

Componente 2. Entregable 2.6.19 Ministerio de Minas y Energía

Proyecto CTO No. 116231-225-2024

Manual Técnico

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

Contenido

1. Introducción	5
2. Objetivo	5
3. Objetivos específicos	6
4. Requerimientos No Funcionales	7
4.1 Cumplimiento de Normativas	7
4.2 Escalabilidad	7
4.3 Experiencia de Usuario	7
4.4 Mantenimiento y Actualización	8
5. Criterios de Aceptación	8
5.1 Acceso	8
5.2 Interfaz y Funcionalidad	8
5.2.1 Módulo de Inicio (Landing Page)	9
5.2.2 Módulo de Consulta y Visualización de Información	11
5.2.3 Módulo de Comparación de Información	12
5.2.4 Módulo de Reportes y Exportación	12
5.3 Experiencia de Usuario	13
5.4 Reportes y Exportaciones	13
6. Reglas de Negocio y Descripción	13
6.1 Interfaz de Selección Dinámica	14
6.2 Comparación	14
7. Almacenamiento y Gestión	15
8. Procesamiento y Visualización de la Información	15
9. Exportación y Reportes	16
10. Entrega del Portal Web	17
11. Paso de instalación de Docker y despliegue del sitio web	18
11.1. Prerrequisitos	18
11.1.1 Cuenta de Docker	18
11.1.2 Puertos de red habilitados	18
11.1.3 Acceso administrativo al servidor	19
12. Pasos antes de la instalación de Docker en sistemas Linux/Ubuntu	19
12.1 Limpieza de versiones anteriores de Docker	19
12.2 Actualización de repositorios Docker en la máquina	20

12.2.1 Agregue la clave GPG oficial de Docker	22
12.2.2 Agregue el repositorio de Docker a las fuentes de APT	23
13. Instalación de Docker	24
13.1 Configuración de permisos para el usuario Docker	26
14. Despliegue de la Aplicación	28
14.1 Creación de la Imagen Docker	28
14.1.1 Navegue hasta la carpeta de la aplicación	28
14.1.2 Construcción de la imagen de Docker	28
14.1.3 Suba la imagen al repositorio de Docker	29
14.2 Creación del archivo docker-compose.yml	30
14.2.1 Navegación hacia el directorio /app	30
14.2.2 Cree y edite el archivo Docker -compose.yml	30
14.2.3 Transfiera el contenido del archivo docker-compose.yml desde su entorno local al servidor remoto	32
14.2.4 Proceda a ejecutar el despliegue utilizando el comando adecuado	32
14.3 Configuración y Ejecución de la Aplicación	33
14.3.1 Procedimiento para reiniciar la aplicación	33
14.4 Verificación	34
14.5 Solución de problemas	36
14.5.1 Error de permisos	36
14.5.2 Puertos bloqueados	37
14.5.3 Problemas con docker-compose	37
15. Aprobación	38

Índice de ilustraciones

Ilustración 1. Interfaz página de inicio. Fuente: Imagen elaborada por ADA.....	10
Ilustración 2. Pantalla de comparación. Fuente: Imagen elaborada por ADA.	10
Ilustración 3. Pantalla indicadores. Fuente: Imagen elaborada por ADA.	11
Ilustración 4. Comando para eliminar versiones antiguas de Docker. Fuente: Imagen elaborada por ADA.	20
Ilustración 5. Actualizar los repositorios comando para eliminar versiones antiguas de Docker. Fuente: Imagen elaborada por ADA.....	21
Ilustración 6. instalación segura de Docker. Fuente: Imagen elaborada por ADA.....	21
Ilustración 7. Clave GPG oficial de Docker. Fuente: Imagen elaborada por ADA.	22
Ilustración 8. Descargar la clave GPG. Fuente: Imagen elaborada por ADA.	22
Ilustración 9. Sistema APT pueda utilizar la clave. Fuente: Imagen elaborada por ADA...	23
Ilustración 10. Agregue el repositorio de Docker. Fuente: Imagen elaborada por ADA.....	24
Ilustración 11. Instalar Docker. Fuente: Imagen elaborada por ADA.....	25
Ilustración 12. Usuario al grupo Docker. Fuente: Imagen elaborada por ADA.	26
Ilustración 13. Usuario al grupo Docker. Fuente: Imagen elaborada por ADA.	27
Ilustración 14. Carpeta de la aplicación. Fuente: Imagen elaborada por ADA.	28
Ilustración 15. Para construir la imagen de Docker. Fuente: Imagen elaborada por ADA.	29
Ilustración 16. Suba la imagen al repositorio de Docker. Fuente: Imagen elaborada por ADA.	29
Ilustración 17. Directorio /app. Fuente: Imagen elaborada por ADA.	30
Ilustración 18. Archivo Docker -compose.yml. Fuente: Imagen elaborada por ADA.	32
Ilustración 19. Despliegue. Fuente: Imagen elaborada por ADA.	33
Ilustración 20. Reiniciar la aplicación Fuente: Imagen elaborada por ADA.....	33
Ilustración 21. Iniciar la aplicación en segundo plano. Fuente: Imagen elaborada por ADA.	34
Ilustración 22. Despliegue. Fuente: Imagen elaborada por ADA.	35
Ilustración 23. Despliegue. Fuente: Imagen elaborada por ADA.	36
Ilustración 24. Error de permisos. Fuente: Imagen elaborada por ADA.	36
Ilustración 25. Puertos bloqueados. Fuente: Imagen elaborada por ADA.....	37
Ilustración 26. Problemas con docker-compose. Fuente: Imagen elaborada por ADA.	38

1. Introducción

Este manual técnico tiene como objetivo detallar el diseño, desarrollo e implementación de una aplicación web, concebida específicamente para ofrecer una plataforma eficiente de consulta y análisis de información relevante relacionada con el sector minero-energético del país. La aplicación está orientada a facilitar el acceso a datos que son de gran interés para diversos actores del sector, permitiendo que la información se consulte de manera rápida y precisa.

El sistema permite a los usuarios acceder a datos estructurados que se presentan de forma clara y comprensible, con el propósito de que cualquier persona pueda acceder a información de interés general del sector minero-energético. Además, garantiza que la aplicación sea altamente eficiente, proporcionando la información necesaria con un mínimo de costos operativos. Esto se logra mediante la integración de herramientas avanzadas de búsqueda, filtrado y análisis de datos, que permiten a los usuarios generar informes, realizar consultas detalladas y obtener visualizaciones interactivas, todo dentro de una interfaz de usuario amigable y de fácil acceso.

El diseño de la aplicación se basa en una arquitectura escalable que permite su crecimiento y expansión a lo largo del tiempo. La arquitectura es flexible, de modo que se adapta a futuras actualizaciones o ampliaciones del sistema sin comprometer su rendimiento ni su capacidad de respuesta.

La aplicación se diseña en conformidad con las políticas dictadas por el Gobierno Digital en cuanto a usabilidad y accesibilidad, garantizando que los usuarios con diferentes tipos de discapacidades o limitaciones puedan interactuar con el sistema de manera eficiente. El cumplimiento de estas políticas no solo asegura la inclusión de todas las personas, sino que también facilita la navegación y el acceso a la información sin barreras.

Cabe aclarar que este informe no se basa en una integración en tiempo real de datos, sino en una implementación técnica que parte de los datos estáticos entregados por el MME. Estos datos se han organizado y adaptado para permitir un análisis tanto visual como espacial, según las necesidades del sector minero-energético.

2. Objetivo

El principal objetivo de esta aplicación web es proporcionar una plataforma robusta y fácil de usar para la consulta de indicadores y determinantes territoriales clave en el sector minero-energético, en el marco de la Transición Energética Justa (TEJ). Este enfoque busca promover el acceso equitativo a la información relevante, brindando a los usuarios las herramientas necesarias para analizar datos geoespaciales y tabulares de manera integral.

La aplicación web ofrecerá a los usuarios la posibilidad de acceder a una base de datos completa y actualizada, que contiene información geoespacial detallada (como mapas,

gráficos y visualizaciones interactivas) y datos tabulares (en formato de tablas o bases de datos estructuradas). La visualización de estos datos es altamente interactiva, permitiendo la comparación de diferentes conjuntos de datos y facilitando su análisis a través de filtros y parámetros personalizables.

Además de la visualización y comparación de datos, la aplicación brinda a los usuarios la opción de exportar la información consultada en varios formatos, como xlsx y PDF lo que permite su utilización fuera de la plataforma para fines de análisis más profundos o para la creación de informes personalizados.

Para garantizar una integración fluida y eficiente, la aplicación web se conecta a bases de datos estructuradas en PostgreSQL y PostGIS. Estas bases de datos están alimentadas por archivos proporcionados por el Ministerio de Minas y Energía (MME), los cuales pueden estar en formatos como Excel o Shapefile. Estos archivos contienen información crucial sobre el sector minero-energético, que se carga y procesa dentro del sistema para su consulta y análisis.

La aplicación no solo facilita el acceso a esta información relevante, sino que también permite la actualización periódica de los datos a medida que nuevos archivos se cargan al sistema. Esta integración de datos de múltiples fuentes garantiza que la información proporcionada sea siempre precisa, actualizada y relevante para los usuarios.

3. Objetivos específicos

- **Proporcionar información clave sobre indicadores y determinantes territoriales:** El sistema tiene como objetivo central ofrecer datos detallados y actualizados que sirven como base para la toma de decisiones estratégicas dentro del ámbito del sector minero-energético. Estos indicadores y determinantes proporcionan una visión precisa y completa, lo que permite una evaluación profunda y la identificación de patrones relevantes que influyen en el desarrollo del sector.
- **Implementación de un sistema web robusto:** El sistema web que se desarrolla está diseñado para ser completamente escalable, lo que significa que puede crecer en función de las necesidades futuras sin comprometer su rendimiento. Además, es seguro y está construido para ofrecer una alta disponibilidad, garantizando que los usuarios puedan acceder a la plataforma en todo momento sin interrupciones. Su compatibilidad con los navegadores web más utilizados, como Chrome, Firefox, Edge y Safari, asegura que los usuarios no enfrenten barreras tecnológicas al interactuar con la plataforma.
- **Facilitar la visualización y comparación de datos:** La plataforma implementa herramientas avanzadas que permiten la visualización clara de la información y su comparación entre diferentes criterios. Entre estas herramientas destacan los filtros avanzados, los acordeones para segmentar la información y la funcionalidad de multi-filtro, lo que facilita una navegación más fluida y precisa. Estas funcionalidades permiten que los

usuarios accedan rápidamente a la información que necesitan de manera ordenada y lógica, lo que mejora la eficiencia en la toma de decisiones.

- **Garantizar la integridad y disponibilidad de los datos:** La integridad de los datos es una prioridad dentro del sistema, asegurando que toda la información almacenada sea precisa, confiable y accesible en todo momento. Además, se habilitan funcionalidades para exportar los datos almacenados en formatos comunes y de uso extendido como PDF y Excel. Esto proporciona a los usuarios flexibilidad en el manejo de los datos, permitiéndoles analizarlos de manera más profunda fuera del sistema.

4. Requerimientos No Funcionales

4.1 Cumplimiento de Normativas

El sistema debe alinearse completamente con las políticas y directrices de **Gobierno Digital**, garantizando su accesibilidad y usabilidad para todos los usuarios. El cumplimiento de estas políticas asegura que el sistema sea eficiente, fácil de usar y accesible desde diversos dispositivos. Además, el sistema debe implementar robustas medidas de seguridad para proteger tanto la integridad como la confidencialidad de los datos. Estas medidas incluyen, al menos, el registro y monitoreo continuo de eventos críticos del sistema, lo cual permite llevar un control detallado y auditable de todas las actividades dentro de la plataforma (log de auditoría), lo que resulta esencial para la detección y prevención de posibles vulnerabilidades.

4.2 Escalabilidad

La escalabilidad es una de las características clave del sistema. Este debe ser capaz de adaptarse a un incremento en el número de usuarios concurrentes y en el volumen de datos, sin perder desempeño ni afectando la experiencia de los usuarios. La capacidad de crecimiento del sistema está asegurada mediante una arquitectura flexible y optimizada. Adicionalmente, el sistema debe ser compatible con los principales navegadores web modernos, como **Google Chrome, Mozilla Firefox, Microsoft Edge y Apple Safari**, para garantizar que los usuarios puedan acceder sin problemas desde diversas plataformas. La estructura tecnológica del sistema permitirá la expansión de sus capacidades conforme aumente la demanda de acceso y la complejidad de los datos procesados.

4.3 Experiencia de Usuario

El diseño de la interfaz del sistema está enfocado en ofrecer una experiencia de usuario que sea fluida, intuitiva y adaptada a diversos dispositivos. La interfaz es completamente **responsiva**, lo que asegura que el sistema se pueda utilizar sin inconvenientes tanto en dispositivos móviles como de escritorio. Se prestará especial atención a la usabilidad del sistema, de modo que los usuarios puedan interactuar de forma eficiente con los filtros y

las herramientas de selección avanzada de indicadores, minimizando la curva de aprendizaje.

La visualización de los datos se optimiza mediante la integración de mapas geoespaciales interactivos, que permiten ver de manera clara y precisa las distribuciones territoriales de los datos relevantes. A su vez, las tablas de resultados se diseñan de forma que faciliten la interpretación y comparación de los indicadores, permitiendo un análisis detallado y eficaz.

4.4 Mantenimiento y Actualización

El mantenimiento y las actualizaciones del sistema se gestionarán en un entorno controlado y organizado, para asegurar que cualquier modificación o mejora no afecte la disponibilidad del servicio. Estas actualizaciones se probarán en un entorno de pruebas, con el fin de evitar posibles interrupciones o inconvenientes. Es importante destacar que la responsabilidad de estas tareas recaerá en el **Ministerio de Minas y Energía (MME)**, asegurando un control adecuado y constante del sistema. Además, se garantizará que las actualizaciones no solo optimicen el funcionamiento del sistema, sino que también mantengan su alineación con las necesidades del sector minero-energético y las políticas gubernamentales.

5. Criterios de Aceptación

5.1 Acceso

El acceso inicial al sistema se habilita exclusivamente dentro de un entorno de pruebas, el cual se encuentra debidamente aislado de los ambientes de producción y orientado a actividades de validación, verificación funcional, pruebas de calidad y ajustes técnicos previos al despliegue definitivo. Este entorno de pruebas se configura con estándares de seguridad informática que garantizan la integridad, disponibilidad y confidencialidad de los datos manejados durante el proceso de evaluación. Asimismo, se asegura la trazabilidad completa de todas las acciones realizadas por los usuarios habilitados, permitiendo un control detallado del comportamiento del sistema en condiciones simuladas.

La administración, mantenimiento y supervisión de este entorno recae directamente bajo la responsabilidad del Ministerio de Minas y Energía (MME), quien lidera la gestión del acceso, define los perfiles de usuario permitidos y monitorea las sesiones activas. Este proceso se realiza conforme a protocolos internos establecidos por el MME, incluyendo políticas de acceso seguro, autenticación por roles, y restricción temporal del uso del entorno.

5.2 Interfaz y Funcionalidad

El sistema aplicativo web que se entrega al Ministerio de Minas y Energía (MME) dispone de una interfaz moderna, responsiva e intuitiva, diseñada bajo principios de usabilidad y accesibilidad que permiten una interacción fluida, eficiente y comprensible por parte de los usuarios, incluso aquellos sin conocimientos técnicos avanzados. La solución incorpora componentes gráficos y funcionales que garantizan la experiencia del usuario en todas las

etapas de navegación, consulta, análisis y exportación de la información disponible sobre el sector minero-energético colombiano.

La aplicación estructura su funcionalidad mediante módulos independientes pero interconectados, optimizados para ofrecer acceso a información clave mediante consultas rápidas, filtros dinámicos y representaciones visuales integradas. A continuación, se describen de forma técnica los módulos principales del sistema:

5.2.1 Módulo de Inicio (Landing Page)

Este módulo constituye la puerta de entrada a la aplicación y se carga automáticamente al acceder a la URL del sistema. Su desarrollo considera aspectos de experiencia de usuario (UX) y diseño de interfaces (UI) para presentar de forma clara y concisa los elementos iniciales del sistema. Las funcionalidades principales incluyen:

- **Acceso rápido a módulos clave:** Se dispone de enlaces directos a las secciones de consulta, análisis comparativo, visualización geográfica y generación de reportes.
- **Selección de determinantes:** El usuario tiene la posibilidad de seleccionar directamente desde el panel izquierdo entre cuatro categorías principales: social, ambiental, sectorial e infraestructura.
- **Selección de indicadores:** El usuario tiene la posibilidad de seleccionar directamente desde la página inicial los principales indicadores de interés, facilitando la navegación contextual.
- **Mapa de Comparación:** Esta funcionalidad permite comparar determinantes e indicadores entre diferentes departamentos o municipios. El usuario puede seleccionar las entidades territoriales que desee contrastar y visualizar los datos en tablas o gráficos comparativos. La presentación de los datos está pensada para facilitar la interpretación visual, haciendo uso de colores, etiquetas y leyendas que permiten entender rápidamente las diferencias o similitudes entre los territorios seleccionados.

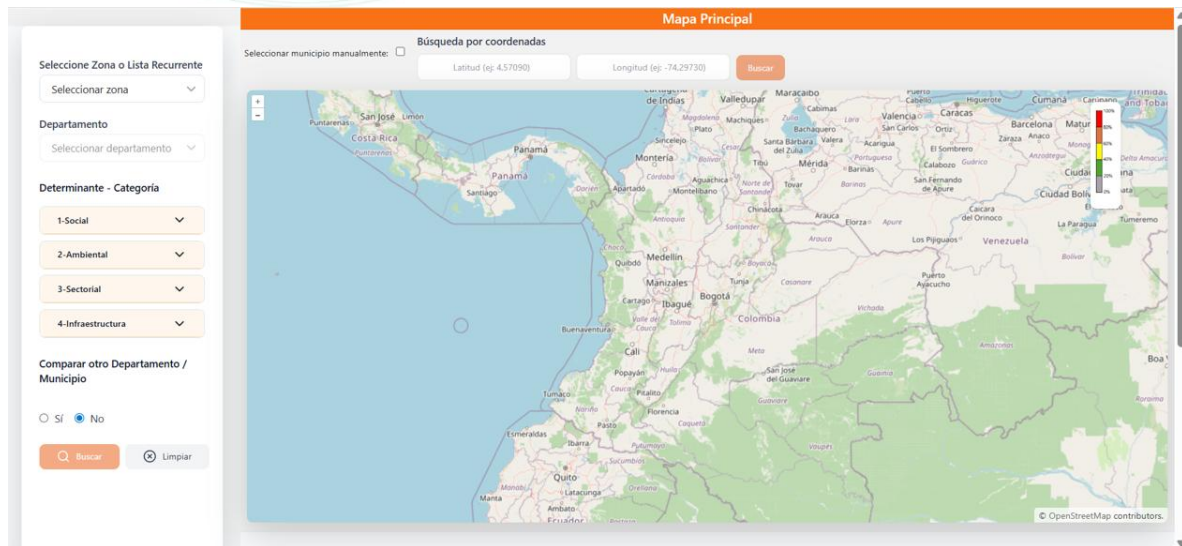


Ilustración 1. Interfaz página de inicio. Fuente: Imagen elaborada por ADA.

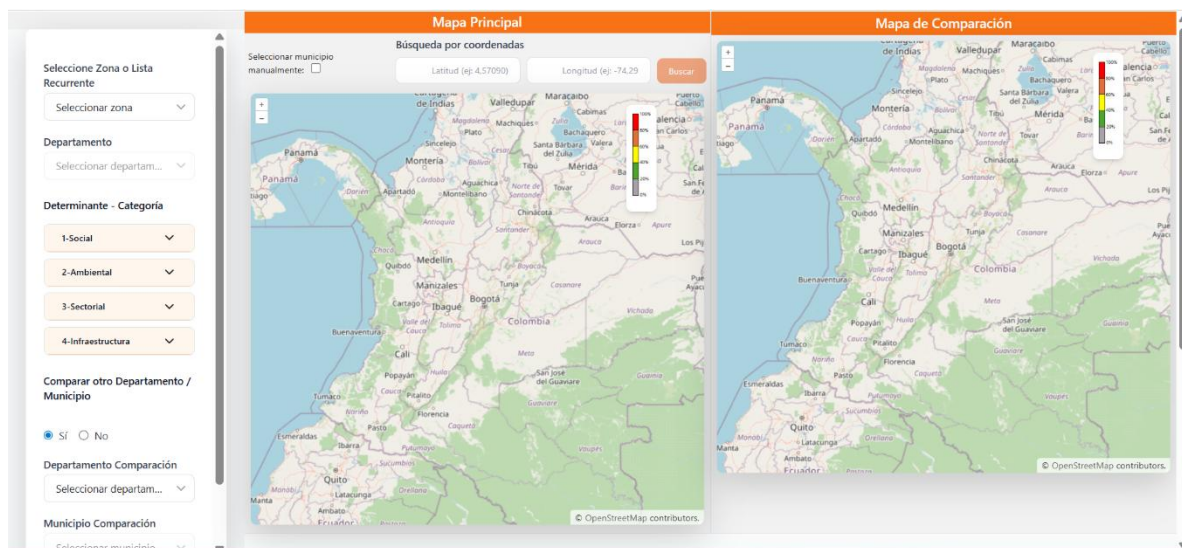


Ilustración 2. Pantalla de comparación. Fuente: Imagen elaborada por ADA.

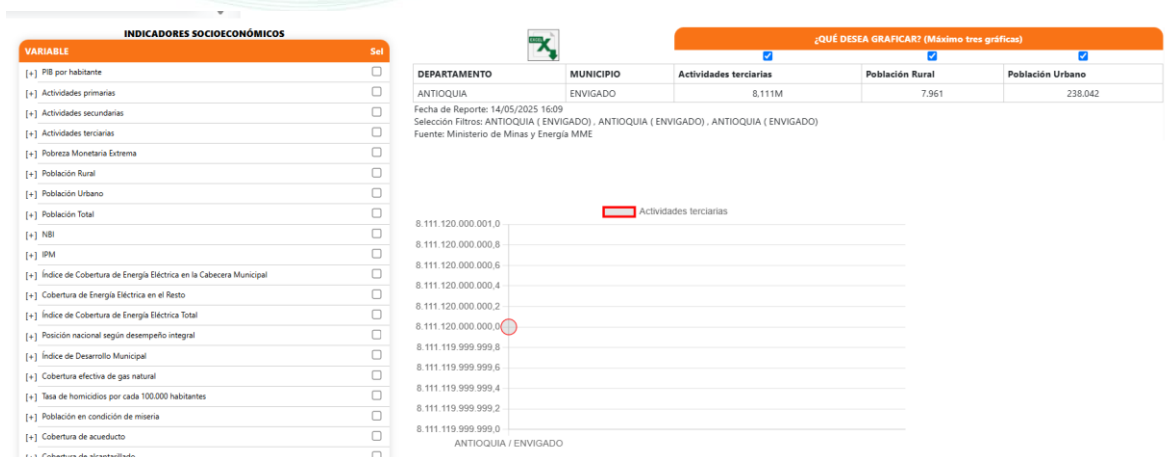


Ilustración 3. Pantalla indicadores. Fuente: Imagen elaborada por ADA.

5.2.2 Módulo de Consulta y Visualización de Información

Este módulo permite al usuario acceder a la base de datos del sistema para consultar y visualizar información estructurada, tanto en formato tabular como en representación geoespacial. El módulo combina funcionalidades de filtrado avanzado con visualización interactiva, y se integra con tecnologías como PostGIS para el manejo de datos geográficos.

Visualización tabular:

- Presenta información organizada por Zona, Departamento y Municipio.
- Permite la navegación rápida a través de registros, mediante funciones de ordenamiento, paginación y búsqueda contextual.
- Habilita la aplicación de filtros múltiples, tanto jerárquicos como cruzados, para refinar los resultados de consulta de acuerdo con los intereses del usuario.

Interacción geográfica:

- Utiliza consultas espaciales con PostGIS para representar la información como objetos geoespaciales en mapas interactivos.
- Ofrece herramientas para la selección territorial, tanto mediante clics directos sobre el mapa como a través de listas desplegables.
- Aplica filtros dinámicos sobre capas específicas para focalizar la visualización en parámetros seleccionados.
- Incorpora capas temáticas superpuestas que permiten analizar simultáneamente múltiples indicadores o variables en un solo entorno visual.

5.2.3 Módulo de Comparación de Información

Este módulo está diseñado para facilitar el análisis comparativo entre diferentes unidades territoriales, permitiendo identificar patrones, diferencias y similitudes en los indicadores clave del sector.

Las funcionalidades específicas que ofrece incluyen:

- Comparación directa entre municipios, departamentos o zonas del país, permitiendo seleccionar múltiples entidades territoriales en paralelo.
- Visualización gráfica de las diferencias entre los valores registrados, mediante gráficos de líneas que facilitan la interpretación.
- Generación de reportes comparativos que combinan tablas de datos con mapas geospaciales para representar los resultados del análisis.
- Aplicación de filtros personalizados que permiten al usuario definir los criterios exactos de comparación, ajustando el enfoque según el objetivo del análisis.

5.2.4 Módulo de Reportes y Exportación

Este módulo centraliza todas las funcionalidades de generación, descarga y exportación de datos e informes, integrando herramientas para que los usuarios puedan obtener reportes estructurados según sus necesidades específicas.

Generación de reportes dinámicos:

- Crea informes automáticos a partir de consultas predefinidas o personalizadas por el usuario.
- Permite la exportación en formato PDF, incluyendo gráficos, tablas, leyendas e información contextual relevante.
- Incluye metadatos en los reportes generados, tales como fecha y hora de emisión, filtros aplicados, variables consultadas y fuentes oficiales de los datos.

Exportación de datos:

- Ofrece la opción de descargar datos filtrados en formato Excel, facilitando su análisis posterior fuera del sistema.
- Exporta representaciones geográficas en formato de imagen o PDF, permitiendo su uso en presentaciones, documentos o estudios técnicos.
- Brinda opciones avanzadas de personalización antes de la descarga, tales como selección de columnas, ordenamiento de registros, idioma del reporte, y formato visual.

5.3 Experiencia de Usuario

Gracias a esta estructura modular y a la integración de tecnologías robustas tanto en el backend como en el frontend, la aplicación garantiza una experiencia satisfactoria al usuario. Se asegura que cualquier persona con acceso al sistema pueda navegar, consultar, analizar y exportar la información de manera precisa, clara y eficiente. La interfaz busca facilitar el entendimiento del sector minero-energético colombiano a partir de datos confiables y visualizaciones intuitivas.

5.4 Reportes y Exportaciones

El sistema implementa funcionalidades robustas para la **generación de reportes personalizados**, permitiendo al usuario acceder a información estructurada y visualmente organizada en formato PDF. Esta capacidad incluye la incorporación de elementos gráficos como tablas, gráficos y logotipos institucionales, facilitando una presentación clara y profesional de los datos procesados. El motor de generación de reportes se integra directamente con los módulos de consulta y análisis, lo que garantiza que los documentos reflejan en tiempo real los filtros y parámetros seleccionados por el usuario. Asimismo, el sistema permite la configuración de plantillas predefinidas y personalizables, asegurando coherencia con la identidad visual de la organización y adaptabilidad a distintos contextos de uso, como auditorías, presentaciones ejecutivas o documentación interna.

Adicionalmente, la plataforma **habilita la exportación de datos filtrados** a hojas de cálculo en formato Excel (.xlsx), optimizando los flujos de trabajo para análisis cuantitativos o procesamiento externo de la información.

Esta funcionalidad respeta los criterios de búsqueda y segmentación definidos por el usuario, exportando únicamente los registros relevantes y asegurando la integridad de los datos exportados. El formato generado mantiene una estructura ordenada, con encabezados claramente definidos, lo cual facilita su comprensión e interpretación por parte de analistas, contadores, o cualquier profesional que requiera manipular los datos con herramientas ofimáticas. Además, el sistema preserva las relaciones entre columnas y puede incluir formatos condicionales básicos para destacar valores clave, contribuyendo a una mejor toma de decisiones basada en la información extraída.

6. Reglas de Negocio y Descripción

El sistema implementa un conjunto de reglas de negocio orientadas a optimizar y perfeccionar los procesos de carga, visualización, interacción y análisis de datos estructurados y geoespaciales. Estas reglas se diseñan con el propósito de garantizar que las funcionalidades respondan de manera eficiente, flexible y escalable a los requerimientos de los usuarios, especialmente en entornos relacionados con la gestión de información del sector público, específicamente en lo concerniente al análisis de indicadores socioeconómicos.

Cada regla de negocio se construye bajo principios de modularidad, personalización y usabilidad, de forma que se facilite la toma de decisiones informadas y contextualizadas. A continuación, se describen las características técnicas fundamentales de dichas reglas:

6.1 Interfaz de Selección Dinámica

Se configura una interfaz gráfica de usuario (GUI) avanzada mediante el uso de un sistema de acordeones participativos, los cuales estructuran la experiencia del usuario en bloques interactivos que permiten la selección simultánea, intuitiva y jerárquica de múltiples determinantes e indicadores. Esta arquitectura de interfaz responde a la necesidad de explorar diversas dimensiones de análisis, proporcionando una navegación eficiente dentro de grandes volúmenes de información categorizada.

Para la gestión y visualización de indicadores socioeconómicos simples y compuestos, se habilita un módulo de selección estructurada, basado en cuadros desplegables jerarquizados y listas de chequeo inteligentes. Este componente permite que el usuario configure escenarios de análisis a través de la combinación libre de variables, ajustando las vistas de resultados a sus necesidades específicas.

El diseño de esta interfaz se fundamenta en principios de experiencia de usuario (UX) y accesibilidad, asegurando compatibilidad con diferentes dispositivos y navegadores, así como fluidez en la interacción, aún en contextos de baja conectividad.

6.2 Comparación

El sistema incorpora un módulo especializado de comparación que permite analizar de forma simultánea dos entidades territoriales a nivel departamental y dos a nivel municipal, conforme a un conjunto de reglas previamente establecidas dentro del sistema o definidas dinámicamente por el usuario según los criterios particulares del análisis requerido. Este componente funcional se diseña para ofrecer un alto grado de flexibilidad, permitiendo al usuario seleccionar parámetros específicos y ajustar las condiciones de evaluación en función de sus necesidades analíticas.

Las reglas de comparación pueden incluir indicadores cuantitativos y cualitativos, métricas geoespaciales, atributos socioeconómicos, datos históricos, entre otros, los cuales se integran en una interfaz amigable que facilita la configuración de comparativas personalizadas. El sistema interpreta estas reglas mediante un motor lógico que evalúa los datos asociados a cada entidad territorial, generando resultados visuales y tabulares que permiten una interpretación clara y precisa de las diferencias y similitudes encontradas. Esta funcionalidad apoya la toma de decisiones estratégicas en el ámbito gubernamental y técnico, al permitir identificar patrones, tendencias y disparidades entre regiones.

7. Almacenamiento y Gestión

La solución tecnológica implementada se estructura sobre un sistema de gestión de bases de datos relacional (SGBDR), específicamente utilizando el motor de base de datos PostgreSQL, el cual se configura con la extensión espacial PostGIS para facilitar el manejo de información georreferenciada y espacial. Esta arquitectura permite almacenar, consultar y analizar grandes volúmenes de datos tanto alfanuméricos como espaciales de forma eficiente, garantizando un desempeño óptimo y una alta disponibilidad de la información.

El modelo de datos se diseña conforme a principios de normalización y buenas prácticas en diseño de bases de datos, lo que asegura una estructura coherente, escalable y fácilmente mantenible. Se implementan políticas rigurosas de control de acceso mediante roles, privilegios y autenticación a nivel de base de datos, garantizando que solo los usuarios autorizados puedan acceder o modificar la información.

Asimismo, se establecen restricciones de integridad referencial y validaciones a nivel de esquema para preservar la consistencia de los datos, minimizando el riesgo de errores durante las operaciones de inserción, actualización o eliminación. La gestión de respaldos se configura de manera automática y periódica, permitiendo la recuperación eficiente de la información en caso de fallos o pérdida de datos.

Durante los procesos de carga de información, el sistema ejecuta mecanismos de validación, transformación y limpieza de datos, especialmente diseñados para procesar archivos de entrada en formatos Excel (.xlsx) y Shapefile (.shp), los cuales son suministrados por el Ministerio de Minas y Energía (MME). Estos mecanismos permiten verificar la estructura, formato y contenido de los datos, asegurando que cumplan con los estándares definidos antes de ser integrados al repositorio central. Además, se realiza un registro detallado de cada operación de carga, lo que facilita la trazabilidad, auditoría y supervisión de los procesos de integración de datos.

8. Procesamiento y Visualización de la Información

El proceso de procesamiento y visualización de la información constituye una etapa fundamental dentro del sistema, ya que permite transformar datos geográficos y estadísticos en representaciones comprensibles, dinámicas y útiles para la toma de decisiones. Este módulo inicia con la carga estructurada de la información geoespacial correspondiente al mapa político-administrativo de Colombia, el cual incluye datos detallados de todos los departamentos y sus respectivos municipios. Esta carga se realiza a través de mecanismos automatizados que garantizan la integridad, consistencia y actualidad de los datos, permitiendo así una representación fiel del territorio nacional.

Durante esta fase, el sistema habilita al usuario para seleccionar de manera interactiva departamentos y municipios específicos, permitiendo realizar análisis focalizados según necesidades puntuales. Esta selección se realiza a través de interfaces gráficas intuitivas

que combinan menús desplegables, filtros jerárquicos y elementos cartográficos interactivos, lo que optimiza la experiencia de navegación y consulta.

Una vez definidos los parámetros de consulta, el sistema procede al procesamiento de los datos mediante algoritmos que permiten organizar, filtrar, agregar y transformar la información en tiempo real. Como resultado de este procesamiento, se generan múltiples formas de representación visual de los datos, entre las cuales se destacan:

- **Tablas de datos dinámicas:** permiten visualizar la información en forma tabular, con opciones de ordenamiento, búsqueda y exportación. Estas tablas se actualizan automáticamente en función de los filtros aplicados por el usuario, asegurando una visualización coherente y contextualizada.
- **Gráficos generados a partir de consultas configurables:** el sistema ofrece un conjunto de herramientas analíticas que permiten construir visualizaciones gráficas (diagramas de líneas) personalizadas según variables seleccionadas. Estas gráficas se generan en tiempo real, a partir de las consultas parametrizadas definidas por el usuario, facilitando la interpretación comparativa de los datos y la identificación de tendencias.
- **Mapas interactivos con filtros avanzados:** se habilita una visualización geográfica de la información mediante mapas temáticos que incorporan múltiples capas de datos. Estos mapas permiten aplicar filtros por categorías y ofrecen funcionalidades como zoom, resaltado de áreas por escala de colores dependiendo de los determinantes seleccionados, ventanas emergentes con detalles contextuales (tooltip), y navegación fluida. La interactividad del mapa garantiza una exploración detallada del territorio y una comprensión espacial más profunda de los fenómenos representados.

Este enfoque técnico y visual, centrado en la experiencia del usuario, mejora la eficiencia en el análisis de la información al brindar una plataforma robusta y confiable para la exploración de datos territoriales y estadísticos.

9. Exportación y Reportes

El sistema implementa funcionalidades avanzadas que permiten la generación de reportes personalizados conforme a los requerimientos específicos de los usuarios autorizados. Esta capacidad de exportación se encuentra habilitada para formatos ampliamente utilizados como PDF y Excel, facilitando la interoperabilidad con otros sistemas de análisis y gestión documental.

Cada reporte generado incorpora de manera estructurada y automatizada un conjunto de metadatos esenciales que garantizan la trazabilidad, integridad y contextualización de la información presentada. Dichos metadatos incluyen, pero no se limitan a:

- **Fecha del reporte:** El sistema registra la fecha exacta en la que se genera el documento, asegurando su validez temporal y permitiendo el seguimiento histórico de los datos presentados.
- **Departamento:** Se especifica el departamento geográfico al que corresponde la información contenida en el reporte, permitiendo una segmentación territorial precisa.
- **Municipio:** Se detalla el municipio correspondiente, ofreciendo un nivel adicional de granularidad en la presentación de los datos.
- **Fuente:** Cada documento señala como fuente oficial el listado completo enviado por el Ministerio de Minas y Energía (MME), validando así la legitimidad de los datos utilizados.

La estructura de los reportes se diseña para ser flexible y escalable, permitiendo la incorporación de filtros dinámicos, gráficos comparativos, y resúmenes analíticos cuando se requiera. Además, el sistema asegura que los datos exportados mantengan su integridad, utilizando mecanismos de validación previos a la generación del archivo.

10. Entrega del Portal Web

El proceso de entrega del portal web se realiza inicialmente dentro de un entorno de desarrollo controlado, el cual es proporcionado directamente por el **Ministerio de Minas y Energía (MME)**. Este entorno permite realizar pruebas funcionales, técnicas y de seguridad, garantizando que la plataforma cumpla con los estándares de calidad establecidos antes de su paso a producción.

Durante esta fase, el sistema se somete a procesos de verificación rigurosos que incluyen pruebas de rendimiento, validación de funcionalidades clave, aseguramiento de la experiencia del usuario (UX), y revisiones de compatibilidad con diferentes navegadores y dispositivos.

Adicionalmente, se establece como requisito que el **Ministerio de Minas y Energía (MME)** disponga de un entorno propio, completamente habilitado y configurado para la implementación definitiva en producción. Este entorno debe cumplir con los lineamientos técnicos de infraestructura, ciberseguridad y disponibilidad establecidos por las políticas institucionales, garantizando así una operación estable, segura y continua del portal en su fase pública.

11. Paso de instalación de Docker y despliegue del sitio web

Esta sección describe de forma técnica y detallada el procedimiento necesario para llevar a cabo la instalación del entorno Docker y la posterior implementación del sitio web dentro de contenedores. Se abordan los requisitos previos indispensables, asegurando que el entorno cumpla con las condiciones mínimas necesarias para evitar errores durante la instalación y el despliegue. Esta guía está dirigida a administradores de sistemas, desarrolladores y personal técnico que requieren configurar un entorno de ejecución basado en contenedores Docker para una aplicación web específica.

11.1. Prerrequisitos

Antes de comenzar con cualquier instalación, resulta esencial validar que el sistema en el que se va a trabajar cumpla con una serie de prerrequisitos técnicos. Estos requisitos garantizan que los servicios a desplegar funcionen correctamente y que los contenedores Docker puedan operar sin restricciones a nivel de red, seguridad o permisos. A continuación, se detallan cada uno de los prerrequisitos necesarios:

11.1.1 Cuenta de Docker

Es indispensable disponer de una cuenta de Docker activa y vinculada a la máquina local desde la cual se realiza el despliegue. Esta cuenta permite autenticar operaciones como la descarga de imágenes desde el Docker Hub, el acceso a repositorios privados (en caso de ser necesario) y la gestión de contenedores a través de la CLI de Docker. En caso de que el usuario aún no posea una cuenta registrada, se proporciona más adelante una sección con instrucciones paso a paso para su creación y configuración.

11.1.2 Puertos de red habilitados

Para que los distintos servicios que conforman la aplicación web puedan comunicarse y estar accesibles tanto interna como externamente, es imprescindible que ciertos puertos estén habilitados en el archivo de configuración `docker-compose.yml`, así como en el firewall del servidor y cualquier otro dispositivo de red intermedio. A continuación, se enumeran los puertos que deben estar disponibles:

- **Puerto 80:** Canal estándar para tráfico HTTP. Permite el acceso público al sitio web sin necesidad de cifrado.
- **Puerto 3306:** Utilizado por el motor de base de datos MySQL. Necesario para la conexión entre los contenedores que manejan la capa de aplicación y la base de datos.
- **Puerto 8080:** Usado comúnmente para servidores de aplicaciones en modo desarrollo o pruebas. Puede estar asociado al backend de la aplicación.

- **Puerto 3030:** Habitualmente empleado por interfaces de usuario, paneles de administración o herramientas de visualización como Grafana o similares.

Cada uno de estos puertos debe estar libre de conflictos con otros servicios del sistema operativo y debe permanecer accesible durante todo el ciclo de vida del entorno de desarrollo o producción.

11.1.3 Acceso administrativo al servidor

Es necesario que el usuario que ejecuta la instalación disponga de privilegios administrativos sobre el sistema operativo del servidor. Esto se traduce en tener permisos sudo, que habilitan la ejecución de comandos críticos como la instalación de paquetes, la configuración de servicios del sistema, la modificación de archivos protegidos y la administración de redes. Sin estos permisos, no será posible llevar a cabo una instalación completa y funcional del entorno Docker.

Nota: En caso de que Docker no esté instalado previamente en el sistema, la próxima sección ofrece un procedimiento detallado para instalar Docker en distintas distribuciones de Linux, así como en entornos Windows y macOS, incluyendo instrucciones para la verificación de instalación y configuración básica post-instalación.

12. Pasos antes de la instalación de Docker en sistemas Linux/Ubuntu

La instalación de Docker en sistemas operativos basados en Linux, como Ubuntu, requiere una preparación cuidadosa del entorno para asegurar la compatibilidad y el correcto funcionamiento del motor de contenedores. Este proceso se inicia con la eliminación de cualquier rastro de versiones antiguas que puedan generar conflictos o interferencias con la nueva instalación. A continuación, se detalla el procedimiento técnico y sistemático para realizar esta limpieza de manera segura y eficiente.

12.1 Limpieza de versiones anteriores de Docker

Antes de proceder con la instalación de una versión actual de Docker, es imprescindible eliminar completamente las versiones previamente instaladas. Esta práctica garantiza que no existan conflictos de dependencias ni configuraciones residuales que puedan comprometer la estabilidad del entorno de contenedores.

Para ello, se ejecuta una iteración sobre una lista específica de paquetes relacionados con Docker y tecnologías similares que pudieran estar presentes en el sistema. Esta lista incluye tanto versiones clásicas como alternativas o herramientas auxiliares que comparten componentes con Docker.

El siguiente bloque de código implementa esta limpieza:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc;  
do  
    sudo apt-get remove $pkg  
done
```

Ilustración 4. Comando para eliminar versiones antiguas de Docker. Fuente: Imagen elaborada por ADA.

Este script utiliza un bucle for en el que se recorre cada uno de los paquetes especificados. Para cada iteración, el sistema invoca apt-get remove con privilegios de superusuario (sudo), lo que permite eliminar el paquete correspondiente del sistema.

- **docker.io:** corresponde a una versión empaquetada de Docker incluida en los repositorios oficiales de Ubuntu. Aunque funcional, esta versión puede no estar actualizada respecto al lanzamiento oficial de Docker.
- **docker-doc:** proporciona la documentación oficial de Docker instalada localmente, la cual no resulta necesaria si se planea obtener la última documentación en línea.
- **docker-compose** y **docker-compose-v2:** representan dos generaciones distintas de la herramienta de orquestación de contenedores Docker Compose. La versión 2 reescribe la funcionalidad en Go y se integra de forma más estrecha con el cliente Docker.
- **podman-docker:** implementa una capa de compatibilidad que permite ejecutar comandos de Docker utilizando el motor Podman. Esta capa puede causar confusión si se desea trabajar exclusivamente con Docker.
- **containerd:** es un runtime de contenedores independiente que Docker utiliza internamente. Su eliminación es recomendable si fue instalado manualmente o como parte de una configuración anterior.
- **runc:** representa una herramienta fundamental en la ejecución de contenedores OCI. Docker la utiliza en segundo plano para la creación y gestión de contenedores a bajo nivel.

La ejecución de este comando no elimina las configuraciones personalizadas ni los volúmenes de Docker, pero libera el sistema de posibles conflictos binarios o de dependencias. Una vez finalizada esta limpieza, el sistema queda listo para continuar con la instalación de la última versión estable de Docker desde su repositorio oficial.

12.2 Actualización de repositorios Docker en la máquina

Como paso inicial y fundamental en el proceso de preparación del entorno para la instalación de Docker, se procede a la actualización de los repositorios de paquetes del sistema operativo. Esta acción garantiza que el sistema obtenga la información más

reciente sobre las versiones disponibles de los paquetes, así como sus respectivas dependencias, desde los repositorios oficiales configurados en el sistema.

Para ello, se ejecuta el siguiente comando desde la terminal con privilegios de superusuario:

```
sudo apt-get update  
sudo apt-get install -y ca-certificates curl
```

Ilustración 5. Actualizar los repositorios comando para eliminar versiones antiguas de Docker. Fuente: Imagen elaborada por ADA.

Este comando solicita al sistema que sincronice los índices de los paquetes con las fuentes configuradas en los archivos localizados en `/etc/apt/sources.list` y en el directorio `/etc/apt/sources.list.d/`. Al hacerlo, se asegura que cualquier instalación posterior utilice las versiones más actualizadas y estables de los paquetes, lo cual es especialmente importante cuando se trata de software que depende de bibliotecas específicas o que se encuentra en constante evolución, como Docker.

Una vez actualizados los índices de los paquetes, se procede a la instalación de un conjunto de utilidades esenciales para la descarga e instalación segura de Docker. En particular, se instalan los paquetes `ca-certificates` y `curl`, utilizando el siguiente comando:

```
bash  
  
sudo apt-get install -y ca-certificates curl
```

Ilustración 6. instalación segura de Docker. Fuente: Imagen elaborada por ADA.

El paquete `ca-certificates` instala una colección de certificados de autoridades de certificación (CA) que permiten establecer conexiones HTTP. Este componente resulta fundamental para garantizar la integridad y autenticidad de las comunicaciones al momento de descargar software desde servidores remotos, como es el caso de los repositorios oficiales de Docker.

Por otro lado, el paquete `curl` proporciona una herramienta de línea de comandos extremadamente versátil, que permite realizar transferencias de datos a través de múltiples protocolos de red, incluidos HTTP, HTTPS, FTP y muchos otros. `curl` se emplea ampliamente en scripts de automatización y en procedimientos de instalación, debido a su capacidad para realizar solicitudes a servidores web y obtener contenido de forma sencilla y eficiente.

Al incluir la opción `-y` en el comando de instalación, se indica al sistema que apruebe automáticamente todas las solicitudes de confirmación que normalmente se presentan durante el proceso. Esto permite que la instalación se lleve a cabo de manera no interactiva,

facilitando la automatización del proceso en entornos donde la intervención humana se desea minimizar, como en servidores o sistemas desplegados en la nube.

Este procedimiento, aunque sencillo en apariencia, establece una base sólida y segura sobre la cual se desarrollan los pasos siguientes en la instalación y configuración de Docker, garantizando que el sistema disponga de los componentes necesarios para acceder de manera segura y eficiente a los repositorios de software requeridos.

12.2.1 Agregue la clave GPG oficial de Docker

Como parte fundamental del proceso de instalación segura de Docker en un entorno basado en Ubuntu, se procede inicialmente a la incorporación de la clave GPG oficial proporcionada por Docker. Este paso garantiza la verificación de la autenticidad e integridad de los paquetes descargados desde su repositorio oficial, protegiendo así el sistema frente a posibles alteraciones o distribuciones no autorizadas.

Para comenzar, se crea de manera explícita el directorio `/etc/apt/keyrings`, el cual se destina a almacenar las claves GPG utilizadas por el sistema de gestión de paquetes APT. Esta acción se realiza con el siguiente comando:

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Ilustración 7. Clave GPG oficial de Docker. Fuente: Imagen elaborada por ADA.

Aquí, la opción `-m 0755` establece los permisos adecuados para el nuevo directorio, otorgando acceso de lectura y ejecución a todos los usuarios, mientras que la escritura se limita únicamente al propietario. Esto garantiza que el sistema mantenga un control de acceso adecuado, reforzando la seguridad del entorno.

A continuación, se procede a descargar la clave GPG oficial directamente desde el servidor de Docker mediante el comando `curl`, que actúa como una herramienta de transferencia de datos desde URLs. La clave se guarda localmente dentro del directorio creado previamente, bajo el nombre `docker.asc`:

```
bash
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

Ilustración 8. Descargar la clave GPG. Fuente: Imagen elaborada por ADA.

Las opciones utilizadas en este comando tienen funciones específicas: `-f` indica que `curl` debe fallar de forma silenciosa en caso de error; `-s` activa el modo silencioso, ocultando la

barra de progreso; -S permite mostrar mensajes de error en caso de fallo; y -L sigue cualquier redirección HTTP que pueda presentarse durante la descarga. Esta configuración asegura una transferencia limpia y sin interrupciones.

Finalmente, se ajustan los permisos del archivo `docker.asc` para garantizar que cualquier usuario pueda leer su contenido, condición necesaria para que el sistema APT pueda utilizar la clave durante la verificación de firmas digitales en los paquetes provenientes del repositorio de Docker:

```
bash

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Ilustración 9. Sistema APT pueda utilizar la clave. Fuente: Imagen elaborada por ADA.

Mediante este comando, se otorgan permisos de lectura a todos los usuarios (`a+r`), asegurando que no existan restricciones en la lectura del archivo por parte de los procesos que intervienen en la instalación y actualización de paquetes.

En conjunto, estos pasos no solo preparan el sistema para utilizar el repositorio oficial de Docker, sino que también establecen una base sólida y segura para la gestión de paquetes firmados criptográficamente, alineándose con las buenas prácticas de administración de sistemas Linux.

12.2.2 Agregue el repositorio de Docker a las fuentes de APT

Con el objetivo de habilitar el sistema para la instalación de Docker directamente desde su fuente oficial, primero incorporo el repositorio correspondiente de Docker a la lista de fuentes confiables del sistema de gestión de paquetes APT. Este procedimiento garantiza que las futuras instalaciones y actualizaciones del software Docker provengan directamente del mantenedor oficial, lo que incrementa la seguridad y confiabilidad del entorno.

Inicio este proceso utilizando un comando `echo` que construye dinámicamente la entrada del repositorio. Esta construcción aprovecha la salida del comando `dpkg --print-architecture` para identificar de manera automática la arquitectura del sistema (por ejemplo, `amd64`, `arm64`, entre otras), lo cual permite asegurar que se descarguen únicamente los paquetes compatibles con la plataforma actual.

Posteriormente, mediante una subshell que evalúa la información contenida en el archivo `/etc/os-release`, obtengo el nombre en clave de la distribución de Ubuntu que se encuentra en ejecución (como `focal`, `jammy`, etc.). Esto se realiza con la instrucción `$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}"),` lo cual garantiza que se

utilice el identificador correcto del sistema operativo para direccionar la versión específica del repositorio Docker que corresponde.

La concatenación de estos valores produce una línea que sigue la sintaxis estándar de las listas de repositorios de APT, e incluye además la instrucción `signed-by=/etc/apt/keyrings/docker.asc`, que especifica el uso de una clave GPG previamente instalada en ese directorio. Esta clave cumple la función de verificar la autenticidad de los paquetes descargados, fortaleciendo así la cadena de confianza del sistema.

A continuación, canalizo esta línea generada hacia el archivo `/etc/apt/sources.list.d/docker.list` utilizando el comando `sudo tee`. Este archivo se convierte en un nuevo origen de paquetes administrado por APT y permite que el sistema consulte directamente los paquetes publicados por Docker Inc. en su repositorio oficial.

Finalmente, ejecuto `sudo apt-get update` con el propósito de actualizar la base de datos local de paquetes del sistema. Este paso es crucial, ya que fuerza al gestor de paquetes a reconocer el nuevo repositorio recién añadido y a incluirlo en las futuras operaciones de búsqueda, instalación y actualización de paquetes. Al realizar esta actualización, el sistema se sincroniza con los metadatos más recientes del repositorio de Docker, asegurando así la disponibilidad de versiones actuales y seguras del software.

```
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://down  
load.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Ilustración 10. Agregue el repositorio de Docker. Fuente: Imagen elaborada por ADA.

13. Instalación de Docker

Para llevar a cabo la instalación del entorno Docker en un sistema operativo basado en Debian o Ubuntu, es fundamental seguir una serie de pasos que garantizan una configuración limpia, segura y funcional. Este proceso permite disponer de una plataforma robusta para la creación, gestión y despliegue de contenedores, facilitando así la implementación de aplicaciones en entornos aislados y reproducibles.

En este paso, se ejecuta el siguiente comando desde la terminal con privilegios de superusuario:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Ilustración 11. Instalar Docker. Fuente: Imagen elaborada por ADA.

Este comando realiza la instalación simultánea de varios componentes esenciales para el funcionamiento completo de Docker:

- **docker-ce (Docker Community Edition):** Constituye el núcleo del motor de Docker. Este paquete incluye el demonio de Docker (dockerd), que se encarga de ejecutar los contenedores en segundo plano, y todas las herramientas necesarias para el manejo de imágenes y contenedores en el sistema.
- **docker-ce-cli (Docker Command Line Interface):** Proporciona la interfaz de línea de comandos docker, a través de la cual el usuario interactúa con el motor de Docker. Esta herramienta permite ejecutar comandos para crear, iniciar, detener, listar y eliminar contenedores, así como construir y gestionar imágenes de contenedor, entre muchas otras funciones.
- **containerd.io:** Instala el runtime de contenedores de bajo nivel utilizado por Docker. Este componente es responsable de la ejecución y supervisión directa de los contenedores, y actúa como una capa intermedia entre el sistema operativo y el motor de Docker, ofreciendo una gestión eficiente y conforme a estándares del ciclo de vida de los contenedores.
- **docker-buildx-plugin:** Incorpora la funcionalidad de BuildKit a través del comando docker buildx, el cual extiende las capacidades de construcción de imágenes. Este plugin permite crear imágenes de contenedor de manera más eficiente, con soporte para arquitecturas múltiples, caché compartido y ejecuciones paralelas.
- **docker-compose-plugin:** Habilita el uso de docker compose como plugin integrado, facilitando el despliegue de aplicaciones multicontenedor definidas en archivos docker-compose.yml. Este plugin simplifica la orquestación de servicios, volúmenes, redes y dependencias en entornos de desarrollo o producción.

Durante la ejecución del comando de instalación, el gestor de paquetes apt resuelve automáticamente las dependencias, descarga los archivos necesarios desde los repositorios oficiales de Docker y configura los servicios requeridos para que el sistema reconozca y utilice Docker correctamente. Es importante asegurarse de que el sistema se encuentra actualizado antes de proceder, y que los repositorios de Docker han sido añadidos previamente de forma segura, lo que garantiza la autenticidad e integridad del software que se descarga e instala.

Una vez finalizada la instalación, se recomienda verificar que el servicio de Docker esté activo y funcionando correctamente mediante el comando `sudo systemctl status docker`. Además, se puede comprobar el correcto funcionamiento de la CLI ejecutando `docker version`, lo cual ofrece información sobre las versiones instaladas del cliente y del demonio.

Este procedimiento constituye la base para el uso posterior de contenedores en proyectos de desarrollo, pruebas y despliegue de aplicaciones modernas en entornos controlados, escalables y portables.

13.1 Configuración de permisos para el usuario Docker

Con el objetivo de optimizar y agilizar la interacción con Docker, es esencial evitar la necesidad de utilizar el prefijo sudo en cada uno de los comandos que ejecutamos a través de esta plataforma de contenedores. Esta práctica no solo mejora la eficiencia de los desarrolladores y administradores de sistemas, sino que también facilita la administración del entorno Docker, evitando la repetición constante de privilegios elevados.

Para lograrlo, el paso inicial consiste en añadir el usuario que desea interactuar con Docker al grupo de usuarios denominado docker, lo que otorga los permisos necesarios para ejecutar los comandos de Docker sin la necesidad de recurrir a sudo. Este paso se puede realizar mediante el uso de la herramienta usermod, que permite modificar los atributos y configuraciones de los usuarios del sistema. En particular, el comando que se utiliza es:

```
sudo usermod -aG docker $USER  
newgrp docker
```

Ilustración 12. Usuario al grupo Docker. Fuente: Imagen elaborada por ADA.

Este comando se desglosa de la siguiente manera:

- sudo: Permite ejecutar el comando con privilegios de superusuario, necesarios para realizar cambios en la configuración del sistema.
- usermod: Herramienta que permite modificar la información de un usuario en el sistema.
- -aG docker: La opción -aG indica que el usuario debe ser añadido (a) al grupo (G) especificado, que en este caso es docker.
- \$USER: Variable que hace referencia al nombre de usuario del que está ejecutando el comando, asegurando que el cambio se aplique al usuario correcto.

Una vez que el usuario ha sido agregado correctamente al grupo docker, es importante aplicar los cambios de manera inmediata sin necesidad de reiniciar el sistema. Para esto, utilizamos el comando:

```
bash
```

```
newgrp docker
```

Ilustración 13. Usuario al grupo Docker. Fuente: Imagen elaborada por ADA.

Este comando cambia temporalmente el grupo de trabajo del usuario a docker, lo que permite que los permisos recién asignados surtan efecto de forma instantánea. Al ejecutar este comando, el usuario puede comenzar a utilizar los comandos de Docker sin la necesidad de emplear sudo, lo que simplifica y agiliza las tareas relacionadas con la gestión de contenedores en el sistema.

Es fundamental recordar que, aunque este procedimiento elimina la necesidad de utilizar sudo para interactuar con Docker, también implica un cambio en la seguridad del sistema. Al otorgar acceso directo a los comandos de Docker, se le confiere al usuario un nivel de privilegio elevado dentro del entorno de contenedores. Por lo tanto, es recomendable aplicar esta configuración solo a usuarios de confianza que necesiten interactuar regularmente con Docker, manteniendo siempre un control estricto sobre los permisos otorgados a cada cuenta en el sistema.

Recomendación: Para que los cambios realizados tengan efecto y sean correctamente aplicados, es esencial cerrar la terminal en uso y proceder a abrirla nuevamente. Esta acción asegura que la sesión actual de la terminal se reinicie, de modo que cualquier modificación en la configuración, variables de entorno o actualizaciones recientes del sistema se carguen adecuadamente en el nuevo entorno de trabajo. Al realizar este paso, se elimina cualquier posible conflicto o desincronización entre el estado actual y el estado actualizado de la terminal, garantizando que todas las configuraciones y ajustes previos sean implementados correctamente en el nuevo inicio.

Este procedimiento es fundamental cuando se efectúan cambios relacionados con la configuración del sistema operativo, la instalación de nuevos programas, la actualización de bibliotecas o la modificación de variables del entorno de trabajo, ya que dichos ajustes a menudo no se reflejan inmediatamente en las sesiones de terminal abiertas previamente. Reabrir la terminal proporciona un entorno limpio y actualizado, asegurando así que todos los cambios recientes se integren sin problemas en las operaciones que se realizarán a continuación.

Este tipo de procedimiento es comúnmente requerido en entornos de desarrollo y administración de sistemas, donde la sincronización de los recursos y configuraciones es crucial para el correcto funcionamiento de las herramientas y aplicaciones utilizadas.

14. Despliegue de la Aplicación

El proceso de despliegue de una aplicación, particularmente en entornos de desarrollo y producción, requiere una serie de pasos que aseguran tanto la portabilidad como la escalabilidad de la misma. Una de las herramientas más eficaces en este tipo de procesos es Docker, que permite empaquetar una aplicación junto con todas sus dependencias en un contenedor independiente. Este contenedor, al ser ejecutado en cualquier entorno compatible con Docker, asegura que la aplicación funcione de manera consistente, independientemente de las diferencias en el sistema operativo o configuraciones de hardware.

14.1 Creación de la Imagen Docker

Para comenzar con la creación de la imagen Docker que representará nuestra aplicación, se siguen una serie de pasos estructurados. Este proceso es crucial para garantizar que la aplicación se empaquete de forma eficiente, cumpliendo con los estándares de operatividad y escalabilidad requeridos.

14.1.1 Navegue hasta la carpeta de la aplicación

El primer paso consiste en posicionarse dentro del directorio raíz de la aplicación en el entorno local donde reside el código fuente y los archivos de configuración relevantes para el proyecto. Este directorio es el punto de partida para la creación de la imagen Docker, ya que desde allí se localizan los archivos esenciales para la configuración del contenedor. Para ello, se ejecuta el siguiente comando en la terminal:

```
cd ruta/de_app/en_local/
```

Ilustración 14. Carpeta de la aplicación. Fuente: Imagen elaborada por ADA.

Este comando `cd` (cambiar directorio) permite acceder al directorio específico donde se encuentra la aplicación, asegurándose de que todos los archivos y configuraciones necesarias sean accesibles en el proceso de construcción de la imagen Docker. En este punto, es importante verificar que el directorio contenga todos los archivos necesarios, como el `Dockerfile`, que define las instrucciones para la creación de la imagen, así como cualquier otro archivo de configuración y dependencias del entorno que serán incorporados al contenedor Docker.

14.1.2 Construcción de la imagen de Docker

El comando `docker build` permite generar una nueva imagen de Docker a partir de un archivo de instrucciones, conocido como `Dockerfile`. Este archivo contiene las configuraciones necesarias que definen el entorno y las dependencias que la imagen debe

incluir. Para iniciar este proceso de construcción, se especifica la ruta hacia el archivo Dockerfile y el nombre o etiqueta de la imagen que se va a crear.

En este caso, el comando se estructura de la siguiente manera:

```
docker build -f Dockerfile -t usuariodocker/municipios-viewer:aws .
```

Ilustración 15. Para construir la imagen de Docker. Fuente: Imagen elaborada por ADA.

14.1.3 Suba la imagen al repositorio de Docker

Para subir una imagen al repositorio de Docker, se debe utilizar el comando docker push, el cual permite enviar una imagen previamente construida y etiquetada a un repositorio remoto de Docker, como el que puede ser proporcionado por Docker Hub o por otros servicios de contenedores. En este caso, se emplea el siguiente comando:

```
docker push usuariodocker/municipios-viewer:aws
```

Ilustración 16. Suba la imagen al repositorio de Docker. Fuente: Imagen elaborada por ADA.

Este comando realiza la transferencia de la imagen denominada municipios-viewer, asociada a la etiqueta aws, al repositorio remoto correspondiente, en este caso, especificado bajo el nombre de usuario usuariodocker. El proceso comienza con la verificación de que la imagen esté correctamente etiquetada y construida en el entorno local antes de ser enviada.

Es importante tener en cuenta que el comando docker push requiere que se haya iniciado sesión previamente en el repositorio remoto con las credenciales adecuadas. Para autenticarte en Docker Hub o cualquier otro servicio de repositorios, se debe utilizar el comando docker login, proporcionando las credenciales correctas. Si la autenticación se realiza de manera exitosa, el sistema permite realizar la operación de subida sin problemas adicionales.

Una vez que el comando docker push es ejecutado, Docker inicia el proceso de subir la imagen. Este proceso implica la verificación de las capas (layers) de la imagen, que se suben de manera incremental. Si alguna capa ya está presente en el repositorio remoto, Docker la reutiliza, evitando la transferencia redundante de datos. Sin embargo, si alguna de las capas no existe, se transfiere al repositorio, asegurando que la imagen completa esté disponible para ser utilizada en cualquier otro entorno que consuma el repositorio.

El proceso de subida se puede monitorizar a través de la salida en consola, donde se muestra el progreso de la transferencia de cada capa. Una vez completada la operación, la imagen queda disponible en el repositorio remoto bajo la etiqueta especificada, lo que facilita su posterior descarga y uso en otros entornos o servidores.

14.2 Creación del archivo docker-compose.yml

14.2.1 Navegación hacia el directorio /app

El primer paso para proceder con la creación del archivo docker-compose.yml consiste en acceder a un directorio específico dentro del sistema de archivos. Este directorio se encuentra en la ruta /app, que es la ubicación designada para realizar las configuraciones y la orquestación de los contenedores que se gestionarán posteriormente a través de Docker Compose.

Para realizar esta acción, se ejecuta el siguiente comando en la terminal o consola de comandos del sistema operativo:

```
cd /app
```

Ilustración 17. Directorio /app. Fuente: Imagen elaborada por ADA.

El comando `cd` (change directory) es una instrucción estándar utilizada en sistemas basados en UNIX y otros entornos de línea de comandos. Su propósito es cambiar el directorio de trabajo actual a la ruta especificada. Al ingresar `/app`, se indica al sistema que se desea navegar al directorio raíz `/` y luego acceder al subdirectorio denominado `app`. Es importante que este directorio exista previamente y contenga los archivos relevantes para la configuración de los contenedores.

Este paso es fundamental, ya que en el directorio `/app` es donde se almacenará el archivo `docker-compose.yml`, el cual es un archivo de configuración utilizado por Docker Compose para definir y ejecutar aplicaciones multi-contenedor. De no encontrarse en el directorio correcto, no se podrá realizar la correcta configuración del entorno de contenedores.

Una vez que se navega correctamente al directorio `/app`, es posible proceder con la creación del archivo `docker-compose.yml`, que establecerá los parámetros de configuración necesarios para la creación y el manejo de los contenedores Docker asociados al proyecto en cuestión.

Este proceso establece la base para la implementación eficiente de los servicios y contenedores dentro del ecosistema Docker, asegurando que las configuraciones sean consistentes y replicables.

14.2.2 Cree y edite el archivo Docker -compose.yml

Para realizar la creación y edición del archivo `docker-compose.yml`, se siguen una serie de pasos que permiten configurar adecuadamente un entorno de contenedores utilizando Docker Compose, una herramienta esencial para definir y ejecutar aplicaciones multi-contenedor. A continuación, se expone de manera técnica y detallada cómo proceder:

1. **Creación del archivo Docker Compose:** En primer lugar, se accede a la terminal del sistema operativo donde se tiene instalado Docker. Dentro de la terminal, se

utiliza el comando `mkdir` para crear un nuevo directorio, que servirá como espacio de trabajo donde se almacenará el archivo `docker-compose.yml`. Este directorio es crucial para organizar los archivos y asegurar que la estructura de la aplicación esté correctamente definida.

El comando `mkdir docker-compose.yml` debe ser ejecutado en el directorio adecuado del sistema de archivos. Este paso asegura que el archivo `docker-compose.yml` esté en la ubicación correcta, optimizando el acceso y la gestión de los recursos relacionados con el contenedor.

2. **Edición del archivo Docker Compose:** Una vez creado el directorio necesario, se pasa a la edición del archivo. Para ello, se utiliza el editor de texto `nano` que permite realizar modificaciones en archivos de manera sencilla y directa desde la terminal. El comando `nano docker-compose.yml` abre el archivo de configuración en el editor, permitiendo la edición del mismo.

Dentro de este archivo, se especifican diversos parámetros esenciales para la configuración de los servicios dentro del contenedor, como la definición de imágenes, redes, volúmenes y variables de entorno. La sintaxis YAML es clave para definir correctamente cada uno de estos aspectos, y cualquier error en su redacción puede llevar a fallos en la ejecución de los contenedores.

3. **Configuración detallada del archivo:** En el archivo `docker-compose.yml` se deben incluir varios elementos que definen el comportamiento de la aplicación, tales como la versión de Docker Compose, los servicios que se desean ejecutar, las dependencias entre contenedores, y la configuración de puertos, volúmenes y redes.
 - **Versión:** Especifica la versión del archivo `docker-compose.yml` que se está utilizando, lo que permite garantizar la compatibilidad con las características del sistema.
 - **Servicios:** Cada servicio dentro del archivo representa un contenedor individual que se ejecutará, permitiendo que Docker Compose gestione de manera eficiente el ciclo de vida de cada uno.
 - **Redes:** Se pueden definir redes personalizadas para permitir la comunicación entre los contenedores, asegurando que los servicios puedan interactuar de forma segura y optimizada.
 - **Volúmenes:** Permiten persistir datos fuera de los contenedores, de modo que los datos se conserven incluso si los contenedores se detienen o se reinician.
4. **Guardado y cierre:** Tras realizar todas las modificaciones necesarias, se guarda el archivo y se cierra el editor. En `nano`, esto se logra presionando las teclas `Ctrl + O` para guardar y `Ctrl + X` para salir.

En resumen, la creación y edición del archivo `docker-compose.yml` implica una serie de pasos bien definidos: desde la creación de un directorio adecuado mediante `mkdir`, hasta la edición del archivo usando un editor de texto como `nano`, garantizando una correcta configuración de los contenedores y servicios para una ejecución fluida y eficiente de la aplicación en Docker.

```
nano docker-compose.yml
```

Ilustración 18. Archivo Docker -compose.yml. Fuente: Imagen elaborada por ADA.

14.2.3 Transfiera el contenido del archivo `docker-compose.yml` desde su entorno local al servidor remoto

El primer paso consiste en copiar de manera eficiente el archivo de configuración `docker-compose.yml` desde su máquina local hacia el servidor donde se llevará a cabo el despliegue de los servicios definidos en el archivo. Este archivo, fundamental para la orquestación de contenedores Docker, contiene las instrucciones y configuraciones necesarias para la creación, ejecución y administración de los servicios de una aplicación basada en contenedores. El proceso de transferencia del archivo se puede realizar a través de diversos métodos, dependiendo de la infraestructura disponible. En escenarios típicos, se utiliza el protocolo SSH (Secure Shell) junto con herramientas como `scp` o `rsync`, las cuales permiten realizar una copia remota segura y eficiente.

Al ejecutar el comando correspondiente, es crucial verificar que el archivo se haya copiado correctamente al servidor, asegurando que no haya pérdidas de información ni alteraciones del contenido del archivo de configuración durante el proceso. Para ello, se recomienda revisar la ruta de destino en el servidor y confirmar que el archivo mantiene la integridad y estructura adecuada, garantizando su funcionalidad en el proceso de despliegue.

14.2.4 Proceda a ejecutar el despliegue utilizando el comando adecuado

Una vez que el archivo `docker-compose.yml` ha sido correctamente copiado al servidor, el siguiente paso es iniciar el despliegue de los contenedores definidos en dicho archivo. Esto se lleva a cabo utilizando el comando `docker compose up -d`, el cual activa la creación y el arranque de los servicios descritos en el archivo de configuración. El parámetro `-d` se utiliza para ejecutar los contenedores en segundo plano, lo que permite que el terminal quede libre para otros procesos, sin bloquearse por la salida de los contenedores en ejecución.

El comando `docker compose up` evalúa el archivo `docker-compose.yml`, crea las redes necesarias, construye las imágenes de los contenedores (si es necesario), y finalmente inicia los contenedores, permitiendo que los servicios definidos en el archivo se pongan en funcionamiento de manera automática. Este proceso de despliegue asegura que todos los servicios estén alineados con las configuraciones establecidas, proporcionando un entorno de ejecución controlado y repetible.

Es recomendable revisar los logs de los contenedores después de la ejecución para identificar posibles errores o problemas en el arranque de los servicios. Este monitoreo garantiza que el entorno de despliegue se haya configurado correctamente y que los servicios se inicien sin inconvenientes.

```
docker compose up -d
```

Ilustración 19. Despliegue. Fuente: Imagen elaborada por ADA.

14.3 Configuración y Ejecución de la Aplicación

En el contexto de la gestión de aplicaciones dentro de entornos de contenedores, resulta fundamental conocer los pasos necesarios para reiniciar y poner en marcha correctamente una aplicación. A continuación, se detallan los procedimientos técnicos específicos que deben llevarse a cabo para reiniciar la aplicación de manera eficiente utilizando Docker y Docker Compose.

14.3.1 Procedimiento para reiniciar la aplicación

Cuando sea necesario reiniciar la aplicación, ya sea por actualizaciones de configuración, cambios en el entorno o por cualquier otra razón que requiera una reconfiguración del contenedor en ejecución, es imprescindible seguir un proceso estructurado para garantizar que la aplicación se reinicie de manera controlada. El procedimiento se realiza a través de dos comandos específicos que Docker Compose gestiona de forma óptima, los cuales se describen a continuación.

1. Detener la aplicación y eliminar los contenedores en ejecución:

El primer paso en este proceso de reinicio es detener la ejecución de todos los contenedores asociados a la aplicación y eliminar sus instancias actuales. Este paso se lleva a cabo utilizando el comando:

```
docker compose down  
docker compose up -d
```

Ilustración 20. Reiniciar la aplicación Fuente: Imagen elaborada por ADA.

Este comando de Docker Compose asegura que todos los contenedores, redes, y volúmenes definidos en el archivo docker-compose.yml sean detenidos y eliminados de manera ordenada. Este comando también garantiza que se liberen los recursos que estaban siendo utilizados por los contenedores, permitiendo que el siguiente ciclo de ejecución inicie con un entorno limpio y sin conflictos previos.

2. Iniciar la aplicación en segundo plano:

Una vez que los contenedores han sido detenidos y eliminados correctamente, el siguiente paso es proceder con el reinicio de la aplicación, que se lleva a cabo mediante el siguiente comando:

```
bash

docker compose up -d
```

Ilustración 21. Iniciar la aplicación en segundo plano. Fuente: Imagen elaborada por ADA.

Este comando inicia todos los servicios y contenedores especificados en el archivo `docker-compose.yml` en segundo plano. El flag `-d` indica que los contenedores se deben ejecutar en modo *detached*, es decir, sin bloquear la terminal actual, lo que permite al usuario continuar trabajando en otras tareas sin interrupciones. Docker Compose se encarga de verificar las configuraciones definidas en dicho archivo y de crear los contenedores necesarios para la ejecución de la aplicación.

Este proceso no solo garantiza el reinicio correcto de la aplicación, sino que también asegura que cualquier cambio en la configuración o en el entorno de los contenedores sea aplicado de manera efectiva.

14.4 Verificación

La fase de verificación del despliegue es un proceso crucial que asegura que todas las configuraciones, componentes y funciones del sistema se implementan correctamente y de acuerdo con los estándares definidos previamente. Es fundamental que, en este proceso, se revisen todos los aspectos involucrados en la implementación para confirmar que no existen errores, fallas o inconsistencias que puedan afectar el rendimiento del sistema o su capacidad para cumplir con los requisitos establecidos.

Durante esta etapa, se lleva a cabo una inspección detallada de cada uno de los elementos que componen el despliegue. Esto incluye no solo la comprobación de los componentes hardware y software involucrados, sino también la validación de las interacciones entre los distintos sistemas, la integración de los módulos y la correcta funcionalidad de las interfaces de usuario. Cada uno de estos aspectos debe ser verificado a través de procedimientos específicos que permiten identificar posibles desviaciones de las especificaciones técnicas y operativas.

Además, la verificación debe incluir una evaluación del entorno en el que el sistema va a operar, asegurando que las condiciones de infraestructura, redes y otros recursos necesarios se encuentren dentro de los parámetros establecidos. Esto permite anticipar cualquier problema que pueda surgir una vez que el sistema esté en funcionamiento y realizar ajustes preventivos para garantizar su estabilidad y eficiencia a largo plazo.

La verificación debe realizarse en varias etapas, comenzando por una comprobación inicial de los componentes básicos del sistema, seguida de una revisión exhaustiva de los procesos de integración y prueba, para finalmente realizar pruebas de validación en escenarios reales de uso. Cada una de estas fases debe documentarse de manera detallada, registrando los resultados obtenidos, las acciones correctivas realizadas y los ajustes necesarios para garantizar que el despliegue sea exitoso y cumpla con todos los requerimientos de calidad y desempeño.

Este proceso no solo valida la correcta implementación del sistema, sino que también proporciona un marco para la mejora continua, permitiendo identificar áreas que puedan ser optimizadas en futuras implementaciones o actualizaciones. La verificación debe ser un proceso iterativo, donde se realicen evaluaciones constantes y se ajusten los procedimientos según sea necesario, garantizando que el despliegue se realice de manera eficiente, sin comprometer la seguridad, la calidad o la funcionalidad del sistema.

```
C:\Windows\System32\cmd.exe
C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>docker build -t frontend_fenog:latest .
[+] Building 1.9s (15/15) FINISHED
=> [internal] load build definition from Dockerfile
=> -- transferring dockerfile: 531B
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [internal] load metadata for docker.io/library/node:20-alpine
=> [internal] load .dockerignore
=> transferring context: 2B
=> [stage-1 1/3] FROM docker.io/library/nginx:alpine@sha256:65645c7bda86d189248b3b8d0743288a18dd2f3f17554ef4b76fb8e2f2a10
=> -- resolve docker.io/library/nginx:alpine@sha256:65645c7bda86d189248b3b8d0743288a18dd2f3f17554ef4b76fb8e2f2a10
=> [builder 1/6] FROM docker.io/library/node:20-alpine@sha256:abfde7f5712172a9c58b8947244722f6faf2876220cada0e444d26ea31c92cf
=> -- resolve docker.io/library/node:20-alpine@sha256:abfde7f5712172a9c58b8947244722f6faf2876220cada0e444d26ea31c92cf
=> [internal] load build context
=> transferring context: 13.43kB
=> CACHED [builder 2/6] WORKDIR /app
=> CACHED [builder 3/6] COPY package*.json ./
=> CACHED [builder 4/6] RUN npm ci
=> CACHED [builder 5/6] COPY . .
=> CACHED [builder 6/6] RUN npm run build
=> CACHED [stage-1 2/3] COPY --from=builder /app/dist /usr/share/nginx/html
=> CACHED [stage-1 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf
=> exporting to image
=> exporting layers
=> exporting manifest sha256:45cc10029c3623aef321b73cfff28a5a5fe50e9a9ff984feed10d23093fdd138
=> exporting config sha256:134a125301001fa975d8028e427caedd426f10faede990b9f9c242ce144ce
=> exporting attestation manifest sha256:c70d83a0ac8f7a84d639d93ad07cfff65ad19c5ad16558091ca2cb950ecc0de
=> exporting manifest list sha256:29cdc8fa6f37585214ff7c2827dae2dd93925ebac78620794d755b6b779ffe1
=> naming to docker.io/library/frontend_fenog:latest
=> unpacking to docker.io/library/frontend_fenog:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ljgcw28pwwv54khyz6m0dxtt

1 warning found (use docker --debug to expand):
- FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)

C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
frontend_fenog      latest      29cdc8fa6f37  18 minutes ago  90.0MB
backend_fenog       latest      3d10e9ef2a30  8 days ago    340MB
postgres            17.2       f4eefc6901dd  5 months ago  614MB

C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>docker save -o frontend_fenog.tar frontend_fenog:latest
```

Ilustración 22. Despliegue. Fuente: Imagen elaborada por ADA.

```

=> CACHED [builder 4/6] RUN npm ci
=> CACHED [builder 5/6] COPY . .
=> CACHED [builder 6/6] RUN npm run build
=> CACHED [stage-1 2/3] COPY --from=builder /app/dist /usr/share/nginx/html
=> CACHED [stage-1 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf
=> exporting to image
=> exporting layers
=> exporting manifest sha256:45ce18029c3623aefa321b73cfff28a5a5fe50a9a9ff904feed10d23903fdd138
=> exporting config sha256:134a12530100b1fa975d8028e427caedd42eef10faede99b09fcc242ce144cee
=> exporting attestation manifest sha256:c70d83a60ac8f7a84d639d93ad07cfff5ad19c55ad16558091ca2cb950ecc0de
=> exporting manifest list sha256:29cdc8fa6f37585214ff7c2827dae2dd939256ebac78620794d755b0b779ffe1
=> naming to docker.io/library/frontend_fenoge:latest
=> unpacking to docker.io/library/frontend_fenoge:latest

View build details: docker-desktop:///dashboard/build/desktop-linux/desktop-linux/ljgcw28pwwv54khyz6m0dwxxt

! warning found (use docker --debug to expand):
- FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)

C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
frontend_fenoge     latest             29cdc8fa6f37       18 minutes ago     90.8MB
backend_fenoge      latest             3d10efef3639       8 days ago         340MB
postgres            17.2               fe4efc6901dd       5 months ago       614MB

C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>docker save -o frontend_fenoge.tar frontend_fenoge:latest

C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog>dir
El volumen de la unidad C es Windows 11 HSL
El número de serie del volumen es: 4803-21B4

Directorio de C:\Users\Jair Lizcano\OneDrive\Escritorio\Frontend_Fenog
15/05/2025 08:02 p. m.    <DIR>          .
15/05/2025 07:58 p. m.    <DIR>          ..
14/05/2025 02:19 p. m.    291          .env
14/05/2025 02:19 p. m.    109          .env.production
14/05/2025 02:19 p. m.    277          .gitignore
14/05/2025 02:20 p. m.    <DIR>          Content
14/05/2025 02:19 p. m.    684          docker-compose.yml
14/05/2025 02:19 p. m.    492          Dockerfile
14/05/2025 02:19 p. m.    767          eslint.config.js
15/05/2025 08:02 p. m.    24,482.816   frontend_fenoge.tar
14/05/2025 02:19 p. m.    790          index.html
14/05/2025 02:19 p. m.    5,613        Indicadores.html
14/05/2025 02:19 p. m.    637          nginx.conf
14/05/2025 02:19 p. m.    205,192      package-lock.json
14/05/2025 02:19 p. m.    1,173        package.json
14/05/2025 02:19 p. m.    87           postcss.config.js
14/05/2025 02:20 p. m.    <DIR>          public
14/05/2025 02:19 p. m.    1,514        README.md
14/05/2025 02:19 p. m.    4,916        README_ARQUITECTURA.md
14/05/2025 02:19 p. m.    <DIR>          ...

```

Ilustración 23. Despliegue. Fuente: Imagen elaborada por ADA.

14.5 Solución de problemas

14.5.1 Error de permisos

Cuando se experimenta un problema relacionado con los permisos al intentar ejecutar Docker sin utilizar el comando sudo, el sistema puede estar indicando que el usuario actual no pertenece al grupo adecuado para ejecutar Docker directamente. En un escenario típico, Docker requiere que el usuario esté incluido en el grupo de usuarios denominado docker, ya que este grupo tiene los permisos necesarios para interactuar con el demonio de Docker y ejecutar contenedores. Si el sistema no reconoce al usuario como parte de este grupo, es posible que se presenten errores de permisos al intentar ejecutar comandos de Docker sin privilegios elevados.

En este caso, una posible solución consiste en reiniciar la sesión del usuario para que el sistema reconozca correctamente los cambios realizados en los grupos a los que pertenece el usuario. Sin embargo, si no se desea cerrar la sesión y se necesita aplicar los cambios de manera inmediata sin necesidad de reiniciar, se puede ejecutar el siguiente comando:

```
newgrp docker
```

Ilustración 24. Error de permisos. Fuente: Imagen elaborada por ADA.

Este comando actualiza el grupo primario del usuario en la sesión actual, permitiendo que los cambios en la pertenencia al grupo docker se apliquen sin necesidad de un reinicio

completo del sistema. Al ejecutar este comando, el usuario puede continuar trabajando sin tener que utilizar sudo para ejecutar comandos relacionados con Docker, ya que los permisos asociados al grupo docker se aplican instantáneamente en la sesión activa. Este procedimiento asegura que Docker pueda operar correctamente sin requerir privilegios de superusuario, mejorando la seguridad y la fluidez de la interacción del usuario con el sistema.

Es importante señalar que, en ocasiones, puede ser necesario verificar que el grupo docker exista en el sistema y que el usuario esté efectivamente agregado a este grupo. Si el problema persiste después de ejecutar el comando mencionado, es recomendable revisar la configuración de los grupos en el sistema y confirmar que no haya restricciones adicionales que impidan el acceso adecuado a los recursos de Docker.

14.5.2 Puertos bloqueados

Para verificar si un puerto específico está siendo bloqueado o si existe alguna restricción en su uso dentro de la red local o del sistema operativo, se debe realizar una consulta utilizando el siguiente comando de la terminal:

```
sudo netstat -tulnp | grep <puerto>
```

Ilustración 25. Puertos bloqueados. Fuente: Imagen elaborada por ADA.

14.5.3 Problemas con docker-compose

Asegúrese de usar la versión correcta:

Para abordar de manera adecuada los problemas relacionados con Docker Compose, es esencial que el usuario verifique con precisión la versión de la herramienta que está utilizando, ya que las versiones incorrectas pueden generar incompatibilidades o errores en la ejecución de los servicios definidos en los archivos docker-compose.yml. Docker Compose es una herramienta fundamental que facilita la definición y ejecución de aplicaciones multi-contenedor, y su correcto funcionamiento depende de que se utilice la versión adecuada, que sea compatible con los parámetros y configuraciones especificadas en los archivos.

Para confirmar que la versión de Docker Compose instalada es la correcta, se recomienda ejecutar el comando `docker compose version` en la línea de comandos. Este comando proporciona información detallada sobre la versión instalada, lo que permite al usuario determinar si es necesario actualizar o ajustar la herramienta para que coincida con los requisitos específicos del entorno o de la aplicación en cuestión.

Es importante recordar que Docker Compose ha pasado por varias actualizaciones a lo largo del tiempo, y algunas funcionalidades pueden variar según la versión utilizada. Por lo tanto, al utilizar una versión incompatible o desactualizada, el usuario puede encontrarse con problemas como la incapacidad de ejecutar ciertos servicios, fallos en la conexión entre

contenedores o la ejecución incorrecta de las configuraciones declaradas. Para evitar estos inconvenientes, es esencial asegurarse de que la versión de Docker Compose coincida con la recomendada por la documentación oficial o por el entorno de desarrollo en el que se trabaja.

En resumen, para solucionar problemas relacionados con Docker Compose, es indispensable verificar que la versión instalada sea la correcta, utilizando el comando `docker compose version`. Esto garantiza que los servicios definidos en los contenedores se ejecuten de manera eficiente y sin interrupciones.

```
docker compose version
```

Ilustración 26. Problemas con docker-compose. Fuente: Imagen elaborada por ADA.

15. Aprobación

Esta sección da fe de la aprobación oficial de este documento. Por favor, firme en el espacio establecido para verificar y dejar constancia de su conformidad con el contenido.

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