



UrbanFlow

**Conectamos tu ciudad
desde las alturas**

Movilidad Inteligente

Contenido

Nosotros	2
Descripción general del requerimiento	3
Contexto sobre transporte por cables aéreos	4
Sistema técnico bajo estudio	6
Estructura de datos IoT de las cabinas	8
Requisitos técnicos de software	10
Arquitectura general	11
Módulo de gestión de usuarios CRUD	11
Módulo de visualización de datos y analítica	12
Módulo de geoportal / mapa	13
Resumen de requisitos funcionales	14
Cronograma de trabajo y entregables	16

Nosotros

Urban Flow Analytics S.A.S

En Urban Flow Analytics S.A.S. somos una empresa operadora y tecnológica dedicada a la gestión del sistema de metro cable urbano por cabinas en ciudades latinoamericanas. Nuestra misión es mejorar la movilidad sostenible en áreas urbanas y periurbanas conectando comunidades que históricamente han contado con pocas opciones de transporte masivo.

No solo nos ocupamos de la operación del sistema de teleférico, también impulsamos la transformación digital de la movilidad. Buscamos optimizar la experiencia de viaje, aumentar la eficiencia del servicio y fortalecer la seguridad de los pasajeros mediante el uso estratégico de datos y plataformas tecnológicas innovadoras.

Sabemos que el sistema de metro cable presenta retos particulares como el monitoreo constante de los tiempos de espera en estaciones, el seguimiento en tiempo real de la ocupación de las cabinas, la predicción de la demanda en horas pico, la planificación de la operación y el mantenimiento preventivo de las líneas. También reconocemos la importancia de ofrecer información clara y oportuna a los ciudadanos frente a retrasos, cierres o contingencias.

Para enfrentar estos desafíos hemos decidido invertir en el desarrollo de una plataforma digital integral que nos permita, junto con nuestros administradores, operadores y usuarios, acceder a información confiable, reportes en tiempo real y análisis predictivos que fortalezcan la operación diaria del sistema.

Con esta iniciativa buscamos consolidarnos como un referente regional en la gestión inteligente de teleféricos urbanos, aportando al desarrollo de ciudades más conectadas, sostenibles y centradas en las personas.

Descripción general del requerimiento

En Urban Flow Analytics S.A.S. requerimos el desarrollo de una plataforma digital integral que nos permita gestionar, monitorear y analizar de manera eficiente el servicio de transporte urbano en metro cable por cabinas. Nuestro objetivo es centralizar toda la información operativa del sistema de teleférico y ponerla a disposición de administradores, operadores y ciudadanos a través de un portal web interactivo.

Necesitamos que la plataforma permita la administración de usuarios y roles con autenticación segura, de forma que podamos garantizar accesos diferenciados para administradores, operadores de transporte y ciudadanos. Igualmente, debe contar con un módulo para la gestión de rutas, estaciones y cabinas, que nos permita registrar, actualizar y consultar la información operativa en tiempo real.

El sistema debe ofrecer un dashboard que presente información clave e indicadores de desempeño y que, adicionalmente, nos muestre en tiempo real la ubicación de las cabinas en movimiento dentro de la red de transporte. Cada cabina que aparezca en el mapa deberá representarse con un color que refleje su estado operativo: verde para indicar que está en condiciones normales, amarillo para señalar situaciones inusuales y rojo para alertar sobre posibles fallos. Esto nos permitirá tener un control visual inmediato sobre la operación y reaccionar de manera oportuna.

Requerimos también un módulo de analítica avanzada que se alimente de los datos obtenidos a partir de la medición de vibraciones de las cabinas. Para ello contamos con sensores que ya se encuentran desarrollados y en funcionamiento, enviando información cada segundo a una de nuestras bases de datos. Como parte del proyecto se les brindará acceso a esta base de datos y al histórico de registros recolectados, con el fin de que puedan entrenar los modelos de inteligencia artificial que detectarán cuándo una cabina presenta fallos durante la operación.

Los resultados de este análisis deberán mostrarse en una vista específica que organice las cabinas en una cuadrícula de tarjetas, donde cada tarjeta refleje el estado de la condición

con los mismos códigos de color verde, amarillo o rojo. Este módulo deberá implementarse como un microservicio independiente en Python con Flask, comunicándose con el backend principal desarrollado en Node.js mediante APIs REST.

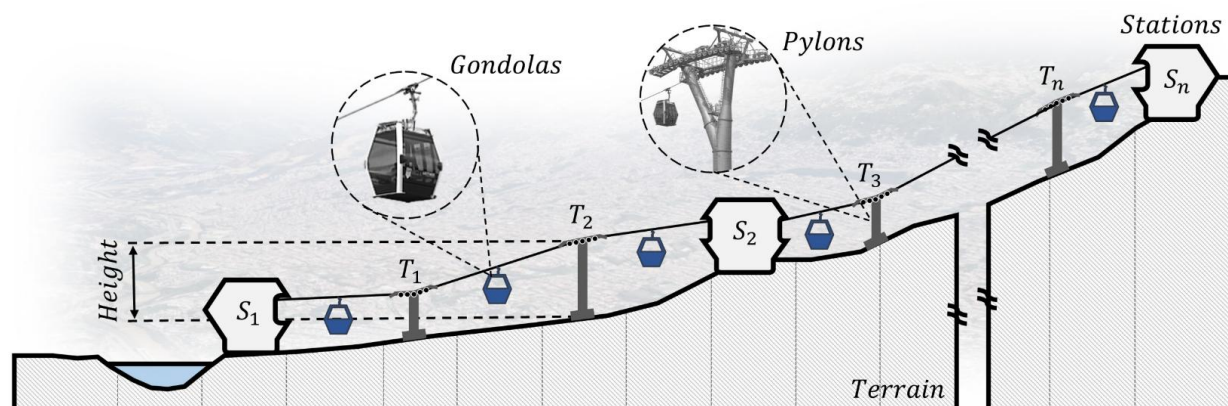
También, requerimos que el sistema siga un patrón de desarrollo MVC y sea implementado en Node JS, tanto para el front como para el back, gestión de vistas y usuarios.

Exigimos que el sistema se desarrolle siguiendo principios de calidad de software, incluyendo la implementación de pruebas unitarias e integración continua, además de una arquitectura clara que nos garantice escalabilidad y mantenibilidad a largo plazo. Finalmente, esperamos que la plataforma se diseñe con un estilo minimalista, accesible y fácil de usar, para que tanto operadores como ciudadanos puedan interactuar de manera intuitiva con la información que se presenta.

Contexto sobre transporte por cables aéreos

El sistema de transporte urbano por cable aéreo de monopinza desenganchable, comúnmente conocido como metro cable o teleférico urbano, constituye una de las soluciones más innovadoras y eficientes para la movilidad en ciudades con topografía compleja y grandes diferencias de altitud. A diferencia de los sistemas tradicionales de transporte masivo en superficie, este modelo emplea cabinas suspendidas por cables aéreos que se desplazan de manera continua a lo largo de una línea, conectando zonas de difícil acceso con los principales corredores de transporte de la ciudad.

El principio de funcionamiento se basa en un cable portador-tractor que recorre toda la línea del sistema. Las cabinas se sujetan al cable mediante una pinza especial, denominada monopinza desenganchable, que garantiza la seguridad del acople durante el trayecto y, al mismo tiempo, permite que la cabina pueda desacoplarse temporalmente en las estaciones para reducir la velocidad, facilitar el embarque y desembarque de pasajeros y volver a engancharse al cable en la salida de la estación sin detener el flujo general del sistema. Este mecanismo es clave para mantener un transporte continuo y eficiente, ya que las cabinas circulan de manera permanente, garantizando una frecuencia alta y tiempos de espera muy reducidos para los usuarios.



Un sistema típico de metro cable urbano opera aproximadamente dieciocho horas diarias, los siete días de la semana y durante prácticamente todo el año, con muy pocas interrupciones programadas para labores de mantenimiento mayor. Esto asegura que la población disponga de un servicio confiable y constante, lo que resulta fundamental en contextos urbanos donde la movilidad es un factor determinante para la productividad y la calidad de vida.

En cuanto a capacidad, cada cabina suele albergar entre ocho y diez pasajeros sentados, y dependiendo de la longitud de la línea y la velocidad del cable —que oscila entre cuatro y seis metros por segundo—, se pueden movilizar entre tres mil y cuatro mil personas por hora y por sentido. Esta capacidad resulta especialmente valiosa en ciudades con áreas montañosas, donde las vías terrestres suelen ser estrechas, congestionadas y de difícil acceso para el transporte público convencional. El cable aéreo no solo logra superar los obstáculos geográficos, sino que lo hace con un nivel de eficiencia energética y ambientalmente sostenible, al operar con motores eléctricos que generan menos emisiones contaminantes en comparación con otros modos de transporte urbano.

La influencia de este sistema en la movilidad urbana es profunda. Para los habitantes de las laderas altas de la ciudad, que tradicionalmente han estado marginados de las redes de transporte masivo, el metro cable representa una conexión directa, rápida y digna con los centros de empleo, educación, salud y cultura. El tiempo de desplazamiento, que en el pasado podía superar la hora debido a las condiciones de las vías, se reduce a pocos minutos gracias a la continuidad y fluidez del sistema aéreo. Además, al integrarse con otros modos de transporte, como el metro o los sistemas de buses articulados, amplía la cobertura de la red y promueve una movilidad más equitativa e inclusiva.

Desde el punto de vista técnico-operativo, la confiabilidad es un aspecto esencial. Los sistemas de monopinza desenganchable incorporan redundancias en los mecanismos de sujeción, monitoreo continuo de los cables y cabinas, así como protocolos de seguridad

que permiten detener la operación de manera controlada en caso de contingencia. Asimismo, la operación prolongada de dieciocho horas diarias requiere una gestión rigurosa de mantenimiento preventivo y correctivo, lo que implica revisar de manera periódica las cabinas, el cable portador-tractor, las estaciones de embarque y desembarque, y los sistemas electromecánicos que sostienen la operación.



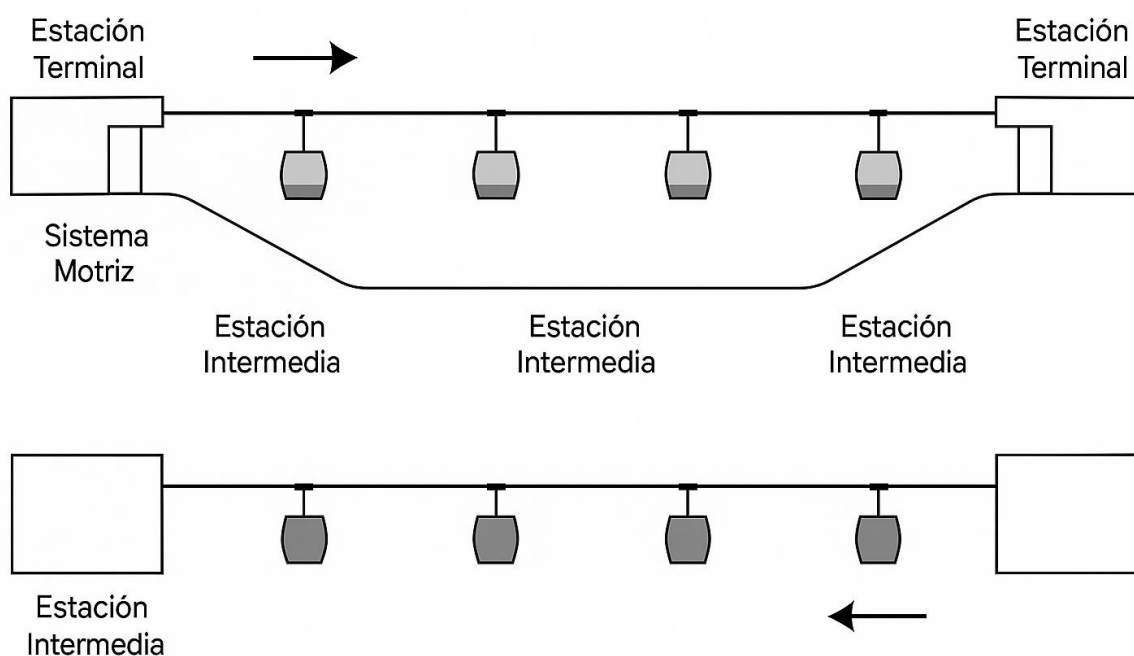
El caso del Metrocable de Medellín constituye un referente internacional del impacto social y urbano de este tipo de transporte. Allí, la implementación de varias líneas de teleférico no solo transformó la movilidad de las comunas de ladera, sino que también generó procesos de inclusión social, revitalización urbana y fortalecimiento del tejido comunitario. Inspirado en ese modelo, el proyecto de UrbanFlow Analytics S.A.S. busca aprovechar las ventajas técnicas y sociales de los sistemas de transporte por cable aéreo, incorporando además herramientas de monitoreo digital y analítica avanzada para optimizar la operación y garantizar que los ciudadanos cuenten con un servicio seguro, eficiente y confiable.

En síntesis, el transporte urbano por cable aéreo de monopinza desenganchable representa un avance tecnológico que responde a necesidades específicas de movilidad en territorios complejos, con un alto impacto positivo en la vida de las comunidades que habitan las partes altas de la ciudad. Su continuidad operativa, su capacidad de movilización y su integración con la planeación urbana lo convierten en un elemento estratégico dentro de los sistemas modernos de transporte masivo.

Sistema técnico bajo estudio

El esquema presentado corresponde al sistema de transporte por cable aéreo de Urban Flow, diseñado bajo la modalidad de monopinza desenganchable en configuración de bucle cerrado. El trazado comprende un total de cinco estaciones: dos terminales y tres intermedias, estratégicamente distribuidas a lo largo del recorrido para garantizar la conectividad entre zonas de ladera y el resto de la red de transporte de la ciudad.

El sistema motriz se encuentra ubicado únicamente en una de las estaciones terminales, donde se concentran los equipos de tracción y control que impulsan el cable portador-tractor a lo largo de todo el circuito. Este diseño permite mantener una operación continua y segura, asegurando la tensión adecuada del cable en el bucle cerrado.



Las estaciones intermedias cumplen la función de puntos de acceso y salida para los usuarios en distintos sectores de la ciudad. Cada una de ellas está equipada con el mecanismo de desacople y reacople de la monopinza, lo que permite que las cabinas reduzcan la velocidad en el interior de la estación para facilitar el embarque y desembarque, sin detener el flujo general del sistema.

El esquema incluye también la representación de cabinas en tránsito entre estaciones. Cada cabina tiene capacidad para transportar entre ocho y diez pasajeros sentados, y circula en intervalos regulares que garantizan una frecuencia constante. El flujo de desplazamiento está indicado mediante flechas: una dirección corresponde al sentido de subida hacia la estación terminal superior y la otra al sentido de bajada hacia la estación terminal inferior.

En condiciones de operación, el sistema funciona dieciocho horas al día, los siete días de la semana y durante la mayor parte del año, con breves interrupciones programadas para

mantenimiento preventivo. Gracias a esta operación prolongada, el sistema es capaz de movilizar miles de usuarios diariamente, representando una solución confiable, sostenible y eficiente para mejorar la movilidad en las zonas de ladera.

Este esquema constituye la base técnica del sistema objeto de estudio, sobre el cual se desarrollará la plataforma digital de monitoreo y analítica avanzada.

Estructura de datos IoT de las cabinas

Cada cabina del sistema de transporte por cable aéreo está equipada con un sensor IoT diseñado específicamente para el monitoreo de condiciones operativas en tiempo real. El sistema puede operar con un número variable de cabinas dependiendo de la longitud de la línea y la capacidad de transporte definida; en una configuración típica, es posible tener entre 90 y 120 cabinas circulando de manera simultánea en el bucle cerrado. Cada sensor se encuentra asociado de manera unívoca al identificador (ID) de la cabina en la que está instalado, lo que garantiza la trazabilidad de los datos y la correcta asociación con la unidad correspondiente. El sensor cuenta con dos subsistemas principales de medición: el módulo de posicionamiento y el módulo de vibraciones.

El módulo de posicionamiento está equipado con tecnología GPS, la cual entrega en tiempo real las variables de latitud, longitud, altitud y velocidad de desplazamiento de la cabina. Estos parámetros permiten no solo ubicar con precisión cada cabina en el mapa, sino también calcular métricas de operación como trayectorias, velocidades promedio, variaciones en altura y sincronización de movimiento respecto al resto del sistema.

El módulo de vibraciones recopila información de la dinámica estructural de la cabina durante su desplazamiento. A partir de la señal de vibración, el sistema extrae indicadores en el dominio del tiempo y de la frecuencia. En el dominio temporal se calculan métricas como el valor RMS (Root Mean Square), la curtosis, la asimetría (skewness), la tasa de cruces por cero (Zero Crossing Rate), el valor pico y el factor de cresta (Crest Factor). En el dominio de la frecuencia se incluyen parámetros derivados del espectro de la señal, tales como la frecuencia media, la frecuencia de mayor energía (frecuencia dominante), la amplitud máxima espectral y la distribución espectral de energía en bandas específicas. Estos indicadores son clave para identificar comportamientos anómalos que puedan anticipar fallas en los componentes de la cabina o del sistema de tracción.

Toda la información recolectada por los sensores se transmite con una frecuencia de un dato por segundo a la base de datos central de Urban Flow, conformando un histórico robusto de datos que permite tanto el monitoreo en tiempo real como el entrenamiento de modelos de inteligencia artificial para la detección temprana de fallos. Este enfoque IoT asegura que cada cabina sea monitoreada de manera independiente, aportando a la

operación un nivel de visibilidad y control que refuerza la seguridad y la confiabilidad del sistema de transporte.

En la base de datos suministrada encontrará las siguientes tablas:

Tabla cabinas

Contiene la información estática de cada cabina del sistema.

Nombre de variable	Tipo de dato	Descripción
cabina_id	INT (PK)	Identificador único de la cabina
codigo_interno	VARCHAR(20)	Código interno de la cabina en la operación
fecha_fabricacion	DATE	Fecha de fabricación de la cabina
estado_actual	VARCHAR(20)	Estado operativo actual (operativo, inusual, alerta, fuera de servicio)

Tabla sensores

Describe cada sensor instalado y lo asocia a la cabina correspondiente.

Nombre de variable	Tipo de dato	Descripción
sensor_id	INT (PK)	Identificador único del sensor
cabina_id	INT (FK)	Referencia a la cabina donde está instalado
modelo	VARCHAR(50)	Modelo del sensor IoT
version_firmware	VARCHAR(20)	Versión de firmware del sensor
fecha_instalacion	DATE	Fecha de instalación del sensor

Tabla mediciones

Tabla central donde se almacena el *payload* de datos que transmite cada sensor cada segundo.

Nombre de variable	Tipo de dato	Descripción
medicion_id	BIGINT (PK)	Identificador único de la medición
sensor_id	INT (FK)	Sensor que reporta el dato
timestamp	TIMESTAMP	Fecha y hora exacta de la medición (sincronizada)
latitud	DECIMAL(9,6)	Coordenada GPS (latitud)
longitud	DECIMAL(9,6)	Coordenada GPS (longitud)
altitud	DECIMAL(8,2)	Altura sobre el nivel del mar (m)
velocidad	DECIMAL(5,2)	Velocidad de la cabina (m/s)
rms	DECIMAL(8,4)	Nivel RMS de vibración
kurtosis	DECIMAL(8,4)	Curtosis de la señal
skewness	DECIMAL(8,4)	Asimetría de la señal

zcr	DECIMAL(8,4)	Tasa de cruces por cero
pico	DECIMAL(8,4)	Valor pico de vibración
crest_factor	DECIMAL(8,4)	Relación pico/RMS
frecuencia_media	DECIMAL(8,4)	Frecuencia media de vibración (Hz)
frecuencia_dominante	DECIMAL(8,4)	Frecuencia con mayor energía (Hz)
amplitud_max_espectral	DECIMAL(8,4)	Amplitud máxima en el espectro
energia_banda_1	DECIMAL(8,4)	Energía espectral en banda baja (ej. 0–50 Hz)
energia_banda_2	DECIMAL(8,4)	Energía espectral en banda media (ej. 50–200 Hz)
energia_banda_3	DECIMAL(8,4)	Energía espectral en banda alta (ej. >200 Hz)
estado_procesado	VARCHAR(20)	Estado resultante tras análisis (operativo, inusual, alerta)

Actualmente Urban Flow cuenta con esta base de datos desplegada en PostgreSQL. El acceso a la base de datos se proveerá a los equipos de desarrollo.

Requisitos técnicos de software

En esta sección se aborda de forma específica los aspectos más relevantes que necesitamos para el desarrollo de la herramienta digital de monitoreo. Esta sección se abordará en los diferentes módulos funcionales y sus requisitos asociados para cada uno. Como ya se había declarado anteriormente, el aplicativo deberá estar desarrollado con una arquitectura MVC (Model – Viewer – Controller), que, en resumen, contiene los siguientes módulos y vistas asociadas:

- A. Módulo de gestión de usuarios CRUD.
 - i. Vista de login / inicio de sesión.
 - ii. Vista de registro de un nuevo usuario.
 - iii. Vista de tabla con usuarios actuales (se puede editar un usuario existente y eliminar usuarios también).
- B. Módulo de geoportal y visualización georeferenciada (monitoreo de cabinas).
 - i. Vista de mapa con monitoreo del movimiento de las cabinas en tiempo real e indicadores de estado de la condición.
- C. Módulo de analítica (dashboard de datos) e inteligencia artificial.
 - i. Vista de dashboard con datos de vibración monitoreada de las cabinas y la predicción del estado de la condición a partir del modelo de IA entrenado con los datos históricos de todas las cabinas.

Arquitectura general

Es necesario que la herramienta se construya en Node JS como framework de desarrollo principal. Allí mismo se integrarán los 3 módulos y las 5 vistas respectivas. También, se deberá integrar el sistema de gestión de usuarios en backend con sus respectivos middlewares para proteger las vistas de dashboard, de registro de nuevo usuario y tabla de usuarios con los roles específicos.

Se espera que el proyecto se estructure con las siguientes carpetas y archivos:



Estructura del proyecto

- **config/** → Archivos de configuración general (ej. conexión a BD, parámetros de aplicación).
- **controllers/** → Lógica de negocio que responde a las peticiones del usuario (intermediario entre models y views).
- **data/** → Datos estáticos o archivos temporales que usa la aplicación (ej. JSON de prueba).
- **db/** → Scripts y configuraciones específicas para la conexión a la base de datos.
- **docs/** → Documentación técnica y manuales del sistema.
- **middlewares/** → Funciones intermedias para validaciones (ej. autenticación, autorización, logging).
- **models/** → Definición de las estructuras de datos y esquemas de base de datos.
- **public/** → Archivos estáticos accesibles desde el cliente (CSS, JS, imágenes).
- **routes/** → Definición de las rutas de la API y del aplicativo web.
- **sql/** → Scripts SQL de creación y carga de la base de datos.
- **utils/** → Funciones auxiliares o utilidades reutilizables en diferentes módulos.
- **views/** → Vistas/renderizado de la interfaz (HTML, plantillas, motores de render).



Archivos raíz

- **.env** → Variables de entorno (ej. credenciales, claves, puertos).
- **app.js** → Archivo principal para levantar la aplicación en Node.js.
- **.gitignore** → Lista de archivos/carpetas a ignorar en Git (ej. node_modules/).
- **LICENSE.md** → Licencia del proyecto (ej. MIT, Apache 2.0).
- **README.md** → Guía inicial de instalación, ejecución y uso del proyecto.

Módulo de gestión de usuarios CRUD

El módulo de gestión de usuarios CRUD tendrá como función principal administrar el ciclo de vida de los usuarios, permitiendo su creación, lectura, actualización y eliminación. Para

garantizar la seguridad en el acceso a las vistas protegidas, se usará un sistema de autenticación basado en JWT (JSON Web Tokens), los cuales se manejarán mediante cookies seguras configuradas con las banderas httpOnly y secure.

El sistema contará con dos roles definidos: usuario y administrador. El rol de usuario podrá acceder al dashboard de datos y al geoportal/mapa, pero no tendrá permisos para ingresar a las vistas de gestión de usuarios, es decir, no podrá registrar, editar ni eliminar usuarios. Estas acciones estarán reservadas únicamente para el rol de administrador.

Para proteger el acceso a las vistas, se implementarán dos middlewares. El primero, verifyToken, validará que la cookie contenga un JWT válido antes de permitir el acceso a las rutas privadas. El segundo, isAdmin, comprobará que el usuario autenticado tenga el rol de administrador para poder acceder a las vistas de gestión de usuarios.

En cuanto al acceso público, solo existirá una ruta abierta: la vista del mapa. Esta será accesible tanto para usuarios registrados como para personas que no se hayan autenticado. Todas las demás vistas estarán protegidas y requerirán autenticación.

La organización de los scripts seguirá la estructura general del proyecto. La lógica del CRUD se desarrollará en el directorio controllers (archivo userController.js). El modelo de usuario, con sus atributos como id, nombre, correo, contraseña encriptada y rol, se definirá en el directorio models (userModel.js). Las rutas se configurarán en routes (userRoutes.js) y estarán protegidas por los middlewares ubicados en middlewares (authMiddleware.js). Las funciones auxiliares, como la generación y verificación de los JWT, se almacenarán en utils (jwtHelper.js). Finalmente, las vistas asociadas al módulo (login, registro y tabla de usuarios) se ubicarán en el directorio views.

Módulo de visualización de datos y analítica

El módulo de dashboard de datos y analítica tendrá como objetivo centralizar y mostrar la información proveniente de los sensores IoT instalados en cada cabina de nuestro sistema de transporte por cable aéreo. Cada sensor transmite datos en tiempo real, que se almacenan en la tabla de mediciones y que utilizaremos tanto para la visualización como para el análisis predictivo.

Esperamos que el tablero nos permita consultar cada cabina de forma individual, mostrando las variables básicas (posición GPS, altitud y velocidad) junto con los parámetros de vibración y espectrales (RMS, curtosis, skewness, tasa de cruces por cero, valor pico, crest factor, frecuencias principales y energías por bandas). Estos datos deberán representarse en gráficos y tablas dinámicas que faciliten la identificación de patrones y comportamientos anormales en la operación de las cabinas.

Entendemos que no somos expertos en analítica de vibraciones, por lo que damos libertad técnica en la elección de metodologías de análisis, visualización y procesamiento intermedio de los datos. Sin embargo, sí requerimos como condición obligatoria la implementación de un modelo de inteligencia artificial capaz de predecir el estado de cada cabina con base en los datos históricos almacenados. Este modelo debe clasificar las cabinas en tres posibles condiciones: modo de fallo, alerta o estado operativo normal.

El resultado del modelo deberá integrarse al dashboard y mostrarse claramente en la vista de cada cabina, de forma que podamos identificar de manera rápida si la operación es segura o si requiere atención. Además, en la base de datos contamos con la columna estado_procesado, donde se registrará el resultado del análisis o predicción para cada medición.

En conclusión, el dashboard de datos deberá cumplir dos funciones principales:

- A. Visualización en tiempo real e histórica de las mediciones de cada sensor asociado a las cabinas.
- B. Predicción automática del estado de la condición de cada cabina, mediante un modelo de inteligencia artificial entrenado con los datos históricos disponibles.

Módulo de geoportal / mapa

El módulo de geoportal será la herramienta central para la visualización geográfica de nuestra red de transporte por cable aéreo. A través de este componente esperamos ver en un mapa la infraestructura completa del sistema y, de manera particular, el movimiento en tiempo real de cada cabina conforme a la información transmitida por los sensores IoT instalados.

Cada cabina deberá representarse como un marcador dinámico en el mapa, el cual debe desplazarse en función de las coordenadas GPS registradas (latitud, longitud y altitud). Además, el color de cada marcador debe variar de acuerdo con el estado de condición de la cabina, reflejando las predicciones generadas en el módulo de analítica: estado operativo normal, alerta o modo de fallo. Esto permitirá identificar rápidamente la situación de cada cabina y facilitará la toma de decisiones.

Si bien definimos estos requerimientos mínimos, también esperamos que el equipo de desarrollo sea propositivo en la incorporación de otras funcionalidades de visualización georreferenciada que puedan generar valor agregado. Ejemplos de estas funcionalidades pueden ser capas adicionales en el mapa, indicadores resumidos de operación por línea, filtros para seleccionar cabinas específicas o vistas históricas del recorrido de una cabina en determinado intervalo de tiempo.

En resumen, el geoportal debe garantizar tres aspectos principales:

- A. La representación de la red de transporte por cable aéreo en un mapa interactivo.
- B. El movimiento en tiempo real de cada cabina en función de los datos GPS.
- C. La diferenciación visual de las cabinas a partir de su estado de condición mediante colores o marcadores dinámicos.

Resumen de requisitos funcionales

ID	Descripción	Módulo
RF1	El sistema debe permitir el registro de nuevos usuarios (solo accesible para administradores).	Gestión de usuarios (CRUD)
RF2	El sistema debe permitir la edición de la información de usuarios existentes (solo accesible para administradores).	Gestión de usuarios (CRUD)
RF3	El sistema debe permitir la eliminación de usuarios (solo accesible para administradores).	Gestión de usuarios (CRUD)
RF4	El sistema debe mostrar una tabla con los usuarios registrados.	Gestión de usuarios (CRUD)
RF5	El sistema debe permitir a cualquier usuario autenticado iniciar sesión mediante login con verificación de credenciales.	Gestión de usuarios (CRUD)
RF6	El sistema debe usar cookies seguras y JWT para la autenticación.	Gestión de usuarios (CRUD)
RF7	El sistema debe implementar el middleware verifyToken para proteger las rutas privadas.	Gestión de usuarios (CRUD)
RF8	El sistema debe implementar el middleware isAdmin para restringir el acceso a las vistas de administración.	Gestión de usuarios (CRUD)
RF9	El rol de usuario debe tener acceso únicamente al dashboard y al geoportal, pero no a la gestión de usuarios.	Gestión de usuarios (CRUD)
RF10	El rol de administrador debe tener acceso total al módulo CRUD de usuarios.	Gestión de usuarios (CRUD)
RF11	El sistema debe permitir la visualización en tiempo real de las mediciones de cada cabina.	Dashboard de datos y analítica

ID	Descripción	Módulo
RF12	El sistema debe permitir la visualización histórica de las mediciones de cada cabina.	Dashboard de datos y analítica
RF13	El sistema debe mostrar las variables de operación básicas: posición GPS, altitud y velocidad.	Dashboard de datos y analítica
RF14	El sistema debe mostrar los parámetros de vibración y espectrales: RMS, curtosis, skewness, zcr, valor pico, crest factor, frecuencias principales, energías por bandas, amplitud espectral.	Dashboard de datos y analítica
RF15	El sistema debe contar con gráficos y tablas dinámicas que permitan explorar los datos.	Dashboard de datos y analítica
RF16	El sistema debe integrar un modelo de inteligencia artificial que prediga el estado de cada cabina (operativo, alerta, fallo).	Dashboard de datos y analítica
RF17	El estado procesado de cada medición debe almacenarse en la columna estado_procesado de la base de datos.	Dashboard de datos y analítica
RF18	El dashboard debe mostrar el resultado del modelo de IA de manera clara en cada vista individual de cabina.	Dashboard de datos y analítica
RF19	El sistema debe mostrar un mapa interactivo con la red de transporte por cable aéreo.	Geoportal y visualización en mapa
RF20	El sistema debe representar cada cabina en el mapa como un marcador dinámico.	Geoportal y visualización en mapa
RF21	Los marcadores de cabina deben moverse en tiempo real según las coordenadas GPS transmitidas.	Geoportal y visualización en mapa
RF22	El color o estado visual de cada cabina debe cambiar en función de su condición (operativo, alerta, fallo).	Geoportal y visualización en mapa

ID	Descripción	Módulo
RF23	El mapa debe contar con al menos una ruta pública accesible sin autenticación.	Geoportal y visualización en mapa
RF24	El sistema debe permitir incorporar funcionalidades adicionales de valor agregado en la visualización (ejemplo: capas adicionales, filtros, vistas históricas).	Geoportal y visualización en mapa

Cronograma de trabajo y entregables

Semana	Actividades	Entregables
1 – 2	- Definición de la arquitectura general del sistema.- Elaboración del backlog inicial.- Construcción de prototipo no funcional en Figma.- Diseño de la base de datos de usuarios.- Documentación técnica inicial.	Documento de arquitectura, backlog inicial, prototipo en Figma, esquema de BD y documentación técnica.
3 – 4	- Desarrollo del backend en Node.js.- Implementación del módulo de gestión de usuarios (CRUD).- Definición y prueba de rutas y controladores.- Implementación de autenticación con JWT y cookies seguras.	Backend funcional con módulo de usuarios, autenticación implementada y documentación actualizada.
5 – 6	- Desarrollo del frontend inicial.- Implementación de vista de login.- Construcción de dashboards básicos.- Primera versión del geoportal con mapa.	Frontend inicial con login, dashboards básicos y mapa funcional.
7 – 8	- Desarrollo de microservicio en Flask.- Implementación de KPIs básicos.- Integración de predicciones simples con datos históricos.	Microservicio Flask operativo, KPIs iniciales y predicciones simples disponibles.
9 – 10	- Integración de backend en Node.js con microservicio Flask.- Integración del modelo	Plataforma integrada con modelo de IA funcional y

Semana	Actividades	Entregables
	de IA con frontend.- Flujo completo de datos sensores → BD → Backend → AI → Frontend.	comunicación Node ↔ Flask ↔ Frontend.
11	- Ejecución de pruebas unitarias y de integración.- Ajustes y refinamientos.- Optimización de seguridad y rendimiento.	Resultados de pruebas documentados, sistema ajustado y versión candidata a entrega final.
12	- Presentación de demo final a Urban Flow SAS.- Entrega oficial de la herramienta.- Documentación técnica final y manual de usuario.	Demo final, sistema completo entregado, documentación técnica y manual de usuario.

