

# Resumen de Proyecto y Conversación (Frontend Vue + NestJS Backend)

## Contexto inicial

Proyecto dividido en microservicios NestJS (auth, incidents, authorization, maintenance, vehicles, drivers) con frontend en Vue 3 + Vite + TypeScript + PrimeVue (tipo Sakai). El cliente quiere un panel administrativo con login, dashboard, gestión de vehículos y conductores.

## Avances frontend

- Se creó el proyecto con Vue 3 + PrimeVue + Tailwind + Pinia + Vue Router.
- Se configuró Login con store de auth, token JWT y guard de rutas.
- Se implementó un layout con Topbar, Sidebar y Dashboard.
- Se añadieron estilos globales (style.css) para adaptar el tema claro (variables de colores, inputs, botones).

## Problemas de estilos

- PrimeVue forzaba dark mode en Dropdown y Calendar → aparecían en negro.
- Se intentó override con CSS (:deep, variables), pero seguía fallando en despleables y bordes.
- Se decidió crear componentes manuales: UiDropdownBasic (HTML/CSS) y UiCalendarBasic (input type=date nativo).
- Resultado: dropdowns y calendarios claros, sin dependencias de PrimeVue.

## Flujo de formularios (DriverFormView / VehicleFormView)

- Formularios estructurados en secciones con dividers.
- Se corrigió alineación de checkboxes y se mantuvo color verde al marcar.
- Se expuso getPayload() en cada form para que el padre pueda pedir un payload ya normalizado.

## Problema crítico: CastError en idDespacho (Mongoose)

- Backend esperaba número, pero llegaba string vacío → Mongoose arrojaba CastError NaN.
- Solución: helper toNumberOrUndefined convierte string → number o lo omite.
- Se añadió cleanEmptyDeep para limpiar vacíos antes de enviar.
- idDespacho nunca se envía como NaN → error solucionado.

## Recomendaciones backend

- Usar ValidationPipe con transform y enableImplicitConversion.
- En DTO de NestJS: @Type(()) => Number) + @IsNumber() @IsOptional() para idDespacho.

## Estado final

- Login funcional con token JWT.
- Layout en modo claro (Topbar + Sidebar).
- Dashboard con métricas centradas y cards.
- Vistas Vehicles y Drivers listas con Pinia stores.
- Forms integrados al backend sin NaN.
- Estilos oscuros de PrimeVue resueltos con componentes propios.

## **Siguiente paso sugerido**

- Probar integración end-to-end: crear usuario, login, crear conductor y vehículo. - Ajustar DTOs backend si faltan conversiones. - Pulir detalles visuales según feedback del cliente.