

Componente 1. Entregable 1.4.5 Ministerio de Minas y Energía

Proyecto: CTO No. 116231-225-2024

Documento de memoria técnica que incluye la planificación de las etapas de la solución.

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

Contenido

1. Introducción	5
2. Objetivo	5
3. Identificación de la necesidad	5
4. Cumplimiento Normativo y Accesibilidad	6
4.1 Marco Legal	6
4.1.1 Accesibilidad	6
4.1.2 Transparencia	6
5. Definición del Aplicativo Web	6
5.1 Objetivo General	6
5.2 Objetivos Específicos	6
5.3 Alcance	6
5.4 Diseño de la Solución	7
5.5 Arquitectura Tecnológica	7
5.5.1 Capa de Presentación (Frontend)	7
5.5.2 Capa Lógica (Backend)	8
5.5.3 Capa de Datos	8
5.6 Diagrama de Flujo General del Aplicativo web	9
5.6.1 Explicación del paso a paso del flujo: detallado	9
5.6.2 Explicación del paso a paso del flujo resumido	11
5.7 Funcionalidad del Aplicativo	11
5.7.1 Carga de Datos: Carga y Persistencia desde Archivos Excel y Shapefiles	11
5.7.1.1 Carga total o actualización de todos los datos	12
5.7.1.2 Inclusión de otro determinante o indicador	12
5.7.2 Tecnologías principales	12
5.7.3 Visualización Geoespacial: Exploración Interactiva del Territorio	12
5.7.4 Tecnologías utilizadas	13
5.8 Filtrado y Consulta: Búsquedas por Indicadores, determinantes y localización	13
5.8.1 Tecnologías utilizadas	13
5.9 Comparación: Análisis Comparativo entre Unidades Territoriales	14
5.9.1 Tecnologías empleadas	14
5.10 Reportes: Generación de Informes en PDF y Excel	14
5.10.1 Tecnologías utilizadas	15

6. Estructura Base del Proyecto Web.....	15
6.1 Estructura de componentes	16
6.2 Interfaces Principales	17
6.2.1 Página de Inicio (Landing Page).....	17
6.2.2 Zonas	18
6.2.3 Lista Recurrente	19
6.2.4 Pantalla de Determinantes	19
6.2.5 Pantalla de Comparación	20
6.2.6 Pantalla de Reportes	20
6.2.7 Pantalla indicadores	22
7. Desarrollo del Aplicativo Web.....	23
7.1 Metodología ágil usada en el desarrollo del proyecto.....	23
7.2 Tecnologías y Herramientas	23
7.2.1 Detalle control de Versionamiento herramientas usadas para el aplicativo web	24
8. Proceso de Desarrollo	24
8.1 Etapa de Recolección de Requisitos	24
8.1.1 Actividades	24
8.1.2 Entregables	25
8.2 Diseño del Aplicativo web.....	25
8.2.1 Actividades	25
8.2.2 Entregables	25
8.3 Puesta en Práctica	26
8.3.1 Actividades	26
8.3.2 Entregables	26
9. Pruebas	26
9.1 Actividades.....	26
9.2 Entregables	27
10. Despliegue.....	27
10.1 Actividades.....	27
10.2 Entregables.....	27
11. Paso a paso del Proceso de Desarrollo	28
11.1 Inicio del Proyecto	28

11.2 Análisis de Requerimientos	28
11.3 Diseño del Sistema	28
11.4 Implementación y Codificación	29
11.5 Pruebas y Validación	29
11.6 Entrega / Cierre del Proyecto	29
12. Compatibilidad con navegadores y dispositivos móviles modernos	30
13. Acompañamiento Técnico	30
13.1 Seguimiento del Progreso: QA (Quality Assurance)	30
13.2 Presentación al líder funcional del Ministerio de Minas y Energía	30
13.3 Pruebas del Aplicativo web	31
13.3.1 Pruebas de aceptación	31
14. Mínimo que se realiza	31
15. Implementación en el ambiente de desarrollo del MME	31
15.1 Prerrequisitos	31
15.2 Instalación de Docker en Linux/Ubuntu	32
15.3 Despliegue de la Aplicación	32
15.4 Verificación y Solución de Problemas	32
16. Aprobación	33

Índice de ilustraciones

Ilustración 1. Estructura de carpetas. Fuente: Imagen elaborada por ADA.	15
Ilustración 2. Estructura de componentes. Fuente: Imagen elaborada por ADA.	17
Ilustración 3. Interfaz página de inicio. Fuente: Imagen elaborada por ADA.	18
Ilustración 4. Interfaz Zonas. Fuente: Imagen elaborada por ADA.	18
Ilustración 5. Listas recurrentes. Fuente: Imagen elaborada por ADA.	19
Ilustración 6. Interfaz determinantes. Fuente: Imagen elaborada por ADA.	19
Ilustración 7. Pantalla de comparación. Fuente: Imagen elaborada por ADA.	20
Ilustración 8. Pantalla de reporte PDF. Fuente: Elaborado por ADA.	21
Ilustración 9. Exportar determinantes Excel. Fuente: Elaborado por ADA.	21
Ilustración 10. Exportar indicadores. Fuente: Elaborado por ADA.	22
Ilustración 11. Indicadores. Fuente: Elaborado por ADA.	22

1. Introducción

El contrato CTO No. 116231-225-2024, suscrito entre ADA S.A.S y el Ministerio de Minas y Energía, tiene como objetivo desarrollar un aplicativo web para el Ministerio de Minas y Energía - MME, que provea información abierta sobre indicadores y determinantes territoriales relevantes para la toma de decisiones en proyectos del sector minero-energético dentro del marco de la Transición Energética Justa - TEJ.

Los datos provendrán exclusivamente de matrices en Excel y Shapefile de municipios, departamentos y polígonos suministrada por el Ministerio de Minas y Energía - MME.

2. Objetivo

El presente proyecto se diseñó y definió las bases para la construcción de un aplicativo web de acceso que permita la reposición, visualización y consulta dinámica de la información de interés estratégico correspondiente al subsector de la minería-energía colombiana. Se propuso construir el aplicativo como un sistema integral de información enfocado a la recopilación y difusión de información relevante sobre los determinantes territoriales y sus subcategorías, así como indicadores socioeconómicos. Por medio de esta plataforma, se desea ofrecer a los públicos administrativos, privados y de la sociedad civil un mecanismo o aplicativo para el acceso en tiempo real a la información confiable y verificable en el marco de la Transición Energética Justa.

3. Identificación de la necesidad

El proyecto surge de la necesidad del Ministerio de Minas y Energía (MME) de contar con una herramienta digital que centralice, estructure y facilite el acceso público a información crítica del sector minero-energético, en el marco de la Transición Energética Justa (TEJ). Esta necesidad se fundamenta en los siguientes aspectos clave:

3.1 Centralización de Datos Estratégicos

- **Problemática:** La información relevante para proyectos del sector (determinantes territoriales e indicadores socioeconómicos) está dispersa en múltiples fuentes (Excel, shapefiles, sistemas internos) y entidades (ANH, UPME, Corporaciones Autónomas Regionales, municipios).
- **Necesidad:** Integrar datos de niveles **nacional, departamental y municipal** en una plataforma única, asegurando su disponibilidad y accesibilidad para consultas públicas.

4. Cumplimiento Normativo y Accesibilidad

4.1 Marco Legal

Alineación con políticas de **Gobierno Digital**, incluyendo:

4.1.1 Accesibilidad

Diseño responsivo compatible con dispositivos móviles y navegadores modernos.

4.1.2 Transparencia

Consulta pública sin restricciones de registro.

5. Definición del Aplicativo Web

5.1 Objetivo General

Diseñar y desarrollar un aplicativo web que brinde acceso libre y organizado a información territorial clave, estructurada en torno a indicadores y determinantes del sector minero-energético. Este aplicativo web público debe servir como un instrumento de información.

5.2 Objetivos Específicos

Con el fin de cumplir con el objetivo general, se establecen los siguientes objetivos específicos, los cuales articulan los componentes funcionales y técnicos requeridos para lograr un aplicativo robusto y eficiente:

- Brindar a los usuarios una experiencia de navegación intuitiva, que permita consultar datos estructurados mediante filtros avanzados y funcionalidades de búsqueda inteligente.
- Habilitar la comparación interactiva de información entre dos departamentos o dos municipios, facilitando análisis territoriales y sectoriales diferenciados.
- Incorporar opciones para visualizar, generar y descargar reportes dinámicos en distintos formatos como PDF o Excel, incluyendo gráficos y mapas personalizados.

5.3 Alcance

El desarrollo del proyecto contempla todas las fases necesarias para entregar un aplicativo web tecnológico completo y funcional, que permita la consulta, visualización y análisis de datos tanto tabulares como geoespaciales, relacionados con indicadores estratégicos del sector minero-energético. El alcance incluye:

- Implementación de interfaces para la consulta de información en formatos estructurados (tablas) y geoespaciales (mapas interactivos).

- Desarrollo de filtros para explorar determinantes e indicadores socioeconómicos por regiones, departamentos y municipios.
- Realizar consultas a través del ingreso de coordenadas latitud y longitud, diligenciando los campos respectivos para ello, el sistema ubicará en el mapa el polígono correspondiente de acuerdo a los valores ingresados.
- Realizar consultas manuales que se realicen a través del botón “Selección Manual” y consultas realizadas desde el menú de filtros de los determinantes, tendrán la funcionalidad de que al dar clic en cualquier zona del mapa se mostrará un tooltip con los datos del polígono, incluyendo las coordenadas X, Y, representadas como latitud y longitud.
- Generación de reportes ajustables según los filtros aplicados, con la inclusión de mapas, gráficos y tablas.
- Habilitación de mecanismos para exportar la información procesada en formatos estándar como PDF y Excel.
- Adaptación del aplicativo web para ser funcional y accesible desde distintos navegadores modernos y dispositivos móviles, garantizando una experiencia de usuario coherente y responsiva.

5.4 Diseño de la Solución

El diseño propuesto parte de una arquitectura modular, que distribuye las responsabilidades del aplicativo web en capas especializadas para lograr eficiencia operativa, escalabilidad futura y cumplimiento de altos estándares de seguridad. Cada componente técnico y funcional se orienta a garantizar el correcto desempeño del aplicativo en ambientes de uso público.

5.5 Arquitectura Tecnológica

La arquitectura seleccionada responde a criterios de flexibilidad, interoperabilidad y sostenibilidad tecnológica. Está compuesta por las siguientes capas:

5.5.1 Capa de Presentación (Frontend)

Corresponde a la interfaz visual que los usuarios finales utilizan para interactuar con el aplicativo web. Su diseño se fundamenta en principios de experiencia de usuario (UX) y diseño de interfaz (UI) modernos:

- **React 18:** Framework de JavaScript moderno que facilita el desarrollo de interfaces dinámicas y reactivas.

- **TypeScript:** Lenguaje que añade tipado estático a JavaScript, mejorando la mantenibilidad y calidad del código.
- **Tailwind CSS:** Aplicativo web de estilos basado en clases utilitarias que permite diseñar interfaces de forma rápida y coherente.
- **OpenLayers:** Librería de código abierto especializada en visualización de mapas geográficos.
- **Lucide React:** Conjunto de íconos minimalistas y accesibles que enriquecen la interfaz del usuario.

Nota: Para conocer el detalle, ver Anexo 1. Documentación técnica del código Front End.

5.5.2 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.

5.5.3 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.

5.6 Diagrama de Flujo General del Aplicativo web

El diagrama de flujo general ilustra la arquitectura y el proceso de interacción de un aplicativo web diseñado para gestionar y visualizar datos geoespaciales de manera eficiente y segura. Su objetivo principal es brindar una experiencia intuitiva al usuario final mediante una interfaz visual dinámica, respaldada por un sistema robusto que integra tecnologías front-end, back-end y bases de datos especializadas en información espacial.

Ver anexo 5. Diagrama de flujo Carga de Datos para conocer el detalle.

5.6.1 Explicación del paso a paso del flujo: detallado

1. Usuario Web

- Inicia una acción (ejemplo: consulta de datos, visualización de mapa).
- Interactúa con la interfaz gráfica proporcionada por el Frontend.

2. Frontend

- Tecnologías: React.js y Tailwind CSS.
- Funciones:
 - Renderiza la interfaz visual (formularios, botones, elementos interactivos).
 - Muestra un mapa interactivo con datos geoespaciales (usando bibliotecas como Leaflet).
- Proceso:
 - Al detectar una acción del usuario (ejemplo: clic en el mapa, envío de formulario), genera una solicitud RESTful (API) al Backend.

3. Backend

- Tecnologías: Python (frameworks como Django).
- Funciones:
 - Recibe la solicitud HTTP desde el Frontend.
 - Aplica lógica de negocio:
 - Valida los datos enviados (ejemplo: campos requeridos, formatos correctos).
 - Ejecuta cálculos o procesamiento específicos (ejemplo: análisis espacial, filtros de datos).
 - Maneja seguridad y control de accesos:

- Autenticación de usuarios (ejemplo: tokens JWT).
- Autorización para operaciones sensibles (ejemplo: edición de datos solo para administradores).
- Proceso:
 - Si la solicitud es válida, realiza consultas a la Base de Datos.
 - Si hay errores (ejemplo: datos incorrectos), retorna un mensaje de error al Frontend.

4. Base de Datos

- Tecnologías: PostgreSQL + PostGIS (extensión para datos geoespaciales).
- Funciones:
 - Almacena datos estructurados (ejemplo: tablas de usuarios, registros de eventos).
 - Gestiona datos geoespaciales:
 - Coordenadas geográficas (puntos, líneas, polígonos).
 - Metadatos asociados a ubicaciones (ejemplo: nombres de ciudades, atributos ambientales).
- Proceso:
 - Recibe consultas SQL/PostGIS desde el Backend.
 - Ejecuta operaciones como:
 - Búsquedas espaciales (ejemplo: "todos los puntos dentro de un radio").
 - Actualización o inserción de datos.
 - Retorna resultados al Backend (ejemplo: JSON con datos georreferenciados).

5. Respuesta al Usuario

- El Backend procesa los datos obtenidos de la Base de Datos y los formatea según el protocolo acordado (ejemplo: JSON).
- Envía la respuesta al Frontend mediante una API RESTful.
- El Frontend actualiza la interfaz visual:
 - Refresca el mapa con nuevos datos.
 - Muestra mensajes de éxito/error al usuario.

5.6.2 Explicación del paso a paso del flujo resumido

1. Un usuario selecciona los filtros de búsqueda de su interés, ejemplo: Región Andina, Cundinamarca, Cajicá (Frontend).
2. El Frontend envía una solicitud GET al Backend con coordenadas geográficas.
3. El Backend valida la solicitud y consulta la Base de Datos para obtener datos relacionados con esa zona.
4. La Base de Datos ejecuta una consulta PostGIS (ejemplo: **SELECT * FROM datos WHERE ST_Contains(geometria, punto)**).
5. El Backend recibe los resultados, los procesa y devuelve un JSON al Frontend.
6. El Frontend actualiza el mapa con información detallada sobre la región seleccionada.

5.7 Funcionalidad del Aplicativo

El aplicativo web se organizará en varias funcionalidades que trabajan de forma integrada para cubrir todas las necesidades requeridas. Cada uno de estas cumplirá un propósito específico en el manejo de los datos: desde su carga inicial hasta la visualización, análisis y exportación. A continuación, se describen cada una de ellas:

5.7.1 Carga de Datos: Carga y Persistencia desde Archivos Excel y Shapefiles

Esta funcionalidad cumplirá una función clave dentro del aplicativo web, ya que permite cargar información al aplicativo tanto al inicio como de manera periódica. Los datos provienen de archivos estructurados como Excel (.xlsx, .csv) o Shapefiles (.shp, .shx, .dbf), que pueden contener información tabular o geoespacial.

La carga de información se llevará a cabo a partir de la información entregada por el MME y se usará la plantilla diseñada por la misma entidad.

Dicha carga de información se hará sobre la base de datos de PostgreSQL respetando la estructura enviada por el MME.

Los datos se almacenarán en la base de datos principal del aplicativo web para su posterior uso.

El almacenamiento se realiza así:

- Los datos tabulares van a **PostgreSQL**.
- Los datos espaciales derivados de los shapefiles se guardan en **PostGIS**, que permite realizar consultas geográficas de forma eficiente.

Para facilitar la carga, se podrá usar el panel de administración de Django o un módulo privado con validaciones automáticas y opción de carga por lotes.

5.7.1.1 Carga total o actualización de todos los datos

El sistema está diseñado para manejar diferentes tipos de actualización: cargas totales, actualización puntual de uno o varios determinantes o indicadores ya existentes, e incluso la incorporación de nuevos indicadores. Estas operaciones se pueden realizar mediante la interfaz de administración (Django Admin) o un módulo privado habilitado para la validación automática de los archivos, permitiendo una gestión eficiente, segura y flexible de los datos territoriales en cada fase de actualización o mantenimiento.

El sistema permite realizar cargas totales de los datos suministrados por el MME. Estas cargas se pueden hacer de forma masiva a través del módulo de administración o un panel privado, utilizando archivos Excel o Shapefiles, lo cual garantiza la actualización completa de la base de datos territorial. El documento anexo del backend, hace referencia a endpoints como `/indicadores-territorial-uno` y `/municipality-data_v2`, los cuales retornan todos los registros de indicadores y determinantes, lo que es funcional para procesos de carga completa de datos desde Excel o Shapefiles.

5.7.1.2 Inclusión de otro determinante o indicador

En el backend, se habilita la posibilidad de ampliar la base de datos mediante la inclusión de nuevos determinantes o indicadores no contemplados inicialmente. Esta función se ejecuta mediante validaciones en la carga y persistencia, asegurando que los nuevos campos sean interpretados correctamente y reflejados en el sistema de consulta geoespacial, como se evidencia en los endpoints `/indicadores-territorial` y `/municipality-data_v2`, del documento anexo del backend, que permiten incluir nuevos campos con nombres personalizados.

5.7.2 Tecnologías principales

- Python + pandas, geopandas, fiona, openpyxl
- PostgreSQL + PostGIS
- Django Admin o interfaz web privada para cargar los archivos

5.7.3 Visualización Geoespacial: Exploración Interactiva del Territorio

Esta funcionalidad ofrecerá una forma visual e intuitiva de consultar la información georreferenciada a través de un mapa interactivo. Permite que el usuario explore datos por regiones, departamentos y municipios, y observe directamente la distribución territorial de los mismos.

Desde lo técnico, el mapa se alimentará con datos extraídos desde una API REST que consulta directamente la base de datos geoespacial (PostGIS). Para el renderizado del mapa en el navegador, se integran bibliotecas como Leaflet.js.

Entre sus funciones principales se destacan:

- Selección dinámica de departamentos y municipios a través de menús o clics en el mapa.
- Carga asincrónica de capas de datos desde el backend, sin recargar la página.
- Posibilidad de superponer dos capas temáticas (basemap y hexágonos) para enriquecer la visualización.

5.7.4 Tecnologías utilizadas

- JavaScript + Leaflet.js
- API REST con Django REST Framework.
- PostgreSQL + PostGIS

5.8 Filtrado y Consulta: Búsquedas por Indicadores, determinantes y localización

Con esta funcionalidad, el usuario puede consultar datos específicos aplicando uno o varios filtros combinados. Esto le permite acceder exactamente a la información que necesita sin necesidad de explorar todo el aplicativo web.

Este componente ofrece:

- Filtros por localización, indicador o determinante.
- Consultas optimizadas a la base de datos usando parámetros, lo que mejora notablemente el rendimiento.
- Filtros que se actualizan en cadena según las elecciones anteriores del usuario.

Para las consultas más complejas, el aplicativo web utilizará vistas SQL o procedimientos almacenados. Desde el backend, los filtros se convierten en consultas SQL dinámicas mediante ORM como Django ORM.

5.8.1 Tecnologías utilizadas

- React con componentes interactivos (dropdowns, checkboxes, sliders)
- Django REST
- PostgreSQL

5.9 Comparación: Análisis Comparativo entre Unidades Territoriales

Esta funcionalidad permitirá comparar dos municipios o departamentos seleccionados por el usuario. Presenta la información en paralelo, lo que facilita identificar diferencias y similitudes entre ambas entidades.

Así es como funciona:

- El usuario elige los dos territorios desde la interfaz
- El aplicativo web lanza consultas separadas para cada uno y obtiene los datos relevantes
- Se muestran los resultados en tablas, gráficos o mapas comparativos

Además, se podrán destacar diferencias porcentuales o mostrar los valores absolutos para cada indicador.

5.9.1 Tecnologías empleadas

- React.js con Recharts para los gráficos
- API REST que permite consultas simultáneas
- PostgreSQL con funciones agregadas para facilitar los cálculos

5.10 Reportes: Generación de Informes en PDF y Excel

Esta funcionalidad permitirá crear reportes personalizados con base en los filtros aplicados por el usuario. El aplicativo web generará automáticamente documentos en formato PDF o Excel listos para ser descargados, impresos o compartidos.

El proceso es el siguiente:

- El usuario define los criterios del reporte: Determinantes, indicadores y localización.
- El backend arma el contenido a partir de plantillas ya definidas.
- El aplicativo web construye el documento con tablas, gráficos y mapas.

Para exportar:

- En PDF se usa la librería de los navegadores llamada jsPDF.
- En Excel, se emplea openpyxl.

Los archivos generados se almacenarán temporalmente (por ejemplo, en una carpeta local o en servicios como Amazon S3) y se habilitará un enlace para que el usuario los descargue.

5.10.1 Tecnologías utilizadas

- Python + WeasyPrint, xlswriter, jinja2.
- API para descarga de archivos.
- Interfaz en React/Vue con botón de exportación.

6. Estructura Base del Proyecto Web.

El proyecto se construirá sobre una arquitectura que separa claramente el backend y el frontend, lo que facilita el trabajo en equipo, el mantenimiento del aplicativo web y su despliegue en producción.

- **backend/**: Aquí se encuentra toda la lógica del servidor desarrollada con Django. Contiene archivos como manage.py, requirements.txt y .env, así como la carpeta carga_datos/ con los archivos clave: models.py, serializers.py, views.py y urls.py.
- **frontend/**: Está desarrollado con React y organizado dentro de src/. Incluye los componentes principales del aplicativo web (Home.jsx, Mapa.jsx, Comparador.jsx, Reportes.jsx), así como App.jsx, index.js y una carpeta docker/ con los Dockerfile que permiten la contenedorización.

Esta estructura modular permite que tanto el backend como el frontend se desarrollen y desplieguen de forma independiente, y además facilita la integración con herramientas de automatización como Docker para ambientes de producción.

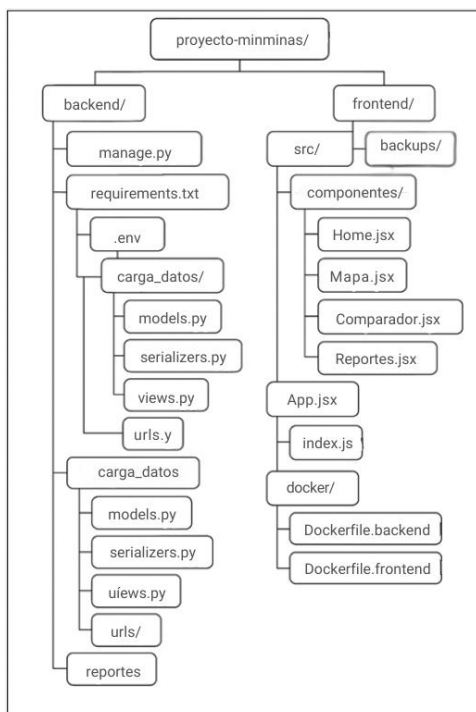


Ilustración 1. Estructura de carpetas. Fuente: Imagen elaborada por ADA.

6.1 Estructura de componentes

La estructura del aplicativo se organizará de forma modular, teniendo como base el componente principal `MainLayout.tsx`, que funciona como la columna vertebral de toda la interfaz. Esta organización no solo facilita el desarrollo, sino que también hace más sencilla la tarea de mantener y escalar la aplicación, ya que cada parte tiene una responsabilidad bien definida.

`MainLayout.tsx` actúa como un contenedor general que agrupa y dispone los diferentes módulos funcionales del aplicativo web. Gracias a este enfoque, es más fácil entender cómo está construido y cómo interactúan sus distintas partes.

La estructura se compone de tres contenedores principales que reúnen los componentes según la función que cumplen dentro del aplicativo web. Esta separación no es solo visual, sino también lógica, y ayuda a que el flujo de la información sea claro y organizado. A continuación, se describe el rol de cada contenedor:

- **Contenedor de filtros (`FilterContainer.tsx`):** Aquí se agrupan todos los elementos que permiten filtrar la información. El usuario puede seleccionar un municipio, uno o varios determinantes, y también comparar datos. Además, se incluyen botones que permiten aplicar los filtros, restablecer la búsqueda o exportar los resultados. Este contenedor es clave para que el usuario pueda definir exactamente qué información quiere consultar, de forma rápida y sencilla.
- **Contenedor de resultados (`ResultsContainer.tsx`):** Una vez aplicados los filtros, los datos se muestran en este módulo. Incluye visores gráficos para ver la información de forma individual o comparativa, y también una tabla con los resultados en formato detallado. Esta parte del aplicativo web está pensada para que el usuario pueda interpretar fácilmente los datos, con opciones para ordenar, filtrar o exportar lo que necesite.
- **Contenedor de reportes (`ReportsContainer.tsx`):** Este módulo está enfocado en los reportes generados a partir de las consultas. Muestra una tabla donde se listan los reportes disponibles, con sus respectivos datos y opciones para descargar o explorar más a fondo. Es una herramienta útil para quienes necesitan dar seguimiento a los resultados o consolidar información en documentos.

Esta forma de organizar el aplicativo web, agrupando componentes por niveles y funciones, permitirá tener una visión clara del recorrido que hacen los datos desde que el usuario los consulta hasta que se presentan en pantalla o se convierten en reportes. Además, hace que cada parte del aplicativo web sea más fácil de desarrollar, probar y mantener de forma independiente, lo cual es fundamental en proyectos que evolucionan constantemente o que necesitan integrar nuevas funcionalidades con el tiempo.

Ver los siguientes anexos para conocer el detalle de las funcionalidades mencionadas anteriormente:

- Anexo 1. Documentación técnica del código Front End para el Ministerio de Minas y Energía.
- Anexo 2. Documentación técnica Back End para Ministerio de Minas y Energía.

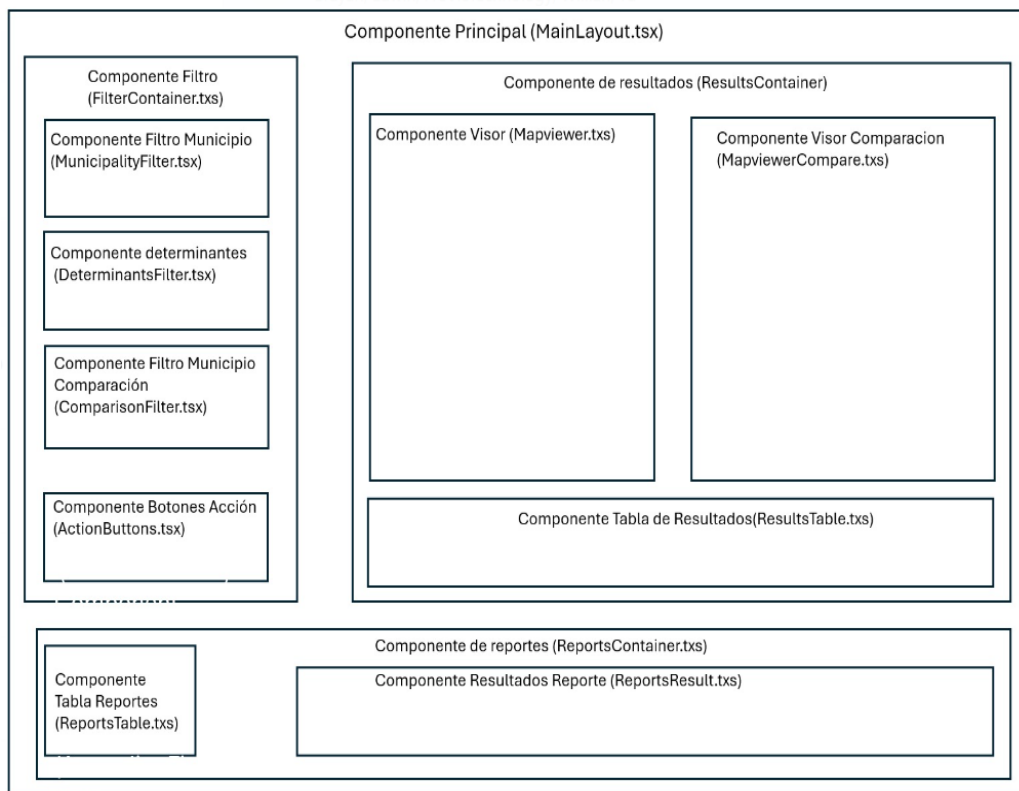


Ilustración 2. Estructura de componentes. Fuente: Imagen elaborada por ADA.

6.2 Interfaces Principales

El aplicativo web estará compuesto por una serie de interfaces clave que permitirán a los usuarios interactuar con el mismo de forma clara, ordenada y eficiente. Cada una de estas interfaces se diseña pensando en la experiencia del usuario, facilitando el acceso a la información y a las funcionalidades más relevantes.

6.2.1 Página de Inicio (Landing Page)

Esta es la puerta de entrada al aplicativo web. Desde aquí, el usuario puede tener una vista general del aplicativo y acceder rápidamente a las distintas secciones disponibles. La página está diseñada para ser visualmente amigable, destacando los módulos principales e incluyendo menús accesibles, botones de acción intuitivos y enlaces útiles para facilitar la navegación.

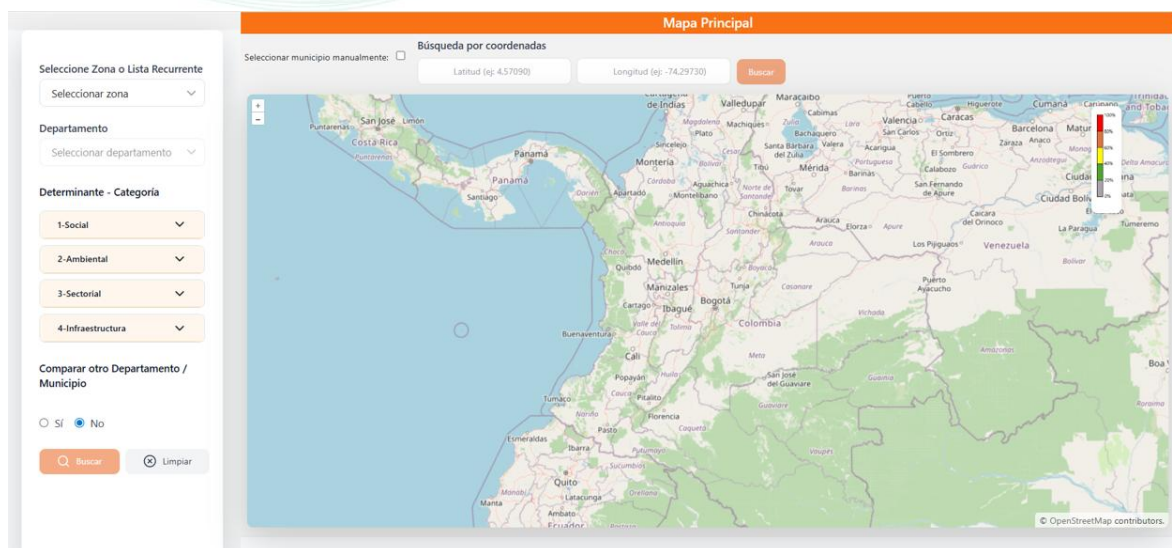


Ilustración 3. Interfaz página de inicio. Fuente: Imagen elaborada por ADA.

6.2.2 Zonas

Esta sección permite visualizar el territorio colombiano dividido por regiones geográficas: Amazonia, Andina, Caribe, Insular, Orinoquia y Pacífica. Cada zona se presenta de manera interactiva, permitiendo al usuario explorar la información territorial disponible para cada una. Esta interfaz no solo organiza los datos por ubicación, sino que también hace uso de herramientas geoespaciales para mostrar la información de forma clara y dinámica.

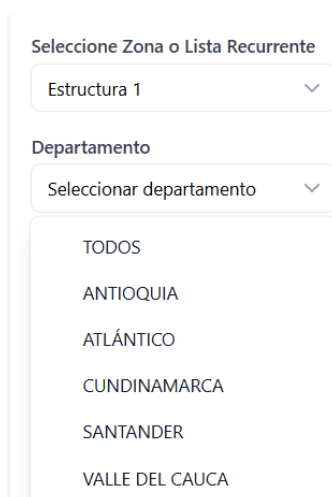


Ilustración 4. Interfaz Zonas. Fuente: Imagen elaborada por ADA.

6.2.3 Lista Recurrente

El aplicativo web incluye tres listas desplegables que permiten al usuario seleccionar criterios específicos para filtrar o enfocar la información que desea consultar. Estas listas están diseñadas para ser prácticas y están interrelacionadas, de modo que las selecciones realizadas en una pueden influir en las opciones disponibles en las otras. Así, se facilita una navegación más lógica y dirigida dentro del aplicativo web.



Ilustración 5. Listas recurrentes. Fuente: Imagen elaborada por ADA.

6.2.4 Pantalla de Determinantes

Aquí se puede consultar un conjunto de determinantes organizados en secciones desplegables tipo acordeón. Esta disposición permite al usuario revisar la información por categorías, sin saturar la pantalla. Al hacer clic en cada acordeón, se despliegan los datos asociados, lo que facilita su consulta sin perder de vista el contexto general.



Ilustración 6. Interfaz determinantes. Fuente: Imagen elaborada por ADA.

6.2.5 Pantalla 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.

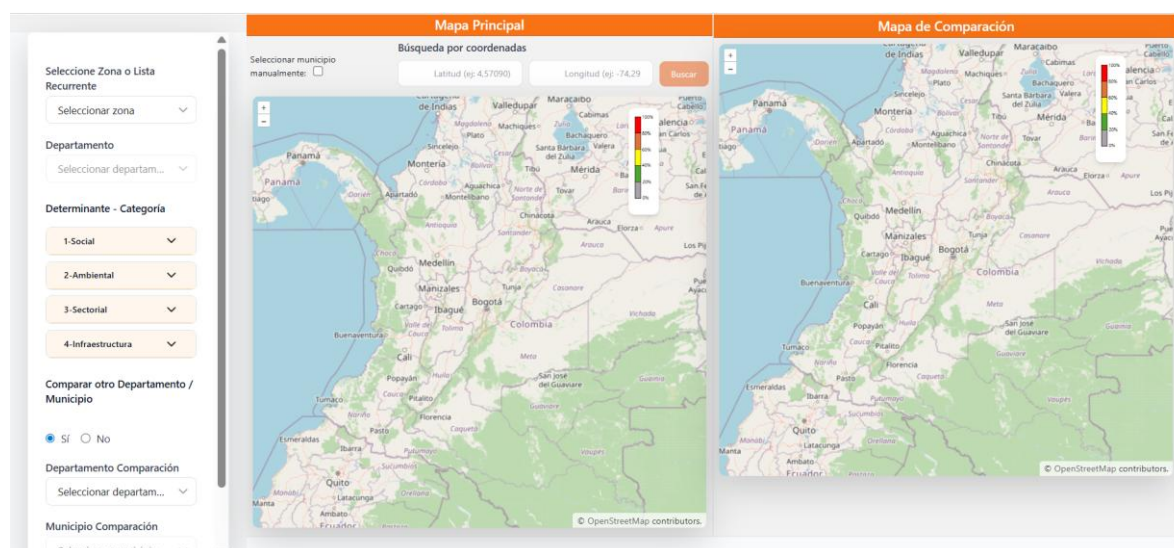


Ilustración 7. Pantalla de comparación. Fuente: Imagen elaborada por ADA.

6.2.6 Pantalla de Reportes

Esta sección permite generar reportes descargables en formatos PDF o Excel, con base en los filtros aplicados por el usuario. Los reportes se pueden personalizar según las necesidades de quien los consulta y están pensados para su uso en análisis, documentación o informes institucionales. El proceso de generación es sencillo: basta con seleccionar el tipo de reporte y el formato deseado para que el aplicativo web lo genere de forma automática y rápida.

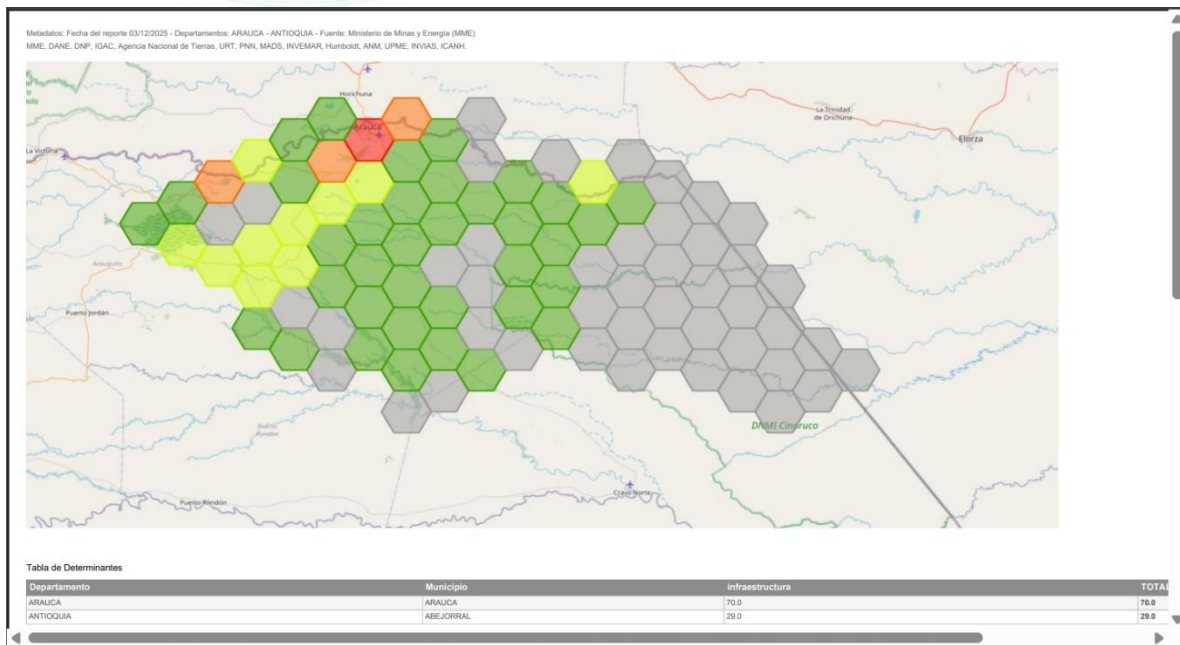


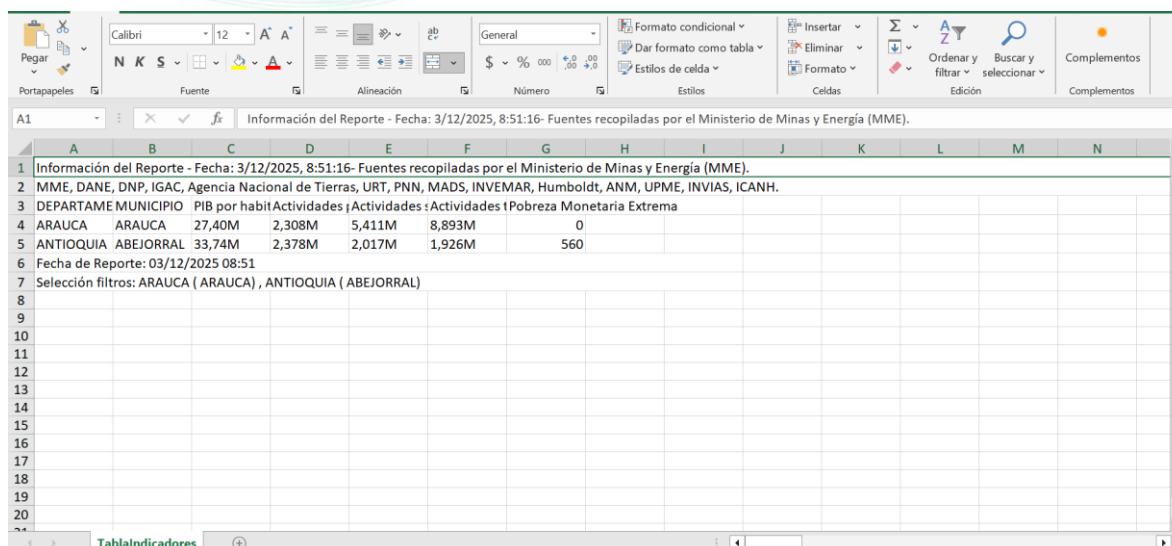
Ilustración 8. Pantalla de reporte PDF. Fuente: Elaborado por ADA.

Información del Reporte - Fecha: 3/12/2025, 8:51:45 - Categorías: 2-Ambiental, 3-Sectorial, 4-Infraestructura - Fuentes recopiladas por el Ministerio de Minas y Energía (MME).

MME, DANE, DNP, IGAC, Agencia Nacional de Tierras, URT, PNN, MADS, INVEMAR, Humboldt, ANM, UPME, INVIAS, ICANH.

Departamen	Municipio	2-Ambiental	3-Sectorial	4-Infraestructura	TOTAL	Categoría	Determinant	Suma
ARAUCA	ARAUCA	135.0	35.0	70.0	240.0	2-Ambiental		
							RUNAP	0.0
							Páramos	31.0
							Bosque Seco	58.0
							Manglar	0.0
							Ley 2/59	0.0
							AICAS	0.0
							ZPRNRMA re	0.0
							Bosques de l	0.0
							POMCAS	0.0
							Bancos de H	0.0
							SIPTA EN EV	0.0
							Licencias API	46.0
						3-Sectorial		
							Título Vigent	6.0
							Solicitudes V	1.0
							Zona Minera	0.0
							Mapa de Tie	23.0
							Gasodutos y	2.0
							Subestaciones	1.0

Ilustración 9. Exportar determinantes Excel. Fuente: Elaborado por ADA.



DEPARTAMENTO	MUNICIPIO	PIB por habitante	Población Rural	Población Total
ANTIOQUIA	ENIGADO	33,74M	7.961	246.003
CUNDINAMARCA	CHÍA	28,85M	25.338	155.541

Ilustración 10. Exportar indicadores. Fuente: Elaborado por ADA.

6.2.7 Pantalla indicadores

Aquí se puede consultar un conjunto de indicadores organizados en una lista, los cuales pueden ser seleccionados a través del botón check, ubicado al frente de cada uno. Luego se da clic en el botón cargar, entonces aparecerá la tabla con los datos de los mismos, que posteriormente pueden ser graficados hasta un máximo de tres al mismo tiempo.

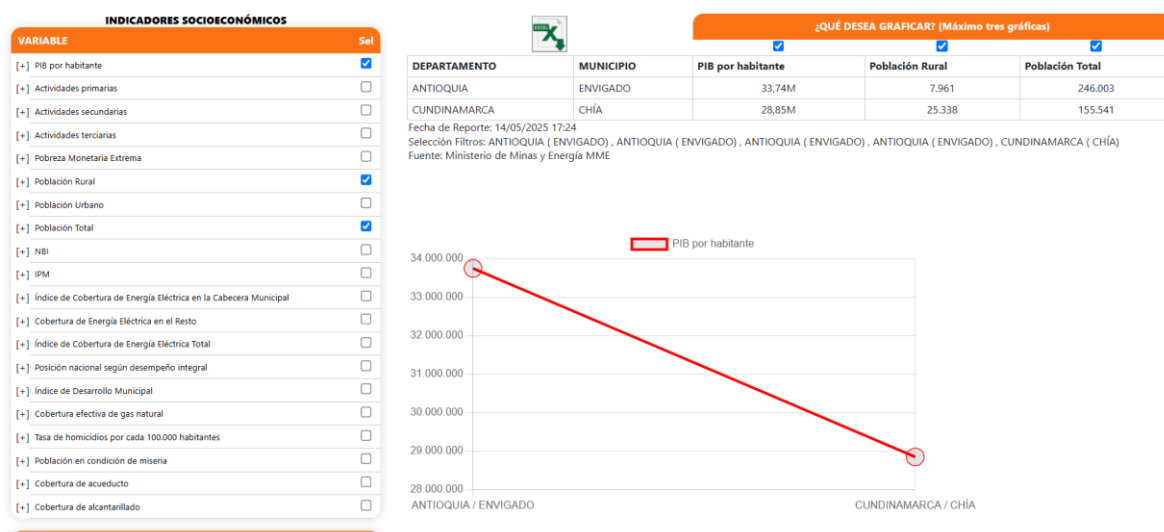


Ilustración 11. Indicadores. Fuente: Elaborado por ADA.

7. Desarrollo del Aplicativo Web

En este apartado se describe de forma detallada el proceso integral que da lugar al desarrollo del aplicativo web. Esta construcción tecnológica se orienta a cumplir con los requerimientos funcionales y no funcionales previamente definidos, procurando que el aplicativo web resultante sea eficiente, escalable, seguro, accesible y fácil de usar para todos los usuarios, sin importar su nivel técnico. El desarrollo contempla tanto el uso de tecnologías modernas como la adopción de metodologías ágiles, lo cual permite mantener un enfoque iterativo, centrado en la mejora continua y en la colaboración estrecha con los equipos del Ministerio de Minas y Energía y la Oficina de Asuntos Ambientales y Sociales (OAAS).

7.1 Metodología ágil usada en el desarrollo del proyecto.

La metodología usada por la fábrica durante el desarrollo del proyecto se llama “ADA REI” que significa Rapid and Easy Implementation, la cual consiste en cuatro fases:

- **Preparar:** Formalización del contrato y el acta de inicio.
- **Explorar:** Asignación de recursos, realización de la Historia de Usuario, casos de uso y diseño del prototipo.
- **Realizar:** Etapa en que se desarrolla el sistema y realización de QA.
- **Desplegar:** Etapa del despliegue sobre el ambiente de pruebas del MME.

Nota: Los tiempos de ejecución de las fases fueron determinados en el cronograma, para conocer el detalle ver entregable 1.2.3 EDT Desarrollo y Afinamiento de Cronograma.

7.2 Tecnologías y Herramientas

La selección de tecnologías y herramientas responde a criterios de eficiencia y mantenibilidad. Cada componente se elige de forma estratégica para cubrir necesidades específicas, optimizando los recursos disponibles y asegurando un alto estándar de calidad.

- **React & TailwindCSS:** Se emplean para construir una interfaz de usuario moderna, responsiva y accesible, que garantiza una experiencia fluida en dispositivos móviles y de escritorio. React permite crear componentes reutilizables y dinámicos, mientras que TailwindCSS acelera el diseño visual con un enfoque utilitario.
- **Django:** Se utilizan como frameworks backend en Python, brindando una base robusta para el desarrollo de APIs RESTful que facilitan la comunicación entre el cliente y el servidor. Destaca por su estructura integral y seguridad incorporada; Flask, por su ligereza y flexibilidad.
- **PostgreSQL y PostGIS:** Constituyen la solución para la gestión de datos tanto tabulares como espaciales. PostgreSQL se reconoce por su confiabilidad y extensibilidad, mientras que PostGIS extiende sus capacidades para manejar información georeferenciada.

- **Pytest / Unittest:** Se implementan para ejecutar pruebas automatizadas, tanto unitarias como de integración, asegurando que cada módulo del aplicativo web funcione correctamente antes de avanzar a la siguiente etapa de desarrollo.
- **Docker:** Se adopta para contenedorización de servicios, lo que facilita la replicación de ambientes, el despliegue ágil y la escalabilidad horizontal del aplicativo en diferentes entornos (desarrollo, pruebas, producción).

7.2.1 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.

8. Proceso de Desarrollo

El desarrollo del aplicativo se estructura a partir de una metodología ordenada que abarca desde la comprensión inicial de las necesidades hasta la entrega final del producto funcional. Cada fase representa un paso crítico hacia la construcción de una solución tecnológica robusta, sostenible y alineada con los objetivos institucionales.

8.1 Etapa de Recolección de Requisitos

En esta primera fase, el equipo de desarrollo se involucró activamente con los actores clave del proyecto para comprender a fondo las expectativas, necesidades y restricciones del aplicativo web.

8.1.1 Actividades

- Se llevaron a cabo reuniones con funcionarios del Ministerio de Minas y Energía y personal de la OAAS, con el fin de entender la visión del proyecto y las

particularidades del contexto institucional. Se realizó un trabajo conjunto entre las partes.

- Se analizó detalladamente como desarrollar el aplicativo web que ayude a implementar la Estrategia de Relacionamento Territorial del Ministerio de Minas y Energía (MME), determinando la mejor forma de aprovechar sus capacidades.
- Se documentaron los requisitos funcionales (RF) y no funcionales (RNF), incluyendo criterios de seguridad, accesibilidad, rendimiento, escalabilidad y compatibilidad tecnológica.
- Se construyeron historias de usuario claras y casos de uso que facilitan la validación de requisitos durante el desarrollo.

8.1.2 Entregables

- Documento de Especificación de Requisitos del Software (ERS), que detalla cada requerimiento identificado y validado.
- Matriz de Trazabilidad de Requisitos, que permite hacer seguimiento al cumplimiento de cada ítem durante el ciclo de vida del proyecto.

8.2 Diseño del Aplicativo web

Con los requisitos claros, se procederá a definir la estructura técnica y visual del aplicativo web, tomando decisiones clave que influirán en su arquitectura, desempeño y experiencia de usuario.

8.2.1 Actividades

- Se diseña la arquitectura del aplicativo web, evaluando enfoques como cliente-servidor tradicional o microservicios, según la necesidad del caso.
- Se modelan las bases de datos mediante diagramas entidad-relación (ER) que permiten visualizar cómo se almacenan, relacionan y consultan los datos.
- Se crean wireframes y prototipos de interfaz (UI/UX), procurando que la experiencia de uso sea intuitiva, inclusiva y consistente con la identidad institucional.

8.2.2 Entregables

- Documento de Arquitectura del Aplicativo web, que incluye la descripción técnica del diseño adoptado.
- Prototipos navegables que permiten explorar las interfaces antes de su codificación.

- La funcionalidad general del aplicativo se encuentra en el diagrama de flujo que documenta el proceso y la lógica interna del aplicativo web. Ver “Anexo 5. Diagrama de flujo Carga de Datos” para conocer el detalle.

8.3 Puesta en Práctica

En esta fase se materializa el diseño del aplicativo web, construyendo tanto la lógica del negocio como la presentación visual.

8.3.1 Actividades

- Se desarrolla el backend, implementando las APIs RESTful, lógica empresarial y mecanismos de autenticación y autorización.
- Se construye la interfaz de usuario, desarrollando componentes accesibles y adaptativos según los prototipos definidos.
- Se habilita un proceso de integración continua, incorporando automatización de pruebas, validación de código y despliegues controlados (CI/CD).
- Se gestionan versiones del código mediante aplicativo web de control como Git, garantizando trazabilidad.

8.3.2 Entregables

- Código fuente documentado, alojado en un repositorio compartido y controlado.
- Scripts para la creación y migración de bases de datos.
- Servicios funcionales integrados, listos para ser validados en ambientes de prueba.

9. Pruebas

Antes de abrir las puertas del aplicativo al público, se llevará a cabo una fase de validación técnica. La plataforma pasa por un conjunto de pruebas estructuradas que tienen como objetivo descubrir, anticipadamente, cualquier debilidad o error que pudiera comprometer su rendimiento o funcionalidad.

9.1 Actividades

- **Pruebas de carga y estrés:** Diseñadas para observar el comportamiento del aplicativo web bajo un tráfico elevado, como el que podría presentarse en jornadas de alta demanda. Para este proceso se usó la herramienta JMeter y se añadió el detalle de las mismas en documento anexo 4 “Prueba de estrés aplicativo web”.

- **Pruebas funcionales:** Confirman que cada componente del aplicativo web se desempeña correctamente y que todas las partes se comunican entre sí sin errores.

Esta metodología tiene como finalidad reducir al mínimo los riesgos al momento del lanzamiento, asegurando que los usuarios finales accedan a una plataforma sólida, fiable y bien preparada.

La fase de pruebas tiene como finalidad asegurar que el aplicativo web cumple con lo esperado, bajo distintos escenarios y condiciones de uso.

9.2 Entregables

- Plan de pruebas y resultados detallados.
- Informe de validación de accesibilidad con observaciones y recomendaciones.
- Registro de errores detectados, con su respectiva documentación y corrección.

10. Despliegue

Finalmente, el aplicativo se preparará para su implementación en el entorno institucional, validando su funcionamiento en condiciones reales.

10.1 Actividades

- Se configurará el ambiente de pruebas facilitado por el Ministerio de Minas y Energía, replicando condiciones similares al entorno de producción.
- Se cargará el aplicativo web en dicho entorno, realizando verificaciones técnicas y funcionales.
- Se brindará acompañamiento técnico a los usuarios clave durante su fase de validación, resolviendo dudas y atendiendo ajustes de último momento.

10.2 Entregables

- Aplicativo web desplegado en el ambiente de pruebas del MME.
- Documentación técnica del proyecto.
- Manual de usuario y manual técnico.
- Documento que evidencie el despliegue en ambiente de pruebas del MME.

11. Paso a paso del Proceso de Desarrollo

A continuación, se incluye un paso a paso que representa las fases del proceso de desarrollo del aplicativo. Este flujo secuencial permite gestionar de manera efectiva el proyecto, asegurando que cada etapa se complete de forma rigurosa antes de avanzar a la siguiente.

- **Análisis de Requerimientos:** Se enfoca en comprender a fondo la necesidad del cliente, representado por el MME.
- **Diseño del Aplicativo web:** Se planifica la arquitectura, se definen las tecnologías a utilizar y se diseñan los componentes.
- **Implementación y Codificación:** Se desarrollan los elementos funcionales del que dan respuesta a las necesidades identificadas.
- **Pruebas y Validación:** Se evalúa el aplicativo web desde una perspectiva técnica y funcional, validando que responde a los requerimientos definidos.

A continuación, se detalla el paso a paso del proceso de desarrollo:

11.1 Inicio del Proyecto

- **Objetivo:** Establecer las bases del proyecto, definir objetivos iniciales y asignar recursos.

11.2 Análisis de Requerimientos

- **Actividades:**
 - Recolección de información con *stakeholders* (clientes, usuarios, equipo técnico).
 - Identificación y documentación de requisitos funcionales (qué debe hacer el sistema) y no funcionales (rendimiento, seguridad, etc.).
 - Establecimiento del alcance del sistema (límites y funcionalidades prioritarias).

11.3 Diseño del Sistema

- **Actividades:**
 - Definición de la arquitectura del sistema (ej: cliente-servidor, microservicios, monolítico).

- Selección de tecnologías: frameworks, lenguajes de programación, herramientas de desarrollo.
- Modelado de base de datos: diseño del modelo entidad-relación, normalización de tablas.
- Diseño de interfaces de usuario: creación de *mockups*, *wireframes* o prototipos interactivos.
- Validación y aprobación del diseño por parte del cliente.

11.4 Implementación y Codificación

- **Actividades:**

- Desarrollo de componentes:
 - **Backend:** Lógica del servidor, APIs, conexión con bases de datos.
 - **Frontend:** Interfaz de usuario, experiencia del usuario (UX).
- Integración de módulos para garantizar funcionamiento conjunto.
- Gestión de versiones con herramientas como Git, SVN, etc.
- Aplicación de buenas prácticas: código limpio, documentación interna, estándares de seguridad.

11.5 Pruebas y Validación

- **Actividades:**

- Ejecución de pruebas:
 - **Unitarias:** Validación de componentes individuales.
 - **Integración:** Verificación de la interacción entre funcionalidades.
 - **Funcionales:** Comprobación de que el sistema cumple con los requisitos.
- Validación de requisitos vs. funcionalidades entregadas.

11.6 Entrega / Cierre del Proyecto

- **Actividades:**

- Implementación del sistema en el entorno de pruebas del MME.
- Capacitación a usuarios finales.

- Entrega de documentación técnica y manuales de usuario.
- Revisión final con el MME y firma de aceptación.

12. Compatibilidad con navegadores y dispositivos móviles modernos

El diseño del aplicativo no solo es funcional, también es responsivo. Se adapta con fluidez a cualquier pantalla y contexto de uso. Gracias a la implementación de principios de diseño, la aplicación puede utilizarse sin inconvenientes desde un celular, una tableta o un computador de escritorio.

Se asegura también la compatibilidad con los navegadores más populares (Chrome, Firefox, Edge, Safari). Esto se logra aplicando los estándares más recientes en desarrollo web —HTML5, CSS3, JavaScript moderno— y realizando pruebas que garantizan una experiencia de usuario coherente y satisfactoria.

13. Acompañamiento Técnico

El éxito de un proyecto digital como este no recae únicamente en su código, es indispensable el trabajo conjunto, el diálogo constante, y la validación de cada paso con el Ministerio de Minas y Energía. Por eso, se establece un acompañamiento técnico cercano, donde se asegure no solo la calidad, sino también la alineación con las expectativas institucionales.

13.1 Seguimiento del Progreso: QA (Quality Assurance)

La entrega pasa por un proceso de verificación, donde se realizan:

- Pruebas funcionales y no funcionales.
- Revisión del cumplimiento de requerimientos.
- Registro y documentación de hallazgos o incidencias.
- Ajustes proactivos con base en observaciones.
- Validación por parte del equipo de QA, como sello final antes de presentar avances al Ministerio de Minas y Energía.

13.2 Presentación al líder funcional del Ministerio de Minas y Energía

Cada entrega se convierte en una oportunidad de diálogo. Se presenta el avance en un entorno de pruebas, se acompaña con documentación y se hace una demostración funcional. El líder funcional del Ministerio de Minas y Energía participa activamente en esta revisión, brindando observaciones y retroalimentación que se registran cuidadosamente para seguir mejorando.

13.3 Pruebas del Aplicativo web

13.3.1 Pruebas de aceptación

Junto al Ministerio de Minas y Energía, se validará que el aplicativo web esté listo. Se comprueba que las funcionalidades esperadas estén presentes, que el rendimiento sea el adecuado y que el aplicativo cumpla con los requisitos de accesibilidad y robustez. Todo se documenta y de ser necesario, se ajusta.

14. Mínimo que se realiza

Como parte del proceso mínimo que se llevará a cabo en esta etapa, se iniciará una revisión detallada de las funcionalidades más relevantes del aplicativo web, en estrecha colaboración con los representantes designados por el Ministerio de Minas y Energía (MME). Esta revisión no se limita a una simple verificación técnica, implica diálogos constantes, intercambios de percepciones y validación de que cada funcionalidad, cada flujo, responde adecuadamente a las necesidades reales del Ministerio de Minas y Energía. Se buscan no solo errores, sino también oportunidades de mejora, enfoques más eficientes y formas más intuitivas de presentar la información o interactuar con el usuario.

En paralelo, se ejecutarán los casos de prueba que han sido previamente definidos durante la fase de planificación. Estos casos se diseñan con base en escenarios concretos de uso, alineados con los requerimientos funcionales establecidos desde el inicio del proyecto. Su ejecución permite identificar comportamientos esperados y detectar cualquier desviación respecto a lo planteado, ya sea en el rendimiento, la lógica del aplicativo web o la experiencia del usuario. Cada resultado se analiza con atención, buscando no solo confirmar que las funcionalidades operan correctamente, sino también anticipar posibles fallos en condiciones menos ideales.

Finalmente, se documentará cuidadosamente todo lo que se observa. Esto incluye las verificaciones realizadas, la retroalimentación recibida por parte del equipo del MME y cualquier observación técnica o funcional que surja durante las pruebas. Además, se registrarán los ajustes que se deban implementar, ya sea por corrección, por mejora o por adaptación a una nueva necesidad detectada en el proceso. Este ejercicio no es solo una formalidad, es una oportunidad para dejar una trazabilidad clara y útil, que permita al equipo técnico mantener una evolución controlada del aplicativo web y al Ministerio de Minas y Energía, tener la certeza de que cada paso se ejecuta con rigor y compromiso.

15. Implementación en el ambiente de desarrollo del MME

15.1 Prerrequisitos

- Asegurar acceso administrativo al servidor (permisos *sudo*).

- Verificar que los puertos necesarios (80, 3306, 8080, 3030) estén habilitados en el firewall.
- Contar con una cuenta de Docker registrada.

15.2 Instalación de Docker en Linux/Ubuntu

- Eliminar versiones antiguas de Docker y dependencias conflictivas.
- Actualizar repositorios del sistema e instalar herramientas como *curl*.
- Agregar la clave GPG oficial de Docker y su repositorio.
- Instalar Docker CE, CLI, *containerd.io* y plugins de Docker Compose.
- Configurar permisos del usuario para ejecutar Docker sin *sudo*.

15.3 Despliegue de la Aplicación

- Navegar al directorio del proyecto y construir la imagen Docker.
- Subir la imagen al repositorio Docker del MME.
- Crear y configurar el archivo *docker-compose.yml* con servicios, redes y volúmenes.
- Ejecutar el despliegue en segundo plano.

15.4 Verificación y Solución de Problemas

- Validar que los contenedores estén activos.
- Revisar logs en caso de errores.
- Solucionar conflictos comunes:
 - **Permisos:** Asegurar que el usuario pertenezca al grupo *docker*.
 - **Puertos bloqueados:** Verificar apertura de puertos en el firewall.
 - **Versión de Docker Compose:** Confirmar compatibilidad con el archivo YAML.

Nota: Ver anexo 3. Evidencia Despliegue en ambiente MME.

16. Aprobación

A continuación, firman aprobando:

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