



UNIVERSITAT DE  
BARCELONA

Treball de fi de Grau

GRAU EN ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica  
Universitat de Barcelona

---

**Projecte GOLEM:**  
**Una aplicació distribuïda per a**  
**l'Avaluació Automàtica d'Exercicis**  
**de Programació**

---

**Autor: Gerard López Ruiz**

**Directors: Dr. Daniel Ortiz Martínez i Dr. Eloi Puertas i Prats**

**Barcelona, 17 de gener de 2024**

## Agraïments

*Primer de tot, vull agrair als meus tutors tota la dedicació que han posat al projecte, els ànims que m'han transmès i totes les reunions a les dues de la tarda dels dilluns que m'alegraven el principi de setmana.*

*També vull agrair a la Katia i al Marcos, tot el suport que m'han donat en els moments que jo no trobava una solució i ells, amb tota la paciència del món, em responien a les preguntes que jo els hi feia.*

*A la família i els amics, que m'han animat a seguir endavant i que han confiat que ho aconseguiria.*

*I finalment i més importat, vull agrair a la meva parella, la Gemma, que m'ha fet entendre, que encara que tot sembli negre sempre hi ha una llum al final del túnel.*

## Resum

Al llarg de la història, els éssers humans han manifestat una constant cerca de l'automatització amb l'objectiu de simplificar i millorar l'eficiència en diverses tasques. Des de les seves arrels amb les eines rudimentàries fins a l'era de les màquines modernes, la tendència cap a l'automatització ha estat una evolució marcada.

En l'actualitat, aquesta tendència s'ha ampliat a àmbits més abstractes, incorporant l'automatització de processos intel·lectuals. Un exemple d'aquesta evolució és la correcció de pràctiques en l'àmbit educatiu, que històricament ha estat una tasca manual i laboriosa. No obstant això, amb els avenços en tecnologia, intel·ligència artificial i aprenentatge automàtic, la idea de la correcció de pràctiques de forma automàtica ha emergit com una àrea d'interès.

En aquest context, he desenvolupat una aplicació web que permet als usuaris carregar les seves pràctiques de programació. A diferència de la correcció manual, aquestes pràctiques són avaluades mitjançant un programa de correcció automàtica. Aquest enfocament no només busca simplificar el procés de correcció sinó també promoure un entorn d'aprenentatge interactiu i eficaç en el món de la programació.

Mitjançant la implementació d'aquesta aplicació, es pretén aportar una solució innovadora a la tradicional correcció manual de pràctiques. Els avantatges van més enllà de l'eficiència, ja que es crea una oportunitat per a estudiants i professors de participar activament en un procés de correcció més dinàmic i automatitzat.

Així, aquest projecte no només s'orienta a la simplificació de tasques, sinó que també contribueix a la transformació de l'educació en àmbits relacionats amb la programació. Amb una interacció més ràpida i dinàmica, l'aprenentatge es converteix en un procés més efectiu i atractiu per als estudiants, promocionant l'adopció de noves tecnologies en l'ensenyament i aprenentatge.

## Resumen

A lo largo de la historia, los seres humanos han buscado constantemente la automatización con el objetivo de simplificar y mejorar la eficiencia en diversas tareas. Desde sus raíces con herramientas rudimentarias hasta la era de las máquinas modernas, la tendencia hacia la automatización ha sido una evolución marcada.

En la actualidad, esta tendencia se ha ampliado a ámbitos más abstractos, incorporando la automatización de procesos intelectuales. Un ejemplo de esta evolución es la corrección de prácticas en el ámbito educativo, que históricamente ha sido una tarea manual y laboriosa. Sin embargo, con los avances en tecnología, inteligencia artificial y aprendizaje automático, la idea de la corrección de prácticas de forma automática ha emergido como un área de interés.

En este contexto, he desarrollado una aplicación web que permite a los usuarios cargar sus prácticas de programación. A diferencia de la corrección manual, estas prácticas son evaluadas mediante un programa de corrección automática. Este enfoque no solo busca simplificar el proceso de corrección, sino también promover un entorno de aprendizaje interactivo y eficaz en el mundo de la programación.

Mediante la implementación de esta aplicación, se pretende aportar una solución innovadora a la tradicional corrección manual de prácticas. Los beneficios van más allá de la eficiencia, ya que se crea una oportunidad para que estudiantes y profesores participen activamente en un proceso de corrección más dinámico y automatizado.

Así, este proyecto no solo se orienta a la simplificación de tareas, sino que también contribuye a la transformación de la educación en ámbitos relacionados con la programación. Con una interacción más rápida y dinámica, el aprendizaje se convierte en un proceso más efectivo y atractivo para los estudiantes, promoviendo la adopción de nuevas tecnologías en la enseñanza y el aprendizaje.

## Abstract

Throughout history, humans have constantly sought automation in order to simplify and improve efficiency in various tasks. From its roots with rudimentary tools to the age of modern machines, the trend towards automation has been a marked evolution.

Today, this trend has expanded into more abstract areas, incorporating the automation of intellectual processes. An example of this evolution is the correction of practices in education, which historically has been a manual and labour-intensive task. However, with advances in technology, artificial intelligence and machine learning, the idea of automatic remediation has emerged as an area of interest.

In this context, I have developed a web application that allows users to upload their programming assignments. In contrast to manual correction, these practices are evaluated by an automatic correction program. This approach not only aims to simplify the correction process, but also to promote an interactive and effective learning environment in the world of programming.

By implementing this application, the aim is to provide an innovative solution to the traditional manual correction of practicals. The benefits go beyond efficiency, as it creates an opportunity for students and teachers to actively participate in a more dynamic and automated correction process.

Thus, this project not only aims at simplifying tasks, but also contributes to the transformation of education in programming-related fields. With faster and more dynamic interaction, learning becomes a more effective and engaging process for students, promoting the adoption of new technologies in teaching and learning.

# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Motivació . . . . .	1
1.2	Objectius . . . . .	2
1.2.1	Objectius Funcionals . . . . .	2
1.2.2	Objectius personals . . . . .	3
<b>2</b>	<b>Anàlisi</b>	<b>5</b>
2.1	Requisits . . . . .	5
2.1.1	Requisits Funcionals . . . . .	5
2.1.2	Requisits no Funcionals . . . . .	6
2.2	Backlog . . . . .	7
<b>3</b>	<b>Arquitectura de l'aplicació</b>	<b>11</b>
3.1	PostgreSQL . . . . .	12
3.2	Mongo DB . . . . .	15
3.3	Sistema de Directoris . . . . .	17
3.4	RabbitMQ . . . . .	19
3.4.1	Problemes Sorgits . . . . .	19
3.5	Eines Utilitzades . . . . .	21
3.5.1	Llenguatges de Programació . . . . .	21
3.5.2	FrameWorks . . . . .	22
3.5.3	Plataformes utilitzades . . . . .	25
<b>4</b>	<b>Implementació i Resultats</b>	<b>26</b>
4.1	Disseny . . . . .	26
4.1.1	Colors . . . . .	34
4.2	Crides a la API . . . . .	36

4.2.1	Crides desde el Front . . . . .	36
4.2.2	Endpoints Backend . . . . .	37
4.3	Sistema d'autenticació . . . . .	39
4.4	Metode de connexió amb la eina de correcció . . . . .	40
4.5	Instal·lació al servidor . . . . .	41
4.5.1	Primers passos . . . . .	41
4.5.2	Prova end-to-end Ping . . . . .	41
4.5.3	Problemes de connexió amb TeamViewer . . . . .	43
4.6	Resultats . . . . .	45
<b>5</b>	<b>Conclusions i treball futur</b>	<b>46</b>
<b>6</b>	<b>Bibliografia</b>	<b>48</b>
<b>7</b>	<b>Llicències</b>	<b>49</b>
7.1	Flowbite . . . . .	49
7.2	Heroicons . . . . .	49
<b>8</b>	<b>Annexos</b>	<b>51</b>
8.1	Enpoints Usuari . . . . .	51
8.2	Enpoints Cursos . . . . .	51

# 1 Introducció

L'àmbit de l'ensenyament de la programació s'enfronta a reptes considerables, i entre ells destaca la correcció de les pràctiques dels estudiants. Amb l'objectiu d'optimitzar aquest procés, aquest Treball de Fi de Grau (TFG) se centra en el desenvolupament d'una eina integral de correcció automàtica. En aquesta fase inicial, l'atenció es concentra en la integració del frontend i el backend de l'eina.

La proposta aborda la necessitat d'eficiència i precisió en l'avaluació de les pràctiques dels estudiants mitjançant una eina que permeti als professors automatitzar aquest procés. En aquest context, la integració del frontend, que representa la interfície d'usuari, amb el backend, que controla la lògica de l'emmagatzematge dels fitxers, sorgeix com un pas crucial cap a la creació d'una eina completa i funcional.

Aquest TFG explora els requisits i les decisions de disseny en la creació del frontend, analitzant la interfície d'usuari amb un enfocament en la usabilitat i l'experiència de l'usuari. Al mateix temps, s'aborda el desenvolupament del backend, centrant-se en la lògica d'emmagatzematge dels fitxers, la comunicació eficient amb el frontend i la comunicació amb el motor de correcció de pràctiques.

A través de l'estudi de la implementació d'aquesta eina, aquest TFG proporciona una visió pràctica sobre com la integració del frontend i el backend contribueix al funcionament cohesiu de l'eina de correcció automàtica de pràctiques en programació. S'analitzen els beneficis potencials, les possibles dificultats i es destaca el camí cap a futurs desenvolupaments i millores. Amb aquesta investigació, s'espera aportar una contribució significativa al camp de les solucions tecnològiques aplicades a l'educació en programació.

## 1.1 Motivació

A l'inici del meu procés de definició del Treball de Fi de Grau (TFG), vaig proposar la idea de desenvolupar un curs introductori de programació destinat als alumnes de batxillerat i institut. Tanmateix, el primer tutor amb qui vaig discutir la meva proposta va considerar que aquesta línia de treball no s'ajustava del tot als paràmetres d'un TFG orientat a la carrera d'Enginyeria Informàtica.

Aquesta primera valoració va representar un punt d'inflexió, ja que vaig haver de replantejar la meva aproximació al projecte. En aquest context, el meu tutor em va presentar a un dels seus companys, el segon tutor, qui ja disposava d'un programari de correcció automàtica de pràctiques. Aquesta eina, tot i ser potent, es feia servir de manera rudimentària a través de comandes de bash.



Aquesta situació va obrir una nova oportunitat i va marcar el gir cap a una direcció diferent per al meu TFG. Amb l'objectiu de satisfer les expectatives de la carrera d'Enginyeria Informàtica, el segon tutor va suggerir que em centrés en el desenvolupament del frontend i el backend per connectar-se amb aquesta eina de correcció automàtica ja existent. Aquesta proposta va captivar la meva atenció, ja que representava una oportunitat tant per millorar una eina ja útil com per aprofundir en l'àmbit de la programació i el desenvolupament de programari.

D'aquesta manera, la transició de la primera idea de TFG a la proposta actual va ser un procés guiat per les orientacions i les necessitats percebudes dels tutors.

A més, aquest TFG està enmarcat dins del projecte d'investigació i desenvolupament denominat GOLEM, amb una durada de tres anys. L'objectiu principal d'aquest projecte és establir un servei de programari accessible a través del campus virtual, destinat a simplificar i automatitzar les tasques d'avaluació d'entregues i exàmens. Aquest TFG representa el punt d'inici d'aquest ambiciós projecte i a més és una motivació afegida que els tutors hagin comptat amb mi.

## 1.2 Objectius

Amb l'objectiu d'assegurar l'èxit de l'aplicació, m'he fixat objectius concrets que guiaran el desenvolupament del Treball de Fi de Grau. Aquests objectius representen la síntesi del meu coneixement adquirit en la carrera d'Enginyeria Informàtica i les habilitats desenvolupades de manera autònoma. Amb això, busco proporcionar una solució eficient per a la correcció automàtica de pràctiques, alhora que contribueixo al meu propi creixement com a desenvolupador de software.

### 1.2.1 Objectius Funcionals

#### 1. Desenvolupament del Frontend:

Crear una interfície d'usuari (frontend) atractiva, intuïtiva i fàcil d'utilitzar per facilitar la interacció dels professors amb l'eina de correcció automàtica. Assegurar una experiència d'usuari eficient que millori la navegació i la comprensió de les funcionalitats de l'eina.

#### 2. Implementació del Backend:

Desenvolupar un backend robust que gestioni de manera eficient la informació que arriba des del frontend. Assegurar una connexió fluida, segura i eficaç que permeti l'intercanvi d'informació necessària per a la correcció automàtica de les pràctiques dels estudiants.

#### 3. Integració amb l'Eina de Correcció Automàtica:

Connectar amb èxit el backend amb l'eina de correcció automàtica preexistent. Assegurar que la informació es transmeti correctament, permetent als professors i als alumnes utilitzar l'eina de manera més àgil i eficient sense necessitat de comandes de bash.

#### 4. **Millora de l'Accessibilitat:**

Optimitzar l'accessibilitat de l'eina de correcció automàtica per als professors, afegint funcionalitats que facilitin la revisió, la correcció i la retroalimentació de les pràctiques dels estudiants. Garantir que l'eina sigui accessible des de diferents dispositius i navegadors.

#### 5. **Seguretat de les Dades:**

Garantir la seguretat i la confidencialitat de les dades dels estudiants. Implementar protocols de seguretat per protegir la informació durant la transmissió i l'emmagatzematge, assegurant el compliment de les normatives de privacitat i seguretat.

#### 6. **Testeig i Depuració:**

Realitzar proves exhaustives del frontend i el backend per assegurar-se que funcionen correctament en diferents escenaris i situacions. Detectar i corregir qualsevol error, bug o malfuncionament per garantir la fiabilitat i la integritat del sistema.

#### 7. **Documentació Detallada:**

Generar una documentació completa i comprensible que descriu tant el frontend com el backend, incloent-hi detalls tècnics, instruccions d'ús i possibles solucions a problemes comuns. Aquesta documentació servirà com a guia per als futurs usuaris i desenvolupadors.

### 1.2.2 Objectius personals

#### 1. **Aprendre React i Tailwind CSS:**

Adquirir coneixements sòlids en React i dominar l'ús de Tailwind CSS amb l'objectiu de crear una interfície d'usuari atractiva i intuïtiva que millori la interacció dels professors amb l'eina de correcció automàtica.

#### 2. **Utilitzar Metodologies Àgils:**

Aplicar principis àgils per millorar l'eficiència del desenvolupament, facilitant l'adaptabilitat a canvis, la col·laboració efectiva i una millora en la interacció amb l'equip de desenvolupament.

#### 3. **Dissenyar una REST API Sòlida amb FastAPI:**

Desenvolupar una interfície backend robusta utilitzant FastAPI per gestionar la comunicació amb el frontend, assegurant una connexió fluida i eficaç entre els components del sistema.

**4. Dissenyar una Base de Dades Robusta i Funcional:**

Aprendre a dissenyar i implementar una base de dades eficient utilitzant SQLAlchemy per a PostgreSQL i MongoDB, fonamental per a garantir la gestió robusta i funcional de les dades.

**5. Instal·lar Correctament el Codi en un Servidor Universitari:**

Configurar i instal·lar el sistema de manera correcta en l'entorn del servidor universitari, assegurant que l'aplicació funcioni de manera fiable i segura en el context de la universitat.

## 2 Anàlisi

Una anàlisi exhaustiu dels requisits d'una aplicació és fonamental per decidir com i per on iniciar el desenvolupament. A aquesta secció, ens endinsarem en la identificació dels requisits, tant funcionals com no funcionals, de l'aplicació. A més, es detallarà el “backlog” que es va establir al principi del projecte, proporcionant una guia estructurada per a la planificació i la implementació. Aquest procés integral d'entendre i documentar els requeriments són els fonaments essencials per garantir un desenvolupament efectiu i una implementació exitosa de l'aplicació.

### 2.1 Requisites

#### 2.1.1 Requisites Funcionals

Els requisits funcionals d'una aplicació són les especificacions i funcionalitats concretes que l'aplicació ha de complir per satisfer les necessitats dels seus usuaris. Aquests requisits descriuen les accions que l'aplicació ha de realitzar i quines funcionalitats ha de proporcionar. En el cas de la nostra aplicació es poden resumir en:

1. **Registre d'usuaris:** Els usuaris s'han de poder registrar a l'aplicació.
2. **Login d'usuaris:** Els usuaris han de poder entrar a l'aplicació amb les credencials.
3. **Visualització de cursos:** Els usuaris han de ser capaços de veure a quins cursos estan assignats.
4. **Visualització de pràctiques:** Els usuaris han de ser capaços de veure les pràctiques creades per aquell curs.
5. **Visualització de la correcció:** Els alumnes han de ser capaços de veure la correcció que torna l'eina de correcció automàtica.
6. **Calendari:** Els usuaris han de ser capaços de veure quan acaben els terminis d'entrega de les pràctiques.
7. **Pujada de fitxers:** Els usuaris han de ser capaços de pujar fitxers al servidor. Tant les pràctiques dels alumnes com els fitxers de correcció dels professors.
8. **Creació de cursos:** Els professors han de ser capaços de crear cursos i assignar de manera automàtica els alumnes que tenen accés al curs.
9. **Creació de pràctiques:** Els professors han de ser capaços de crear pràctiques en un curs en concret i assignar a aquella pràctica un llenguatge de programació.

10. **Assignar rols:** Els administradors han de ser capaços d'assignar rols als professors, tant de creació de cursos com de pràctiques.

### 2.1.2 Requisits no Funcionals

Els requisits no funcionals d'una aplicació són característiques i restriccions que no estan relacionades amb funcionalitats específiques, sinó que se centren en altres aspectes com el rendiment, la seguretat, la fiabilitat i l'experiència de l'usuari. Aquests requisits defineixen les qualitats globals i les restriccions del sistema. En el cas de la nostra aplicació són:

1. **Seguretat:** La seguretat de l'aplicació és una prioritat crítica, sobretot considerant la naturalesa dels fitxers universitaris. Amb el codi allotjat en un servidor de la universitat, és imperatiu evitar qualsevol possible vulnerabilitat de seguretat. S'han d'implementar mesures robustes per garantir la protecció integral de les dades sensibles.
2. **Escalabilitat:** És crucial que l'aplicació exhibeixi una capacitat d'escalabilitat eficient per facilitar futures ampliacions de funcionalitats sense problemes operatius. Aquesta escalabilitat assegurarà que l'aplicació pugui evolucionar i adaptar-se a les necessitats creixents sense afectar el seu rendiment.
3. **Usabilitat:** La interfície de l'aplicació ha de ser no només senzilla i intuïtiva, sinó també considerada com a factor crític, especialment en el context d'un nou mètode de correcció. És la nostra responsabilitat assegurar que l'ús de l'aplicació sigui accessible i eficaç amb una experiència d'usuari optimitzada.
4. **Rendiment:** L'aplicació ha de ser capaç de gestionar amb eficàcia una càrrega de treball amb múltiples usuaris connectats simultàniament. S'ha de garantir un temps de resposta significativament baix per mantenir un rendiment òptim, assegurant que els usuaris experimentin l'aplicació de forma fluïda i sense interrupcions.
5. **Integració:** L'aplicació ha de demostrar una compatibilitat fiable i una connexió efectiva amb l'eina de correcció automàtica i altres serveis addicionals que pugui requerir. La seva capacitat per integrar-se amb altres components del sistema és crucial per a la seva funcionalitat global i l'èxit del projecte.

## 2.2 Backlog

Un “backlog” és una llista prioritzada de tasques, funcionalitats o elements que han de ser abordats o desenvolupats en un projecte. Serveix com a inventari dinàmic que pot ser actualitzat i reordenat segons les necessitats del projecte. Aquesta llista permet als membres de l’equip tenir una visió clara de les feines pendents i proporciona una guia per a la planificació i la implementació de noves funcionalitats.

User Story	Descripció	Criteris d’acceptació
US1	Com a usuari no registrat vull accedir a l’aplicació	<ul style="list-style-type: none"><li>- Un usuari no registrat no pot accedir a l’aplicació</li><li>- Un usuari registrat pot accedir a l’aplicació</li></ul>
US2	Com a usuari vull veure a quins cursos estic assignat	<ul style="list-style-type: none"><li>- L’usuari pot accedir a la pàgina principal i veure els cursos als que està assignat</li><li>- L’usuari no pot veure els cursos als que no està assignat</li></ul>
US3	Com a usuari assignat a un curs vull veure quines pràctiques té aquell curs	<ul style="list-style-type: none"><li>- L’usuari pot veure el curs a la pàgina principal</li><li>- L’usuari pot entrar al detall d’aquell curs</li><li>- L’usuari pot veure quines pràctiques estan creades en aquell curs</li></ul>
US4	Com a usuari vull veure al calendari les pràctiques dels cursos als que estic assignat	<ul style="list-style-type: none"><li>- L’usuari, des de qualsevol lloc de l’aplicació, pot accedir al calendari</li><li>- L’usuari pot veure la pàgina del calendari</li><li>- L’usuari pot veure marcades al calendari les pràctiques i quan acaben</li></ul>

User Story	Descripció	Criteris d'acceptació
US5	Com a estudiant vull poder penjar una pràctica per veure la correcció automàtica	<ul style="list-style-type: none"> <li>- L'estudiant pot accedir a la pràctica específica del seu treball</li> <li>- L'estudiant pot veure el detall de la pràctica</li> <li>- L'estudiant pot pujar un fitxer a través de la pàgina de la pràctica</li> <li>- L'estudiant pot veure la correcció automàtica al cap de uns minuts</li> <li>- L'estudiant pot veure la correcció completa després que s'acabi el termini de la pràctica</li> </ul>
US6	Com a professor vull poder crear cursos nous per als meus estudiants	<ul style="list-style-type: none"> <li>- El professor pot accedir des del menú principal a la pàgina de creació de cursos</li> <li>- El professor pot adjuntar un fitxer plantilla amb els alumnes que vol assignar al curs</li> <li>- El professor es pot descarregar el fitxer plantilla</li> <li>- El professor pot crear el curs, que es guardi de manera correcta i veure'l a la pàgina principal</li> </ul>
US7	Com a professor vull poder crear pràctiques d'un curs en concret per a que els alumnes les vegin	<ul style="list-style-type: none"> <li>- El professor pot accedir al curs</li> <li>- El professor, des de dins del curs, pot accedir a la pàgina de creació de pràctiques</li> <li>- El professor pot adjuntar fitxers a aquella pràctica</li> <li>- El professor pot crear la pràctica, que es guardi de manera correcta i veure-la a la pàgina del curs</li> </ul>

User Story	Descripció	Criteris d'acceptació
US8	Com a professor vull poder editar una pràctica	<ul style="list-style-type: none"> <li>- El professor pot accedir al detall de la pràctica</li> <li>- El professor pot pujar més fitxers a la pràctica</li> <li>- El professor pot canviar els detalls de la pràctica</li> <li>- El professor pot veure quins fitxers estan pujats per aquella pràctica</li> <li>- El professor pot eliminar fitxers de la pràctica</li> </ul>
US9	Com a professor vull poder activar la correcció completa quan vulgui	<ul style="list-style-type: none"> <li>- El professor pot accedir al detall de la pràctica</li> <li>- El professor pot activar la correcció automàtica</li> <li>- El professor pot veure els resultats de la correcció automàtica de cada estudiant</li> </ul>
US10	Com a professor vull poder eliminar una pràctica	<ul style="list-style-type: none"> <li>- El professor pot accedir al detall de la pràctica</li> <li>- El professor pot veure un botó per eliminar la pràctica</li> <li>- El professor pot eliminar la pràctica, que es guardin els canvis de manera correcta i veure-ho a la pàgina del curs</li> </ul>
US11	Com a administrador vull poder donar rols de professor	<ul style="list-style-type: none"> <li>- L'admin pot accedir a la pàgina de donar rols des de qualsevol pàgina</li> <li>- L'admin pot donar el rol de professor a un usuari i que es guardi de manera correcta</li> </ul>



User Story	Descripció	Criteris d'acceptació
US12	Com a administrador vull poder veure tots els cursos creats	<ul style="list-style-type: none"> <li>- L'administrador pot veure la pàgina principal</li> <li>- L'administrador pot veure tots els cursos creats</li> </ul>
US13	Com a administrador vull poder borrar cursos	<ul style="list-style-type: none"> <li>- L'administrador pot veure la pàgina principal</li> <li>- L'administrador pot entrar al curs que vol borrar</li> <li>- L'administrador pot borrar aquell curs i que es guardi correctament</li> </ul>
US14	Com a estudiant vull poder veure totes les qualificacions a la vegada	<ul style="list-style-type: none"> <li>- L'estudiant pot accedir a la pàgina de qualificacions des de qualsevol lloc de la pràctica</li> <li>- L'estudiant pot veure tots els cursos als que està assignat</li> <li>- L'estudiant pot veure les pràctiques de cada curs i quina nota té associada</li> </ul>
US15	Com a professor vull poder veure els fitxers extra que em proporciona la correcció automàtica	<ul style="list-style-type: none"> <li>- El professor pot accedir al detall de la pràctica</li> <li>- El professor pot accedir als fitxers que ha proporcionat l'eina de correcció automàtica</li> </ul>
US16	Com a professor vull poder afegir nous estudiants a un curs	<ul style="list-style-type: none"> <li>- El professor pot accedir al detall del curs</li> <li>- El professor pot veure tots els estudiants associats a aquell curs</li> <li>- El professor pot adjuntar un nou fitxer plantilla amb els estudiants nous</li> <li>- El professor pot guardar els canvis de forma correcta i veure'ls en els estudiants associats al curs</li> </ul>

Taula 4: Backlog de l'aplicació

### 3 Arquitectura de l'aplicació

En aquesta secció presentaré l'arquitectura de l'aplicació i els motius pels quals he triat aquestes opcions, que es poden veure a la **Figura 1**. Parlarem de què és RabbitMQ, de com s'han estructurat les dades a MongoDB, quines taules hi ha i com es relacionen entre elles a PostgreSQL, com és el sistema de directoris de l'aplicació i, finalment, quins llenguatges i tecnologies hem utilitzat per realitzar tot això.

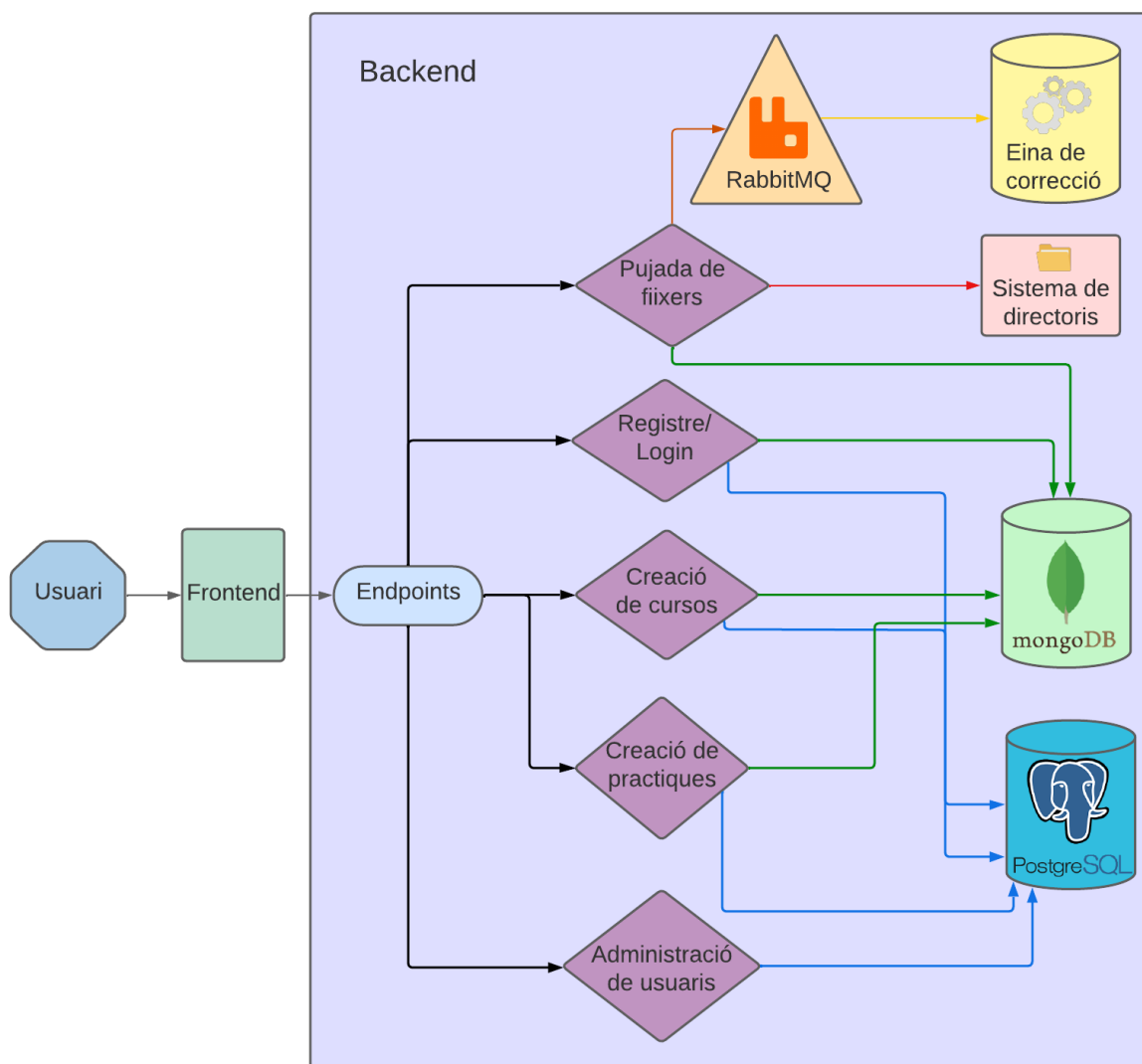


Figura 1: Arquitectura de l'aplicació

## 3.1 PostgreSQL

PostgreSQL [8], conegut com a "Postgres", és un sistema de gestió de bases de dades relacional d'objectes àmpliament utilitzat i de codi obert. Destaca per la seva versatilitat en el tractament de dades estructurades i la seva àmplia gamma de tipus de dades, incloent-hi funcionalitats avançades com ara tipus compostos i geoespacionals. Amb suport a transaccions ACID, PostgreSQL assegura la consistència i fiabilitat de les dades, sent una opció robusta per al tipus d'aplicació que estem dissenyant.

A nivell de rendiment, PostgreSQL destaca per la seva escalabilitat, permetent gestionar bases de dades petites i grans amb eficiència. L'optimitzador de consultes avançat millora l'eficiència en l'accés a les dades, i les opcions de replicació i alta disponibilitat garanteixen una operativitat contínua.

L'aplicació triada per gestionar de manera eficaç les dades de PostgreSQL ha estat PGAdmin **Fig 2**. La seva interfície amigable i funcionalitats potents han jugat un paper clau en la simplificació de les tasques de gestió de bases de dades amb la seva consola SQL integrada i la capacitat de gestionar múltiples connexions.

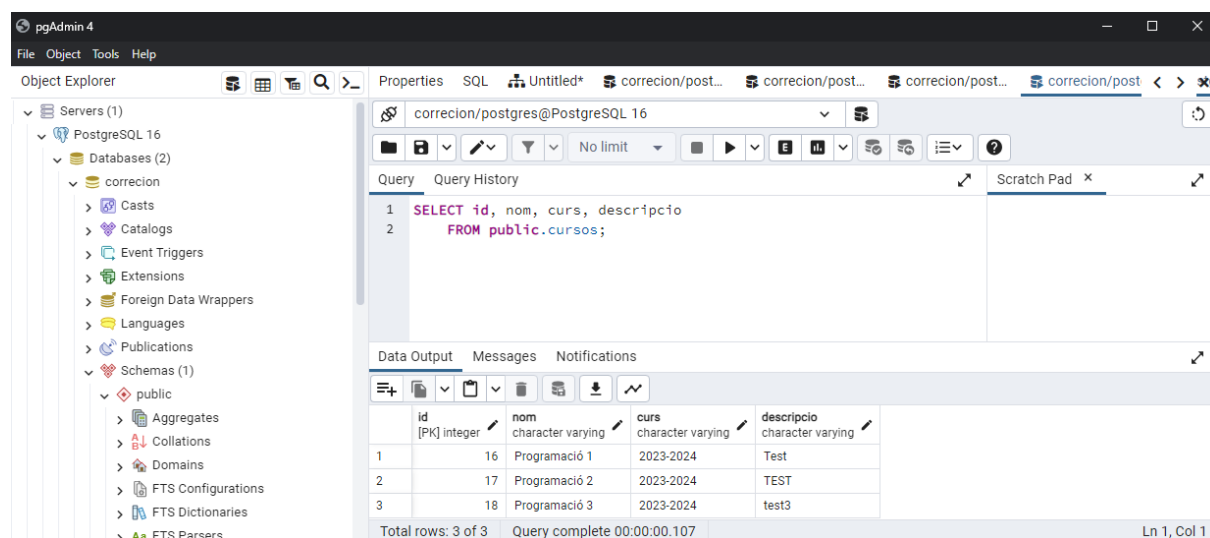


Figura 2: PGAdmin

Per últim, explicaré les taules que he creat per emmagatzemar les meves dades i les relacions que tenen entre elles representades a la **Fig 3**.

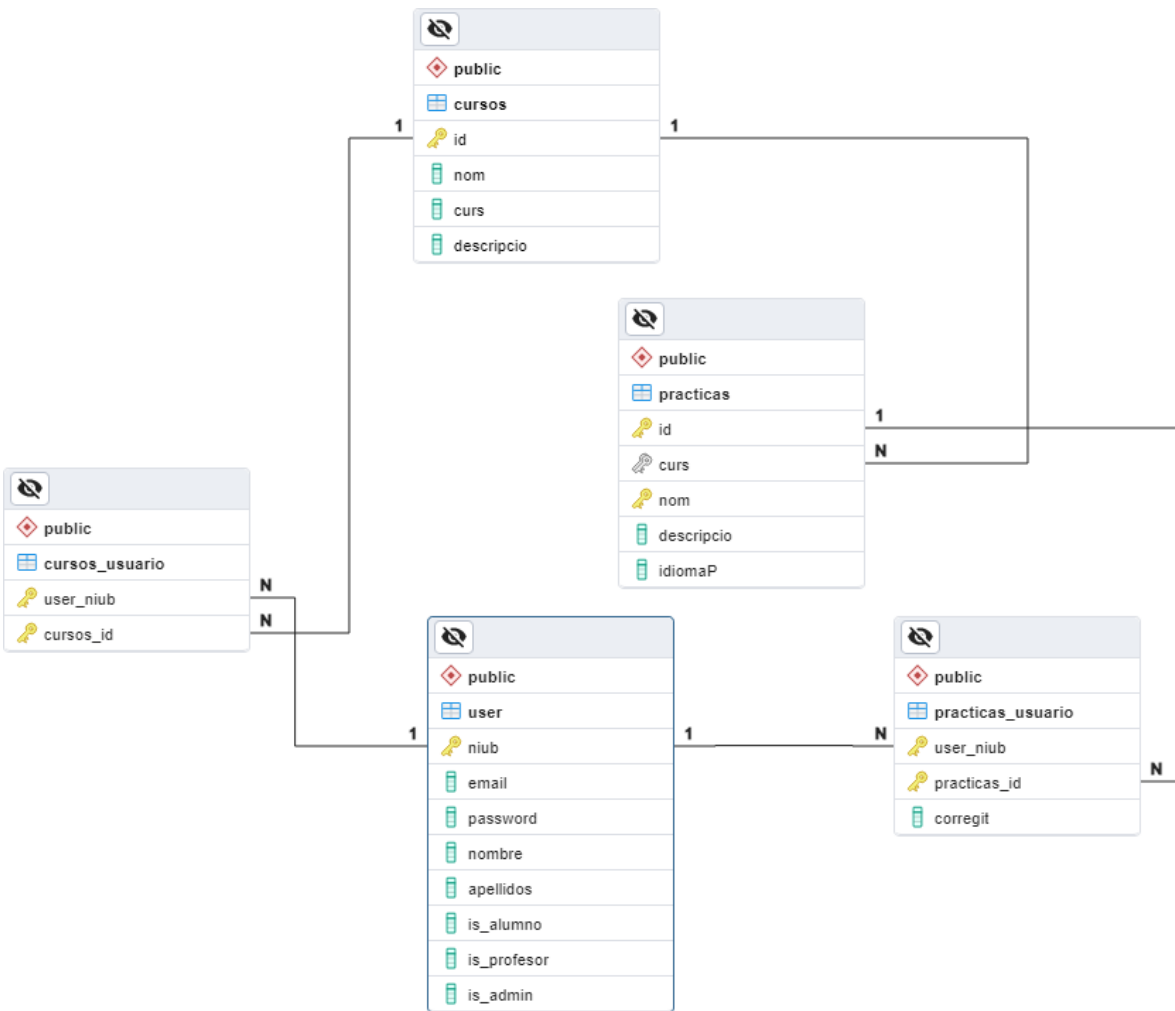


Figura 3: Diagrama Entitat-Relació

- La taula "user" és la taula principal, emmagatzema els usuaris que tenen accés a l'aplicació i conté la informació necessària per a poder fer un ús correcte de l'app. La clau primària (CP) és el niub del estudiant o el niu del professor i és la que ens permetrà saber amb quin usuari estem tractant en l'app. I tant el niub com l'e-mail són els que ens permetran connectar a l'app. L'altra informació que es guarda en aquesta taula són la contrasenya de l'usuari, que es guarda en forma de hash, el nom i el cognom de l'usuari i per últim, però bastant important, estan les columnes: `is_alumno`, `is_profesor`, `is_admin`, que són les columnes que guarden quins rols té aquell usuari i són les que ens permetran amagar o inutilitzar funcions de l'aplicació depenent del rol que tenen.
- La taula "cursos" és la taula que emmagatzema la informació dels cursos que

s'han creat. La clau principal és l'id del curs, que en ser autoincremental i fer-la un índex, ens permetrà accedir a la informació de forma molt més ràpida. També en aquesta taula es guarda el nom, el curs (l'any acadèmic) actual i una breu descripció, informació necessària per a després mostrar-la en el frontend.

- La taula "**practicas**" és una taula derivada de cursos que emmagatzema la informació de les pràctiques que tenen els cursos. La seva clau principal és un id únic autoincremental i indexat. També té una columna, **curs**, que emmagatzema l'id del curs al qual està lligat aquesta pràctica i dues columnes d'informació per al frontend com són les columnes **nom** i **descripció**. Per últim, està la columna **idiomaP**, aquesta columna és molt important, ja que emmagatzema en quin idioma de programació s'ha de fer la pràctica, aquest valor és el que passarem al corrector automàtic i és el que determinarà com s'ha de corregir la pràctica.
- La taula "**cursos\_usuario**" estableix vincles entre els usuaris i els cursos, utilitzant claus foranes per connectar les claus primàries de les taules "**user**" i "**cursos**". La combinació d'aquestes claus foranes forma una clau primària composta, assegurant relacions úniques entre usuaris i cursos.
- La taula "**practicas\_usuario**" estableix la connexió entre usuaris i les pràctiques que han realitzat. Aquesta taula utilitza claus foranes per referenciar les taules "**user**" i "**practicas**", amb una columna addicional que indica si la pràctica ha estat corregida o no.

## 3.2 Mongo DB

MongoDB [6] destaca com a base de dades NoSQL orientada a documents, popularment utilitzada en entorns moderns com aplicacions web i mòbils. Distingint-se dels sistemes relacionals tradicionals, emmagatzema dades en documents JSON anomenats "col·leccions" oferint una gran flexibilitat en l'esquema de dades. La principal fortalesa de MongoDB rau en la seva capacitat d'escalabilitat horitzontal, permetent gestionar grans volums de dades i adaptar-se a càrregues de treball intenses. A més, la seva flexibilitat d'esquema permet afegir o modificar camps sense la necessitat de migrar dades, fent aquesta una de les primeres raons per les quals em vaig decantar a l'hora d'escollir-la com a base de dades addicional.

Per poder fer ús de les bases de dades Mongo es pot fer de diferents maneres, però em vaig decantar pel seu servei al núvol. MongoDB Atlas, un servei totalment gestionat, facilitant l'accés a MongoDB, eliminant la complexitat de la gestió i el desplegament de la infraestructura de manera local. L'ús de MongoDB Atlas comporta avantatges com la configuració sense complicacions, el monitoratge eficient, còpies de seguretat fiables i escalabilitat. Tots aquests avantatges han estat fonamentals per establir i gestionar de manera segura i eficient la base de dades del projecte. Amb aquest servei, s'ha creat una plataforma completa i gestionada, adaptada a les necessitats requerides del projecte, garantint una gestió de bases de dades efectiva al núvol.

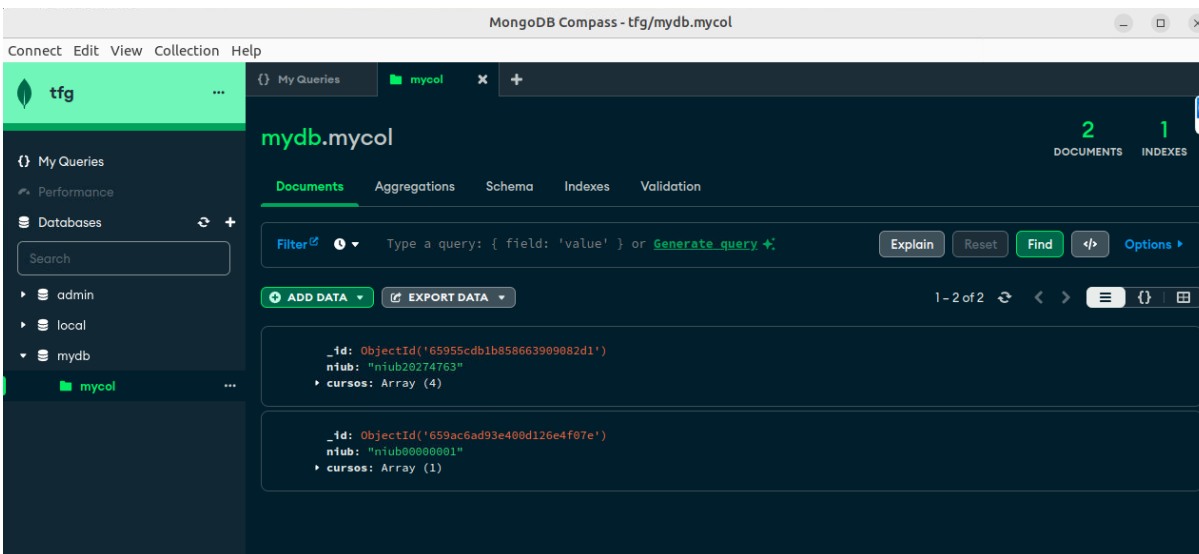


Figura 4: MongoDB Compass

L'eina gràfica que he utilitzat per veure com estava l'estat de la base de dades ha estat MongoDB Compass **Fig 4**. Aquesta aplicació és una eina dissenyada per simplificar la interacció amb bases de dades MongoDB, oferint una interfície intuïtiva i funcionalitats que milloren la productivitat dels desenvolupadors i administradors. Destaca la seva adapta-

bilitat a consultes avançades, la gestió visual de dades i la facilitat d'ús, tot contribuint a una experiència d'usuari eficient i satisfactòria. En la meva experiència personal, MongoDB Compass ha millorat significativament la manera com exploro, analitzo i manipulo les dades en el projecte.

Finalment, explicaré com s'estructuren les dades a la meva MongoDB, representades a la **Fig 5**. Cada usuari té el seu document personal d'on traurà la informació de les correccions:

```
  _id: ObjectId('659ac6ad93e400d126e4f07e')
  niub: "niub00000001"
  cursos: Array (1)
    0: Object
      id: 5
      practicas: Array (1)
        0: Object
          id: 2
          fichero: "./api/correcciones/alumnos/2023-2024/Test/Test/niub00000001/submission..."
          correccion: Object
```

Figura 5: Estructura de un document

- **niub** : Aquest camp es comparteix amb la taula "user" i és la que serveix per identificar de quin usuari és aquest document.
- **cursos** : Aquest camp és una llista que guardarà la informació de tots els cursos als quals està assignat l'usuari.
- **id** : Aquest camp és l'identificador del curs, es comparteix amb l'id de la taula "cursos" i ens servirà per tenir control de quin curs estem recuperant o canviem informació.
- **practiques** : Aquest camp és una llista que guardarà la informació de totes les pràctiques associades a aquell curs.
- **id** : Aquest camp és l'identificador de la pràctica, es comparteix amb l'id de la taula "practicas" i ens servirà per tenir el control de quina pràctica estem recuperant o canviant informació.
- **fichero** : Aquest camp és el que guarda el path d'on està guardat la pràctica de l'alumne o pel contrari, si l'usuari és un professor el que fa és guardar el path on estan guardats els fitxers de correcció.
- **correccion** : Aquest camp és el que guarda l'objecte JSON que torna el servidor de correcció automàtica.

### 3.3 Sistema de Directoris

Un problema crucial que ens va sorgir inicialment durant el TFG era determinar la ubicació adequada per emmagatzemar els fitxers de correcció carregats pels professors i les pràctiques carregades pels alumnes. En un primer moment, vaig considerar l'ús de sistemes d'emmagatzematge al núvol. No obstant això, ràpidament vam descartar aquesta opció, ja que el volum d'informació podria créixer significativament amb l'escalada de l'aplicació i, per tant, no volíem dependre d'altres recursos externs més enllà del servidor de la universitat. En conseqüència, vam optar per la decisió de fer que el servidor també funcionés com a sistema d'emmagatzematge per a tots els fitxers carregats des del frontend. A més, aquesta solució ha simplificat el procés de transport dels fitxers cap a l'eina de correcció automàtica, ja que ara no depenem de cap protocol SFTP o d'un pas addicional per al transport dels fitxers; el servidor agafa els fitxers directament dels directoris mitjançant la seva ruta completa.

A continuació, detallarem el sistema de directoris que es va generar cada vegada que un professor crea un curs o una pràctica o quan un alumne carrega una pràctica.

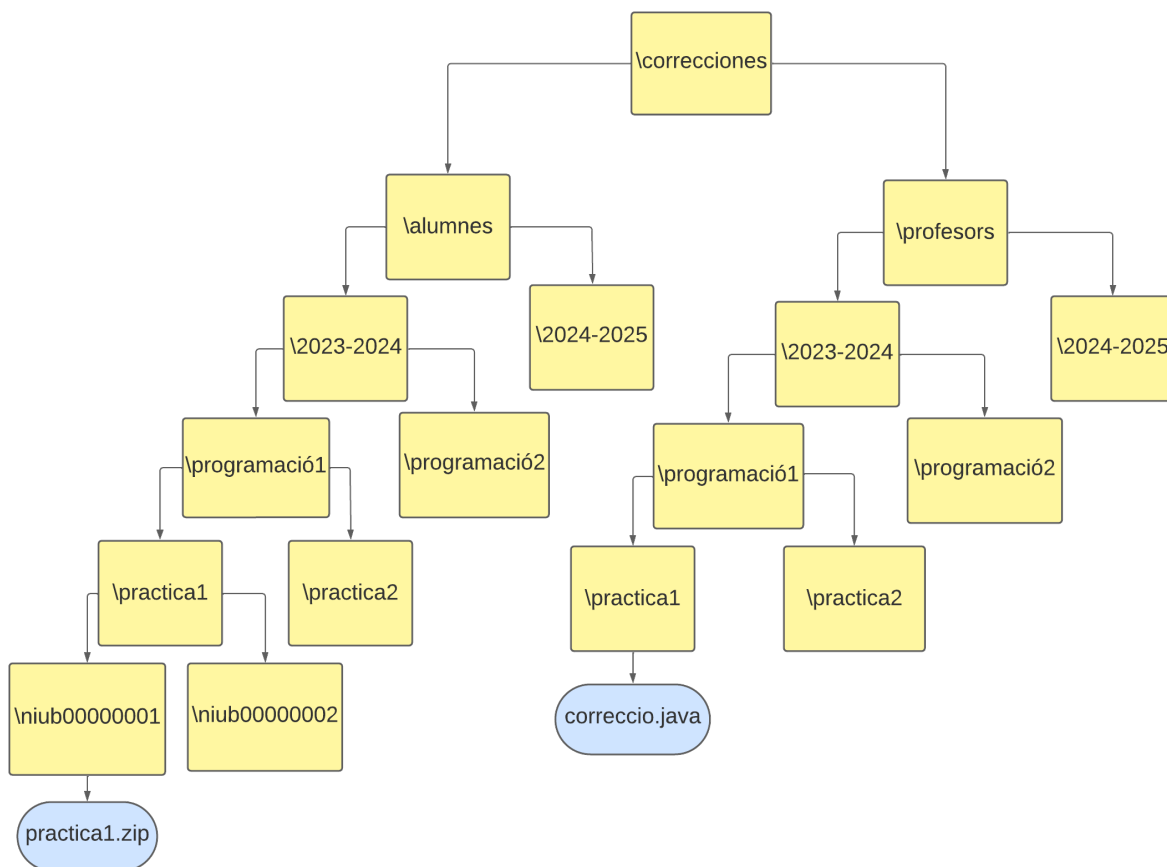


Figura 6: Sistema de Directoris



Com es pot apreciar a la Fig 6, disposem d'un directori principal anomenat "correccions" que es bifurca en diverses carpetes. Les primeres carpetes que emergeixen són "alumnes i "professors". Aquesta diferenciació de carpetes resultarà beneficiosa per mantenir separats els fitxers de correcció de les pràctiques, facilitant així la gestió per part de l'eina de correcció. A més, aquests directoris són estàtics, no varien. A partir d'aquí, tots els directoris es generen de manera que es van creant assignatures i pràctiques. Com es pot observar, els directoris comencen a ser anomenats de manera similar tant per a la part dels alumnes com per a la part dels professors.

Inicialment, es crea el directori del curs amb el format "any-any". Encara que aquest directori podria semblar irrelevant, ja que tant els alumnes com els professors estan centrats en un curs acadèmic a la vegada, té una importància més gran del que pot semblar. A l'hora de passar de curs, tots els fitxers utilitzats queden obsolets i aquesta estructura facilita l'eliminació de la informació continguda en aquests directoris. El següent directori és l'assignatura, que servirà per contenir les pràctiques. Com els noms de les pràctiques són únics, no hi haurà problemes de compatibilitat.

Aquest mateix principi es pot aplicar als següents directoris, que representen les pràctiques. Gràcies als noms únics, no es generen conflictes, i el directori pare és l'assignatura a la qual estan lligades. Aquí es fa una distinció entre alumnes i professors, ja que la branca dels professors acaba en aquest punt. Tots els fitxers de correcció s'emmagatzemen dins del directori de la pràctica, mentre que els alumnes tindran un últim directori amb el seu número identificatiu universitari (NIUB), establint així la distinció d'on es guardarà cada pràctica. Aquesta estructura serà útil a l'hora de realitzar la primera correcció automàtica. Ja que aquesta es fa immediatament després de pujar el fitxer, aleshores tenir-ho per separat pel (NIUB) fa que es puguin passar les correccions de manera individual.

### 3.4 RabbitMQ

RabbitMQ[10] és un sistema de missatgeria de codi obert que facilita la comunicació entre diversos components d'una aplicació distribuïda. Implementant el protocol de missatgeria avançat (AMQP), RabbitMQ possibilita un intercanvi eficient de missatges entre productors i consumidors.

Mitjançant RabbitMQ, els missatges es poden enviar i rebre de manera asíncrona, afavorint una integració flexible i escalabilitat en entorns distribuïts. A més, RabbitMQ suporta diversos patrons de missatgeria, incloent-hi cues, temes i canals directes, oferint opcions diverses per gestionar el flux de missatges.

En el context de la nostra aplicació, RabbitMQ ha ocupat un paper fonamental en la connexió entre el backend i l'eina de correcció automàtica. Aprofitant el seu sistema de cues, cada vegada que un alumne carrega una pràctica, s'envia un missatge a l'eina de correcció, activant el seu programari i retornant una resposta amb la correcció de la pràctica. Per a la implementació d'aquesta funcionalitat, hem optat pel protocol de Remote Procedure Call (RPC) **Fig 7**, que permet l'execució de procediments en una màquina externa (en el nostre cas, un altre usuari de Linux). Aquest enfocament simula eficaçment una trucada a una funció del programa, proporcionant una resposta que posteriorment emmagatzemem a MongoDB. Encara que tot això ha portat problemes que hem solucionat i explicaré a continuació.

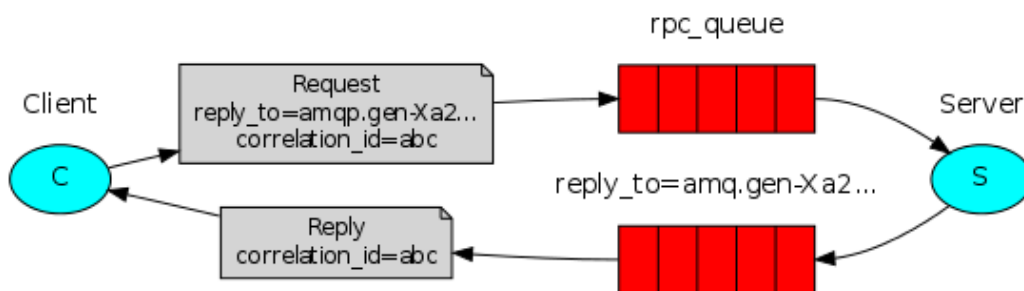


Figura 7: Protocol RPC

#### 3.4.1 Problemes Sorgits

Al ser la primera vegada que treballava amb una cua de missatges, ha estat tot un repte fer-la funcionar correctament. Un dels principals obstacles que vam enfrontar va ser la gestió de la concurrència en l'ús d'una cua de missatges amb el protocol RPC. El primer problema que va sorgir va ser que quan s'intentava fer una crida a la cua de missatges,

aquesta col·lapsava tota l'API, ja que Python, al ser un llenguatge interpretat, executava el codi línia per línia, bloquejant tot el procés fins que arribava la resposta per la cua.

Per abordar aquest problema, vaig intentar implementar fils d'execució per gestionar les crides a la cua de missatges i esperar les respostes. Això va permetre alliberar el fil principal i mantenir l'API en funcionament. No obstant això, vaig enfrontar un nou problema: Pika, la llibreria que gestiona aquestes crides, no és "threadsafe".

#### **3.4.1.1 Pika**

Pika[7] és una llibreria de Python que actua com a client per a RabbitMQ, facilitant la interacció amb aquest sistema de missatgeria. Tot i això, Pika presenta limitacions a l'hora de gestionar l'asincronia del codi. Després de diversos intents sense èxit de solucionar aquest problema, vaig descobrir Aio-Pika, una derivada de Pika orientada a la programació asíncrona.

#### **3.4.1.2 Aio-Pika**

Aio-Pika [1] és una llibreria Python que ofereix suport per a la programació asíncrona en l'ús de RabbitMQ. Adaptada a entorns "asyncio", Aio-Pika millora el rendiment i l'eficiència en comparació amb la seva contrapartida sincrònica. Amb l'ajuda d'aquesta llibreria i FastAPI es van solucionar els problemes de concurrència a l'aplicació. Ara, amb la crida a RabbitMQ, un fil d'execució espera que tot el procés amb la cua finalitzi, i l'ús d'async i await proporcionats per FastAPI permet que els endpoints es puguin utilitzar mentre es corregeixen pràctiques.

## 3.5 Eines Utilitzades

Per realitzar el projecte, he utilitzat diverses eines per ajudar-me a escriure un codi més net i comprensible, facilitant així futurs canvis o ampliacions per a les persones que treballin en el projecte en un futur. A continuació, explicaré les diferents tecnologies, llenguatges de programació i frameworks que he fet servir.

### 3.5.1 Llenguatges de Programació

#### 3.5.1.1 HTML

HTML [4], o Llenguatge de Marcatge d'Hipertext, és l'estàndard utilitzat per crear i organitzar contingut en una pàgina web. A través d'etiquetes i elements, HTML determina com s'estructuren i presenten textos, imatges, enllaços i altres elements multimèdia en una pàgina web. És fonamental per construir interfícies web i ofereix l'estructura bàsica d'un lloc web.

#### 3.5.1.2 CSS

CSS [3] és un llenguatge utilitzat per estilitzar i donar format a les pàgines web creades amb HTML. Mitjançant regles de selecció i declaracions, CSS permet controlar l'aparença i la disposició del contingut en una pàgina. Aquest llenguatge ofereix la possibilitat de personalitzar fonts, colors, marges, alineacions i altres propietats visuals, proporcionant flexibilitat i coherència estètica a través d'un lloc web o aplicació web. En resum, CSS compleix un paper crucial en millorar la presentació i l'experiència de l'usuari a les pàgines web.

#### 3.5.1.3 JavaScript

JavaScript [5] és un llenguatge de programació fonamental per a la creació de pàgines web interactives. Opera al costat del navegador i permet als desenvolupadors gestionar dinàmicament els elements HTML i CSS, així com respondre als esdeveniments de l'usuari.

Pel que fa a l'enllaç amb el backend, JavaScript utilitza llibreries de crides com Axios o Fetch. Aquestes llibreries faciliten la comunicació asíncrona amb el backend, permetent al frontend obtenir i enviar dades sense necessitat de recarregar la pàgina. Això millora l'experiència de l'usuari i l'eficiència de l'aplicació web.

En conjunt, JavaScript i les seves llibreries són essencials per al desenvolupament web modern, permetent la creació d'aplicacions dinàmiques i interconnectades tant al costat del client com al servidor.

### 3.5.1.4 Python

Python [9] és un llenguatge de programació versàtil i de propòsit general amb moltes característiques notables que el destaquen en diversos àmbits. La seva sintaxi fàcil i llegible fa que sigui accessible per als desenvolupadors, tant novells com experimentats. La capacitat de ser interpretat, sense necessitat de compilació, agilitza significativament el procés de desenvolupament, permetent una iteració ràpida i una prova immediata del codi.

En el marc del nostre projecte, Python mostra la seva utilitat destacada en el desenvolupament d'APIs REST, especialment amb l'ús de frameworks com FastAPI i SQLAlchemy. FastAPI ofereix rendiment i facilitat d'ús, mentre que SQLAlchemy simplifica la interacció amb bases de dades relacionals mitjançant l'ús d'Object-Relational Mapping (ORM). Aquesta combinació de tecnologies permet crear de manera eficient APIs potentment funcionals.

En resum, Python no només destaca per la seva eficiència i facilitat d'ús, sinó que també és una elecció potent i versàtil en el món de la programació. La seva adaptabilitat i l'abast de les seves llibreries el converteixen en una opció ideal per a projectes de desenvolupament web, així com per a tasques que involucren anàlisi de dades i aprenentatge automàtic. És per això que he escollit Python per desenvolupar el meu projecte.

## 3.5.2 FrameWorks

Un framework en el desenvolupament de programari és una estructura reutilitzable que simplifica la creació d'aplicacions, permetent als desenvolupadors centrar-se en la seva lògica específica. Amb eines predefinides i convencions, facilita el desenvolupament i promou la coherència del codi.

### 3.5.2.1 React

React [11] és una potent biblioteca de JavaScript desenvolupada per Facebook, àmpliament utilitzada per construir interfícies d'usuari interactives i eficients. El seu enfocament principal es basa en la creació de components reutilitzables, que actuen com a blocs de construcció per representar parts específiques de la interfície d'usuari. Aquesta estructura modular permet a cada component gestionar el seu propi estat i comportament, facilitant la construcció i el manteniment d'aplicacions complexes.

React adopta un enfocament basat en components, descomponent la interfície d'usuari en petites peces independents, anomenades components. Aquesta abstracció millora la modularitat i la reutilització del codi, essencial per al desenvolupament eficient d'aplicacions.

El paradigma de programació declarativa de React permet als desenvolupadors descriure com hauria de semblar la interfície d'usuari en funció de l'estat de l'aplicació, delegant a React la responsabilitat de gestionar els canvis de manera eficient. Això millora la predictibilitat i la sostenibilitat del codi.

Un aspecte clau és l'ús del DOM virtual, una tècnica que permet a React realitzar actualitzacions de manera eficient al comparar el DOM virtual amb el DOM real i aplicar només els canvis necessaris. Aquesta optimització contribueix a la rapidesa i l'eficiència de les actualitzacions d'interfície d'usuari.

JSX és una extensió de la sintaxi de JavaScript utilitzada a React per descriure com ha de ser renderitzada la interfície d'usuari. Amb JSX, els desenvolupadors poden escriure codi amb una sintaxi semblant a HTML dins dels components de React. Tot i que aquests elements semblen etiquetes HTML, en realitat es converteixen en crides a funcions de React per crear elements virtuals. Això proporciona una manera llegible i expressiva de definir l'estructura de la interfície d'usuari dins del codi JavaScript, millorant la sostenibilitat i la comprensió del codi.

A més, React utilitza el DOM virtual per optimitzar les actualitzacions del DOM real, millorant encara més l'eficiència i la velocitat de l'aplicació. En resum, React, juntament amb JSX i el DOM virtual, proporciona una eina versàtil i potent per al desenvolupament web modern, amb una estructura modular, rendiment eficient i gestió elegant de l'estat de l'aplicació.

### 3.5.2.2 FastAPI

FastAPI [2] és un framework web modern per al desenvolupament ràpid d'APIs amb Python basat en anotacions. Aquesta eina permet crear APIs eficients i amb un alt rendiment amb una sintaxi clara i fàcil d'entendre.

Amb FastAPI, es poden definir ràpidament les rutes de l'API utilitzant anotacions per especificar els tipus de dades i els paràmetres que s'esperen. Aquesta informació no només facilita la comprensió del codi, sinó que també permet a FastAPI generar automàticament documentació interactiva Swagger **Fig 8**.



users	
POST	/users/profesor Give Profesor
DELETE	/users Deleteuser
POST	/users/ Create User
POST	/users/login Login
POST	/users/uploadfile Login
cursos	
GET	/cursos Get Cursos
POST	/cursos Crear Nou Curs
POST	/cursos/practica Crear Practicas
GET	/cursos/csv Get Csv
GET	/cursos/excel Get Excel
GET	/cursos/curs/practicas/{id} Get Practiques

Figura 8: Swagger

FastAPI aprofita les funcionalitats de les anotacions per validar les dades d'entrada i sortida de les rutes, millorant la seguretat i reduint errors potencials. A més, utilitza asincronia per gestionar moltes connexions concurrents amb eficàcia, assegurant un alt rendiment fins i tot en aplicacions amb càrregues de treball exigents.

En resum, FastAPI ofereix una manera eficient i moderna de construir APIs amb Python, prioritzant la simplicitat, la seguretat i el rendiment.

### 3.5.2.3 SQLAlchemy

SQLAlchemy [12] és un framework de mapeig relacional d'objectes (ORM) per a Python que facilita la interacció amb bases de dades relacionals. Aquesta eina ofereix una abstracció eficient i flexible per gestionar les dades a través de consultes en lloc d'utilitzar directament SQL.

Amb SQLAlchemy, els desenvolupadors poden definir models d'objectes Python que es mapegen directament a taules de bases de dades. Aquest mapeig simplifica la gestió de les dades i permet fer servir Python com a llenguatge de consulta en lloc de SQL.

SQLAlchemy proporciona una interfície potent per a realitzar consultes complexes, facilitant l'obtenció i la manipulació de dades d'una manera intuïtiva. A més, ofereix suport

per a diverses bases de dades relacionals, com MySQL, PostgreSQL i SQLite, permetent als desenvolupadors treballar amb diferents sistemes de gestió de bases de dades sense haver de canviar radicalment el codi.

En resum, SQLAlchemy simplifica la interacció amb bases de dades relacionals mitjançant un ORM potent i flexible, permetent als desenvolupadors gestionar les dades de manera eficient fent servir Python.

#### **3.5.2.4 Tailwind**

Tailwind [13] CSS és utilitzat per simplificar i accelerar el desenvolupament d'interfícies d'usuari. En lloc de fer servir classes predefinides, com en altres frameworks, Tailwind ofereix un abordatge diferent basat en utilitats.

En Tailwind, les utilitats són classes de baix nivell que s'apliquen directament als elements HTML. Això permet als desenvolupadors estilitzar els components i les pàgines de manera ràpida i senzilla, sense necessitat de definir classes prèvies al CSS.

Aquest framework proporciona una gran quantitat de classes utilitats que cobreixen una àmplia gamma de propietats CSS, com mides, colors, espais, alineacions, etc. Això ofereix flexibilitat i agilitat a l'hora de construir i estilitzar elements.

Tailwind també facilita la personalització, permetent als desenvolupadors configurar els seus propis estils i fer servir-los a través de les classes utilitats.

En resum, Tailwind CSS és una eina eficient i personalitzable per a l'estilització d'interfícies d'usuari, amb un enfocament que agilitza el procés de desenvolupament i permet una gran flexibilitat en el disseny i l'estil. Permetent als desenvolupadors gestionar les dades de manera eficient fent servir Python.

### **3.5.3 Plataformes utilitzades**

#### **3.5.3.1 Github**

GitHub és una plataforma de desenvolupament de programari que ofereix serveis de control de versions utilitzant Git. Ha estat una eina essencial per a mi, ja que l'he fet servir per emmagatzemar i gestionar les diferents versions del meu codi.

A més, GitHub ha estat fonamental per facilitar la transició del meu codi des del meu ordinador personal al servidor de la universitat. Amb l'ús de repositoris remots, he pogut transferir el codi de manera eficient i mantenir una sincronització actualitzada entre els diferents entorns de treball. Això no només ha simplificat el procés de desenvolupament, sinó que també ha proporcionat una plataforma centralitzada per a la col·laboració i el seguiment del projecte.



## 4 Implementació i Resultats

### 4.1 Disseny

La concepció central del disseny de l'aplicació parteix de la intenció de crear una interfície extremadament senzilla, facilitant l'adaptació de l'usuari a una nova plataforma. Amb aquest objectiu en ment, vaig optar per desenvolupar una plataforma que se semblés al Moodle, però amb una simplificació encara més pronunciada tant en els seus components com en la navegació. Aquest enfocament té com a objectiu proporcionar als usuaris una experiència àgil i fàcil d'utilitzar, inspirant-se en les funcionalitats de Moodle però reduint-ne la complexitat per afavorir una transició més suau i una interacció més intuïtiva.

A continuació, aniré explicant de forma detallada cada pàgina o finestra que té la meva aplicació i el pensament que té darrere.

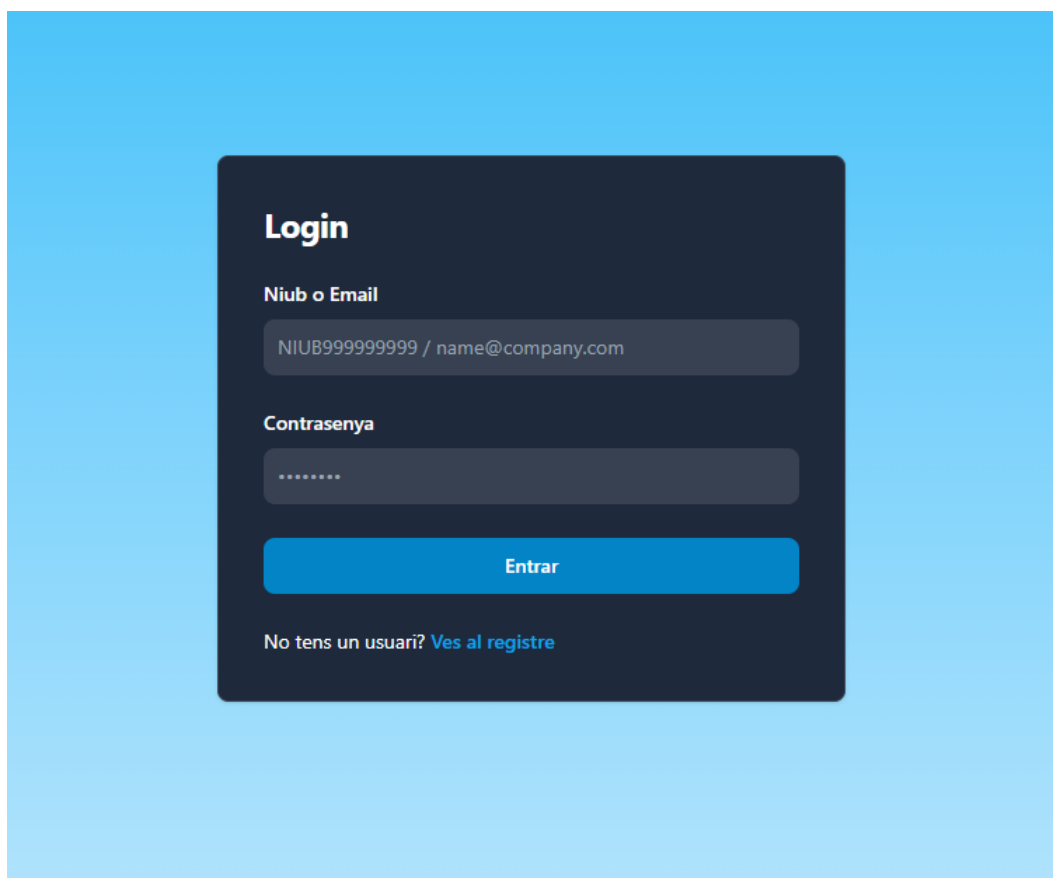


Figura 9: Login

En accedir al lloc web, el primer que trobem és la pantalla d'inici de sessió **Fig 9**. Aquesta interfície de login és bastant senzilla, amb dos camps d'entrada: un per introduir el niub o

l'email amb el qual t'has registrar i l'altre per la contrasenya. A més, disposem d'un botó que permet intentar iniciar sessió i, just a sota d'aquest botó, hi ha l'opció de dirigir-se a la pàgina de registre en cas que encara no tinguis un compte d'usuari creat.

The image shows a registration form on a dark blue background. At the top, the title 'Registre' is displayed in white. Below the title, there are two input fields: 'NIUB' with the value 'NIUB999999999' and 'Nom' with the value 'Albert'. A small text block below these fields reads: 'El niub està al carnet d'estudiant, si no en tens. [Fes click aquí.](#)'. The next row contains 'Cognoms' with 'Roca Perez' and 'Email' with 'name@company.com'. Below this is a 'Contrasenya' field with masked characters '\*\*\*\*\*'. A 'Confirmació' field contains the placeholder text 'Torna a escriure la contrasenya'. Underneath the confirmation field, there are five error messages, each preceded by a red 'X' icon: 'La contrasenya té més de 8 caràcters.', 'La contrasenya té caràcters especials.', 'La contrasenya té un número.', 'La contrasenya té una lletra majúscula.', and 'Les contrasenyes coincideixen.'. At the bottom of the form, there are two large blue buttons: 'Registrar-se' and 'Tornar al Login'.

Figura 10: Registre

A la pàgina de registre **Fig 10**, trobem una interfície semblant a la de Login, però amb més camps d'entrada. Aquests camps són essencials per a la creació d'un nou usuari, destacant-ne tres: el niub, que inclou una ajuda a sota per facilitar la seva correcta introducció; i els camps de contrasenya i confirmació. Aquests dos últims, amb l'assistència de les indicacions que es mostren a sota, guien l'usuari a crear una contrasenya segura i fiable, seguint les normes establertes.

A més dels camps, es presenten dos botons rellevants: “Registrar-se”, que crida al backend per crear un nou usuari, i “Tornar al Login”, que, com indica el seu text, redirigeix a la pàgina d’inici de sessió.

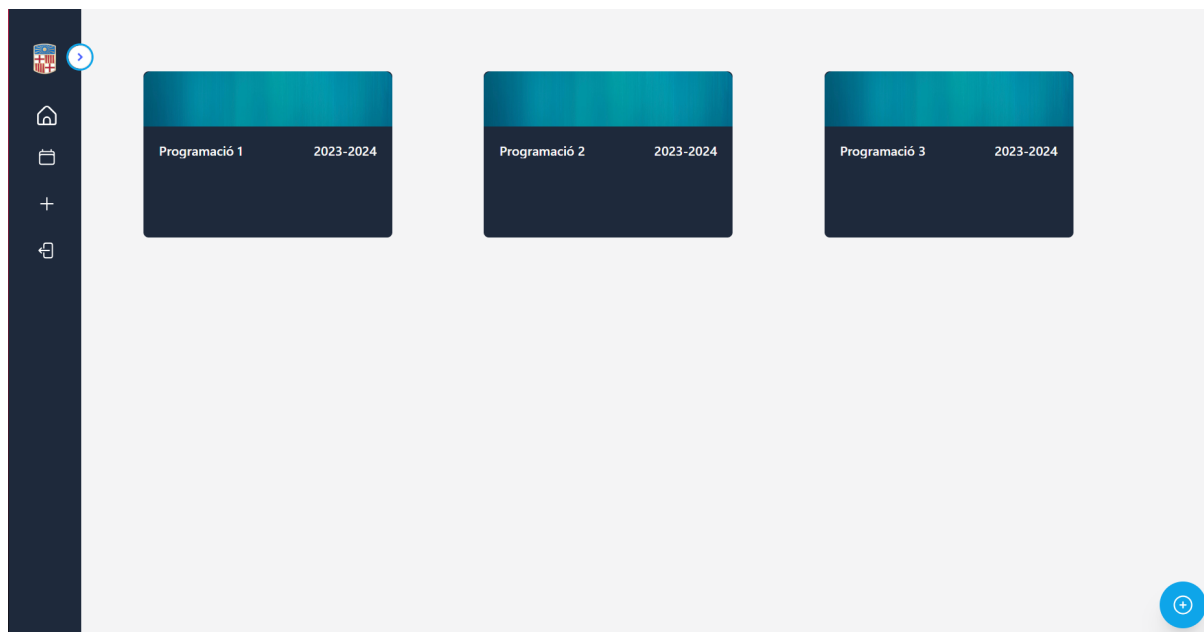


Figura 11: Pàgina Principal

Després d’iniciar sessió, accedim a la pàgina principal, com es mostra a la **Fig 11**, la qual és compartida per tots els usuaris, excepte pel botó situat a la part inferior dreta, visible només pels administradors i professors, el qual explicarem més endavant la seva funcionalitat.

A la pàgina principal, es poden apreciar tres targetes (cards), totes elles amb el mateix format: una imatge a la part superior, el nom de l’assignatura a l’esquerra i el curs a la dreta. A més, una barra lateral (sidebar) també està present, proporcionant una navegació convenient a través de l’aplicació.

El botó situat a la part inferior dreta té la funció d’obrir un modal per a la creació de cursos. Aquesta funcionalitat està destinada a proporcionar una manera eficient i ràpida de crear nous cursos. És important destacar que aquest botó només és visible per als administradors i professors, assegurant un accés restringit a la funcionalitat.

En termes de disseny, la pàgina principal es caracteritza per la simplicitat i l’absència d’elements extravagants. Les targetes amb informació clara sobre assignatures i cursos faciliten la comprensió del contingut. La interfície global busca ser clara i senzilla, optant per una presentació netament estructurada.

**Creació Curs**

Nom Curs

Curs

Descripció

Arrossega aquí els fitxers o

Figura 12: Modal Creació Curs

El modal de creació de cursos, com es mostra a la **Figura 12**, proporciona dos camps de text lliure per al nom i la descripció del curs. Seguidament, trobem un desplegable que permet seleccionar l'any acadèmic del curs. A més d'això, s'incorpora una funcionalitat de “drag and drop”, que facilita la càrrega de fitxers de manera intuïtiva. Aquests fitxers, que es poden carregar mitjançant aquesta funcionalitat, són les plantilles que contenen la llista d'alumnes que seran assignats a aquest curs.

Val la pena destacar que la funcionalitat de “drag and drop” millora significativament l'experiència de càrrega de fitxers, proporcionant una interacció visual i directa que facilita als usuaris la pujada d'aquestes plantilles sense complicacions.

Finalment, es disposa de dos botons d'acció: un per tancar el modal sense efectuar la creació del curs i un altre per desar els canvis realitzats. Aquesta interfície d'usuari opta per una organització clara i intuïtiva dels elements.

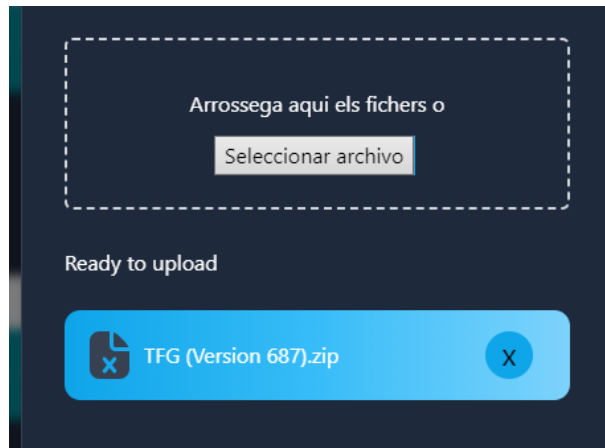


Figura 13: Drag and Drop

El que es mostra a la **Figura 13** és la interfície de tots els “Drags and Drops” disponibles a l’aplicació. Es poden distingir dos elements claus: la zona de caixa per a la pujada de fitxers i una llista situada a la part inferior que mostra els fitxers que s’han carregat. Els elements corresponents als fitxers pujats es poden eliminar de la llista utilitzant el botó en forma de “x” que hi ha a la dreta de cada element. Cal destacar dues funcionalitats del “drag and drop”: en el cas de la creació de cursos només es permet carregar una plantilla de fitxers, de manera que si s’intenta pujar nous fitxers, aquests substitueixen els antics. En canvi, en la creació d’una pràctica, on es poden pujar múltiples fitxers, cada fitxer nou crea un objecte individual a la llista.

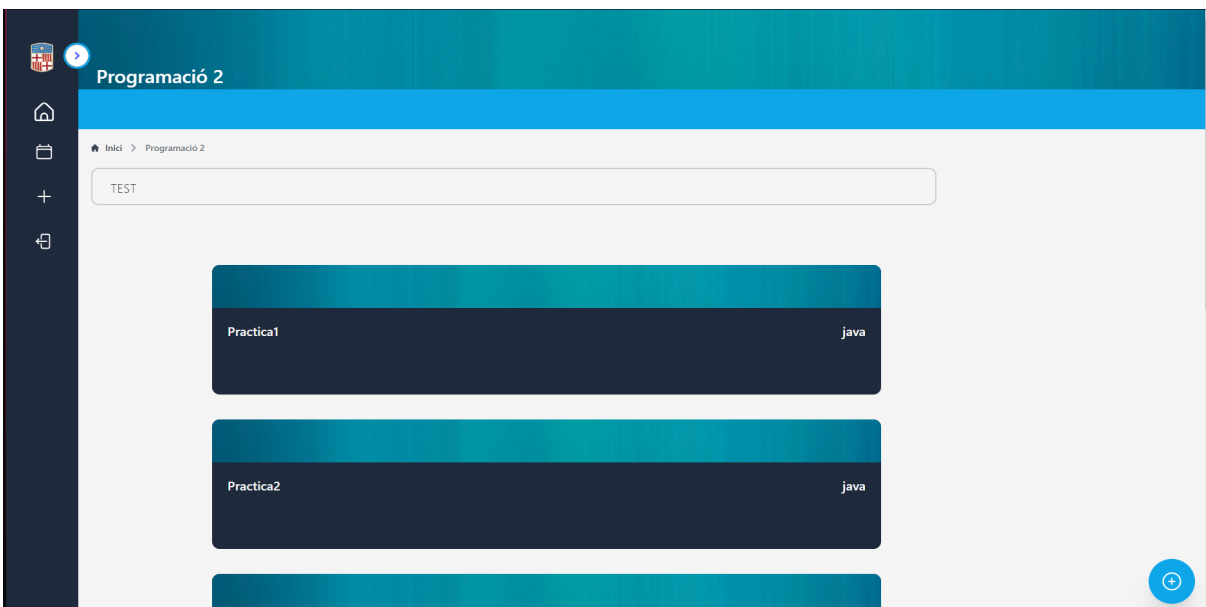


Figura 14: Pàgina Curs

A l'accedir a un curs específic, arribem a una pàgina com la que es mostra a la **Figura 14**. En aquesta imatge, la capçalera presenta la mateixa imatge que la card de la pàgina principal, juntament amb el nom del curs situat a sobre la imatge. A continuació, es troba una barra de color blau que separa el contingut de la capçalera. Just després, es poden veure uns “breadcrumbs” que indiquen la ubicació actual a l'aplicació. A sota d'aquests “breadcrumbs”, trobem un marc amb un vorejat de color gris que conté la descripció del curs. En aquest cas, la descripció és “TESTi s'ha dissenyat per adaptar-se a la longitud del text mantenint la mida de la capsa.

Seguidament, arribem a les cards de les pràctiques. Aquestes redirigeixen a la pàgina del detall de cada pràctica. Com es pot observar, les cards són similars a les de la pàgina inicial, però allargades per proporcionar una aparença simplificada al frontend. Finalment, es destaca un altre botó a la part inferior dreta, visible només pels professors o administradors. Aquest botó obre el modal per a la creació de noves pràctiques.

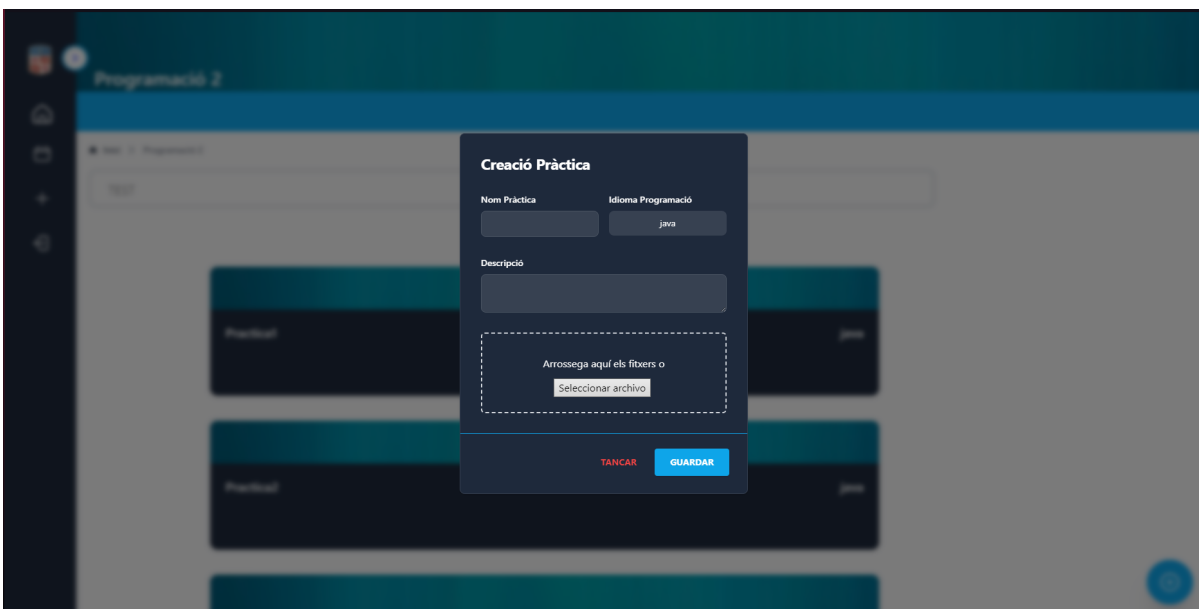


Figura 15: Modal Pràctica

El modal de creació de pràctiques **Fig 15** és molt similar al de creació de cursos, amb l'única diferència que les pràctiques estan associades a un llenguatge de programació en comptes d'un curs, com es pot observar. També és evident que no hi ha botons per descarregar plantilles, ja que aquesta funcionalitat no és necessària en aquest context. Aquesta consistència en el disseny dels modals busca simplificar la interacció de l'usuari amb l'aplicació.

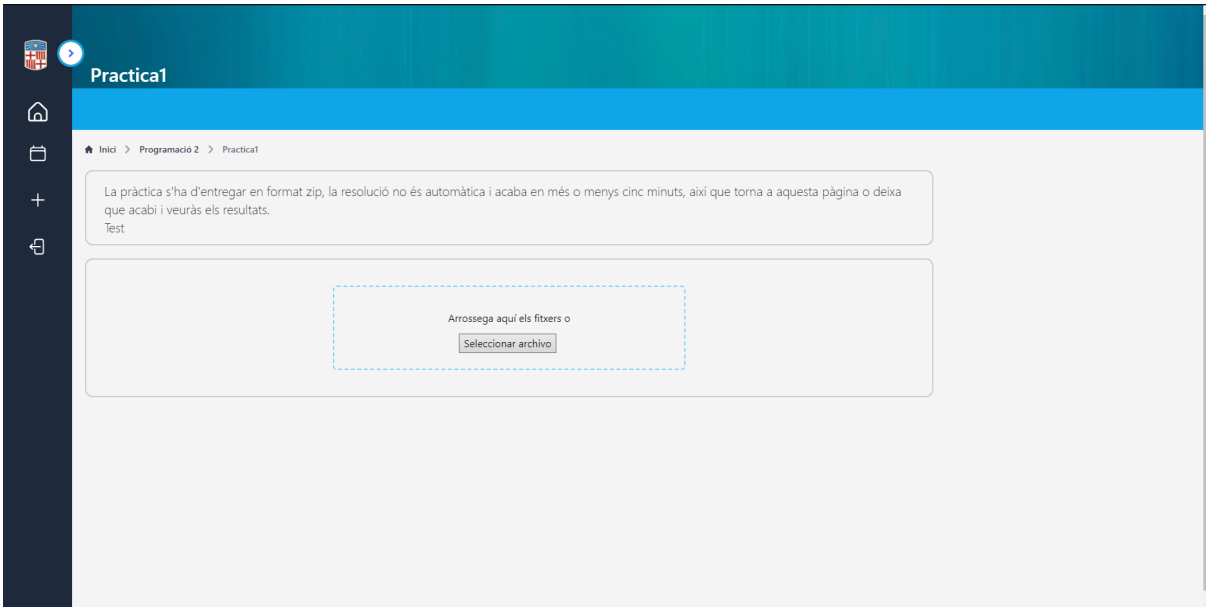


Figura 16: Vista Pràctica

La vista de la pràctica és la que menys s'assembla a les altres, però encara comparteix alguns elements. Manté la mateixa capçalera que la vista d'un curs, canviant únicament el nom del curs pel nom de la pràctica. Els breadcrums s'actualitzen per reflectir l'estat actual de la pàgina.

Apareix un requadre gris amb un text estàndard per a totes les pràctiques, demanant paciència amb les correccions automàtiques. Posteriorment, segueix la descripció detallada de la pràctica. Un següent requadre similar conté una funcionalitat de “drag and drop”. Quan es carrega un fitxer a través d'aquesta funcionalitat, s'activa un botó per fer el lliurament del fitxer o fitxers, depenent de si l'usuari és un alumne o un professor.

Encara que la secció pugui semblar buida inicialment, hi ha un requadre que apareix després que la correcció automàtica ha estat realitzada, mostrant un “OK” que indica que la pràctica ha superat la correcció.

Aquesta vista específica opta per una estructura diferent per adaptar-se a les necessitats particulars de les pràctiques, però manté la coherència amb altres vistes en termes d'elements comuns, proporcionant una experiència d'usuari unificada a través de l'aplicació.

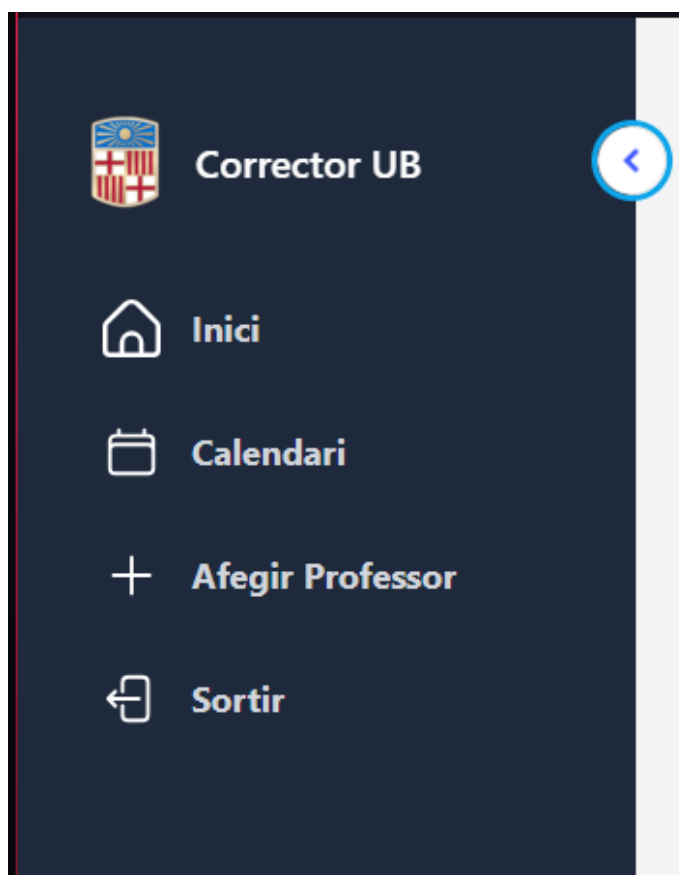


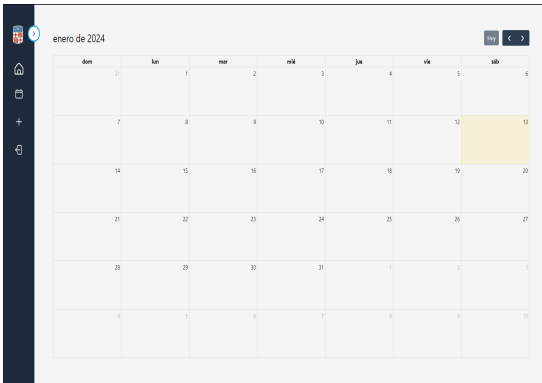
Figura 17: SideBar

La SideBar desplegada, com es mostra a la **Fig 17**, ocupa un paper fonamental en la navegació de l'aplicació. Mitjançant el botó “Inici”, els usuaris poden accedir a la pàgina principal des de qualsevol secció de l'aplicació, garantint una navegació àgil i intuïtiva. Per a la visualització del calendari, l'única opció disponible és utilitzar el botó “Calendari”.

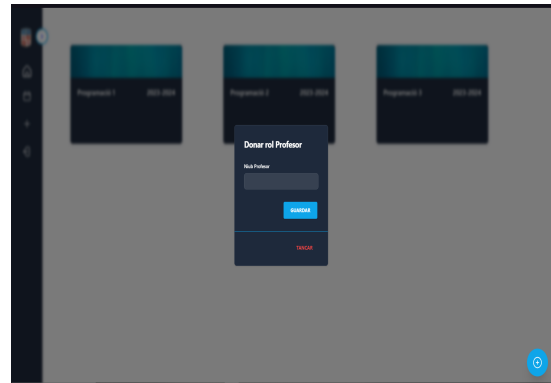
Un aspecte rellevant és la visibilitat selectiva de l'element “Afegir Professor”, reservat únicament per als administradors. Aquest element permet obrir un modal específic per a la gestió dels rols d'usuari, aportant una funcionalitat clau per a l'administració de la plataforma.

Finalment, el botó “Sortir” facilita una sortida ràpida de l'aplicació, redirigint l'usuari directament al formulari del Login. Aquest disseny de la SideBar, amb funcionalitats específiques, contribueix a una experiència de navegació eficient i cohesionada.





(a) Vista Calendari



(b) Modal Rols Professor

Figura 18: Vistes de la aplicació

Les dues últimes vistes s'expliquen molt més fàcilment. La vista **Fig 18a** és un calendari que destaca el dia en el qual ens trobem i permet la navegació cap enrere o cap endavant. La vista **Fig 18b** ens mostra l'últim modal de l'aplicació, que és exclusiu per als administradors. Aquest modal permet introduir qualsevol niub o niu i canviar els rols d'un usuari d'alumne a professor. Aquesta funcionalitat és essencial per als administradors per gestionar els rols dels usuaris i garantir que tots tinguin l'accés adequat dins de la plataforma.

#### 4.1.1 Colors

Els colors en una aplicació són una part que queda en el fons, però són una part crucial. Poden influir en com l'usuari experimenta i percep la plataforma. Els colors poden generar una sensació de tranquil·litat i comoditat o, contràriament, provocar que l'usuari se senti incòmode utilitzant l'aplicació.

Tenint en compte la proximitat de l'aplicació amb la UB, he decidit fer servir colors propers als blaus. Aquesta elecció busca no només mantenir una coherència estètica amb l'entorn universitari, sinó també proporcionar una experiència visual agradable i coherent per als usuaris. L'ús de tailwind també ha ajudat, ja que tenen predefinitos molts colors i moltes tonalitats d'aquests colors fent-ho senzill a l'hora d'escollir els colors.

■ **#1E293B Slate-800**. És el color que predomina a l'aplicació, és una tonalitat fosca que, tot i no ser negra, aporta un fons elegant i agradable als ulls. Aquesta elecció de color més fosc pot ajudar a reduir la fatiga visual, permetent que l'usuari utilitzi l'aplicació de manera més còmoda i sense cansament excessiu. A més, el to fosc pot destacar bé altres elements de la interfície, com textos i botons, creant una jerarquia visual clara. En general, aquest color contribueix a crear una estètica atractiva i funcional per a l'aplicació.

- **#0EA5E9 Sky-500.** El color Sky-500 de tailwind, utilitzat com a complement al color principal de pissarra, té un to més lluminós i vibrant. Aquest contrast entre el color principal i aquest color complementari permet millorar la visibilitat dels elements clau, fent que sigui més fàcil per als usuaris identificar zones d'interès i realitzar accions específiques. A més, el color complementari té una similitud amb el color blau de la Universitat de Barcelona, donant una connexió visual amb la institució.
- **#F4F4F5 Zinc-100.** Aquest color blanc amb una lleuger matís és ideal per al fons de la pantalla, ja que manté una aparença neta i ordenada sense generar una llum excessiva que pugui resultar incòmoda per als ulls.

## 4.2 Crides a la API

Per establir la connexió entre el frontend i el backend és essencial tenir en compte dues components clau: les crides del frontend i els endpoints del backend.

Les crides del frontend són les sol·licituds que el client (navegador web) fa al servidor backend per obtenir o enviar dades. Aquestes crides poden ser realitzades mitjançant diferents mètodes HTTP com GET, POST, PUT o DELETE, segons la naturalesa de la sol·licitud.

D'altra banda, els endpoints del backend són les adreces URL específiques que el servidor proporciona per gestionar les crides del frontend. Cada endpoint està associat a una funcionalitat o recurs específic de l'aplicació. Per exemple, podríem tenir un endpoint `/api/obtenirdades` que respongui amb les dades sol·licitades pel frontend.

En aquest apartat explicaré com ho he fet per a la meva aplicació, ja que he hagut d'adaptar les crides tant del frontend com del backend perquè acceptin fitxers.

### 4.2.1 Crides desde el Front

```
const create = e => {
  e.preventDefault()
  const formDataToSend = new FormData();
  formDataToSend.append('nom', formData.nom);
  formDataToSend.append('idiomaP', selectedcurs);
  formDataToSend.append('descripcio', formData.descripcio);
  formDataToSend.append('id_curs', localStorage.getItem('curs_id'))
  fileList.forEach(files => {
    formDataToSend.append('files', files);})
  axios
    .post('http://127.0.0.1:8000/cursos/practica', formDataToSend, {
      headers: {
        'Authorization': localStorage.getItem('token'),
        'Content-Type': 'multipart/form-data'
      }
    })
    .then((response) => {
      setOpenModal(false)
    })
    .catch((error) => {
      console.error('Error al enviar la sol·licitud:', error);
    });
}
```

L'extracte de codi que veiem, es tracta d'una crida de creació d'una pràctica. He agafat aquest, ja que és una de les crides que més coses significatives té. Podem veure que està dins d'una funció create, aquesta s'activa quan premem el botó de guardar vist a la **Fig 15**. Seguidament, s'instància un nou objecte FormData anomenat formDataToSend, que és utilitzat per construir dades de formulari i enviar-les amb XMLHttpRequest.

A continuació, es fan diverses crides al mètode append de formDataToSend per afegir diferents camps del formulari. Aquests camps són "nom", "idiomaP", "descripció", "id\_curs" i "files", tots els camps són de text o de números excepte els files, que aquests són una llista dels fitxers adjunts que pot tenir una pràctica. Per poder afegir els fitxers es fa una iteració sobre la llista de fitxers (filelist) amb un bucle forEach, i cada fitxer s'afegeix al formulari amb la clau "files".

A continuació, es fa una crida a la llibreria Axios amb el mètode POST, enviant les dades del formulari a una URL específica ('http://127.0.0.1:8000/cursos/practica'). S'inclouen headers com "Authorization" amb un token emmagatzemat a localStorage i 'Content-Type' com a 'multipart/form-data' per indicar que s'està enviant un formulari amb arxius adjunts.

Finalment, es tracta el resultat de la crida a la promesa. Si la crida és exitosa (.then), es tanca un modal amb setOpenModal(false). Si hi ha un error (.catch), es mostra un missatge d'error a la consola (console.error).

En general, les crides a les API solen fer-se utilitzant dades en format JSON. No obstant això, en aquest escenari específic, és necessari fer ús de l'objecte FormData. Aquesta elecció és obligatòria, ja que es tracta de l'enviament de fitxers des del frontend al backend.

## 4.2.2 Endpoints Backend

```
@router.post("/practica")
async def crear_practicas(user: auth,
                          nom: str = Form(...),
                          idiomaP: str = Form(...),
                          descripcio: str = Form(...),
                          files: Optional[list[UploadFile]] = File(None),
                          id_curs: str = Form(...),
                          db: Session = Depends(get_db),
                          mongo: Session = Depends(get_mongodb_client)):

    if user['is_alumno']:
        raise HTTPException(401, "Unauthorized")

    p_path = prof_path + "/" + curs.curs + "/" + curs.nom + "/" + nom
    a_path = almn_path + "/" + curs.curs + "/" + curs.nom + "/" + nom
```

```

if files != None:

    for file in files:
        directorio = p_path + "/" + file.filename
        with open(directorio, 'wb') as f:
            chunk_size = 1024 * 1024
            while True:
                chunk = file.file.read(chunk_size)
                if not chunk:
                    break
                f.write(chunk)

```

Aquest endpoint decorat amb `@router.post("/practica")` pertany a un fragment del codi que he desenvolupat, per explicar les parts importants.

La funció `crear_practicas` rep diversos paràmetres que provenen de les dades enviades pel client mitjançant una sol·licitud POST. Aquests paràmetres inclouen informació com el nom de la pràctica (`nom`), l'idioma (`idiomaP`), la descripció (`descripcio`), una llista de fitxers (`files`), l'identificador del curs (`id_curs`), i altres que es llegeixen de la sol·licitud, aquesta forma de llegir els paràmetres és únic per un `FormData`.

Abans de processar les dades, es realitza una comprovació de l'accés de l'usuari, aquesta es fa amb el token que es porta desde el frontend. Si l'usuari és un alumne, es genera una excepció HTTP amb codi d'estat 401 (`Unauthorized`), ja que un alumne no ha de ser capaç de crear una pràctica.

A continuació, es construeixen les rutes dels directoris on es guardarà la informació de la pràctica, tant per als professors com per als alumnes. Les rutes es generen utilitzant algunes variables com `prof_path`, `curs.curs`, `curs.nom`, i `nom`, el curs de l'assignatura i el seu nom venen d'una query que està en el codi, però que en aquest exemple està retallada.

Després, es verifica si hi ha fitxers per carregar. En cas afirmatiu, es procedeix amb la iteració sobre la llista de fitxers rebuts (`files`). Per a cada fitxer, es construeix la ruta completa i s'escriu el contingut del fitxer al sistema de fitxers local.

### 4.3 Sistema d'autenticació

En el sistema d'autenticació d'usuaris implementat, s'ha optat per utilitzar JSON Web Tokens (JWT) com a mètode de gestió de l'autenticació dels usuaris. El JWT és un estàndard obert (RFC 7519) que proporciona una manera compacta i segura de representar la informació entre dues parts.

En primer lloc, quan un usuari es registra o inicia sessió amb èxit al sistema, se li genera un JWT que conté informació rellevant com ara el seu niub i el rol d'usuari a través de booleans amb el "is\_alumno", "is\_profesor" o el "is\_admin". Aquest JWT se signa fent servir una clau secreta al servidor, garantint així la seva integritat.

Després de generar el JSON Web Token (JWT) al backend amb les dades d'autenticació de l'usuari, aquest és enviat al frontend, on es gestiona per mantenir-lo. En aquest cas fem servir el localStorage que ens proporciona React, que no deixa de ser un diccionari de claus i valors per a cada usuari.

En les crides futures que el frontend realitzi al backend, el JWT es vincula com a part de la sol·licitud mitjançant l'encapçalament d'autorització (Bearer Token). Aquesta pràctica permet al backend autenticar l'usuari i determinar els seus permisos d'acord amb la informació continguda al JWT.

Els endpoints del backend que requereixen autorització poden fer servir el JWT per comprovar els rols de l'usuari. Si un usuari intenta accedir a un endpoint sense els rols adequats, el backend pot rebutjar la sol·licitud amb un codi d'estat HTTP 401 (Unauthorized) o 403 (Forbidden), assegurant-se que només els usuaris amb els permisos correctes puguin utilitzar aquests recursos.

## 4.4 Mètode de connexió amb la eina de correcció

```
await rpc_client.connect()
result = await rpc_client.call("java_checks",
                               curs.nom,
                               curs.curs,
                               practica.nom,
                               user['niub'],
                               a_path,
                               p_path)

string = result.decode("utf-8")
resposta = json.loads(string)
mongo.correccion(user['niub'], curs.id, practica.id, resposta)
```

L'extracte de codi que es presenta correspon a la gestió dels missatges amb RabbitMQ en el context de l'endpoint encarregat de pujar una tasca. Analitzarem les característiques clau d'aquest fragment de codi:

En primer lloc, cal destacar que l'endpoint espera establir una connexió amb el client RPC abans de continuar. Aquesta connexió és crítica, ja que s'inicia en un fil d'execució diferent de la resta de l'aplicació. Aquesta pràctica evita que la resta de l'aplicació es bloquegi mentre s'espera la resposta de la connexió RPC, assegurant una execució fluida.

Un cop establerta la connexió, l'endpoint procedeix a enviar un missatge a través de RabbitMQ i espera la resposta de l'eina remota. És rellevant subratllar que aquesta operació també es realitza en un fil d'execució separat, evitant possibles bloquejos durant l'espera de la resposta.

En arribar el missatge de resposta, aquest es presenta inicialment com una cadena de text (string) i, per tant, és necessari dur a terme una descodificació per interpretar correctament el contingut. A més, cal tenir en compte que la resposta de l'eina remota es proporciona en format JSON, i per això és necessari convertir-la a un objecte de tipus JSON per a un processament adequat.

Finalment, després de fer la descodificació i la conversió de la resposta a un objecte JSON, aquesta informació s'emmagatzema a la base de dades MongoDB.

## 4.5 Instal·lació al servidor

Perquè tot el treball fet tant al frontend com al backend funcioni completament, és molt important establir la capacitat d'enviar i rebre missatges a través de RabbitMQ. L'inconvenient és el fet que l'eina de correcció automàtica només opera en un entorn local, mentre que el servidor encarregat de gestionar els missatges de Rabbit per executar les correccions automàtiques ha de funcionar exclusivament en el servidor de la universitat. Per abordar aquesta qüestió, vam prendre la decisió de transferir tot el codi que tenia al meu portàtil i instal·lar-lo en el servidor. Encara que no fos difícil, vam haver de fer diverses coses perquè tot el codi funcionés correctament. En aquest apartat detallaré tots els passos que vam seguir perquè en l'actualitat tot estigui funcionant correctament.

### 4.5.1 Primers passos

El primer que vam fer va ser instal·lar tant el frontend com el backend (sense encara connectar-ho a la Rabbit). Vam crear un directori anomenat “software” que es dedicaria a emmagatzemar el directori Git anomenat “TFG”. Aquí vam enfrontar-nos al primer problema, ja que el que volíem era que la sessió de GitHub es mantingués oberta permanentment per poder transportar els canvis quan fos necessari. Això va requerir la creació d'una clau SSH per a l'usuari i, a través del compte de GitHub, vam vincular les credencials.

Vam guardar el codi en local i, seguidament, vam executar el fitxer `requirements.txt` per a la instal·lació automàtica de totes les llibreries de Python que s'utilitzen. També ho vam fer per al frontend a través del `npm install`. En aquest moment, vam intentar executar el codi. El frontend funcionava correctament, mostrava el formulari de login i tot semblava estar correcte, però l'api no havia aconseguit executar-se correctament.

El problema era que havíem oblidat d'instal·lar el PostgreSQL localment i, per tant, no reconeixia l'URL de creació de la base de dades. Vam descarregar-nos Postgres, vam crear un usuari amb contrasenya i vam crear la base de dades. Després d'això tot va funcionar correctament.

### 4.5.2 Prova end-to-end Ping

Per facilitar aquest procés de les proves end-to-end, vaig proposar l'adaptació del codi del servidor per incloure una funció de ping. Això ens permetria verificar la connexió amb el servidor abans de provar tot el codi relacionat amb les correccions automàtiques.

La prova va ser un èxit. Aleshores, vam passar a la següent tasca: establir una connexió remota amb el servidor. Inicialment, vam utilitzar SSH per a la connexió remota, però aquest mètode presenta problemes: estant connectat a través de SSH, es pot iniciar el backend, però no es té la capacitat de posar en marxa el frontend, ja que no és pot



obrir un navegador per visualitzar-ho. Per solucionar aquest problema vaig pensar en el TeamViewer.

TeamViewer **Fig 19** és una aplicació que permet l'accés remot a ordinadors i altres dispositius. Amb TeamViewer, pots connectar-te a un dispositiu remot des de qualsevol lloc amb connexió a Internet. Aquesta eina és àmpliament feta servir per oferir assistència remota, dur a terme reunions virtuals, compartir arxius i, fins i tot, accedir al teu propi ordinador des de lluny. El principal avantatge de TeamViewer és que proporciona la capacitat de visualitzar la sortida per pantalla generada pel dispositiu remot, facilitant així una interacció completa amb el sistema a distància.

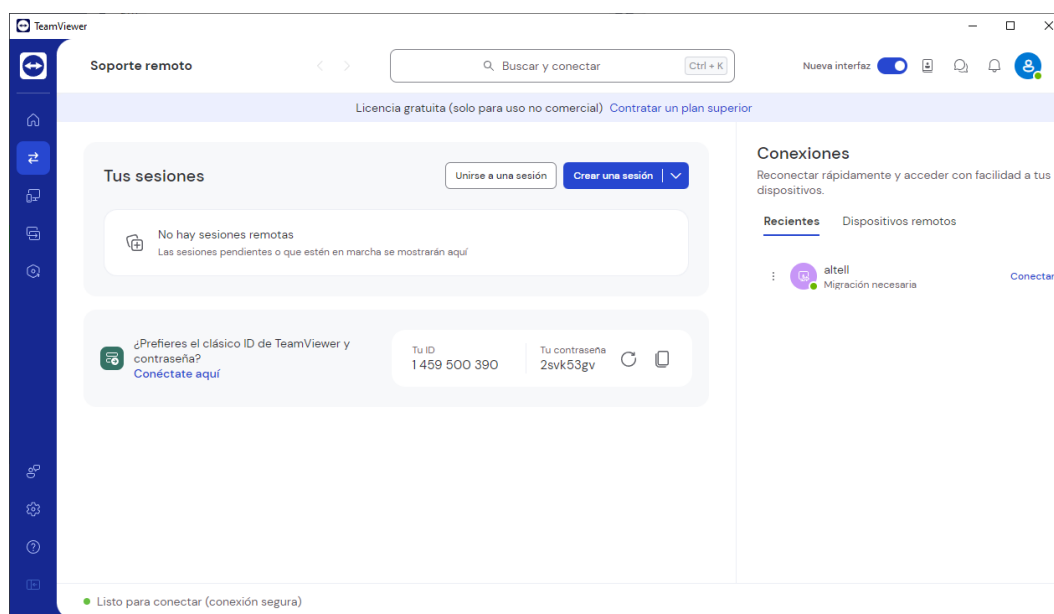


Figura 19: TeamViewer

Vaig instal·lar TeamViewer per a servidors, al servidor de la universitat, ja que no es necessiten funcionalitats més avançades que les que permeten compartir la pantalla del servidor. Això ens va permetre utilitzar una versió adequada a les nostres necessitats. Després, vaig descarregar TeamViewer al meu portàtil personal, vaig crear un compte d'usuari i vaig intentar connectar-me al servidor.

Inicialment, vaig tenir problemes, ja que havia creat el compte de TeamViewer mitjançant el meu compte de Google, i aquesta opció no estava suportada pel TeamViewer del servidor. Per solucionar això, vaig crear un compte d'usuari normal. Posteriorment, vaig fer servir TeamViewer per a crear una sessió i em va donar un codi que vaig utilitzar per connectar-me al servidor amb èxit. A través del meu ordinador, podia veure tot el que succeïa al servidor i tenia la capacitat d'editar fitxers, executar comandes de Bash i tenir control total sobre l'ordinador.

Vaig fer una prova tancant la sessió de TeamViewer i tornant-la a obrir, i semblava que

tot funcionava correctament. Així que vam decidir deixar-ho de moment i continuar fent canvis al servidor des del meu ordinador a través de TeamViewer.

### 4.5.3 Problemes de connexió amb TeamViewer

Més endavant, vaig intentar connectar-me novament al servidor a través del TeamViewer. No obstant això, en intentar-ho, em deia que el servidor estava apagat. Vaig pensar que podia ser a causa d'un tall de llum planificat, com m'havia informat un dels tutors. Malgrat això, vaig decidir verificar-ho connectant-me a través de SSH al servidor. Sorprenentment, el servidor estava en línia i en funcionament. Així que vaig començar a investigar i vaig trobar els possibles errors que podíem haver comès:

1. Vam tancar la sessió de l'usuari de Linux sense deixar una opció marcada al TeamViewer del servidor, que manté la connexió oberta sempre.
2. Vam desconnectar la pantalla del servidor i no vam configurar-ho perquè el mateix TeamViewer generés la sortida de pantalla de manera virtual. TeamViewer comparteix la sortida per pantalla i, si no hi ha una pantalla connectada, no pot compartir res i, per tant, no es pot connectar, per això s'ha de configurar que es pugui generar una sortida per pantalla virtual.

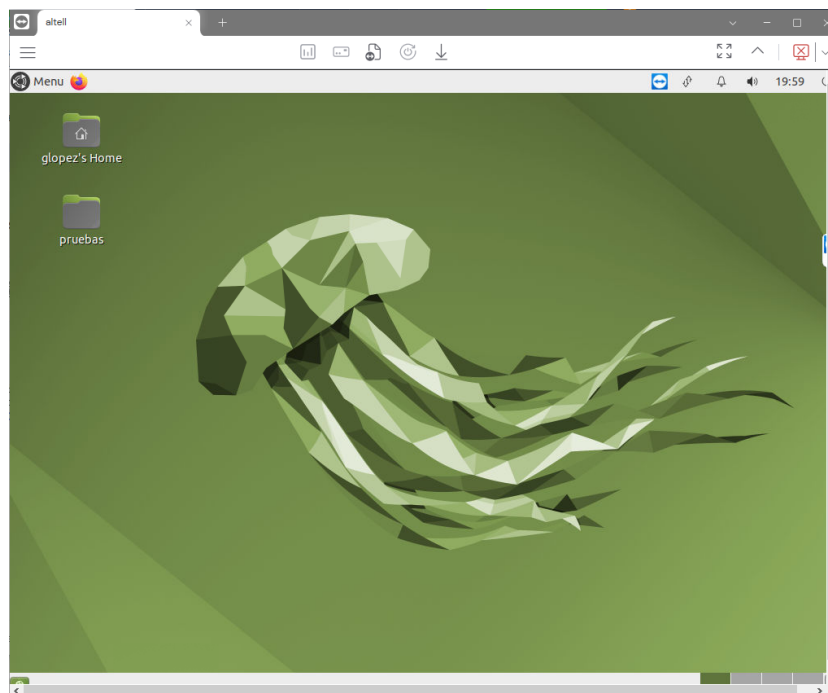


Figura 20: Pantalla del Servidor

Per resoldre aquests dos problemes, vaig procedir a reinstal·lar TeamViewer al servidor. Vaig activar la funció que permet que TeamViewer romangui actiu entre sessions d'usuaris de Linux. A més, vaig mantenir la pantalla connectada, però apagada; d'aquesta manera, no es mostra a la pantalla del servidor, però la GPU reconeix la presència de la pantalla i genera una imatge que TeamViewer pot capturar amb èxit. **Fig 20.**

## 4.6 Resultats

L'últim que hem fet, ha estat fer la prova final d'un fluxe complet d'operacions. El fluxe a seguir ha estat:

1. Crear un professor nou
2. Crear un alumne nou
3. Crear un curs nou afegint a l'alumne
4. Crear una pràctica amb els fitxers de correcció
5. Entrar amb l'alumne nou
6. Pujar una pràctica

Al principi no va funcionar del tot bé, ja que la correcció automàtica no tornava res. El que vam pensar era que podien ser privilegis de l'usuari de Linux del tutor i la impossibilitat de llegir o escriure del meu usuari. La primera opció que vam provar va ser donar permisos d'escriptura i lectura a tots els usuaris. Aquesta solució va funcionar, però pensàvem que era un perill de seguretat.

Així que el que vam fer va ser crear un grup d'usuaris amb permisos conjunts, agrupant l'usuari del tutor i el meu i, amb això, es va solucionar el problema de seguretat i vam comprovar que tot funcionava correctament.

El codi es troba en aquest repositori de github : <https://github.com/GerardLopezRuiz/TFG>. Encara que es poden provar la majoria de les funcionalitats de l'aplicació amb uns petits canvis, no es pot provar l'eina de correccions automàtiques perquè, com està explicat al document, només està instal·lada al servidor de la universitat.

## 5 Conclusions i treball futur

L'objectiu principal d'aquest treball ha estat utilitzar tots els coneixements que he anat adquirint a la carrera, per a fer un treball a l'alçada que s'espera d'un alumne d'últim curs.

He aconseguit aprendre a utilitzar React i en conjunt amb Tailwind, crear un frontend senzill i agradable a la vista. He pogut establir connexions des d'un frontend a un backend, així com implementar un sistema de directoris ordenats per emmagatzemar els fitxers carregats pels usuaris. També he aconseguit connectar dos serveis per a que s'enviïn missatges seguint un protocol, a més d'aconseguir que la producció i consumició dels missatges es produeixin de forma concurrent a l'aplicació.

Si bé tot això compleix amb el que pot ser un Treball de Fi de Grau complet, hi ha coses que s'havien plantejat inicialment i no s'han arribat a implementar o idees que han sorgit després d'establir el "backlog" i que m'agradaria esmentar:

1. Fer que les pràctiques creades tinguin una data de finalització, que a partir d'aquella data no es puguin pujar més fitxers i que la correcció automàtica comenci en aquell instant.
2. Fer que el professor pugui iniciar la correcció automàtica de totes les pràctiques en qualsevol moment.
3. Veure en un curs la llista d'usuaris que estan assignats i fer una distinció entre professors i alumnes, a més de poder editar aquella llista, eliminant o afegint nous alumnes o professors.
4. Mostrar la correcció completa que dona l'eina de correcció.
5. Fer una pàgina de qualificacions perquè els alumnes puguin veure com van globalment.
6. Fer el calendari funcional i que es mostrin les pràctiques en les dates de quan acaben.
7. Que els professors i els administradors puguin editar tant els cursos com les pràctiques.
8. Crear uns nous subdirectoris a l'estructura ja creada, per a poder rebre els fitxers de correcció que crea l'eina.
9. Tant per els alumnes com els professors, que siguin capaços de saber quins fitxers han pujat i si els volen eliminar del servidor.
10. Dockeritzar tot el projecte, per a no tenir problemes a l'hora de posar en marxa el projecte.

11. Portar el projecte a producció i que es puguin fer proves reals amb alumnes i professors.

Per concloure tot vull recalcar que m'ho he passat molt bé desenvolupant aquest projecte, ha estat un repte apassionant i crec que té molt futur, ja no per la universitat sinó per tots els àmbits de l'educació, ja que crec que treure càrrega de correcció als professors implicaria que podrien dedicar encara més hores a preparar les classes i inventant noves formes de fer l'educació divertida i enriquidora. També espero poder seguir treballant en el projecte i poder veure'l finalitzat .

## 6 Bibliografía

- [1] Aio-Pika. Aio-pika. <https://aio-pika.readthedocs.io/en/latest/>.
- [2] FastAPI. Fastapi. <https://fastapi.tiangolo.com>.
- [3] MDNWebDocs. Css. <https://developer.mozilla.org/es/docs/Web/CSS>.
- [4] MDNWebDocs. Html. <https://developer.mozilla.org/es/docs/Web/HTML>.
- [5] MDNWebDocs. Javascript. <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- [6] MongoDB. Mongoddb. <https://www.mongodb.com/es>.
- [7] Pika. Pika. <https://pika.readthedocs.io/en/stable/>.
- [8] PostgreSQL. Postgresql. <https://www.postgresql.org>.
- [9] Python. Python. <https://www.python.org>.
- [10] RabbitMQ. Rabbitmq. <https://rabbitmq.com>.
- [11] React. React. <https://es.react.dev>.
- [12] SQLAlchemy. Sqlalchemy. <https://www.sqlalchemy.org>.
- [13] Tailwind. Tailwind. <https://tailwindcss.com>.

## 7 Llicències

### 7.1 Flowbite

Copyright (c) Themesberg (Bergside Inc.)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 7.2 Heroicons

MIT License

Copyright (c) 2020 Refactoring UI Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN



AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 8 Annexos

### 8.1 Enpoints Usuari

Endpoint	Descripció	Parametres i Cos	Mètode
api/users/	Creació d'un usuari nou	Dades del usuari	POST
api/users/login	Comprovació de dades per l'inici de sessió	Dades del usuari	POST
api/users/profesor	Donar rols de professor a un NIUB	Dades del professor	POST

### 8.2 Enpoints Cursos

Endpoint	Descripció	Parametres i Cos	Mètode
api/cursos/	Llista de tots els cursos d'un usuari	-	GET
api/cursos/	Creació d'un nou curs	Dades del nou curs	POST
api/cursos/practica	Creació d'una nova pràctica	Dades de la nova pràctica	POST
api/cursos/csv	Obté plantilla en csv	-	GET
api/cursos/excel	Obté la plantilla en excel	-	GET
api/cursos/curs/practicas/id	Obté les pràctiques d'un curs	id del curs	GET
api/cursos/curs/practicas/id	Obté les pràctiques d'un curs	id del curs	GET
api/cursos/curs/id	Obté la informació d'un curs	id del curs	GET
api/cursos/practica/id	Obté la informació d'una pràctica	id de la pràctica	GET

api/cursos/upload	Guarda els fitxers penjats per els usuaris	Dades de la pràctica i fitxer adjunt	GET
-------------------	--	--------------------------------------	-----