



**Proyecto Final Bases de Datos
Primera entrega**

Integrantes:

Jhonnier Andres Becerra Ballesteros – 201940979
Lina Vanessa Cosme Arce – 202436459

Docente:
Jefferson A. Peña Torres

Universidad del Valle
Santiago de Cali
Escuela de Ingeniería de Sistemas y Computación
12 de noviembre, 2025

Introducción

En el siguiente documento se encontrará la primera parte realizada para cumplir con los requerimientos del proyecto final para la materia de bases de datos.

El proyecto consta de realizar un sistema para el monitoreo de contaminantes, donde se pueda recopilar, procesar y visualizar datos provenientes de una red de estaciones distribuidas en la ciudad de Cali que miden la calidad del aire a través de sensores de manteniendo un registro histórico para el análisis preciso del estado ambiental de la ciudad.

Desarrollo del proyecto

Lo primero que se debe realizar es la identificación de los requerimientos con ayuda de los objetivos planteados en el documento del proyecto, primero, nos dicen que el sistema debe de ser confiable ya que se deben de tomar decisiones importantes con los datos que este va a almacenar y procesar, además, debe de garantizar que el sistema contribuye a tener un ambiente más saludable, fortalecer la participación ciudadana y mejorar la planificación de las políticas públicas en torno a la gestión ambiental.

Esta parte es muy importante, ya que, por más que se tenga que realizar un software para una función, hay que conocer su trasfondo y su impacto en la sociedad para poder tener una mejor idea de quien va a ser el usuario final.

Al ser un sistema complejo, se debe integrar con varias funciones. No será utilizado solo por una persona, se tendrán varias instituciones y cada una tendrá autonomía de aceptar o rechazar las estaciones de monitoreo que desee integrarse a la red, por ende debe de tener un sistema de autorización de credenciales para iniciar sesión para que el administrador de cada estación seleccione qué institución está asociada con qué punto, además, debe de tener la localización, el tipo de sensor instalado, la variable que mide, el responsable técnico de la estación, así como documentos o certificados de calibración y mantenimiento.

Una vez sea aceptada la solicitud de integración, el sistema se encarga de almacenar la información suministrada permitiendo su consulta. Los usuarios autorizados podrán acceder a reportes detallados que incluyen tendencias históricas de contaminantes, zonas críticas, proyecciones de riesgo y alertas tempranas frente a episodios de contaminación.

Entre los reportes que se almacenarán están:

- Reportes de Calidad del Aire y Estado Ambiental
- Reportes de Tendencias
- Reportes de Alertas Crítica

- Reportes de Infraestructura y Mantenimiento

Para realizar todo esto se utilizará el stack tecnológico recomendado por el docente encargado de la materia.

Backend:

- Framework: Se utilizará Python + Django, aprovechando su robustez, seguridad integrada y el ecosistema de django-rest-framework para la rápida construcción de una API RESTful eficiente.
- Seguridad: La autenticación y autorización se gestionarán mediante JSON Web Tokens (JWT). Adicionalmente, se implementarán las medidas de seguridad estándar de Django para la protección contra ataques comunes (SQL Injection, XSS).

Base de Datos:

- Gestor: Se empleara PostgreSQL como sistema gestor de base de datos relacional, dada su fiabilidad y escalabilidad para manejar grandes volúmenes de datos históricos.
- Datos Espaciales: Se activará la extensión PostGIS, un requisito indispensable para almacenar eficientemente las coordenadas de las estaciones y realizar consultas espaciales complejas.

Frontend (Web):

- Framework: La interfaz de usuario web será una Single-Page Application (SPA) desarrollada en ReactJS por su alta interactividad y ecosistema.
- Estilos: Se utilizará Tailwind CSS para garantizar un diseño responsive (accesible desde móviles y escritorio) y permitir un desarrollo ágil de interfaces modernas.

Frontend (Móvil):

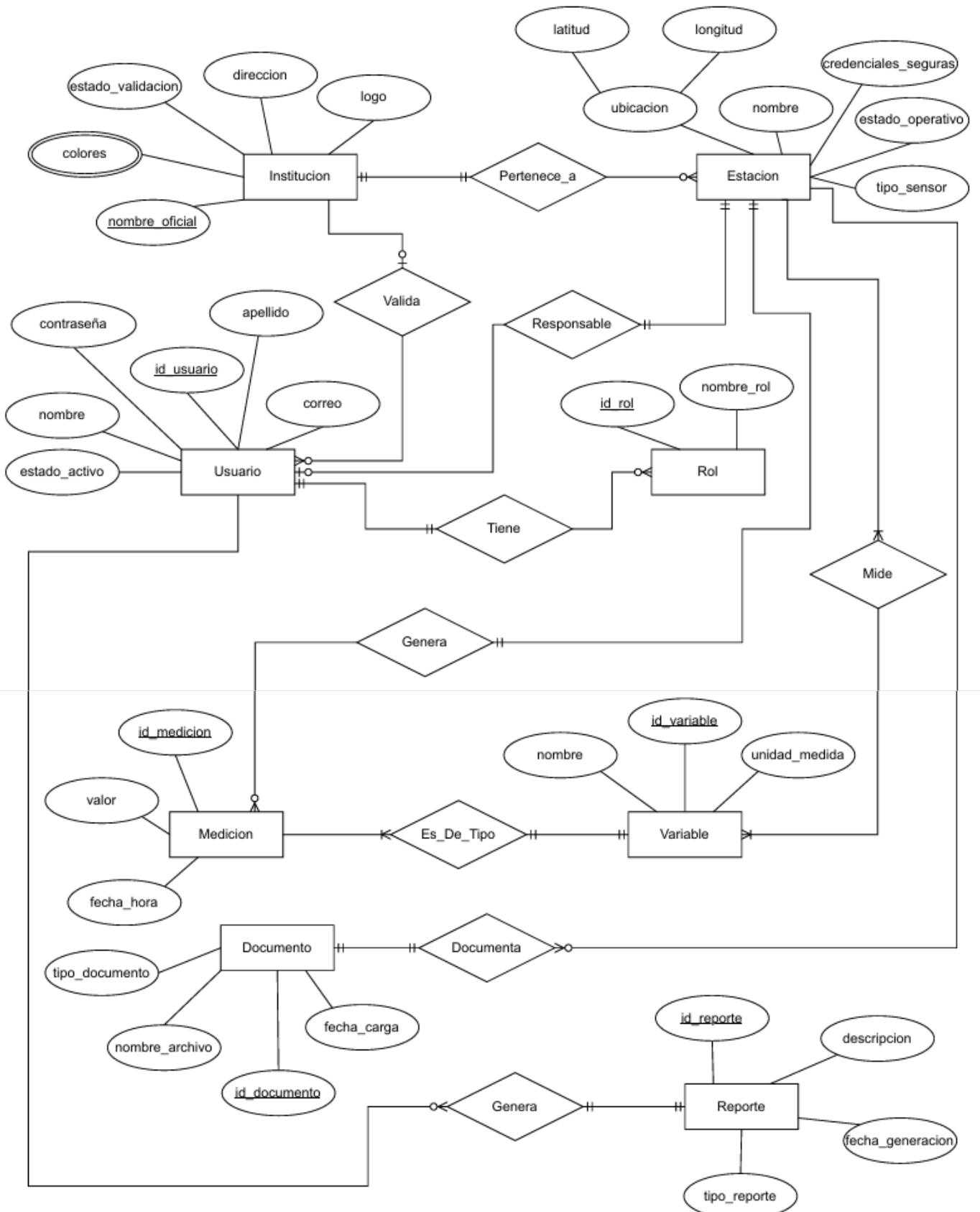
- Framework: La aplicación móvil se desarrollará usando Expo + React Native, permitiendo reutilizar la lógica de React y compilar de forma nativa para iOS y Android.
- Cliente HTTP: Se usará Axios para gestionar todas las comunicaciones basadas en promesas con la API RESTful del backend.

Despliegue (DevOps):

- Contenedores: Todo el proyecto (Frontend, Backend, Base de Datos) se empaquetan en contenedores aislados usando Docker.
- Orquestación: Se empleará Docker Compose para definir, gestionar y facilitar el despliegue coordinado de todos los servicios de la aplicación.

Ya con esto definido pasamos a los entregables solicitados para esta primera parte del proyecto:

1. Un (1) Diagrama (o modelo) Entidad-Relación (ER), que incluya las entidades, atributos y relaciones básicas.
 - Utilice la notación vista en clase (Chen + Cardinalidad crowfoot)
 - No se debe olvidar los factores de calidad



2. A partir del diagrama ER, presente un (1) esquema relacional.

- Establezca los tipos de datos
- Las restricciones de llave primaria y foránea
- Incluya el Diccionario de datos: Data elements

Una vez validado el "mapa" (el ER), este se utilizó para construir el Esquema Relacional. Este esquema ya no es un boceto, sino el "plano técnico" detallado de la base de datos.

En esta fase se convirtieron las ideas conceptuales en una estructura de tablas real. Se definieron los nombres exactos de las columnas, el tipo de dato específico para cada una (texto, número, fecha) y, de forma crucial, cómo se conectarán las tablas mediante Llaves Primarias y Foráneas.

Este paso es el que garantiza la coherencia e integridad de los datos, evitando, por ejemplo, que se puedan registrar mediciones de una estación que no existe.

Institucion		
PK	<u>id_institucion</u>	INT
	nombre_oficial	VARCHAR
	logo	VARCHAR
	direccion	VARCHAR
	color	VARCHAR
	estado_validacion	BOOLEAN

Rol		
PK	<u>id_rol</u>	INT
	nombre_rol	VARCHAR

Usuario_Rol		
PK FK	<u>id_usuario</u>	INT
PK FK	<u>id_rol</u>	INT

Variable		
PK	<u>id_variable</u>	INT
	nombre	VARCHAR
	unidad_medida	VARCHAR

Usuario		
PK	<u>id_usuario</u>	INT
	nombre	VARCHAR
	apellido	VARCHAR
	correo	VARCHAR
	contraseña	VARCHAR
	estado_activo	BOOLEAN
FK	<u>id_institucion</u>	INT

Estacion		
PK	<u>id_estacion</u>	<u>INT</u>
	nombre	VARCHAR
	ubicacion	VARCHAR
	latitud	DECIMAL
	longitud	DECIMAL
	tipo_sensor	VARCHAR
	estado_operativo	VARCHAR
	credenciales_seguras	TEXT
FK	<u>id_institucion</u>	<u>INT</u>
FK	<u>id_usuario_responsable</u>	<u>INT</u>

Medicion		
PK	<u>id_medicion</u>	<u>INT</u>
	valor	DECIMAL
	fecha_hora	TIMESTAMP
FK	<u>id_estacion</u>	<u>INT</u>
FK	<u>id_variable</u>	<u>INT</u>
FK	<u>id_usuario_generador</u>	<u>INT</u>

Documento		
PK	<u>id_documento</u>	<u>INT</u>
	nombre_archivo	VARCHAR
	tipo_documento	VARCHAR
	fecha_carga	TIMESTAMP
FK	<u>id_medicion</u>	<u>INT</u>

Reporte		
PK	<u>id_reporte</u>	<u>INT</u>
	tipo_reporte	VARCHAR
	fecha_generacion	TIMESTAMP
	descripcion	TEXT
FK	<u>id_documento</u>	<u>INT</u>

Con el diagrama ER terminado, se elaboró el Diccionario de Datos. Si el esquema es el plano, el diccionario es el "manual de usuario" o la leyenda que explica ese plano.

El esquema indica que existe un campo estado operativo, pero el diccionario explica su significado (ej. "Aquí se guarda si la estación está 'Activa', 'En Mantenimiento', etc.").

Este documento se creó para que todo el equipo (presente y futuro) sepa exactamente para qué sirve cada campo y qué información debe contener. Funciona como una guía de referencia esencial para no perder la consistencia y el orden de los datos a largo plazo.

Nombre de Archivo:	Institucion	Descripción	Almacena la información de las instituciones que se registran y validan en la plataforma VriSA
Campo	Tamaño	Tipo de Dato	Descripción
id_institucion	11	Numérico (INT)	Identificador único de la institución.
nombre_oficial	255	Caracter (VARCHAR)	Nombre legal o completo de la institución.
logo	255	Caracter (VARCHAR)	URL o ruta al archivo de imagen del logo.
direccion	255	Caracter (VARCHAR)	Dirección física de la institución.
colores	100	Caracter (VARCHAR)	Colores corporativos (ej. "#FFFFFF").
estado_validacion	1	Booleano	Indica si la institución ha sido validada (0=No, 1=Si).
Campos Clave: id_institucion		(PK)Relaciones: Ninguna.	

Nombre de Archivo:	Rol	Descripción	Tabla catálogo que define los roles de usuario en el sistema.
Campo	Tamaño	Tipo de Dato	Descripción
id_rol	11	Numérico (INT)	Identificador único del rol.
nombre_rol	100	Caracter (VARCHAR)	Nombre del rol (ej. "Admin", "Investigador", "Técnico").
Campos Clave: id_rol		(PK)Relaciones: Ninguna.	

Nombre de Archivo:	Usuario	Descripción	Almacena los datos de los usuarios que acceden a la plataforma.
Campo	Tamaño	Tipo de Dato	Descripción
id_usuario	11	Numérico (INT)	Identificador único del usuario.
nombre	100	Caracter (VARCHAR)	Nombre de pila del usuario.
apellido	100	Caracter (VARCHAR)	Apellido del usuario.
correo	255	Caracter (VARCHAR)	Correo electrónico del usuario, usado para login (Único).
contraseña	255	Caracter (VARCHAR)	Contraseña del usuario (debe almacenarse hasheada).
estado_activo	1	Booleano	Indica si la cuenta del usuario está activa (0=No, 1=Si).
id_institucion	11	Numérico (INT)	Institución que valida o a la que pertenece el usuario.
Campos Clave: id_usuario		(PK)Relaciones: id_institucion con Institucion(id_institucion)	

Nombre de Archivo:	Usuario_Rol	Descripción	intermedia (relación M:N) que asigna uno o más roles a cada usuario.
Campo	Tamaño	Tipo de Dato	Descripción
id_usuario	11	Numérico (INT)	Identificador del usuario (FK).
id_rol	11	Numérico (INT)	Identificador del rol (FK).
Campos Clave: id_usuario, id_rol		(PK)Relaciones: id_usuario con Usuario(id_usuario), id_rol con Rol(id_rol)	

Nombre de Archivo:	Estacion	Descripción	Contiene la información de cada estación de monitoreo registrada.
Campo	Tamaño	Tipo de Dato	Descripción
id_estacion	11	Numérico (INT)	Identificador único de la estación.
nombre	100	Caracter (VARCHAR)	Nombre descriptivo de la estación.
ubicacion	255	Caracter (VARCHAR)	Descripción textual de la ubicación (ej. Comuna 5).
latitud	9,6	Numérico (DECIMAL)	Coordenada geográfica de latitud.
longitud	9,6	Numérico (DECIMAL)	Coordenada geográfica de longitud.
tipo_sensor	100	Caracter (VARCHAR)	Describe el tipo de sensor principal instalado.
estado_operativo	50	Caracter (VARCHAR)	Estado actual de la estación (ej. "Activa", "Mantenimiento").
credenciales_seguras	N/A	Caracter (TEXT)	Token o API Key para la autenticación del hardware.
id_institucion	11	Numérico (INT)	Institución a la que pertenece la estación.
id_usuario_responsable	11	Numérico (INT)	Usuario técnico responsable de la estación.
Campos Clave: id_estacion		(PK)Relaciones:id_institucion con Institucion(id_institucion), id_usuario_responsable con Usuario(id_usuario)	

Nombre de Archivo:	Variable	Descripción	Tabla catálogo de las variables meteorológicas y contaminantes que se pueden medir.
Campo	Tamaño	Tipo de Dato	Descripción
id_variable	11	Numérico (INT)	Identificador único de la variable.
nombre	100	Caracter (VARCHAR)	Nombre de la variable (ej. "Temperatura", "PM2.5").
unidad_medida	50	Caracter (VARCHAR)	Unidad en que se mide la variable (ej. "°C", "µg/m³").
Campos Clave: id_variable		(PK)Relaciones: Ninguna.	

Nombre de Archivo:	Medicion	Descripción	Almacena cada dato individual transmitido por las estaciones. Es la tabla más grande
Campo	Tamaño	Tipo de Dato	Descripción
id_medicion	11	Numérico (INT)	Identificador único de la medición.
valor	10,4	Numérico (DECIMAL)	Valor numérico registrado en la medición.
fecha_hora	N/A	Fecha/Hora (TIMESTAMP)	Fecha y hora exactas en que se tomó la medición.
id_estacion	11	Numérico (INT)	Estación que tomó la medida.
id_variable	11	Numérico (INT)	Variable que se midió (ej. "PM2.5").
id_usuario_generador	11	Numérico (INT)	Usuario que generó la medición (si fue manual).
Campos Clave: id_medicion	(PK)Relaciones: id_estacion con Estacion(id_estacion), id_variable con Variable(id_variable), id_usuario_generador con Usuario(id_usuario)		

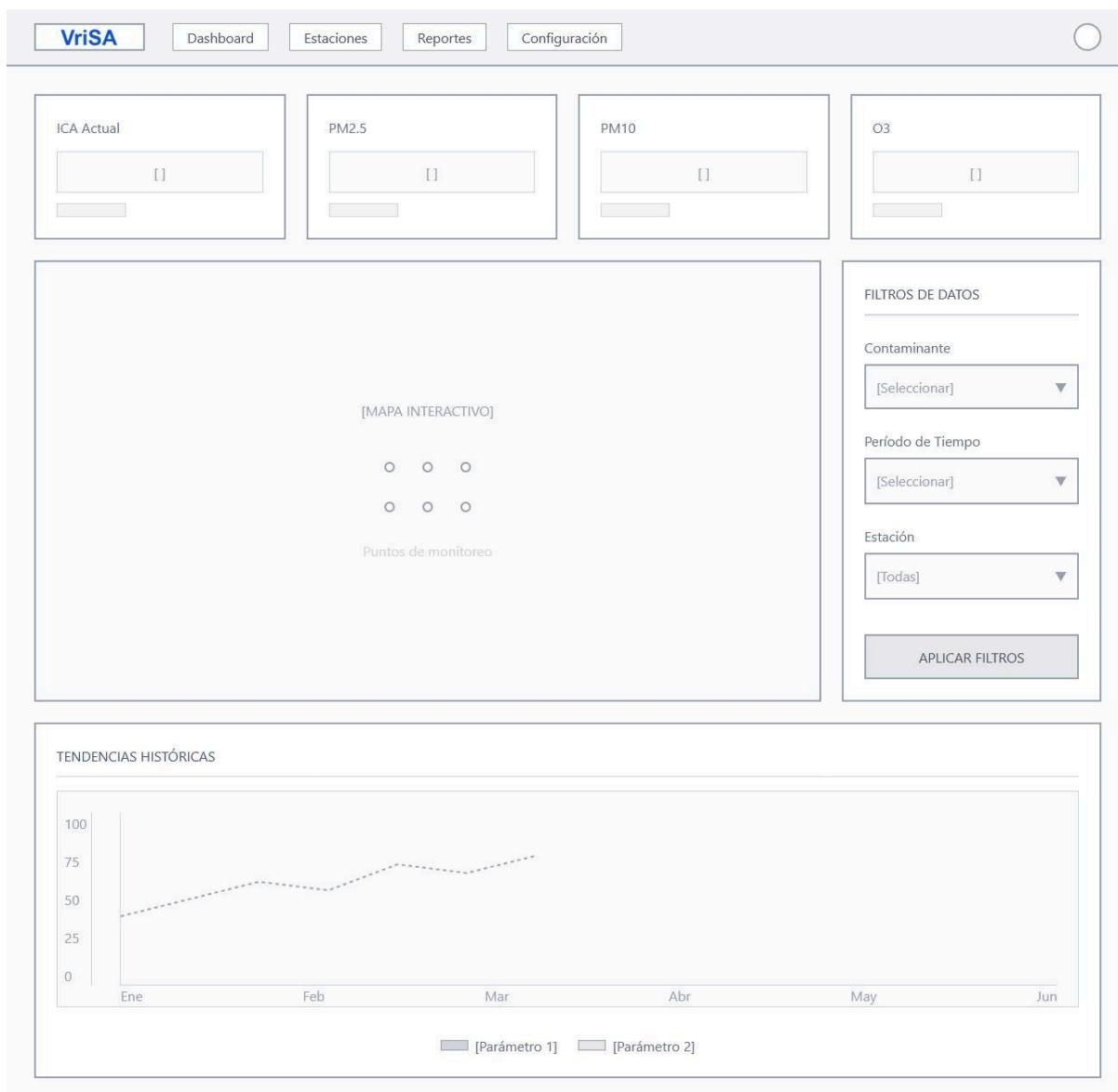
Nombre de Archivo:	Reporte	Descripción:	Contiene los reportes generados por el sistema (alertas, tendencias, etc.).
Campo	Tamaño	Tipo de Dato	Descripción
id_reporte	11	Numérico (INT)	Identificador único del reporte.
tipo_reporte	100	Caracter (VARCHAR)	Tipo de reporte (ej. "Alerta Crítica", "Tendencia Semanal").
fecha_generacion	N/A	Fecha/Hora (TIMESTAMP)	Fecha y hora en que se generó el reporte.
descripcion	N/A	Caracter (TEXT)	Contenido o descripción detallada del reporte.
id_documento	11	Numérico (INT)	Documento que genera o respalda este reporte.
Campos Clave: id_reporte	(PK)Relaciones: id_documento con Documento(id_documento)		

3. Un (1) archivo con los diseños de la Interfaz de Usuario (UI): Presentar bocetos o prototipos de la interfaz de usuario para al menos algunas de las pantallas principales de la aplicación.

- Estos pueden ser diseños de baja fidelidad. (Hecho en Figma).

El diseño de la interfaz de usuario no es un elemento puramente estético, sino la traducción funcional de los requerimientos del negocio y del modelo de datos. Los siguientes prototipos de baja fidelidad (wireframes) se diseñaron para validar los flujos de trabajo clave de la plataforma VriSA y asegurar que la arquitectura de datos propuesta pueda satisfacer las necesidades de los distintos perfiles de usuario.

Dashboard Monitoreo Ambiental



Gestión Institucional

Solicitudes de Registro de Instituciones			
Nombre Institución	Estado	Fecha Solicitud	Acciones
Universidad Nacional del Norte	Pendiente	2025-10-28	<input checked="" type="checkbox"/> Aceptar <input type="checkbox"/> Rechazar
Instituto Tecnológico Central	Pendiente	2025-10-29	<input checked="" type="checkbox"/> Aceptar <input type="checkbox"/> Rechazar
Colegio San Martín	Pendiente	2025-10-30	<input checked="" type="checkbox"/> Aceptar <input type="checkbox"/> Rechazar
Academia de Ciencias Aplicadas	Pendiente	2025-11-01	<input checked="" type="checkbox"/> Aceptar <input type="checkbox"/> Rechazar

Registro Estación

Registro de Nueva Estación

Nombre de la Estación

Tipo de Sensor

Latitud

Longitud

VARIABLES MEDIDAS

- PM2.5
- Temperatura
- SO2

Responsable Técnico

Generar Credenciales Seguras

DOCUMENTACIÓN

Cargar Certificado de Calibración

Archivos Cargados:

(Ningún archivo cargado)

Cancelar

Registrar

4. Un (1) diagrama de la arquitectura de Alto Nivel: Presente un gráfico de la arquitectura de alto nivel de la aplicación, incluyendo las tecnologías y lenguajes de programación que se planea utilizar.

- Considere un diagrama de bloques o cajas para describir los módulos, componentes y/o servicios de la plataforma

VriSA

1. Clientes (Usuarios)



Admin / Instituciones
(Navegador Web)



Investigadores / Autoridades
(Navegador Web)



Ciudadanía
(iOS / Android)



(Solicitan Vistas / Interactúan)

2. Aplicaciones Frontend

App Web (Admin)

- ReactJS
- HTML5/CSS3
- Autenticación

App Web (Consulta)

- ReactJS
- Bootstrap/Tailwind
- Dashboards / Mapas

App Móvil

- Expo / React Native
 - Axios (para API)
 - Alertas / Reportes

↓
(Peticiones HTTP / Solicitan Datos)

3. API RESTful (Punto de Entrada)

Protocolo de comunicación: **HTTPS / JSON**

↓
(Enruta Petición)

4. Entorno de Despliegue (DOCKER)

5. Servidor Backend (Contenedor 1)

Tecnología: **Python + Django**

- API REST (Django REST Framework)
- Lógica de Negocio (Validaciones, Roles, Alertas)
- Seguridad y Autenticación (JWT)
- Generación de Reportes
- Endpoint de Ingesta de Datos (de Sensores)

(Lee/Escribe Datos)

6. Base de Datos Relacional (Contenedor 2)

Tecnología: PostgreSQL

- Extensión PostGIS (Para datos espaciales)
- Datos de Usuarios, Instituciones, Estaciones
- Datos de Mediciones (Históricos y Tiempo Real)
- Reportes, Documentos, Roles

(Transmisión de datos ← Tiempo Real)

7. Fuentes de Datos Externas

Estaciones de Monitoreo (IoT)

- Sensores: PM2.5, \$SO_2\$, Temp, etc.
- Envía datos vía HTTPS