

**DESIGNIOUS**

**"Transformando cultivos, transformando vidas"**

**Diseño básico de componentes**



**Popayán, Cauca  
2024**

# Índice

<b>Índice</b>	<b>2</b>
<b>1. Enfoque de arquitectura</b>	<b>3</b>
<b>2. Frontend (Interfaz de Usuario)</b>	<b>3</b>
<b>3. Backend (API Restful)</b>	<b>4</b>
3.1. Lenguajes y Frameworks . . . . .	4
3.2. Servicios API . . . . .	4
3.3. Procesamiento de Imágenes . . . . .	5
<b>4. Base de Datos</b>	<b>5</b>
4.1. NoSQL . . . . .	5
4.2. SQL (Opcional) . . . . .	5
<b>5. Almacenamiento de Imágenes</b>	<b>5</b>
<b>6. Visualización de Imágenes Satelitales Segmentadas</b>	<b>6</b>
<b>7. Servicios de Nube y DevOps</b>	<b>6</b>
7.1. Plataformas de Nube . . . . .	6
<b>8. Monitoreo y Escalabilidad</b>	<b>7</b>
8.1. Monitoreo . . . . .	7
8.2. Escalabilidad . . . . .	7

## 1. Enfoque de arquitectura

Para el desarrollo de Designious, los componentes clave que conectaremos son fundamentales para garantizar el éxito de la startup en la recolección, análisis y venta de información sobre cultivos ilícitos. En primer lugar, el frontend (web y móvil) permitirá a los usuarios reportar cultivos ilícitos en tiempo real usando mapas satelitales. El backend será la columna vertebral del sistema, gestionando los reportes, procesando imágenes satelitales y conectando con bases de datos como MongoDB o PostgreSQL para almacenar los datos de ubicación y análisis. La integración con APIs como Google Maps y Google Earth Engine permitirá obtener imágenes satelitales y hacer segmentación de áreas críticas, mientras que servicios de almacenamiento en la nube como AWS S3 o Google Cloud Storage facilitarán el manejo eficiente de grandes volúmenes de imágenes si es necesario, puede ser opcional. Además, es crucial implementar autenticación segura con Firebase Authentication para proteger la información sensible de los reportes, y asegurar que solo usuarios autorizados accedan a los datos. El uso de tecnologías en la nube como AWS o Google Cloud Platform permitirá escalar el sistema a medida que aumente el volumen de datos y usuarios, mientras que Docker y Kubernetes proporcionarán la flexibilidad y escalabilidad necesarias para mantener la infraestructura de manera eficiente y segura. Estos componentes son esenciales para que Designious pueda ofrecer un servicio confiable, seguro y escalable desde su inicio.

## 2. Frontend (Interfaz de Usuario)

El frontend será la plataforma que permite a los usuarios reportar cultivos ilícitos y acceder a los informes geospaciales generados por **Designious**. Las tecnologías recomendadas incluyen:

- **React.js**: Para crear una interfaz rápida y eficiente para la web donde los usuarios pueden registrar los reportes.
- **React Native o Flutter**: Para el desarrollo de una aplicación móvil que permita el reporte de cultivos ilícitos desde dispositivos móviles.
- **Leaflet.js o Google Maps JavaScript API**: Para visualizar mapas interactivos y permitir a los usuarios ubicar los cultivos ilícitos en

coordenadas específicas.

- **Axios o Fetch API:** Para realizar solicitudes HTTP al backend y manejar la interacción con la API de **Designious**.
- **Material-UI o Bootstrap:** Para proporcionar un diseño responsivo y profesional a la interfaz del usuario.

### 3. Backend (API Restful)

El backend será responsable de recibir y procesar los reportes de los cultivos ilícitos, además de gestionar la generación de informes y el acceso a los datos satelitales.

#### 3.1. Lenguajes y Frameworks

- **Node.js con Express.js:** Ideal para construir una API RESTful escalable que gestione los reportes de cultivos ilícitos.
- **Python con Flask o Django:** Alternativa para el backend si se requiere mayor control en el procesamiento de imágenes satelitales y análisis de datos.

#### 3.2. Servicios API

- **Google Maps Geocoding API:** Para convertir las ubicaciones de los reportes en coordenadas geográficas precisas.
- **Google Earth Engine API:** Para la obtención de imágenes satelitales de las zonas reportadas y la ejecución de análisis geoespaciales.
- **OpenCV o Scikit-Image:** Para realizar la segmentación de las imágenes satelitales y detectar patrones anómalos o cultivos ilícitos.

### 3.3. Procesamiento de Imágenes

- **TensorFlow o Keras:** Para construir y entrenar modelos de machine learning que automaticen la identificación de cultivos ilícitos en las imágenes satelitales.
- **GDAL (Geospatial Data Abstraction Library):** Para procesar grandes volúmenes de datos geoespaciales y realizar análisis raster.

## 4. Base de Datos

La base de datos de **Designious** almacenará los reportes, las coordenadas de los cultivos ilícitos, y los metadatos asociados a las imágenes satelitales.

### 4.1. NoSQL

- **MongoDB:** Almacenará la información de los reportes de forma flexible, permitiendo manejar grandes volúmenes de datos de forma no estructurada.

### 4.2. SQL (Opcional)

- **PostgreSQL con PostGIS:** Para almacenar datos geoespaciales de forma estructurada y ejecutar consultas geográficas avanzadas.

## 5. Almacenamiento de Imágenes

Para almacenar las imágenes satelitales procesadas y los informes generados, se utilizarán servicios en la nube:

- **Amazon S3 (AWS):** Ideal para almacenar imágenes satelitales y documentos de informes de manera eficiente.

- **Google Cloud Storage:** Alternativa que permite almacenar y servir imágenes y documentos directamente desde la infraestructura de Google.

## 6. Visualización de Imágenes Satelitales Segmentadas

Los usuarios podrán acceder a las imágenes segmentadas de cultivos ilícitos a través de la plataforma web o móvil de **Designious**:

- **Google Maps JavaScript API:** Para superponer las imágenes satelitales segmentadas en mapas interactivos, permitiendo la navegación y análisis de las áreas de interés.
- **Leaflet.js:** Alternativa open-source para la visualización de mapas interactivos y capas de imágenes.

## 7. Servicios de Nube y DevOps

**Designious** se beneficiará de la escalabilidad y flexibilidad que ofrecen las plataformas de nube, además de los servicios de DevOps para mantener la infraestructura.

### 7.1. Plataformas de Nube

- **AWS o Google Cloud Platform:** Para alojar el backend, la base de datos y los servicios de procesamiento de imágenes.
- **Docker:** Para contenerizar las aplicaciones y asegurar que el entorno de producción sea consistente y reproducible.
- **Kubernetes:** Para escalar de forma automática las instancias del backend según la carga de usuarios y tráfico de datos.

## 8. Monitoreo y Escalabilidad

Para garantizar que la plataforma de **Designious** funcione de forma óptima y sea capaz de manejar un crecimiento en usuarios y datos:

### 8.1. Monitoreo

- **AWS CloudWatch o Google Cloud Monitoring:** Para monitorear el uso de recursos, el tráfico en la plataforma y el rendimiento de los servidores.

### 8.2. Escalabilidad

- **Kubernetes:** Para ajustar automáticamente la capacidad de procesamiento según la demanda de los usuarios y el volumen de datos.