Plan de Implementación del Proyecto

1. Objetivo

Este documento describe el plan de implementación y la arquitectura tecnológica seleccionada para el proyecto de "Análisis Geoespacial de Potencial Solar en Territorios PDET". El objetivo es cumplir con los requerimientos de la UPME, utilizando una metodología reproducible y un stack de tecnología basado en soluciones NoSQL.

2. Selección de Tecnologías (Technology Stack)

Para cumplir con los requisitos de escalabilidad, flexibilidad y capacidad de análisis geoespacial, se ha seleccionado el siguiente stack tecnológico:

- **Base de Datos NoSQL: MongoDB**
    - **Justificación:** Se elige MongoDB por ser una base de datos orientada a documentos (JSON), lo cual se alinea perfectamente con el formato de los datos geoespaciales (GeoJSON). Su principal fortaleza para este proyecto es su potente motor de consultas geoespaciales (índices 2dsphere), que permitirá realizar las operaciones espaciales de manera nativa y eficiente. Esto es crucial para manejar los miles de millones de huellas de edificios.
- **Lenguaje de Programación: Python**
    - **Justificación:** Python es el estándar de la industria para la ciencia de datos y el análisis geoespacial.
    - **Bibliotecas Clave:**
        - **Pymongo**: El driver oficial para interactuar (leer/escribir) con la base de datos MongoDB.
        - **Geopandas y Pandas**: Para leer, limpiar y transformar los datos de origen (los Shapefiles del DANE y los CSV/GeoJSON de Google/Microsoft) antes de cargarlos a la base de datos. Se usará Geopandas para calcular el área de las geometrías de Microsoft.
        - **Jupyter Notebooks**: Para el desarrollo, la exploración de datos (EDA) y la documentación del flujo de análisis.
- **Control de Versiones: Git y GitHub**
    - **Justificación:** Es un requisito del proyecto para la entrega de los avances semanales.

3. Fases de Implementación (Alineadas con las Entregas)

El proyecto se ejecutará en cinco fases principales, correspondiendo a las entregas definidas:

Fase 1: Diseño de la Base de Datos (Entrega 1 - 27 Oct)

- **Acciones:**
    1. Finalizar este plan de implementación.
    2. Completar el modelado de datos, identificando las entidades: Municipios y Edificios.
    3. Diseñar los esquemas JSON para las colecciones de MongoDB, asegurando que puedan estandarizar los datos de Google y Microsoft.

Fase 2: Integración de Límites PDET (Entrega 2 - 3 Nov)

- **Acciones:**
    1. Adquirir los datos del Marco Geoestadístico Nacional (MGN) del DANE.
    2. Identificar y filtrar los municipios que son territorios PDET.
    3. Escribir un script en Python (usando Geopandas) para leer los Shapefiles de los municipios PDET y cargarlos en la colección municipios de MongoDB usando el esquema GeoJSON diseñado.

Fase 3: Carga de Huellas de Edificios (Entrega 3 - 10 Nov)

- **Acciones:**
    1. Descargar muestras de los datos de Microsoft y Google.
    2. Desarrollar los scripts de ETL (Extracción, Transformación y Carga) para procesar ambos datasets.
    3. **Transformación Clave:** El script estandarizará ambos datasets al esquema JSON edificios diseñado. Esto incluye calcular el área para los polígonos de Microsoft, ya que esa fuente no la provee.
    4. Cargar los datos en la colección edificios y crear el índice geoespacial (2dsphere) para optimizar las consultas.

Fase 4: Flujo de Análisis Geoespacial (Entrega 4 - 17 Nov)

- **Acciones:**
    1. Desarrollar el flujo de análisis reproducible.
    2. Ejecutar las consultas geoespaciales en MongoDB. Por cada municipio PDET en la colección municipios, se usará su geometría para contar ($count) y sumar ($group, $sum) el área de los edificios en la colección edificios que estén contenidos espacialmente ($geoWithin) en él.
    3. Este análisis se ejecutará dos veces: una para la fuente google y otra para microsoft.
    4. Generar las tablas y mapas de salida con los resultados.

Fase 5: Reporte Técnico Final (Entrega 5 - 24 Nov)

- **Acciones:**
    1. Consolidar toda la metodología, código y resultados en el reporte técnico.
    2. Presentar los hallazgos comparativos y las visualizaciones de datos.
    3. Redactar las recomendaciones finales para la UPME sobre la idoneidad de los datasets y los resultados encontrados.

Modelado de datos

Para este proyecto, hemos identificado dos entidades principales y la relación espacial entre ellas.

Entidades Principales

1. Entidad Municipio Esta entidad representa las unidades administrativas que son el foco del análisis.

- **Descripción:** Representa un territorio municipal colombiano, específicamente aquellos designados como PDET.
- **Fuente de Datos:** Proviene del Marco Geoestadístico Nacional (MGN) del DANE.
- **Atributos Clave:**
    - **Identificador Único:** Un código (ej. código DANE) que identifica al municipio.
    - **Nombre:** El nombre oficial del municipio.
    - **Indicador PDET:** Un atributo (booleano) para marcar si es un territorio PDET, facilitando el filtrado.
    - **Geometría:** El polígono (o multipolígono) que define los límites administrativos exactos del municipio. Este es el atributo crucial para el análisis espacial.

2. Entidad: Edificio Esta entidad es el objeto principal del análisis. El objetivo es contar cuántos de estos edificios existen y sumar su área de techo.

- **Descripción:** Representa la huella o contorno de un solo edificio en la superficie.
- **Fuentes de Datos:** Proviene de dos conjuntos de datos separados que deben ser comparados: Microsoft Building Footprints y Google Open Buildings.
- **Atributos Clave:**
    - **Geometría:** El polígono que define la huella del edificio. Este es el atributo usado para la operación espacial.
    - **Área de Techo (m²):** Un valor numérico que representa el área estimada del techo. Este es el atributo clave para la agregación.
    - **Fuente:** Un atributo de texto (ej. "Google" o "Microsoft") para identificar de qué conjunto de datos provino el registro, permitiendo la comparación.

Relación entre Entidades

La relación fundamental del proyecto es una **relación espacial**:

- Un **Municipio** (específicamente su geometría) **"contiene"** múltiples geometrías de **Edificio**.
- La consulta principal del proyecto será determinar, para cada Municipio PDET, qué Edificios están espacialmente **"dentro de"** (o *within*) su polígono de límites.

Diseño de Esquema y Pertinencia

3.1. Diseño del Esquema (Schema Design)

Se crearán dos colecciones principales: municipios y edificios.

Colección: municipios

Esta colección almacenará los polígonos de los municipios PDET, que servirán como áreas de consulta.

- **Propósito:** Almacenar las geometrías de los municipios PDET del DANE.

Estructura del Documento:

```
JSON
{
  "codigo_dane_municipio": "54810",
  "nombre": "Tibú",
  "geometria": {
    "type": "MultiPolygon",
    "coordinates": [
      [ /* Arreglo de polígonos de coordenadas [lng, lat] */ ]
    ]
  }
}
```

Colección: edificios

Esta colección contendrá los miles de millones de huellas de edificios de ambas fuentes, estandarizadas en un único formato.

- **Propósito:** Almacenar las huellas de edificios de Google y Microsoft para su posterior análisis.

Estructura del Documento (Ejemplo):

```
JSON
{
  "fuente": "google", // o "microsoft"
  "area_m2": 120.5,
  "geometria": {
    "type": "Polygon",
    "coordinates": [
      [ /* Arreglo de coordenadas [lng, lat] */ ]
    ]
  }
}
```

3.2. Estrategia de Indexación

Para garantizar "operaciones espaciales eficientes", la indexación es fundamental.

1. **Índice Geoespacial :**
    - **Colección:** edificios
    - **Campo:** geometria
    - **Tipo:** 2dsphere (Índice geoespacial de MongoDB).
    - **Justificación:** Este es el índice más importante. Permitirá a la base de datos ejecutar consultas $geoWithin (encontrar edificios *dentro* de un polígono municipal) de forma extremadamente rápida, incluso con miles de millones de documentos.
2. **Índice de Comparación:**
    - **Colección:** edificios
    - **Campo:** fuente

    - **Tipo:** Estándar (ascendente).
    - **Justificación:** Acelerará las consultas de filtrado para separar los análisis de Google y Microsoft.
3. **Índice de Búsqueda:**
    - **Colección:** municipios
    - **Campo:** codigo_dane_municipio
    - **Tipo:** Único (unique).
    - **Justificación:** Asegura la integridad de los datos y permite una búsqueda rápida de los municipios por su código.

3.3. Pertinencia del Diseño (Appropriateness)

Este diseño de esquema es pertinente y cumple con todos los objetivos del proyecto:

- **Cumple con el requisito NoSQL:** Utiliza un modelo de documentos (JSON) flexible, ideal para los datos semiestructurados de las diferentes fuentes.
- **Permite Operaciones Espaciales Eficientes:** El uso nativo de GeoJSON y el índice 2dsphere en la colección edificios aborda directamente el requisito de "operaciones espaciales eficientes".
- **Facilita la Comparación de Fuentes:** El campo fuente en la colección edificios es la clave para cumplir con el mandato de "comparar los resultados de cada fuente". El análisis se puede ejecutar fácilmente filtrando por este campo.
- **Estandariza los Datos:** El esquema edificios crea un formato unificado. Lo más importante es el campo area_m2. Para los datos de Google, este campo se poblará directamente desde su fuente. Para los datos de Microsoft (que solo proveen la geometría), este campo se calculará durante el proceso de carga (ETL), asegurando que ambos datasets sean comparables para el objetivo final de "estimar el área total del techo".
- **Es Escalable:** La arquitectura de MongoDB está diseñada para escalar horizontalmente, lo que es necesario para manejar los "miles de millones de contornos de edificios" mencionados en el proyecto.