



**Documento de Diagnóstico y Plan de Acción**  
**Implementación de un Sistema de Alerta Temprana (SAT) de Deslizamientos**

**Octubre de 2025**

**Felipe Ruiz Zea**

**Ciencia de Datos e Ingeniería (Prueba Técnica para el puesto de analista de datos en mantenimiento)**

## Contenido

<b>1.</b>	<b><i>Resumen Ejecutivo</i></b>	<b>5</b>
1.1.	Contexto de la Problemática	5
1.2.	Alcance General	5
1.3.	Alcance Específico	6
1.4.	Consideraciones	6
1.5.	Resultado Esperado	7
<b>2.</b>	<b><i>Fuentes y Estructura de Datos</i></b>	<b>7</b>
2.1.	Panorama General de las Fuentes de Información	7
2.2.	Modelo de Datos Unificado	7
2.3.	Pipeline ETL Serverless: Especificación Técnica	8
<b>3.</b>	<b><i>Análisis Exploratorio y Definición del Modelo de Riesgo</i></b>	<b>9</b>
3.1.	Objetivo del Análisis	9
3.2.	Procesamiento de Datos Geoespaciales a Gran Escala	9
3.3.	Cálculo del Riesgo Estático Multifactorial	10
3.4.	Definición del Modelo de Alerta Dinámica: Matriz de Riesgo	11
3.5.	Validación a través del Prototipo Técnico	12
<b>4.</b>	<b><i>Propuesta de Arquitectura Técnica Escalable (AWS)</i></b>	<b>13</b>
4.1.	Principios de Diseño	14
4.2.	Arquitectura de Referencia Cloud-Native (AWS)	14
4.2.1.	Frontend (Capa de Visualización para el CSM)	14
4.2.2.	Backend (Capa de Ingesta y Lógica)	15
4.2.3.	Base de Datos (Capa de Persistencia y "Puente")	15
4.3.	Flujo de Datos en Producción	16
<b>5.</b>	<b><i>Plan de Acción e Implementación (Fase 1)</i></b>	<b>16</b>
5.1.	Objetivo de la Fase	16
5.2.	Cronograma de Implementación Detallado	17
5.3.	Entregables Finales de la Fase 1	17
5.4.	Requisitos y Supuestos para la Implementación	17
<b>6.</b>	<b><i>Criterios de Éxito y Próximos Pasos</i></b>	<b>18</b>
6.1.	Criterios de Éxito (Fase 1)	18
6.2.	Riesgos Potenciales y Plan de Mitigación	18
6.3.	Visión a Futuro y Próximos Pasos (Fase 2)	19
6.3.1.	Hoja de Ruta Tecnológica Detallada:	19

<b>6.3.2. Entrenamiento del Modelo de Machine Learning</b>	<b>19</b>
<b>6.3.3. Despliegue del Modelo como un Microservicio</b>	<b>20</b>
TABLE 1: FUENTES DE DATOS PARA EL SISTEMA DE ALERTA TEMPRANA .....	7
TABLE 2: ESTRUCTURA DE LA TABLA MAESTRA TORRES.....	8
TABLE 3 LÓGICA DE DECISIÓN PARA GENERACIÓN DE ALERTAS .....	11
TABLE 4 UMBRALES DE PRECIPITACIÓN POR NIVEL DE AMENAZA (EJEMPLO) .....	11
TABLE 5 CRONOGRAMA DE IMPLEMENTACIÓN .....	17
TABLE 6: MATRIZ DE RIESGOS.....	18
FIGURE 1-1:MAPA DE ARAUCA DONDE SE SIMULAN DÓNDE ESTÁN LAS TORRES TOLEDO SAMORÉ Y SAMORÉ – BANADÍA .....	5
FIGURE 2-1 FLUJO ETL .....	8
FIGURE 3-1 MAPA NACIONAL DE AMENAZA.....	9
FIGURE 3-2 DISTRIBUCIÓN DE VALORES DE AMENAZA.....	10
FIGURE 3-3: PROTOTIPO EN STREAMLIT -> VISUAL GENERAL DE DATOS .....	12
FIGURE 3-4: PROTOTIPO EN STREAMLIT -> MAPA CON ACTIVOS DE ISA Y SUS ALERTAS TEMPRANAS .....	13
FIGURE 3-5: PROTOTIPO EN STREAMLIT -> MATRIZ DE RIESGO VS PRECIPITACIÓN .....	13

### **Acrónimos de Entidades y Conceptos**

- SAT: Sistema de Alerta Temprana.
- SGC: Servicio Geológico Colombiano.
- IDEAM: Instituto de Hidrología, Meteorología y Estudios Ambientales (mencionado en el documento, es el nombre completo de la entidad).
- CSM: Centro de Supervisión y Maniobra.
- ETL: Extract-Transform-Load (Extraer, Transformar y Cargar).
- MDE: Modelo Digital de Elevación.

### **Acrónimos de Tecnologías y Servicios**

- AWS: Amazon Web Services.
- API: Application Programming Interface (Interfaz de Programación de Aplicaciones).
- RDS: Relational Database Service (Servicio de Base de Datos Relacional de AWS).
- ECR: Elastic Container Registry (Servicio de registro de contenedores de AWS).
- ALB: Application Load Balancer (Balanceador de Carga de Aplicaciones de AWS).
- VPC: Virtual Private Cloud (Nube Privada Virtual de AWS).
- IAM: Identity and Access Management (Gestión de Identidades y Accesos de AWS).

### **Otros Acrónimos**

- NDVI: Normalized Difference Vegetation Index (Índice de Vegetación de Diferencia Normalizada).
- SMAP: Soil Moisture Active Passive (Misión satelital de la NASA para medir la humedad del suelo).

# 1. Resumen Ejecutivo

## 1.1. Contexto de la Problemática

El departamento de Arauca, Colombia, presenta condiciones geográficas y climáticas que lo hacen altamente susceptible a fenómenos de movimientos en masa y deslizamientos de terreno, especialmente durante los períodos de lluvias intensas. Estas situaciones afectan de manera directa la infraestructura de transmisión de energía eléctrica, generando riesgos sobre torres, líneas y estaciones de ISA INTERCOLOMBIA (en adelante ISA).

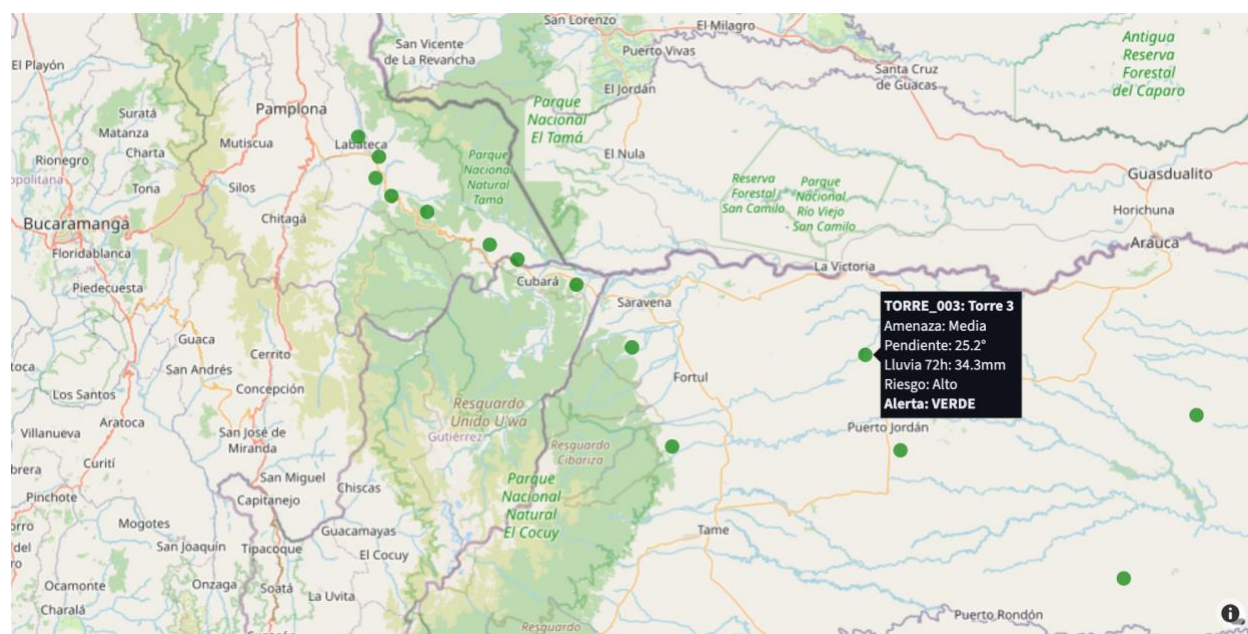


Figure 1-1: Mapa de Arauca donde se simulan dónde están las torres Toledo Samoré y Samoré – Banadía

En este contexto, se propone desarrollar un Sistema de Alertas Tempranas (SAT) orientado a identificar zonas críticas con alta probabilidad de deslizamientos. El objetivo es apoyar la toma de decisiones operativas durante eventos climáticos extremos, proporcionando señales oportunas sobre la vulnerabilidad del terreno y la exposición de activos críticos del sistema de transmisión, ver en la Figure 1-1.

La iniciativa se enmarca en los objetivos 2040 de ISA “Consolidar la transmisión de energía eléctrica” a partir de la gestión de riesgo climático y resiliencia operativa, alineados con la estrategia de seguridad energética y continuidad del servicio energético de Colombia.

## 1.2. Alcance General

Diseñar una arquitectura técnica de referencia para un sistema de alertas tempranas de deslizamientos, utilizando un enfoque de ciencia de datos, herramientas geoespaciales y tecnologías en la nube AWS, asumida como plataforma base.

### 1.3. Alcance Específico

- Analizar fuentes de datos relevantes (meteorológicas, topográficas, geológicas y de infraestructura eléctrica).
- Evaluar la disponibilidad y accesibilidad de los datos abiertos del Servicio Geológico Colombiano (SGC) y del IDEAM, identificando brechas y requerimientos.
- Simular un conjunto de datos representativos para validar el flujo de procesamiento y análisis espacial.
- Prototipar una visualización exploratoria en Streamlit que permita demostrar la integración y potencial del modelo.
- Proponer una arquitectura de nube escalable que soporte ingesta continua, modelado predictivo y visualización operacional.
- Definir un plan de acción a corto plazo (en días) que establezca los pasos iniciales para pasar del diagnóstico al desarrollo técnico.

### 1.4. Consideraciones

El presente documento corresponde a la fase diagnóstica y exploratoria del proyecto. Durante esta fase:

- Se asume el uso de AWS como entorno de nube principal para el despliegue de los componentes del sistema (ETL, modelado, almacenamiento y visualización).
- Para validar los flujos de integración, se utilizaron datos simulados del Servicio Geológico Colombiano (SGC) debido a la naturaleza de su disponibilidad. En cuanto a los datos meteorológicos, si bien el IDEAM es la fuente oficial y cuenta con una API pública, para agilizar el desarrollo de este prototipo se empleó temporalmente la API de Open-Meteo por su facilidad de implementación. La arquitectura propuesta está diseñada para conectarse a los servicios del IDEAM en la fase productiva.
- La arquitectura propuesta es agnóstica, y puede replicarse en GCP o Azure sin cambios estructurales significativos.
- El prototipo Streamlit desarrollado tiene un carácter demostrativo, y no pretende ser el producto final, sino un vehículo para validar la factibilidad técnica del flujo de datos y visualización geoespacial.
- Los conjuntos de datos del SGC no ofrecen actualizaciones en tiempo real, limitando la capacidad predictiva inmediata.
- Se carece de registros consistentes de eventos de deslizamiento georreferenciados para entrenar modelos supervisados.
- Se debe garantizar interoperabilidad con los sistemas internos de ISA (por ejemplo, SCADA o bases de datos de mantenimiento).

## 1.5. Resultado Esperado

La entrega de este diagnóstico tiene como propósito dejar definido un modelo técnico conceptual y funcional, que establezca los componentes mínimos para la implementación de un Sistema de Alertas Tempranas (SAT) con capacidad de:

- Recibir, procesar y analizar datos geoespaciales y meteorológicos.
- Identificar áreas críticas de amenaza.
- Emitir alertas configurables a partir de umbrales definidos.
- Servir como base para la implementación futura en producción en la nube (AWS).

## 2. Fuentes y Estructura de Datos

### 2.1. Panorama General de las Fuentes de Información

Para el desarrollo del sistema de alertas tempranas de deslizamientos en Arauca, se requiere la integración de múltiples fuentes de datos heterogéneas, que combinan información geológica, topográfica, meteorológica e infraestructural.

Estas fuentes tienen distintos niveles de disponibilidad, frecuencia de actualización y formatos de entrega, lo que implica la necesidad de una arquitectura flexible de ingesta y almacenamiento en la nube.

En la Table 1, se presentan las principales fuentes identificadas, diferenciando su estado de disponibilidad real, su formato original, y las suposiciones o simulaciones utilizadas en esta fase.

Table 1: Fuentes de Datos para el Sistema de Alerta Temprana

Fuente	Tipo	Formato / Acceso	Frecuencia	Disponibilidad real	Uso en prototipo
Servicio Geológico Colombiano (SGC) – Zonificación de Amenaza	Estático	GeoTIFF / Shapefile	Anual	Parcial (sin API pública)	Simulada (categorías oficiales)
IDEAM – Pronóstico Meteorológico	Dinámico	API REST (JSON)	Diario / Horario	publica	Simulada (valores estimados)
Infraestructura de ISA INTERCOLOMBIA	Estático	CSV / Base de datos corporativa	Variable	Restringida	Simulada (15 torres de prueba)
Historial de Deslizamientos	Histórico	CSV / Excel	Eventual	Escasa / No pública	Simulada con registros sintéticos
Modelo Digital de Elevación (MDE – NASA / SRTM)	Estático	GeoTIFF	Fijo	Pública (copias en AWS Open Data)	Usada directamente

### 2.2. Modelo de Datos Unificado

La estrategia central de la arquitectura de datos es la creación de un Modelo de Datos Unificado. En lugar de procesar datos de múltiples fuentes en tiempo real durante cada consulta, un proceso de backend se encargará de consolidar toda la información relevante en una única tabla maestra Table 2 en la base de datos. Esta tabla, que

llamaremos torres, será el corazón del sistema, donde cada registro representa una torre enriquecida con todas sus variables de riesgo.

Table 2: Estructura de la Tabla Maestra torres

Columna	Descripción	Tipo de Dato	Fuente / Cálculo	Frecuencia
id_torre	Identificador único del activo.	VARCHAR	Infraestructura ISA	Estático
nombre	Nombre o designación de la torre.	VARCHAR	Infraestructura ISA	Estático
geometria	Coordenadas geográficas.	GEOMETRY(Point)	Infraestructura ISA	Estático
amenaza_sgc	Nivel de amenaza (Baja, Media, Alta).	VARCHAR	SGC (Pre-procesado)	Estático
pendiente_grados	Inclinación del terreno en grados.	FLOAT	MDE (Pre-procesado)	Estático
lluvia_acum_72h_mm	Precipitación acumulada en las últimas 72h.	FLOAT	API Meteorológica	Cada hora
nivel_alerta	Estado de alerta actual (Verde, Amarilla, Roja).	VARCHAR	Matriz de Riesgo	Cada hora
ultima_actualizacion	Timestamp de la última actualización de datos.	TIMESTAMP	Proceso Backend	Cada hora

2.3. Pipeline ETL Serverless: Especificación Técnica

El núcleo del sistema es un pipeline ETL (Extract-Transform-Load) automatizado y serverless para minimizar la gestión y el costo.

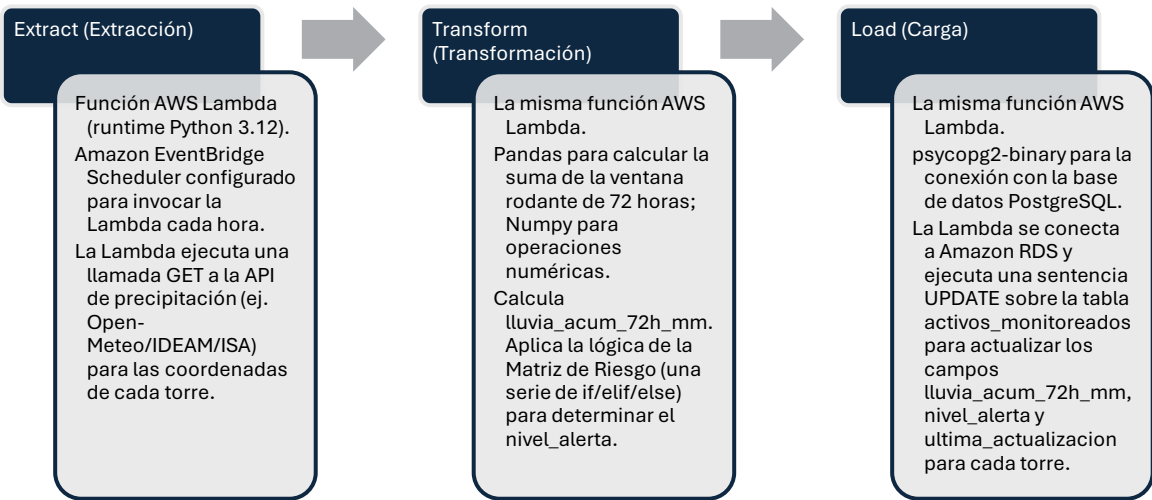


Figure 2-1 Flujo ETL



### 3. Análisis Exploratorio y Definición del Modelo de Riesgo

#### 3.1. Objetivo del Análisis

El objetivo de esta fase fue traducir datos geográficos y meteorológicos crudos en una lógica de alerta operacional. Las metas específicas fueron:

- Procesar el Mapa Nacional de Amenaza (AmeMM\_100k.tif) a escala completa para entender su distribución estadística real.
- Cuantificar múltiples factores de riesgo estático para cada torre.
- Definir una Matriz de Alerta dinámica que combine el riesgo estático con el disparador de precipitación.
- Validar la lógica y la visualización a través de un prototipo técnico funcional.

#### 3.2. Procesamiento de Datos Geoespaciales a Gran Escala

El insumo principal, el Mapa Nacional de Amenaza (AmeMM\_100k.tif), es un archivo GeoTIFF de alta resolución. Para optimizar el uso de recursos computacionales, se implementó una estrategia de procesamiento por bloques (chunking) utilizando la librería Rasterio en Python.

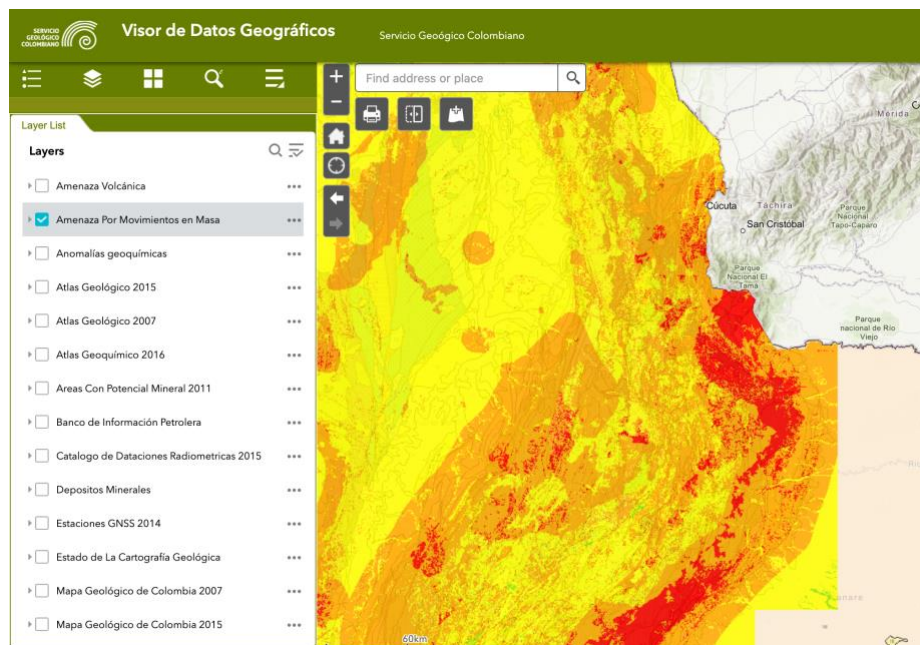


Figure 3-1 Mapa Nacional de Amenaza

Este método consiste en leer y procesar el archivo en fragmentos secuenciales, lo que permite realizar cálculos sobre la totalidad del dataset sin necesidad de cargarlo por completo en memoria. Este enfoque no solo previene el consumo excesivo de RAM, sino que también garantiza que la metodología sea robusta y escalable para datasets de escala nacional.

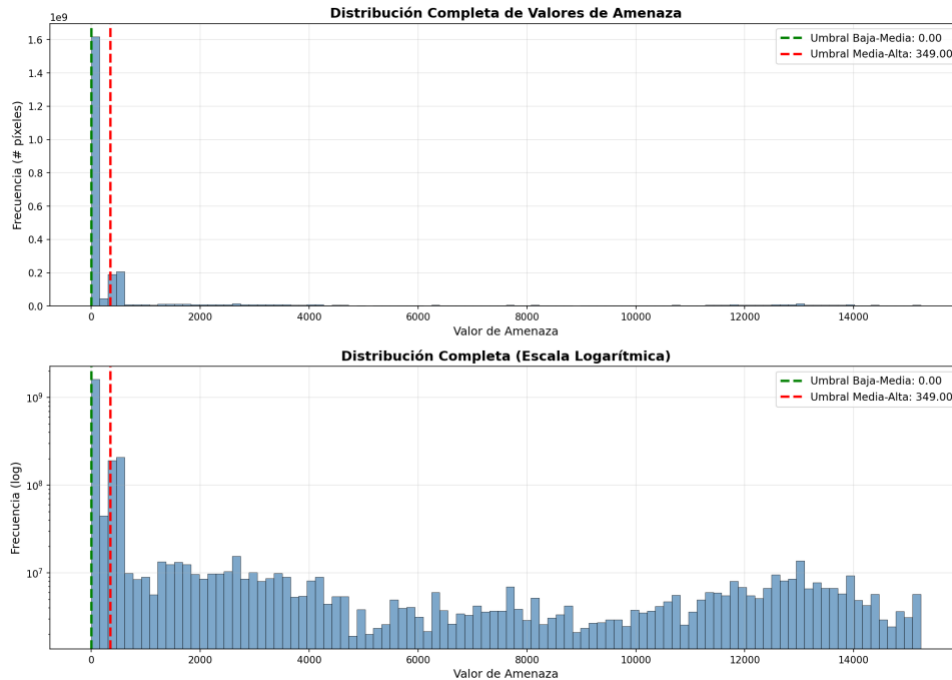


Figure 3-2 Distribución de valores de amenaza

Si bien el análisis a nivel de píxel. Figure 3-2 fue exitoso para caracterizar la amenaza, se detectaron desafíos en la correcta localización espacial de los datos, lo que introduce un riesgo en la precisión del modelo.

Como acción inmediata, se ha definido la necesidad de consultar al SGC para clarificar el sistema de coordenadas y asegurar una carga de datos geográficamente precisa.

### 3.3. Cálculo del Riesgo Estático Multifactorial

Con un entendimiento completo de la distribución de los datos del SGC, se procedió a construir un modelo de riesgo estático multifactorial para determinar la vulnerabilidad inherente de cada torre, combinando dos factores clave.

- **Amenaza Geológica (SGC):** Se propone utilizar el Mapa Nacional de Amenaza como insumo base. La metodología consistiría en cruzar las coordenadas de cada torre con este mapa para asignarle una de las cinco categorías de amenaza definidas por el SGC: Muy Baja, Baja, Media, Alta o Muy Alta.
- **Factor Topográfico (Pendiente):** Se propone calcular la pendiente del terreno en la ubicación exacta de cada torre. Este valor, expresado en grados, se derivaría de un Modelo Digital de Elevación (MDE). Este factor es un indicador crítico, ya que se asume que una mayor pendiente incrementa sustancialmente el riesgo de inestabilidad del terreno.

Se propone que estos dos factores se combinen para generar una Clasificación\_Riesgo unificada (ej. Bajo, Medio, Alto). Este indicador representaría el riesgo base de cada torre, sin considerar aún la influencia de la precipitación.

### 3.4. Definición del Modelo de Alerta Dinámica: Matriz de Riesgo

El núcleo del sistema es un motor de reglas que genera la alerta operacional de forma dinámica. Esta lógica, implementada en la función calcular\_nivel\_alerta, integra dos dimensiones:

- Un factor de riesgo estático, correspondiente al nivel de Amenaza SGC de la torre.
- Un disparador dinámico, basado en la precipitación acumulada en las últimas 72 horas.

El modelo opera consultando una tabla de configuración (umbrales\_lluvia.csv) que contiene los umbrales de lluvia específicos para cada categoría de amenaza. El proceso de decisión se resume en la siguiente tabla:

Table 3 Lógica de Decisión para Generación de Alertas

Paso del Proceso	Nivel de Amenaza (SGC)	Precipitación Acumulada (72h)	Alerta Resultante
1. Búsqueda de Umbrales	Se busca el nivel de amenaza de la torre en la tabla de configuración.	-	Se obtienen los umbrales Amarillo y Rojo para ese nivel.
2. Evaluación de Alerta Roja	(Umbrales encontrados)	Lluvia $\geq$ Umbral_Rojo	ROJA ●
3. Evaluación de Alerta Amarilla	(Umbrales encontrados)	Umbral_Amarillo $\leq$ Lluvia < Umbral_Rojo	AMARILLA ○
4. Condición Normal	(Umbrales encontrados)	Lluvia < Umbral_Amarillo	VERDE ●
Caso por Defecto	(No se encuentran umbrales para el nivel de amenaza)	Cualquiera	VERDE ●

Table 4 Umbrales de Precipitación por Nivel de Amenaza (Ejemplo)

Nivel de Amenaza SGC	Alerta ● AMARILLA	Alerta ● ROJA
Muy Baja	> 250 mm	> 300 mm
Baja	> 200 mm	> 250 mm
Media	> 150 mm	> 200 mm
Alta	> 100 mm	> 120 mm
Muy Alta	> 80 mm	> 100 mm

Este enfoque granular es significativamente más preciso, ya que reconoce que una torre en una zona de amenaza "Muy Alta" puede entrar en estado crítico con una fracción de la lluvia que necesitaría una torre en una zona de amenaza "Baja". Esta es la lógica exacta que se implementará en la función AWS Lambda.

### 3.5. Validación a través del Prototipo Técnico

El prototipo en Streamlit fue la herramienta para validar este modelo multifactorial de extremo a extremo. Su función no fue ser el producto final, sino mitigar el riesgo del proyecto al confirmar que Es factible integrar múltiples capas de datos (coordenadas, amenaza, pendiente, lluvia) en un único sistema.



Figure 3-3: Prototipo en Streamlit -> visual general de datos

Es factible la integración de datos heterogéneos: Se validó con éxito la capacidad de unificar capas de información geoespacial (coordenadas de torres), datos ráster (amenaza SGC), datos derivados (pendiente del terreno) y datos dinámicos de una API externa (precipitación) en un único modelo de datos coherente.

La lógica del modelo de riesgo es operacionalmente viable: La implementación de la matriz de riesgo dinámica (calcular\_nivel\_alerta) demostró que la lógica de umbrales puede procesar los datos para generar alertas consistentes en tiempo real.



Figure 3-4: Prototipo en Streamlit -> Mapa con activos de ISA y sus alertas tempranas

La visualización de resultados es efectiva: Las herramientas gráficas, como el mapa geoespacial interactivo y los análisis detallados por torre, fueron validadas como una interfaz intuitiva y de alto valor para comunicar la conciencia situacional a los operadores del CSM, cumpliendo con el objetivo principal de la visualización.

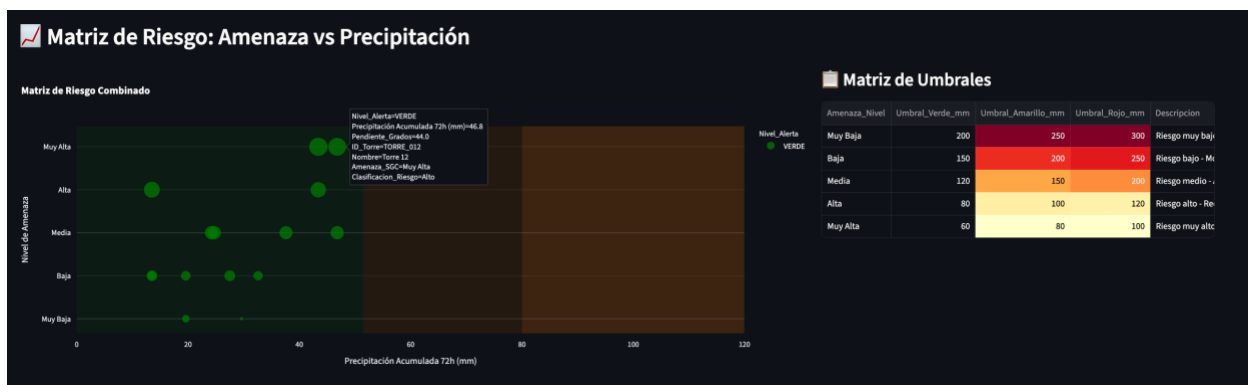
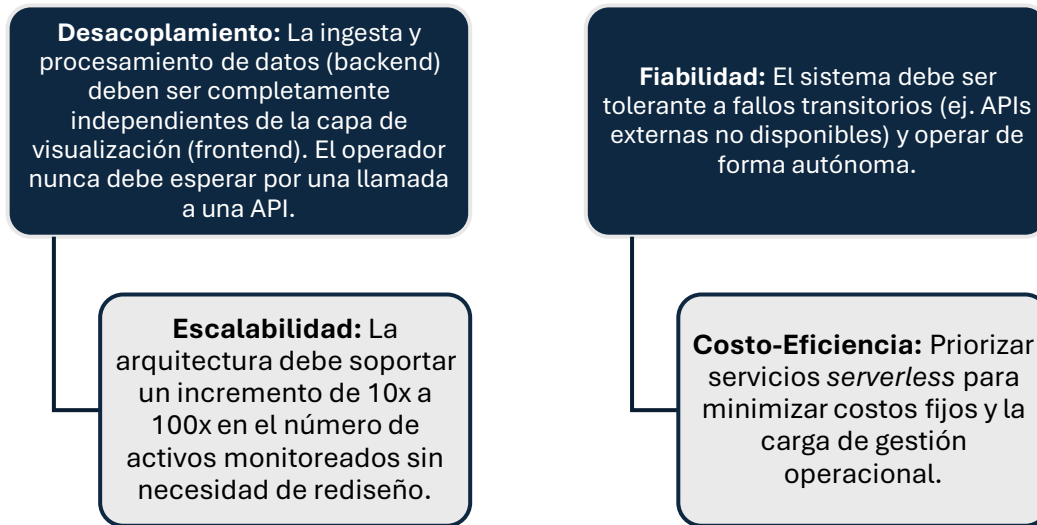


Figure 3-5: Prototipo en Streamlit -> Matriz de riesgo Vs precipitación

## 4. Propuesta de Arquitectura Técnica Escalable (AWS)

El prototipo técnico validó la lógica del modelo, pero su arquitectura monolítica (donde la interfaz de usuario está acoplada al procesamiento de datos) no es viable para un entorno de producción debido a la alta latencia y la falta de escalabilidad. Para la solución operacional, se propone una arquitectura de microservicios desacoplada y nativa de la nube, diseñada para un rendimiento óptimo y una alta fiabilidad.

#### 4.1. Principios de Diseño

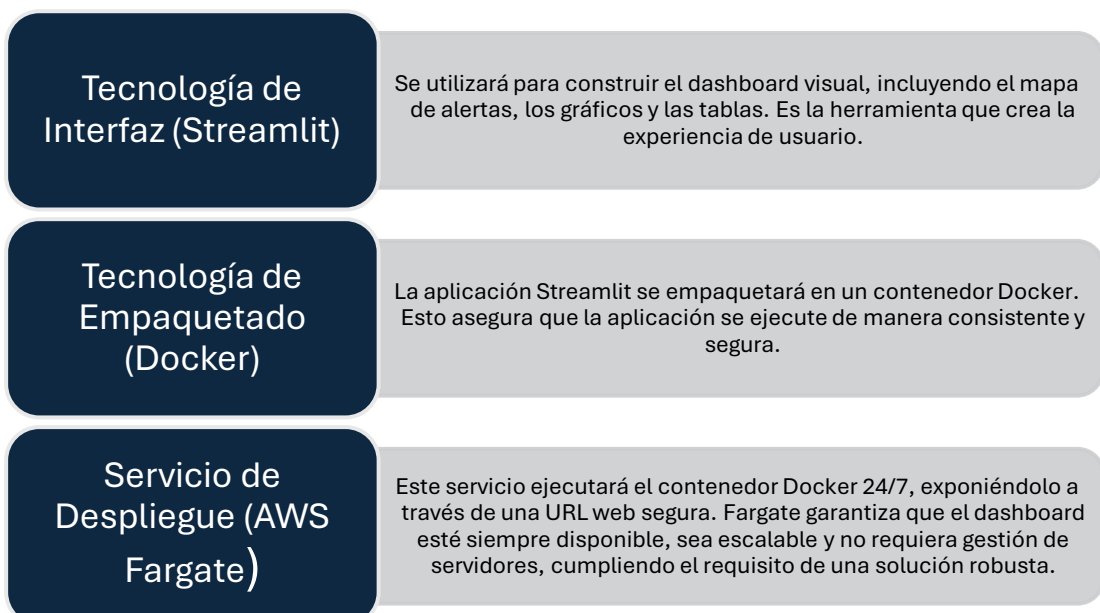


#### 4.2. Arquitectura de Referencia Cloud-Native (AWS)

La solución se desplegará íntegramente en Amazon Web Services, utilizando servicios gestionados para cada capa lógica.

##### 4.2.1. Fronted (Capa de Visualización para el CSM)

Esta capa es la interfaz de usuario final que consumirán los operadores del CSM. Su único propósito es traducir los datos procesados por el backend en una visualización gráfica, interactiva y fácil de interpretar, cumpliendo con el requisito principal del proyecto.



#### 4.2.2. Backend (Capa de Ingesta y Lógica)

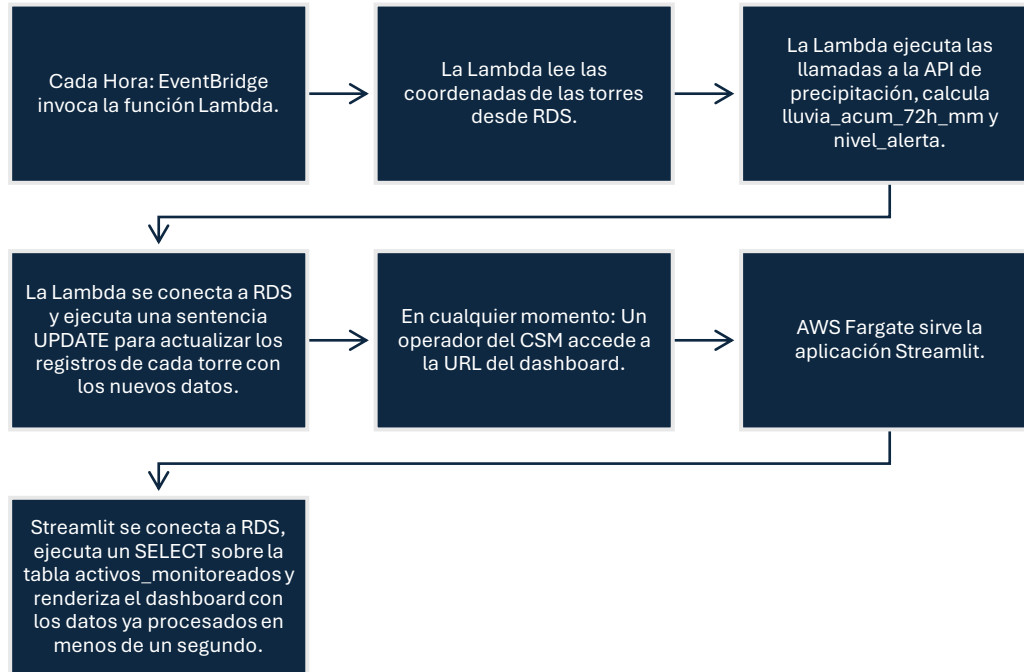
Tecnología:	Función AWS Lambda con runtime de Python 3.12.
Orquestación	Amazon EventBridge Scheduler. Se configurará una regla para invocar la función Lambda de forma periódica, es decir, al inicio de cada hora.
Función	Esta función contendrá la lógica de negocio principal: consultar la API de precipitación para cada torre, calcular los acumulados y aplicar la Matriz de Riesgo para determinar el nivel de alerta

#### 4.2.3. Base de Datos (Capa de Persistencia y "Puente")

Tecnología	Amazon RDS (Relational Database Service) con el motor PostgreSQL y la extensión geoespacial PostGIS.
El backend	(Lambda) escribe en ella el estado actualizado de cada torre cada hora.
El frontend	(Streamlit) solo lee de ella, ejecutando una consulta SQL simple y rápida.

### 4.3. Flujo de Datos en Producción

El flujo de trabajo en la arquitectura propuesta es el siguiente:



## 5. Plan de Acción e Implementación (Fase 1)

### 5.1. Objetivo de la Fase

El objetivo de esta fase es desplegar la arquitectura descrita en la Sección 4 para tener un Prototipo Operacional Mínimo Viable (MVP) completamente funcional en un plazo de 5 días hábiles. Este plan está diseñado para una ejecución ágil por un equipo con experiencia en AWS y Python.



## 5.2. Cronograma de Implementación Detallado

El plan se divide en tareas diarias, con entregables específicos para validar el progreso.

Table 5 Cronograma de implementación

Día	Actividad Clave	Tecnologías / Servicios	Entregable
1	Configuración de Infraestructura Base	AWS RDS (PostgreSQL + PostGIS), VPC, Security Groups, IAM Roles	Instancia de base de datos aprovisionada y accesible, con el esquema de tablas creado. Roles de seguridad definidos.
2	Carga de Datos Estáticos y Containerización	Python (psycopg2), Docker, AWS ECR	1. Datos de torres, amenaza (raster clasificado) y pendiente cargados en RDS. 2. Imagen Docker del dashboard de Streamlit construida y subida a ECR.
3	Despliegue del Frontend	AWS Fargate, Application Load Balancer (ALB)	URL pública y funcional del dashboard, mostrando los datos estáticos leídos desde RDS.
4	Despliegue del Backend Serverless	AWS Lambda, Amazon EventBridge Scheduler	Función Lambda desplegada y configurada para ejecutarse cada hora, actualizando exitosamente los datos de precipitación en RDS.
5	Integración, Seguridad y Entrega	AWS Cognito, Pruebas End-to-End	1. Sistema 100% funcional e integrado. 2. Acceso al dashboard restringido por autenticación. 3. Sesión de entrega y handover al CSM.

## 5.3. Entregables Finales de la Fase 1

Al concluir el quinto día, el equipo del CSM tendrá acceso a:

- Un Dashboard Operacional Funcional: Una URL segura para acceder al sistema de monitoreo en tiempo real.
- Infraestructura como Código (IaC) - Opcional: Plantillas de Terraform o AWS CloudFormation que definen toda la infraestructura, permitiendo su replicación o modificación de manera controlada.
- Documentación Técnica y de Usuario: Un resumen de la arquitectura desplegada y una guía rápida para los operadores del CSM.

## 5.4. Requisitos y Supuestos para la Implementación

- Permisos de AWS: Se requiere acceso a la cuenta de AWS de ISA con permisos suficientes para crear los recursos mencionados (RDS, Fargate, Lambda, IAM).
- Equipo Técnico: Se asume la disponibilidad de al menos un ingeniero con experiencia en DevOps/Cloud y Python.

- Datos de Entrada: Se debe contar con el archivo de coordenadas de las torres y el raster clasificado de amenaza al inicio del Día 1.

## 6. Criterios de Éxito y Próximos Pasos

### 6.1. Criterios de Éxito (Fase 1)

El éxito de esta fase de implementación rápida se medirá a través de criterios operacionales y técnicos cuantificables:

1. Entrega a Tiempo: El sistema completo está desplegado y 100% funcional en la infraestructura de AWS en el plazo de 5 días hábiles.
2. Rendimiento del Frontend: El tiempo de carga inicial del dashboard de Streamlit y las interacciones del usuario deben ser inferiores a 2 segundos.
3. Fiabilidad del Backend: La tasa de ejecución exitosa de la función AWS Lambda debe ser superior al 99.9%. Los datos en el dashboard deben tener una antigüedad máxima de 1 hora.
4. Adopción por el Usuario: El equipo del Centro de Supervisión y Maniobra (CSM) debe validar formalmente que la herramienta es intuitiva, fiable y que aporta un valor tangible a su conciencia situacional durante la operación.

### 6.2. Riesgos Potenciales y Plan de Mitigación

Se han identificado los siguientes riesgos técnicos para la Fase 1, junto con sus planes de mitigación:

Table 6: Matriz de riesgos

Riesgo	Impacto	Plan de Mitigación
<b>Fiabilidad de la API de Precipitación</b>	Medio	La función AWS Lambda se implementará con una política de reintentos con exponential backoff. Se configurará una alarma en Amazon CloudWatch para notificar al equipo técnico en caso de fallos persistentes de la API externa.
<b>Precisión de los Umbrales Iniciales</b>	Bajo	Los umbrales de la Matriz de Riesgo se almacenarán como parámetros configurables. El sistema está diseñado para que estos puedan ser ajustados fácilmente a medida que se recopilen datos de eventos reales y se obtenga retroalimentación de campo.
<b>Calidad del Mapa de Amenaza (SGC)</b>	Bajo	El riesgo de una baja resolución en el mapa SGC se mitiga al incorporar la pendiente del terreno como una segunda variable de riesgo estático. Esto proporciona una capa adicional de granularidad y validación.

### 6.3. Visión a Futuro y Próximos Pasos (Fase 2)

La arquitectura de la Fase 1 está diseñada explícitamente para ser el cimiento de la Fase 2: Evolución a un Modelo Predictivo. El sistema inicial es determinístico: responde a la pregunta "¿Se cumplen las condiciones de riesgo?". El siguiente paso estratégico es evolucionar hacia un sistema que utilice machine learning para responder a una pregunta mucho más potente: "¿Cuál es la probabilidad de que ocurra un evento en esta ubicación?".

Esta transición convierte el SAT de un sistema de monitoreo a una herramienta de pronóstico inteligente.

#### 6.3.1. Hoja de Ruta Tecnológica Detallada:

**Consolidación del Historial de Eventos:** El activo más valioso que se generará es el dataset de verdad terreno (ground truth dataset). La prioridad será alimentar la tabla eventos\_historicos en Amazon RDS con datos de deslizamientos reales (fecha, coordenada, magnitud). Este historial es indispensable, ya que es la única forma de validar objetivamente la precisión del modelo y, más importante, de entrenarlo para que reconozca patrones que llevaron a fallas reales en el pasado.

**Ingeniería de Características (Feature Engineering):** Para que el modelo "aprenda" mejor, necesitamos darle más contexto. Se integrarán nuevas fuentes de datos para crear características (features) que capturen la física del fenómeno de manera más directa que solo la lluvia:

**Humedad del Suelo Satelital:** Utilizando datos de misiones como SMAP de la NASA (disponibles en AWS Open Data), pasaremos de inferir la saturación del suelo a partir de la lluvia a tener una medida directa. Esto es un salto cualitativo en la precisión del modelo.

**Índice de Vegetación (NDVI):** Calculado a partir de imágenes de Sentinel-2. Una vegetación sana y densa (alto NDVI) suele estar correlacionada con una mayor estabilidad del suelo. Un cambio brusco en el NDVI podría ser un precursor de inestabilidad.

#### 6.3.2. Entrenamiento del Modelo de Machine Learning

**Tecnología:** Se utilizará el ecosistema de Amazon SageMaker para gestionar todo el ciclo de vida del modelo (entrenamiento, evaluación y despliegue). Esto elimina la necesidad de gestionar infraestructura de cómputo para el entrenamiento.

**Algoritmo:** Se entrenará un modelo de clasificación basado en árboles de decisión potenciados por gradiente, como XGBoost o LightGBM. Estos algoritmos son el estándar de la industria para datos tabulares (nuestro caso de uso) porque son extremadamente eficientes y capaces de capturar interacciones complejas y no lineales entre las variables (por ejemplo, cómo 100mm de lluvia afectan de manera diferente a una pendiente de 20° vs una de 40°).

### 6.3.3. Despliegue del Modelo como un Microservicio

El modelo entrenado se desplegará como un endpoint en Amazon SageMaker, que es esencialmente una API privada y escalable para hacer predicciones. La arquitectura del sistema evolucionará de la siguiente manera:

La función AWS Lambda será modificada. En lugar de ejecutar la lógica de la Matriz de Riesgo, su nueva tarea será:

1. Recopilar todas las características en tiempo real (lluvia, humedad del suelo, etc.).
2. Enviar estos datos como una solicitud al endpoint de SageMaker.
3. Recibir una respuesta con la probabilidad de deslizamiento (ej. {probabilidad: 85%}).

Este puntaje de riesgo probabilístico (0-100%) es mucho más granular y accionable para el CSM que una simple alerta roja, permitiendo una priorización mucho más fina de los recursos.