

# PROJECTE DE DESENVOLUPAMENT D'APLICACIONS WEB



Enable intelligent Neural Assistant



I.E.S. Maria Enríquez  
Gandia

Presentat per **JAVIER PEIRO**

Cicle Formatiu de Grau Superior

Curs 2024-2025





## INDEX

1	Introducció.....	6
1.1	Informació general.....	7
1.2	Objectius del projecte.....	7
2	Contextualització del projecte.....	8
2.1	Viabilitat i objectius concrets.....	8
2.2	Tecnologies avaluades.....	9
2.3	Tecnologies alternatives.....	11
3	Planificació del projecte.....	13
3.1	Temporalització.....	13
3.2	Cronograma.....	15
4	Desenvolupament del projecte.....	16
4.0	Fase prèvia.....	16
4.0.1	Estudi d'alternatives.....	16
4.0.2	Instal·lació i configuració d'eines.....	16
4.0.3	Creació del repositori GitHub.....	17
4.1	Especificació i disseny previ.....	17
4.1.1	Requisits i funcionalitats principals.....	17
4.1.2	Model E/R i diagrames UML.....	18
4.1.3	Prototip inicial amb Figma.....	18
4.1.4	Disseny de rutes i controladors a Laravel.....	20
4.2	Desenvolupament Backend.....	24
4.2.1	Creació de models, migracions i relacions.....	24
4.2.2	Autenticació amb Laravel Sanctum.....	24
4.2.3	Controladors i serveis per gestió d'usuaris.....	25
4.3	Integració de Gemini 2.5.....	25
4.3.1	Configuració API.....	26

4.3.2	Comunicació amb la IA.....	26
4.3.3	Emmagatzematge i processament de converses.....	26
4.4	Implementació del Xat i WebSockets .....	28
4.4.1	Desenvolupament inicial de les vistes .....	28
4.4.1.1	Configuració inicial amb Blade i Tailwind .....	28
4.4.1.2	Estils bàsics amb CSS.....	28
4.4.2	Implementació del xat amb Livewire/AJAX.....	28
4.4.3	Connexió frontend-backend.....	28
4.4.4	Ampliació i refinament de les vistes.....	29
4.4.4.1	Vistes específiques d'alumnes.....	29
4.4.4.2	Vistes específiques del professor.....	29
4.4.5	Implementació Broadcasting amb Laravel Echo i Pusher.....	29
4.5	Control i gestió del professor .....	29
4.5.1	Panell professor (supervisió, bloqueig IA).....	30
4.5.2	Historial de converses per alumne.....	30
4.6	Seguretat, Optimització i Pujada.....	31
4.6.1	Revisió seguretat API i autenticació.....	31
4.6.2	Optimització rendiment.....	31
4.6.3	Correcció errors.....	31
4.6.4	Ajustos finals.....	32
4.6.5	Pujada GitHub.....	32
4.7	Documentació tècnica.....	33
5	Interfície final de l'aplicació.....	33
5.1	Vista de l'alumne.....	33
5.2	Vista del professor.....	34
6	Bibliografia .....	36



6.1	Llibres i vídeos.....	36
6.2	Manuals web.....	36
6.3	Fòrums i blogs.....	36
7	Anàlisi i valoració d'alternatives.....	37
7.1	Anàlisi i conclusions.....	37
7.1.1	Objectius.....	37
7.1.2	Conclusions.....	37
7.1.3	Autoavaluació.....	37
7.2	Millores o alternatives proposades.....	37
7.3	Agraïments.....	38
7.4	Dificultats trobades.....	38
7.5	Relació amb els mòduls estudiats.....	39
8	Annexos.....	40
8.1	Estructura de carpetes.....	40
8.2	Diagrames.....	43
8.2.1	Diagrama E/R.....	45
8.2.2	Diagrama de classes.....	46
8.2.3	Diagrama de flux d'aplicació.....	46

## 1. INTRODUCCIÓ

En un principi, la idea original era desenvolupar un projecte relacionat amb mapes interactius i jocs de guerra, una temàtica que sempre m'ha interessat per la seua complexitat tècnica i les seues possibilitats d'interacció visual. No obstant això, durant la fase inicial del curs, el professor Javier Garcia Sabater va proposar una idea que em va semblar no sols innovadora, sinó també molt útil i amb un gran potencial pràctic: crear una eina basada en intel·ligència artificial per a l'entorn educatiu.

Aquesta proposta partia d'una realitat que coneixíem de primera mà: molts professors es troben desbordats davant la quantitat de continguts i canvis constants que es produeixen en l'àmbit educatiu. En algunes classes, per molt bona voluntat que hi haja, és impossible atendre adequadament tots els alumnes. Per això, vaig veure clar que aquest projecte podia ser una gran oportunitat per aportar una solució real a un problema actual.

L'objectiu no era només desenvolupar una eina que ajudara els estudiants, sinó també proporcionar al professorat un instrument de suport i control, capaç de supervisar l'ús de la IA per part dels alumnes i afavorir-ne una adopció responsable. Amb aquesta eina, el professor pot controlar el context de la conversa, limitar el nombre d'interaccions i revisar les converses dels alumnes així com limitar l'ús durant activitats específiques.

A més, aquest projecte també busca reduir la por o desconfiança que moltes persones encara tenen envers la intel·ligència artificial. Ben utilitzada, pot ser una ferramenta poderosa per complementar l'ensenyament tradicional, i aquest és precisament un dels pilars fonamentals del projecte EINA.



## 1.1 Informació General

**Nom de l'alumne:** Javier Peiró Aguado

**Cicle:** Desenvolupament d'Aplicacions Web

**Curs escolar:** 2024 – 2025

**Denominació del projecte:** EINA - Eina Intel·ligent per a l'Aula

**Projecte vinculat a FCT:** No vinculat

**Tutor individual:** Eloy Gutiérrez Paredes

**Tutor col·lectiu:** Enrique Savall Seguí

## 1.2 Objectius del projecte

### Objectius generals

- Desenvolupar una aplicació web funcional amb Laravel que permeti la interacció entre alumnes i una IA generativa amb la supervisió del professor.
- Facilitar al professorat una eina de seguiment i control de les converses entre alumnat i la intel·ligència artificial.
- Fomentar l'ús responsable i supervisat de la IA en l'entorn educatiu.
- Reduir la resistència i la preocupació a la IA mitjançant una implementació pràctica, clara i controlada.

### Objectius específics

- Permetre al professor assignar un context a la IA per a cada classe o tema.
- Limitar el nombre d'interaccions dels alumnes segons les condicions establides pel docent.
- Crear un sistema d'autenticació i rols amb Laravel Breeze i Sanctum per diferenciar professorat i alumnat.
- Registrar totes les converses mantingudes amb la IA per a la seua posterior consulta.
- Implementar funcionalitats en temps real com la visualització de missatges i canvis de context mitjançant Laravel Echo i Pusher.

- Dissenyar una interfície clara, responsive i usable mitjançant Blade, Tailwind CSS i JavaScript.
- Permetre al professor consultar l'historial de converses de cada alumne i controlar quan pot fer ús de la IA.

## 2. CONTEXTUALITZACIÓ DEL PROJECTE

### 2.1 Viabilitat i objectius concrets

El projecte EINA naix amb la intenció de resoldre una necessitat real detectada en l'àmbit educatiu: la manca d'eines efectives per a controlar i aprofitar les interaccions amb intel·ligència artificial a l'aula. Amb l'arribada massiva de models d'IA com ChatGPT, Gemini o Copilot, molts centres educatius es troben desbordats per l'ús descontrolat que en poden fer els alumnes, especialment en contextos on el professorat no pot estar pendent de cadascun d'ells en temps real.

La viabilitat del projecte es fonamenta en l'ús de tecnologies madures, estables i ben documentades com Laravel, Tailwind, JavaScript i Pusher. A més, el fet que el desenvolupament es base en el framework Laravel facilita una implementació modular i escalable, adequada per a ser adaptada a les necessitats de cada centre.

**JavaScript** **Laravel**



**tailwindcss**



EINA permet al professorat:

- Assignar un context concret a la IA perquè responga d'acord amb una situació educativa definida (per exemple, actuar com a professor de CSS o com a guia de resolució de problemes).
- Controlar la quantitat d'interaccions disponibles per a cada alumne.
- Bloquejar o activar l'accés a la IA segons la dinàmica de la classe.
- Visualitzar les converses dels alumnes seleccionats.

Els alumnes, per la seua banda, poden:

- Conversar amb la IA dins d'un context docent controlat.

Aquest plantejament equilibra la llibertat de l'alumne amb el control necessari per part del docent, convertint-se així en una eina pedagògica poderosa, segura i adaptable. Tot això fa que el projecte siga completament viable en entorns educatius reals, tant a nivell tècnic com metodològic.

## 2.2 Tecnologies avaluades

Per al desenvolupament d'EINA, s'ha fet ús de tecnologies estables, modernes i àmpliament utilitzades en el desenvolupament web. La tria d'aquestes tecnologies ha estat condicionada tant per l'experiència prèvia com per la seua documentació, comunitat activa i compatibilitat entre components.

### Laravel (PHP)

Framework backend utilitzat per la seua robustesa, facilitat d'organització de codi, gestió de rutes, autenticació integrada i gran ecosistema. Ha sigut la base per a tota la lògica de negoci del projecte.

### Laravel Breeze + Sanctum

Utilitzat per a la gestió d'autenticació, rols i sessions segures, amb suport per a SPA i API tokens. Breeze facilita l'estructura d'autenticació amb Blade i Tailwind.

**MySQL**

Base de dades relacional triada per la seua estabilitat i integració nativa amb Laravel mitjançant Eloquent ORM.

**Tailwind CSS (amb plugins)**

Framework de disseny utilitzat per a donar estil a la interfície de forma modular i responsive. Ha permés aplicar disseny mobile-first i adaptar ràpidament els components visuals. A més del nucli bàsic, s'han instal·lat els plugins `@tailwindcss/forms` per millorar l'estil dels formularis i `@tailwindcss/line-clamp` per limitar visualment la quantitat de línies de text mostrades en vistes com les descripcions de context.

**Blade (motor de plantilles)**

Permet integrar codi PHP amb HTML de manera senzilla i mantenible. S'ha utilitzat per a totes les vistes del projecte.

**JavaScript (i Alpine.js)**

Usat per a interaccions dinàmiques a les vistes. Especialment útil per a gestionar el xat en temps real, mostrar o amagar components i actualitzar l'estat de l'interfície.

**Laravel Echo + Pusher**

Tecnologia per a la comunicació en temps real via broadcasting. Permet que el professor veja missatges nous i canvis de context sense necessitat de recarregar la pàgina.

**Laravel Telescope**



S'ha utilitzat per monitoritzar les peticions HTTP, emissions d'esdeveniments broadcast i execució de serveis durant la fase de desenvolupament i proves. Ha resultat essencial per a la depuració de funcionalitats en temps real i comprovació del correcte funcionament de les cues, emissions i respostes de l'API.

### **Gemini 2.5 Flash (Google AI)**

Model d'intel·ligència artificial triat per la seua capacitat de mantenir contextos persistents i pel seu pla gratuït amb accés via API. S'ha integrat mitjançant peticions HTTP personalitzades amb una clau API pròpia, i s'està estudiant la transició a un sistema d'autenticació OAuth per a futures versions.

### **Vite**

Utilitzat com a bundler per a compilar i optimitzar els fitxers CSS i JS del projecte, en lloc de Webpack. S'integra nativament amb Laravel des de les últimes versions.

### **Composer i npm**

Gestors de dependències PHP i JavaScript, essencials per a instal·lar paquets com Breeze, Tailwind o Echo.

## **2.3 Tecnologies Alternatives**

Durant el desenvolupament del projecte també es valoraren altres tecnologies que finalment foren descartades per motius de compatibilitat, mantenibilitat o viabilitat tècnica.

### **beyondcode/laravel-websockets**

Inicialment es va intentar implementar comunicació en temps real mitjançant aquest paquet, però es va descartar per incompatibilitat amb Laravel 12. Aquest fet

va motivar la migració definitiva cap a Pusher com a servei extern per al broadcasting.

### Laravel Jetstream

Es va estudiar l'ús de Jetstream com a alternativa a Breeze per la seua integració amb Livewire i funcionalitats avançades. No obstant això, es va considerar massa pesat per a l'abast i simplicitat del projecte actual.

### Firebase Realtime Database

Tot i que ofereix funcionalitats en temps real integrades, es va descartar per dependències externes no necessàries i per incompatibilitat amb la lògica basada en Laravel i MySQL.

### Chat GPT

Va ser avaluat com a opció principal, però degut a la poca capacitat del model de la API per fer consultes gratuïtes, i a que els models amb capacitat de mantindre context requerien fins a 10 vegades més s'ha obtat per **Gemini 2.5 Flash**.

### Gemini Flash 2.5

Necessita un compte, però els seus preus eren significativament més assequibles al moment de començar amb la API. Gemini a part permetia 15 sol·licituds per minut, front a les 3 de **Chat GPT**. Amés **Chat GPT** requeria de tindre un compte de empresa amb un límit de tokens gastats, mentre que Gemini no. Per totes estes raons es va optar per **Gemini 2.5 Flash**.



## 3. PLANIFICACIÓ DEL PROJECTE

### 3.1 Temporalització

#### 0. Setmana 1 (21 - 27 de març) → Fase 0: Fase Prèvia

- Estudi d'alternatives, informació documental amb Gemini 2.0.
- Instal·lació i configuració d'eines (Laravel, MySQL, etc.).
- Creació del repositori en GitHub i planificació del projecte.

#### 1. Setmana 2 i 3 (28 de març - 10 d'abril) → Fase 1: Especificació i Disseny Previ

- Definició de requisits finals, abast i funcionalitats principals.
- Model entitat-relació (E/R) i diagrames UML.
- Prototip inicial de la interfície amb Figma.
- Disseny de les rutes i els controladors a Laravel.

#### 2. Setmana 4 i 5 (11 - 25 d'abril) → Fase 2: Desenvolupament del Backend (Autenticació i Base de Dades)

- Creació de models, migracions i relacions en MySQL
- Implementació del sistema d'autenticació amb Laravel Sanctum
- Desenvolupament de controladors i serveis per a la gestió d'usuaris

#### 3. Setmana 6 (26 d'abril - 2 de maig) → Fase 3: Integració de Gemini 2.0

- Configuració de l'API i gestió de tokens.
- Creació de la comunicació amb la IA.
- Emmagatzematge i processament de converses.

#### 4. Setmana 7 (3 - 9 de maig) → Fase 4: Implementació del Xat i WebSockets

- Desenvolupament de les vistes amb Blade, Tailwind i CSS.
- Implementació de xat en temps real.
- Connexió entre frontend i backend amb AJAX, pusher i echo.

**5. Setmana 8 (10 - 15 de maig) → Fase 5: Control i Gestió del Professor**

- Panell de professor amb funcions de supervisió.
- Funció de bloqueig/desbloqueig de la IA en temps real.
- Historial de converses per alumne (Falta vore el context holder de Gemini).

**6. Setmana 9 (16 - 20 de maig) → Fase 6: Seguretat, Optimització i Pujada.**

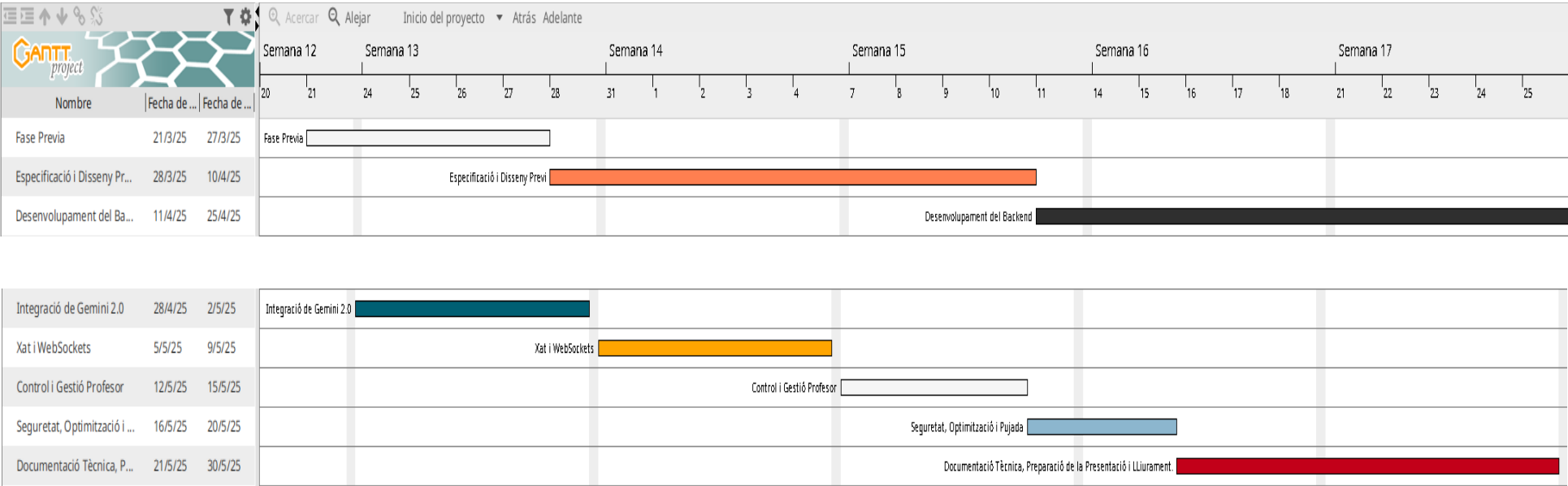
- Revisió de seguretat en l'API i autenticació.
- Optimització de rendiment i gestió de consultes.
- Correcció d'errors.
- Pujada a Github.

**7. Setmana 10, 11 i 12 (21 de maig - 1 de juny) → Fase 7: Documentació Tècnica, preparació de la presentació i lliurament.**

- Redacció de la documentació tècnica i manual d'usuari.
- Elaboració del Document a entregar i de la presentació.
- Lliurament.



3.2 Cronograma



## 4. DESENVOLUPAMENT DEL PROJECTE

En aquest apartat es descriu com s'ha anat creant EINA, pas a pas. Des de la fase prèvia fins a l'entrega, el desenvolupament ha anat evolucionant de forma iterativa, tornant enrere quan feia falta, provant funcionalitats i millorant-les quan es podia.

Cada fase descriu què s'ha fet, amb quines eines i per què s'han pres algunes decisions. També inclou captures, exemples de codi i explicacions visuals per entendre millor el procés.

### 4.0 Fase prèvia

Durant aquesta fase inicial s'ha posat en marxa tota l'estructura bàsica del projecte i s'ha decidit quina tecnologia s'anava a fer servir. També s'han començat a establir les primeres idees generals del projecte, així com l'organització del codi i el control de versions.

#### 4.0.1 Estudi d'alternatives

Inicialment, s'han valorat diversos tipus de projectes com ja s'ha exposat a la introducció. Però com ja s'ha exposat, Javier García va proposar la idea d'EINA. Hi han al mercat ferramentes paregudes com Duolingo Max o Botpress, però ninguna enfocada al entorn educatiu de classe a la educació pública.

L'objectiu ha sigut crear una aplicació que ajudara tant als alumnes com als professors a utilitzar la IA de forma educativa i controlada. Això també podia ajudar a perdre la por a emprar aquestes eines i, a més, oferir al professorat un mitjà per controlar com s'utilitza i amb quina finalitat.

#### 4.0.2 Instal·lació i configuració d'eines

Abans de començar el desenvolupament com a tal, s'han instal·lat totes les eines necessàries:



- Laravel HA12, amb PHP 8.3 i MySQL com a base de dades.
- Laravel Breeze amb Blade i Tailwind CSS.
- Vite com a bundler de JavaScript i CSS.
- Composer per a la gestió de dependències PHP.
- npm per a dependències de frontend com Tailwind i Laravel Echo.

A més, s'ha creat un entorn de desenvolupament local funcional i s'ha començat a preparar la base del projecte.

### 4.0.3 Creació del repositori GitHub

S'ha creat un repositori públic a GitHub per controlar les versions i facilitar la col·laboració. S'han fet commits freqüents per registrar cada canvi, tant de funcionalitats com d'estils, estructures de carpetes i proves. Això també ha ajudat a revisar errors i fer alguns canvis.

## 4.1 Especificació i disseny previ

Aquesta fase s'ha centrat en definir exactament què havia de fer l'aplicació, com havia de comportar-se i com seria la seua estructura tant visual com interna. A més, s'han documentat els requisits i s'han fet les primeres proves visuals i tècniques per garantir que el projecte fora viable.

### 4.1.1 Requisits i funcionalitats principals

S'ha establert una llista clara dels requisits essencials:

- Suport per a rols d'alumne i professor.
- Control de la IA per part del professor.
- Persistència de context amb Gemini 2.5.
- Visualització i supervisió de converses.
- Xat en temps real amb interaccions limitades per alumne.

### 4.1.2 Model E/R i diagrames UML

A partir dels requisits s'ha creat el model entitat-relació que inclou taules com usuaris, converses, missatges i contexts. També s'han fet diagrames UML per visualitzar les relacions entre models, especialment les dependències entre usuari-conversa-context. Els camps com email, rol, usuari\_id, context\_id, creat\_per, conversa\_id, actiu i gemini\_history\_id s'han marcat com a indexats per millorar el rendiment.



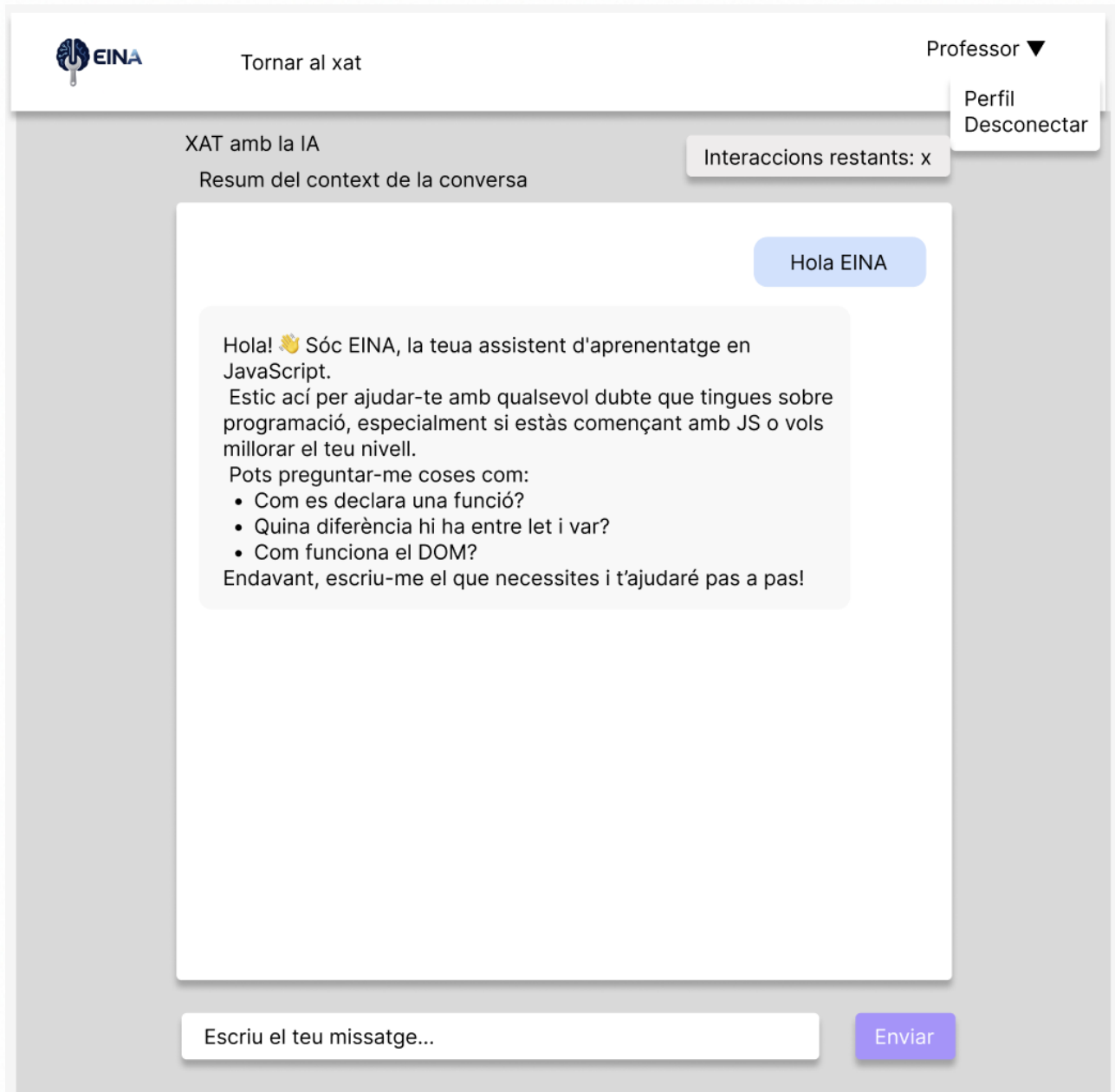
### 4.1.3 Prototip inicial amb Figma

Per a la part visual, s'ha utilitzat **Figma** per fer un esbós inicial de com seria el panell del professor i la vista del xat per a l'alumne. Aquest prototip ha ajudat a



identificar possibles problemes d'usabilitat i ha servit de base per a les primeres vistes Blade.

Xat del Alumne:



Aquesta vista es la del Xat del alumne, tota la interacció d'aquest amb l'aplicació serà a través d'aquesta vista i del perfil d'usuari. Hi haurà un camp per a introduir els missatges de l'alumne i la IA contestara al xat.

The screenshot displays the 'Panell del professor' (Professor Panel) for the EINA project. The header includes the EINA logo, the panel title, a link to 'Configuració de contextos', and a user profile dropdown for 'Professor' with options for 'Perfil' and 'Desconnectar'.

**Configuració activa**

- Context actiu
- Nom del context: WebEINA
- Descipció Curta: Professor de HTML y CSS
- Interaccions màximes: 7
- Configuracions IA

**Llista d'alumnes**

Alumne1	27-03-2026	16:35
Alumne2	27-03-2026	17:13

**Chat Window:**

Hola EINA

Hola! 🙌 Sóc EINA, la teua assistent d'aprenentatge en JavaScript. Estic ací per ajudar-te amb qualsevol dubte que tingues sobre programació, especialment si estàs començant amb JS o vols millorar el teu nivell.

Pots preguntar-me coses com:

- Com es declara una funció?
- Quina diferència hi ha entre let i var?
- Com funciona el DOM?

Endavant, escriu-me el que necessites i t'ajudaré pas a pas!

Aquest será el panell de configuració del professor. Aquí podrà veure el context actiu, seleccionar l'alumne i veure tant la seua conversa actual com les anteriors.

#### 4.1.4 Disseny de rutes i controladors a Laravel

S'han començat a planificar les rutes de Laravel seguint la lògica RESTful i s'han estructurat els controladors com `ConversaController`, `MissatgeController`,



ContextController i PanellProfessorController, així com alguns serveis com GeminiService. Aquesta estructura s'ha anat refinant al llarg del projecte.

### Algunes rutes a web.php

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use Illuminate\Support\Facades\Auth;
5  use App\Http\Controllers\ProfileController;
6  use App\Http\Controllers\ContextController;
7  use App\Http\Controllers\ConversaController;
8  use App\Http\Controllers\ConfiguracioIAController;
9  use App\Http\Controllers\GeminiController;
10 use Illuminate\Support\Facades\Broadcast;
11
12 /*Redirecció inicial segons rol*/
13 Route::get('/', function () {
14     return redirect()->route('login'); // Mostra login si no autenticat
15 });
16
17 Route::get('/redirect-segons-rol', function () {
18     $usuari = Auth::user();
19
20     if ($usuari->rol === 'professor') {
21         return redirect()->route('configuracio.index');
22     }
23
24     if ($usuari->rol === 'alumne') {
25         return redirect()->route('alumne.chat');
26     }
27
28     abort(403, 'Rol desconegut');
29 }->middleware(['auth', 'verified']->name('home')); // Laravel Breeze usa aquesta per defecte
30
31 /* Perfil d'usuari (Breeze) */
32 Route::middleware('auth')->group(function () {
33     Route::get('/profile', [ProfileController::class, 'edit']->name('profile.edit'));
34     Route::patch('/profile', [ProfileController::class, 'update']->name('profile.update'));
35     Route::delete('/profile', [ProfileController::class, 'destroy']->name('profile.destroy'));
36 });
37
38 /*Panell del professor (configuració IA)*/
39 Route::middleware(['auth'])->group(function () {
40     //Route::get('/panell-professor', [ConfiguracioIAController::class, 'index']->name('configuracio.index'));
41     Route::get('/panell-professor', [ConversaController::class, 'panell']->name('configuracio.index'));
42     Route::get('/panell-professor/conversa/{id}', [ConversaController::class, 'carregarPerProfessor'])
43     ->name('professor.carregarConversa');
44     Route::get('/panell-professor/{id}/edit', [ConfiguracioIAController::class, 'edit']->name('configuracio.edit');
45     Route::put('/panell-professor/{id}', [ConfiguracioIAController::class, 'update']->name('configuracio.update');
46     Route::post('/converses/{conversa}/actualitzar-context', [ConversaController::class, 'actualitzarContext'])
47     ->name('converses.actualitzarContext');
48     Route::get('/panell-professor/conversa-id/{id}', [ConversaController::class, 'mostrarConversaPerId']);
49 });
50
51 /*Gestió de contextos*/
52 Route::middleware(['auth'])->group(function () {
```

Algun controlador de ContextController.php:

```
10 class ContextController extends Controller
11 {
12     public function __construct()
13     {
14         $this->middleware('auth');
15
16         $this->middleware(function ($request, $next) {
17             if (auth()->user()->rol !== 'professor') {
18                 abort(403, 'Accés només per a professors');
19             }
20             return $next($request);
21         }->except('show'));
22     }
23
24     public function create()
25     {
26         return view('contextes.create');
27     }
28     public function index()
29     {
30         $this->autoritzarProfessor();
31
32         $context_actiu = ContextClasse::where('actiu', true)
33             ->where('creat_per', auth()->id())
34             ->first();
35
36         return view('panell-professor.index', compact('context_actiu'));
37     }
38     public function store(Request $request)
39     {
40         $request->validate([
41             'titol' => 'required|string|max:255',
42             'descripcio' => 'nullable|string',
43             'interaccions_max' => 'nullable|integer|min:1'
44         ]);
45
46         ContextClasse::create([
47             'titol' => $request->input('titol'),
48             'descripcio' => $request->input('descripcio'),
49             'interaccions_max' => $request->input('interaccions_max', 10),
50             'actiu' => false,
51             'creat_per' => auth()->id(),
52         ]);
53
54         return redirect()->back()->with('success', 'Context creat exitosament');
55     }
56 }
```



## Part de GeminiService.php:

```
1 <?php
2
3 namespace App\Services;
4
5 use Illuminate\Support\Facades\Http;
6 use Illuminate\Support\Facades\Log;
7
8 class GeminiService
9 {
10     protected string $endpoint = 'https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash-preview-04-17:generateContent';
11
12     // Enviar el missatge amb l'historial i opcional historyId
13     public function enviarMissatge(array $historial, string $apiKey, ?string $historyId = null): array
14     {
15         $payload = [
16             'contents' => $historial,
17         ];
18
19         if ($historyId) {
20             $payload['history'] = [
21                 'id' => $historyId,
22             ];
23         }
24
25         $response = Http::post($this->endpoint . '?key=' . $apiKey, $payload);
26
27         if ($response->successful()) {
28             $data = $response->json();
29
30             return [
31                 'text' => $data['@var mixed $data content']['parts'][0]['text'] ?? 'Error de resposta',
32                 'history_id' => $data['history']['id'] ?? null
33             ];
34         }
35
36         Log::error('Gemini API error', [
37             'status' => $response->status(),
38             'body' => $response->body()
39         ]);
40
41         return [
42             'text' => 'Error de comunicació amb Gemini',
43             'history_id' => null
44         ];
45     }
46
47     // Inicie una conversa i recupere un history_id explícitament
48     public function iniciarConversa(string $context, string $apiKey): ?string
49     {
50         $response = Http::post($this->endpoint . '?key=' . $apiKey, [
51             'contents' => [[
```

## 4.2 Desenvolupament backend

En aquesta fase s'ha centrat a construir tota la lògica del servidor, les connexions amb la base de dades i la gestió de la informació entre les diferents entitats. S'ha assegurat que les relacions entre models estiguen ben definides i que es puguin recuperar fàcilment des de les vistes.

### 4.2.1 Creació de models, migracions i relacions

S'han generat les migracions per a les taules usuaris, contexts, converses i missatges, definint clarament les relacions entre elles amb claus foranes.

- Cada **usuari** pot tindre diverses converses i si es professor pot crear diversos contextos.
- Cada **conversa** està associada a un context i un usuari.
- Cada **missatge** pertany a una conversa concreta.
- Els **contexts** defineixen el comportament de la IA i poden ser reutilitzats per diversos alumnes.

Aquestes relacions també s'han declarat als models Eloquent per a permetre consultes dinàmiques i senzilles des del backend.

### 4.2.2 Autenticació amb Laravel Sanctum



Per a la gestió de sessions i seguretat s'ha implementat Laravel Breeze amb Laravel Sanctum. El sistema d'autenticació inclou gestió de rols (alumne i professor) i protecció de rutes mitjançant middleware específics per a cada tipus d'usuari.

### 4.2.3 Controladors i serveis per a la gestió d'usuaris

S'han creat controladors responsables de gestionar:

- La creació i visualització de converses.
- L'enviament i registre de missatges.
- La visualització i gestió de contextos.

També s'ha desenvolupat un servei anomenat GeminiService, que enumerem a continuació. Aquest servei es l'encarregat de centralitzar totes les peticions a l'API de Gemini. D'aquesta manera, si es realitzen canvis a la configuració de l'API o a la lògica de missatges, només cal modificar aquest component.

## 4.3 Integració De Gemini 2.5

Durant aquesta fase s'ha integrat l'assistent d'intel·ligència artificial de Google anomenat Gemini 2.5 Flash. L'objectiu ha sigut permetre que cada alumne pugui conversar amb la IA dins d'un context educatiu, establert prèviament pel professor, mantenint una comunicació coherent i segura.

### 4.3.1 Configuració de l'API

S'ha configurat la connexió amb l'API de Gemini mitjançant una clau d'API definida al fitxer .env. Per centralitzar i aïllar la lògica de comunicació, s'ha creat un servei (GeminiService) que gestiona tant la construcció de la petició com el processament de la resposta.

Gemini 2.5 permet utilitzar identificadors d'historial (historyId) per mantindre la coherència en la conversa. Aquesta funcionalitat s'ha aprofitat per a reduir la quantitat de dades enviades i per a donar una experiència més fluida a l'alumne.

### 4.3.2 Comunicació amb la IA

Cada vegada que un alumne envia un missatge, es construeix l'historial complet de la conversa i s'envia a l'API. La resposta es desa automàticament com un nou missatge, associat a la conversa en curs.

A més, es controla el nombre d'interaccions restants en cada conversa. Quan aquest valor arriba a zero, es bloqueja temporalment l'enviament de nous missatges fins que el professor active un nou context o augmente el límit.

### 4.3.3 Emmagatzematge i processament de converses

Les converses es guarden a la base de dades amb informació com el context\_id, el gemini\_history\_id i les interaccions restants. Totes les respostes de la IA es registren a la taula de missatges, amb el remitent ia.



```

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('missatges', function (Blueprint $table) {
            $table->id();
            $table->foreignId('conversa_id')->constrained('converses')->onDelete('cascade')->index();
            $table->enum('remitent', ['alumne', 'ia']);
            $table->text('cos');
            $table->timestamps();
        });
    }
}

```

You, 3 weeks ago • Primer Commit EINA

També s'ha implementat un sistema bàsic de validació per evitar duplicats, truncaments o errors de format en les respostes de l'assistent.

```

// Funció per a parsejar format
function parseGeminiFormat(text) {
    return text
        .replace(/\*(.*?)\*/g, "<strong>$1</strong>")
        .replace(/\n/g, "<br>");
}

```

You, 1 second ago • Uncommitted changes

```

// Cree la instancia
Missatge::create([
    'conversa_id' => $conversa->id,
    'remitent' => 'alumne',
    'cos' => $nouMissatge,
]);

// Guarde la resposta
$missatgeNovaIA = Missatge::create([
    'conversa_id' => $conversa->id,
    'remitent' => 'ia',
    'cos' => $resposta['text'],
]);

// Emet l'esdeveniment per broadcast
event(new NouMissatgeAlumne($missatgeNovaIA));

```

## 4.4 IMPLEMENTACIÓ DEL XAT I WEBSOCKETS

Aquesta fase s'ha centrat a construir la funcionalitat del xat entre l'alumne i la IA. Ha sigut una de les parts més importants del projecte, ja que ha requerit la integració entre backend i frontend, així com la prova de tecnologies en temps real com Laravel Echo i Pusher. També s'ha treballat en l'estètica i estructura de les vistes per a fer el xat còmode i intuïtiu tant per a l'alumnat com per al professorat.

### 4.4.1 Desenvolupament inicial de les vistes

#### 4.4.1.1 Configuració inicial amb Blade i Tailwind

S'han creat les primeres vistes Blade com `chat.blade.php` per als alumnes i `index.blade.php` dins del panell del professor. S'ha utilitzat Tailwind CSS per estructurar les interfícies i permetre un disseny responsive. També s'ha separat el disseny en components reutilitzables.

#### 4.4.1.2 Estils bàsics amb CSS

S'han afegit estils propis mitjançant una fulla `custom.css` per ajustar colors, gradients, transicions i elements visuals com botons o el panell del xat. També s'ha aplicat un sistema de mode fosc amb canvis dinàmics a través de JavaScript.

### 4.4.2 Implementació del xat amb Livewire/AJAX

Tot i que inicialment es va considerar Livewire, finalment s'ha optat per AJAX pur amb JavaScript per enviar missatges i rebre respostes de Gemini. Aquest sistema ha permès integrar el control de la conversa, actualitzar el DOM amb les respostes i mantindre la interacció fluïda sense recarregar la pàgina.

### 4.4.3 Connexió frontend-backend

El fitxer `chat.js` s'encarrega de gestionar els esdeveniments de formulari, enviar les peticions POST al controlador `GeminiController` i actualitzar la vista amb la resposta. També controla el nombre d'interaccions restants i bloqueja l'enviament



si s'esgoten. S'ha garantit la connexió fluida entre frontend i backend amb gestió d'errors i feedback visual.

#### 4.4.4 Ampliació i refinament de les vistes

##### 4.4.4.1 Vistes específiques d'alumnes

S'ha dissenyat una vista simplificada per a l'alumnat, on es mostra únicament el xat i les interaccions restants. Aquesta vista s'ha optimitzat perquè siga clara, lleugera i usable des de qualsevol dispositiu.

##### 4.4.4.2 Vistes específiques del professor

Al panell del professor s'ha afegit un sistema per seleccionar un alumne i veure les seues converses en temps real, incloent missatges i context actiu. També s'ha afegit la possibilitat de bloquejar l'accés a la IA o canviar el context actiu.

#### 4.4.5 Implementació Broadcasting amb Laravel Echo i Pusher

Per a la comunicació en temps real s'ha utilitzat Laravel Echo amb Pusher. S'ha configurat `broadcasting.php` per usar el canal privat `conversa.{id}` i `xat-alumne.{id}`. S'han emés esdeveniments com `NouMissatgeAlumne` i `ContextCanviat` per actualitzar la vista del professor i la del xat de l'alumne de forma instantània. També s'ha provat amb Laravel WebSockets (BeyondCode), però es va descartar per problemes amb Laravel 12.

### 4.5 Control i gestió del professor

Durant aquesta fase s'han desenvolupat les funcionalitats específiques per al rol de professor. L'objectiu ha sigut oferir un panell des del qual es puga supervisar l'activitat dels alumnes, gestionar els contextos i bloquejar l'accés a la IA quan siga necessari.

### 4.5.1 Panell professor

S'ha creat una vista específica dins la carpeta panell-professor que actua com a panell de control. Aquesta vista permet:

- Veure la configuració activa de la IA i el context seleccionat.
- Visualitzar un llistat d'alumnes reals registrats a l'aplicació.
- Seleccionar un alumne per accedir al seu xat.
- Canviar a conversacions antigues dels alumnes segons les necessitats del professor.
- Controlar el número de interaccions dels alumnes per al context aplicat, pugen bloquejar-lo actualitzant el context a 0 interaccions.

Aquestes funcionalitats s'han implementat combinant rutes protegides amb middleware i components dinàmics que s'actualitzen amb AJAX o Laravel Echo.

### 4.5.2 Historial de converses per alumne

Quan el professor selecciona un alumne, es mostra a la dreta un panell amb totes les converses que ha tingut eixe alumne. Aquest llistat inclou:

- La data de creació de cada conversa.
- El context actiu que estava seleccionat en aquell moment.
- L'estat de la conversa (activa o arxivada).

També s'ha preparat la funcionalitat per carregar cada conversa individual amb els seus missatges, la qual cosa permet revisar de manera detallada què s'ha parlat i com ha interaccionat l'alumne amb la IA.



## 4.6 Seguretat, Optimització i Pujada

Aquesta fase ha tingut com a objectiu revisar la seguretat del projecte, millorar el seu rendiment i preparar-lo per a l'entorn de producció.

### 4.6.1 Revisió seguretat API i autenticació

S'ha revisat que totes les rutes sensibles estiguen protegides amb middleware d'autenticació (auth:sanctum) i que els rols d'alumne i professor tinguin accés exclusiu a les funcionalitats que els pertocuen. A més, s'ha aplicat control d'accés a nivell de controladors per evitar que un usuari interactue amb dades que no li corresponen.

S'han aplicat també bones pràctiques com l'ús de tokens segurs, validació estricta de les dades d'entrada i gestió d'errors personalitzada per evitar exposar informació interna.

### 4.6.2 Optimització rendiment

Per millorar el rendiment del sistema, s'han indexat els camps clau de la base de dades (usuari\_id, context\_id, creat\_per, conversa\_id, actiu, etc.). Això ha permès accelerar les consultes Eloquent i millorar la càrrega de les vistes que mostren informació dependent d'altres models.

A més, s'ha evitat la càrrega innecessària de relacions no usades i s'han afegit consultes amb with() per fer eager loading i evitar el problema N+1.

### 4.6.3 Correcció errors

Durant la fase de proves s'han detectat i corregit errors visuals, problemes amb el recompte d'interaccions, fallades en el broadcasting i errors en les rutes. També s'han corregit comportaments inesperats en el xat, com missatges duplicats o bloquejos quan s'esgotaven les interaccions.



S'ha utilitzat Laravel Telescope per ajudar a depurar i fer seguiment dels esdeveniments, respostes de l'API i comportament general de l'aplicació.

#### 4.6.4 Ajustos Finals.

Abans de pujar el projecte a producció, s'han aplicat diversos ajustos menors que han contribuït a polir l'experiència d'ús i garantir la consistència del sistema. Entre aquestes millores s'inclouen:

- **Sistema d'enviament de missatges** refinat, controlant millor el nombre d'interaccions i les condicions de bloqueig.
- **Traducció d'etiquetes i missatges d'interfície** al valencià, per garantir una coherència lingüística en tota l'aplicació.
- **Millores visuals en el mode fosc**, especialment en botons, scrolls i contrastos.
- **Corregits errors menors** com duplicats visuals, icones mal posicionades o vistes que no es carregaven bé.

Aquests canvis finals han sigut importants per oferir una aplicació sòlida, funcional i agradable de fer servir abans de la pujada definitiva.

#### 4.6.5 Pujada GitHub

Un cop completat el desenvolupament, s'ha pujat el projecte a GitHub com a còpia definitiva. El repositori inclou tota la documentació, fitxers .env, i scripts útils per posar en marxa l'entorn. Això permet tindre una còpia segura, fàcilment clonable i preparada per a possibles millores futures o presentacions.

Es pot accedir mitjançant el enllaços següents:

<https://projecteeina.es/>

<https://github.com/jpeiroaguado/Projecte-EINA>

## 4.7 Documentació tècnica

Aquesta mateixa documentació cobreix de manera detallada els aspectes tècnics del projecte: estructura, fases de desenvolupament, eines emprades i decisions preses. Per aquest motiu, no s'ha elaborat un manual d'usuari específic, ja que les funcionalitats es descriuen juntament amb les seues vistes i comportaments dins d'aquesta mateixa memòria, a part de que la complexitat d'ús no es alta.

# 5. INTERFÍCIE FINAL DE L'APLICACIÓ

## 5.1 Estils personalitzats i mode fosc

L'aplicació ha sigut dissenyada amb **Tailwind CSS** seguint una estètica pròpia i coherent, amb l'objectiu de garantir una bona experiència d'usuari. S'han aplicat els següents elements visuals destacats:

- **Tipografia Roboto** com a font principal.
- **Botons personalitzats** amb gradient i estils adaptats a mode clar i fosc.
- **Scrolls estilitzats**, formularis cuidats i missatges de sistema visuals.
- **Transicions suaus** per a canvis de tema i elements interactius.

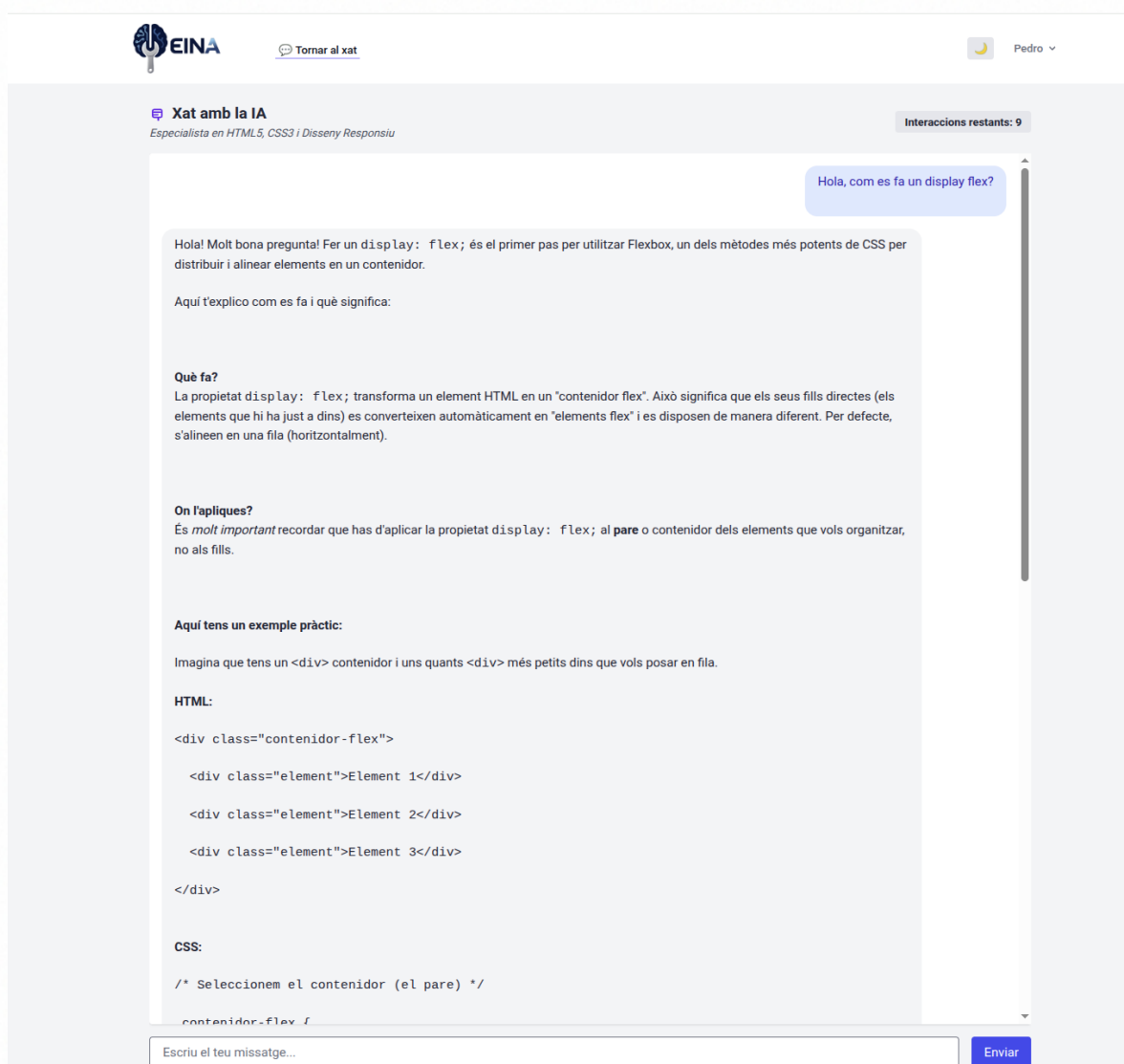


El mode fosc ha sigut implementat per defecte, permetent un ús còmode en entorns amb poca llum i reforçant l'aspecte modern de la interfície.

## 5.2 Interfície final de l'aplicació

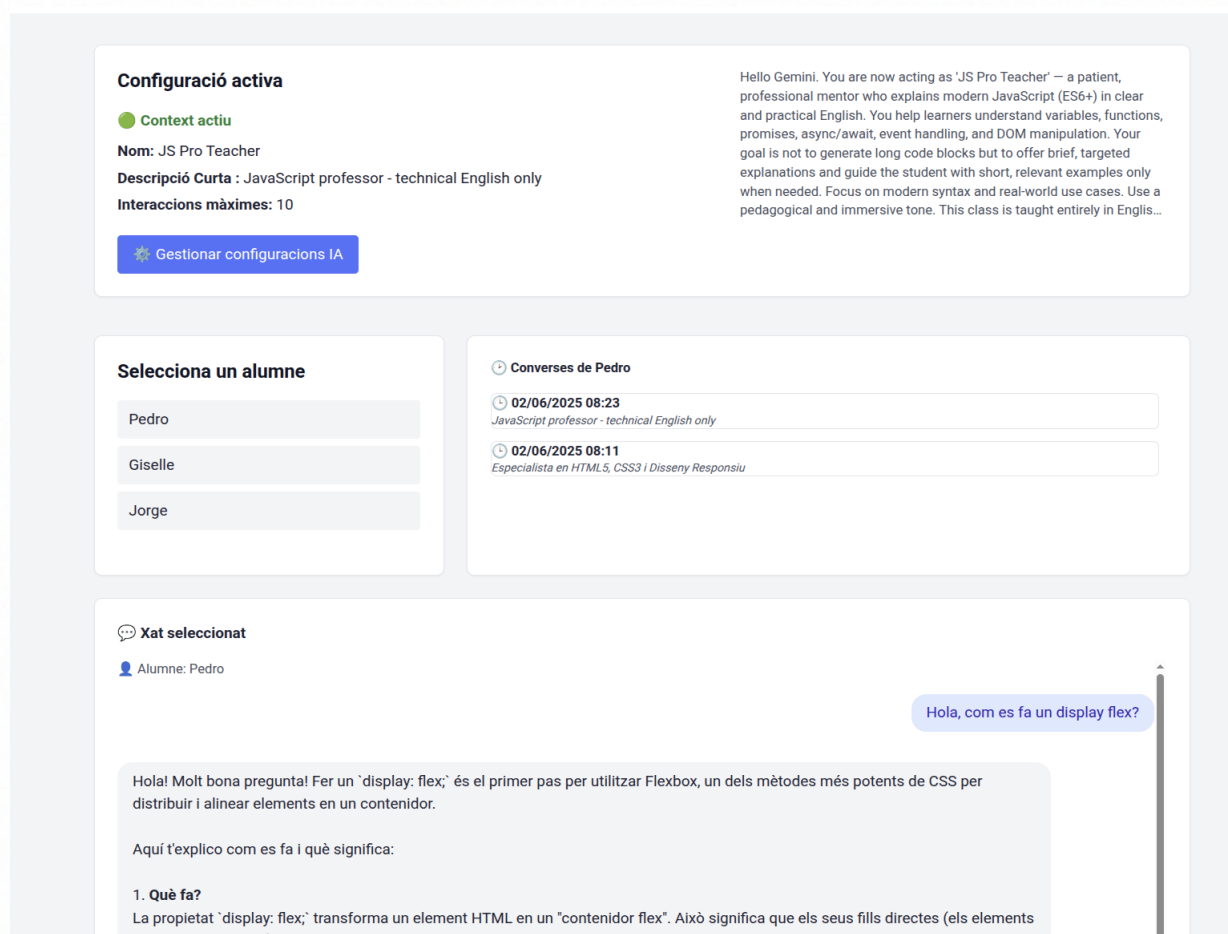
A continuació es mostren diverses **captures de la versió final de l'aplicació**:

- **Vista de l'alumne**, amb el xat en temps real amb la IA, el comptador d'interaccions i l'adaptació responsive.





- **Vista del professor**, amb el panell de control per a gestionar contextos, visualitzar converses i controlar l'activitat dels alumnes.



- Diferents seccions comunes amb **disseny consistent**, traduccions al valencià i elements clars i accessibles.

Aquestes pantalles representen l'aspecte real de l'aplicació en funcionament després del desplegament final.

## 6. BIBLIOGRAFIA

### 6.1 Llibres i vídeos

- [\*Tailwind CSS: From Zero to Production.\*](#)
- Vídeos de [Laravel Breeze](#) i [Laravel Echo](#).

### 6.2 Manuals web

- <https://laravel.com/docs> – Documentació oficial de Laravel.
- <https://tailwindcss.com/docs> – Guia completa de Tailwind CSS.
- <https://pusher.com/docs> – Manual per a integrar Pusher amb Laravel.
- <https://ai.google.dev/gemini-api/> Documentació oficial de Gemini API.
- <https://laravel-livewire.com/docs> – (opcional) per a proves amb Livewire.

### 6.3 Fòrums i blogs

- **Stack Overflow:** dubtes sobre Laravel, Tailwind i APIs.
- **GitHub Discussions** en els repositoris oficials de Laravel i Pusher.
- **Laravel.io** i **Laracasts Forum**.



## 7 ANÀLISI I VALORACIÓ D'ALTERNATIVES

### 7.1 Anàlisi i conclusions

#### 7.1.1 Objectius

Els objectius inicials del projecte EINA eren clars: desenvolupar una eina per a l'aula que permetera als alumnes comunicar-se amb una IA educativa dins d'un entorn controlat pel professorat. Aquesta IA havia de ser capaç de mantenir contextos persistents, limitar el nombre d'interaccions i adaptar-se a diferents rols pedagògics. A més, es buscava facilitar la supervisió docent i reduir la barrera d'entrada a la IA en el context educatiu.

#### 7.1.2 Conclusions

S'han aconseguit la gran majoria dels objectius marcats. El sistema implementat permet un ús real de la IA en l'aula, amb control del nombre d'interaccions, canvis de context, bloqueig de l'assistent i visió global del que fan els alumnes. A nivell tècnic, el projecte ha servit per posar en pràctica coneixements de Laravel, gestió d'API, WebSockets i integració amb serveis externs com Gemini.

#### 7.1.3 Autoavaluació

El projecte s'ha desenvolupat dins del temps previst, tot i haver tingut fases molt complicades com la configuració del broadcasting. La funcionalitat cobreix els requisits marcats i el resultat és usable. Tot i això, es podria millorar l'aspecte visual de certes vistes i optimitzar millor les consultes per a escenaris amb molts alumnes o converses. També deuria de formar-me millor amb tecnologies d'emissió d'esdeveniments i consumidors.

### 7.2 Millores o alternatives proposades

- Integrar OAuth per gestionar millor l'accés a l'API en lloc d'usar una clau pública.
- Implementar un panell d'estadístiques per al professorat.

- Millorar la usabilitat per a mòbils i afegir funcionalitats com cerca de converses.
- Explorar WebSockets propis en lloc de Pusher, si la compatibilitat amb Laravel millora.
- Al final, EINA es un prototip per demostrar el poder que pot tindre una aplicació controlant una IA a l'aula. Amb models propis o lliures, i un servidor web a la Generalitat que la contingui, les possibilitats son cassi infinites.

### 7.3 Agraïments

Gràcies al professor Eloy Gutiérrez Paredes pel seu suport constant i orientació durant tot el procés de desenvolupament. També a l'equip docent de DAW i a Javier García Sabater per aportar la idea i fer un 'brainstorming'. Sense aquest entorn col·laboratiu, aquest treball no hauria sigut el mateix.

I per últim agrair al meu tutor Kike Savall tota l'ajuda durant el curs, inclús abans...

### 7.4 Dificultats trobades

Al principi, s'ha creat funcions al controlador de GeminiController que després conforme ha anat evolucionant l'aplicació s'ha fet obvi que hi havia que **re-ubicar-les** a controladors més específics.

Un problema que s'ha trobat i que s'ha repetit durant el projecte ha sigut la **interoperabilitat del sistema**, amb els **servicis d'autorització** i els **serveis externs**. Moltes vegades l'aplicació fallava silenciosament.

Tots aquest problemes venen de que **Laravel 12** ha **perdut compatibilitat** amb WebSockets i s'ha de fer amb **Pusher i Echo**, el que dificulta el control de les cues d'esdeveniments.

S'ha sol·ventat fent funcions dins del servei de autorització al que deixaven accedir al professor amés de l'alumne, estant el professor identificat per a poder accedir als xats dels alumnes.



Els estils de **Tailwind** maceraven el CSS natiu que s'intentava inserir.

S'ha sol-ventat fent eixes modificacions directament al @layer utilites, que modifica els estils nadius de tailwind.

## 7.5 Relació amb els mòduls estudiats

Tots els moduls pràctics que he donat a aquest curs han sigut claus per a la creació d'aquest projecte.

**Disseny d'Interfícies Web:** Els estils de Tailwind CSS no deixen de ser una abreviatura de CSS. Tailwind treballa amb **css grid** i així ha sigut estructurada la aplicació visualment. Sense el coneixement de com funciona el grid, no haguera sigut possible treballar amb Tailwind ni tampoc tindre coneixements sobre els selectors interns d'aquest.

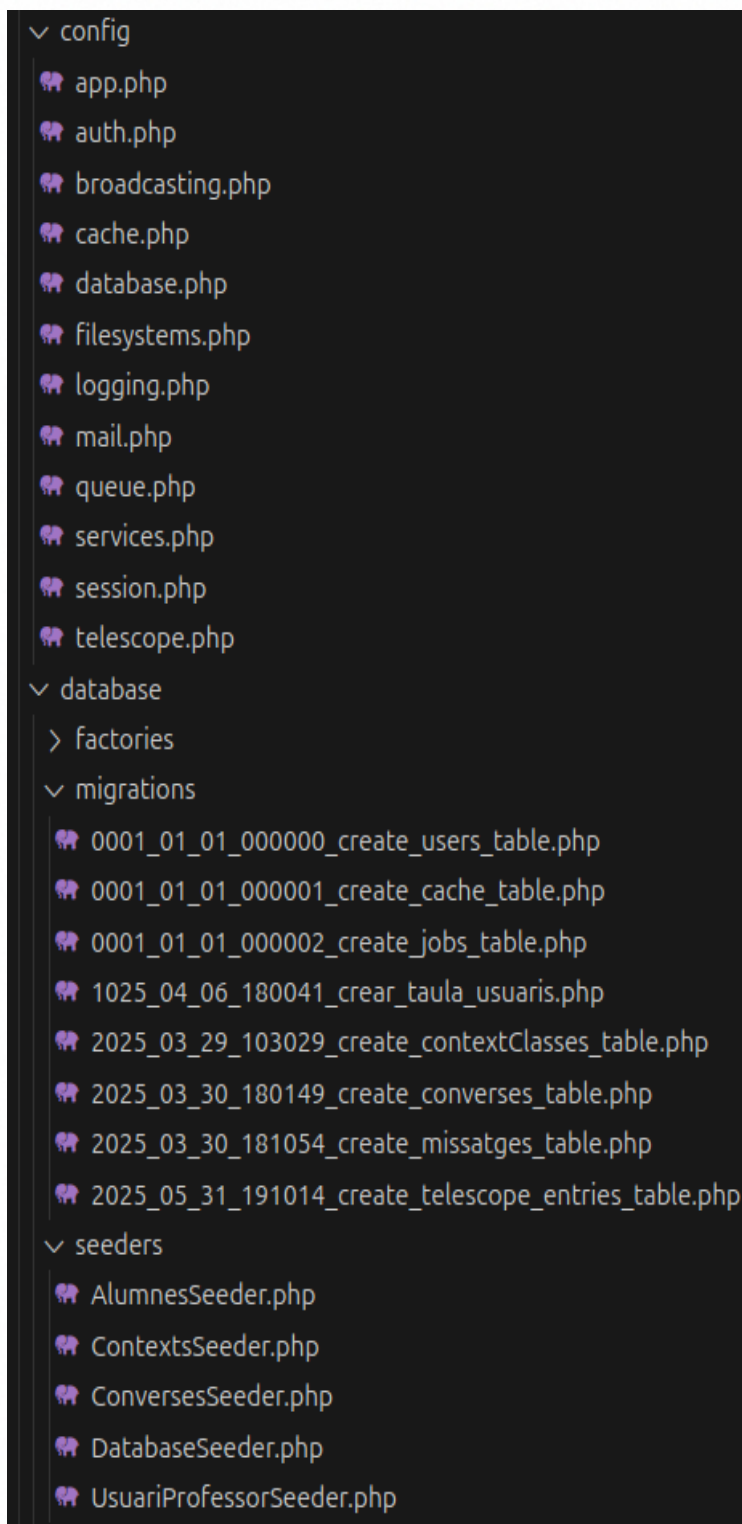
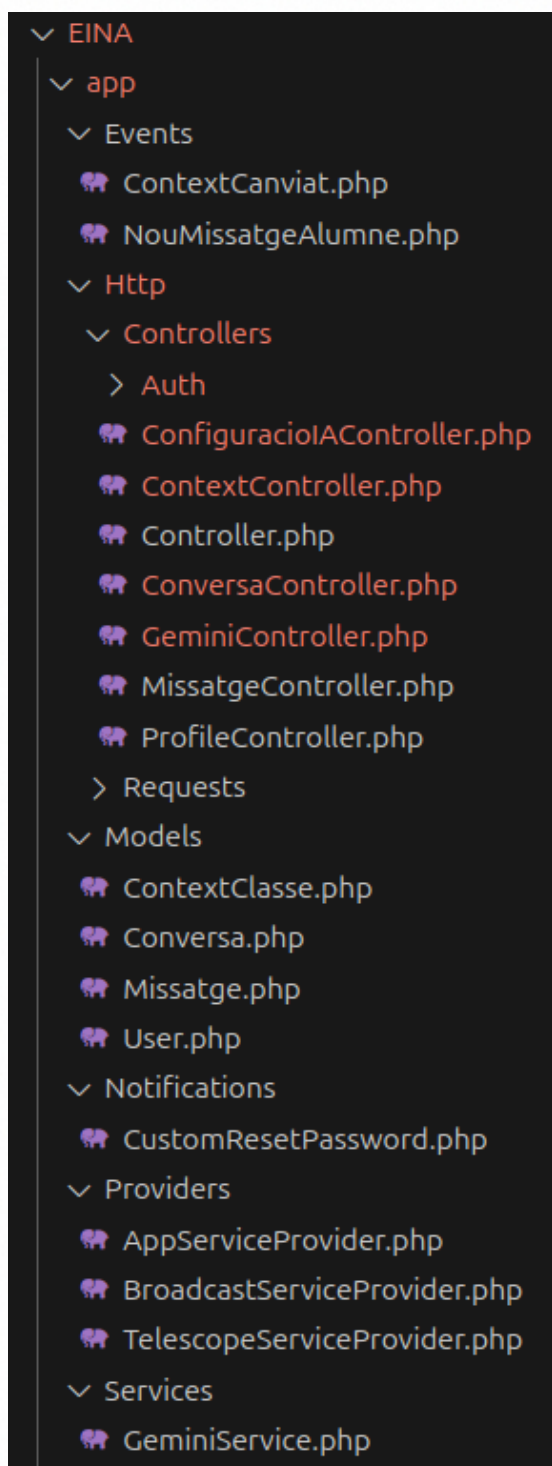
**Desplegament d'Aplicacions Web:** També ha sigut necessari ja que s'ha instal·lat un servidor Nginx amb certificats SSL i s'ha configurat tot l'entorn en local, per a després pujar-lo a producció.

**Desenvolupament Web en entorn Client:** Clau per al funcionament interactiu de l'aplicació i sobre tot de la comunicació amb Gemini, gràcies a la experiència obtinguda en interaccions amb API/Json ha sigut un element molt important del projecte.

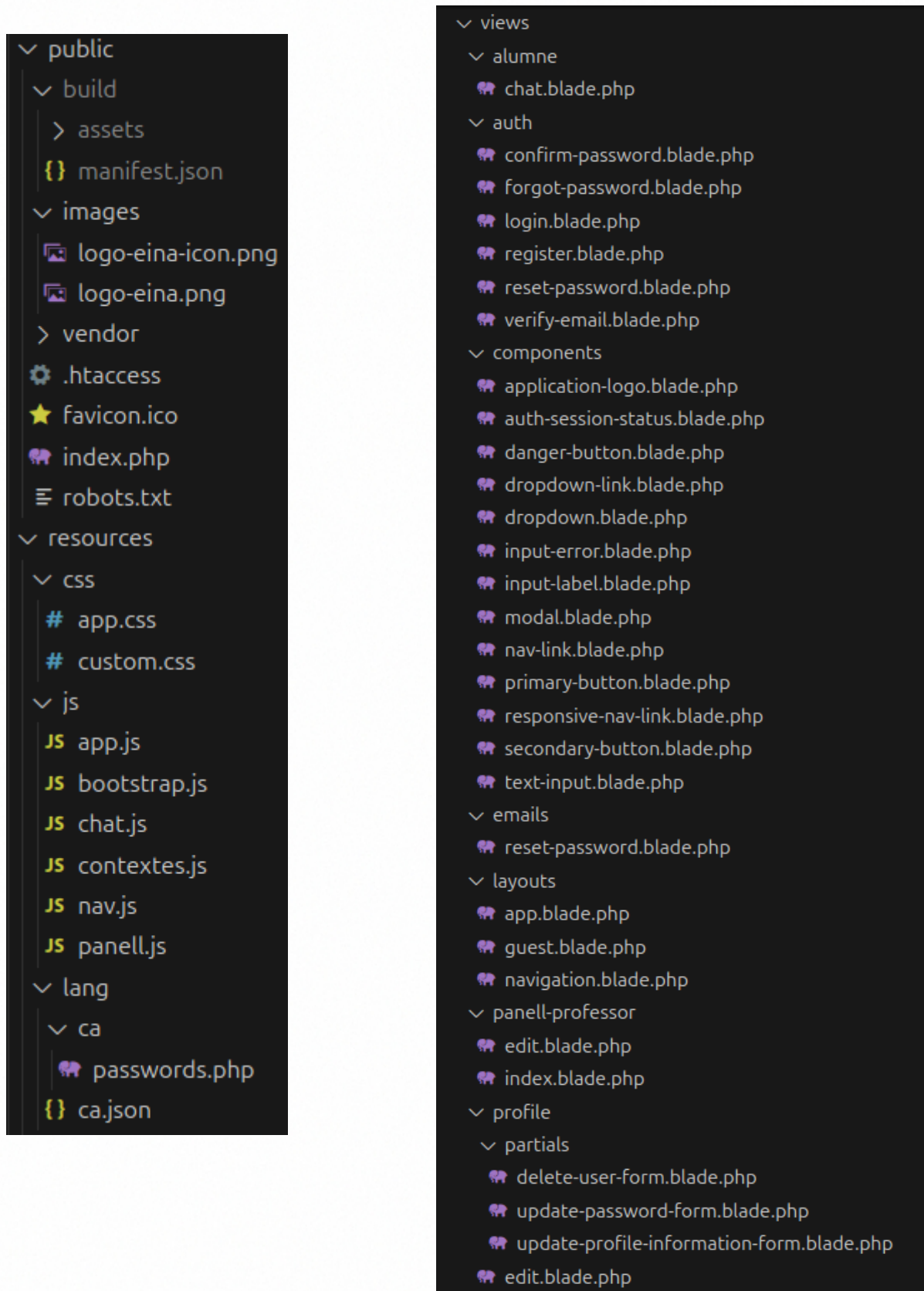
**Desenvolupament WEB en entorn servidor:** La columna vertebral de l'aplicació. Construït amb LARAVEL ha sigut el mòdul més destacat a aquest projecte. Controladors, Rutes, Models, Taules, Vistes i un llarg etc. Controlar tots els elements ha sigut desafiant, però sense aquest coneixement no haguera sigut possible EINA.

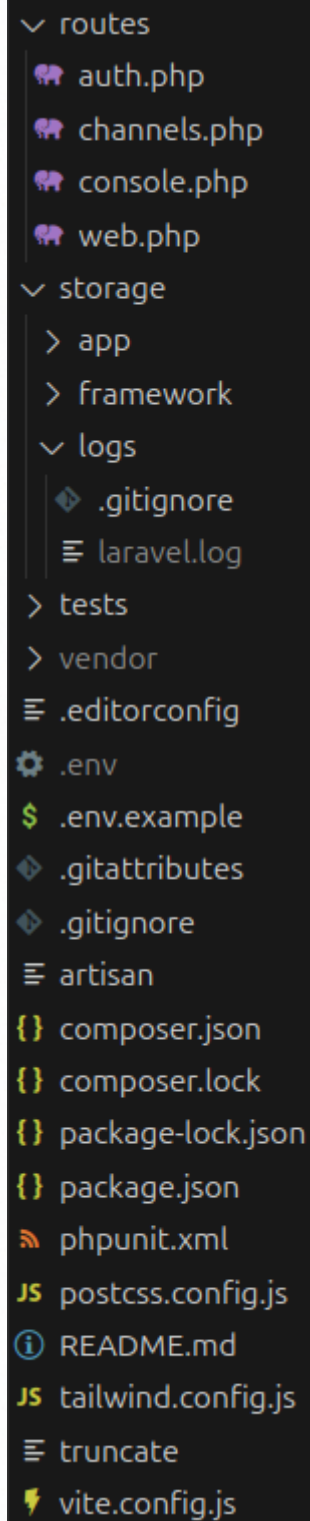
## 8 ANNEXOS

### 8.1 Estructura de carpetes









A screenshot of a file explorer window showing the directory structure of a Laravel project. The files and folders are listed with their respective icons: a folder icon for directories and various file icons (elephant for PHP, gear for environment files, dollar sign for example files, diamond for git files, notepad for config, and lightning bolt for vite config). The structure is as follows:

- routes
  - auth.php
  - channels.php
  - console.php
  - web.php
- storage
  - app
  - framework
- logs
  - .gitignore
  - laravel.log
- tests
- vendor
- .editorconfig
- .env
- .env.example
- .gitattributes
- .gitignore
- artisan
- composer.json
- composer.lock
- package-lock.json
- package.json
- phpunit.xml
- postcss.config.js
- README.md
- tailwind.config.js
- truncate
- vite.config.js



## 8.2 Diagrames

### 8.2.1 Diagrama E/R

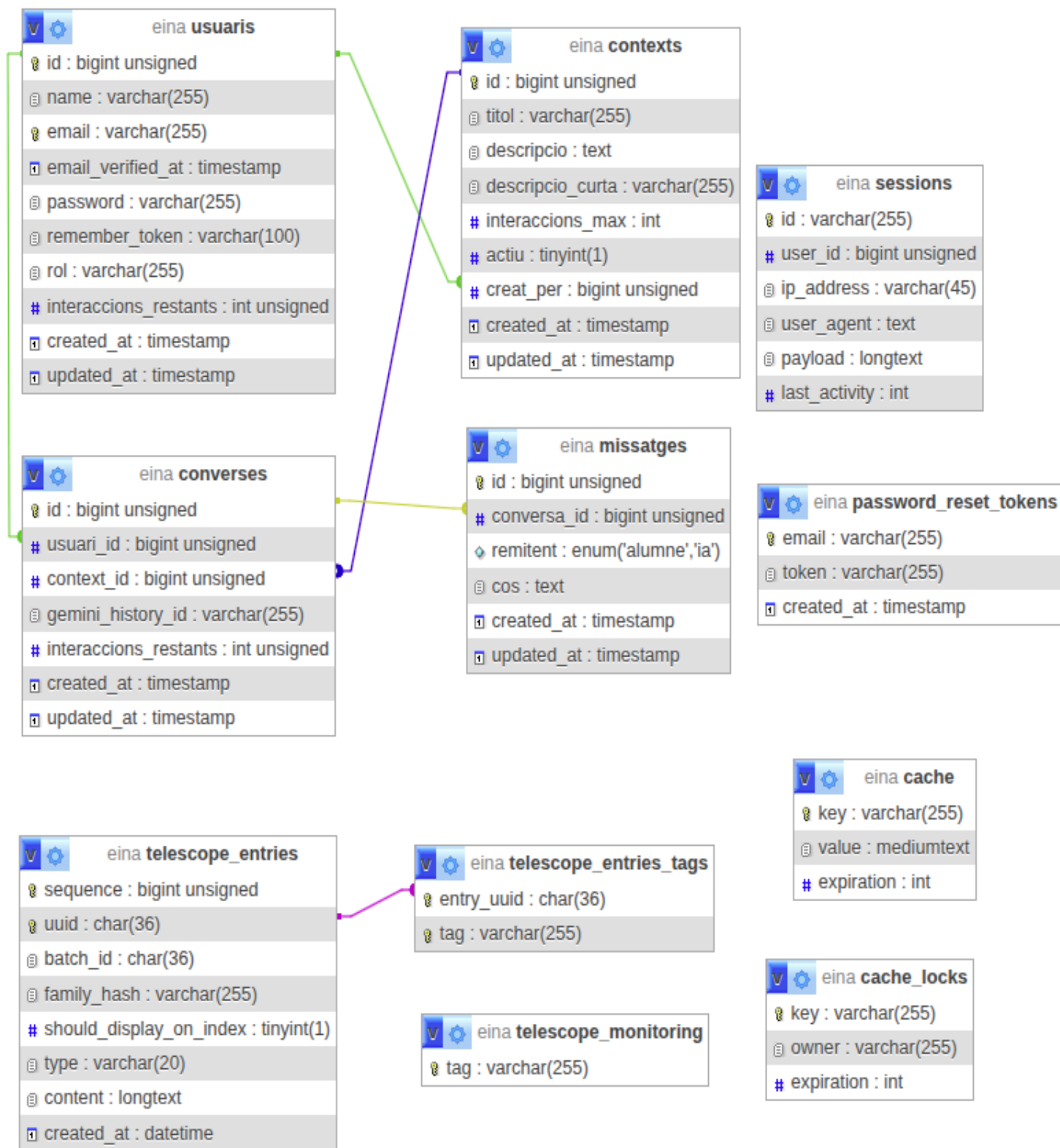
El model Entitat-Relació del projecte EINA està compost per quatre taules principals: usuaris, contexts, converses i missatges. A continuació es descriu cada taula i la seua relació amb les altres:

- **usuaris**
  - Conté la informació bàsica de cada usuari registrat: nom, email (únic), contrasenya i rol (alumne o professor).
  - Disposa d'un camp `interaccions_restants`, tot i que esta funcionalitat finalment s'ha gestionat des del model converses.
  - Relacions:
    - Un usuari pot **crear múltiples contextos** (`contexts.creat_per`).
    - Un usuari pot **tindre múltiples converses** (`converses.usuari_id`).
  - Indexats:
    - email (per a login).
    - rol (per a gestió de permisos i vistes).
- **contexts**
  - Representa un context pedagògic definit pel professor, amb un títol, descripció completa i resumida.
  - Inclou el nombre màxim d'interaccions per conversa (`interaccions_max`) i un booleà per marcar si està actiu (`actiu`).
  - Relacions:
    - Cada context **pertany a un usuari (professor)** mitjançant `creat_per`.
    - Pot estar **associat a múltiples converses** (`converses.context_id`).
  - Indexats:
    - actiu (per recuperar ràpidament el context actual).
    - creat\_per (per filtrar els contextos d'un professor).
- **converses**

- Representa una conversa entre un alumne i la IA.
- Emmagatzema la referència al context usat (context\_id), l'usuari que la comença (usuari\_id), i el gemini\_history\_id per mantenir el context persistent amb Gemini.
- També registra el nombre d'interaccions restants.
- Relacions:
  - Cada conversa **pertany a un usuari i a un context**.
  - Pot contindre **molts missatges** (missatges.conversa\_id).
- Indexats:
  - usuari\_id, context\_id, gemini\_history\_id.
- **missatges**
  - Conté els missatges que formen part d'una conversa.
  - Guarda el cos del missatge i qui és el remitent (alumne o ia).
  - Relacions:
    - Cada missatge **pertany a una conversa**.
  - Indexat:
    - conversa\_id.

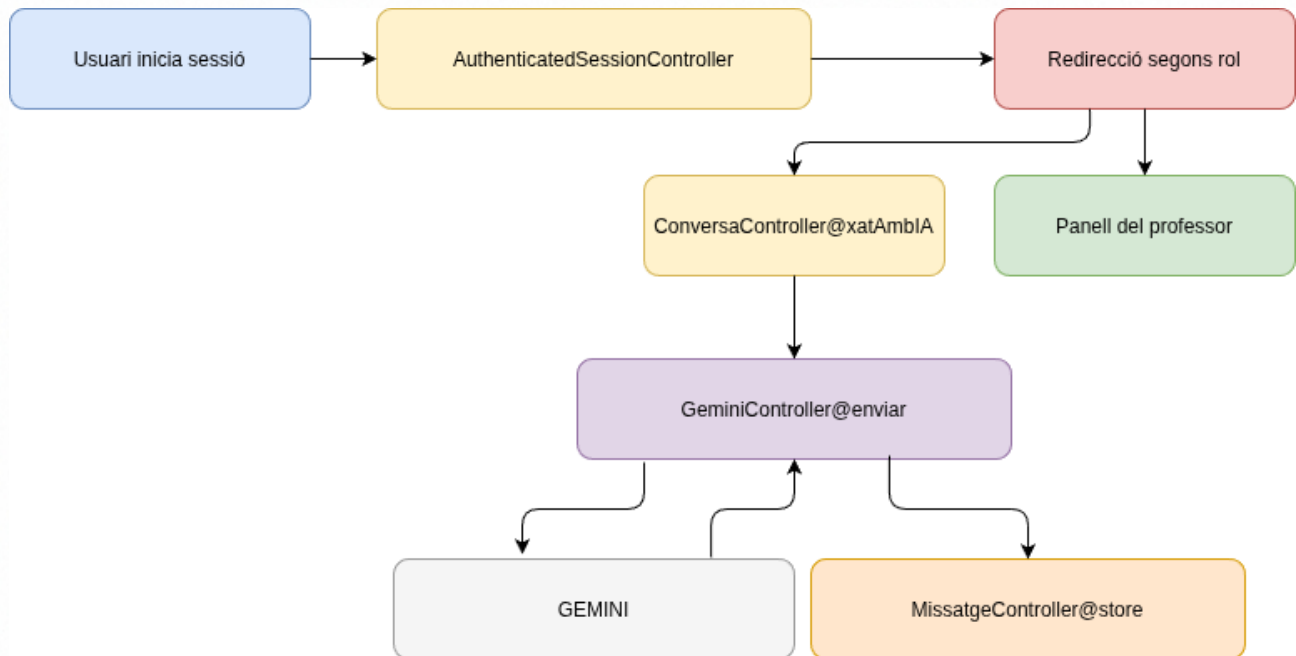


## 8.2.1 Diagrama E/R



•

### 8.2.2 Diagrama de classes:



### 8.2.3 Diagrama de Flux de l'aplicació:

