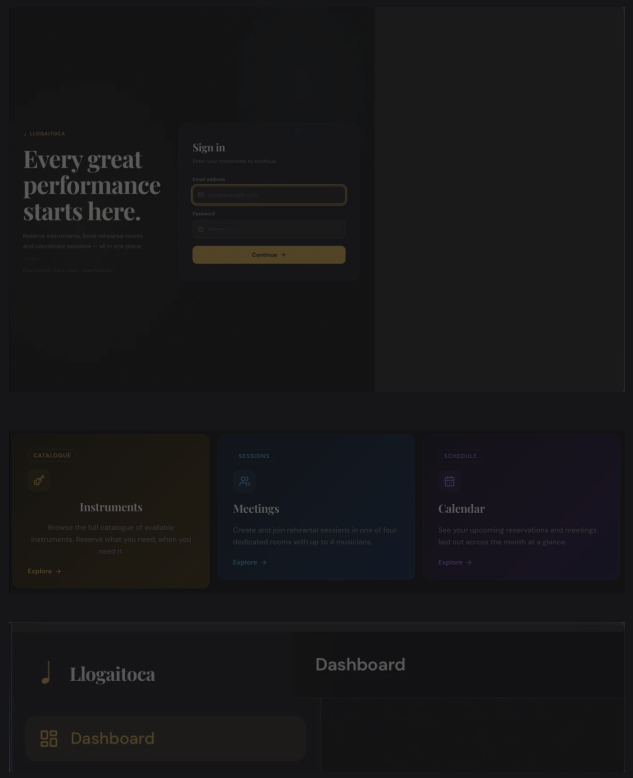
♩ Llogaitoca

IA en el Desenvolupament Frontend

Com he fet servir la intel·ligència artificial
per construir una interfície de producció real

Claude | ChatGPT | React + Vite | Laravel 12



Model IA seleccionat

Quin model vaig escollir i per quin motiu

ChatGPT

Backend & Estructura de dades

Vaig fer servir ChatGPT per tot el que tenia a veure amb el backend en Laravel: la definició d'endpoints, la lògica de negoci, els schemas de la base de dades i els serveis d'autenticació.

La seva rapidesa per generar codi PHP i explicar conceptes de forma pràctica el feia ideal per a la fase inicial del projecte, on calia construir una base sòlida de servidor.

Claude (Anthropic)

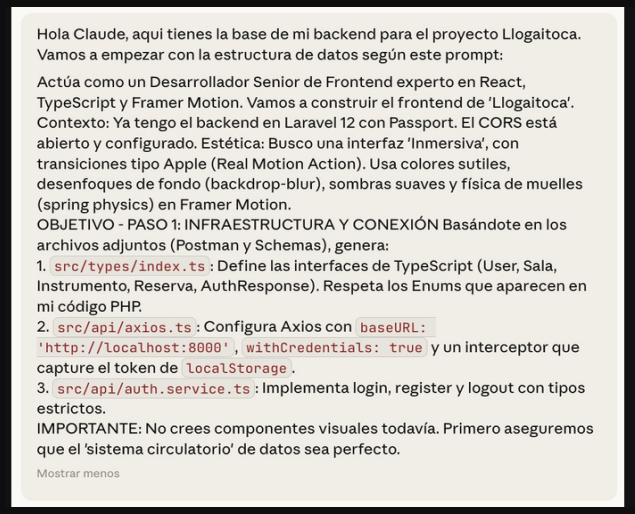
Frontend & Disseny Visual

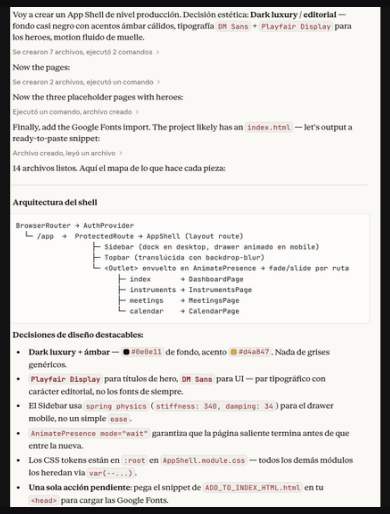
Per construir el frontend vaig canviar a Claude, i la diferència va ser notable des del primer prompt. Sense necessitat de descripcions molt extenses, Claude generava components amb un disseny acurat, consistent i visual —res de layouts genèrics.

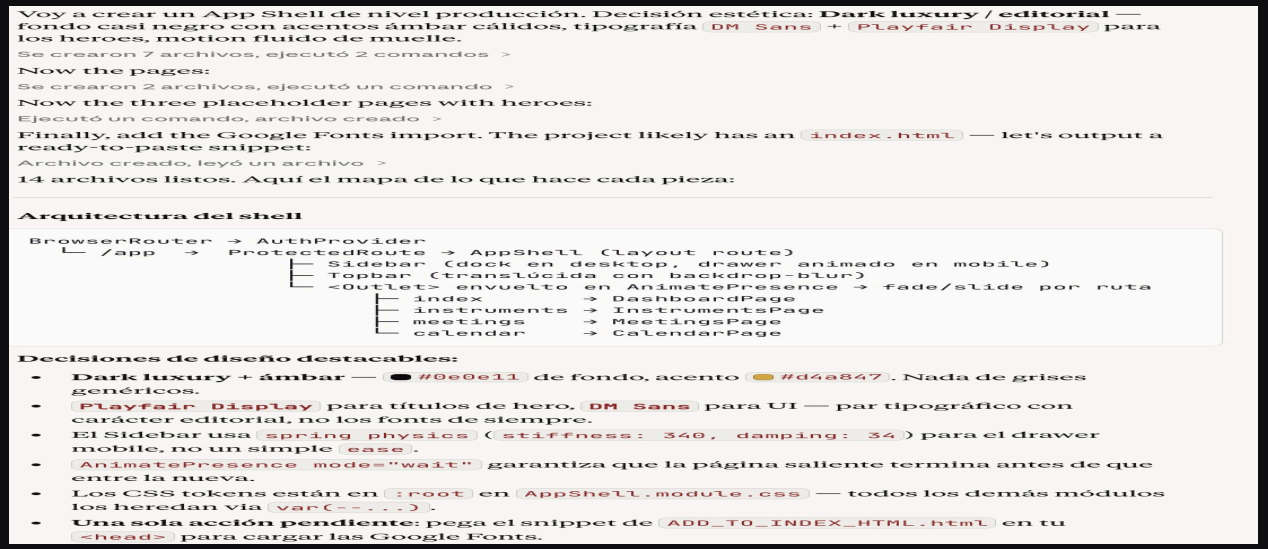
La seva capacitat per entendre el context estètic del projecte (dark luxury, Framer Motion, spring physics) i mantenir la coherència entre fitxers el convertia en la millor opció per al frontend.

Registre d'Interaccions

Els prompts més rellevants i les respostes que van marcar el ritme del projecte







Infraestructura

Primer prompt a Claude: Actua com a dev senior... construeix el sistema circulatori del frontend. Resultat: 3 fitxers TypeScript llestos.

App Shell

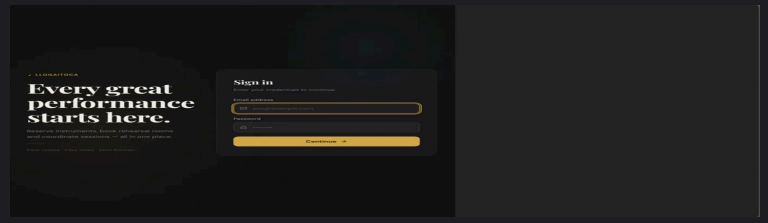
Prompt de l'AppShell amb sidebar, topbar i transicions Framer Motion. Claude va generar 14 fitxers en una sola resposta.

Resolucio de bugs

Cada bug reportat amb captura -> Claude identificava la causa arrel i entregava el fix minim, sense trencar res al voltant. Si era necessari, afegia un croquis gràfic.

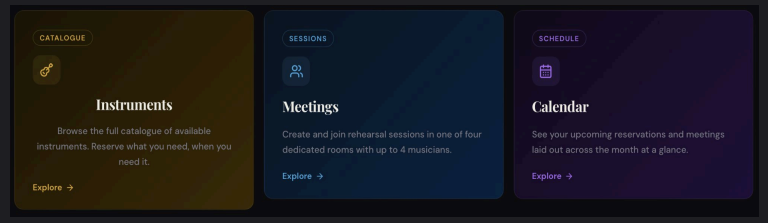
Anàlisi del Codi Generat

Bugs trobats, ajustos realitzats i com els vam resoldre



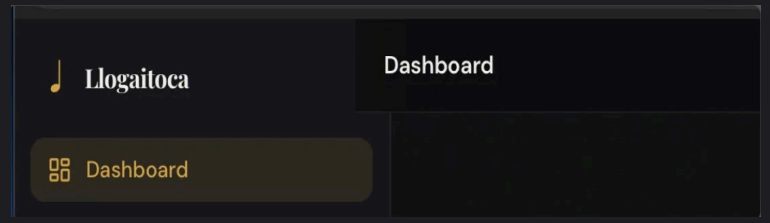
Bug #1 — Amplada de pantalla

El layout no ocupava el 100% de l'ample. La meitat dreta quedava en gris. Causa: el body de Vite per defecte tenia display:flex + place-items:center, que encongía el #root.



Bug #2 — Cards del Dashboard

Les feature cards no ocupaven tot l'ample disponible i quedaven centrades com a mini-cards en lloc de seccions horitzontals a pantalla completa.



Bug #3 — Topbar desalineada

El Sidebar tenia z-index:30 i el Topbar z-index:40. El backdrop-blur del Topbar s'encimava al borde del Sidebar de forma incorrecta.

Solucions implementades

Fix width body/root

body { margin:0; min-width:320px; min-height:100vh; }
#root { width:100%; min-height:100vh; }

Fix topbar alignment

height: var(--topbar-h, 56px) al .brand del Sidebar + border-bottom consistent

Fix API missing endpoint

Endpoint afegit al backend per actualitzar image_url dels instruments (oblidat en la versió inicial)

Connexió Frontend ↔ Backend

Com vam connectar React amb Laravel 12 + Passport

1

Sistema circulatori

src/types/index.ts, src/api/axios.ts i auth.service.ts — la base de dades de TypeScript i el client Axios amb interceptor de token.

2

AuthContext + rutes protegides

Context de React per gestionar la sessió. La ruta /app comprova isAuthenticated() abans de renderitzar el contingut.

3

App Shell i navegació

AppShell.tsx amb Sidebar, Topbar i AnimatePresence per a les transicions entre rutes. 14 fitxers generats en un sol pas.

4

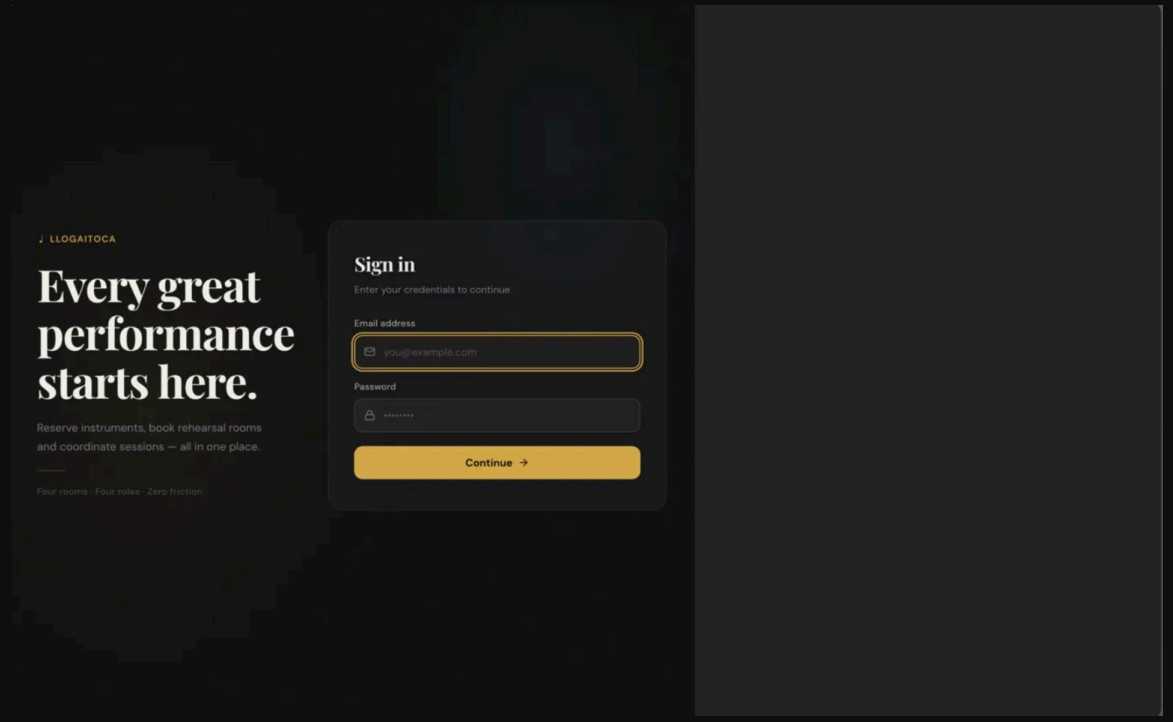
Pàgines funcionals

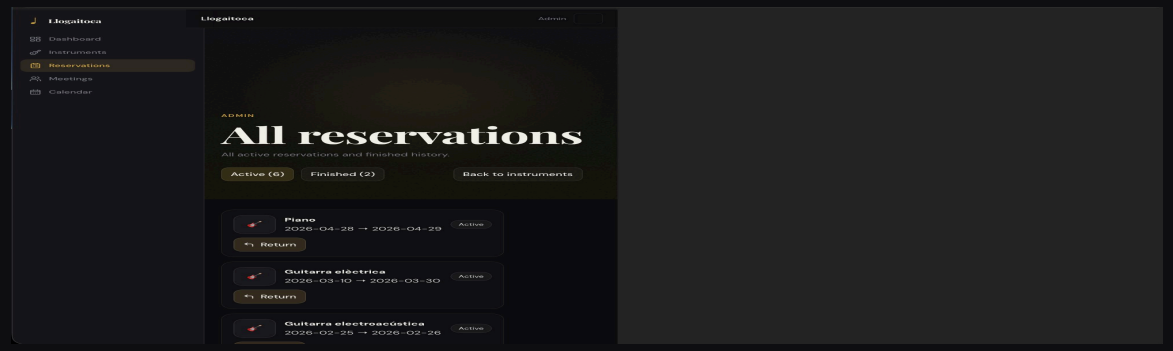
InstrumentsPage, ReservationsPage, MeetingsPage i CalendarPage connectades als seus serveis API respectius.

5

Ajustos d'endpoints

Algunes funcionalitats revelaven endpoints que faltaven al backend (ex: PATCH image_url). Iteració ràpida entre frontend i backend.





Reflexió sobre el Procés d'Aprenentatge

En el meu cas, el punt de reflexió ha estat la IA en general. L'he hagut d'utilizar al 100% en aquesta segona entrega i he entés la seva utilitat, practicitat per a la productivitat i tot el que això estalvia. Tot i així, m'he trobat en molts moments inverting molt de temps en les explicacions (necessàriament concretes) per a tindre el resultat esperat (i obligatóriament correcte).
Un altre punt, ha estat el git flow. La IA no ha sapigut o no ho ha tingut en compte i això ha fet que jo haguès d'estar atent.
Tret d'això, faig les paus amb la IA i m'emporto les claus indicades aqui al lateral.

I ja, en general, segons el meu parer; he tingut la sensació de que BE i FE s'han de dissenyar junts.

Takeaways clau

- Prompting estructurat = menys iteracions
- La IA no substitueix el criteri, l'acelera
- Cada bug resolt va ser una lliçó de CSS/React
- Backend i frontend s'han de dissenyar junts