



UNIVERSITAT DE  
BARCELONA

Treball de Final de Grau

**Projecte GOLEM XIV Versió 3.0:**  
**Millora d'Arquitectura, Noves**  
**Funcionalitats i Nova Interfície d'Usuari**

**Autor:**

Sergio Marín Herrera

**Director:**

Dr. Daniel Ortiz Martínez

**Realitzat a:**

Departament de Matemàtiques i Informàtica

**Facultat de Matemàtiques i Informàtica**  
**Universitat de Barcelona**

Barcelona, 10 de juny de 2025

# Índex

---

|                                            |           |
|--------------------------------------------|-----------|
| <b>Resum</b>                               | <b>4</b>  |
| <b>Resumen</b>                             | <b>5</b>  |
| <b>Abstract</b>                            | <b>7</b>  |
| <b>1. Introducció</b>                      | <b>8</b>  |
| 1.1. Motivació                             | 9         |
| 1.2. Objectius                             | 9         |
| 1.2.1 Objectius Funcionals                 | 10        |
| 1.2.1 Objectius Personals                  | 11        |
| <b>2. Anàlisi</b>                          | <b>12</b> |
| 2.1. Requisits                             | 12        |
| 2.1.1. Requisits Funcionals                | 12        |
| 2.1.1. Requisits No Funcionals             | 14        |
| 2.2. Backlog                               | 16        |
| <b>3. Arquitectura de l'aplicació</b>      | <b>24</b> |
| 3.1. Backend                               | 25        |
| 3.1.1. FastAPI                             | 25        |
| 3.1.2. SQLModel                            | 27        |
| 3.1.1. Alembic                             | 28        |
| 3.2. Base de dades                         | 29        |
| 3.2.1. PostgreSQL                          | 29        |
| 3.3. Sistema de Processament de Pràctiques | 30        |
| 3.3.1. RabbitMQ                            | 31        |
| 3.3.2. Worker                              | 32        |
| 3.3.3. Servidor RPC                        | 33        |
| 3.4. Emmagatzematge d'arxius               | 35        |
| 3.4.1. Servidor SFTP                       | 35        |
| 3.5. Frontend                              | 36        |
| 3.5.1. Next.js                             | 36        |
| 3.5.2. TailwindCSS                         | 37        |
| 3.5.3. HeroUI                              | 38        |
| 3.6. Arquitectura Distribuïda al Núvol     | 38        |
| 3.6.1. Backend                             | 39        |
| 3.6.2. Base de Dades                       | 40        |
| 3.6.3. Worker                              | 41        |
| 3.6.4. Servidor RPC                        | 41        |

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| 3.6.5. Frontend                                                       | 41        |
| 3.6.6. Sistema de Missatgeria                                         | 42        |
| 3.6.7. Servidor SFTP                                                  | 43        |
| <b>3.7. Desplegament Continu</b>                                      | <b>43</b> |
| 3.7.1. Components automatitzats en el CD                              | 43        |
| <b>4. Implementació i resultats</b>                                   | <b>45</b> |
| <b>4.1. Implementació del Backend</b>                                 | <b>45</b> |
| 4.1.1. Reestructuració de l'arquitectura del projecte                 | 45        |
| 4.1.2. Canvi d'ORM: de SQLAlchemy a SQLModel                          | 47        |
| 4.1.4. Unificació i millora de l'estructura de la base de dades       | 48        |
| 4.1.5. Millora del sistema autorització                               | 50        |
| 4.1.6. Gestió de fitxers via SFTP                                     | 51        |
| 4.1.7. Endpoints i funcionalitats implementades                       | 52        |
| 4.1.6. Gestió de dependències                                         | 56        |
| <b>4.2. Implementació del Worker</b>                                  | <b>56</b> |
| 4.2.1. Refactorització i processament concurrent                      | 56        |
| 4.2.2. Comunicació amb el backend i el servidor RPC                   | 57        |
| 4.2.3. Gestió asíncrona de base de dades                              | 58        |
| 4.2.4. Controls de fiabilitat amb RabbitMQ                            | 58        |
| 4.2.5. Gestió de dependències                                         | 60        |
| <b>4.3. Implementació del Frontend i Experiència d'Usuari (UI/UX)</b> | <b>60</b> |
| 4.3.1. Estructura de navegació i arquitectura visual                  | 61        |
| 4.3.2. Pàgina principal (Dashboard)                                   | 62        |
| 4.3.3. Pàgina de cursos                                               | 64        |
| 4.3.4. Pàgina detall d'un curs                                        | 65        |
| 4.3.5. Pàgina de pràctiques                                           | 67        |
| 4.3.6. Pàgina detall d'una pràctica                                   | 67        |
| 4.3.7. Gestió de cursos i pràctiques                                  | 69        |
| 4.3.7. Calendari de pràctiques                                        | 71        |
| 4.3.8. Configuració de l'usuari                                       | 72        |
| 4.3.9. Sistema de notifikacions en temps real                         | 73        |
| 4.3.10. Millora global de l'experiència d'usuari (UX)                 | 75        |
| <b>4.4. Resultats</b>                                                 | <b>78</b> |
| 4.4.1. Millora de l'experiència d'usuari (UI/UX)                      | 78        |
| 4.4.2. Rendiment del sistema de correcció                             | 78        |
| 4.4.3. Fiabilitat i estabilitat del sistema                           | 79        |
| 4.4.4. Accessibilitat i portabilitat                                  | 79        |
| 4.4.5. Resum dels resultats obtinguts                                 | 80        |
| <b>5. Conclusions i treball futur</b>                                 | <b>81</b> |
| <b>5.1. Conclusions</b>                                               | <b>81</b> |

|                                    |    |
|------------------------------------|----|
| 5.2. Treball futur                 | 82 |
| 6. Bibliografia                    | 84 |
| Apèndix A. Repositori de codi font | 86 |

## Resum

---

Al llarg dels darrers anys, la integració de la tecnologia en l'àmbit educatiu ha permès automatitzar processos que tradicionalment requerien una elevada càrrega manual. Un exemple d'això és la correcció de pràctiques de programació, una tasca repetitiva i costosa per als docents. Aquest Treball de Fi de Grau presenta Golem XIV, una plataforma web que, en la seva tercera versió, proporciona una solució completa per a la gestió i autocorrecció d'aquestes pràctiques, facilitant molt la feina del professorat i oferir als alumnes una resposta més ràpida.

L'aplicació permet als professors pujar els fitxers d'autocorrecció específics de cada pràctica i, posteriorment, els estudiants poden enviar les seves solucions a través del sistema. Així, la creació d'aquesta eina no només simplifica la tasca per als professors, sinó que també millora l'experiència d'aprenentatge dels estudiants, proporcionant un entorn més dinàmic i interactiu.

Una de les prioritats d'aquesta versió del projecte ha estat renovar completament la interfície d'usuari i experiència d'usuari per fer-la més clara, intuïtiva i agradable d'utilitzar. Tant els estudiants com els professors poden consultar de forma visual el progrés de les pràctiques, accedir a un calendari amb els terminis d'entrega, nous panells de control pel professorat i rebre notificacions quan hi ha canvis en l'estat d'alguna pràctica. Així com, una pàgina principal en format tauler per tenir la informació important recollida en un sol lloc.

Així mateix, s'ha refactoritzat l'estructura interna de l'aplicació per fer-la més estable i fàcil de mantenir en el futur gràcies a la seva modularitat. El sistema s'ha distribuït en diferents serveis per funcionar de manera més eficient i ha estat desplegat al núvol perquè es pugui accedir-hi des de qualsevol lloc.

En definitiva, Golem XIV no només optimitza el procés de correcció, sinó que transforma l'experiència educativa, oferint una eina moderna, eficient i alineada amb les necessitats actuals del món acadèmic.

## Resumen

---

A lo largo de los últimos años, la integración de la tecnología en el ámbito educativo ha permitido automatizar procesos que tradicionalmente requerían una elevada carga manual. Un ejemplo de esto es la corrección de prácticas de programación, una tarea repetitiva y costosa para los docentes. Este Trabajo de Fin de Grado presenta Golem XIV, una plataforma web que, en su tercera versión, proporciona una solución completa para la gestión y autocorrección de estas prácticas, facilitando mucho el trabajo del profesorado y ofrecer a los alumnos una respuesta más rápida.

La aplicación permite a los profesores subir los ficheros de autocorrección específicos de cada práctica y, posteriormente, los estudiantes pueden enviar sus soluciones a través del sistema. Así, la creación de esta herramienta no solo simplifica la tarea para los profesores, sino que también mejora la experiencia de aprendizaje de los estudiantes, proporcionando un entorno más dinámico e interactivo.

Una de las prioridades de esta versión del proyecto ha sido renovado completamente la interfaz de usuario y experiencia de usuario para hacerla más clara, intuitiva y agradable de utilizar. Tanto los estudiantes como los profesores pueden consultar de forma visual el progreso de las prácticas, acceder en un calendario con los plazos de entrega, nuevos paneles de control por el profesorado y recibir notificaciones cuando hay cambios en el estado de alguna práctica. Así como, una página principal en formato tablero para tener la información importante recogida en un solo lugar.

Así mismo, se ha refactorizado la estructura interna de la aplicación para hacerla más estable y fácil de mantener en el futuro gracias a la su modularidad. El sistema se ha distribuido en diferentes servicios para funcionar de manera más eficiente y ha sido desplegado a la nube porque se pueda acceder desde cualquier lugar.

En definitiva, Golem XIV no solo optimiza el proceso de corrección, sino que transforma la experiencia educativa, ofreciendo una herramienta moderna, eficiente y alineada con las necesidades actuales del mundo académico.

## Abstract

---

In recent years, the integration of technology in education has enabled the automation of traditionally manual tasks. One clear example is the grading of programming assignments, a repetitive and time-consuming job for educators. This Final Degree Project introduces Golem XIV, a web platform which, in its third version, offers a comprehensive solution for managing and auto-grading such assignments, significantly easing the instructors' workload and providing students with faster feedback.

The platform allows teachers to upload custom correction files for each assignment, and students to submit their solutions directly through the system. This tool not only simplifies the teachers' tasks, but also enhances the students' learning experience by offering a more dynamic and interactive environment.

A major focus in this version has been the complete redesign of the user interface and user experience, making it clearer, more intuitive, and pleasant to use. Both students and teachers can visually track progress, access calendars with deadlines, use new dashboards, and receive real-time notifications when there are updates to assignment statuses. A central dashboard consolidates essential information in one place.

Additionally, the application's internal structure has been refactored for improved stability and maintainability thanks to its modular architecture. The system has been distributed into separate services to maximize efficiency and deployed in the cloud to ensure global accessibility.

Ultimately, Golem XIV not only optimizes the correction process but also transforms the educational experience by providing a modern, efficient tool aligned with today's academic needs.

# 1. Introducció

---

En l'actualitat, l'ensenyament de la programació es troba en una etapa de constant evolució, i amb ella, els processos d'avaluació i correcció de les pràctiques dels estudiants. Un dels principals reptes als quals s'enfronten els docents és la càrrega de treball associada a la correcció manual d'aquestes pràctiques, un procés que pot ser tediós, lent i propens a errors. La necessitat d'automatitzar aquesta tasca i augmentar l'eficiència de l'avaluació és, doncs, més evident que mai.

Aquest Treball de Fi de Grau se centra en la tercera versió de Golem XIV, una plataforma web desenvolupada amb l'objectiu de facilitar la gestió i autocorrecció de pràctiques de programació. En aquesta nova versió, no només s'han ampliat i perfeccionat les funcionalitats existents [1], sinó que també s'ha redissenyat la interfície i s'ha refactoritzat tota l'arquitectura tècnica del sistema per millorar-ne la mantenibilitat i escalabilitat.

Una de les prioritats d'aquest projecte ha estat implementar una arquitectura moderna basada en serveis desacoblats, millorant la modularitat del backend i introduint un sistema de processament concurrent de les pràctiques. Paral·lelament, s'ha desenvolupat una nova interfície amb tecnologies com Next.js, HeroUI i TailwindCSS, orientada a millorar l'experiència d'usuari tant per a estudiants com per a professors.

Aquest TFG analitza els canvis introduïts en el sistema, les decisions de disseny adoptades i les tecnologies escollides per donar resposta als reptes plantejats. A més, s'estableixen les bases per a la seva evolució futura com a eina educativa eficient i escalable, contribuint d'aquesta manera a la millora dels processos d'ensenyament i aprenentatge en l'àmbit de la programació.

## **1.1. Motivació**

La motivació principal per abordar aquest projecte ha estat la voluntat d'anar més enllà de l'automatització funcional de les tasques, incorporant una capa d'interacció visual atractiva i accessible per a l'usuari. Sovint, les eines educatives es desenvolupen amb una visió tècnica molt enfocada a la funcionalitat, deixant de banda l'experiència d'usuari.

Amb la tercera versió de Golem XIV, he volgut capgirar aquest enfocament, dotant la plataforma d'una interfície moderna, intuïtiva i agradable que faciliti la navegació i interacció dels usuaris. Tant estudiants com professors es beneficien d'un entorn visualment clar i organitzat, que redueix la corba d'aprenentatge i afavoreix l'eficiència en l'ensenyament i el seguiment acadèmic.

Aquesta motivació està reforçada per la necessitat de millorar l'arquitectura interna de l'aplicació, inicialment creada amb una estructura monolítica poc escalable. Refactoritzar el backend, introduir concurrència i desplegar l'aplicació com a sistema distribuït ha estat clau per garantir la seva evolutivitat. Amb aquestes millores, s'ha buscat no només una solució funcional, sinó un producte robust, escalable i preparat per adaptar-se a les necessitats futures del context educatiu.

## **1.2. Objectius**

Els objectius d'aquest Treball de Fi de Grau se centren en el desenvolupament d'una plataforma web moderna i escalable per a la gestió i autocorrecció de pràctiques de programació. A nivell tècnic, ha estat buscar una millora substancial del sistema tant pel que fa al rendiment com a la qualitat de la interfície. Paral·lelament, el projecte suposa una oportunitat d'adquirir nous coneixements i aprofundir en tecnologies i pràctiques avançades de desenvolupament. Per això, els objectius s'han dividit en dues categories: funcionals i personals.

### 1.2.1 Objectius Funcionals

- **Redisseny de la interfície d'usuari:**  
Redissenyar completament la interfície d'usuari amb tecnologies modernes com Next.js, TailwindCSS i HeroUI, millorant-ne la usabilitat i consistència visual.
- **Notificacions en temps real:**  
Incorporar un sistema de notificacions en temps real per a estudiants mitjançant Supabase Realtime, amb notificacions visibles a la interfície (Toast, Popover i Badge) i canvis visuals en la navegació.
- **Refactorització del backend:**  
Reestructurar l'arquitectura del backend per fer-lo modular, net i escalable, usant FastAPI, SQLAlchemy i separació clara de capes (configuració, models, serveis, API), millorant la llegibilitat i mantenibilitat del codi.
- **Unificació de base de dades:**  
Simplificar l'estructura de la base de dades, passant de dues instàncies a una sola basada en PostgreSQL, per centralitzar, optimitzar la gestió de dades i eliminant redundàncies.
- **Millora del processament de pràctiques:**  
Millorar el sistema de processament de pràctiques mitjançant concurrència amb asyncio i aio\_pika, permetent gestionar diversos missatges simultàniament en funció dels nuclis del processador.
- **Integració amb CloudAMQP:**  
Desplegar la cua de missatges RabbitMQ a CloudAMQP per a la comunicació entre serveis de manera fiable, escalable i distribuïda.
- **Servidor SFTP propi:**  
Configurar un servidor SFTP en una màquina virtual Linux a Azure per gestionar els arxius relacionats amb les entregues de les pràctiques i poder integrar el backend amb el servidor de correcció de pràctiques.
- **Desplegament distribuït amb CD:**  
Distribuir la plataforma en diversos serveis desplegats en el núvol (Azure, Supabase, Vercel), amb desplegament continu (CD), incloent-hi les migracions de base de dades.

### 1.2.1 Objectius Personals

- **Millora de l'experiència d'usuari:**  
Enfocar el projecte cap a la millora de l'experiència d'usuari, prioritzant la usabilitat i el disseny d'interfície sense perdre de vista la funcionalitat.
- **Bones pràctiques de desenvolupament:**  
Aprendre a estructurar i mantenir un projecte complex usant bones pràctiques de programació i arquitectura de software.
- **Treball amb stack modern:**  
Treballar amb tecnologies modernes de frontend i backend per ampliar coneixements en entorns reals de desenvolupament.
- **Aprentatge de React i Next.js:**  
Dominar el framework de React mitjançant Next.js, adoptant les seves millors pràctiques.
- **Implementació de RabbitMQ:**  
Aprendre i implementar RabbitMQ com a sistema de missatgeria entre serveis, entenent la seva arquitectura basada en cues i consumidors.
- **Ús de Docker:**  
Utilitzar Docker per a la contenerització d'aplicacions i facilitar-ne el desplegament, testatge i portabilitat.
- **Coneixement d'Azure:**  
Conèixer i experimentar amb els serveis de Microsoft Azure per al desplegament i monitoratge d'aplicacions en el núvol.
- **Sistema distribuït i CI/CD:**  
Adquirir experiència en l'organització d'un sistema distribuït i en la gestió del desplegament automatitzat.
- **Construcció d'una eina educativa:**  
Desenvolupar una solució útil, escalable i amb impacte real en el context de l'ensenyament de la programació.

## 2. Anàlisi

---

A l'hora de modificar una aplicació web ja existent, resulta essencial fer una anàlisi acurada per identificar les àrees que requereixen millora, definir nous requisits i establir les bases per a un desenvolupament estructurat i coherent. Aquesta fase és clau per assegurar que les decisions tècniques i de disseny es prenen amb una visió clara del context i de les necessitats dels usuaris finals. A més, facilita la creació d'un pla de treball amb un backlog ben definit, que guiarà tot el procés de desenvolupament i ajudarà a reduir imprevistos.

### 2.1. Requisits

#### 2.1.1. Requisits Funcionals

Els requisits funcionals especifiquen què ha de fer exactament l'aplicació perquè respongui a les necessitats dels seus usuaris. En aquest projecte, es tractava de transformar i enriquir una aplicació ja operativa, amb un èmfasi en la millora de la usabilitat i la funcionalitat de la plataforma.

1. **Gestió de fitxers d'autocorrecció:** Permetre als professors pujar els fitxers d'autocorrecció específics de cada pràctica.
2. **Seguiment d'estat de pràctiques:** Mostrar a estudiants i professors l'estat de cada pràctica com: *No entregada*, *Entregada*, *Corregint-se*, *Corregida*, *Rebutjada*.
3. **Enviament de pràctiques per part d'estudiants en format ZIP:** Permetre als estudiants enviar la seva pràctica en format ZIP a través de la plataforma.
4. **Tornar a tramitar una pràctica:** Permetre tornar a tramitar una pràctica si aquesta es troba amb l'estat de corregida o rebutjada.
5. **Pàgina d'inici amb format Dashboard:** Dashboard per a estudiants i professors amb informació adaptada: cursos recents, calendari setmanal navegable, i taula amb pràctiques pròximes.
6. **Sistema de notifikacions en temps real:** Mostrar canvis en l'estat de les pràctiques via Toasts, i una llista de notifikacions dintre d'un Popover activat per un botó amb icona de campana a la barra superior (Navbar),

que a la seva vegada conté un Badge amb el nombre de notificacions que té l'usuari.

- 7. Vista de calendari:** Calendari setmanal amb pràctiques com a esdeveniments clicables amb popover informatiu i accés directe a la pràctica.
- 8. Gestió de cursos, pràctiques i alumnes:** Permetre al professor crear, editar i eliminar els seus cursos i pràctiques, i gestionar els seus alumnes. També permetre a l'administrador fet tot el mencionat abans però de manera global.
- 9. Visualització estructurada de pràctiques a la pàgina de pràctiques:** Organitzar les pràctiques per estat en una vista amb pestanyes.
- 10. Detall de pràctiques i feedback:** Accés al detall d'una pràctica, visualització del feedback rebut i poder descarregar la pràctica tramitada.
- 11. Filtres i cercadors contextuais:** Implementar filtres i sistemes de cerca a les llistes de cursos, pràctiques i alumnes.
- 12. Barra lateral (SideNav):** Navegació lateral amb seccions "General" (Dashboard, Cursos, Pràctiques, Calendari) i "Configuració" (Configuració, Tanca sessió). Possibilitat d'expandir o encongir la barra de navegació lateral amb un botó.
- 13. Barra superior (Navbar) :** Incloure cerca global (amb selector de context), mode fosc/clar, botó de notificacions, i informació de l'usuari (avatar, nom, email) a la navbar.
- 14. Drag & Drop:** Sistema de pujada d'arxius amb funcionalitat d'arrossegar i deixar (Drag & Drop) a totes les seccions on sigui necessari.
- 15. Navegació amb breadcrumbs:** Rastre de navegació amb jerarquia visual (ex. Dashboard > Cursos).
- 16. Visualització de targetes (cards):** Cursos i pràctiques representats amb targetes visuals amb informació rellevant, barra de progrés, i accés a accions com editar (segons permisos).
- 17. Accés a entregues dels estudiants:** El professor pot consultar les entregues de cada alumne des de la vista de pràctica.

### 2.1.1. Requisites No Funcionals

Els requisits no funcionals defineixen les característiques del sistema que no estan directament relacionades amb les funcionalitats específiques, sinó amb aspectes generals com el seu rendiment, la seguretat, la fiabilitat i l'experiència de l'usuari. Aquests requisits estableixen les condicions i les limitacions que influiran en la qualitat global del sistema.

1. **Concurrencia i rendiment:** El sistema ha de permetre el processament simultani d'entregues sense pèrdua de rendiment.
2. **Arquitectura modular:** El backend, frontend i serveis han d'estar estructurat per mòduls separats i clars que facilitin el manteniment i l'escalabilitat.
3. **Desplegament distribuït:** Backend desplegat a Azure Web App, frontend a Vercel, base de dades a Supabase, cues RabbitMQ a CloudAMQP i servidor SFTP a màquina virtual de Linux a Azure.
4. **Temps de resposta eficient:** El sistema ha de respondre de manera ràpida, especialment en processos intensius com la correcció de pràctiques.
5. **Contenerització amb Docker:** Tots els serveis han d'estar empaquetats com a contenidors per facilitar la portabilitat i el desplegament, menys el frontend.
6. **Seguretat i permisos:** Gestió d'autenticació i control d'accés d'usuari segons rol, tant per a accions com per a visualització de dades.
7. **Notificacions sincronitzades:** Fiabilitat i rapidesa en l'enviament d'alertes a l'usuari a través de la integració amb Supabase Realtime.
8. **Middleware aplicat:** Fer ús del middleware per garantir una gestió eficient d'autenticació, control d'accés, i comportaments comuns en peticions al frontend.
9. **Nomenclatura estandarditzada en anglès:** Tot el projecte ha d'estar refactoritzat per utilitzar de manera consistent la nomenclatura en anglès. Això inclou noms de taules i columnes a la base de dades, variables, funcions, serveis i qualsevol altra entitat dins del codi. Aquesta uniformitat

facilita la col·laboració, mantenibilitat del codi i l'alineació amb bones pràctiques de desenvolupament internacional.

- 10. Protecció en la comunicació entre serveis:** S'han implementat mesures per garantir la seguretat en la transmissió de dades entre serveis com el backend, el worker i el servidor de correcció de pràctiques mitjançant cues de missatges (RabbitMQ), evitant exposicions innecessàries i afavorint un intercanvi de dades controlat.
- 11. Escalabilitat mitjançant microserveis:** Gràcies a la separació entre frontend, backend i altres serveis, el sistema ha de permetre l'ampliació o millora d'elements individuals sense afectar el conjunt.
- 12. Millora de la usabilitat:** S'ha de treballar específicament a millorar l'experiència de l'usuari final, oferint una interfície intuïtiva per facilitar la navegació o la gestió en general.
- 13. Rendiment estable i entorns aïllats:** L'ús de Docker assegura un comportament previsible en diversos entorns i redueix problemes derivats de configuracions inconsistentes.
- 14. Integració fluida amb serveis interns:** El sistema ha d'estar dissenyat per interactuar de forma robusta amb l'eina de correcció automàtica, la base de dades, els canals de notificació i altres serveis complementaris de forma coherent i extensible.

## 2.2. Backlog

Un “backlog” és una llista ordenada de tasques, funcionalitats o elements que cal desenvolupar o completar durant un projecte desglossat en forma de *user stories* que expressen el valor que aporten als usuaris. Cada entrada inclou una descripció funcional concreta i uns criteris d'acceptació que permeten verificar-ne la correcta implementació. Aquest enfocament permet prioritzar el desenvolupament i estructurar les tasques de manera incremental, seguint bones pràctiques de desenvolupament àgil.

| User Story | Descripció                                                                             | Criteris d'acceptació                                                                                                                                                                                                                                                               |
|------------|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>US1</b> | Com a professor, vull poder pujar fitxers d'autocorrecció per pràctica                 | <ul style="list-style-type: none"> <li>- Els fitxers es pugen correctament.</li> <li>- Els fitxers queden vinculats a una pràctica específica.</li> <li>- L'usuari rep una confirmació visual de la càrrega.</li> <li>- Si hi ha un error es mostra un missatge d'error.</li> </ul> |
| <b>US2</b> | Com a estudiant, vull poder entregar una pràctica per obtenir una correcció automàtica | <ul style="list-style-type: none"> <li>- L'arxiu es desa correctament.</li> <li>- L'estat de la pràctica canvia a “Entregada”.</li> <li>- El sistema encola la pràctica per a correcció i retorna resposta.</li> <li>- Si hi ha un error es mostra un missatge d'error.</li> </ul>  |
| <b>US3</b> | Com a usuari, vull accedir a una pàgina amb tots els cursos als quals tinc accés       | <ul style="list-style-type: none"> <li>- Es mostra una graella targetes de cursos amb informació rellevant (títol, descripció, curs acadèmic, llenguatges de programació, nombre d'estudiants, progrés, avatars de professors).</li> </ul>                                          |

|            |                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |                                                                            | <ul style="list-style-type: none"> <li>- Cada targeta de curs inclou un botó per accedir al detall i, si és professor del curs/administrador, també un botó per editar.</li> <li>- Si l'usuari és administrador, es mostren tots els cursos del sistema.</li> <li>- Si és estudiant o professor, només es mostren els cursos als quals està assignat.</li> </ul>                                 |
| <b>US4</b> | Com a usuari assignat a un curs, vull veure la informació general del curs | <ul style="list-style-type: none"> <li>- Es mostra el nom, descripció, curs acadèmic, semestre i professor responsable.</li> <li>- Es permet accedir a les pràctiques i participants del curs.</li> </ul>                                                                                                                                                                                        |
| <b>US5</b> | Com a usuari assignat a un curs, vull veure quines pràctiques té un curs   | <ul style="list-style-type: none"> <li>- Es mostra el llistat de pràctiques del curs.</li> <li>- Es pot filtrar per estat, buscar per títol i ordenar per data.</li> <li>- Cada targeta mostra llenguatge de programació, estat i data límit.</li> <li>- Cada targeta de pràctica inclou un botó per accedir al detall i, si és professor del curs/administrador, un botó per editar.</li> </ul> |
| <b>US6</b> | Com a usuari assignat a un curs, vull veure quins participants té un curs  | <ul style="list-style-type: none"> <li>- Es mostra la llista completa de participants del curs.</li> <li>- Es poden filtrar per rol (Tots, Professors, Estudiants).</li> <li>- Es pot cercar un usuari concret.</li> </ul>                                                                                                                                                                       |

|             |                                                                                       |                                                                                                                                                                                                                                                                       |
|-------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                                                       | <ul style="list-style-type: none"> <li>- És possible contactar amb un participant per correu.</li> </ul>                                                                                                                                                              |
| <b>US7</b>  | Com a usuari assignat a una pràctica, vull veure la informació general de la pràctica | <ul style="list-style-type: none"> <li>- Es mostra la descripció, estat, dates, documents adjunts i enllaços rellevants.</li> <li>- Si l'usuari és estudiant, pot veure el seu estat i accedir a la seva entrega.</li> </ul>                                          |
| <b>US8</b>  | Com a professor, vull veure totes les entregues d'una pràctica                        | <ul style="list-style-type: none"> <li>- Es mostra la llista d'estudiants.</li> <li>- En clicar un estudiant, es mostra informació de l'entrega: estat, data, arxius, i opcions de descàrrega.</li> <li>- El llistat es pot ordenar o filtrar per estat.</li> </ul>   |
| <b>US9</b>  | Com a usuari, vull tenir una vista de calendari amb entregues de pràctiques           | <ul style="list-style-type: none"> <li>- El calendari mostra visualment les entregues segons l'estat (amb colors).</li> <li>- Es pot navegar per setmanes.</li> <li>- En clicar una entrega, es mostra informació detallada i accés directe a la pràctica.</li> </ul> |
| <b>US10</b> | Com a estudiant, vull rebre notificacions de canvis d'estat d'una pràctica            | <ul style="list-style-type: none"> <li>- Quan una pràctica canvia d'estat, es mostra una notificació immediata via Toast.</li> <li>- Si el nou estat és "Corregida", s'activa un ping visual al botó de Pràctiques.</li> </ul>                                        |
| <b>US11</b> | Com a usuari, vull una navegació clara                                                | <ul style="list-style-type: none"> <li>- La SideNav i la Navbar estan disponibles en tot moment.</li> <li>- La SideNav es pot col·lapsar i expandir.</li> </ul>                                                                                                       |

|             |                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                                          | <ul style="list-style-type: none"> <li>- La Navbar pot aparèixer i desaparèixer quan l'usuari fa scroll a la pàgina.</li> <li>- Els botons de la SideNav porten correctament a les vistes corresponents.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                           |
| <b>US12</b> | Com a usuari, vull visualitzar les notificacions rebudes i gestionar-les | <ul style="list-style-type: none"> <li>- En fer clic al botó amb icona de la campana a la navbar s'obre un popover amb dues pestanyes: No llegides i Totes.</li> <li>- Les notificacions no llegides es mostren primer i tenen un estil destacat.</li> <li>- En clicar una notificació no llegida, es marca automàticament com a llegida.</li> <li>- Si es clica una notificació ja llegida, redirigeix l'usuari a la pàgina de la pràctica corresponent.</li> <li>- Hi ha un botó per esborrar totes les notificacions.</li> <li>- Hi ha un botó per marcar totes com a llegides.</li> </ul> |
| <b>US13</b> | Com a usuari, vull un dashboard amb informació rellevant                 | <ul style="list-style-type: none"> <li>- Mostra cursos accedits recentment, calendari setmanal i taula de pràctiques properes.</li> <li>- El contingut es personalitza segons l'usuari autenticat.</li> <li>- El calendari és setmanal i s'adapta a la mida de pantalla.</li> </ul>                                                                                                                                                                                                                                                                                                           |

|             |                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                                | <ul style="list-style-type: none"> <li>- Només es mostra el rang horari relatiu a l'hora actual.</li> <li>- El dia actual està ressaltat amb una marca visual blava.</li> <li>- Si un dia té pràctiques amb la mateixa data de lliurament, es marca amb un indicador.</li> <li>- Es pot seleccionar un dia per visualitzar-lo.</li> <li>- La taula de pràctiques mostra <i>Pràctica, Curs, Professor, Estat, Data de venciment, Accions</i> pels alumnes, els professors i admins no tenen la columna <i>Estat</i>.</li> <li>- La taula es pot filtrar per cerca de nom, rang de dates properes i estat de la pràctica.</li> </ul> |
| <b>US14</b> | Com a professor, vull poder gestionar cursos                   | <ul style="list-style-type: none"> <li>- Es poden crear, editar i eliminar cursos.</li> <li>- En eliminar un curs es mostra una notificació de confirmació.</li> <li>- L'edició afecta el nom, descripció i semestre.</li> <li>- Els canvis es reflecteixen immediatament a la plataforma.</li> </ul>                                                                                                                                                                                                                                                                                                                              |
| <b>US15</b> | Com a professor, vull poder gestionar pràctiques d'un curs meu | <ul style="list-style-type: none"> <li>- Es poden afegir, editar i eliminar pràctiques dins un curs.</li> <li>- En eliminar una pràctica es mostra una notificació de confirmació.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                      |

|             |                                                                                                                 |                                                                                                                                                                                                                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             |                                                                                                                 | <ul style="list-style-type: none"> <li>- S'han d'associar fitxers d'autocorrecció per crear la pràctica.</li> <li>- Es defineixen les dates de lliurament i les descripcions.</li> </ul>                                                                            |
| <b>US16</b> | Com a professor, vull poder gestionar els usuaris d'un curs meu                                                 | <ul style="list-style-type: none"> <li>- Es poden afegir o eliminar usuaris al curs.</li> <li>- En eliminar un usuari es mostra una notificació de confirmació.</li> <li>- Els canvis tenen efecte immediat.</li> </ul>                                             |
| <b>US17</b> | Com a usuari, vull cercar cursos o pràctiques fàcilment                                                         | <ul style="list-style-type: none"> <li>- Hi ha un cercador global amb selector de context (cursos o pràctiques).</li> <li>- Els resultats es mostren en un popover dinàmic sota el camp de cerca.</li> <li>- El cercador és accessible des de la navbar.</li> </ul> |
| <b>US18</b> | Com a usuari (professor o estudiant), vull accedir a la configuració des del botó de configuració de la SideNav | <ul style="list-style-type: none"> <li>- En fer clic al botó "Configuració" s'obre un modal amb dues pestanyes: <i>Perfil</i> i <i>Contrasenya</i>.</li> <li>- El modal es pot tancar fàcilment i manté l'estat actual si es reobre.</li> </ul>                     |
| <b>US19</b> | Com a usuari, vull modificar el meu perfil des de la pestanya <i>Perfil</i> al modal de configuració d'usuari   | <ul style="list-style-type: none"> <li>- Es mostren els camps <i>Nom</i>, <i>Cognoms</i> i <i>Email</i> preomplerts.</li> <li>- Es poden editar i desar els canvis.</li> <li>- Es mostra una notificació de confirmació o error segons el resultat.</li> </ul>      |

|             |                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>US20</b> | Com a usuari, vull canviar la meua contrasenya des de la pestanya <i>Contrasenya</i> al modal de configuració d'usuari | <ul style="list-style-type: none"> <li>- Es requereix introduir la contrasenya actual.</li> <li>- Cal introduir dues vegades la nova contrasenya.</li> <li>- El sistema valida que coincideixin i mostra un missatge d'error si no és així.</li> <li>- Si és correcte, es desa i es notifica a l'usuari.</li> <li>- Si no és correcte la contrasenya actual mostra un missatge d'error.</li> </ul> |
| <b>US21</b> | Com a administrador, vull veure una tercera pestanya d' <i>Administració</i> dins del modal de configuració d'usuari   | <ul style="list-style-type: none"> <li>- Només visible si el rol de l'usuari autenticat és <i>Admin</i>.</li> <li>- Mostra selector amb una llista d'usuaris del sistema.</li> </ul>                                                                                                                                                                                                               |
| <b>US22</b> | Com a administrador, vull poder modificar el rol d'un usuari des de la pestanya <i>Administració</i>                   | <ul style="list-style-type: none"> <li>- Es pot seleccionar un usuari.</li> <li>- Es pot canviar el seu rol (estudiant, professor, administrador) amb diferents switchs un per cada rol.</li> <li>- El canvi es desa i queda reflectit immediatament.</li> </ul>                                                                                                                                   |
| <b>US23</b> | Com a administrador, vull poder restablir la contrasenya d'un usuari des de la pestanya <i>Administració</i>           | <ul style="list-style-type: none"> <li>- El botó de restablir contrasenya reinicia la contrasenya de l'usuari seleccionat.</li> <li>- Es genera una contrasenya per defecte.</li> <li>- Es mostra una notificació de confirmació.</li> </ul>                                                                                                                                                       |

|             |                                                                                                                                     |                                                                                                                                                                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>US24</b> | Com a usuari, vull poder canviar entre mode fosc i mode clar per adaptar la visualització de la plataforma a les meves preferències | <ul style="list-style-type: none"><li>- Hi ha un botó a la navbar per alternar entre mode fosc i mode clar.</li><li>- El mode seleccionat es guarda localment perquè es mantingui en futures visites.</li><li>- L'estil visual de la plataforma canvia automàticament segons el mode seleccionat.</li></ul> |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 3. Arquitectura de l'aplicació

Per tal de garantir un sistema escalable, modular, mantenible i fàcilment desplegable, s'ha dissenyat l'arquitectura de la versió 3 de Golem XIV basant-se en una separació clara de responsabilitats mitjançant serveis distribuïts. Aquesta arquitectura segueix un enfocament tipus microserveis, on diferents components del sistema interactuen de forma desacoblada, permetent un desenvolupament independent, una millor escalabilitat i una alta disponibilitat.

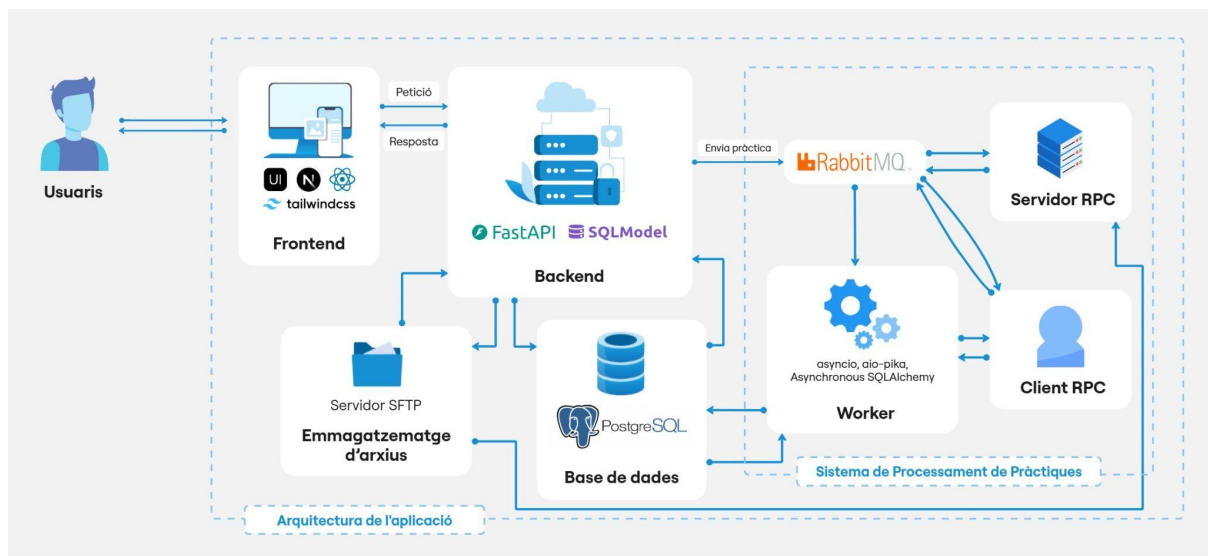


Figura 1. Arquitectura de l'aplicació

#### Flux de Treball del Sistema

1. L'estudiant puja una pràctica a través del frontend.
2. El backend valida i emmagatzema la informació.
3. El backend genera un missatge a la cua RabbitMQ amb la informació de la pràctica necessària per al servidor RPC.
4. El worker consumeix el missatge i inicia el processament.
5. El worker es comunica amb el servidor RPC per a la correcció mitjançant una instància del client RPC i RabbitMQ.
6. Una vegada el client RPC rep la resposta del servidor de correcció, els resultats i el nou estat de la pràctica de l'estudiant es desen a la base de dades des del worker.
7. Es generen notificacions en temps real per a l'usuari.

## 3.1. Backend

El backend de Golem XIV constitueix el nucli funcional del sistema, encarregat de gestionar la lògica de negoci, validar les dades, mantenir la persistència i coordinar la comunicació amb la resta de serveis. Està desenvolupat amb FastAPI i estructurat de manera modular per garantir una alta mantenibilitat i escalabilitat.

Aquest backend actua com a pont entre el frontend i el sistema de correcció, exposant múltiples endpoints per a operacions relacionades amb usuaris, curso i pràctiques. També és responsable de la gestió de fitxers, la seguretat (autenticació i autorització), i l'enviament de missatges al sistema de cues RabbitMQ per desencadenar processos de correcció.

### 3.1.1. FastAPI

FastAPI [\[2\]](#) és un framework web modern i d'alt rendiment per a la construcció d'APIs web amb Python 3.7 o superior [\[14\]](#). Ha estat dissenyat per ser fàcil d'utilitzar, però a la vegada potent i molt eficient. Està construït sobre dues tecnologies sòlides i molt reconegudes en l'ecosistema Python:

- **Starlette** [\[25\]](#): per a la gestió de peticions HTTP, middleware, rutes, WebSockets i més.
- **Pydantic** [\[12\]](#): per a la validació i serialització de dades mitjançant models fortament tipats.

FastAPI aprofita el sistema d'anotacions de tipus de Python per generar automàticament validacions de dades, documentació interactiva (OpenAPI) i integració nativa amb sistemes asincrònics (com asyncio), cosa que el fa ideal per a aplicacions modernes i escalables.

#### Característiques destacades:

- **Generació automàtica de documentació interactiva** mitjançant Swagger UI [\[18\]](#) i ReDoc [\[19\]](#).
- **Validació automàtica de dades** d'entrada i sortida amb Pydantic.
- **Serialització i deserialització** transparent i tipada

- **Suport natiu per a asincronia** (async/await), permetent una alta concurrència amb menys recursos.
- **Simplicitat i llegibilitat** del codi, amb integració fluida entre lògica d'API, models i dades.

### Ús en l'Aplicació:

En aquest projecte, FastAPI és la columna vertebral del backend, i el seu ús és extens i fonamental per al funcionament de tot el sistema, proporcionant:

- **Creació d'Endpoints RESTful**
  - Gestió d'entitats com usuaris, cursos i pràctiques.
  - L'ús de *routers* i *prefixes* permet una estructura clara i escalable per l'API.
- **Autenticació i autorització**
  - Implementació de sistemes de login amb tokens (JWT).
  - Control d'accés segons el tipus d'usuari: estudiant, professor o administrador.
  - Integració amb middleware personalitzat per a protecció de rutes i validació de credencials.
- **Validació i serialització de dades**
  - Models Pydantic per definir i validar totes les dades rebudes pel servidor.
  - Reutilització dels models per a respostes API, garantint coherència entre client i servidor.
- **Documentació automàtica**
- **Gestió d'arxius**
  - Implementació d'endpoints per a la pujada, obtenció i eliminació de fitxers.
  - Integració amb el servidor SFTP per enviar o recuperar documents (tant d'estudiants com professors).
  - Validació del tipus de fitxer.

- **Integració amb altres serveis**
  - Enviament de missatges a RabbitMQ en lliuraments de pràctiques.
  - Comunicació amb la base de dades PostgreSQL via SQLAlchemy.

### 3.1.2. SQLAlchemy

SQLModel [10] és una llibreria de Python per interactuar amb bases de dades relacionals creada pel mateix autor de FastAPI. Combina les millors característiques de dues tecnologies molt consolidades:

- **Pydantic [12]**, per a la validació de dades i tipatge estàtic.
- **SQLAlchemy [24]**, per a la interacció amb la base de dades i gestió d'ORM (Object Relational Mapping).

SQLModel simplifica la definició de models de dades que poden servir simultàniament com a models de validació, esquemes d'entrada/sortida i models de base de dades, fet que redueix redundàncies i manté el codi coherent i mantenible.

#### Característiques destacades:

- **Declaració de models unificats:** els mateixos models serveixen per interactuar amb l'API (entrada/sortida) i per persistència a la base de dades.
- **Integració nativa amb FastAPI:** permet passar directament els models SQLAlchemy als endpoints com si fossin Pydantic Models.
- **Basat en SQLAlchemy:** conserva tota la potència d'aquest ORM (consultes, joins, sessions, relacions, etc.).
- **Tipatge totalment estàtic:** permet detectar errors abans d'executar el codi.

#### Ús en l'Aplicació:

SQLModel s'utilitza com a ORM principal per definir, validar i consultar totes les entitats persistents del sistema, com ara usuaris, cursos i pràctiques.

SQLModel s'ha implementat per substituir l'anterior ús separat de SQLAlchemy i Pydantic, proporcionant:

- **Models Unificats:** Els models es defineixen només una vegada i serveixen tant per:
  - Emmagatzemar dades a la base de dades
  - Validar dades d'entrada i definir respostes d'API
- **Relacions entre entitats:** Definició de relacions entre taules mitjançant claus foranes i Relationship que es reflecteixen de forma bidireccional i eficient a l'ORM
- **Respostes Tipificades:** Els endpoints de FastAPI retornen directament models SQLAlchemy
- **Validació Integrada:** Validació automàtica en entrada i sortida de dades
- **Manteniment Simplificat:** Menys codi per mantenir i menor possibilitat d'inconsistències
- **Migracions amb Alembic:** Ús de les migracions amb Alembic cosa que permet versionar els esquemes i aplicar canvis progressius a la base de dades PostgreSQL gràcies a la compatibilitat amb SQLAlchemy

### 3.1.1. Alembic

Alembic [\[11\]](#) és una eina de migració de bases de dades per a Python, dissenyada específicament per funcionar amb SQLAlchemy. Permet versionar l'estructura d'una base de dades, aplicar canvis progressius als esquemes de manera controlada i mantenir sincronitzada la definició dels models amb la implementació real de la base de dades.

És una eina fonamental quan es treballa amb ORM, ja que evita la necessitat de recrear la base de dades manualment cada vegada que es modifica un model.

#### Funcionalitats clau:

- Migracions automàtiques i manuals
- Versionat d'esquemes de base de dades
- Control des de línia de comandes
- Integració amb SQLAlchemy/SQLAlchemy
- Rollback de migracions

- Seguretat i traçabilitat

### Ús en l'Aplicació:

S'utilitza per gestionar l'evolució estructural de la base de dades PostgreSQL. El projecte ha anat creixent i afegint funcionalitats, i Alembic ha estat essencial per implementar tots aquests canvis de manera segura i ordenada.

- **Generació automàtica de migracions**
- **Aplicació de migracions**
- **Control de versions i revertibilitat**
- **Desplegament Automatitzat:** Migracions automàtiques en el procés de desplegament

## 3.2. Base de dades

### 3.2.1. PostgreSQL

PostgreSQL [\[13\]](#) és un sistema de gestió de base de dades relacional (RDBMS) de codi obert àmpliament reconegut per la seva robustesa, consistència i rendiment elevat. S'ha convertit en una de les solucions més utilitzades tant en entorns acadèmics com en aplicacions empresarials per la seva compatibilitat amb l'estàndard SQL, la seva capacitat d'escalar i la seva capacitat d'extensió mitjançant mòduls, funcions definides per l'usuari i tipus de dades personalitzats.

És especialment adequada per a projectes que requereixen consistència de dades, control transaccional i operacions complexes, com és el cas de Golem XIV, una aplicació centrada en la gestió educativa i l'avaluació automàtica.

### Característiques destacades:

- Compliment dels principis ACID
- Suport natiu per a tipus de dades avançats (arrays, enum, UUIDs, JSON)
- Índexs avançats i optimització de consultes
- Extensibilitat i funcions personalitzades
- Replicació i alta disponibilitat
- Seguretat i control d'accés

### Ús en l'Aplicació:

En el projecte Golem XIV, PostgreSQL actua com a component central de persistència de dades. Inicialment, l'aplicació es va dissenyar amb dues bases de dades separades, fet que introduïa complexitat i dificultava el manteniment. Com a part de la refactorització en aquesta tercera versió, s'ha consolidat tota la persistència a una sola base de dades PostgreSQL, millorant l'eficiència, la coherència de dades i la simplicitat de la infraestructura.

- **Dades d'Usuaris:** Informació d'identificació, rols (estudiant, professor, administrador) i credencials.
- **Gestió Acadèmica:** Cursos i pràctiques
- **Integració amb SQLModel:** Tota la interacció amb PostgreSQL es realitza mitjançant SQLModel
- **Notificacions en temps real:** PostgreSQL permet activar triggers en canvis d'estat a determinades taules (per exemple, quan una pràctica d'un estudiant passa a "Corregida").

### 3.3. Sistema de Processament de Pràctiques

Un dels components més crítics de Golem XIV és el sistema de processament de pràctiques, que s'encarrega de gestionar l'autocorrecció de les entregues dels estudiants. Aquest sistema s'ha dissenyat per operar de manera desacoblada, asíncrona i escalable, assegurant que les pràctiques es puguin corregir de forma eficient, fins i tot en moments de gran volum de càrrega.

Aquest sistema es basa en una arquitectura distribuïda formada per tres components principals:

1. **RabbitMQ:** Un sistema de cues de missatges que permet la comunicació desacoblada entre el backend i els serveis de processament.
2. **Worker:** Un servei consumidor que rep les pràctiques des del backend amb RabbitMQ, les processa en paral·lel i comunica amb el servidor RPC.

- 3. Servidor RPC:** El component que executa la lògica de correcció i retorna els resultats, que posteriorment el worker emmagatzema a la base de dades.

Aquesta arquitectura permet tolerància a fallades, concurrència elevada, i control exhaustiu de l'estat de cada pràctica.

### 3.3.1. RabbitMQ

RabbitMQ és un sistema de missatgeria lleuger i d'alt rendiment que implementa el protocol AMQP [\[20\]](#) (Advanced Message Queuing Protocol). Permet la comunicació desacoblada entre serveis mitjançant cues de missatges, cosa que el fa ideal per a arquitectures distribuïdes i basades en microserveis.

A diferència de la comunicació directa (per exemple via API HTTP), on client i servidor han d'estar disponibles al mateix temps, RabbitMQ permet enviar missatges que poden ser rebuts i processats de manera asíncrona, tolerant latències, errors temporals i processos independents.

#### Característiques principals:

- **Fiabilitat:** Garantia de lliurament de missatges
- **Flexibilitat:** Múltiples protocols i patrons de missatgeria
- **Escalabilitat:** Clustering i alta disponibilitat
- **Gestió:** Interfície web per a monitoratge
- **Persistència:** Emmagatzematge durable de missatges
- **Control avançat d'errors:** TTL, reintents limitats, cues mortes (DLQ), etc.

#### Ús en l'Aplicació:

RabbitMQ juga un paper fonamental en l'arquitectura de Golem XIV, ja que coordina la comunicació entre el backend, el worker i el servidor RPC. Aquest sistema de missatgeria garanteix una separació clara de responsabilitats i permet que la correcció de pràctiques es pugui realitzar de forma escalable, controlada i segura.

### 3.3.2. Worker

El worker és un component clau en l'arquitectura de Golem XIV, encarregat de consumir els missatges provinents del sistema de missatgeria RabbitMQ i gestionar el procés de correcció automatitzada de pràctiques. Aquest servei ha estat profundament refactoritzat per passar d'un model seqüencial a un de concurrent i asíncron, capaç d'aprofitar al màxim els recursos de la màquina on s'executa.

L'objectiu principal d'aquesta refactorització ha estat augmentar l'eficiència, escalabilitat i tolerància a errors del sistema de correcció.

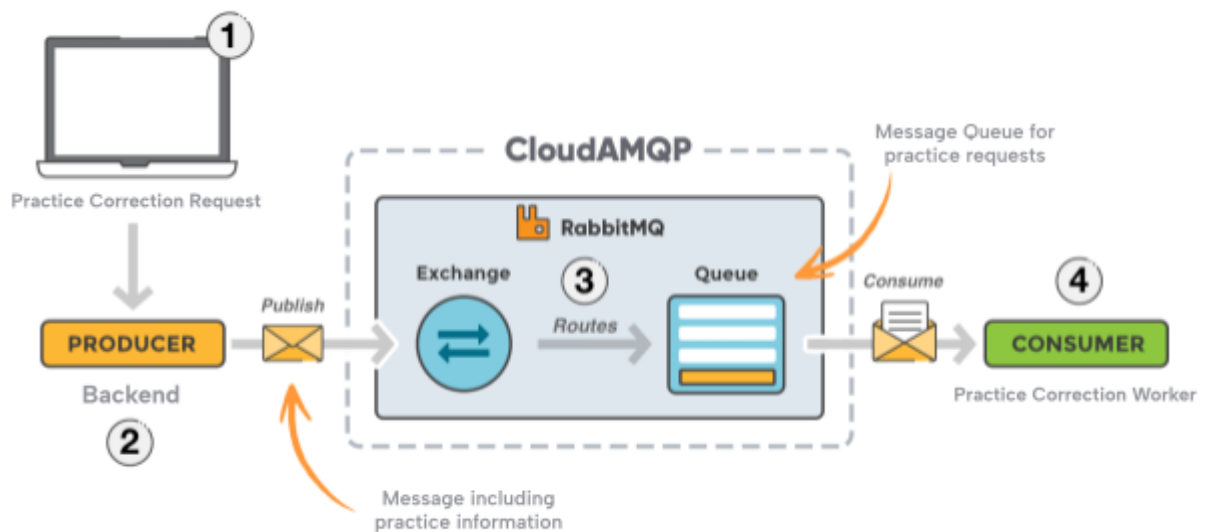


Figura 2. Diagrama del funcionament del paradigma productor-consumidor en l'aplicació amb RabbitMQ

#### Processament Concurrent:

##### Estat anterior:

- El worker només podia processar un missatge a la vegada, cosa que limitava significativament el rendiment, especialment quan hi havia diverses entregues pendents.

##### Estat actual:

- S'ha implementat un sistema de processament concurrent, que permet gestionar múltiples missatges de forma paral·lela.
- El límit de concurrència es defineix com a nuclis de la CPU  $\times 4$ , aprofitant tant el paral·lisme del sistema com l'escalabilitat de Python asíncron.

### **Tecnologies utilitzades:**

- **asyncio** [15]: Llibreria nativa de Python per a programació asíncrona, que permet llançar múltiples *tasks* sense bloqueig.
- **aio\_pika** [16]: Client AMQP asíncron basat en `asyncio`, utilitzat per connectar-se i consumir missatges de RabbitMQ de manera eficient.

### **Semàfor de concurrència:**

- Es fa ús d'un semàfor (`asyncio.Semaphore`) per limitar el nombre de tasques actives simultànies i evitar una saturació incontrolada del sistema.

### **Gestió d'Errors i Reintents:**

Per garantir la resiliència del sistema, s'ha incorporat reintents controlats, cues de missatges morts (DLQ) i temps de vida (TTL) per assegurar la gestió segura i eficient dels errors, evitant bloquejos i garantint l'estabilitat del sistema.

### **Gestió Asíncrona de Base de Dades:**

Perquè el worker pugui actualitzar correctament l'estat de les pràctiques a la base de dades en temps real, s'ha integrat SQLAlchemy [24] en mode asíncron mitjançant el mòdul `sqlalchemy.ext.asyncio`. Aquesta arquitectura ofereix múltiples avantatges:

- Integració amb el paradigma asíncron de Python (`async/await`).
- Evita el bloqueig del fil principal durant operacions de lectura i escriptura a la base de dades.
- Millora el rendiment general del worker quan ha de tractar simultàniament amb múltiples entregues.
- Permet aplicar transaccions, control d'errors i gestió eficient de sessions dins d'un flux asíncron.

### **3.3.3. Servidor RPC**

El concepte de RPC (Remote Procedure Call) es basa en la possibilitat d'executar funcions o procediments remots com si fossin crides locals dins del programa que les invoca. En altres paraules, permet que un programa invoqui una rutina

que s'executa en un altre servidor o procés, sense necessitat de gestionar explícitament la connexió o la comunicació de baix nivell.

Aquest model resulta molt útil en arquitectures distribuïdes, ja que permet desacoblar completament components especialitzats i reutilitzables, tot mantenint una comunicació eficient i controlada.

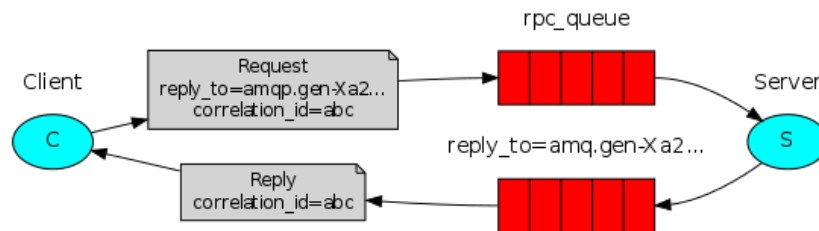


Figura 3. Diagrama del funcionament del protocol RPC

En el cas del projecte el servidor RPC és una peça clau dins del sistema de correcció automatitzada de pràctiques. Està dissenyat com un microservei especialitzat i independent, el qual està totalment enfocat a processar els arxius d'entrega dels estudiants i retornar els resultats de la seva execució, així com el feedback associat.

Aquest servidor no forma part directa del backend ni del worker, però hi interacciona constantment a través d'un canal de comunicació específic. Aquesta separació ofereix una sèrie d'avantatges arquitectònics:

- **Escalabilitat independent:** el servidor RPC pot escalar-se o redeplegar-se sense afectar la resta del sistema.
- **Extensibilitat:** es poden afegir nous llenguatges de correcció, millorar el motor d'anàlisi o aplicar noves rúbriques sense modificar el backend.
- **Aïllament de riscos:** mantenir-lo fora del backend redueix l'impacte de possibles errors o vulnerabilitats.

### Ús en l'Aplicació:

El servidor RPC és l'encarregat d'executar el procés de correcció automàtica. Rep tota la informació necessària d'una pràctica, executa el codi de l'estudiant en un

entorn aïllat, aplica l'autocorrecció i retorna el resultat, que el consumidor worker registra i vincula amb l'estat de la pràctica.

Aquest servidor ha estat desenvolupat pel director del TFG, i està previst que es publiqui pròximament com a programari de codi obert.

### **3.4. Emmagatzematge d'arxius**

#### **3.4.1. Servidor SFTP**

SFTP [\[21\]](#) (Secure File Transfer Protocol) és un protocol de xarxa que permet la transferència de fitxers de manera segura sobre una connexió SSH (Secure Shell). A diferència de l'FTP tradicional, SFTP xifra tant les dades com les credencials durant la transmissió, garantint la confidencialitat i la integritat de la informació. És molt utilitzat en entorns on la seguretat és prioritària, ja que evita exposar els fitxers i les contrasenyes a la xarxa en text pla.

SFTP funciona com un canal segur per pujar, descarregar, moure o eliminar arxius en un servidor remot, oferint una interfície de comandes o biblioteques que permeten integrar-lo programàticament a aplicacions com la que s'ha desenvolupat en aquest projecte.

#### **Ús en l'Aplicació:**

S'ha optat per utilitzar SFTP com a sistema d'emmagatzematge i intercanvi de fitxers, ja que és el protocol compatible amb el servidor RPC i ofereix una transmissió segura mitjançant connexions xifrades. Per tant, actua com a repositori segur per a tots els fitxers relacionats amb les pràctiques:

- Els professors poden pujar arxius d'autocorrecció associats a una pràctica. Aquests arxius s'envien al servidor SFTP des del backend.
- Els estudiants poden entregar les seves pràctiques mitjançant el frontend. El backend rep aquests fitxers i també els emmagatzema al SFTP.
- El servidor RPC, durant el procés de correcció, accedeix al SFTP per obtenir tant l'arxiu de l'estudiant com el fitxer d'autocorrecció.

- Els professors poden descarregar les entregues dels estudiants en qualsevol moment des del frontend, mitjançant una connexió controlada al SFTP a través del backend.

## 3.5. Frontend

El frontend de Golem XIV ha estat dissenyat per oferir una experiència d'usuari moderna, intuïtiva i eficient, aprofitant les últimes tecnologies del desenvolupament web. El sistema està implementat amb Next.js, estilitzat amb TailwindCSS, i enriquit amb components visuals gràcies a HeroUI. Aquesta combinació ha permès construir una interfície clara, ràpida, modular i altament mantenible.

### 3.5.1. Next.js

Next.js [\[6\]](#) és un framework de React orientat a entorns de producció, dissenyat per facilitar el desenvolupament d'aplicacions web escalables i altament optimitzades.

#### Característiques principals:

- **Server-Side Rendering (SSR):** Generació dinàmica del contingut al servidor, útil per a millorar el SEO i reduir el temps de càrrega inicial.
- **Static Site Generation (SSG):** Prerenderització de pàgines amb contingut que canvia poc, millorant el rendiment.
- **File-based Routing:** Rutes basades en l'estructura de fitxers.
- **Optimització Automàtica:** Imatges, fonts i bundles
- **Hot Reloading:** Les modificacions es reflecteixen instantàniament durant el desenvolupament.
- **Optimització automàtica:** Gestió intel·ligent de *chunks*, compressió, càrrega *lazy* de components i optimització d'imatges.
- **API Routes:** Possibilitat de definir rutes backend dins el mateix projecte si cal.

### Ús en l'Aplicació:

Next.js és la base tecnològica del frontend de Golem XIV, i s'utilitza per:

- **Renderització optimitzada:** Les pàgines clau com el dashboard i el calendari utilitzen SSR per millorar el rendiment inicial.
- **Navegació dinàmica:** Implementació de rutes dinàmiques per a cursos i pràctiques, amb comportament fluid i sense recàrrega de pàgina.
- **Gestió d'estats i dades:** Comunicació amb el backend mitjançant API REST per mostrar cursos, pràctiques, notificacions i usuaris.
- **Mode fosc/clar:** Commutació automàtica gràcies a la integració amb TailwindCSS.

### 3.5.2. TailwindCSS

TailwindCSS [\[7\]](#) és un framework CSS utility-first que permet construir interfícies personalitzades i estilitzades directament des del codi HTML mitjançant classes petites i composables. A diferència de frameworks com Bootstrap, no imposa cap estil per defecte i ofereix màxima flexibilitat de disseny.

#### Filosofia i Avantatges:

- **Utility-First:** Estilització mitjançant classes específiques (p-4, bg-blue-500, text-center, etc.).
- **Customització:** Configuració altament personalitzable
- **Responsive per defecte:** Suport complet a breakpoints amb prefixos
- **Dark Mode integrat:** Canvi de temes automàtic amb preferències del navegador o toggle manual.
- **Purging automàtic:** Eliminació del CSS no utilitzat durant el build per minimitzar la mida dels arxius finals.
- **JIT Engine (Just-In-Time):** Compilació instantània i dinàmica de les classes utilitzades.

### Ús en l'Aplicació:

TailwindCSS permet crear una interfície consistent i responsive:

- **Design System:** Paleta de colors, espaiat i tipografia consistent
- **Responsive:** Adaptació automàtica a diferents mides de pantalla

- **Components Reutilitzables:** Classes utility combinades per crear components
- **Performance:** CSS mínim gràcies al purging automàtic
- **Productivitat:** Desenvolupament ràpid sense sortir de l'HTML

### 3.5.3. HeroUI

HeroUI [\[8\]](#) és una llibreria de components UI per a React, dissenyada per funcionar perfectament amb TailwindCSS. Ofereix components moderns, accessibles i altament personalitzables, optimitzats per a rendiment i integració amb TypeScript.

#### Característiques principals:

- **Components preconstruïts:** Modals, pestanyes, dropdowns, botons, toasts, popovers, etc.
- **Accessibilitat garantida:** Compliment total de les directrius WCAG amb suport per teclat i lectors de pantalla.
- **Estils adaptables:** Integració directa amb la configuració de temes i colors de Tailwind.
- **Dark Mode:** Suport natiu per temes clar i fosc.
- **Tipat estàtic:** Compatible amb TypeScript per a més seguretat i desenvolupament assistit.

#### Ús en l'Aplicació:

HeroUI ha estat emprada per construir la majoria de components visuals reutilitzables de la plataforma, com ara modals de configuració, sistemes de pestanyes, notificacions, menús desplegable i botons, mantenint una coherència visual i millorant l'accessibilitat.

## 3.6. Arquitectura Distribuïda al Núvol

Una de les millores més importants introduïdes en la tercera versió de Golem XIV ha estat la migració cap a una arquitectura distribuïda desplegada al núvol, amb l'objectiu de millorar l'escalabilitat, la modularitat, i garantir una alta disponibilitat.

Aquesta decisió ha permès separar funcionalment els diferents components de l'aplicació, assignant-los recursos i entorns d'execució independents segons les seves necessitats.

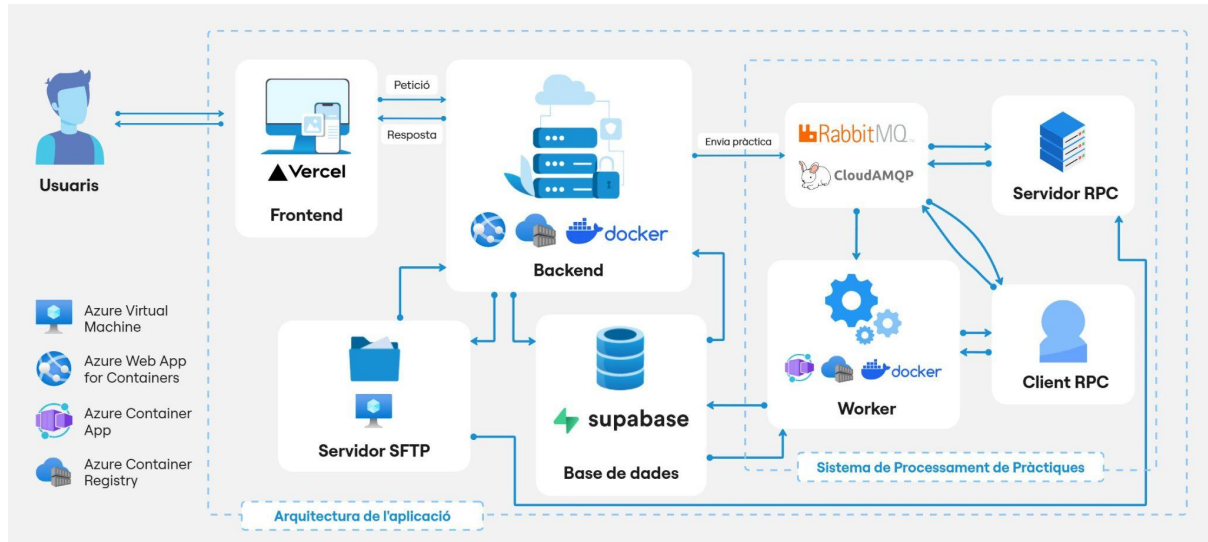


Figura 4. Arquitectura distribuïda al núvol de l'aplicació

### Visió general de l'arquitectura distribuïda

El sistema es compon de diversos serveis interconnectats però desplegats per separat, cadascun en plataformes cloud especialitzades. Aquesta arquitectura permet que cada component s'escali, s'actualitzi o es desplegui de manera independent, sense afectar la resta de l'aplicació.

A continuació es detalla el desplegament de cada component.

#### 3.6.1. Backend

El backend ha estat empaquetat dins d'un contenidor Docker [9], garantint que es pot executar de manera idèntica en qualsevol entorn, independentment del sistema operatiu, biblioteques o configuracions locals. Aquesta contenidorització assegura portabilitat, reproduïbilitat i facilita el desplegament automatitzat dins d'entorns cloud.

Per a la seva execució en producció, s'ha utilitzat el servei Azure Web App for Containers [3], una plataforma de Microsoft Azure que permet executar aplicacions web Dockeritzades amb una infraestructura gestionada. Aquesta

plataforma ofereix monitoratge, actualització contínua, escala automàtica, i una integració fluida amb sistemes de control de versions, cosa que ha facilitat enormement el cicle de desplegament i manteniment.

### 3.6.2. Base de Dades

La base de dades s'ha desplegat amb Supabase [4], una plataforma de Backend-as-a-Service que proporciona una instància gestionada de PostgreSQL, a més de funcionalitats addicionals com autenticació, emmagatzematge de fitxers, i notificacions en temps real mitjançant WebSocket. Aquesta elecció ha permès:

- Desplegar la base de dades sense necessitat de configurar ni mantenir manualment la infraestructura.
- Fer ús de **Supabase Realtime**, que permet escoltar canvis a taules concretes i emetre notificacions instantànies al frontend, com quan una pràctica passa a estat "corregint-se", "corregida" o "rebutjada".
- Gestionar esquemes i accés a dades des d'un entorn gràfic, a més de suportar l'ús avançat via psq1 (terminal interactiva per PostgreSQL), clients SQL o ORM com SQLAlchemy.

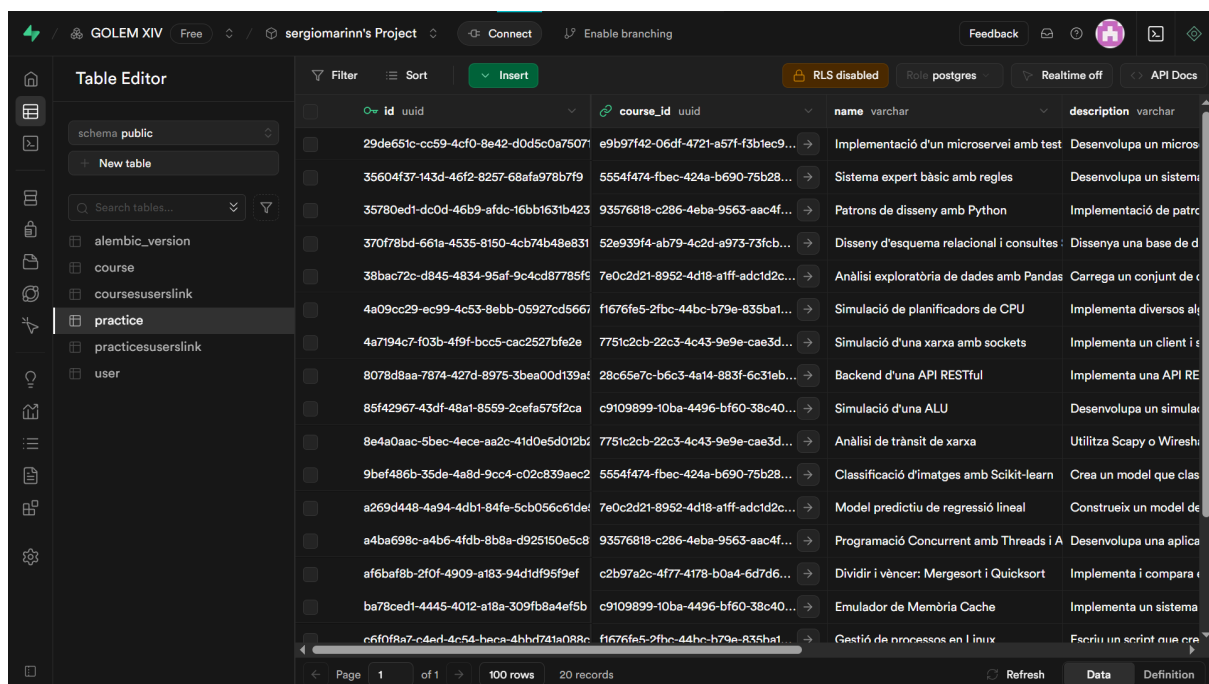


Figura 5. Panell d'edició de taules de la base de dades a Supabase

### 3.6.3. Worker

El worker també ha estat empaquetat en una imatge Docker pròpia. Aquest contenidor inclou totes les dependències necessàries i permet executar múltiples instàncies si cal escalar el processament.

El seu desplegament s'ha dut a terme a Azure Container Apps [\[22\]](#), un servei serverless de Microsoft Azure dissenyat per executar contenidors de forma escalable i gestionada. Aquesta plataforma ofereix control automàtic de recursos segons la càrrega de treball (scaling a zero inclòs), monitoratge de logs en temps real i comunicació fluida amb la resta de serveis de la infraestructura Azure.

Aquesta elecció permet que el worker funcioni de forma robusta i paral·lela, sense necessitat de mantenir servidors propis o configurar balancejadors de càrrega.

### 3.6.4. Servidor RPC

El servidor RPC, encarregat d'executar el codi de correcció de pràctiques de manera aïllada i controlada, s'ha desplegat en una màquina pròpia de la facultat. Aquest component pot mantenir-se en entorn intern perquè:

- No necessita exposar-se públicament.
- Requereix recursos puntuals de càlcul, però no gestiona dades sensibles.
- Pot accedir a les pràctiques a través del servidor SFTP i retornar resultats al worker.

La seva execució en un entorn aïllat proporciona seguretat addicional, especialment quan es tracta d'executar codi enviat per estudiants, i facilita la seva integració amb el sistema docent de la institució.

### 3.6.5. Frontend

El frontend, desenvolupat amb Next.js, s'ha desplegat a Vercel [\[17\]](#), una plataforma dissenyada específicament per a aquest framework. Vercel ofereix:

- **SSL i domini inclòs**, amb monitoratge de performance i estat.
- **Escalat automàtic**, servint les pàgines des de múltiples regions per millorar la latència.

- Funcionalitats de previsualització (preview deployments), útils per testing abans de fer canvis en producció.

Aquesta elecció ha facilitat molt la publicació contínua del frontend i ha garantit una experiència d'usuari fluida i ràpida.

### 3.6.6. Sistema de Missatgeria

El sistema de cues de missatges s'ha desplegat amb CloudAMQP [5], una plataforma especialitzada que ofereix instàncies gestionades de RabbitMQ al núvol. La seva integració ha estat fonamental per garantir la comunicació desacoblada i asíncrona entre el backend, el worker i el servidor RPC.

CloudAMQP proporciona:

- Panell de control complet per visualitzar cues, missatges pendents, reintents i errors.
- Integració segura via AMQPS (connexió xifrada).
- Suport per TTL, DLQ, reintents amb x-death i cues paral·leles.
- Logs i alertes en temps real.

Aquesta elecció evita la necessitat de mantenir la infraestructura de RabbitMQ pròpia i garanteix una alta disponibilitat.

| Virtual host | Name            | Type    | Features                                       | State   | Ready | Unacked | Total | incoming | deliver | get    | ack |
|--------------|-----------------|---------|------------------------------------------------|---------|-------|---------|-------|----------|---------|--------|-----|
| rsxzzzyg     | java            | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     | 0.00/s   | 0.00/s  | 0.00/s |     |
| rsxzzzyg     | java_checks     | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     | 0.00/s   | 0.00/s  | 0.00/s |     |
| rsxzzzyg     | ping            | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     | 0.00/s   | 0.00/s  | 0.00/s |     |
| rsxzzzyg     | practicas       | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     | 0.00/s   | 0.00/s  | 0.00/s |     |
| rsxzzzyg     | practicas.dlq   | classic | D Args default rsxzzzyg-max-length             | running | 15    | 0       | 15    | 0.00/s   |         |        |     |
| rsxzzzyg     | python          | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     |          |         |        |     |
| rsxzzzyg     | python_checks   | classic | D Args default rsxzzzyg-max-length             | running | 0     | 0       | 0     |          |         |        |     |
| rsxzzzyg     | retry.practicas | classic | D TTL DLX DLK Args default rsxzzzyg-max-length | running | 0     | 0       | 0     | 0.00/s   |         |        |     |
| rsxzzzyg     | rpc_queue       | classic | Args default rsxzzzyg-max-length               | running | 0     | 0       | 0     | 0.00/s   | 0.00/s  | 0.00/s |     |

Figura 6. RabbitMQ Manager de la instància desplegada a CloudAMQP

### 3.6.7. Servidor SFTP

El servidor SFTP s'ha desplegat a una màquina virtual amb Linux dins de Microsoft Azure [23]. Aquesta màquina funciona com a repositori centralitzat i segur d'arxius, on es desen:

- Les pràctiques entregades pels estudiants.
- Els fitxers d'autocorrecció carregats pels professors.

## 3.7. Desplegament Continu

Per tal de garantir un procés de desenvolupament àgil, fiable i automatitzat, s'ha incorporat al projecte Golem XIV un sistema de Continuous Deployment (CD) que permet que qualsevol canvi fet al repositori de codi es desplegui automàticament als entorns de producció corresponents. Aquest enfocament minimitza errors humans, accelera el temps de resposta davant canvis i facilita el manteniment del sistema en entorns distribuïts.

### 3.7.1. Components automatitzats en el CD

- **Frontend:**

Desplegat automàticament a Vercel. Qualsevol canvi a la branca principal del repositori main del GitHub genera una nova versió del frontend publicada en pocs segons. Vercel també ofereix versions de previsualització per fer proves abans de la publicació definitiva.

- **Backend i Worker:**

Ambdós estan contenidoritzats amb Docker i desplegats a Azure Web App for Containers i Azure Container Apps, respectivament. El sistema de CD es gestiona amb GitHub Actions, configurat perquè qualsevol commit a la branca main del repositori de GitHub executi automàticament:

- Construcció de la imatge Docker.
- Pujada de la imatge a Azure Container Registry.
- Desplegament del contenidor actualitzat al servei d'Azure.

- **Migracions de Base de Dades:**

Les migracions pendents amb Alembic s'executen de manera automatitzada com a pas dins del workflow de GitHub Actions associat a la branca main. Això assegura que l'esquema de la base de dades PostgreSQL a Supabase estigui sincronitzat amb la versió actual del codi desplegat, i evita inconsistències entre versions.

## 4. Implementació i resultats

---

Per tal de garantir un sistema escalable, modular, mantenible i fàcilment desplegable, s'ha dissenyat l'arquitectura de la versió 3 del projecte Golem XIV basant-se en una separació clara de responsabilitats mitjançant serveis distribuïts. Aquesta arquitectura segueix un enfocament tipus microserveis, on diferents components del sistema interactuen de forma desacoblada, permetent un desenvolupament independent, una millor escalabilitat i una alta disponibilitat.

### 4.1. Implementació del Backend

El backend de Golem XIV ha estat objecte d'una refactorització profunda respecte a les versions anteriors del projecte. Tot i que l'aplicació ja disposava d'una base funcional construïda amb FastAPI [\[2\]](#), s'ha reestructurat completament per aconseguir una arquitectura modular, escalable i fàcilment mantenible, optimitzada tant a nivell de codi com d'organització interna. En conjunt, el backend no només ha estat millor estructurat i optimitzat, sinó que ha experimentat una millora qualitativa profunda respecte a les versions anteriors, sent ara més robust, segur, escalable i preparat per a futures ampliacions.

#### 4.1.1. Reestructuració de l'arquitectura del projecte

Una de les millores estructurals més destacables d'aquesta nova versió de Golem XIV ha estat la reorganització completa de l'arquitectura del codi al backend, que ha passat d'un disseny monolític, amb codi poc modular i dispers, a una estructura neta, organitzada i plenament modularitzada. L'objectiu principal ha estat millorar la mantenibilitat, escalabilitat i llegibilitat del projecte, separant clarament la responsabilitat de cada component i alineant-lo amb bones pràctiques de desenvolupament professional.

#### **Abans**

La versió anterior del projecte presentava una arquitectura més plana i desestructurada. Tot el codi residia dins la carpeta `api/`, incloent models, controladors, client RPC i altres funcionalitats sense separació clara. A més,

existien fitxers amb noms inconsistents i una barreja de lògica de negoci, configuració i connexions en fitxers globals com `settings.py` o `database.py`. Això dificultava la reutilització, el testing i l'evolució del projecte.

## Ara

La nova estructura del projecte segueix un enfocament clarament desacoblat i escalable, organitzat dins la carpeta principal `app/`, amb submòduls específics per a cada responsabilitat:

- `app/api/`: conté els endpoints públics de l'API, agrupats per domini funcional (`courses.py`, `practices.py`, `users.py`, etc.) dins de la subcarpeta `routes/`. També inclou el fitxer `deps.py`, on es defineixen les dependències compartides com l'obtenció de l'usuari actual o els permisos per rol.
- `app/models/`: inclou tots els models de dades implementats amb `SQLModel`, separats per entitat (`course.py`, `user.py`, `practice.py`) i amb fitxers independents per a les taules intermitges (`CoursesUsersLink.py`, `PracticesUsersLink.py`). Aquesta separació millora la mantenibilitat i permet l'evolució modular del model relacional.
- `app/core/`: conté la configuració central del projecte, com ara la connexió a la base de dades (`db.py`), la configuració (`config.py`) i la gestió de seguretat i autenticació (`security.py`).
- `app/crud/`: aquest nou mòdul encapsula l'accés directe a la base de dades, separant la lògica de consultes i persistència de la lògica de negoci. Inclou submòduls per a cursos, pràctiques i usuaris. Aquest disseny segueix el patró CRUD (Create, Read, Update, Delete), molt habitual en APIs REST.
- `app/services/`: conté la lògica de negoci especialitzada, desacoblada dels endpoints. Per exemple, `practice_service.py` centralitza les operacions relacionades amb pràctiques, i `sftp_service.py` encapsula la gestió de fitxers amb el servidor SFTP.
- Altres fitxers com `initial_data.py` i `utils.py` completen l'aplicació amb utilitats generals i scripts d'inicialització.

A més, el sistema de migracions Alembic s'ha mantingut dins la carpeta `alembic/`, amb una estructura clara per controlar l'evolució del model de dades a través de versions incrementals.

En resum, aquesta reestructuració ha transformat l'arquitectura del projecte en un sistema modular, escalable i organitzat per responsabilitats, afavorint la col·laboració, les proves, el desplegament automatitzat i el creixement futur del projecte.

#### **4.1.2. Canvi d'ORM: de SQLAlchemy a SQLAlchemyModel**

En la versió anterior de Golem XIV, el backend feia ús de SQLAlchemy [\[24\]](#) com ORM per interactuar amb la base de dades. Tot i ser una eina potent, SQLAlchemy presenta certa complexitat quan es tracta d'unificar models de persistència i models de serialització (esquemes d'entrada i sortida), requerint sovint la definició duplicada de models (un per a la base de dades i un altre per a l'API). Aquest enfocament fragmentava la lògica de dades i dificultava el manteniment.

En aquesta nova versió, s'ha substituït SQLAlchemy per SQLAlchemyModel, una llibreria desenvolupada pel creador de FastAPI, que combina les capacitats de SQLAlchemy amb la simplicitat i la validació de Pydantic. Això permet definir un únic model per a cada entitat, que serveix tant per persistir les dades com per intercanviar-les mitjançant l'API.

Aquest canvi ha suposat un gran avenç en coherència i mantenibilitat, amb beneficis clars:

- S'han eliminat les redundàncies entre models de la base de dades i esquemes de resposta o validació.
- S'ha millorat la traçabilitat entre els components de l'aplicació (frontend, backend i base de dades).
- El codi s'ha simplificat i fet més declaratiu i llegible.

A més, SQLAlchemyModel permet integrar de manera senzilla tipus avançats i camps controlats, com enums o propietats calculades. Per exemple, camps com `semester`, `color` o `programming_language` es defineixen utilitzant enums

pròpiament tipats (`SemesterEnum`, `ColorEnum`, `ProgrammingLanguageEnum`), que es tradueixen a columnes SQL amb valors restringits i per defecte.

Aquesta estructura garanteix la validació automàtica, tant a nivell d'API com de base de dades, i evita errors de coherència en valors textuais.

D'altra banda, gràcies a l'ús de propietats calculades com `students_count` o `programming_languages`, els models poden oferir informació derivada de les relacions sense necessitat de fer càlculs externs o duplicar lògica en serveis o controladors. Aquestes propietats són especialment útils per generar respostes més completes i eficients cap al frontend.

També s'ha incorporat un control explícit sobre relacions many-to-many mitjançant `link_model`, com ara entre `User` i `Course`, o `User` i `Practice`, mantenint les taules intermitges gestionades pel mateix ORM, facilitant molt la gestió.

En resum, aquest canvi d'ORM ha comportat una modernització i neteja profunda del model de dades, alineant el projecte amb les bones pràctiques actuals i facilitant tant el desenvolupament com la futura evolució de l'aplicació.

#### **4.1.4. Unificació i millora de l'estructura de la base de dades**

Un dels canvis estructurals més significatius en aquesta nova versió de Golem XIV ha estat la reconstrucció total del model de dades, tant a nivell de l'esquema relacional com del modelatge a través de l'ORM. En versions anteriors, el sistema estava dividit en dues bases de dades separades, cadascuna amb una estructura poc normalitzada, amb noms inconsistents, redundàncies i sense cap control explícit sobre l'estat de les pràctiques ni l'evolució de l'usuari dins dels cursos. Aquesta divisió afegia complexitat innecessària, dificultava la sincronització entre dades, i dificultava les consultes i comprometia la mantenibilitat del sistema.

En aquesta nova versió, tota la informació s'unifica en una única base de dades PostgreSQL, clarament estructurada i dissenyada amb una nomenclatura normalitzada en anglès per mantenir coherència amb l'estil general del projecte i bones pràctiques de desenvolupament. L'ús d'enums tipats per representar valors

controlats, com l'estat de les pràctiques, els idiomes de programació o els colors dels cursos, reforça la validació semàntica de les dades i la seva extensibilitat.

### Millores clau respecte a l'estructura antiga

En el model antic, trobem taules com cursos, practicas, user, cursos\_usuario i practicas\_usuario, amb noms en castellà, camps sovint redundants, i una estructura simplificada que limitava l'expressivitat de les dades. Per exemple, el camp corregit a practicas\_usuario era només un booleà, fet que no permetia reflectir l'estat real del procés de correcció.

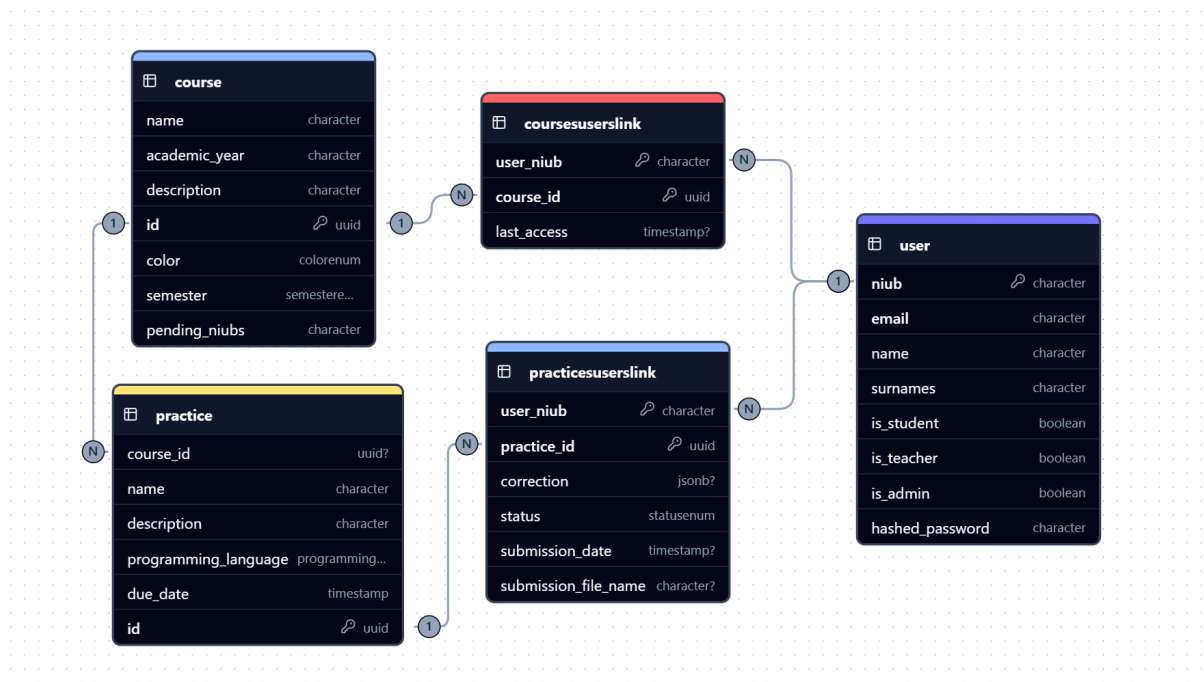


Figura 7. Esquema de la nova base de dades de Golem XIV

### A la nova versió, s'ha optat per relacions més riques i explícites:

- Les entitats `course`, `practice` i `user` es troben ara relacionades per taules intermitges (`coursesuserslink`, `practicesuserslink`) que no només connecten entitats, sinó que emmagatzemen metadades útils com `last_access`, `submission_date`, `status`, o `correction`, oferint així una traçabilitat completa de l'activitat de l'usuari dins del sistema.
- El camp `status` ja no és un simple booleà, sinó un enumerat (`statusenum`) amb cinc valors possibles que reflecteixen fidelment el cicle de vida d'una entrega:

NOT\_SUBMITTED, SUBMITTED, CORRECTING, CORRECTED, REJECTED.

Això permet implementar una lògica molt més acurada, detallada i reactiva, per exemple en la generació de notificacions i la visualització de l'estat de cada pràctica.

- S'han introduït també altres enumerats, com `colorenum` per identificar de manera consistent el color d'un curs pel frontend, `programminglanguageenum` per especificar el llenguatge de programació de cada pràctica, i `semesterenum` per codificar de manera explícita el semestre (TARDOR o PRIMAVERA).
- A nivell de disseny, les taules tenen ara claus primàries ben definides, com UUIDs per a `course` i `practice`, cosa que ofereix més robustesa a l'hora d'escalar el sistema o replicar-lo. També s'han eliminat camps confusos com `idiomaP` o `entrega`, consolidant-los en camps com `programming_language` o `due_date`.

#### 4.1.5. Millora del sistema autorització

En versions anteriors, el backend ja utilitzava autenticació amb tokens JWT per identificar l'usuari i validar-ne la sessió. En aquesta nova versió, s'ha reforçat el sistema d'autorització mitjançant la creació de dependències personalitzades que permeten controlar l'accés a cada endpoint de manera clara i reutilitzable.

Mitjançant l'ús de Depends de FastAPI, s'han implementat filtres específics per validar els rols dels usuaris. Per exemple, funcions com `get_current_teacher` o `get_current_active_superuser` asseguren que només els professors o administradors puguin accedir a determinades operacions, com ara la gestió de cursos, pràctiques o usuaris.

A més, s'ha definit un objecte reutilitzable anomenat `CurrentUser`, que permet injectar de manera automàtica l'usuari autenticat a qualsevol endpoint. Aquesta dependència facilita tant l'accés a la informació del mateix usuari que fa la petició (per exemple, per actualitzar el seu perfil, llistar els cursos o pràctiques), com l'aplicació de validacions específiques segons el context funcional. Això aporta

coherència, evita duplicació de codi i reforça la seguretat i mantenibilitat de tot el sistema.

#### 4.1.6. Gestió de fitxers via SFTP

En lloc d'emmagatzemar els arxius de forma local al servidor del backend o integrar-los dins la base de dades com es feia en versions anteriors, aquesta nova versió ha incorporat un sistema de gestió de fitxers mitjançant SFTP, allotjat en una màquina virtual Linux d'Azure, que permet:

- Pujar, baixar i eliminar arxius relacionats amb entregues i autocorreccions.
- Integració directa amb el servidor RPC, que consumeix els fitxers necessaris per a la correcció, mitjançant rutes proporcionades pel backend.
- Mantenir l'accés controlat i indirecte a través del backend, evitant exposar directament el servidor de fitxers a l'usuari final o al client RPC.

La gestió de directoris segueix una estructura jeràrquica clara i normalitzada, basada en el rol de l'usuari (estudiant o professor), l'any acadèmic, el nom del curs, la pràctica concreta, i el niub en el cas que sigui el directori d'estudiants. Exemple:

```
corrections_files/
├── students/
│   ├── 2023-2024/
│   │   ├── Algorismica/
│   │   │   ├── niub12345678
│   │   │   └── niub12344675
│   │   └──
│   └──
└── submission.zip
    ├── teachers/
    │   ├── 2023-2024/
    │   │   ├── Algorismica/
    │   │   │   ├── checks
    │   │   │   ├── scores.csv
    │   │   │   └── ...
    │   └──
    └──
```

#### 4.1.7. Endpoints i funcionalitats implementades

A nivell funcional, el backend exposa una API RESTful organitzada de manera estructurada i coherent, amb rutes agrupades segons el domini funcional que cobreixen: autenticació, gestió d'usuaris, cursos i pràctiques. Aquesta separació facilita tant el desenvolupament com el manteniment, i permet una gestió eficient dels permisos i de les funcionalitats de cada rol dins de la plataforma.

Les rutes principals de l'API estan agrupades sota quatre àrees: `/login`, `/users`, `/courses` i `/practices`.

La ruta `/login` és la responsable de la gestió del sistema d'autenticació mitjançant JSON Web Tokens (JWT). Aquesta ruta proporciona els endpoints necessaris perquè un usuari pugui iniciar sessió a la plataforma i obtenir un token d'accés vàlid, que posteriorment haurà d'incloure en les seves peticions a l'API per accedir a recursos protegits. A més, s'inclou un endpoint addicional per validar que un token determinat és vàlid i actiu, molt útil pel middleware del client.

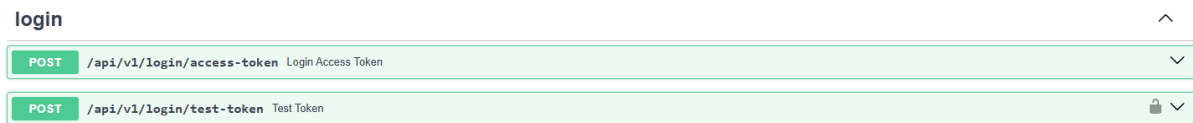


Figura 8. Endpoints de la ruta login

La ruta `/users` inclou els endpoints relacionats amb la creació, consulta i modificació d'usuaris. A diferència d'altres blocs de l'API, aquesta secció conté tant funcionalitats obertes com protegides. Qualsevol usuari pot registrar-se a través d'un endpoint públic, així com modificar el seu propi perfil i la seva contrasenya. En canvi, la resta d'operacions (com consultar usuaris, modificar rols o gestionar drets d'accés) estan restringides a usuaris amb permisos elevats, com ara els administradors. Aquesta segmentació garanteix que els usuaris només tinguin accés a les dades que els corresponen i que el sistema mantingui una separació estricta entre privilegis.

| users  |                           |                     | ^   |
|--------|---------------------------|---------------------|-----|
| GET    | /api/v1/users/            | Read Users          | 🔒 ↓ |
| POST   | /api/v1/users/            | Create User         | 🔒 ↓ |
| GET    | /api/v1/users/students    | Read Students Users | 🔒 ↓ |
| GET    | /api/v1/users/me          | Read User Me        | 🔒 ↓ |
| DELETE | /api/v1/users/me          | Delete User Me      | 🔒 ↓ |
| PATCH  | /api/v1/users/me          | Update User Me      | 🔒 ↓ |
| PATCH  | /api/v1/users/me/password | Update Password Me  | 🔒 ↓ |
| POST   | /api/v1/users/signup      | Register User       | ↓   |
| GET    | /api/v1/users/{user_niub} | Read User By Id     | 🔒 ↓ |
| PATCH  | /api/v1/users/{user_niub} | Update User         | 🔒 ↓ |
| DELETE | /api/v1/users/{user_niub} | Delete User         | 🔒 ↓ |

Figura 9. Endpoints de la ruta users

Un dels blocs més complets és el que fa referència a /courses, on es gestionen tant els cursos com la relació amb els seus participants. Els professors responsables de cada curs, així com els administradors, poden crear, modificar i eliminar cursos. També poden afegir o eliminar alumnes de manera individual, o bé descarregar plantilles en format CSV o XLSX per gestionar la incorporació massiva d'alumnes. A més, es poden obtenir llistats complets de participants, diferenciats per rol, així com consultar totes les pràctiques vinculades a un curs. Cal destacar que si un alumne és eliminat d'un curs, també se l'elimina automàticament de totes les pràctiques associades a aquest curs, mantenint així la coherència de les relacions de dades.

| course |                                             |                            | ^ |
|--------|---------------------------------------------|----------------------------|---|
| GET    | /api/v1/courses/                            | Read Courses               | 🔒 |
| POST   | /api/v1/courses/                            | Create Course              | 🔒 |
| GET    | /api/v1/courses/search                      | Search Courses             | 🔒 |
| GET    | /api/v1/courses/me                          | Read My Courses            | 🔒 |
| GET    | /api/v1/courses/me/recent                   | Read My Recent Courses     | 🔒 |
| GET    | /api/v1/courses/{course_id}                 | Read Course                | 🔒 |
| PUT    | /api/v1/courses/{course_id}                 | Update Course              | 🔒 |
| DELETE | /api/v1/courses/{course_id}                 | Delete Course              | 🔒 |
| GET    | /api/v1/courses/{course_id}/users           | Read Course Users          | 🔒 |
| GET    | /api/v1/courses/{course_id}/teachers        | Read Course Teachers       | 🔒 |
| GET    | /api/v1/courses/{course_id}/practices       | Read Course Practices      | 🔒 |
| POST   | /api/v1/courses/{course_id}/students/{niub} | Add Student By Niub        | 🔒 |
| DELETE | /api/v1/courses/{course_id}/students/{niub} | Delete Student From Course | 🔒 |
| PATCH  | /api/v1/courses/me/{course_id}/access       | Update Course Last Access  | 🔒 |
| GET    | /api/v1/courses/students-template/csv       | Get Students Template Csv  | 🔒 |
| GET    | /api/v1/courses/students-template/xlsx      | Get Students Template Xlsx | 🔒 |

Figura 10. Endpoints de la ruta courses

La ruta `/practices` permet gestionar pràctiques de programació dins dels cursos. Igual que amb els cursos, només els professors vinculats al curs i els administradors poden accedir a les operacions de creació, actualització i eliminació de pràctiques. A més, els professors poden consultar el llistat d'usuaris assignats a cada pràctica, accedir a l'estat de les entregues, i descarregar els arxius relacionats amb les entregues dels estudiants. Aquests arxius es poden descarregar de manera individual —per estudiant concret— o de forma agrupada, incloent-hi tant els fitxers d'entrega com els fitxers d'autocorrecció associats a la pràctica. D'altra banda, els estudiants tenen accés als seus propis arxius de submissió i poden descarregar-los o revisar-ne l'estat i el feedback rebut.

| practices |                                                                   |                                     | ^ |
|-----------|-------------------------------------------------------------------|-------------------------------------|---|
| GET       | /api/v1/practices/                                                | Read Practices                      | 🔒 |
| POST      | /api/v1/practices/                                                | Create Practice                     | 🔒 |
| GET       | /api/v1/practices/search                                          | Search Courses                      | 🔒 |
| GET       | /api/v1/practices/me                                              | Read My Practices                   | 🔒 |
| GET       | /api/v1/practices/me/corrected                                    | Read My Corrected Practices         | 🔒 |
| GET       | /api/v1/practices/me/uncorrected                                  | Read My Uncorrected Practices       | 🔒 |
| GET       | /api/v1/practices/{practice_id}                                   | Read Practice                       | 🔒 |
| PUT       | /api/v1/practices/{practice_id}                                   | Update Practice                     | 🔒 |
| DELETE    | /api/v1/practices/{practice_id}                                   | Delete Practice                     | 🔒 |
| GET       | /api/v1/practices/{practice_id}/users                             | Read Practice Users                 | 🔒 |
| GET       | /api/v1/practices/{practice_id}/course                            | Read Practice Course                | 🔒 |
| GET       | /api/v1/practices/{practice_id}/correction-files-info             | Read Practice Correction Files Info | 🔒 |
| GET       | /api/v1/practices/{practice_id}/submission-file-info              | Read Practice File Info             | 🔒 |
| GET       | /api/v1/practices/{practice_id}/users/{niub}/submission-file-info | Read User Submission File Info      | 🔒 |
| GET       | /api/v1/practices/{practice_id}/{user_niub}                       | Read Practice Student               | 🔒 |
| POST      | /api/v1/practices/{practice_id}/upload                            | Upload Practice File                | 🔒 |
| DELETE    | /api/v1/practices/{practice_id}/submission/{user_niub}            | Delete Practice Submission          | 🔒 |
| GET       | /api/v1/practices/{practice_id}/download/me                       | Download My Files                   | 🔒 |
| GET       | /api/v1/practices/{practice_id}/download/all                      | Download All Files                  | 🔒 |
| GET       | /api/v1/practices/{practice_id}/download/{user_niub}              | Download User Files                 | 🔒 |
| POST      | /api/v1/practices/send-practice-data/{practice_id}/{niub}         | Send Practice Data                  | 🔒 |

Figura 11. Endpoints de la ruta practices

Així mateix, les rutes `/courses` i `/practices` tenen un endpoint per cercar els cursos o pràctiques els quals es troben inscrits, tot i que, els administradors poden cercar qualsevol.

En general, la nova API ha estat desenvolupada amb una atenció especial a la seguretat i la gestió de rols, assegurant que cada acció només pugui ser executada pels usuaris autoritzats. Aquesta estratègia d'autenticació i autorització, combinada amb una estructura clara i modular d'endpoints, permet que el sistema sigui escalable, segur i fàcil d'ampliar. Els endpoints, juntament amb la documentació automàtica generada per FastAPI (mitjançant Swagger UI i ReDoc), faciliten la comprensió i el test de l'API per part dels desenvolupadors, millorant així l'experiència de desenvolupament i integració.

#### 4.1.6. Gestió de dependències

Per garantir una gestió de dependències moderna, eficient i reproduïble, s'ha optat per utilitzar uv com a gestor de paquets i entorns virtuals en lloc del tradicional pip.

uv és una eina escrita en Rust, que ofereix temps d'instal·lació molt més ràpids, millor resolució de versions i una millor integració amb `pyproject.toml` com a font única de configuració. A més, genera automàticament un fitxer `uv.lock` que assegura entorns de desplegament consistents i predicibles, especialment útils en escenaris de contenidorització amb Docker com l'utilitzat pel backend desplegat, Azure Web App for Containers. Aquesta decisió ha contribuït a millorar l'eficiència del procés de desenvolupament i desplegament del backend.

### 4.2. Implementació del Worker

El worker, com ja s'ha mencionat anteriorment a la memòria, és un component clau dins de l'arquitectura distribuïda de Golem XIV. La seva responsabilitat principal és consumir missatges de correcció de pràctiques enviats pel backend mitjançant una cua de missatgeria RabbitMQ, i processar-los de manera eficient mitjançant un sistema asíncron i concurrent.

#### 4.2.1. Refactorització i processament concurrent

A diferència de versions anteriors del projecte, on el `worker.py` funcionava de manera seqüencial, és a dir, només podia processar una pràctica a la vegada, en aquesta nova versió s'ha dut a terme una refactorització completa per introduir concurrència i asincronia. Això ha permès multiplicar el rendiment i escalabilitat, ja que el worker ara és capaç de processar diversos missatges simultàniament, amb un límit dinàmic que s'adapta al maquinari disponible (actualment definit com a nombre de nuclis de CPU  $\times$  4). Per exemple, en una màquina amb 4 nuclis desplegada a Azure Container App, el sistema pot arribar a processar fins a 16 pràctiques de manera paral·lela. Exemple real d'inicialització del worker:

```
02-06-2025 16:57:53 INFO@worker.py >> Using event loop:
```

```
<asyncio.windows_events.WindowsSelectorEventLoopPolicy object at 0x000002E9EE080170>  
02-06-2025 16:57:53 WARNING@worker.py >> Starting worker...  
02-06-2025 16:57:55 INFO@worker.py >> Worker started with capacity for 64 concurrent  
tasks  
02-06-2025 16:57:55 WARNING@worker.py >> Waiting for messages. To exit press CTRL+C
```

Per implementar aquesta millora s'han utilitzat les llibreries `asyncio` i `aio_pika`, que permeten treballar amb operacions de I/O no bloquejants i consumir missatges de RabbitMQ de forma asíncrona. La concurrència es gestiona a través d'un semàfor (`asyncio.Semaphore`), que limita el nombre de tasques actives per evitar la sobrecàrrega del sistema. Això permet optimitzar l'ús dels recursos del servidor i mantenir l'estabilitat del servei fins i tot amb un gran volum d'entregues simultànies.

#### 4.2.2. Comunicació amb el backend i el servidor RPC

El worker no manipula directament fitxers, sinó que processa metadades i rutes de directori. Quan rep un missatge del backend, aquest conté informació estructurada sobre la pràctica entregada per l'estudiant: identificadors, llenguatge de programació, i sobretot els directoris al servidor SFTP on es troben els fitxers d'entrega i d'autocorrecció. Aquestes rutes ja han estat generades pel backend i són accessibles pel servidor RPC.

El worker a l'hora de processar una pràctica el primer que fa és actualitzar l'estat de la pràctica a CORRECTING (corregint-se) a la base de dades, per reflectir que el procés de correcció està en marxa. Tot seguit, crea una instància del client RPC, que s'encarrega de formatar les dades i transmetre-les al servidor RPC a través de RabbitMQ. Aquest servidor, desplegat en una màquina de la facultat, és l'encarregat d'accedir als fitxers a partir dels directoris proporcionats i executar la correcció. Exemple real de processament d'una pràctica al worker:

```
02-06-2025 02:06:13 INFO@worker.py >> Received message: {"program":"Programacio_II",  
"year":"2024-2025", "task":"MVC", "task_id":"de9af56b-af1d-4ea3-a9aa-1e18e9ba25b2",  
"student_id":"niub20601353", "language":"java",  
"student_dir":"/uploads/corrections_files/students/2024-2025/Programacio_II/MVC/niub2060  
1353", "teacher_dir":"/uploads/corrections_files/teachers/2024-2025/Programacio_II/MVC"}  
02-06-2025 02:06:13 INFO@worker.py >> Requesting  
02-06-2025 02:06:13 INFO@rpc_client.py >> Connected and ready  
02-06-2025 02:06:13 INFO@rpc_client.py >> Sending request to 'java' queue:
```

```
Programacio_II,2024-2025,MVC,niub20601353,  
/uploads/corrections_files/students/2024-2025/Programacio_II/MVC/niub20601353,/uploads/c  
orrections_files/teachers/2024-2025/Programacio_II/MVC  
02-06-2025 02:06:13 INFO@rpc_client.py >> Request sent to java, waiting for response  
(correlation_id: cdb88cc8-a86f-4b5f-882b-d8c407812bb3)
```

Quan el servidor RPC retorna el resultat, el worker ho rep de manera asíncrona i actualitza la base de dades, canviant l'estat de la pràctica a CORRECTED (corregit) o REJECTED (rebutjat) i desant la correcció obtinguda, produint-se tot aquest flux de forma desacoblada i optimitzada, assegurant una resposta fluida i escalable.

#### 4.2.3. Gestió asíncrona de base de dades

Per realitzar operacions de lectura i escriptura a la base de dades de manera no bloquejant, el worker utilitza sessions asíncrones gràcies a la llibreria asíncrona de SQLAlchemy (`sqlalchemy.ext.asyncio`). Aquesta implementació és coherent perquè permet mantenir el paradigma d'asincronia en tot el flux de processament.

Cada vegada que el worker processa un missatge, obre una sessió temporal asíncrona amb la base de dades, com he mencionat abans, per actualitzar l'estat de la pràctica durant el processament i posteriorment desar els resultats retornats pel servidor RPC.

Exemple d'usos reals dins del worker:

- Canvi d'estat de la pràctica (de *Entregada* a *Corregint-se*, *Corregida*, etc.).
- Inserció de resultats de correcció rebuts del servidor RPC.

En definitiva, aquesta integració asíncrona amb la base de dades resulta essencial per mantenir la fluïdesa i capacitat de resposta del sistema sota càrrega.

#### 4.2.4. Controls de fiabilitat amb RabbitMQ

Per augmentar la resiliència del sistema, fent el sistema més robust i tolerant a errors, s'han implementat diversos mecanismes de control sobre la cua de missatges de RabbitMQ:

- **Reintents limitats** mitjançant l'ús del header x-death, que permet detectar i limitar el nombre de vegades que un missatge pot tornar-se a intentar processar, per exemple tres cops.
- **Dead Letter Queue (DLQ)**: si un missatge excedeix el nombre de reintents permès, es redirigeix automàticament a una cua de missatges morts, evitant bucles infinits i bloquejos del sistema.
- **TTL (Time To Live)** que estableix un límit de temps (per exemple, 5 segons) abans que el missatge torni a la cua principal de correcció, cosa que permet un control eficient dels temps de reexecució i evitar bloquejos infinits.

A més, per facilitar el monitoratge i el diagnòstic de problemes, s'ha implementat un sistema de logging (registre) detallat al worker, que registra cada pas del procés amb diferents nivells de severitat (INFO, WARNING, ERROR). Per exemple:

- Quan es rep un missatge: registre del contingut i del context de la pràctica.
- Quan es connecta amb el servidor RPC: confirmació i estat de la connexió.
- Si es produeix una fallada: captura de l'error i nombre de reintents acumulats.
- Si un missatge és enviat a la DLQ: registre explícit indicant que s'ha superat el límit permès de reintents.

Exemple real dels diferents logs que es poden trobar worker:

```
30-05-2025 02:08:37 INFO@worker.py >> Response from RPC server: Error: Return code 1.
30-05-2025 02:08:37 ERROR@worker.py >> Error processing message: Error from RPC server:
Error: Return code 1.
30-05-2025 02:08:37 WARNING@worker.py >> Failed practice correction for NIUB
niub20601353 and practice MVC.
30-05-2025 02:08:37 WARNING@worker.py >> Retrying (3/3)...
30-05-2025 02:08:42 INFO@worker.py >> Received {"subject": "Programacio_II", "year":
"2024-2025", "task": "MVC", "task_id": "de9b3f8e-e315-4b1f-9516-8511c90e6c3c",
"student_id": "niub20601353", "language": "java", "student_dir":
"/uploads/corrections_files/students/2024-2025/Programacio_II/MVC/niub20601353",
"teacher_dir": "/uploads/corrections_files/teachers/2024-2025/Programacio_II/MVC"}
30-05-2025 02:08:42 INFO@rpc_client.py >> Requesting
30-05-2025 02:08:42 INFO@rpc_client.py >> Connected and ready
30-05-2025 02:08:43 INFO@rpc_client.py >> Sending request to 'java' queue:
Programacio_II,2024-2025,MVC,niub20601353,/uploads/corrections_files/students/2024-2025/
Programacio_II/MVC/niub20601353,/uploads/corrections_files/teachers/2024-2025/Programaci
o_II/MVC
30-05-2025 02:08:43 INFO@rpc_client.py >> Request sent to java, waiting for response
```

```
(correlation_id: 49c58c69-dfee-4565-9ac4-61bd6a2ae86f)
30-05-2025 02:08:43 INFO@rpc_client.py >> Response received in RPC client: Error: Return
code 1.
30-05-2025 02:08:43 INFO@rpc_client.py >> Closing RPC connection
30-05-2025 02:08:43 INFO@rpc_client.py >> RPC connection closed
30-05-2025 02:08:43 INFO@worker.py >> Response from RPC server: Error: Return code 1.
30-05-2025 02:08:43 ERROR@worker.py >> Error processing message: Error from RPC server:
Error: Return code 1.
30-05-2025 02:08:43 WARNING@worker.py >> Failed practice correction for NIUB
niub20601353 and practice MVC.
30-05-2025 02:08:43 WARNING@worker.py >> Permanent failure after 3 attempts. Sending to
DLQ.
```

A l'exemple es mostra un cas on, després de tres intents de correcció sense èxit, el sistema informa de manera clara la fallada, notifica l'error rebut del servidor RPC (Return code 1) i redirigeix el missatge a la cua de DLQ. Aquest seguiment en temps real resulta essencial per detectar patrons de fallada, identificar pràctiques problemàtiques i mantenir la integritat del sistema.

En conjunt, aquests mecanismes garanteixen que el sistema es mantingui estable, tolerant a errors i fàcil de monitorar, fins i tot en escenaris amb errors de comunicació, dades incompletes o fallades temporals al servidor RPC.

#### 4.2.5. Gestió de dependències

Igual que el backend, el worker utilitza uv com a gestor de dependències, cosa que ha millorat notablement el temps d'instal·lació, la consistència dels entorns i la reproducció fiable dins de contenidors Docker.

### 4.3. Implementació del Frontend i Experiència d'Usuari (UI/UX)

Un dels objectius principals d'aquesta tercera versió de Golem XIV ha estat la renovació completa del frontend, tant des del punt de vista tècnic com visual. S'ha desenvolupat una interfície totalment nova amb Next.js, TailwindCSS i HeroUI, pensada per oferir una experiència clara, ràpida i intuïtiva tant per a estudiants com per a professors.

L'anterior interfície, limitada i visualment arcaica, ha estat reemplaçada per una nova aplicació moderna, dissenyada des de zero amb una clara orientació a la usabilitat (UX) i a una presentació estètica coherent (UI).

El resultat ha sigut una plataforma amb una experiència més rica i interactiva, adaptada a les necessitats reals dels usuaris finals, estudiants i professors.

#### 4.3.1. Estructura de navegació i arquitectura visual

La nova interfície de Golem XIV ha estat construïda sobre una arquitectura de navegació clara i funcional, basada en dues àrees principals: una barra lateral de navegació (SideNav) i una barra superior (Navbar). Aquesta combinació proporciona accés ràpid a les funcionalitats clau i millora l'eficiència de l'usuari a l'hora de moure's per l'aplicació.

La SideNav es divideix en dues seccions:

- **General:** amb accés directe a les seccions principals com el Dashboard, Cursos, Pràctiques i Calendari.
- **Configuracions:** amb accés a la configuració del perfil d'usuari i a l'opció de tancar la sessió.

Aquesta barra lateral es pot expandir o col·lapsar segons les preferències de l'usuari, permetent aprofitar millor l'espai en pantalles petites o centrant l'atenció en el contingut principal.

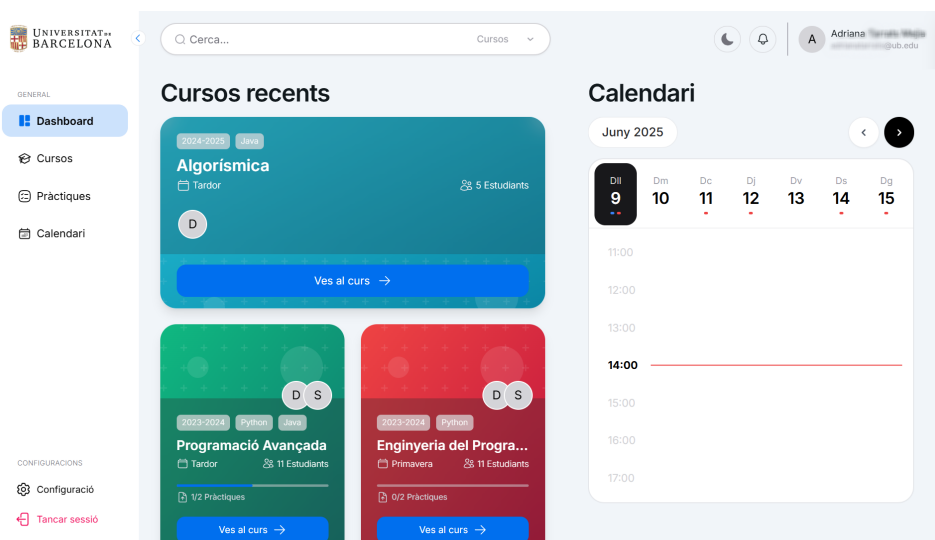


Figura 12. SideNav expandida

La Navbar superior complementa la navegació lateral i afegeix funcionalitats essencials:

- Desapareix a l'hora de fer scroll cap avall, i torna a aparèixer quan fas scroll cap a dalt. Gràcies a això l'usuari té més espai per visualitzar el contingut de la pàgina.
- Un buscador global contextual, amb selector per alternar entre la cerca de cursos o pràctiques.
- Un commutador de tema que permet passar de mode clar a mode fosc i viceversa.
- Una icona de notificacions (campana) amb badge, que indica el nombre de notificacions no llegides.
- Un bloc d'identificació de l'usuari amb l'avatar, nom complet i adreça electrònica.

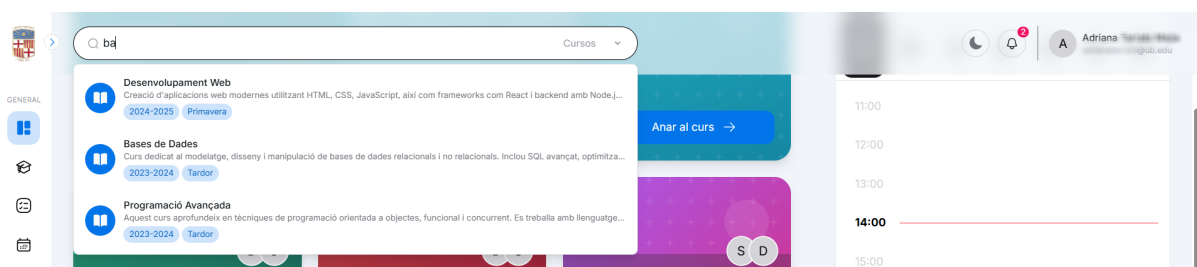


Figura 13. Detall de la Navbar

#### 4.3.2. Pàgina principal (Dashboard)

La pàgina principal ha estat redissenyada amb una clara orientació a la productivitat i el resum visual de la informació rellevant per a l'usuari. Aquesta pàgina és comuna per a estudiants i professors, però mostra continguts adaptats a l'usuari autenticat. El dashboard inclou:

- Una secció amb els cursos recents, presentats com targetes visuals. El nombre de targetes visibles i la seva amplada s'adapta a la resolució de la pantalla.
- Un calendari setmanal navegable, amb marques que indiquen dies amb pràctiques, i una marca especial pel dia actual, mostrant solament un rang d'hores del dia relatiu a l'hora actual.

- Una taula de pràctiques properes, amb columnes com pràctica, curs, professor, estat, data de venciment i accions, amb possibilitat de filtrar per nom, estat o rang de dates.

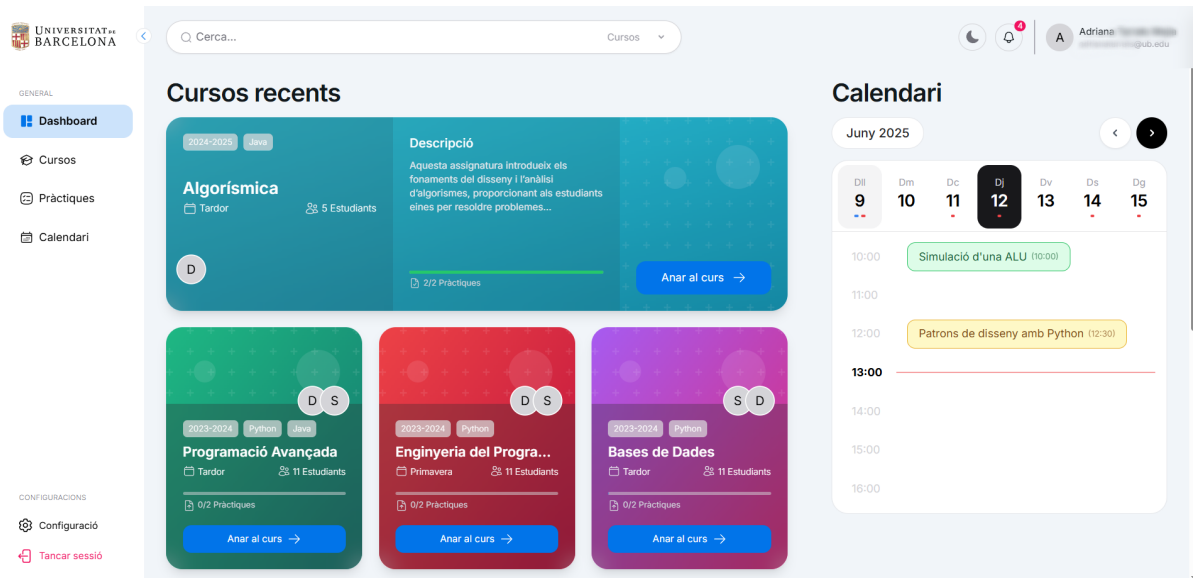


Figura 14. Vista general del dashboard amb cursos recents i calendari

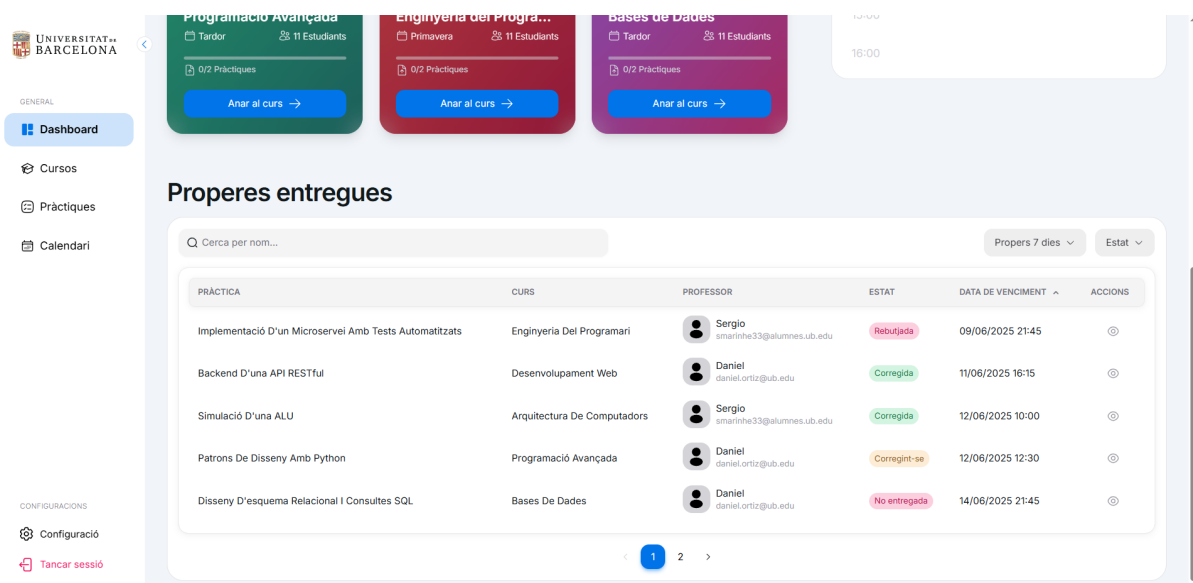


Figura 15. Vista general del dashboard amb properes entregues

En el cas que l'usuari autenticat sigui un professor o administrador la taula de pràctiques properes mostrarà el mateix que pels estudiants, però amb la diferència de què el filtre i columna estat no estaran i tindrà una acció més, un accés directe a editar la pràctica.

**Properes entregues**

Q Cerca per nom...

Propers 30 dies ▾

| PRÀCTICA                                               | CURS                        | PROFESSOR                                                                                                             | DATA DE VENCIMENT | ACCIONS                                                                                                                                                                 |
|--------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aplicació CRUD Amb SQLite                              | Bases De Dades              |  Sergio<br>smarinhe33@alumnes.ub.edu | 05/06/2025 22:45  |   |
| Implementació D'un Microservei Amb Tests Automatitzats | Enginyeria Del Programari   |  Sergio<br>smarinhe33@alumnes.ub.edu | 09/06/2025 21:45  |   |
| Backend D'una API RESTful                              | Desenvolupament Web         |  Daniel<br>daniel.ortiz@ub.edu       | 11/06/2025 16:15  |   |
| Simulació D'una ALU                                    | Arquitectura De Computadors |  Sergio<br>smarinhe33@alumnes.ub.edu | 12/06/2025 10:00  |   |
| Patrons De Disseny Amb Python                          | Programació Avançada        |  Daniel<br>daniel.ortiz@ub.edu       | 12/06/2025 12:30  |   |

< 1 2 3 >

Figura 16. Vista general del dashboard amb propers entregues

### 4.3.3. Pàgina de cursos

A la vista de cursos, l'usuari pot consultar tots els cursos als quals està assignat, visualitzats en forma de targetes. Cada targeta ofereix:

- Avatars dels professors assignats al curs.
- Etiquetes amb el curs acadèmic i llenguatges de programació de les pràctiques del curs.
- El semestre i el nombre total d'estudiants al curs.
- Una barra de progrés sobre el nombre de pràctiques corregides.

UNIVERSITAT DE BARCELONA

Q Cerca...

Cursos ▾

Dashboard > Cursos

**Els meus cursos**

Explora els teus cursos i accedeix a les seves pràctiques

Q Cerca per nom de curs...

\$1 Per Nom ▾

Curs Acadèmic ▾

</> Llenguatge ▾

2024-2025 Java

**Algorismica**

Tardor 5 Estudiants

2/2 Pràctiques

Anar al curs →

2023-2024 Python

**Arquitectura de Computadors**

Primavera 11 Estudiants

1/2 Pràctiques

Anar al curs →

2023-2024 Python

**Bases de Dades**

Tardor 11 Estudiants

0/2 Pràctiques

Anar al curs →

2024-2025 Python

**Ciència de Dades**

Primavera 11 Estudiants

0/2 Pràctiques

Anar al curs →

2024-2025 Python

**Desenvolupament Web**

Primavera 12 Estudiants

1/2 Pràctiques

Anar al curs →

2023-2024 Python

**Enginyeria del Programari**

Primavera 11 Estudiants

0/2 Pràctiques

Anar al curs →

2025-2026 Python

**Inteligència Artificial**

Tardor 12 Estudiants

2/2 Pràctiques

Anar al curs →

2023-2024 Python Java

**Programació Avançada**

Tardor 11 Estudiants

1/2 Pràctiques

Anar al curs →

CONFIGURACIONS

Configuració

Tancar sessió

Figura 17. Pàgina de cursos amb targetes per a estudiants

En el cas que l'usuari sigui professor o administrador, es mostra un botó "Editar" a cada curs, així com un botó a dalt a la dreta de "Afegir nou curs" per crear-ne de nous.

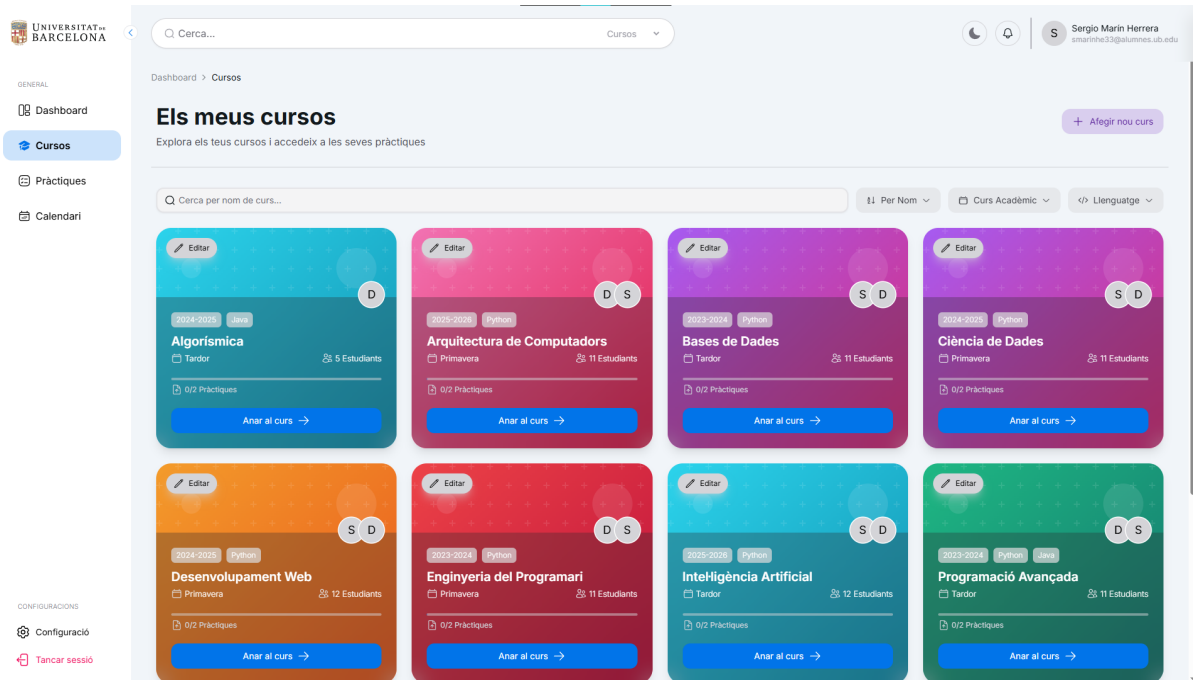


Figura 18. Pàgina de cursos amb targetes i botons d'acció per a professors i admins

#### 4.3.4. Pàgina detall d'un curs

Aquesta pàgina ofereix una vista centralitzada del curs seleccionat. Aquí es poden consultar:

- Informació detallada del curs: nom, descripció, any acadèmic, semestre i professor del curs.
- Les pràctiques del curs, amb estats i terminis.
- Els participants, diferenciats per pestanyes (tots, professors, estudiants), amb cercador per nom i opció d'enviar missatges.

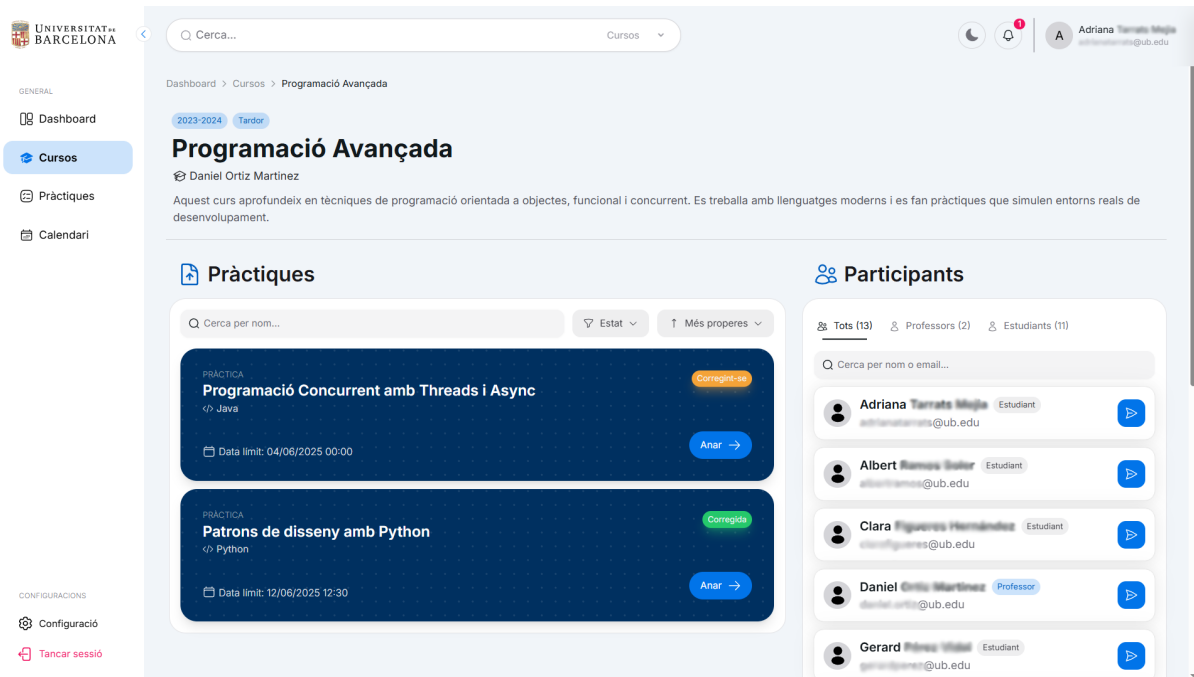


Figura 19. Vista detallada d'un curs per a estudiants

Els professors a més tenen accés a accions com gestionar pràctiques, editar el curs i afegir o eliminar estudiants.

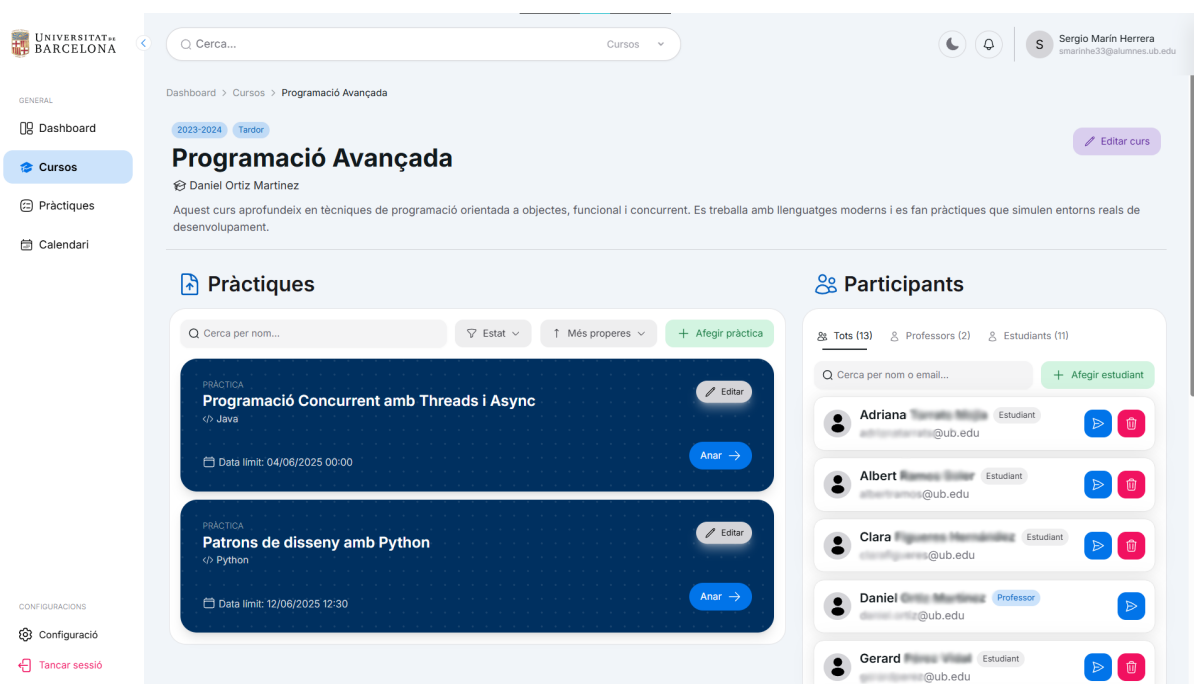


Figura 20. Vista detallada d'un curs per a professors i admins

### 4.3.5. Pàgina de pràctiques

La pàgina de pràctiques permet als usuaris consultar totes les pràctiques agrupades per estat mitjançant pestanyes: No entregades, Entregades, Corregint-se, Corregides, Rebutjades, i una pestanya inicial amb Totes. La pestaña principal és la de Corregides i es, per tant, la que es mostra quan entres a la pàgina.

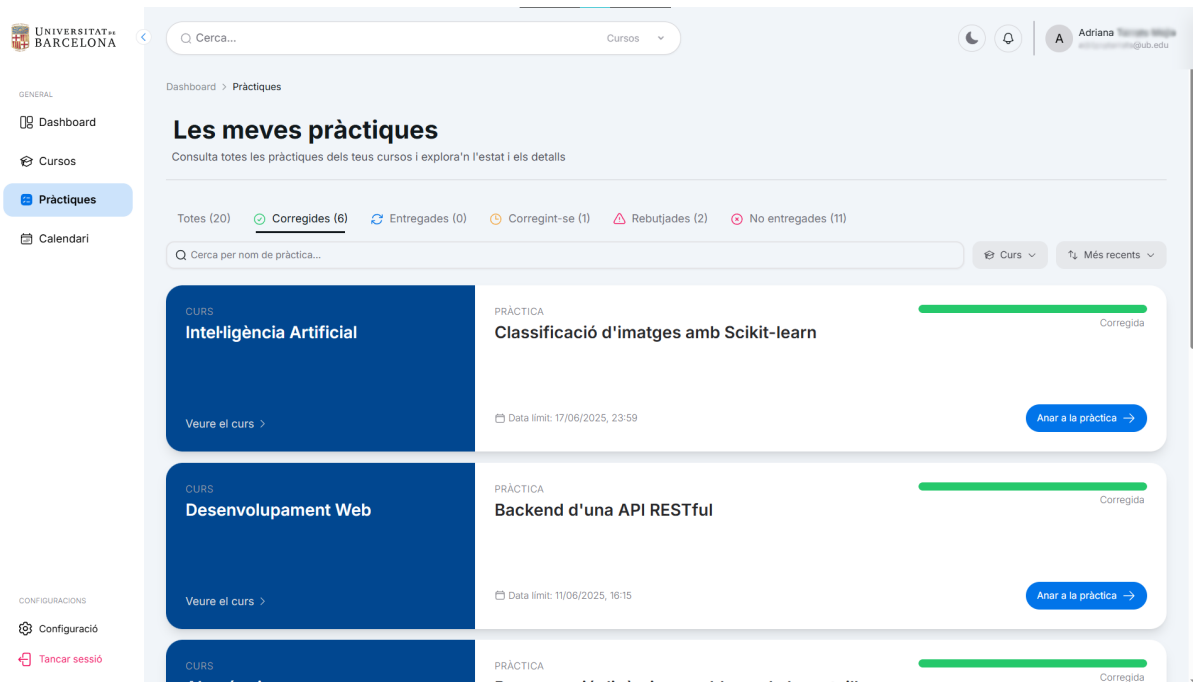


Figura 21. Pàgina de pràctiques amb pestanyes i estat visual

### 4.3.6. Pàgina detall d'una pràctica

La vista detallada d'una pràctica proporciona una experiència completa i adaptada al rol de l'usuari, ja sigui estudiant o professor. Aquesta pàgina centralitza tota la informació rellevant i les accions possibles associades a una pràctica concreta.

Per a tots els usuaris, s'hi pot consultar:

- La informació general de la pràctica, com ara el nom, descripció, data límit d'entrega, llenguatge de programació i la data de lliurament, si existeix.
- La interfície de pujada d'arxius, amb funcionalitat de drag & drop per facilitar l'entrega.

- L'estat actual de la pràctica (No entregada, Entregada, Corregint-se, Corregida, Rebutjada).
- El resultat de la correcció automàtica, un cop finalitzat el procés.
- L'opció de descarregar els arxius ja entregats, tant per revisar-los com per conservar-ne una còpia.

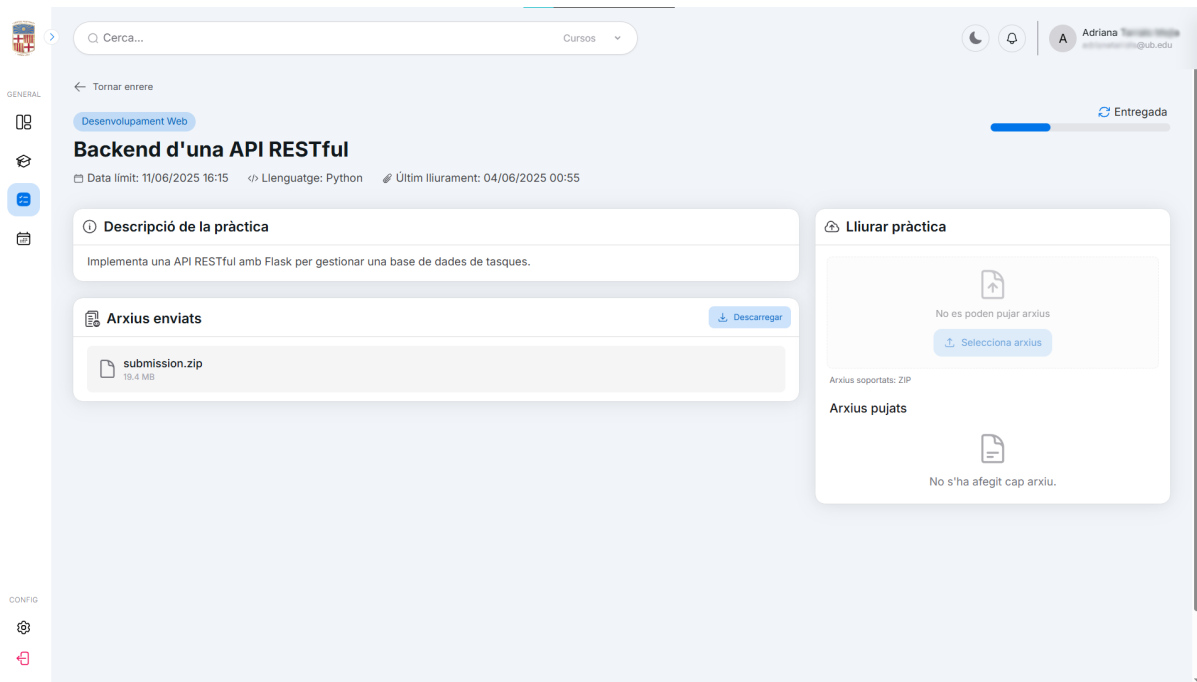


Figura 22. Vista d'una pràctica per a un estudiant que l'ha entregat

Per als professors (o usuaris amb permisos d'administració), la pàgina incorpora funcionalitats addicionals:

- Visualització del llistat d'alumnes que han entregat la pràctica, amb accés individualitzat a cadascuna de les seves entregues.
- Per a cada entrega es poden consultar les dates de lliurament, estat actual i arxius enviats.
- Si l'estat de la pràctica no és "Corregint-se", el professor pot:
  - Eliminar l'entrega de l'alumne, en cas d'errors o per requeriment acadèmic.
  - Forçar una nova correcció, reiniciant el procés d'autocorrecció des del servidor RPC.

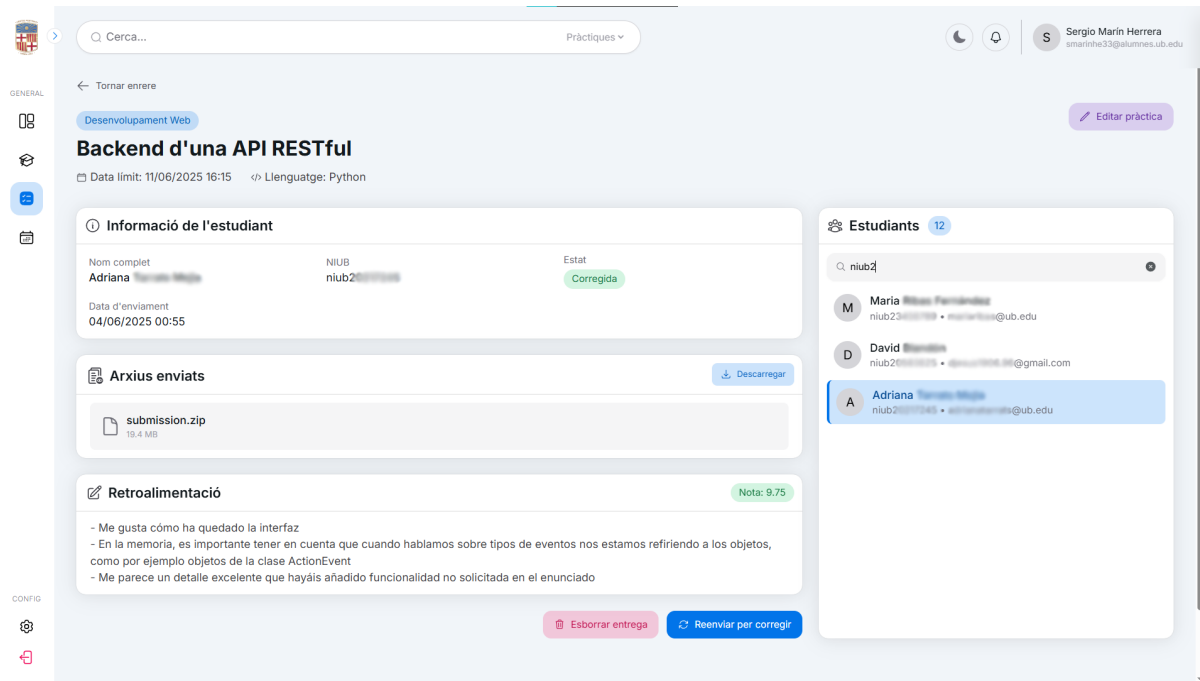


Figura 23. Vista d'una pràctica per a un professor que ha seleccionat un estudiant amb una pràctica corregida

#### 4.3.7. Gestió de cursos i pràctiques

Els usuaris amb permisos (professors o administradors) poden crear i editar cursos i pràctiques mitjançant una interfície clara i intuïtiva. Quan es fa clic als botons "Afegir curs" o "Afegir pràctica", es desplega un drawer lateral a la dreta de la pantalla que conté un formulari interactiu i validat.

En el cas dels cursos, el formulari permet:

- Escollir un color identificatiu per a la targeta del curs (seguint la paleta comuna de la plataforma).
- Introduir informació bàsica com el nom del curs, una descripció breu, l'any acadèmic i el semestre.
- Pujar el full de càlcul amb el llistat d'alumnes que es vol afegir mitjançant arrossegat i deixar anar (drag & drop) o botó.

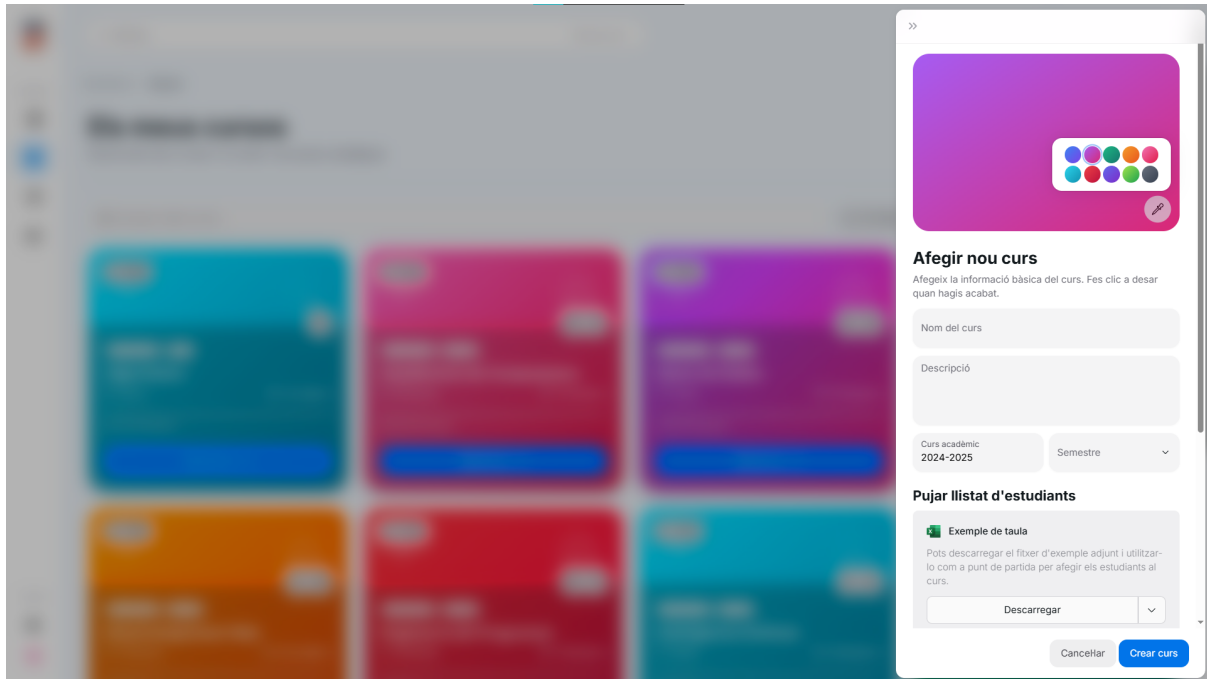


Figura 24. Drawer amb el formulari de creació d'un curs

En el cas de les pràctiques, a més dels camps anteriors (excepte el color, que depèn del curs, i el llistat d'alumnes), es permet:

- Pujar els fitxers d'autocorrecció corresponents a la pràctica mitjançant arrossegar i deixar anar (drag & drop) o botó.
- Establir la data límit d'entrega, que es reflecteix automàticament en els components de l'aplicació com el calendari o la dashboard.
- Indicar el llenguatge de programació específic per a la pràctica.

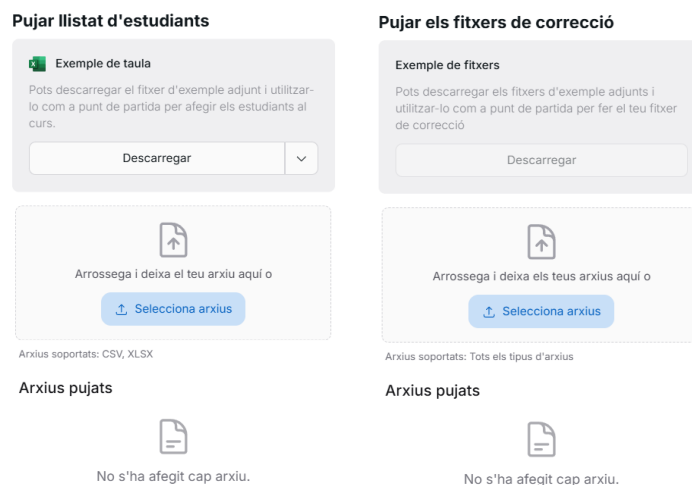


Figura 25. Secció de pujada de fitxers d'un curs i d'una pràctica respectivament

A més del mode de creació, la plataforma també incorpora un mode d'edició, accessible a través del botó "Editar" que apareix a cada targeta de curs o pràctica per als usuaris autoritzats, o dintre de la pàgina detall del curs o la pràctica. Aquest mode reutilitza el mateix drawer i formulari, però amb els camps precarregats amb la informació existent, permetent actualitzar qualsevol valor de manera ràpida i consistent. A més inclou l'opció de poder d'eliminar el curs o la pràctica.

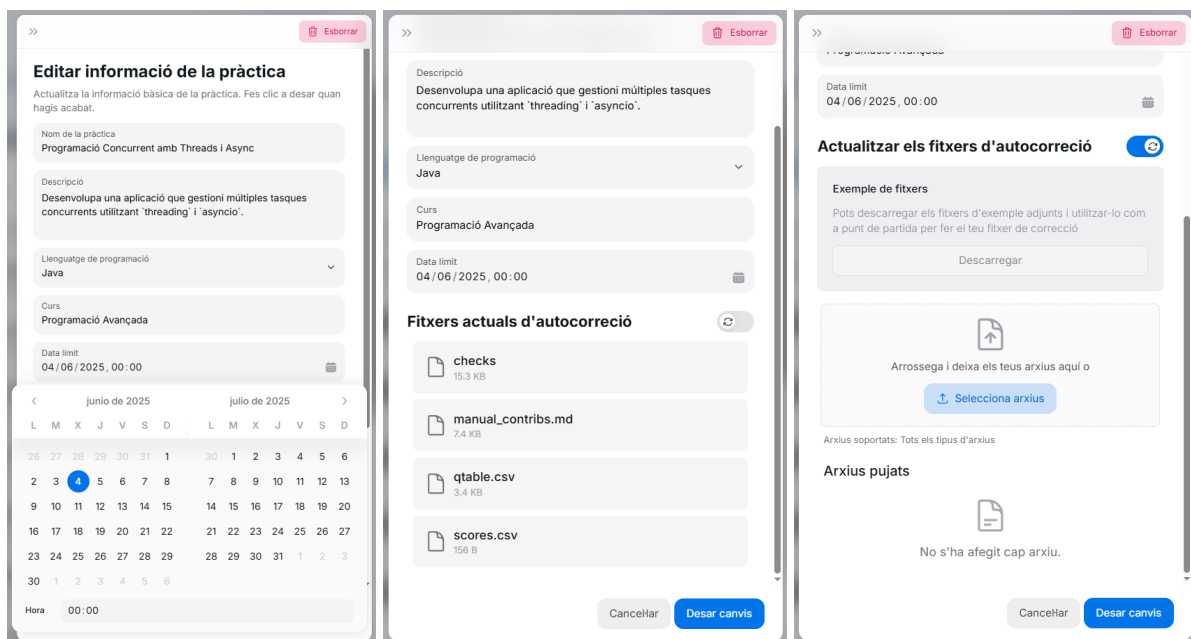


Figura 26. Vista d'una pràctica en mode edició

#### 4.3.7. Calendari de pràctiques

La secció calendari mostra un calendari en format setmanal on cada entrega de pràctica es representa amb un color segons el seu estat, esquema de colors ja vist en anteriors parts de la web:

- No entregat
- Entregat
- Corregint-se
- Corregit
- Rebutjada

Quan l'usuari fa clic sobre una entrega de pràctica programada, es mostra un popover amb informació detallada i un botó per accedir directament a la pràctica com a la Figura 27.

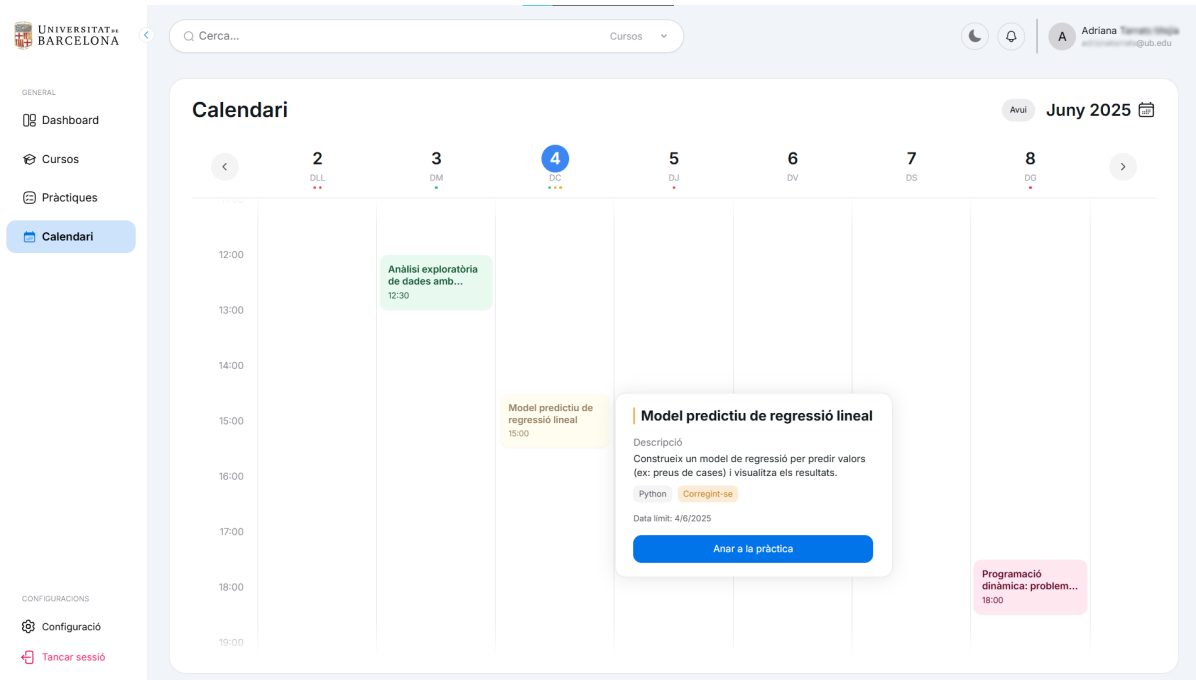


Figura 27. Calendari amb pràctiques de diferents estats i popover obert d'una entrega

#### 4.3.8. Configuració de l'usuari

Des del botó de configuració de la SideNav, s'obre un modal amb pestanyes:

- **Perfil:** per modificar nom, cognoms i correu, i visualitzar els rols de l'usuari.
- **Contrasenya:** amb doble entrada de nova contrasenya i validació de l'actual.

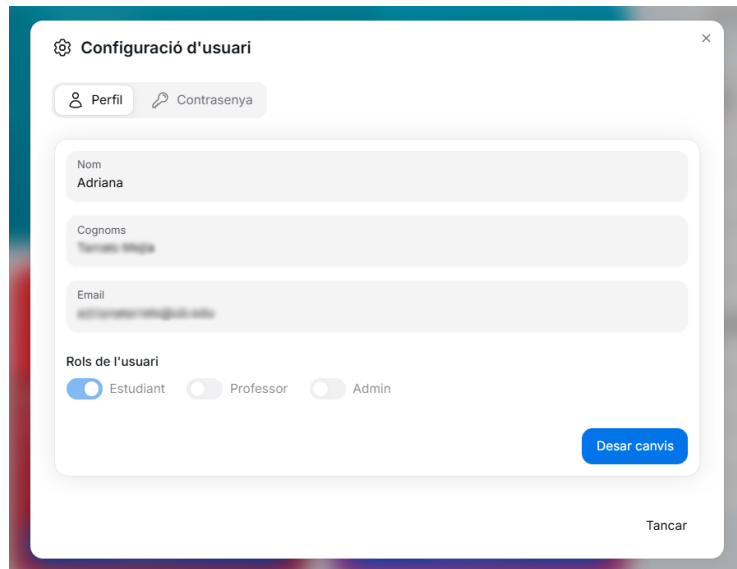


Figura 28. Modal de configuració d'usuari amb pestanyes Perfil i Contrasenya

- **Administració:** per a usuaris administradors, amb accés a la gestió de rols i restabliment de contrasenyes d'altres usuaris.

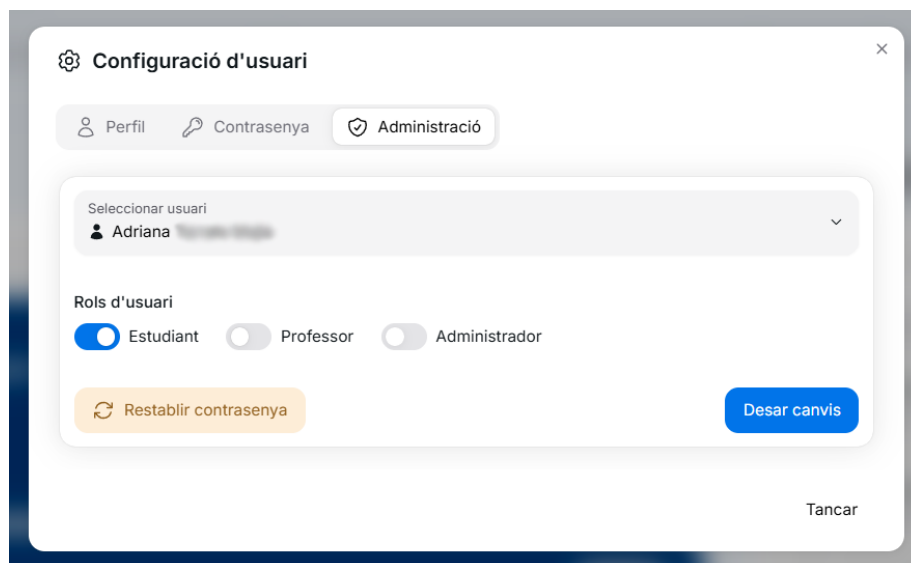


Figura 29. Pestanya d'Administració per a gestió d'usuaris

#### 4.3.9. Sistema de notificaciones en temps real

El sistema de notificaciones és un dels punts més innovadors de la plataforma, implementat mitjançant Supabase Realtime, que permet capturar canvis en l'estat de les pràctiques de l'usuari de la base de dades en temps real mitjançant WebSocket.

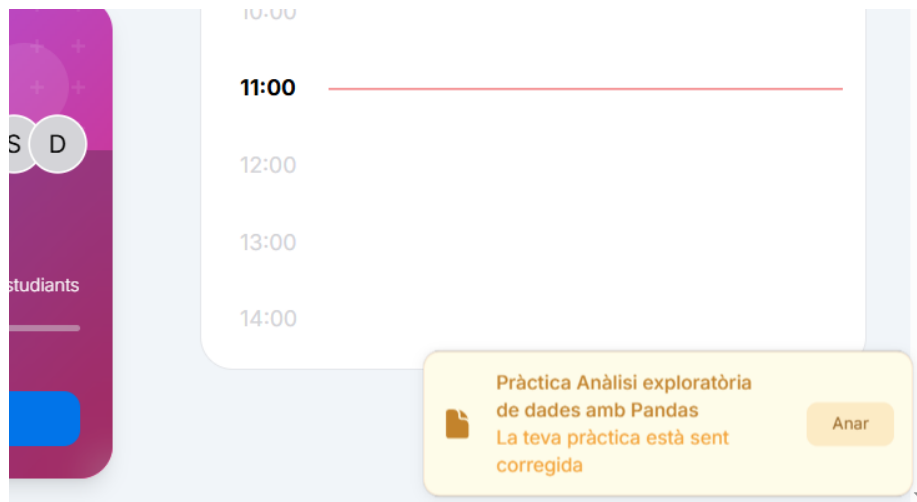


Figura 30. Toast emergent mostrant notificació en temps real

Quan una pràctica canvia d'estat (CORRECTING, CORRECTED, REJECTED), l'usuari rep un Toast immediat com el de la Figura 30, i a més:

- A la campana de notificacions de la navbar, es mostra un badge amb el nombre de notificacions no llegides.
- En fer clic, s'obre un popover amb dues pestanyes: "No llegides" i "Totes".
- Es pot fer clic sobre una notificació per marcar-la com llegida i si està llegida redirigir-te a la pràctica corresponent.
- També hi ha opcions per marcar totes com a llegides o eliminar-les totes.
- Quan es corregeix una pràctica, es mostra un efecte visual de "pulse" al botó de Pràctiques de la SideNav, per cridar l'atenció de l'usuari i indicar que hi ha nova informació pendent de veure.

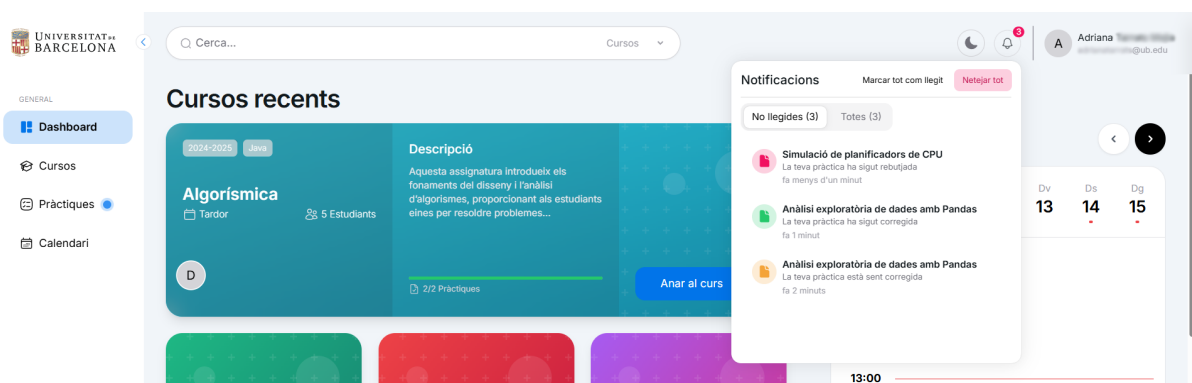


Figura 31. Popover de notificacions i indicador de notificació actiu al botó lateral de Pràctiques

### 4.3.10. Millora global de l'experiència d'usuari (UX)

Un dels principis de disseny aplicats ha estat la coherència cromàtica entre components. La paleta de colors associada als estats de les pràctiques (vermell, groc, blau, verd) no només s'utilitza al calendari, sinó que es repeteix de manera sistemàtica a diversos punts de la interfície: taules de pràctiques, targetes, notifikacions, popovers d'informació, i fins i tot als indicadors d'estat dins dels detalls d'una pràctica. Aquesta decisió reforça la familiaritat visual, redueix la corba d'aprenentatge i ofereix una experiència molt més intuïtiva i predictiva per a l'usuari.

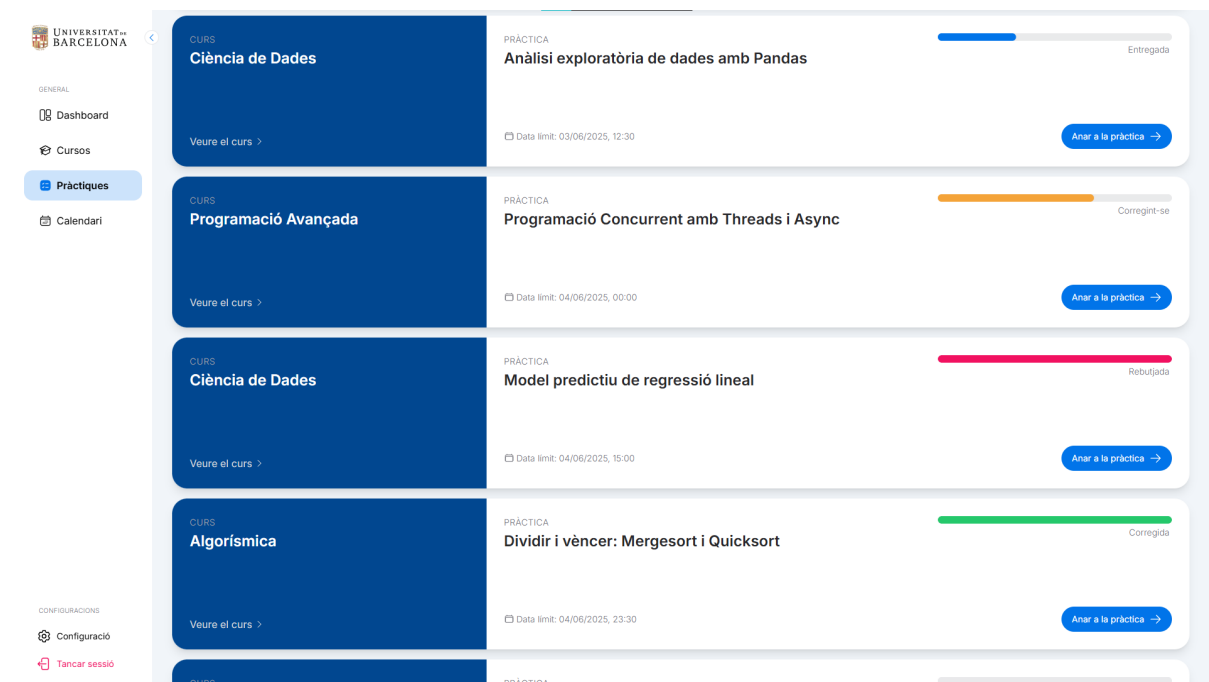


Figura 32. Targetes de pràctiques amb codificació de colors per estat

| <div> <div>Q Cerca per nom...</div> <div> <div>Propers 7 dies</div> <div>Estat</div> </div> </div> |                           |                                                                                                            |              |                   |         |
|----------------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------------------|--------------|-------------------|---------|
| PRÀCTICA                                                                                           | CURS                      | PROFESSOR                                                                                                  | ESTAT        | DATA DE VENCIMENT | ACCIONS |
| Model Predictiu De Regressió Lineal                                                                | Ciència De Dades          | <div>                      Daniel<br/>                     daniel.ortiz@ub.edu                 </div>      | Entregada    | 04/06/2025 15:00  | ⓘ       |
| Dividir I Vèncer: Mergesort I Quicksort                                                            | Algorísmica               | <div>                      Daniel<br/>                     daniel.ortiz@ub.edu                 </div>      | Rebutjada    | 04/06/2025 23:30  | ⓘ       |
| Aplicació CRUD Amb SQLite                                                                          | Bases De Dades            | <div>                      Sergio<br/>                     smarinh33@alumnos.ub.edu                 </div> | No entregada | 05/06/2025 22:45  | ⓘ       |
| Programació Dinàmica: Problema De La Motxilla                                                      | Algorísmica               | <div>                      Daniel<br/>                     daniel.ortiz@ub.edu                 </div>      | Corregint-se | 08/06/2025 18:00  | ⓘ       |
| Implementació D'un Microservei Amb Tests Automatitzats                                             | Enginyeria Del Programari | <div>                      Sergio<br/>                     smarinh33@alumnos.ub.edu                 </div> | Corregida    | 09/06/2025 21:45  | ⓘ       |

Figura 33. Taula amb columnes d'estat codificades visualment

També s’ha treballat de manera especial la gestió d’estats buits, pensant en aquells moments en què l’usuari no té dades per visualitzar. Quan, per exemple, un estudiant encara no està inscrit a cap curs, no té pràctiques assignades, no s’han trobat resultats en una cerca o filtre, o encara no ha accedit recentment a cap curs, la interfície mostra missatges informatius clars, empàtics i contextuals.

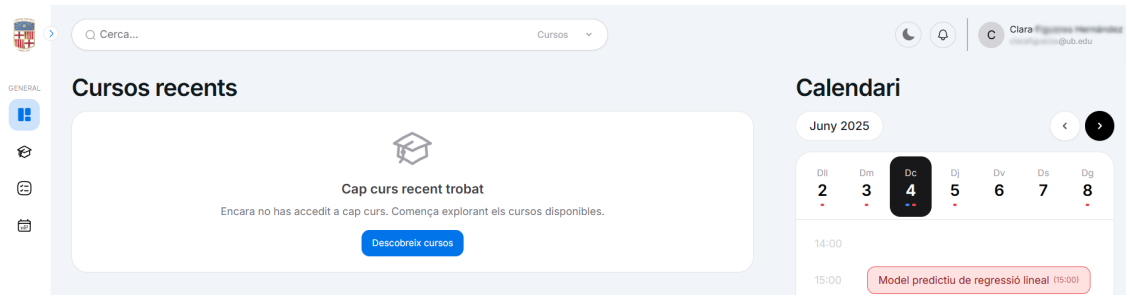


Figura 34. Estat en què l’usuari no s’ha accedit a cap curs



Figura 35. Estat buit a les pràctiques d’un curs

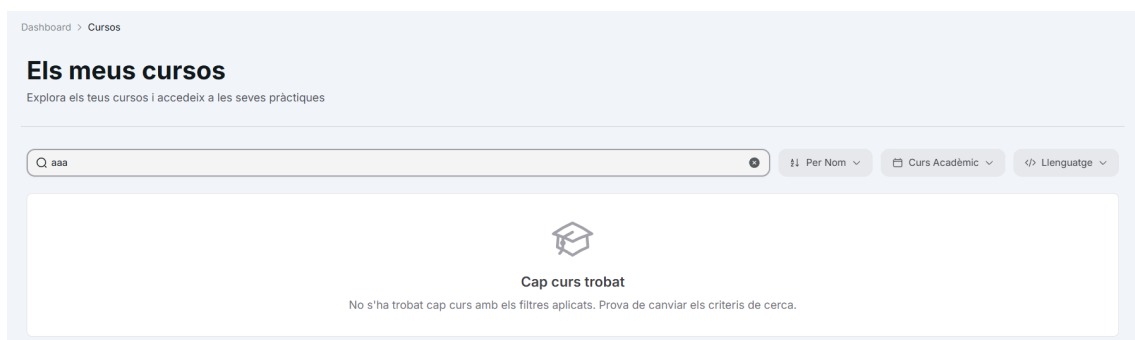


Figura 36. Estat buit a la cerca de la llista de cursos

Aquests missatges no només eviten la sensació d’error o confusió, sinó que ofereixen orientacions d’acció, com ara suggeriments per buscar cursos o consells per modificar els filtres.

Un altre aspecte clau ha estat la validació d'accions permeses. En el cas de les pràctiques, si l'usuari intenta fer una entrega després de la data límit, es mostra un missatge d'error visible que explica que ja no es pot enviar la pràctica perquè ha caducat el termini. Aquest tipus de validacions ajuden a evitar errors, ofereixen un retorn immediat i reforcen la confiança de l'usuari en el sistema.

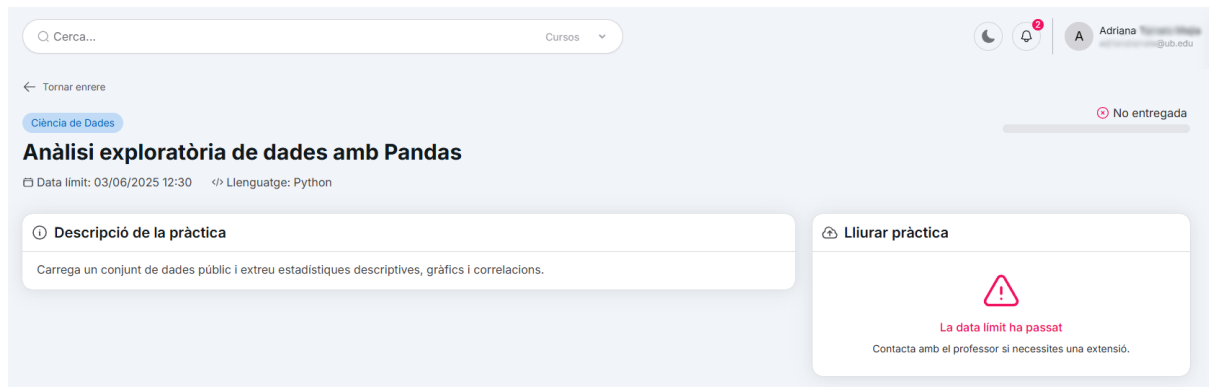


Figura 37. Error quan s'intenta pujar una pràctica fora de termini

Finalment, s'ha cuidat especialment la coherència visual i la intuïció de les interaccions: animacions suaus, feedback visual immediat (hover, clics, estats de càrrega), distribució clara de la informació, breadcrumbs per facilitar la navegació jeràrquica, i compatibilitat completa amb temes foscos i clars. Tots aquests detalls sumen per convertir aquesta versió de Golem XIV en una eina molt més moderna, accessible i eficient per a l'ús acadèmic del dia a dia.

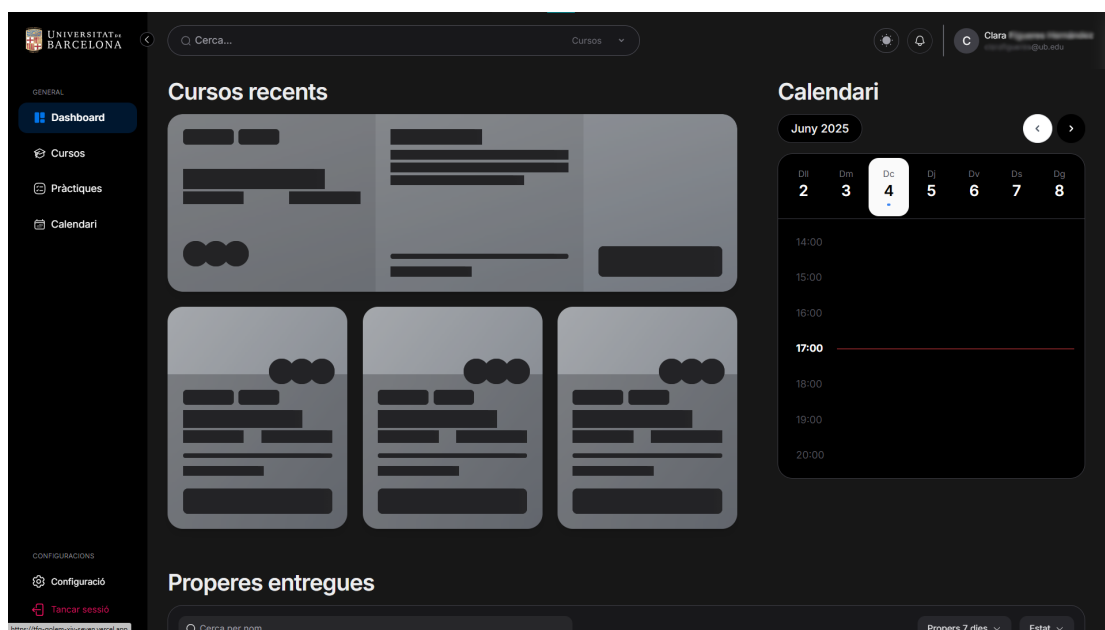


Figura 38. Esquelets de càrrega a la dashboard en mode fosc

## 4.4. Resultats

La nova versió de Golem XIV ha introduït millores substancials tant a nivell tècnic com d'interacció d'usuari, amb impactes visibles en la qualitat de l'experiència educativa, la càrrega de treball dels professors i el rendiment general del sistema. A continuació es presenten els resultats més destacats després de la implementació de totes les funcionalitats i l'arquitectura distribuïda.

### 4.4.1. Millora de l'experiència d'usuari (UI/UX)

La redissenyada interfície ha obtingut una millora clara en termes de navegabilitat, claror visual i fluïdesa d'ús. Les funcionalitats destacades que han contribuït a aquesta millora són:

- Dashboard personalitzat amb pràctiques properes, cursos recents i calendari integrat.
- Codificació cromàtica coherent per als estats de pràctiques en múltiples components.
- Missatges informatius contextuais en estats buits o accions invalidades.
- Sistema de notifikacions en temps real que connecta automàticament amb l'estat de les pràctiques.

Aquestes millores fan l'aplicació més intuïtiva, reduint el temps d'aprenentatge per part de nous usuaris i facilitant una interacció més àgil amb el sistema.

### 4.4.2. Rendiment del sistema de correcció

El canvi de model seqüencial a processament asíncron i concurrent al consumidor de pràctiques, juntament amb la modularització de serveis, ha permès:

- Processar múltiples pràctiques simultàniament, amb una capacitat màxima de nombre de nuclis del processador multiplicat per 4 ( $CPU \times 4$ ). Aquesta configuració no es basa en l'ús de múltiples fils reals (multithreading), sinó en l'aprofitament del model asíncron d'asyncio, que permet gestionar moltes tasques alhora en un sol fil d'execució. Com que la major part del temps el worker el fa servir en operacions I/O (com lectura/escriptura a la

base de dades, comunicació amb el servidor RPC o esperant confirmacions de la cua RabbitMQ), aquestes tasques no consumeixen intensament la CPU. Això permet tenir més tasques actives que nuclis disponibles permetent que mentre una tasca està esperant, la CPU aprofiti per processar una altra tasca. Per tant, el factor  $\times 4$  s'ha triat com a valor empíricament eficient que equilibra rendiment i estabilitat, oferint un bon rendiment sense arribar a sobrecarregar el bucle d'esdeveniments d'asyncio. I que, en entorns com Azure Container Apps on està desplegat, aquesta relació també ofereix un bon equilibri entre rendiment i estabilitat.

- Reduir dràsticament el temps mitjà de correcció, ja que diverses entregues poden processar-se en paral·lel en lloc de formar una cua seqüencial.
- Registrar i rastrejar errors amb més precisió gràcies al sistema de logs.

També s'ha assegurat que en escenaris de fallada (per exemple, errors del servidor RPC o problemes en el format d'una entrega), els missatges es gestionen de manera segura gràcies als mecanismes de reintents limitats, TTL i DLQ, assegurant la resiliència del sistema i evitant bloquejos indefinits.

#### **4.4.3. Fiabilitat i estabilitat del sistema**

Amb el desplegament distribuït, el sistema és ara molt més resiliència i escalable. Els serveis independents (frontend, backend, worker, servidor RPC, base de dades i missatgeria) poden escalar de manera separada i ser monitorats de forma individual.

L'ús de contenidors Docker i el Continuous Deployment amb GitHub Actions assegura que els canvis es despleguin de manera ràpida i controlada, mantenint la base de dades sincronitzada mitjançant migracions gestionades automàticament.

#### **4.4.4. Accessibilitat i portabilitat**

El frontend és completament adaptat a temes foscos i clars, i compatible amb diferents resolucions de pantalles d'ordinador. Les funcionalitats com el drag &

drop, popovers, toasts i navegació lateral col·lapsable aporten accessibilitat i comoditat d'ús a estudiants i professors amb perfils diversos.

#### 4.4.5. Resum dels resultats obtinguts

| Àmbit                     | Millora implementada                          | Impacte                                                 |
|---------------------------|-----------------------------------------------|---------------------------------------------------------|
| UI/UX                     | Redisseny total de la interfície              | Navegació més intuïtiva i visual                        |
| Rendiment                 | Worker asíncron i concurrent                  | Correcció paral·lela i temps reduït                     |
| Estabilitat               | Arquitectura distribuïda i gestió de fallides | Sistema robust, fàcilment escalable i tolerant a errors |
| Monitoratge               | Sistema de logs detallat i contextualitzat    | Diagnòstic i traçabilitat més eficients                 |
| Accessibilitat            | Missatges contextuais i feedback visual       | Experiència millorada per a tots els rols               |
| Desplegament continu (CD) | Desplegament automàtic amb GitHub Actions     | Integració i desplegament fiables, traçables i àgils    |

## 5. Conclusions i treball futur

---

### 5.1. Conclusions

El desenvolupament d'aquesta tercera versió de Golem XIV ha representat un pas endavant molt significatiu tant pel que fa a l'arquitectura interna del sistema com a la interfície d'usuari i experiència d'ús. El projecte ha evolucionat des d'una base funcional cap a una plataforma modular, escalable i visualment cuidada, orientada a facilitar la feina del professorat i a millorar la interacció amb l'alumnat.

S'han assolit amb èxit els objectius plantejats inicialment:

- El **refactor complet del backend** ha permès una estructura neta, escalable i fàcilment mantenible, amb models `SQLModel` que simplifiquen la integració amb la base de dades i els endpoints d'API.
- El **nou worker asíncron i concurrent**, juntament amb la incorporació de mecanismes de reintents i una DLQ, ha millorat de forma notable l'eficiència i robustesa del sistema de correcció automàtica.
- L'**arquitectura distribuïda desplegada al núvol**, amb contenidors Docker i CD automatitzat, assegura que cada servei es pot desplegar i escalar de manera independent.
- El **frontend redissenyat** amb Next.js i TailwindCSS ha portat la plataforma a un nivell molt més modern, usable i coherent visualment, amb noves funcionalitats com la dashboard, el calendari, el sistema de notificacions i una navegació clara i fluida.

A més, el projecte ha tingut una forta vessant d'aprenentatge tècnic a nivell personal: des de l'adopció de tecnologies com React/Next.js, Docker o RabbitMQ, fins al desplegament i manteniment d'una arquitectura al núvol mitjançant Azure i Vercel.

En conjunt, es pot considerar que aquesta nova versió de Golem XIV és més madura, estable i preparada per a un ús real dins d'un entorn docent.

## 5.2. Treball futur

Tot i les millores implementades a la tercera versió de Golem XIV, el sistema encara té marge per seguir evolucionant.

A nivell de **funcionalitat i experiència d'usuari**, es proposa:

- **Millorar l'editor d'entregues**, afegint funcions com la previsualització de fitxers, detecció d'errors abans de l'enviament i suggeriments automàtics per a l'estudiant.
- **Permetre configuracions de pràctiques amb deadlines flexibles**, tant per grups com per estudiants individuals, així com opcions per lliurar fora de termini amb control docent.
- **Mostrar un historial complet de canvis per pràctica**, incloent entregues anteriors, correccions rebudes i modificacions a l'enunciat original.
- **Incloure un sistema d'avaluació qualitativa**, amb comentaris del professor, puntuacions manuals i rúbriques personalitzables.
- **Afegir una pàgina d'estadístiques** amb gràfiques i indicadors de progrés, rendiment i participació, tant per a estudiants com per a docents.
- **Ampliar el suport de llenguatges de programació**, per donar cobertura a més assignatures i escenaris docents dins de l'àmbit tècnic.
- **Incloure un xat o fòrum dins de cada curs o pràctica**, fomentant la comunicació directa entre estudiants i professors.

Pel que fa a la **interacció i accessibilitat**, caldria:

- **Reestructurar la interfície per adaptar-la a dispositius mòbils**, amb disseny per millorar la usabilitat en pantalles petites i amb interacció tàtil.
- **Acompanyar les notificacions en temps real amb alertes per correu electrònic**, per assegurar que l'usuari estigui informat encara que no estigui connectat.
- **Establir missatges contextuais de confirmació o assistència** en accions clau, com pujar fitxers, eliminar entregues o editar pràctiques.

Des del punt de vista de **seguretat i control d'accés**:

- **Implementar un sistema de rols i permisos més granular**, que permeti definir accés específic per funcionalitat, curs o context.
- **Establir grups d'estudiants dins d'un mateix curs**, per permetre pràctiques diferenciades i una gestió acadèmica més flexible.
- **Integrar l'autenticació multifactor (MFA)** i registres d'activitat (audit log), per augmentar la seguretat i la traçabilitat d'accions sensibles.

En l'àmbit **tècnic i arquitectònic**, es podrien dur a terme les següents millores:

- **Millorar el temps de pujada dels fitxers**, optimitzant el procés de transferència amb tècniques de compressió, paral·lelització o l'ús de mecanismes de transferència més eficients.
- **Substituir el sistema actual d'SFTP per una solució d'emmagatzematge més moderna i escalable**, que millori el control d'accés, el rendiment i el manteniment.
- **Implementar diferents tipus de tests automatitzats**: tests unitaris, tests d'integració i tests end-to-end, cobrint així els diferents nivells del sistema i assegurant-ne el bon funcionament davant canvis i regressions
- **Establir un sistema de Continuous Integration (CI)** que executi aquests tests automàticament en cada canvi de codi (commit o pull request), afavorint un desenvolupament més fiable, segur i àgil.
- **Integrar la plataforma amb el campus virtual de la universitat**, permetent sincronitzar automàticament cursos, usuaris i pràctiques amb l'ecosistema docent institucional, estalviant duplicació de dades i processos manuals.

Aquestes millores no només permetrien enriquir funcionalitats i millorar l'experiència de tots els usuaris, sinó que també consolidarien Golem XIV com una plataforma docent integral, robusta, versàtil i preparada per escalar i evolucionar a llarg termini.

## 6. Bibliografia

---

- [1] Rocío Barrera Robles. *Dipòsit Digital de la Universitat de Barcelona: Projecte Golem XIV v2.0: Desenvolupament d'una aplicació basada en una arquitectura de microserveis*. <https://diposit.ub.edu/dspace/handle/2445/221225>
- [2] FastAPI. *High-performance web framework for building APIs with Python*. <https://fastapi.tiangolo.com>
- [3] Azure Web App for Containers. *Microsoft Cloud-native Hosting for Web App for Containers*. <https://azure.microsoft.com/en-us/products/app-service/containers/>
- [4] Supabase. *The open source Firebase alternative*. <https://supabase.com>
- [5] CloudAMQP. *LavinMQ and RabbitMQ Hosting in the Cloud*. <https://www.cloudamqp.com>
- [6] Next.js. *The React Framework by Vercel*. <https://nextjs.org>
- [7] TailwindCSS. *Utility-first CSS framework for rapid UI development*. <https://tailwindcss.com>
- [8] HeroUI. *React UI Library*. <https://www.heroui.dev>
- [9] Docker. <https://www.docker.com>
- [10] SQLAlchemy. *ORM and data validation library for Python based on SQLAlchemy and Pydantic*. <https://sqlmodel.tiangolo.com>
- [11] Alembic. *Database migration tool for SQLAlchemy*. <https://alembic.sqlalchemy.org>
- [12] Pydantic. *Data parsing and validation using Python type annotations*. <https://docs.pydantic.dev>
- [13] PostgreSQL. *The world's most advanced open-source relational database*. <https://www.postgresql.org>
- [14] Python. *General-purpose programming language*. <https://www.python.org>

- [15] asyncio. *Asynchronous I/O in Python*.  
<https://docs.python.org/3/library/asyncio.html>
- [16] aio-pika. *AMQP Client for asyncio and Python*. <https://aio-pika.readthedocs.io>
- [17] Vercel. *Frontend Cloud Platform*. <https://vercel.com>
- [18] Swagger UI. *Interactive API documentation for RESTful web services*.  
<https://swagger.io>
- [19] ReDoc. *OpenAPI-generated API Documentation*. <https://redocly.com>
- [20] AMQP Protocol. *Advanced Message Queuing Protocol*. <https://www.amqp.org>
- [21] SFTP Protocol. *SSH File Transfer Protocol*.  
<https://datatracker.ietf.org/doc/html/draft-ietf-secsh-filexfer>
- [22] Azure Container Apps. *Microsoft Cloud-native Hosting for Web Applications*.  
<https://azure.microsoft.com/en-us/products/container-apps/>
- [23] Azure Virtual Machine. *Scalable Cloud VMs by Microsoft Azure*.  
<https://azure.microsoft.com/en-us/products/virtual-machines/>
- [24] SQLAlchemy. *The Python SQL Toolkit and ORM*. <https://www.sqlalchemy.org>
- [25] Starlette. *Lightweight ASGI framework/toolkit*. <https://www.starlette.io>

## Apèndix A. Repositori de codi font

---

El codi font complet del projecte Golem XIV – Versió 3 es troba disponible públicament en un repositori de GitHub. Aquest inclou tots els components que formen part de l'arquitectura de l'aplicació:

- **Frontend**
- **Backend**
- **Worker**
- **Servidor RPC de correcció de pràctiques simulat**
- **Migracions de base de dades**
- **Scripts de desplegament i configuració**

**Enllaç al repositori:**

<https://github.com/sergiomarinn/TFG-GOLEM-XIV>