

SRS

8C

Equipo 3:
Quintana Zepeda Andrea Guadalupe
Martinez Zavala Jonathan
Garcia Lopez Miguel Isaac
Ochoa Huerta Cesia Nuemi
Ponce Duarte Jose de Jesus
Avalos Garcia Juan Antonio

Índice

Introducción	2
Propósito	3
Alcance	3
Personal Involucrado	3
Resumen	4
Descripción general	4
Requerimientos Funcionales.....	5
Requerimientos No Funcionales	11
Restricciones.....	13
Requisitos de rendimiento	14
Seguridad	14
Fiabilidad	16
Disponibilidad	16
Mantenibilidad	17
Especificación de los Requisitos	18
Portabilidad	18
Otros requisitos	19
Requisitos Legales.....	19
Aprobaciones	20
Apéndices	21
Avance de Backend	29
Avance de Landing page	33
Avance de Frontend	37
Aprobaciones de avance	38

Introducción

El presente documento describe las especificaciones de requerimientos del software para el desarrollo de una plataforma destinada al monitoreo y gestión del consumo de agua en entornos industriales. Esta iniciativa surge como respuesta a la necesidad crítica del sector industrial de contar con herramientas tecnológicas que permitan supervisar, optimizar y reducir el uso de recursos hídricos, promoviendo así prácticas sostenibles y cumpliendo con normativas ambientales vigentes.

El sistema propuesto, tentativamente denominado “**Droply**”, está orientado a proporcionar una solución centralizada que facilite la visualización en tiempo real del consumo de agua por área o proceso, la generación de alertas, bitácoras automáticas, control de usuarios con distintos niveles de acceso, así como la configuración personalizada de la gestión operativa. Todo ello permitirá a las empresas tomar decisiones informadas para mejorar su eficiencia hídrica, reducir costos operativos y alinear sus procesos con estándares como la norma ISO 14001.

Este documento servirá como guía base para el desarrollo, implementación y validación del sistema, asegurando que se cumplan tanto los objetivos funcionales como los no funcionales definidos por el equipo de desarrollo.

Propósito

Desarrollar un sistema que permite a las empresas del sector industrial medir, monitorear y optimizar el consumo de agua por área o proceso productivo, permitiendo identificar desperdicios, implementar estrategias de uso eficiente, y cumplir con normativas ambientales.

Alcance

El sistema de software a desarrollar se denomina “Plataforma para el Monitoreo y Gestión del Consumo de Agua en Entornos Industriales” (Por definirse “Droply”)

El sistema de software a desarrollar se enfocará en ofrecer una plataforma centralizada que permita a las industrias:

- Gestión industrial personalizada.
- Reducción de costos operativos asociados al consumo de agua.
- Cumplimiento con normativas ambientales (ISO 14001).
- Visualización de datos.
- Configuración automática de alertas.
- Generación de bitácoras.
- Control de usuarios y roles.

Personal Involucrado

Integrante	Responsabilidad
Martinez Zavala Jonathan	Desarrollador Fullstack, QA
Quintana Zepeda Andrea Guadalupe	Desarrollador Frontend, UI/UX, QA
Avalos García Juan Antonio	Desarrollador Frontend, UI/UX
Ponce Duarte José De Jesús	Desarrollador Backend, DBA
García López Miguel Isaac	Desarrollador Backend, DevOps

Resumen

El sistema de software que se desea desarrollar tiene como propósito principal apoyar al sector industrial en la medición, monitoreo y optimización del consumo de agua, alineándose con los principios de sostenibilidad y responsabilidad ambiental. Esta iniciativa responde directamente a los Objetivos de Desarrollo Sostenible (ODS) definidos por las Naciones Unidas, específicamente al ODS N.º 6: “Agua limpia y saneamiento”, cuyo objetivo es garantizar la disponibilidad de agua y su gestión sostenible.

Descripción general

El sistema de software propuesto permitirá a los usuarios, principalmente personal técnico y administrativo de empresas del sector industrial, solicitar, visualizar y gestionar de manera eficiente el consumo de agua en sus procesos operativos. Este sistema busca optimizar la gestión hídrica mediante la automatización de la recolección de datos, el análisis en tiempo real y la generación de reportes clave para la toma de decisiones y cumplimiento de normativas ambientales.

Funcionalidades Principales:

- Registro de usuarios.
- Registro de áreas industriales.
- Configuración de medidores digitales.
- Generación de alertas personalizadas.
- Generación de bitácoras.
- Carga de datos (archivos).

Requerimientos Funcionales

1. Gestión de Usuarios y Roles

RF-01: Registro de usuarios

- **Descripción:** El sistema debe permitir que cualquier usuario cree una cuenta. Al registrarse, el usuario podrá definir su propio rol o conjunto de permisos, sin depender de un administrador.
- **Prioridad:** Alta
- **Entradas:** Nombre, correo electrónico, contraseña, nombre del rol (opcional), configuración de permisos (opcional).
- **Salidas:** Cuenta activa en la base de datos con rol y permisos definidos por el propio usuario.

RF-02: Autenticación y control de acceso

- **Descripción:** El sistema debe permitir el inicio de sesión mediante credenciales. Después de iniciar sesión, el usuario accederá a su propio entorno, configurado según los permisos que él mismo definió. El sistema debe permitir que el usuario cree, modifique o elimine sus propios roles personalizados y defina qué acciones puede realizar cada uno.
- **Prioridad:** Alta
- **Entradas:** Correo electrónico, contraseña.
- **Salidas:** Acceso al sistema; visualización y gestión de dashboards, registros, y configuraciones de roles/permisos según lo que el usuario haya definido para sí mismo o para otros.

2. Monitoreo de Consumo

RF-03: Integración con medidores digitales

- **Descripción:** Conexión con sensores vía API o IoT para capturar datos de consumo en tiempo real.
- **Prioridad:** Crítica
- **Protocolos soportados:** MQTT, HTTP.

RF-04: Visualización de datos

- **Descripción:** Gráficos históricos y en tiempo real por área/proceso (mínimo: líneas, barras y tablas).
- **Prioridad:** Media

3. Alertas y Notificaciones

RF-05: Configuración de umbrales

- **Descripción:** Permitir definir límites de consumo (ej: 1000L/hora) para generar alertas.
- **Prioridad:** Alta

RF-06: Notificaciones multicanal

- **Descripción:** Enviar alertas cuando se supere un umbral.
- **Prioridad:** Media

4. Reportes y Cumplimiento

RF-07: Generación de bitácoras automáticas

- **Descripción:** Registro diario de consumos por área en formato CSV/PDF.
- **Prioridad:** Baja

RF-08: Reportes para normativas (ISO 14001)

- **Descripción:** Plantillas predefinidas con métricas de sostenibilidad.
- **Prioridad:** Media

5. Gestión de Áreas

RF-09: Asociación de medidores a áreas

- **Descripción:** Cada área podrá asociarse a uno o varios medidores de consumo para segmentar la información.
- **Prioridad:** Media

6. Gestión de Datos e Importación

RF-10: Carga manual de datos históricos

- **Descripción:** El usuario podrá subir archivos CSV o Excel con datos previos de consumo para integrarlos al sistema.
- **Prioridad:** Media

RF-11: Validación automática de archivos

- **Descripción:** El sistema debe validar el formato de los archivos cargados y notificar errores si hay inconsistencias.

- **Prioridad:** Media

7. Panel de Administración y Visualización

RF-12: Panel de control personalizado

- **Descripción:** Los usuarios podrán personalizar sus dashboards para mostrar las métricas que más les interesen.
- **Prioridad:** Media

RF-13: Filtros de visualización

- **Descripción:** Permitir filtros por fecha, área, consumo específico, límites, entre otros.
- **Prioridad:** Alta

8. Seguridad y Auditoría

RF-14: Registro de actividad del usuario (logs)

- **Descripción:** El sistema debe mantener un historial de acciones realizadas por los usuarios (ingresos, ediciones, eliminaciones, etc.).
- **Prioridad:** Alta

9. Gestión de Costos y Tarifas

RF-15: Configuración de tarifas de agua

- **Descripción:** El sistema debe permitir definir tarifas escalonadas de agua (por m³) asociadas a rangos de consumo, para calcular costos operativos.
- **Prioridad:** Media
- **Entradas:** Rango mínimo/máximo (m³), costo por m³, fecha de vigencia.
- **Salidas:** Registro de tarifas aplicables a reportes financieros.

RF-16: Cálculo automático de costos

- **Descripción:** Generar proyecciones de gastos por área/proceso basadas en consumo real y tarifas vigentes.
- **Prioridad:** Baja

10. Mantenimiento de Medidores

RF-17: Registro de mantenimiento preventivo/correctivo

- **Descripción:** Programar y registrar mantenimientos de medidores (fechas, tipo, proveedor).
- **Prioridad:** Media
- **Entradas:** Tipo de mantenimiento, fecha programada, proveedor, costo.
- **Salidas:** Historial de mantenimientos por medidor.

RF-18: Alertas de mantenimiento pendiente

- **Descripción:** Notificar cuando un medidor requiera mantenimiento según su vida útil o fallas reportadas.
- **Prioridad:** Baja

11. Gestión de Inventario

RF-19: Control de inventario de medidores

- **Descripción:** Registrar datos de compra, garantía y ubicación en bodega de medidores.
- **Prioridad:** Baja
- **Entradas:** Fecha de compra, vida útil (meses), ubicación.
- **Salidas:** Reporte de medidores activos/en bodega/descartados.

12. Metas de Sostenibilidad

RF-20: Configuración de metas de consumo

- **Descripción:** Definir objetivos de reducción de agua por área/proceso (m³ y plazo).
- **Prioridad:** Media
- **Entradas:** Consumo objetivo (m³), fecha límite, área asociada.
- **Salidas:** Comparativo consumo real vs. meta en dashboards.

13. Cumplimiento Normativo *(Nueva Sección)*

RF-21: Registro de normativas aplicables

- **Descripción:** Almacenar estándares legales (ej: ISO 14001) y vincularlos a reportes automáticos.
- **Prioridad:** Baja
- **Entradas:** Nombre de normativa, descripción, fecha actualización.
- **Salidas:** Alertas de incumplimiento en reportes.

Requerimientos No Funcionales

1. Rendimiento y Escalabilidad

RNF-01: Tiempo de respuesta

- El sistema debe responder a solicitudes del usuario en menos de 2 segundos bajo condiciones normales de carga.

RNF-02: Escalabilidad

- La arquitectura debe ser escalable horizontal y verticalmente para adaptarse al crecimiento del número de usuarios y sensores conectados.

2. Seguridad

RNF-03: Encriptación de datos

- Toda la información sensible (contraseñas, datos de consumo) debe almacenarse y transmitirse cifrada (ej: HTTPS, AES).

RNF-04: Autenticación segura

- El sistema debe implementar mecanismos de autenticación como JWT y políticas de contraseña segura.

3. Usabilidad

RNF-05: Interfaz amigable

- La plataforma debe ser intuitiva, responsiva y accesible para usuarios con distintos niveles de experiencia.

RNF-06: Soporte multilinguaje

- La interfaz deberá estar disponible en al menos dos idiomas: español e inglés.

4. Compatibilidad

RNF-07: Navegadores soportados

- La aplicación debe funcionar correctamente en los navegadores modernos: Chrome, Firefox, Edge y Safari.

RNF-08: Dispositivos compatibles

- La plataforma debe ser responsive, funcional tanto en escritorio como en dispositivos móviles y tablets.

5. Disponibilidad y Confiabilidad

RNF-09: Disponibilidad del sistema

- El sistema debe estar disponible al menos el 99.5% del tiempo mensual.

RNF-10: Recuperación ante fallos

- Debe contar con mecanismos de respaldo automático y recuperación de datos en caso de caída del sistema.

6. Mantenibilidad y Actualización

RNF-11: Documentación técnica

- El código debe ir acompañado de documentación técnica que facilite su mantenimiento.

RNF-12: Gestión de versiones

- Las actualizaciones deben manejarse a través de un sistema de control de versiones con despliegue continuo (ej: Git + CI/CD).

Restricciones

Resumen de limitaciones y consideraciones del sistema que se requiere desarrollar con todas las funcionalidades mencionadas previamente. Estas restricciones deben ser consideradas para asegurar la viabilidad, estabilidad y cumplimiento normativo del sistema.

1. Requiere dispositivos de medición.

El sistema depende de sensores físicos para recolectar datos de consumo de agua de manera automática.

2. Conexión a internet.

La funcionalidad en tiempo real depende de una conexión estable a la red.

3. Cumplimiento legal.

El sistema debe ajustarse a normativas ambientales locales e internacionales.

4. Compatibilidad limitada.

Solo se garantiza funcionamiento en navegadores modernos y actualizados.

5. Acceso restringido.

Solo usuarios autorizados, según su rol, podrán acceder a funciones específicas.

6. Mantenimiento

Se realizarán pausas planificadas para mantenimiento y actualización.

Requisitos de rendimiento

1. Tiempo de respuesta

- El 95% de las búsquedas de equipos deben completarse en menos de 2 segundos.
- El tiempo promedio para procesar una solicitud de préstamo no debe superar los 5 segundos.

2. Capacidad

- El sistema debe soportar hasta 500 usuarios concurrentes durante las horas pico.
- El sistema debe poder procesar un promedio de 100 transacciones por minuto.

3. Disponibilidad

- El sistema debe tener una disponibilidad del 99.9% (es decir, un tiempo de inactividad máximo de 8.76 horas al año).

Seguridad

1. Autenticación y Autorización

- **Autenticación fuerte:** Implementar mecanismos de autenticación robustos, como contraseñas seguras con políticas de complejidad y autenticación de dos factores.
- **Autorización basada en roles:** El usuario asigna permisos específicos a cada rol de usuario para limitar el acceso a datos y funcionalidades.
- **Gestión de sesiones:** Establecer tiempos de espera para las sesiones y requerir autenticación periódica.

2. Control de Acceso

- **Listas de control de acceso (ACL):** Definir ACLs precisas para cada recurso del sistema, permitiendo solo las acciones autorizadas.
- **Principio de mínimo privilegio:** Otorgar a cada usuario solo los permisos necesarios para realizar sus tareas. (NO SE BIEN JONA)

3. Cifrado

- **Cifrado de datos en reposo:** Proteger los datos almacenados en la base de datos mediante el cifrado.
- **Cifrado de datos en tránsito:** Utilizar protocolos seguros como HTTPS para proteger la comunicación entre el cliente y el servidor.
- **Gestión de claves:** Implementar un sistema seguro para la generación, almacenamiento y rotación de claves criptográficas.

4. Integridad de Datos

- **Sumas de comprobación:** Calcular sumas de comprobación para verificar la integridad de los archivos.
- **Firma digital:** Utilizar firmas digitales para garantizar la autenticidad y la integridad de los mensajes.
- **Registro de auditoría:** Mantener un registro detallado de todas las acciones realizadas en el sistema para detectar posibles intrusiones o modificaciones no autorizadas.

5. Protección contra Ataques

- **WAF (Web Application Firewall):** Implementar un WAF para proteger el sistema contra ataques web comunes como inyección SQL, XSS y CSRF.
- **Protección contra DDoS:** Implementar medidas para mitigar ataques de denegación de servicio distribuido.

01/0/2025

Desarrollo Web Profesional

Equipo 3

8C

- **Seguimiento de amenazas:** Mantenerse actualizado sobre las últimas amenazas y vulnerabilidades, aplicando parches y actualizaciones de seguridad de forma oportuna.

6. Gestión de Vulnerabilidades

- **Análisis de vulnerabilidades:** Realizar análisis de vulnerabilidades de forma regular para identificar y corregir cualquier debilidad en el sistema.

7. Copias de Seguridad y Recuperación

- **Copias de seguridad regulares:** Realizar copias de seguridad de los datos del sistema de forma regular y almacenarlas en un lugar seguro.
- **Planes de recuperación de desastres:** Desarrollar planes detallados para restaurar el sistema en caso de una falla o desastre.

Fiabilidad

- **Funcionamiento continuo:** El sistema debe operar sin interrupciones prolongadas.
- **Recuperación rápida:** En caso de falla, el sistema debe poder recuperarse rápidamente y reanudar sus operaciones.
- **Precisión de los datos:** Los datos almacenados en el sistema deben ser precisos y consistentes.

Disponibilidad

- **99.9% de disponibilidad:** Significa que el sistema debe estar disponible el 99.9% del tiempo, lo que equivale a un máximo de 8.76 horas de inactividad al año.

Mantenibilidad

1. Tipos de Mantenimiento

- **Correctivo:** Corregir errores o fallos en el sistema.
- **Adaptativo:** Modificar el sistema para adaptarse a cambios en el entorno o en los requisitos del negocio.
- **Perfectivo:** Mejorar el rendimiento, la fiabilidad o la usabilidad del sistema.
- **Preventivo:** Realizar tareas de mantenimiento de forma regular para prevenir fallos y prolongar la vida útil del sistema.

2. ¿Quién Realiza el Mantenimiento?

- **Usuarios:** Pueden realizar tareas de mantenimiento básicas, como cambiar contraseñas o actualizar sus perfiles.
- **Administradores del sistema:** Se encargan de tareas más complejas, como configurar el sistema, realizar copias de seguridad y resolver problemas técnicos.
- **Desarrolladores:** Realizan modificaciones al código fuente del sistema para corregir errores, añadir nuevas funcionalidades o mejorar el rendimiento.

3. ¿Cuándo Realizar el Mantenimiento?

La frecuencia del mantenimiento depende de diversos factores, como el tamaño y la complejidad del sistema, el número de usuarios y la criticidad de las funciones. Algunos ejemplos de tareas de mantenimiento y su frecuencia:

- **Generación de estadísticas:** Semanal, mensual o anualmente.
- **Realización de copias de seguridad:** Diaria, semanal o mensualmente.
- **Aplicación de parches de seguridad:** Tan pronto como estén disponibles.
- **Actualización de software:** Anualmente o según sea necesario.

- **Revisión del rendimiento del sistema:** Mensual o trimestralmente.

Especificación de los Requisitos

- **Documentación:** El sistema debe estar bien documentado, incluyendo diagramas de flujo, código fuente comentado y manuales de usuario.
- **Modularidad:** El sistema debe estar dividido en módulos bien definidos y desacoplados para facilitar los cambios.
- **Estándares de codificación:** Se deben seguir estándares de codificación consistentes para mejorar la legibilidad y mantenibilidad del código.
- **Herramientas de desarrollo:** Se deben utilizar herramientas de desarrollo adecuadas para facilitar la gestión del código, la depuración y las pruebas.
- **Acceso a la información:** Se debe facilitar el acceso a la información necesaria para realizar tareas de mantenimiento, como la configuración del sistema y los logs de errores.

Portabilidad

1. independencia de la Plataforma

- **Lenguajes de Programación:** Utilizar lenguajes de programación con amplia compatibilidad entre plataformas, como Javascript, Python o C#.
- **Frameworks:** Emplear frameworks que sean multiplataforma y ofrezcan abstracciones de las características específicas de cada sistema operativo.
- **Bases de Datos:** Utilizar bases de datos relacionales con drivers disponibles para múltiples sistemas operativos, como PostgreSQL, MySQL o MongoDB.

2. Minimización de Dependencias

- **Librerías de Terceros:** Evitar o minimizar el uso de librerías específicas de una plataforma. Si es necesario, utilizar librerías multiplataforma.

- **Interfaces de Programación de Aplicaciones (APIs):** Emplear APIs estándar y bien documentadas para interactuar con el sistema operativo y otros componentes del sistema.

3. Modularidad

- **Arquitectura Modular:** Diseñar el sistema en módulos bien definidos y desacoplados, facilitando la sustitución de componentes específicos de una plataforma.

4. Configuración Externa

- **Archivos de Configuración:** Utilizar archivos de configuración externos para almacenar parámetros específicos de cada entorno, evitando la necesidad de modificar el código fuente.

Otros requisitos

- **Usabilidad:** La interfaz de usuario debe ser intuitiva y acorde con las costumbres y expectativas culturales de los usuarios.
- **Normas Sociales:** El sistema debe respetar las normas sociales y éticas de la comunidad en la que se utiliza.

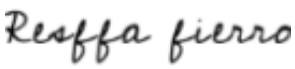

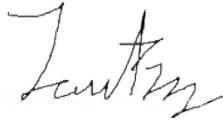

Requisitos Legales

- **Privacidad de Datos:** El sistema debe cumplir con las leyes de protección de datos y garantizar la confidencialidad de la información personal de los usuarios.
- **Propiedad Intelectual:** El sistema debe respetar los derechos de autor y las licencias de software.
- **Normativas Sectoriales:** El sistema debe cumplir con las normativas específicas del sector en el que se utiliza.

01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

Aprobaciones

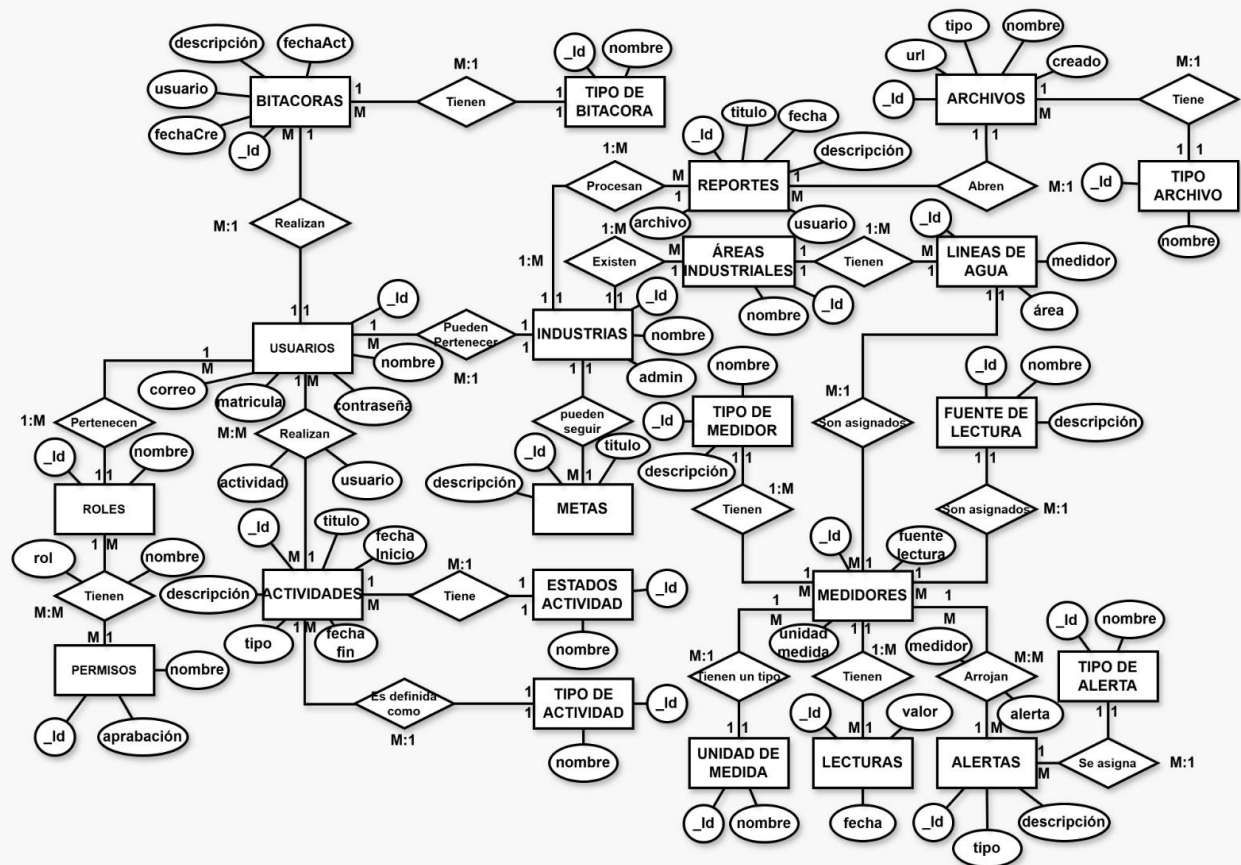
Nombre	Cargo	Firma
Dr. Ray Brunett Parra Galaviz	Desarrollo Web Profesional	
M.C Resffa Elizabeth Fierro Garcia	Seguridad en el desarrollo de aplicaciones	
MTI Juan Carlos Rico Resendiz	Planeación y organización del trabajo	
MCA Laura Trejo Medina	Administración de base de datos	
Lider del Equipo	Líder del Equipo	

01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

Apéndices

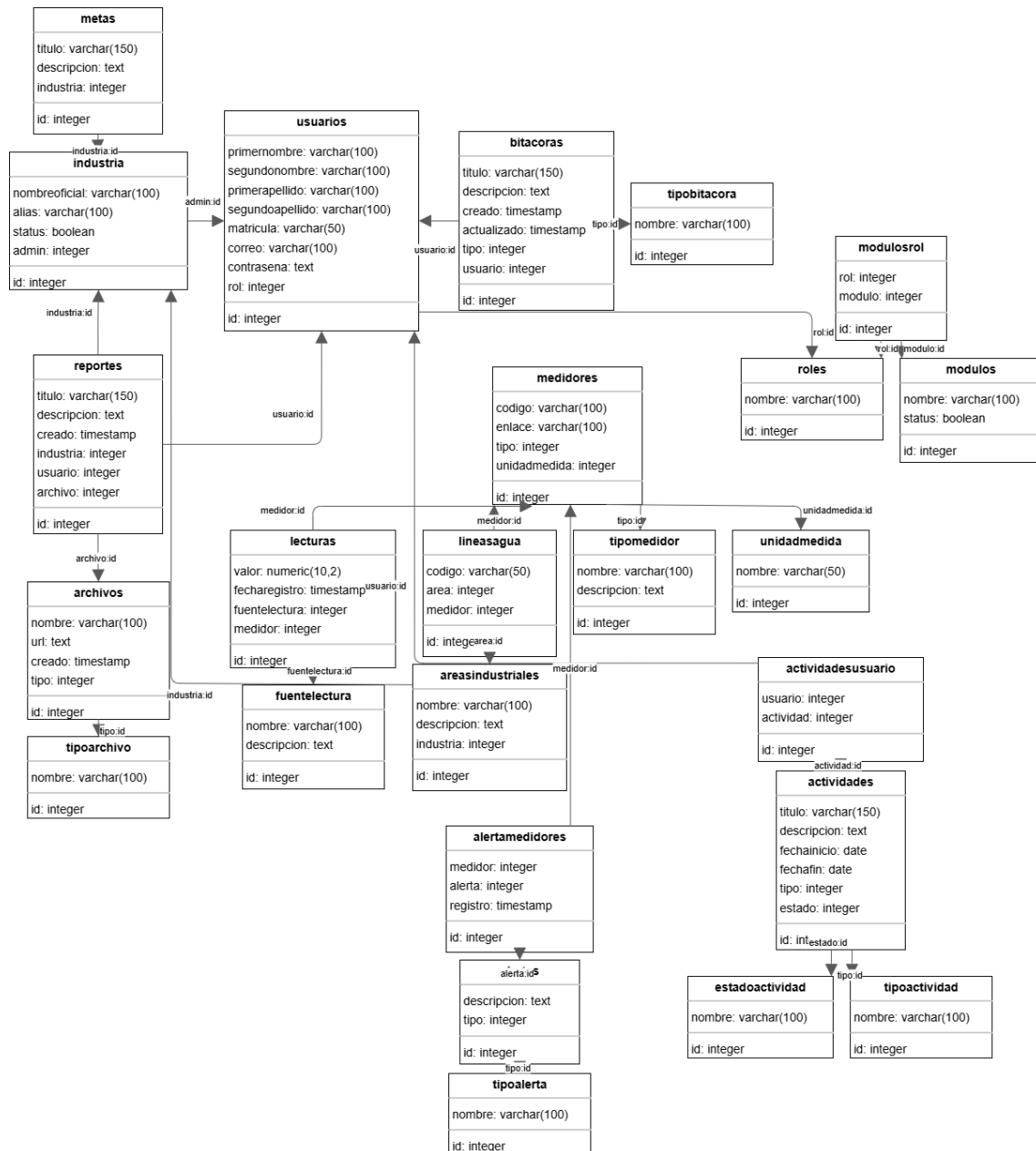
1. Apéndice A: Diagrama Entidad-Relación (DER)



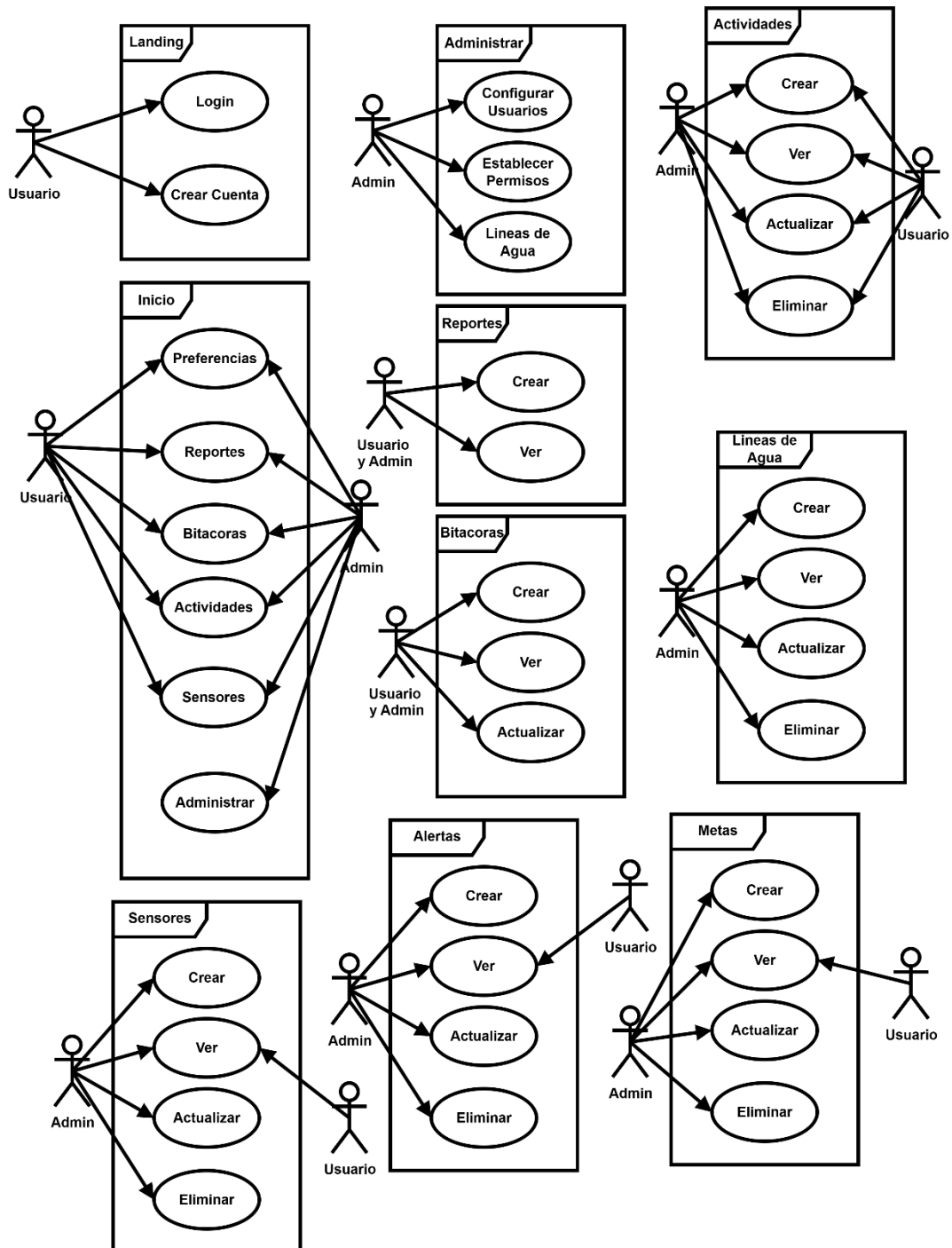
01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

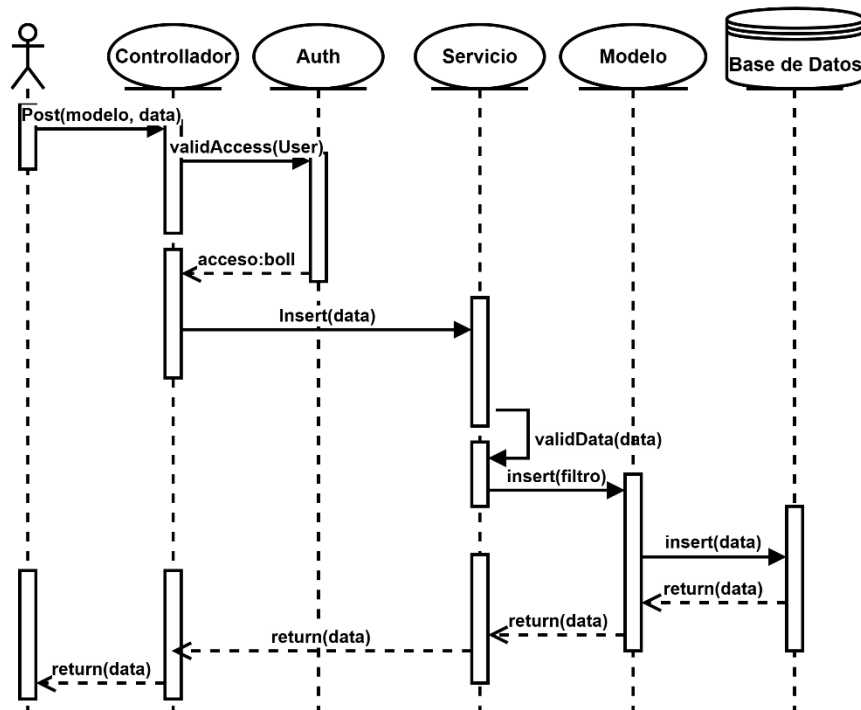
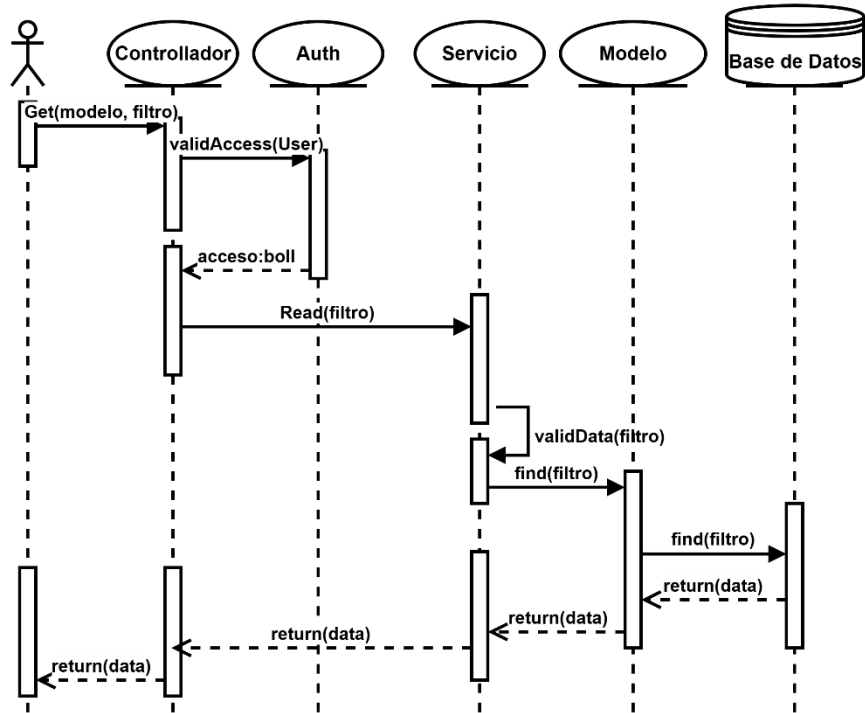
2. Apéndice B: Diagrama Relacional (DR)



3. Apéndice C: Diagramas Casos de Uso



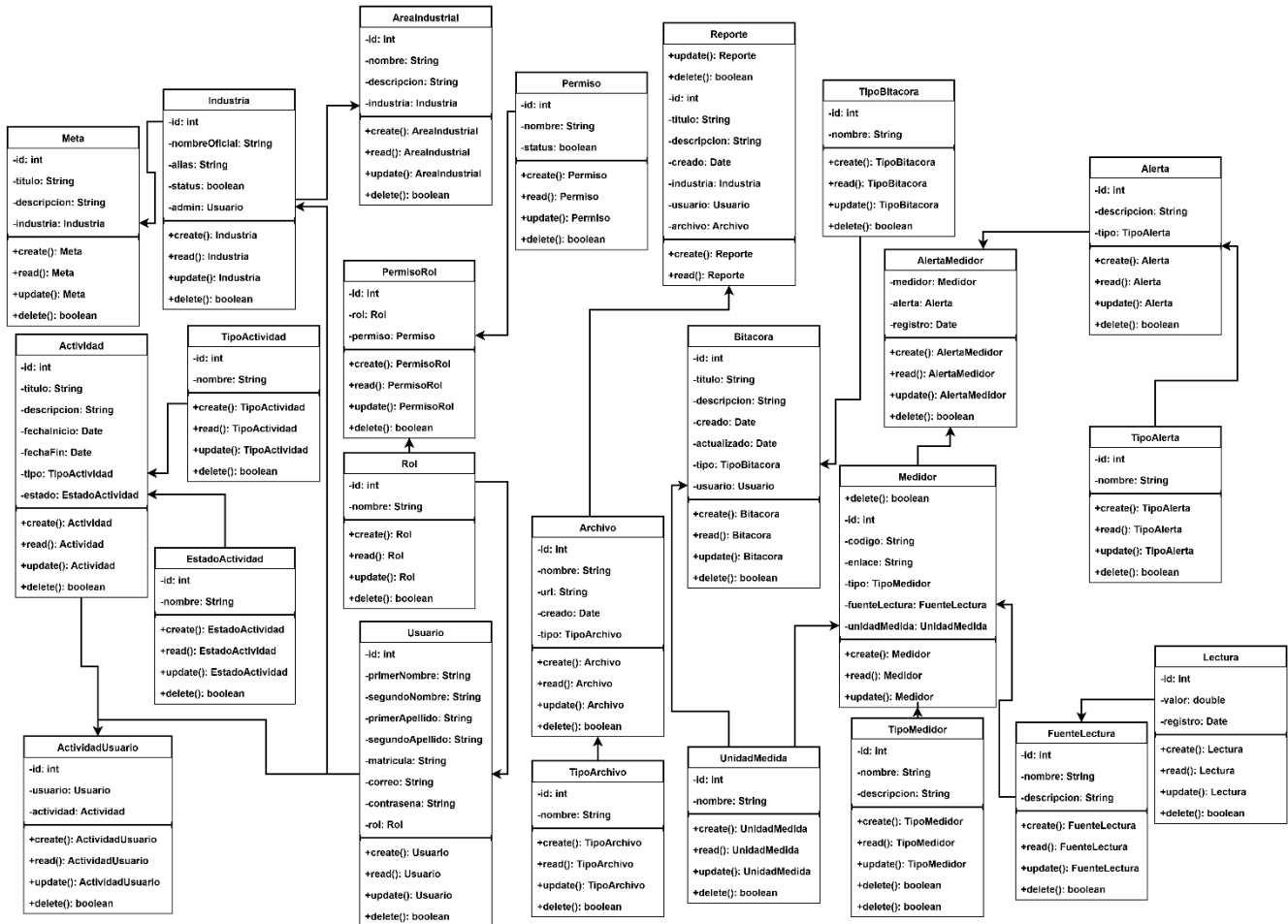
4. Apéndice D: Diagramas de Secuencia



01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

5. Apéndice E: Diagrama de Clases



6. Apéndice F: Planeación a Futuro

Como parte del desarrollo progresivo del sistema, se ha proyectado una nueva etapa dentro del plan de trabajo del proyecto de fin de cuatrimestre. En esta fase se contempla una expansión significativa de la plataforma, orientada hacia el monitoreo y la gestión del consumo energético, con énfasis particular en el uso de la energía eléctrica dentro de los entornos industriales. Esta evolución responde tanto a la necesidad de diversificar las capacidades del sistema como a los compromisos actuales de sostenibilidad y eficiencia que enfrentan las empresas. La ampliación hacia el consumo energético implicará el diseño e implementación de nuevas funcionalidades, muchas de ellas inspiradas en las que ya se han desarrollado previamente para la gestión del recurso hídrico. Entre estas se encuentran la recolección de datos en tiempo real a través de sensores inteligentes, la generación de alertas automáticas ante escenarios de sobreconsumo o fallos en el sistema, la elaboración de reportes periódicos que faciliten el análisis histórico del consumo, así como el cumplimiento de estándares y normativas internacionales, como la ISO 50001, enfocada en sistemas de gestión de la energía y la mejora continua de su uso. A mediano plazo, esta expansión se enmarca dentro de una visión más ambiciosa: convertir la plataforma en un sistema integral de sostenibilidad industrial. Esto permitirá a las organizaciones supervisar y controlar diversos recursos —como agua, energía eléctrica, gas, y eventualmente otros insumos críticos— desde una única interfaz centralizada. La implementación de este enfoque integral no solo busca optimizar la operación diaria de las empresas, sino también fomentar una cultura organizacional basada en la eficiencia energética, la responsabilidad ambiental y la reducción sostenible de costos operativos. Así, la plataforma se posiciona como una herramienta estratégica que contribuye tanto a la competitividad empresarial como al cumplimiento de los compromisos globales en materia de sostenibilidad y cambio climático.

7. Apéndice G: Definiciones, acrónimos y abreviaturas

SRS: Software Requirements Specification, documento que describe de forma detallada los requisitos funcionales y no funcionales del sistema

Frontend: Parte visual del sistema con la que interactúa el usuario. Implementada con tecnologías como Angular o React, consume datos desde el backend.

Backend: Parte lógica del sistema encargada del procesamiento de datos, reglas de negocio, autenticación, y exposición de APIs.

API: Conjunto de servicios web que permiten la comunicación entre el frontend, el backend y dispositivos externos como sensores IoT.

Base de Datos Relacional: Estructura de almacenamiento organizada en tablas.

DevOps: Práctica que combina desarrollo y operaciones para automatizar pruebas, integraciones y despliegues usando herramientas como GitHub Actions.

Docker: Plataforma para contenerizar aplicaciones, permitiendo empaquetar y desplegar el sistema de forma portátil y consistente.

MQTT: Protocolo ligero usado para la transmisión de datos en tiempo real desde sensores IoT hacia el backend.

IoT: Conjunto de dispositivos inteligentes conectados a internet, como sensores de flujo, caudalímetros o medidores de presión de agua.

Broker MQTT: Servidor intermediario que gestiona el envío de mensajes entre dispositivos IoT y el sistema Droply. Ej: Mosquitto, HiveMQ.

Dashboard: Interfaz gráfica personalizable que muestra métricas clave, consumo de agua, alertas y reportes.

Alerta Automatizada: Sistema de notificación que se activa cuando se detectan consumos anormales o sobrepasos de límites configurados.

Token de Autenticación: Código generado por el backend para identificar de forma segura a un usuario o dispositivo autenticado.

TLS: Protocolo criptográfico que garantiza la privacidad de los datos en tránsito, usado para cifrar las conexiones MQTT o HTTPs.

Escalabilidad Horizontal: Capacidad del sistema para aumentar su rendimiento añadiendo más instancias o nodos a la arquitectura.

Alta Disponibilidad: Diseño del sistema para minimizar tiempos de inactividad mediante redundancia y monitoreo continuo.

Balanceador de Carga: Componente que distribuye el tráfico entre diferentes instancias del backend para evitar sobrecargas.

Microservicios: Arquitectura que divide la aplicación en componentes pequeños e independientes que se comunican entre sí.

Logs: Registros detallados de eventos del sistema utilizados para auditoría y solución de problemas.

Backup: Proceso automático de copia de seguridad de los datos del sistema para recuperación ante fallos.

ISO 14001: Estándar internacional para sistemas de gestión ambiental. El sistema debe facilitar la generación de reportes de cumplimiento.

CSV / PDF Export: Funcionalidad que permite exportar los reportes generados en formatos compatibles con auditorías o análisis externos.

Firewall: Sistema de seguridad que regula el tráfico de red permitiendo o bloqueando comunicaciones según reglas definidas.

Avance de Backend

Avance de Base de Datos:

1.1. Tablas Principales

Se ha diseñado un esquema de base de datos relacional que consta de 25 tablas principales organizadas en varios módulos funcionales:

- **Gestión de usuarios y roles:** roles, usuarios, modulos, modulosRol
- **Actividades y bitácoras:** tipoActividad, estadoActividad, actividades, actividadesUsuario, tipoBitacora, bitacoras
- **Gestión industrial:** industria, areasIndustriales
- **Monitoreo y mediciones:** tipoMedidor, fuenteLectura, unidadMedida, medidores, lecturas, lineasAgua
- **Reportes y alertas:** metas, tipoArchivo, archivos, reportes, tipoAlerta, alertas, alertaMedidores

1.2. Relaciones y Claves Foráneas

Todas las tablas están relacionadas mediante claves foráneas que garantizan la integridad referencial. Por ejemplo:

- Los usuarios tienen un rol asignado (usuarios.rol referencia roles.id)
- Las actividades tienen tipo y estado (actividades.tipo y actividades.estado)
- Los medidores tienen tipo y unidad de medida (medidores.tipo, medidores.unidadMedida)
- Las lecturas están asociadas a medidores (lecturas.medidor)

2. Mejoras Implementadas

2.1. Correcciones de Estructura

Realizamos las siguientes correcciones y mejoras en el diseño original:

1. **Normalización de relaciones:** Se eliminaron relaciones circulares (como las referencias entre medidores y lecturas que estaban comentadas)

2. Tipos de datos adecuados:

- Usé SERIAL para claves primarias autoincrementales
- VARCHAR con longitudes apropiadas para textos
- BOOLEAN para estados binarios
- TIMESTAMP para fechas y horas precisas
- DECIMAL(10,2) para valores numéricos con precisión

3. Restricciones de integridad:

- NOT NULL en campos obligatorios
- UNIQUE para campos que no deben repetirse (como correos electrónicos)

2.2. Sistema de Permisos Modular

Implementamos un sistema de permisos modular más robusto:

1. Tabla modulos con 108 módulos/premisos diferentes
2. Tabla modulosRol que asigna permisos a roles
3. Inserción inicial de 4 roles con diferentes niveles de acceso:
 - Desarrollador (acceso completo)
 - Administrador (acceso amplio pero no total)
 - Gerente (acceso operativo)
 - Auditor (solo lectura)

3. Datos de Prueba Iniciales

Incluimos un conjunto completo de datos de prueba para facilitar el desarrollo y las pruebas:

3.1. Usuarios

- 8 usuarios de prueba con diferentes roles
- Incluye los datos del equipo de desarrollo con correos institucionales
- Contraseñas iniciales básicas (deben ser cambiadas en producción)

3.2. Configuración Industrial

- 2 industrias de ejemplo con diferentes estados (activo/inactivo)

01/0/2025

Desarrollo Web Profesional

Equipo 3
8C

- Áreas industriales asociadas a cada industria
- Líneas de agua con medidores asignados

3.3. Medición y Monitoreo

- 3 tipos de medidores (flujo, temperatura, presión)
- 2 fuentes de lectura (manual, automática)
- 3 unidades de medida (litros, °C, PSI)
- 3 medidores de ejemplo con diferentes configuraciones

3.4. Actividades y Bitácoras

- 2 tipos de actividad (inspección, mantenimiento)
- 2 estados de actividad (pendiente, finalizado)
- 2 actividades de ejemplo asignadas a usuarios
- 2 tipos de bitácora (producción, mantenimiento)
- 2 entradas de bitácora con datos completos

3.5. Reportes y Alertas

- 2 tipos de archivo (PDF, CSV)
- 2 archivos de ejemplo con URLs simuladas
- 2 reportes generados asociados a archivos
- 2 tipos de alerta (crítica, advertencia)
- 2 alertas configuradas asociadas a medidores

4. Scripts de Mantenimiento

Incluimos secciones comentadas con sentencias DROP TABLE para facilitar la recreación de la base de datos durante el desarrollo, organizadas en el orden correcto para respetar las dependencias.

5. Consideraciones de Seguridad

1. **Contraseñas:** Las contraseñas están almacenadas como texto plano (solo para desarrollo). En producción serán:
 - Hasheadas con algoritmos seguros (bcrypt, PBKDF2)
 - Incluir salt

- Nunca almacenarse en texto plano
- 2. **Permisos:** El sistema de módulos/permisos permite un control granular de acceso:
 - CRUD completo para cada entidad
 - Asignación flexible por roles
 - Fácil expansión para nuevos módulos

Se implementaron aproximadamente 25 CRUDs, aprovechando la modularidad de los paquetes base desarrollados. Esta estructura permitió crear un servicio genérico y extensible para los diferentes modelos, lo que facilitó un desarrollo más ágil y eficiente.

Capas	Descripcion
Modelo	La capa de modelo define la estructura de los datos en la base de datos.
Service	La lógica de negocio se encarga de filtrar y procesar estos datos antes de que sean utilizados por otras capas.
Controller	El acceso a los métodos se realiza a través de solicitudes HTTP.
Router	El enrutador (router) establece los paths correspondientes a cada endpoint, vinculando cada ruta con la función específica del servicio que será ejecutada en el controlador (controller).

01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

Avance de Landing page

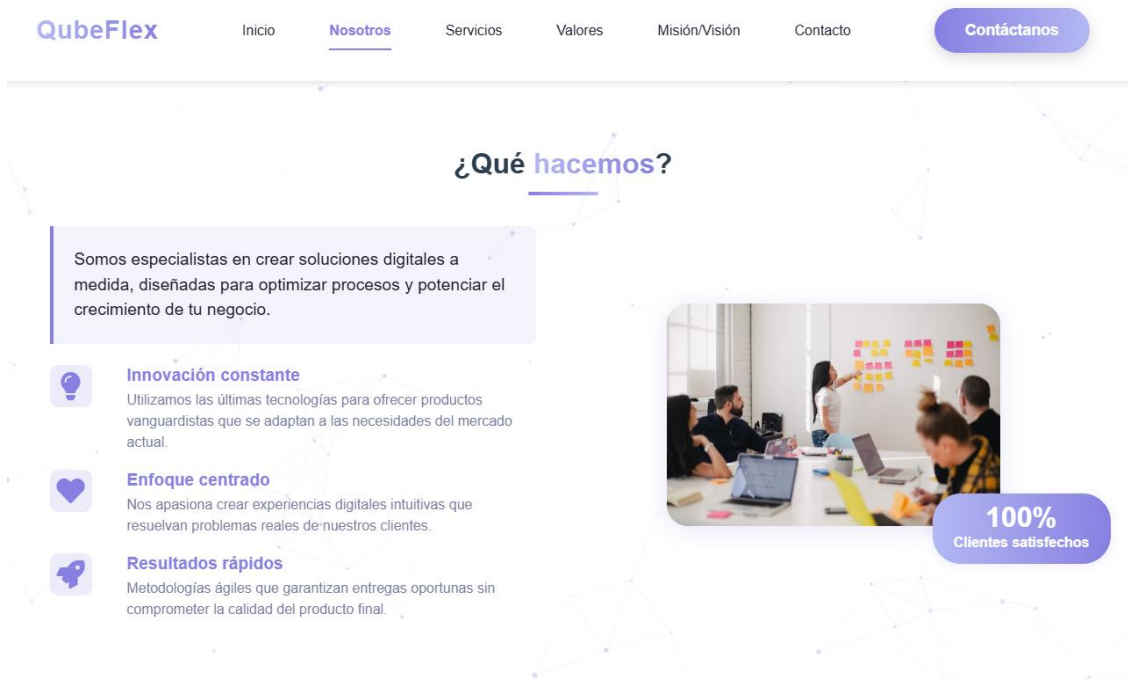
1. Sección Hero (Principal)

Mensaje claro con título impactante y botones accionables. Buen uso de métricas para credibilidad. Sugiero añadir una imagen representativa o video corto para mayor engagement visual. El diseño actual es limpio, pero algo plano.



2. Qué hacemos

Tres pilares bien definidos con lenguaje directo. Estructura funcional que podría mejorarse con iconos ilustrativos y mejor jerarquía visual. El contenido comunica bien el valor diferencial.



01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

3. Servicios

Listado organizado de 4 áreas clave. Descripciones precisas pero diseño muy lineal. Recomiendo tarjetas con hover effects y acceso a más detalles. Falta componente visual que acompañe el texto.



01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

4. Valores

Lista completa de principios corporativos. Texto denso que beneficiaría de un formato más gráfico (íconos, layout circular o interactivo). Buen contenido pero necesita más dinamismo.



5. Misión/Visión

Textos inspiradores, pero presentación básica. Podría usar diseño de citas destacadas o formato split-screen con imágenes. El mensaje es fuerte, el contenedor visual puede mejorarse.



01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

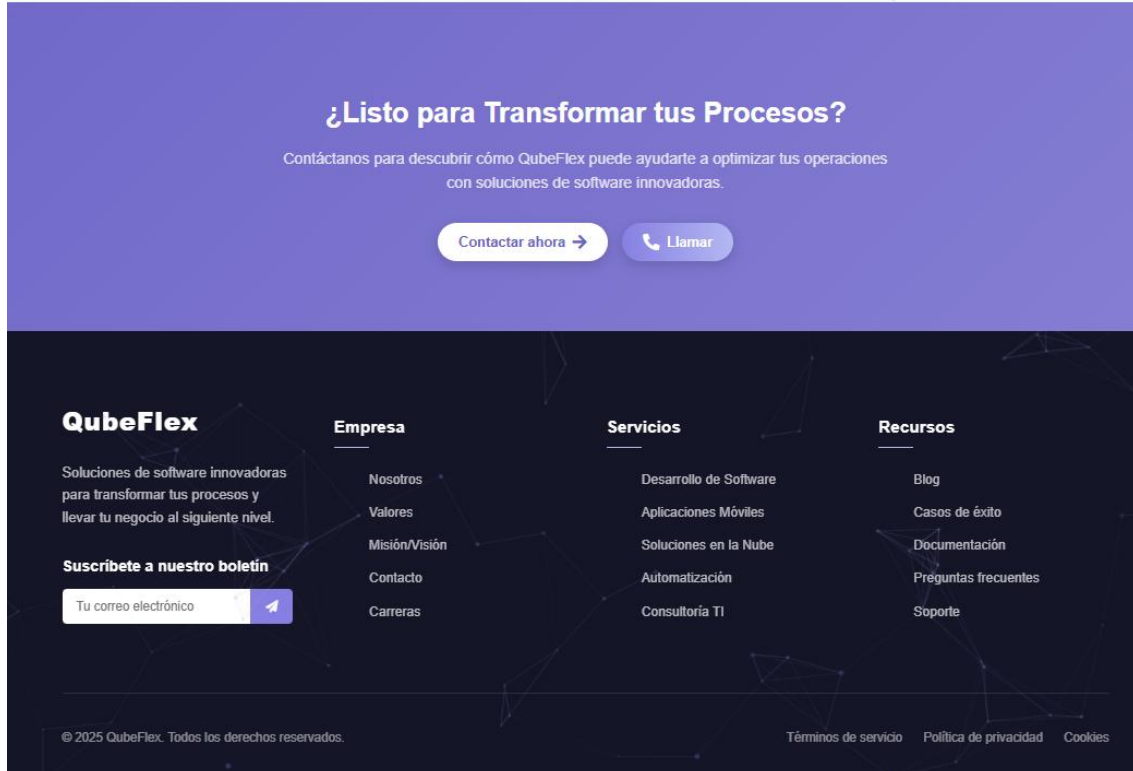
6. Testimonios

Citas auténticas con datos concretos. Falta diseño profesional (fotos, logos, rating). Considera formato carrusel o video testimonios para mayor impacto.



7. Footer

Todos los elementos necesarios pero apretados. Mejorar organización en columnas claras y añadir redes sociales. El CTA de newsletter necesita más prominencia.



01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

Avance de Frontend desarrollo (rama actual)

Rama principal de trabajo donde se integran todos los módulos desarrollados. Aquí se hacen pruebas antes de pasar a producción.

main

Rama principal del proyecto, pensada para versiones estables o productivas. Solo se actualiza desde desarrollo una vez que todo ha sido probado correctamente.

Ramas de características (prefijo FEAT-)

Cada rama FEAT- representa un módulo específico del sistema. Estas ramas son independientes y se enfocan en desarrollar una funcionalidad concreta.

Rama	Descripción
FEAT-componentes	Contiene componentes genéricos y reutilizables utilizados en toda la app.
FEAT-menú-y-navegación	Implementa el menú lateral, rutas, y navegación entre vistas.
FEAT-modulo-actividades	Lógica y vistas para el manejo de actividades (estilo Kanban o tareas).
FEAT-modulo-medidores	Funcionalidad para mostrar o administrar datos de sensores o medidores.
FEAT-modulo-notificaciones	Implementación de alertas y notificaciones del sistema.
FEAT-modulo-usuarios	Gestión de usuarios, perfiles, autenticación y permisos.




Relación entre ramas

Las ramas de características (FEAT-*) se trabajan de forma independiente y luego se integran a la rama desarrollo mediante merge o merge --squash. Una vez que desarrollo está estable, se fusiona con main para subir a producción.

01/0/2025
Desarrollo Web Profesional

Equipo 3
8C

Aprobaciones de avance

Nombre	Cargo	Firma
Dr. Ray Brunett Parra Galaviz	Desarrollo Web Profesional	
M.C Resffa Elizabeth Fierro Garcia	Seguridad en el desarrollo de aplicaciones	
MTI Juan Carlos Rico Resendiz	Planeación y organización del trabajo	
MCA Laura Trejo Medina	Administración de base de datos	