Comparativa de Soluciones para la Aplicación de Denuncia de Cultivos Ilícitos según diferentes tecnologías

1. Opción 1: Aplicación con React, TypeScript, Google Maps API, Spring Boot y Power BI (Nuestra Elección)

Descripción de la Solución:

- **Frontend:** React con TypeScript
- **Geolocalización:** Google Maps API
- **Backend:** Spring Boot con base de datos SQL
- **Análisis de Datos:** Power BI

Ventajas:

- **Escalabilidad:** React y Spring Boot permiten crear una aplicación escalable, capaz de manejar una gran cantidad de usuarios y datos sin comprometer el rendimiento.
- **Precisión en la Geolocalización:** Google Maps API proporciona mapas detallados y precisos, facilitando la interacción del usuario con los mapas para localizar los cultivos ilícitos.
- **Mantenimiento Sencillo:** El uso de TypeScript ayuda a identificar errores durante el desarrollo, asegurando un código limpio y fácil de mantener.
- **Capacidad Analítica Avanzada:** Power BI ofrece informes dinámicos, visualización de datos en tiempo real y mapas de calor para facilitar el análisis geoespacial de los cultivos.
- **Solución Integral:** Al integrar Google Maps API y Power BI, la solución cubre tanto el aspecto de visualización geográfica como el análisis de datos, optimizando la toma de decisiones y proporcionando una visión completa.

Desventajas:

- **Curva de aprendizaje:** TypeScript y React pueden tener una curva de aprendizaje inicial para los desarrolladores sin experiencia en estas tecnologías.
- **Costos de API y Licencias:** Google Maps API y Power BI pueden tener costos asociados según el uso, especialmente cuando la cantidad de datos o usuarios aumenta significativamente.

Justificación de Elección: Elegimos esta solución porque nos ofrece un equilibrio entre escalabilidad, mantenibilidad y funcionalidades avanzadas. React con TypeScript es una opción sólida para el frontend, que garantiza una aplicación

robusta y fácilmente escalable. Google Maps API es la mejor opción para garantizar precisión en la geolocalización, y Power BI nos permite crear análisis visuales sin necesidad de desarrollar soluciones analíticas internas desde cero. Aunque los costos de las APIs pueden aumentar, consideramos que el valor añadido justifica la inversión.

---

2. Opción 2: Aplicación con Angular, Node.js, OpenStreetMap, MongoDB y Grafana

Descripción de la Solución:

- **Frontend:** Angular
- **Geolocalización:** OpenStreetMap (alternativa gratuita a Google Maps)
- **Backend:** Node.js con base de datos NoSQL (MongoDB)
- **Análisis de Datos:** Grafana

Ventajas:

- **Frontend Dinámico y Modular:** Angular es un framework robusto que ofrece herramientas nativas para gestionar componentes y hacer que la aplicación sea modular y fácilmente extensible.
- **Base de Datos NoSQL:** MongoDB permite una alta flexibilidad en la estructura de datos, lo que es útil si se anticipan cambios frecuentes en los tipos de datos que se manejan.
- **Geolocalización Gratuita:** OpenStreetMap es una alternativa gratuita y de código abierto a Google Maps, lo que reduce costos.
- **Visualización con Grafana:** Grafana permite generar gráficos y paneles de control visuales a partir de los datos, similar a Power BI, pero de código abierto.

Desventajas:

- **Precisión Limitada de Mapas:** OpenStreetMap, aunque gratuito, no tiene la misma precisión o nivel de detalle que Google Maps en algunas áreas geográficas, lo cual podría comprometer la exactitud de las denuncias.
- **Mayor Complejidad Técnica:** Node.js junto con MongoDB y Angular puede tener una curva de aprendizaje mayor, y la integración de herramientas como Grafana requiere una mayor personalización.
- **Problemas de Consistencia:** Al usar una base de datos NoSQL, la consistencia de los datos puede ser un desafío, especialmente cuando se

requiere realizar transacciones o mantener integridad entre las diferentes denuncias.

Justificación de Descarto: Si bien esta solución tiene la ventaja de reducir costos con herramientas de código abierto como OpenStreetMap y Grafana, la precisión de OpenStreetMap no es suficiente para un proyecto en el que la geolocalización precisa es clave. Además, la falta de transacciones en MongoDB y la complejidad adicional en la integración de Grafana hacen que esta solución no sea la mejor opción para nuestros objetivos.

---

3. Opción 3: Aplicación con Vue.js, Flask, PostgreSQL y Mapbox

Descripción de la Solución:

- **Frontend:** Vue.js
- **Geolocalización:** Mapbox (alternativa a Google Maps con costos más bajos)
- **Backend:** Flask (Python)
- **Base de Datos Relacional:** PostgreSQL

Ventajas:

- **Frontend Ligero:** Vue.js es un framework liviano y fácil de integrar, ideal para proyectos que requieren flexibilidad en la interfaz de usuario sin la sobrecarga de frameworks más pesados.
- **Backend Simplificado:** Flask es un framework minimalista que permite la rápida creación de APIs, ideal para aplicaciones web pequeñas y medianas que no requieren una arquitectura compleja.
- **Base de Datos Relacional Sólida:** PostgreSQL es una base de datos relacional avanzada, conocida por su capacidad para manejar grandes volúmenes de datos y realizar consultas complejas.
- **Mapbox:** Ofrece funcionalidades similares a Google Maps, pero con opciones de personalización y precios más bajos, lo que lo hace ideal para proyectos con un presupuesto ajustado.

Desventajas:

- **Menos Soporte para Vue.js:** Aunque Vue.js es sencillo y flexible, no tiene el mismo nivel de soporte ni la comunidad que React o Angular, lo que puede ser un problema a medida que la aplicación crezca.
- **Menos Funcionalidades que Spring Boot:** Flask es un framework minimalista, lo que implica menos funcionalidades "out of the box" en

comparación con Spring Boot, lo que puede aumentar el tiempo de desarrollo si se requieren soluciones adicionales.

- **Costos de Mapbox:** Aunque más barato que Google Maps, Mapbox sigue siendo un servicio de pago, lo cual puede resultar costoso si el número de usuarios aumenta.

Justificación de Descarto: Esta solución ofrece algunas ventajas, como la simplicidad de Vue.js y Flask, y una alternativa más económica a Google Maps con Mapbox. Sin embargo, carece de las funcionalidades robustas que ofrece React y Spring Boot, especialmente para aplicaciones que requieren escalabilidad y mantenimiento a largo plazo. Además, aunque PostgreSQL es una base de datos sólida, el ecosistema de desarrollo de Flask no se adapta tan bien a nuestros requisitos como lo hace Spring Boot.