

Dashboard Web de Visualización y Análisis de Movilidad Urbana para Berlín

Roy Sandoval, Susana Toro, Valentina Rendón

Este proyecto consiste en una aplicación web interactiva que permite a usuarios visualizar, consultar y analizar información actualizada sobre rutas, paradas y frecuencias del transporte público de Berlín. Los datos provienen de una [API REST para el sistema de transporte público de Berlín BVG](#). Al integrar una API de BVG, incluye todo el tráfico local de Berlín y Brandeburgo, así como algunos trenes de larga distancia que circulan por la zona. En esencia, devuelve todos los datos que muestra la aplicación de BVG, incluyendo retrasos e interrupciones en tiempo real.

1. Objetivos del Proyecto

Objetivo General

Desarrollar una aplicación web que integre datos de movilidad urbana provenientes de la API <https://v6.bvg.transport.rest> para ofrecer análisis, visualizaciones interactivas y consulta de rutas de transporte público en Berlín.

Objetivos Específicos

- Visualizar rutas, paradas y horarios en mapas interactivos.
- Permitir análisis de uso y congestión por horarios y paradas.
- Ofrecer reportes y descargas de datos relevantes.
- Facilitar la toma de decisiones a usuarios y autoridades.

2. Proceso de Desarrollo

2.1 Reunión con el Cliente

Objetivo: Levantar requerimientos y restricciones reales a través de reuniones con el cliente periódicas a lo largo de un periodo de 6 meses bajo la metodología SCRUM.

Puntos clave:

- Identificar el usuario final (ciudadanos, gobierno, empresas).
- Determinar necesidades de análisis y visualización (mapas, estadísticas, predicciones).
- Especificar funcionalidades mínimas (búsqueda de rutas, reportes, descarga de datos).
- Definir requerimientos de autenticación y roles.
- Establecer la frecuencia de actualización de los datos.

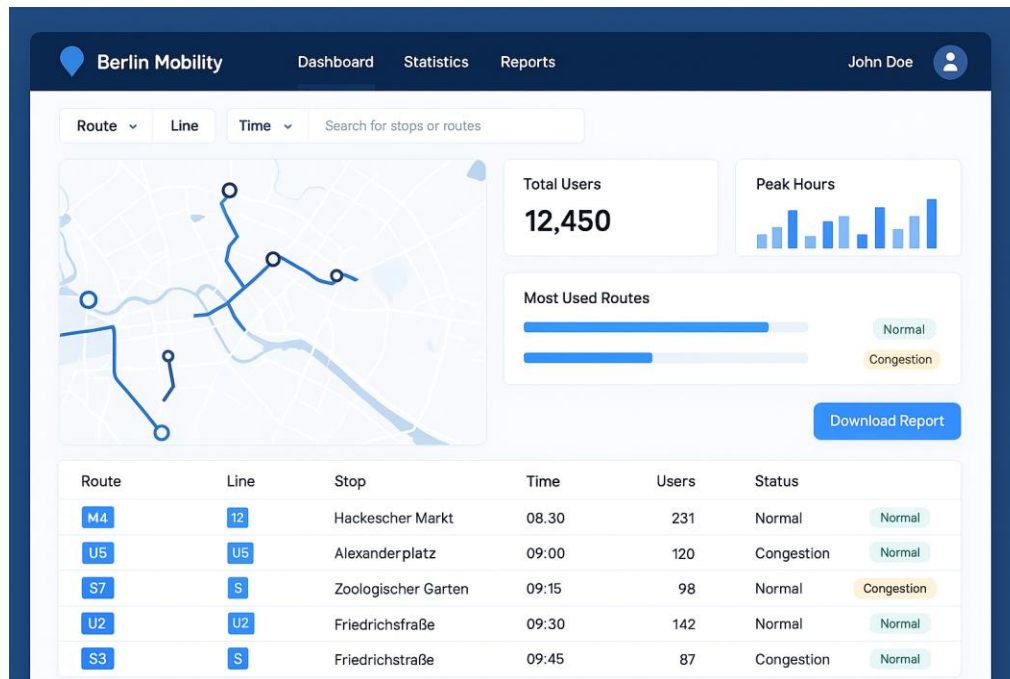
Entregable: Documento de requerimientos funcionales y no funcionales.

2.2 Diseño de Interfaz

Pantallas principales:

- Login / Registro (opcional).
- Dashboard principal con:
 - Mapa interactivo de Berlín (Leaflet/Mapbox).
 - Filtros de rutas, líneas y horarios.
 - Estadísticas de uso (gráficos de barras, líneas, pastel).
 - Buscador de paradas/rutas.
 - Botón de descarga de reportes.
- Vista de detalle de ruta/parada:
 - Horarios.

- Estadísticas históricas.
- Mapa centrado en la ruta/parada.
- Panel administrativo (opcional): gestión de usuarios, logs, configuración.



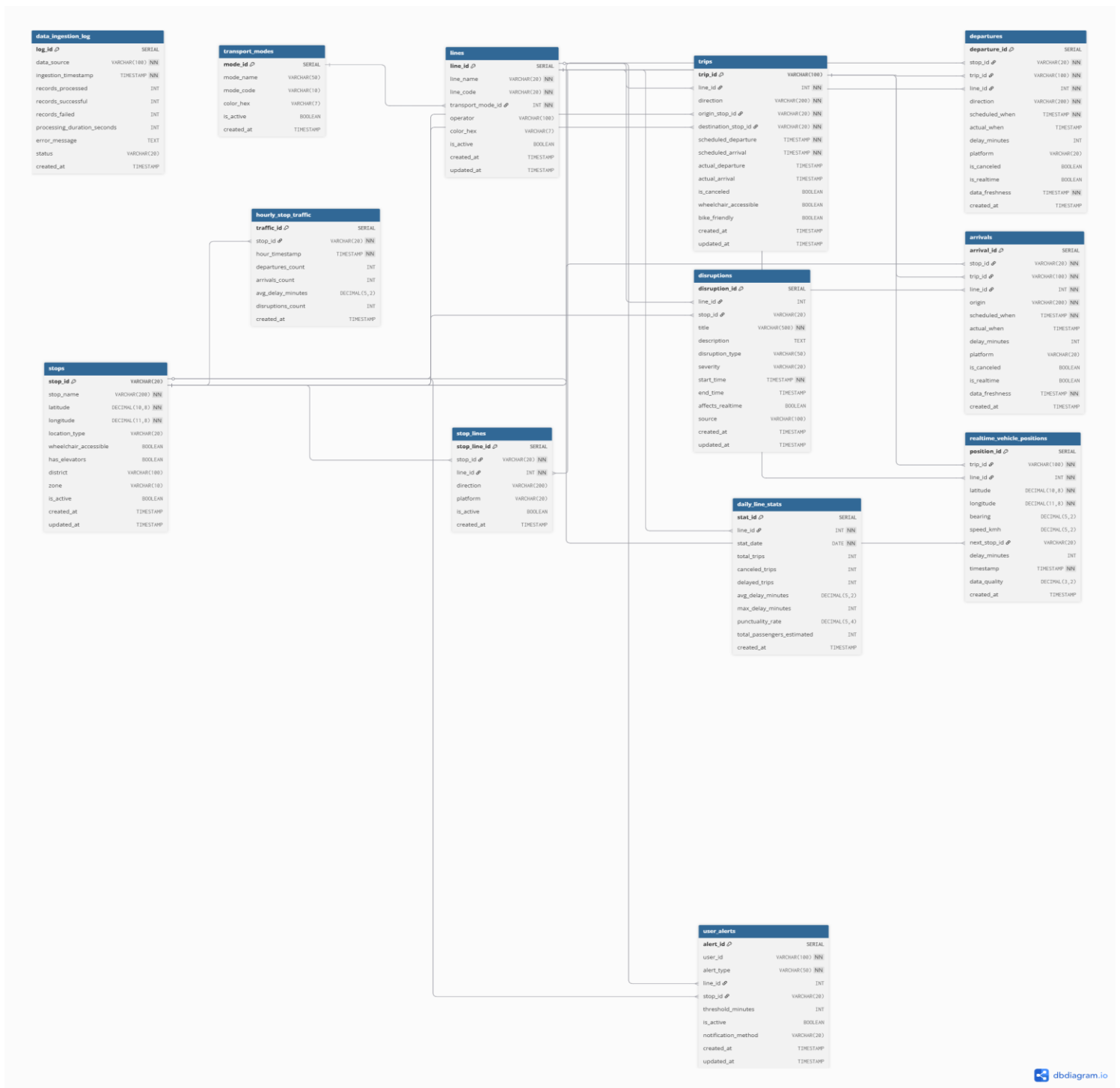
2.3 Diseño de Base de Datos (basada en Snowflake)

El pipeline ETL ya crea las tablas en Snowflake, que sirven de backend para la app. Ejemplo de tablas relevantes:

- routes: id, nombre, color, tipo
- stops: id, nombre, latitud, longitud
- trips: id, route_id, service_id, dirección, fecha/hora
- stop_times: trip_id, stop_id, arrival_time, departure_time
- statistics: stop_id, fecha, total_subidas, total_bajadas

Diagrama simplificado:

routes —< trips —< stop_times >— stops statistics



2.4 Despliegue y Documentación

- Uso de Docker para backend y frontend.

- Documentación técnica y manual de usuario.
- Despliegue en plataforma cloud (Render, Heroku, AWS, etc).

3. Ejemplo de Caso de Uso

Búsqueda de ruta más rápida entre dos paradas:

- El usuario selecciona origen y destino.
- El backend consulta Snowflake para calcular rutas posibles.
- El frontend muestra la ruta óptima en el mapa, con horarios y detalles.