El Código Fuente del desarrollo correspondiente al proyecto, será entregado como un documento adjunto para su revisión y análisis en formato txt. Dicho documento estará guardado en el servidor en la siguiente ruta:

/home/admgeor/Documents/Instaladores/Codigo fuente/codigo fuente.txt

5.4 Perfil del profesional idóneo para realizar despliegues del aplicativo

Ingeniero/a con dominio avanzado de la herramienta Docker, usada para construir y gestionar imágenes Compose/kubernetes, configuración de entornos CI/CD, integración con sistemas de control de versiones, optimización de imágenes, gestión de volúmenes y garantizar despliegues seguros y escalables.

6. Criterios de Aceptación

El sistema debe ser capaz de proporcionar a los usuarios una visualización clara y precisa de los datos segmentados por zona, departamento y municipio, lo que permitirá una comprensión detallada de la información geoespacialmente organizada. Los usuarios podrán seleccionar y comparar información de determinantes e indicadores a través de mapas y gráficos interactivos.

El sistema también será capaz de generar informes en formatos Excel y PDF, que incluyan tanto gráficos como metadatos, proporcionando un contexto adicional sobre la información presentada.

7. Aprobación

A continuación, firman aprobando:

Nombre	Cargo	Firma	Fecha
Responsable de Área (MME)			

Nombre	Cargo	Firma	Fecha
Adriana Patricia Cante Chaparro	Supervisora Contrato designada por FENOGE.		
María Fernanda Ramírez	Supervisor Convenio. MME-OAAS		