



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

**Studia: Una aplicació web de e-learning
basada en l'aprenentatge autoregulat**

Marcos Palomino Vivar

Director: Eloi Puertas Prats
Realitzat a: Departament de
Matemàtiques i Informàtica
Barcelona, 13 de juliol de 2023

Resumen

La educación tal como la conocemos a día de hoy, no ha sufrido grandes cambios si la comparamos como era hace 100 años. Sin embargo, en estas últimas décadas, debido a la reciente innovación tecnológica ha tenido que ir adaptándose para satisfacer las nuevas necesidades de los usuarios.

A pesar de ello, el modelo utilizado por los profesores no ha cambiado, y se sigue aplicando un esquema tradicional donde lo más importante es el temario y el profesor no está demasiado preocupado por si el estudiante desarrolla otras capacidades y habilidades cognitivas, meta-cognitivas y emocionales vinculadas al aprendizaje autorregulado.

Éste concepto no es un término genérico ni un concepto general, sino un constructo bastante sofisticado que se ha establecido en el campo de la investigación educativa, donde el objetivo principal es desarrollar habilidades que permitan a los estudiantes convertirse en aprendices independientes y eficientes a lo largo de su vida. Para lograr esto, los estudiantes deben establecer metas de aprendizaje, planificar y organizar su tiempo, utilizar estrategias de aprendizaje efectivas, monitorear su progreso, regular su motivación y evaluar su propio aprendizaje.

Studia, nace de la idea de querer implementar de una manera sencilla este aprendizaje autorregulado a los alumnos universitarios, intentando mejorar el actual sistema moodle y tratando de innovar el sistema educativo que se imparte, para ello, en este documento explicaré el desarrollo de una aplicación web en colaboración con la facultad de educación, basada en el estudio encontrado en el artículo “How to Embed SRL in Online Learning Settings? Design Through Learning Analytics and Personalized Learning Design in Moodle” [19] y en el proyecto “Plataforma E-Learning para la Autorregulación del Aprendizaje” [22] realizado por Elena Cano García que ha estado presente en todo el desarrollo de este proyecto.

Para desarrollar esta innovadora aplicación he utilizado tecnologías modernas y con alta demanda en el mercado, he utilizado Django para implementar una REST API en la parte del back-end, React para la parte del front-end, y finalmente Tailwind como framework de CSS, podemos encontrar el código del proyecto en el siguiente repositorio de github

<https://github.com/Topiksit0/Studia>

Keywords: Learning Analytics, self-regulated learning, django, react, web development, web app

Abstract

Education as we know it today has not undergone significant changes compared to what it was like 100 years ago. However, in recent decades, due to recent technological innovation, it has had to adapt to meet the new needs of users.

Nevertheless, the model used by teachers has not changed, and a traditional scheme is still applied where the subjects of the course are the most important, and the teacher is not too concerned about whether the student develops other cognitive, metacognitive, and emotional abilities linked to self-regulated learning.

This concept is not a generic or general term but a rather sophisticated construct that has been established in the field of educational research, with the main objective being to develop skills that enable students to become independent and efficient learners throughout their lives. To achieve this, students must set learning goals, plan and organize their time, use effective learning strategies, monitor their progress, regulate their motivation, and evaluate their own learning.

Studia was born from the idea of wanting to implement this self-regulated learning to university students in a simple way, attempting to improve the current Moodle system and innovate the educational system being taught. Therefore, in this document, I will explain the development of a web application in collaboration with the Faculty of Education, based on the study found in the article "How to Embed SRL in Online Learning Settings? Design Through Learning Analytics and Personalized Learning Design in Moodle" [\[19\]](#) and the project "E-Learning Platform for Self-regulated Learning" [\[22\]](#) carried out by Elena Cano García, which has been present throughout the development of this project.

To develop this innovative application, I have used modern technologies in high demand in the market. I have used Django to implement a REST API on the back-end, React for the front-end, and finally, Tailwind as the CSS framework, you can find the code on this github repository <https://github.com/Topiksit0/Studia>

Keywords: Learning Analytics, self-regulated learning, django, react, web development, web app

Resum

L'educació tal com la coneixem avui en dia no ha sofert grans canvis si la comparem amb com era fa 100 anys. No obstant això, en aquestes últimes dècades, a causa de la recent innovació tecnològica, s'ha hagut d'adaptar per satisfer les noves necessitats dels usuaris.

Tot i això, el model utilitzat pels professors no ha canviat i s'aplica encara un esquema tradicional on el més important és el temari i el professor no està massa preocupat si l'estudiant desenvolupa altres capacitats i habilitats cognitives, metacognitives i emocionals vinculades a l'aprenentatge autoregulat.

Aquest concepte no és un terme genèric ni un concepte general, sinó un constructe bastant sofisticat que s'ha establert en el camp de la investigació educativa, on l'objectiu principal és desenvolupar habilitats que permetin als estudiants convertir-se en aprenents independents i eficients al llarg de la vida. Per aconseguir-ho, els estudiants han d'establir metes d'aprenentatge, planificar i organitzar el seu temps, utilitzar estratègies d'aprenentatge efectives, monitoritzar el seu progrés, regular la seva motivació i avaluar el seu propi aprenentatge.

Studia neix de la idea de voler implementar d'una manera senzilla aquest aprenentatge autoregulat als alumnes universitaris, intentant millorar l'actual sistema Moodle i tractant d'innovar el sistema educatiu que s'imparteix. Per això, en aquest document explicaré el desenvolupament d'una aplicació web en col·laboració amb la facultat d'educació, basada en l'estudi trobat a l'article "How to Embed SRL in Online Learning Settings? Design Through Learning Analytics and Personalized Learning Design in Moodle" [19] i en el projecte "Plataforma E-Learning per a l'Autoregulació de l'Aprenentatge" [22] realitzat per Elena Cano García, que ha estat present en tot el desenvolupament d'aquest projecte.

Per desenvolupar aquesta innovadora aplicació, he utilitzat tecnologies modernes i amb alta demanda en el mercat. He utilitzat Django per implementar una REST API a la part del backend, React per a la part del frontend i, finalment, Tailwind com a framework de CSS, podem trobar el codi del projecte en el següent repositori de github <https://github.com/Topiksit0/Studia>.

Keywords: Learning Analytics, self-regulated learning, django, react, web development, web app

1. Introducció	7
1.1. Motivació	7
1.2. Context	8
1.3. Objectius	9
1.3.1 Objectius principals	9
1.3.2 Objectius específics	9
1.4. Estructura del document	10
2. State of art	11
3. Anàlisi	13
3.1. Requeriments	13
3.1.1. Requeriments funcionals	13
3.1.2. Requeriments no funcionals	14
3.2. Backlog	14
3.3 Diagrama de Gantt	21
4. Disseny	22
4.1. Arquitectura de l'aplicació	22
4.2. Tecnologies utilitzades	30
4.2.1. Llenguatges de programació (i altres)	30
4.2.1.1. HTML	30
4.2.1.2. CSS	30
4.2.1.3. Javascript	30
4.2.1.4. Python	31
4.2.2. Frameworks	31
4.2.2.1. React	32
4.2.2.2. Redux	33
4.2.2.3. Django	34
4.2.2.4. Djoser	35
4.2.2.5. Tailwind	35
4.2.3 Plataformes	35
4.2.3.1. Github	35
4.2.3.2. Figma	37
4.3. Wireframes	37
4.4. Prototips	39
5. Implementació i resultats	45
5.2. Serveis externs	46
5.2.1. OpenAI API	46
5.2.2. Google Forms API	47
5.3.1 Docker	49
5.3.2 Docker Compose	51
6. Conclusions i treball futur	55
7. Bibliografia	56
8. Annexos	58

1. Introducció

Actualment, l'aprenentatge autoregulat és considerat una habilitat clau per a l'èxit acadèmic i professional. Aquesta habilitat permet als estudiants adquirir i aplicar nous coneixements de manera autònoma, la qual cosa els permet desenvolupar la seva capacitat d'aprenentatge i adaptar-se millor a les exigències d'un món en constant canvi.

En aquest context, sorgeix la necessitat de desenvolupar eines i recursos que permetin als estudiants practicar i millorar la seva capacitat d'autoaprenentatge de manera efectiva. Aquest procés d'aprenentatge és un camp molt estudiat en la branca educativa, cada vegada major nombre d'investigacions subratllen la importància que té de cara a aconseguir que els estudiants siguin aprenents autònoms i d'èxit, el fet que siguin capaços de regular el seu propi procés d'aprenentatge, així com la forta relació existent entre l'aprenentatge autoregulat que inclou tres components: (1) les estratègies metacognitives; (2) les estratègies cognitives; i (3) la direcció i control de l'esforç. [\[20\]](#).

Basat en els diferents estudis previs, s'ha desenvolupat una aplicació WEB d'e-learning enfocada en l'autoaprenentatge, anomenada Studia.

Studia és una aplicació innovadora que té com a objectiu proporcionar als estudiants universitaris una plataforma en la qual puguin practicar i millorar les seves habilitats d'aprenentatge a l'hora de realitzar un curs d'estudis. Aquesta aplicació web s'ha desenvolupat en conjunt amb la facultat d'educació de la UB, que en tot moment ens han donat feedback i han supervisat el projecte per validar que aquest procés educatiu es trobi present tant per l'estudiant com pel professor.

El projecte comença amb la recerca i anàlisi d'aplicacions de e-learning que tinguin un objectiu similar a la nostra, amb els nostres requisits clars fem una primera iteració del backlog complet i continuem amb una proposta pel que fa al disseny intern incloent un estudi de l'arquitectura del sistema, esquemes de bases de dades, arquitectura del backend i també un disseny del front començant pels wireframes del sistema i acabant amb uns prototips d'alta fidelitat. Un cop implementades aquestes funcionalitats, farem un deployment amb Docker per poder desplegar l'aplicació en qualsevol servei de Cloud.

1.1. Motivació

Aquest projecte neix de diverses motivacions personals i acadèmiques. En primer lloc, sempre m'ha apassionat el desenvolupament web, especialment el frontend. Per aquesta raó tenia clar que volia emprendre un projecte d'aquesta tema, amb aquesta oportunitat que em van oferir no tenia cap dubte que podría sortir un projecte molt interessant.

A més, aquest treball em permetria aplicar els coneixements adquirits en la meva formació universitària i en diversos cursos que he realitzat en un projecte pràctic i real, crec que resulta en una

pràctica interessant per consolidar les meves habilitats tècniques, així com aprendre noves tecnologies i eines.

Una altra motivació important és la creixent demanda d'eines i plataformes d'e-learning, especialment en el context actual de l'educació a distància, el projecte m'ofereix l'oportunitat de contribuir en aquest àmbit, oferint una eina útil i accessible intentant millorar les alternatives que trobem actualment en el mercat. És un projecte interessant també perquè no és una plataforma de e-learning més, sinó que és un projecte amb un fonament i un estudi darrere, amb el suport d'un grup educatiu expert en el tema.

Finalment, em motiva el repte d'abastar tot el cicle de vida del projecte, des de la definició de les user stories fins al deployment del producte final en una plataforma de producció, tot això em permet adquirir una visió global i completa del procés de desenvolupament d'un projecte de software.

1.2. Context

El context d'aquest treball s'emmarca en la creixent necessitat d'implementar noves eines tecnològiques en el camp de l'educació. La pandèmia del COVID-19 ha portat a una acceleració en l'adopció de tecnologies d'aprenentatge a distància i e-learning, la qual cosa ha posat en relleu la importància de comptar amb eines eficaces per a l'autoaprenentatge i l'avaluació entre parells.

Donant context en l'autoregulació, aquest aprenentatge el podem dividir en tres fases [\[21\]](#). Previsió, Execució i Autoreflexió. Aquests passos són seqüencials, de manera que l'aprenent autoregulat segueix aquestes fases en l'ordre esmentat quan aprèn alguna cosa.

- **Fase de Previsió:** En aquesta etapa, l'aprenent autoregulat es prepara per a l'aprenentatge. Analitza la tasca que abordarà i estableix metes d'aprenentatge. A més, desenvolupa una planificació estratègica per aconseguir aquestes metes i cultiva creences d'automotivació respecte a la seva capacitat i els resultats esperats.
- **Fase d'Execució:** En aquesta fase, l'aprenent posa en pràctica l'aprenentatge. Utilitza estratègies d'autocontrol per gestionar el seu propi procés d'aprenentatge. Això implica aplicar les estratègies planificades durant la fase de previsió. També és important que l'aprenent s'observi i es monitori a si mateix, mitjançant l'autoobservació o l'automonitorització, per avaluar el seu progrés i ajustar les estratègies si és necessari.
- **Fase d'Autoreflexió:** En aquesta última etapa, l'aprenent reflexiona sobre el seu procés d'aprenentatge. Realitza una autoavaluació, tant cognitiva com afectiva. En l'avaluació cognitiva, analitza l'èxit o fracàs de l'aprenentatge, identificant les causes que l'han influït. A escala efectiva, avalua el seu nivell de satisfacció personal. L'autoreflexió permet a l'aprenent determinar si ha assolit les seves metes d'aprenentatge i obtenir retroalimentació sobre el seu rendiment i experiència d'aprenentatge.

L'aplicació està dirigida a estudiants i cerca oferir una experiència d'aprenentatge més flexible i interactiva, que els permet avançar al seu propi ritme i aconseguir una retroalimentació constant tant del professorat com dels seus mateixos companys seguint la seqüència de fases mencionada anteriorment.

1.3. Objectius

Durant el desenvolupament d'aquest projecte, ens adherim a uns objectius, en aquesta secció parlarem tant de l'objectiu principal com dels específics.

1.3.1 Objectius principals

L'objectiu principal d'aquest projecte és crear una plataforma nova pels estudiants d'un curs on puguin fer un procés educatiu mitjançant l'autoregulació gràcies a una contínua resposta del professorat. A més a més, la pàgina ha d'estar composta per un disseny atractiu per l'estudiant i ha d'incloure una sèrie d'eines per facilitar l'aprenentatge.

1.3.2 Objectius específics

També identifiquem els objectius específics relacionats amb el coneixement que he anat adquirint al llarg de la carrera:

- Utilitzar un framework de front-end modern com React per desenvolupar la interfície d'usuari de la plataforma, proporcionant una experiència interactiva i àgil, amb l'ajuda d'un framework de CSS com Tailwind.
- Dissenyar i implementar una REST API amb Django que permeti la comunicació entre la plataforma d'e-learning i altres serveis externs:
 - Definir els endpoints i les operacions necessàries per a l'intercanvi de dades rellevants per a l'autoregulació de l'aprenentatge.
 - Assegurar una connexió segura i eficient entre la plataforma i altres serveis a través de la REST API.
- Implementar un sistema d'autenticació mitjançant React Redux i Djoser que permet als usuaris administrar els seus propis comptes.
- Disseny d'una arquitectura robusta i escalable amb la integració de diverses bases de dades, com MongoDB o PostgreSQL.
- Disseny de wireframes, prototips de mitja fidelitat i d'alta fidelitat mitjançant Figma.

- Identificar els requeriments dels usuaris i construir un backlog adaptant-se en tot moment a les necessitats d'aquests.

1.4. Estructura del document

El contingut que trobem en aquesta memòria està format pels següents ítems:

- **State of art:** Identificació de les plataformes més competents que trobem en l'àmbit del e-learning, analitzant els seus punts forts.
- **Anàlisis:** Recerca dels requeriments funcionals i no funcionals que necessita la plataforma i desglossament del backlog.
- **Disseny:** Arquitectura de l'aplicació i explicació de les tecnologies utilitzades, bases de dades , deployment de l'aplicació i finalment es mostraran tant els primers wireframes de l'aplicació com els prototips finals.
- **Implementació i resultats:** Funcionalitats internes en l'àmbit tècnic, característiques i lògica de l'aplicació, a més a més, s'inclourà els resultats finals obtinguts de la pàgina.
- **Conclusions i treball futur:** Conclusions del treball amb funcionalitats extra que es poden implementar en un futur.
- **Referències:** Referències de l'origen de la informació utilitzada per desenvolupar el projecte.
- **Annexos:** Documentació de l'aplicació i resultats finals.

2. State of art

En el camp de l'e-learning, existeixen diverses plataformes que ofereixen solucions d'aprenentatge en línia per a estudiants i professors. Una de les plataformes més conegudes és Moodle LMS4, la qual proporciona eines per a la gestió de cursos en línia i la creació de continguts educatius. Encara que Moodle és una plataforma molt popular i utilitzada en molts centres educatius, incloent el nostre, la seva interfície pot resultar confusa i poc intuïtiva per a alguns usuaris, actualment, amb la recent arribada de la seva versió 4.0, han estat treballant molt en aquesta part tant d'accessibilitat i experiència d'usuari, però en la seva essència encara és el mateix com podem observar en la **Fig. 1**.

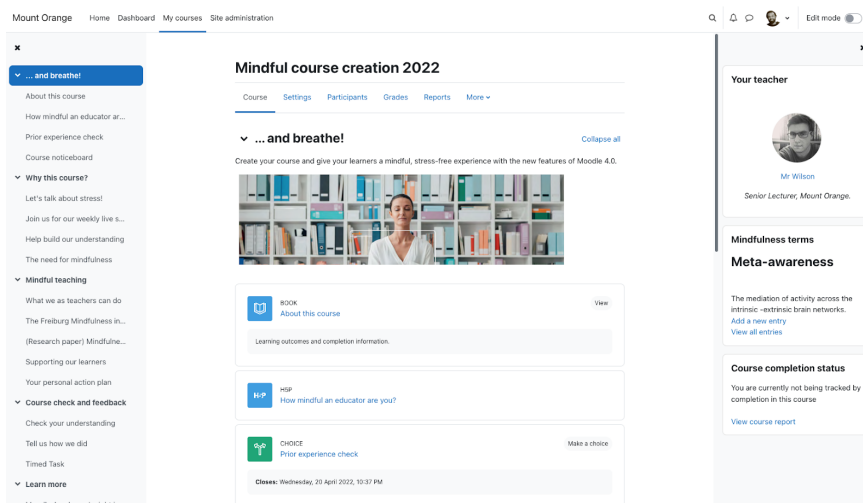


Fig. 1. Home page del nou Moodle 4.0.

Una altra plataforma d'e-learning destacada és Udemy, que se centra en la creació i venda de cursos en línia per part dels mateixos instructors. Si bé Udemy és molt popular entre els estudiants que busquen aprendre noves habilitats i coneixements (especialment els relacionats amb àmbits de la programació), sovint els cursos disponibles no estan organitzats en un pla d'estudis coherent i no compten amb la tutorització necessària. En la **Fig. 2**, podem observar la home page que trobem en Udemy a l'intentar accedir a un curs, no segueix una estructura similar a Moodle ja que tampoc estan enfocades al mateix target.

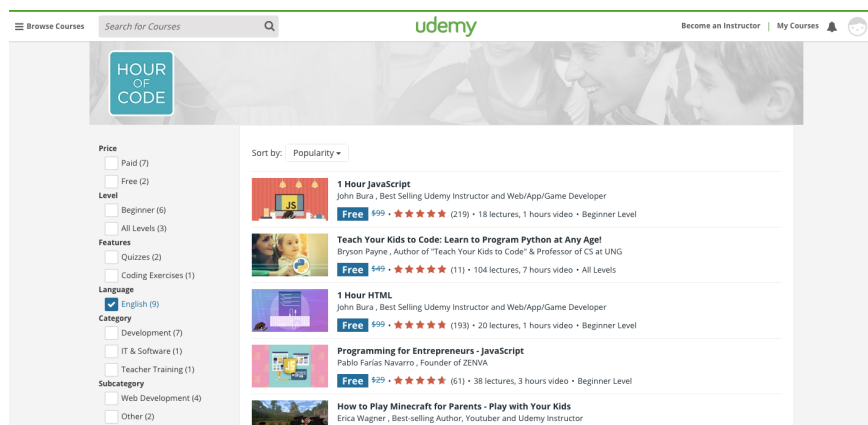


Fig. 2. Home page d'un curs de Udemy.

Finalment, una plataforma d'e-learning que ha guanyat molta popularitat en els últims anys és Coursera, la qual se centra a oferir cursos en línia d'universitats de tot el món. A diferència de Moodle i Udemy, els cursos de Coursera solen estar més estructurats i compten amb una tutorització més completa. No obstant això, la seva oferta de cursos està més limitada a les universitats i sovint requereixen un pagament per a obtenir certificació.

En comparació amb aquestes plataformes, Studia té com a objectiu oferir una interfície intuïtiva i fàcil d'usar perquè els estudiants puguin centrar-se en l'aprenentatge i els professors puguin gestionar el contingut de manera eficient. A més a més, la nostra plataforma té com a eina principal l'aprenentatge autoregulat sent el principal atractiu de la plataforma, ja que comptem amb un equip de professionals en l'àrea de l'educació que ens han pogut guiar en com seria una plataforma amb un mètode autoreglatiu centrat en l'aprenentatge de l'estudiant.

3. Anàlisi

El disseny d'aquesta aplicació implica tenir en compte tant les necessitats dels usuaris com els principis teòrics que recolzen la metodologia d'aprenentatge. En proporcionar una interfície intuïtiva i funcionalitats específiques per a fomentar l'autoregulació, es busca promoure la independència i la responsabilitat dels estudiants en el seu procés d'aprenentatge.

3.1. Requeriments

Per a aconseguir una aplicació eficient i efectiva, és crucial establir requisits clars i precisos que guiïn el desenvolupament i la implementació de la plataforma, en aquest apartat, dividirem els requisits en no funcionals i funcionals.

3.1.1. Requeriments funcionals

Els requisits funcionals són les funcionalitats i accions específiques que una aplicació o sistema ha de tenir per complir amb les necessitats i expectatives dels usuaris. Són aquells que indiquen què pot fer el sistema i quines interaccions permet als usuaris realitzar. En el cas de Studia, podem analitzar els següents requisits:

- **Registre d'usuaris:** Els usuaris han de poder registrar-se a la plataforma i fer log-in.
- **Perfils d'usuari:** Els usuaris han de poder veure el seu perfil.
- **Visualitzar cursos:** Els estudiants han de poder visualitzar el contingut d'un curs del seu principi a fi.
- **Chatbot:** Els estudiants han de poder conversar amb una intel·ligència artificial estil chatGPT adaptada a les dades del nostre servei.
- **Peer-review:** Els estudiants han de poder utilitzar una eina de peer-review per poder aplicar l'avaluació entre iguals en cas que el professor o convingui necessari.
- **Qualificacions:** Els estudiants han de poder consultar les qualificacions de les seves entregues juntament amb el feedback rebut pel professorat.
- **Entregues:** Els estudiants han de poder realitzar les entregues dins un curs.
- **Calendari:** Els estudiants han de poder visualitzar un calendari on siguin visibles les deadlines de les entregues i on puguin ells afegir alguna anotació.
- **Timeline:** Els estudiants han de poder visualitzar de manera dinàmica les deadlines de les entregues.

- **Fòrum:** Els professors han de poder agregar posts al fòrum de la seva assignatura.
- **Administració d'un curs:** Els professors han de poder administrar un curs, poden eliminar seccions, afegir-les, agregar entregues, eines per l'autoregulatge, etc.
- **Creació d'un curs:** Els administradors han de poder crear un curs.

3.1.2. Requeriments no funcionals

D'altra banda, els requisits no funcionals són les característiques més generals o transversals d'un sistema. Se centren en com el sistema ha de ser, més que en què ha de fer. Aquests requisits aborden aspectes com la usabilitat, el rendiment, la seguretat, la fiabilitat i altres aspectes de qualitat que afecten l'experiència global de l'usuari i el funcionament de la plataforma. En el nostre cas, podem analitzar els següents requisits:

- **Usabilitat:** La plataforma ha de ser intuïtiva i fàcil d'utilitzar per a usuaris de diferents nivells d'habilitat tècnica incloent una interfície atractiva i adaptada a diferents dispositius, com ara ordinadors d'escriptori, tauletes i telèfons mòbils.
- **Rendiment:** La plataforma ha de ser capaç de gestionar múltiples usuaris concurrents sense experimentar retard significatiu amb temps de càrrega ràpids i eficients.
- **Seguretat:** La plataforma ha d'implementar mesures de seguretat per protegir la informació personal i les dades dels usuaris, incloent-hi el xifratge de contrasenyes i la protecció contra accessos no autoritzats.
- **Escalabilitat:** L'arquitectura de la plataforma ha de ser escalable, permetent un creixement futur sense comprometre el rendiment.
- **Integració:** La plataforma ha de ser compatible amb serveis externs, com ara sistemes de gestió de l'aprenentatge existents, serveis d'autenticació única (SSO) i eines de col·laboració.

3.2. Backlog

En aquest apartat, es presenta el concepte del backlog adaptat al projecte i la seva rellevància en la planificació i desenvolupament de la plataforma. El backlog es converteix en una llista de funcionalitats prioritzades que inclou els requisits funcionals i no funcionals necessaris per a aconseguir els objectius proposats. Mitjançant el backlog, s'estableix un enfocament clar, es prioritzen les tasques i es fomenta la col·laboració entre els membres de l'equip de desenvolupament.

Aquest apartat explora com l'ús del backlog en el meu TFG permet una gestió eficient del projecte, garantint que les necessitats dels usuaris i els objectius del treball s'abordin de manera

efectiva. En la següent **Fig. 3**, podem visualitzar el backlog complet de totes les funcionalitats d'aquest projecte.

User Story	Descripció	Criteris d'Acceptació
U-1	Com a usuari no registrat vull poder accedir al lloc	<ul style="list-style-type: none"> - L'usuari no pot accedir a la pàgina principal sense haver-se registrat. - La pàgina inicial conté una regió que redirigirà l'usuari a una segona pàgina per poder-se registrar. - L'usuari pot navegar per la pàgina inicial sense interrupcions o errors.
URE-1	Com a estudiant vull poder veure el meu perfil	<ul style="list-style-type: none"> - L'usuari ha de poder accedir al seu perfil des de la pàgina d'inici del lloc web. - El perfil ha de mostrar la informació facilitada per l'usuari durant el registre, com ara el nom, adreça de correu electrònic i qualsevol altra informació rellevant. - L'usuari ha de poder editar qualsevol informació errònia o desactualitzada en el seu perfil. - Els canvis realitzats per l'usuari s'han de guardar i reflectir en el seu perfil de manera efectiva. - Un usuari no registrat no tindrà l'opció de poder accedir al seu perfil.
URE-2	Com a estudiant vull poder veure el meu progrés	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una pàgina o secció de la plataforma que mostri el seu progrés en els cursos en línia. - La pàgina de progrés ha de mostrar informació detallada sobre el progrés de l'estudiant en cada curs, inclòs el percentatge completat, les qualificacions obtingudes en els exàmens i tasques, i qualsevol altra informació rellevant. - L'estudiant ha de poder veure el seu progrés en temps real mentre avança en cada curs. - La pàgina de progrés ha de ser fàcil d'utilitzar i navegar, amb una presentació clara i concisa de la informació.

		<ul style="list-style-type: none"> - El sistema ha de guardar i registrar de manera fiable i precisa el progrés de l'estudiant, de manera que la informació sigui accessible i actualitzada en tot moment.
URE-3	Com a estudiant vull poder pujar tasques	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una secció o funcionalitat del lloc web per pujar tasques o treballs per ser avaluats pel professor. - El formulari de càrrega de tasques ha de permetre a l'estudiant adjuntar fitxers o introduir contingut relacionat amb la tasca. - Després d'enviar una tasca, l'estudiant ha de rebre una confirmació visual o un missatge de confirmació que la tasca s'ha enviat amb èxit. - El sistema ha de registrar de manera precisa la data i l'hora d'enviament de la tasca per a posteriors avaluacions. - El professor ha de poder veure les tasques enviades pels estudiants en una interfície adequada i tenir accés als fitxers adjunts o contingut relacionat. - El professor ha de poder assignar una qualificació o proporcionar comentaris a cada tasca enviada pels estudiants.
URE-4	Com a estudiant vull poder consultar el meu calendari acadèmic	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una pàgina o secció de la plataforma que mostri un calendari acadèmic. - El calendari ha de mostrar totes les activitats i lliuraments que l'estudiant tingui en un període determinat.
URE-5	Com a estudiant vull poder modificar una entrega	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una entrega. - Un botó redirigirà l'estudiant a una finestra emergent amb l'explorador de fitxers que permetrà afegir una nova entrega. - En acceptar els canvis, l'entrega serà modificada.
URE-6	Com a estudiant vull poder accedir a un xat amb una intel·ligència artificial de tipus chatGPT	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una pàgina o secció de la plataforma que mostri un xat amb una intel·ligència artificial. - El chatbot ha de ser capaç de respondre totes les preguntes que faci l'estudiant de manera ràpida.

URE-7	Com a estudiant vull poder accedir a un timeline amb esdeveniments de data límit per tal d'evitar oblidar-me de fer els lliuraments	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una pàgina o secció de la plataforma que mostri un timeline amb els lliuraments recents i l'activitat de l'usuari. - El timeline ha de mostrar clarament les dates límits dels lliuraments pendents i ressaltarà aquells que estiguin a punt d'acabar. -L'estudiant ha de rebre una notificació si està a punt d'acabar.
URE-8	Com a estudiant vull poder accedir al fòrum de notícies per tal de veure les últimes novetats dels cursos	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una secció de notícies de cada curs. - Els missatges del fòrum han de estar ordenats cronològicament.
URE-9	Com a estudiant, vull poder veure un resum de les analítiques de tots els cursos per comprovar si estic progressant adequadament.	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a un apartat anomenat "Dashboard". - El "Dashboard" mostrarà un resum de les activitats i el progrés de l'estudiant.
URE-10	Com a estudiant, vull poder realitzar qüestionaris en un curs perquè el professor pugui avaluar el meu coneixement o graduar el meu procés educatiu.	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a la secció de qüestionaris d'un curs. - Els qüestionaris han de ser accessibles i navegables per a l'estudiant. - Les respostes enviades pels estudiants han de ser enregistrades i accessibles pel professor per a l'avaluació. - Els resultats dels qüestionaris han de ser mostrats a l'estudiant.
URE-11	Com a estudiant, vull poder realitzar activitats d'avaluació entre iguals per avaluar els meus companys i que ells em puguin avaluar a mi.	<ul style="list-style-type: none"> - L'estudiant ha de tenir accés a una secció per a realitzar activitats d'avaluació entre iguals - L'estudiant ha de poder avaluar als seus companys i rebre avaluacions d'ells. - Les activitats d'avaluació han de ser accessibles i fàcils d'utilitzar per a l'estudiant.

URE-12	Com a estudiant, vull poder visualitzar un resum de les meves qualificacions en tots els cursos per veure les meves notes.	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una secció que mostri un resum de les seves qualificacions en tots els cursos. - Les qualificacions han de ser mostrades de manera clara i concisa per a l'estudiant.
URE-13	Com a estudiant, vull poder descarregar fitxers d'un curs pujats pel professorat per poder accedir al material educatiu.	<ul style="list-style-type: none"> - L'estudiant ha de tenir accés a una funcionalitat per a descarregar fitxers dels cursos. - Els fitxers han de ser fàcilment accessibles i descarregables per a l'estudiant. - Els fitxers han de correspondre amb el material educatiu del curs.
URE-14	Com a estudiant, vull poder tancar la sessió de l'aplicació per sortir del meu compte.	<ul style="list-style-type: none"> - L'estudiant ha de poder accedir a una opció de "Tancar sessió" a la interfície de l'aplicació. - En fer clic a "Tancar sessió", l'estudiant ha de ser redirigit a la pàgina d'inici de sessió i haver tancat la sessió de manera efectiva.
A-1	Com a administrador, vull poder crear i publicar cursos en línia.	<ul style="list-style-type: none"> - L'administrador ha de poder accedir a una pàgina o secció de l'aplicació per a crear nous cursos. - L'administrador ha de poder introduir el nom, la descripció i les dates d'inici i finalització del curs. - L'administrador ha de poder afegir i organitzar materials d'estudi, com ara textos, vídeos i presentacions. - L'administrador ha de poder publicar el curs i fer-lo visible per als estudiants inscrits.
URP-1	Com a professor, vull poder qualificar tasques i exàmens en línia per poder avaluar el progrés dels estudiants.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui accedir a les entregues de les tasques dels alumnes. - El professor ha de poder qualificar les tasques i exàmens realitzats pels estudiants.
URP-2	Com a professor, vull poder afegir estudiants al meu curs per poder-los avaluar.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui afegir estudiants al seu curs. - El professor ha de poder introduir les dades dels estudiants, com ara nom, correu electrònic, etc.

		<ul style="list-style-type: none"> - Els estudiants afegits han de quedar registrats com a participants del curs.
URP-3	Com a professor, vull poder editar un curs per poder modificar-lo al meu gust.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui editar les dades d'un curs existent. - El professor ha de poder modificar el nom, la descripció, les dates, els materials, etc., del curs. - Els canvis fets pel professor han de quedar guardats i reflectits de manera efectiva en el curs.
URP-4	Com a professor, vull poder afegir una publicació al fòrum de notícies del meu curs per poder notificar als meus alumnes les novetats.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui afegir una publicació al fòrum de notícies del curs. - El professor ha de poder redactar i publicar la publicació amb la informació o les novetats que vol compartir amb els alumnes. - La publicació ha de ser visible pels estudiants del curs.
URP-5	Com a professor, vull poder carregar les notes d'una tasca a través d'un fitxer Excel automàticament per poder agilitzar el procés.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui carregar les notes d'una tasca mitjançant un fitxer Excel. - El sistema ha de processar automàticament les dades del fitxer Excel i actualitzar les notes dels estudiants a la plataforma. - Les notes actualitzades han de quedar registrades de manera efectiva per a cada estudiant i tasca corresponents.
URP-6	Com a professor, vull poder carregar les notes d'una tasca a través d'un fitxer Excel automàticament per poder agilitzar el procés.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui carregar les notes d'una tasca mitjançant un fitxer Excel. - El sistema ha de processar automàticament les dades del fitxer Excel i actualitzar les notes de l'estudiant per a la tasca corresponent. - Les notes actualitzades han de quedar registrades de manera efectiva i visible per als estudiants i el professor.

URP-7	Com a professor, vull poder crear activitats de qüestionaris, entregues i avaluació entre iguals per poder avaluar els meus alumnes.	<ul style="list-style-type: none"> - El professor ha de poder accedir a una pàgina o secció de l'aplicació on pugui crear activitats de qüestionaris, entregues i avaluació entre iguals. - El professor ha de poder definir les característiques de l'activitat, com ara la data límit, el nombre de preguntes, el tipus de lliurament, etc. - Les activitats creades han de quedar enregistrades i ser accessibles per als estudiants corresponents.
URP-8	Com a professor, vull poder diferenciar les 3 etapes de l'aprenentatge autoregulat perquè els alumnes siguin conscients en tot moment de quina etapa es troben.	<ul style="list-style-type: none"> - El professor ha de poder proporcionar als alumnes una representació visual de les 3 etapes de l'aprenentatge autoregulat (com a gràfic o indicador visual). - L'aplicació ha de mostrar i actualitzar la posició de l'estudiant en les etapes de manera dinàmica i precisa. - Els alumnes han de poder identificar clarament en quina etapa es troben i quines accions suggereix cada etapa.

Fig. 3. Backlog complet de Studia.

3.3 Diagrama de Gantt

Un diagrama de Gantt és una representació visual que mostra les activitats d'un projecte en forma de barres horitzontals en un eix de temps ,s'utilitza per planificar, programar i controlar el progrés de les tasques d'un projecte, mostrant les dates d'inici i finalització de cada activitat.

En la següent **Fig. 4** podem veure el Diagrama de Gantt de la planificació d'aquest projecte.

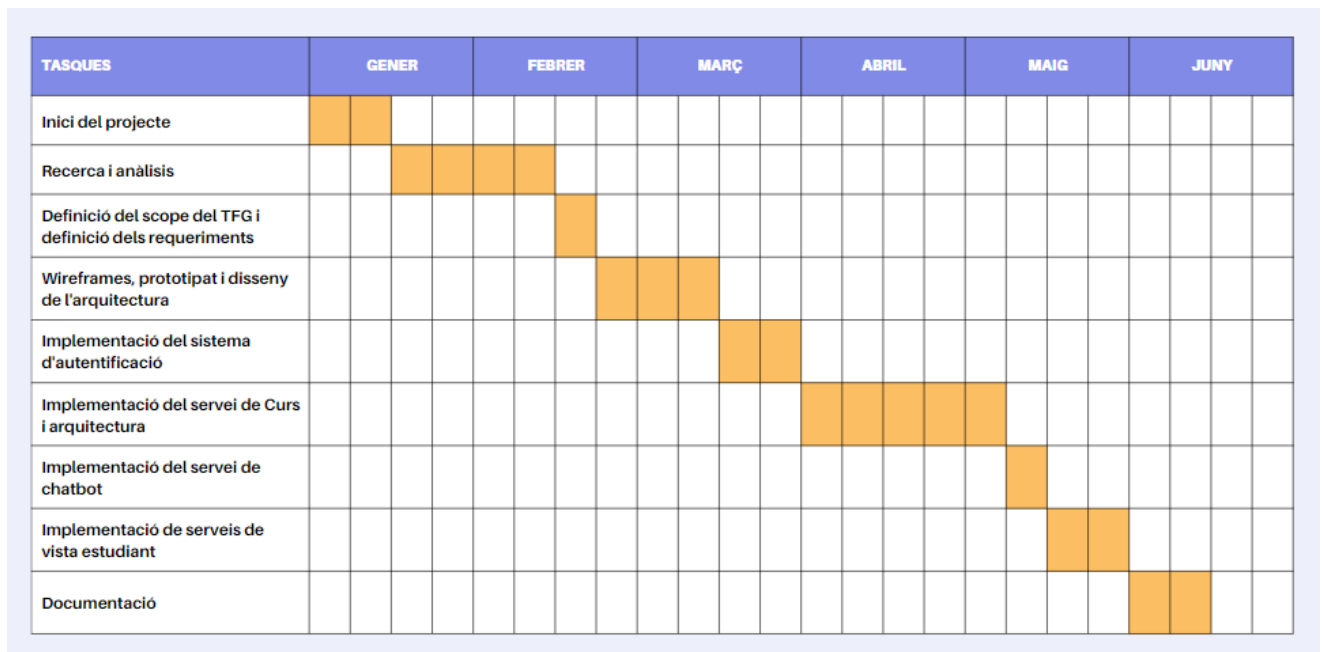


Fig. 4. Diagrama de Gantt

Observem que he dedicat gran part del temps en el disseny, ja que he volgut que sigui prou robust i escalable perquè sigui fàcilment ampliable. Per la part de la implementació, tot el sistema dels cursos ha sigut el més costós de realitzar, perquè és el principal atractiu de la pàgina i ha sigut complicat reflectir les diferents fases educatives.

4. Disseny

En aquesta secció introduïrem el disseny de la plataforma, des de l'arquitectura de l'aplicació fins als prototips d'alta fidelitat que acabaran conformant l'aplicació final.

4.1. Arquitectura de l'aplicació

En la **Fig. 4**, es pot observar l'arquitectura que fa servir la web, així com els seus serveis i bases de dades que s'han utilitzat.

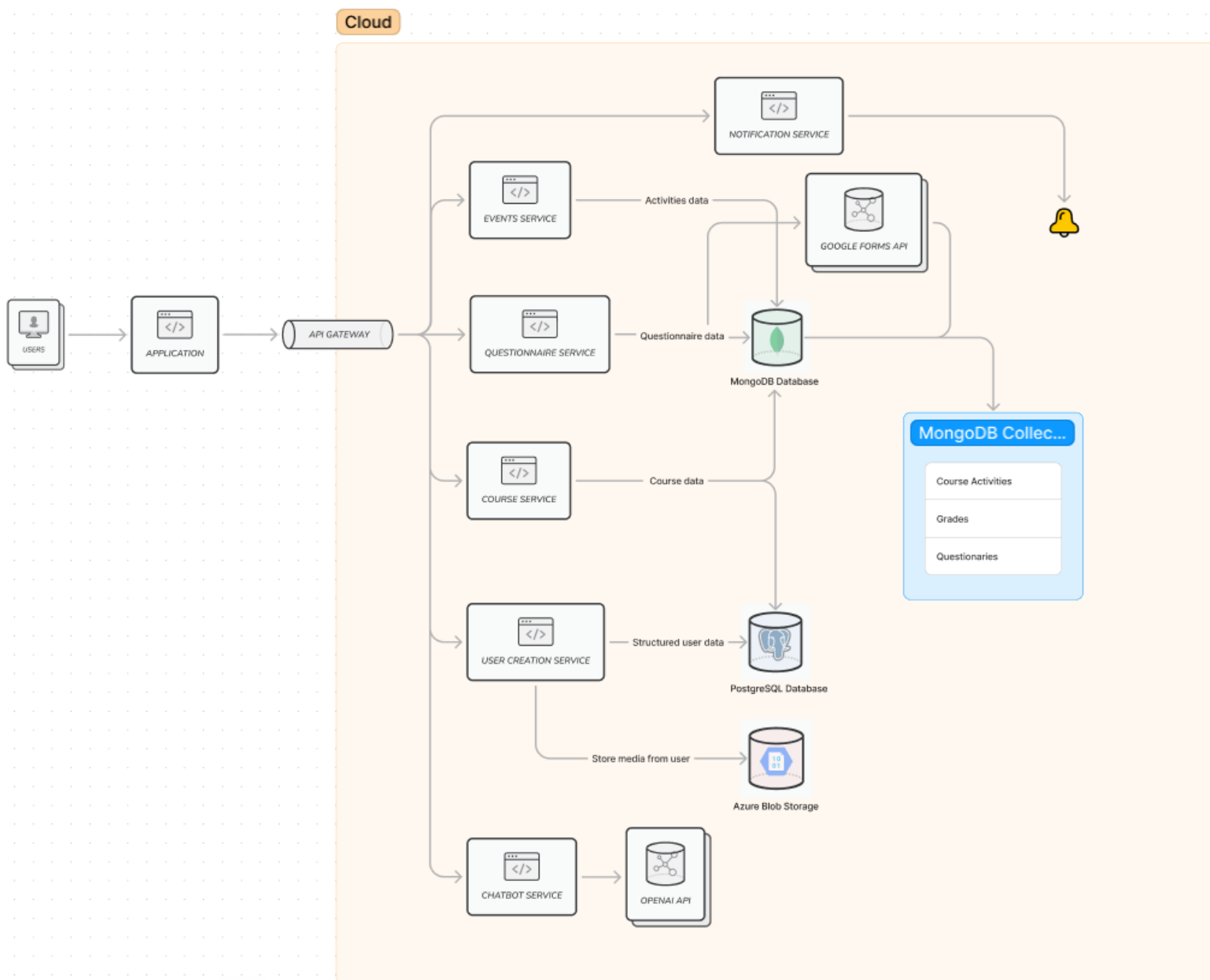


Fig. 4. Arquitectura de Studia.

Com podem veure, tenim una sèrie d'usuaris que interactuen amb la web, aquesta demandarà la informació a l'API Gateway, aquesta es comunica amb diferents serveis que tenen diverses funcions.

Començant per 'User Creation Service', aquest serà l'encarregat de gestionar la creació i edició dels comptes dels diferents usuaris. Aquest servei té un contacte directe amb dues de les nostres bases de dades que utilitzem en el sistema, una relacional PostgreSQL [4] i un sistema de storage en cloud com és Azure Blob Storage.

En la nostra base de dades relacional, emmagatzemarem tota la informació estructurada dels usuaris i la informació base dels cursos, en la [Fig. Anexos 5](#), podem observar la relació que tenim en la nostre PostgreSQL.

Analitzem que el nostre model és bastant extens i la majoria de les taules tenen relació entre elles, a continuació explicarem les taules més importants que trobem.

- La taula "UserAccount" és part de l'aplicació "accounts" i representa els usuaris registrats al sistema. Conté informació detallada sobre cada usuari, com ara la seva adreça de correu electrònic, nom, contrasenya i altres atributs relacionats amb el seu perfil. A més, aquesta taula hereta camps i relacions de les taules "AbstractBaseUser" i "PermissionsMixin" de l'aplicació "django.contrib.auth". Els usuaris poden pertànyer a grups i tenir permisos específics associats, la qual cosa s'estableix mitjançant relacions de molts a molts amb les taules relacionades.
- La taula "ContentType" present a l'aplicació "django.contrib.contenttypes", emmagatzema informació sobre els tipus de contingut disponibles al sistema. Els tipus de contingut representen els models o les entitats de dades a l'aplicació Django. Cada registre en aquesta taula conté metadades sobre un tipus de contingut específic, com ara el nom de l'aplicació i el nom del model. Aquesta taula s'utilitza en altres relacions per establir associacions entre els models i els permisos, com podrem veure en les següents taules.
- La taula "Session" es troba a l'aplicació "django.contrib.sessions" i s'utilitza per emmagatzemar informació sobre les sessions dels usuaris al sistema. Cada registre representa una sessió d'usuari activa i conté dades relacionades amb la sessió, com ara la clau de sessió, la data de caducitat i les dades de la sessió en si. La taula hereta camps i relacions de la taula "AbstractBaseSession".
- La taula "Permission" es troba a l'aplicació "django.contrib.admin" i emmagatzema registres d'activitat de l'administrador. Cada registre representa una entrada de registre que enregistra una acció realitzada al panell d'administració de Django. Algunes de les accions enregistrades poden incloure la creació, actualització o eliminació d'objectes al sistema. La taula està relacionada amb la taula "UserAccount" mitjançant una clau externa per identificar l'usuari

responsable de l'acció registrada. També està relacionada amb la taula "ContentType" per associar el tipus de contingut de l'objecte afectat per l'acció.

- La taula "Group" pertany a l'aplicació "django.contrib.auth" i s'utilitza per emmagatzemar permisos al sistema. Cada registre en aquesta taula representa un permís específic que es pot assignar a un usuari o grup al sistema. Els permisos controlen l'accés i l'autorització a diferents parts i funcions del sistema. La taula està relacionada amb la taula "ContentType" per associar el tipus de contingut al qual s'aplica el permís.
- Finalment, trobem la taula "Course" aquesta conté diverses columnes rellevants per al sistema. Entre aquestes columnes es troba "id", que actua com a clau primària de la taula. Aquesta columna proporciona un identificador únic per a cada curs enregistrat al sistema. Un altre camp important és "title", que emmagatzema el títol del curs. Aquesta informació és crucial per a l'ús dels usuaris, ja que els ajuda a identificar i seleccionar els cursos en els quals estan inscrits. Una altra columna rellevant és "professor", que és una clau externa que fa referència a la taula "UserAccount". Aquest camp estableix una relació entre el curs i el professor que l'imparteix. La columna "course_photo" conté una URL que apunta a una imatge relacionada amb el curs que pot representar una miniatura del curs o proporcionar informació visual addicional sobre el mateix. Finalment, amb les columnes "description", "start_date" i "end_date" permet proporcionar informació detallada als usuaris sobre els continguts i la durada del curs. També permeten als estudiants planificar el seu temps i prendre decisions informades sobre la seva participació en el curs.

Continuant amb el servei de "User Creation", com hem dit les imatges s'emmagatzemen en PostgreSQL com un URL, això es deu al fet que utilitzem un servei extern de Storage, ja que és bastant costós guardar totes aquestes imatges en una BBDD.

En aquest cas, hem optat per fer ús les eines que ens ofereix l'ecosistema d'Azure, en aquest cas, Azure Blob Storage on hem creat un contenidor d'imatges anomenat 'studiainages' que podem veure en la **Fig. 5**.

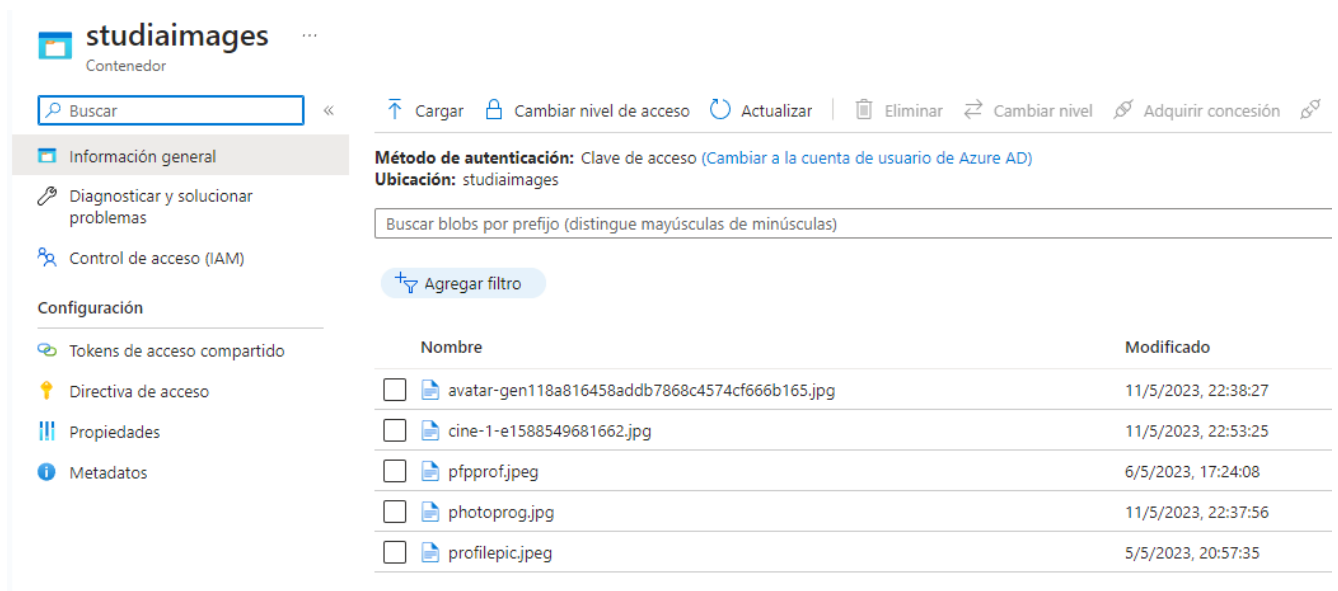


Fig. 5. Contenedor de imágenes de la web.

Azure Blob Storage ofereix diverses característiques tècniques rellevants. Per començar, suporta diversos nivells de rendiment i durabilitat segons les necessitats de l'aplicació. Això inclou l'opció d'utilitzar discos SSD per a un accés ràpid a les dades o discos magnètics per a un emmagatzematge més econòmic.

A més, també proporciona replicació de dades automàtica i redundància per garantir la durabilitat de les dades. Les dades es poden replicar de forma síncrona o asíncrona en diverses regions geogràfiques per a una major resistència a fallades i una millor disponibilitat dels recursos.

El servei també ofereix controls avançats d'accés i seguretat per protegir les dades emmagatzemades. Això inclou autenticació basada en claus d'accés, polítiques d'accés compartit i integració amb Azure Active Directory per a una gestió centralitzada d'autoritzacions.

Azure Blob Storage és altament escalable, permetent l'emmagatzematge i recuperació eficient de grans volums de dades on també els clients poden augmentar o reduir la capacitat d'emmagatzematge segons les seves necessitats i només paguen pels recursos que consumeixen, que en el nostre cas s'assumeix un cost molt baix, ja que pujar media no és el principal atractiu de l'aplicació (al contrari que una xarxa social estil instagram per exemple).

Un altre servei que trobem en l'arquitectura és el "Course Service", aquest serà l'encarregat de proporcionar la informació relacionada amb els cursos incloent també les activitats que es troben dins d'aquests en PostgreSQL com la descripció, títol, professor, data d'inici, data de finalització, etc. També agafa altres dades no estructurades de la nostra base de dades en MongoDB.

MongoDB [5] és un sistema de gestió de bases de dades NoSQL, orientat a documents. A diferència dels sistemes de bases de dades relacionals tradicionals com el nostre PostgreSQL, MongoDB emmagatzema les dades en forma de documents JSON flexibles, anomenats "col·leccions", en comptes de taules amb files i columnes.

És àmpliament utilitzat en entorns moderns de desenvolupament de software, especialment en aplicacions web i mòbils, ja que ofereix diverses característiques clau que el fan atractiu per a aquests casos d'ús. Un dels principals avantatges és la seva escalabilitat horitzontal, que permet gestionar grans volums de dades i suportar càrregues de treball intensives. A més, MongoDB proporciona una flexibilitat notable en l'esquema de dades pel fet que no requereix un esquema fix prèviament definit. Això significa que des de l'inici del desenvolupament de Studia, he tingut la facilitat de poder afegir o modificar els camps dels documents de manera flexible, sense haver de migrar les dades existents. Per utilitzar mongoDB en el cloud, he optat per utilitzar els serveis de Atlas MongoDB.

Atlas MongoDB és un servei de base de dades al núvol completament administrat que ofereix MongoDB. Està dissenyat per simplificar la gestió i el desplegament de bases de dades MongoDB, permetent-te centrar-te en el desenvolupament de la teva aplicació en lloc de preocupar-te per la infraestructura subjacent.

Aquest proporciona una solució integral per allotjar, gestionar i escalabilitzar bases de dades MongoDB al núvol. Fent servir Atlas, pots aprofitar els beneficis de MongoDB sense haver-te de preocupar per la configuració i el manteniment de servidors, el monitoratge, la còpia de seguretat i l'escalabilitat, el qual facilita molt la feina de cara a un futur deployment. En la **Fig. 6** podem veure el deployment de la nostra aplicació de MongoDB junt amb estadístiques de monitoratge com les lectures i escriptures que es fan en cada moment. En cas que la nostra aplicació augmentés de forma exponencial és de gran utilitat per analitzar els colls d'ampolla que poden haver-hi entre altres coses.

MARCOS'S ORG - 2023-04-16 > PROJECT 0

Database Deployments

Find a database deployment...

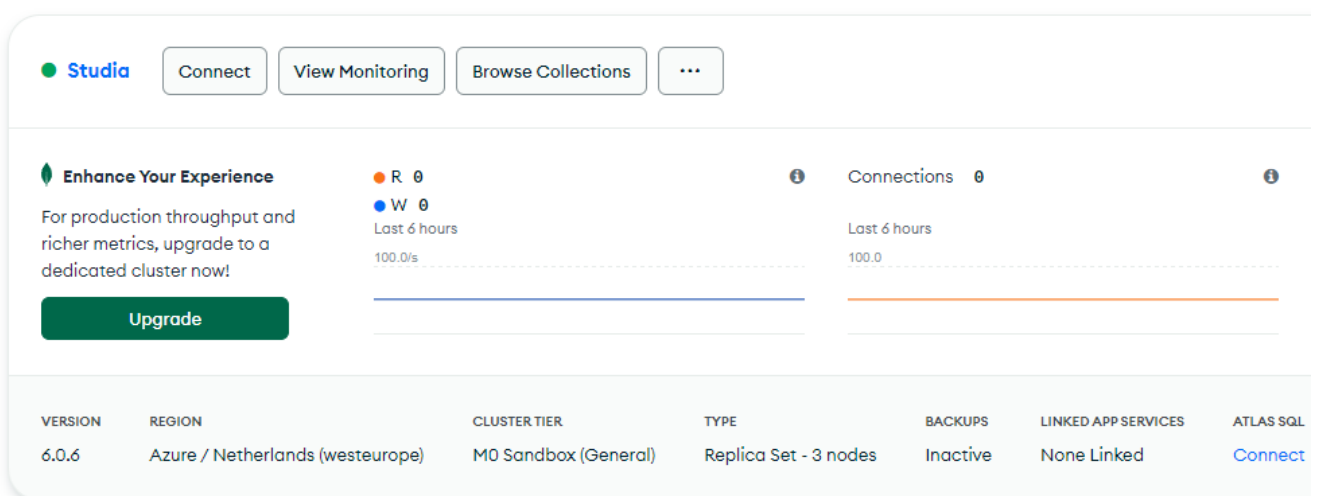


Fig. 6. Deployment de Studia MongoDB en Atlas.

Finalment, per gestionar el nostre MongoDB he utilitzat l'eina de MongoDB Compass, podem veure la interfície que utilitza en la **Fig.7**, aquest és una eina gràfica i intuïtiva que s'utilitza per interactuar i administrar bases de dades MongoDB. Proporciona una interfície visual que facilita l'exploració, l'anàlisi i la manipulació de les dades emmagatzemades en una base de dades MongoDB. Un dels principals avantatges és la seva capacitat de visualitzar i comprendre l'estructura de les dades. Permet explorar les col·leccions, els documents i els camps d'una base de dades de manera intuïtiva, la qual cosa facilita la comprensió de l'organització i la relació entre les dades. A més, ofereix la possibilitat de realitzar consultes avançades utilitzant una interfície gràfica o utilitzant consultes en format JSON.

Una altra característica destacada de MongoDB Compass és la seva capacitat de efectuar operacions de CRUD (crear, llegir, actualitzar, eliminar) de forma senzilla. A través de la interfície gràfica, pots inserir nous documents, actualitzar camps existents, eliminar registres i dur a terme consultes per recuperar dades específiques. Això simplifica el procés d'interacció amb la base de dades i agilitza el desenvolupament d'aplicacions.

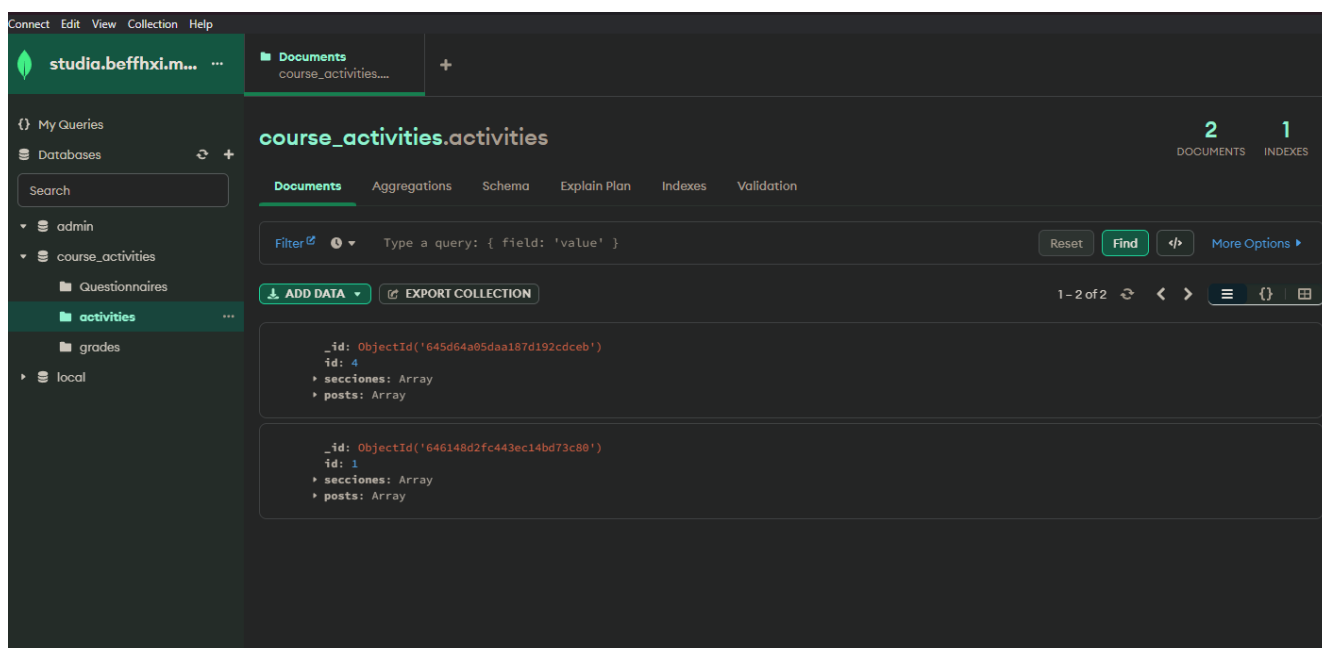


Fig. 7. Interfície gràfica de MongoDB Compass.

Cada camp que trobem en la interfície en la col·lecció de 'activities' representa les dades dins un curs, en el JSON de la **Fig. 8** podem veure un exemple de com guardem els cursos dins de mongoDB.

```
{
  "id":1,
  "secciones":[
    {
      "titulo":"Sección 1: Algebra Relacional",
```



```

    "subsecciones":[
      {
        "titulo":"MSLQ or DLSQ self-regulation questionnaire",
        "fase":"Forethought",
        "finished":"True",
        "fecha_inicio":"01-05-2023",
        "fecha_fin":"07-05-2023",
        "duracion":"1h",
        "contenido":[
          (...)
        ]
      }
    ]
  }
]

```

Fig. 8. JSON generat d'un curs.

Aquesta en una versió resumida d'un curs de prova que tenim actualment a l'aplicació web, aquest està format pels següents camps:

- id: Aquest valor identifica de manera única el curs.
- secciones: Aquesta és una llista que conté les seccions del curs. En aquest cas, només hi ha una secció.
- titulo: Aquest camp indica el títol de la secció, que és "Sección 1: Algebra Relacional".
- subsecciones: Aquesta és una llista que conté les subseccions dins de la secció. En aquest cas, només hi ha una subsecció.
- titulo: Aquest camp indica el títol de la subsecció.
- fase: Aquest camp indica la fase de la subsecció, que pot representar tres valors "Forethought", "Performance", "Self-Reflection", que representen les 3 fases de l'aprenentatge autoregulat.
- finished: Aquest camp indica si la subsecció s'ha completat o no. En aquest cas, té el valor "True", el que significa que la subsecció ha estat completada.
- fecha_inicio: Aquest camp indica la data d'inici de la subsecció, això vol dir que si la data actual < fecha_inicio, aquesta subsecció no serà accessible pels estudiants.
- fecha_fin: Aquest camp indica la data de finalització de la subsecció.
- duracion: Aquest camp indica la durada estimada per completar la subsecció. Aquesta informació ajuda a planificar i organitzar el temps necessari per a aquesta part del curs.

- contenido: Aquest camp conté el contingut específic de la subsecció, pot incloure tant text, com diferents de les activitats mencionades anteriorment (qüestionaris, avaluació entre iguals, entregues, fitxers).

Tant el "Questionnaire Service" com el "Event Service" també escriuen i llegeixen en la mongoDB, ja que aquests serveis formen part d'informació interna d'un curs.

Finalment, el notification service serà l'encarregat d'escoltar els esdeveniments que succeeixen en la pàgina, com per exemple nous posts en el fòrum, pujada de notes o noves entregues, i farà una notificació als usuaris.

4.2. Tecnologies utilitzades

En aquest apartat, exposaré les diferents tecnologies que he utilitzat al llarg d'aquest projecte web, incloure tant els diferents llenguatges de programació, com els Frameworks utilitzats i finalment les plataformes que m'han sigut d'utilitat pel desenvolupament de principi a fi.

4.2.1. Llenguatges de programació (i altres)

Els següents llenguatges que exposarem ens permeten tenir una base sòlida per a desenvolupar aplicacions web interactives, personalitzades i funcionals. Cada llenguatge té el seu propòsit específic i conjunt de funcionalitats, i utilitzar-los en combinació pot permetre crear aplicacions web més enriquidores

4.2.1.1. HTML

HTML [\[11\]](#) és el llenguatge estàndard utilitzat per crear i estructurar el contingut d'una pàgina web. Mitjançant etiquetes i elements, HTML defineix la manera en què s'organitzen i es presenten els textos, les imatges, els enllaços i altres elements multimèdia en una pàgina web. És la base per a la construcció d'interfícies web i proporciona l'estructura bàsica d'un lloc web.

4.2.1.2. CSS

CSS [\[10\]](#) és un llenguatge de fulls d'estil que complementa HTML. Amb CSS, pots controlar l'aparença i el disseny d'una pàgina web. Mitjançant regles i seleccions, CSS permet aplicar estils com colors, tipus de lletra, marges, dimensions i altres propietats visuals als elements HTML. Això permet personalitzar l'aparença i la presentació d'un lloc web, aportant estil i coherència visual.

4.2.1.3. Javascript

JavaScript [\[9\]](#) és un llenguatge de programació orientat a objectes que s'executa al costat del client (al navegador web). Una de les característiques més destacades de JavaScript és la seva capacitat per afegir interactivitat i dinamisme a les pàgines web amb aquest pots respondre a esdeveniments com clics de botons, moviment del ratolí o enviaments de formularis. Això et permet

validar dades d'entrada, realitzar canvis en temps real a la pàgina, carregar contingut de manera asíncrona (utilitzant AJAX) i interactuar amb l'usuari en general.

A més, JavaScript té una ampla gamma de biblioteques i frameworks com jQuery, React i Angular, que faciliten el desenvolupament d'aplicacions web més complexes i potents, oferint una gran varietat de possibilitats i sobretot, reduint de manera molt significativa el temps que es necessita per programar una pàgina web, en el següent apartat parlarem més en profunditat.

4.2.1.4. Python

Python [\[12\]](#) és un llenguatge de programació versàtil i de propòsit general. Una de les característiques més rellevants de Python és la seva sintaxi llegible i elegant, que facilita la comprensió del codi i el treball en equip. És un llenguatge interpretat, el que significa que no cal compilar-lo abans de l'execució, el que agilitza el cicle de desenvolupament.

Python ofereix una gran quantitat de llibreries i mòduls que cobreixen una àmplia varietat d'àrees, com l'anàlisi de dades (NumPy, Pandas), la visualització de dades (Matplotlib, Seaborn), l'aprenentatge automàtic (Scikit-learn, TensorFlow) i la web (Django, Flask) aquest és el que realment és interessant de cara el nostre projecte, ja que ens permet de manera molt ràpida i senzilla realitzar una REST API .

Gràcies a la seva ampla comunitat de desenvolupadors, és fàcil trobar suport, tutorials i exemples de codi per a una gran varietat de tasques i problemes, també és multiplataforma, el que significa que el mateix codi es pot executar en diferents sistemes operatius sense haver de realitzar grans canvis.

Python també destaca per la seva facilitat d'ús i rapidesa en l'escriptura de codi, de fet, aquesta és la principal raó per la qual m'he decantat per utilitzar aquest llenguatge, penso que realment és molt important utilitzar eines amb les quals et trobis realment còmode, en aquest cas python és un bon exemple afegint també que té molt potencial en l'entorn web.

4.2.2. Frameworks

Els frameworks són conjunts d'eines, llibreries i codi predefinit que proporcionen una estructura i funcionalitats per desenvolupar aplicacions més ràpidament. Simplifiquen el procés de desenvolupament i permeten als desenvolupadors centrar-se en la lògica de l'aplicació en lloc de gestionar aspectes tècnics i repetitius. En aquest apartat mostrarem aquells que hem utilitzat en Studia.

4.2.2.1. React

React [6] és una llibreria de JavaScript de codi obert que es fa servir per construir interfícies d'usuari (UI) interactives i reutilitzables, en aquest context no és ben bé un framework ja que et proporciona una sèrie d'eines per desenvolupar, però no 'obliga' a utilitzar alguna d'aquestes, tampoc és estricta amb la sintaxi com podrías ser el cas de Angular. React va ser desenvolupada i mantinguda per Facebook, és coneguda per la seva eficiència, rendiment i facilitat d'ús, a més a més, és una de les llibreries més populars utilitzada per grans empreses com airBnb, Uber, Netflix, Twitter, Reddit o Paypal.

Una de les principals característiques de React és la seva arquitectura basada en components. Amb React, pots descompondre l'aplicació en components reutilitzables, cada un dels quals té la seva pròpia lògica i interfície d'usuari. Això permet una gestió més senzilla de l'estat de l'aplicació i facilita la seva evolució i manteniment.

React utilitza JSX, una extensió de sintaxi que combina JavaScript i HTML, per a la creació de components. Això permet als desenvolupadors escriure codi semblant a HTML per descriure la interfície d'usuari, mentre que al mateix temps pot utilitzar JavaScript per gestionar la lògica i les dades de l'aplicació. A més, React utilitza un algorisme de reconciliació eficient per actualitzar només els components que han canviat, minimitzant la càrrega de renderització i optimitzant el rendiment de l'aplicació.

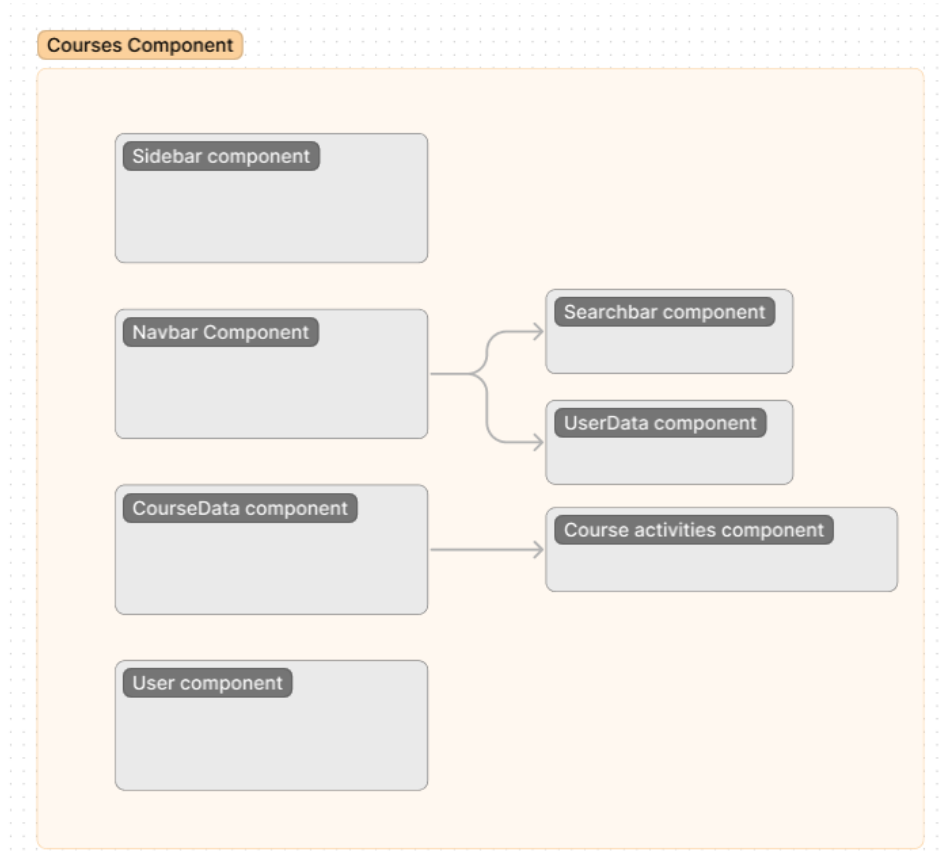


Fig. 9. Exemple de component d'un curs de Studia.

En la **Fig. 9**, podem veure un exemple real d'un component de la web, en aquest cas de la pàgina principal on es mostren tots els cursos de l'usuari. Observem que està format per una sèrie de components que inclús alguns d'aquests també estan formats per altres components.

La raó per la qual m'he decantat per utilitzar React, es la prèvia experiència que tenia en aquesta llibreria per altres projectes que he realitzat, afegint també que m'agrada la llibertat i facilitat que et dona a l'hora de fer qualsevol cosa com podria simplement fer un Route, exportar un component o guardar l'estat d'una variable dinàmicament.

4.2.2.2. Redux

Redux [16] és una llibreria de gestió d'estat previsible per a aplicacions JavaScript, que s'utilitza sovint en conjunt amb React. Si bé Redux no és estrictament un framework, és una eina potent per gestionar l'estat global de l'aplicació, que és especialment útil en aplicacions de mida mitjana a gran.

L'objectiu principal de Redux és facilitar la gestió de l'estat de l'aplicació i la comunicació entre components. Utilitza un patró de disseny anomenat Flux, que estableix una direcció unidireccional per al flux de dades a l'aplicació.

A Redux, l'estat de l'aplicació es guarda en un magatzem únic anomenat "store". Els components poden enviar accions a aquest magatzem, que descriuen els canvis que es volen realitzar a l'estat. Redux utilitza "reducers" per processar les accions i actualitzar l'estat de l'aplicació. Una vegada que l'estat s'actualitza, els components React connectats a Redux reben les actualitzacions i es tornen a renderitzar de manera eficient.

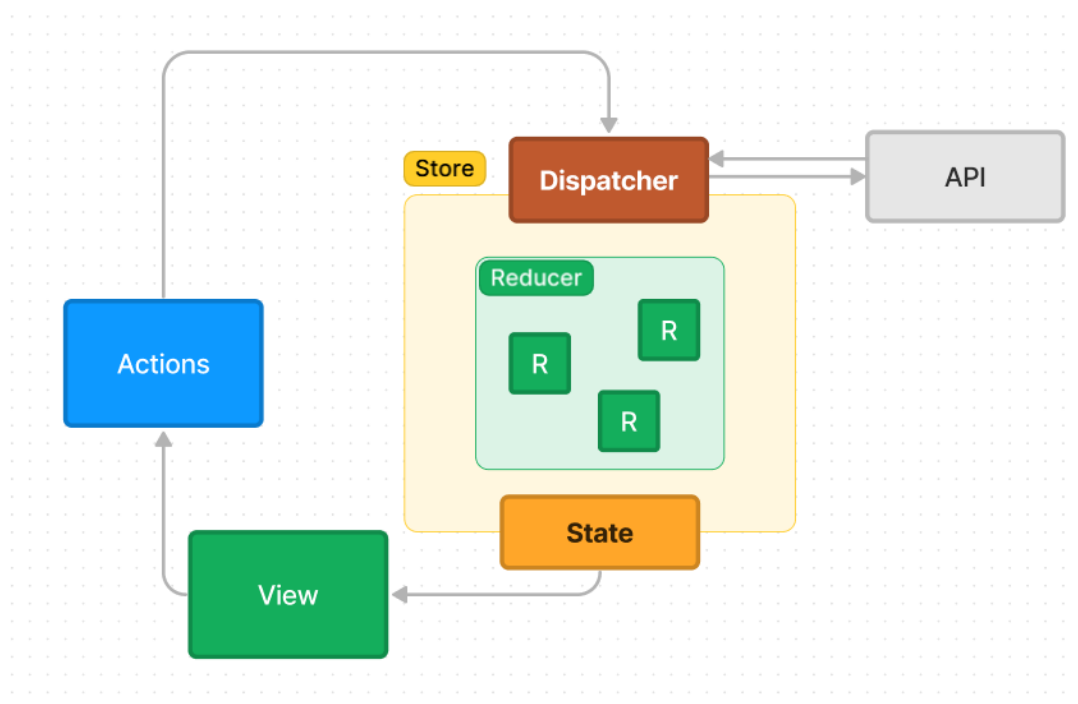


Fig. 10. Diagrama de flux de Redux.

En la **Fig. 10**, veiem el descrit anteriorment, afegint les accions que son produïdes en la vista de manera asíncrona per part dels usuaris de l'aplicació.

Entre altres coses, redux ha sigut fundamental en Studia per elaborar tot el relacionat amb el tema de la autenticació i control d'estat dels usuaris.

4.2.2.3. Django

Django [\[15\]](#) és un framework de desenvolupament web de codi obert basat en Python. És conegut per la seva simplicitat, eficiència i escalabilitat, i és àmpliament utilitzat per construir aplicacions web de tota mena.

Django proporciona una estructura d'aplicació robusta i completa, que facilita la creació d'aplicacions web sofisticades i segures. El framework ofereix un conjunt de components predefinits i biblioteques que agilitzen el desenvolupament, com ara autenticació d'usuaris, gestió de bases de dades, enrutament de URL, generació de formularis, etc.

També ofereix la possibilitat de definir models de dades per representar les entitats de un sistema, en el meu cas, m'ha permès definir el model de usuaris i cursos. Django s'encarrega de gestionar la capa de persistència de dades, permetent-te interactuar amb bases de dades relacionals de manera senzilla i eficient. A més, Django proporciona un ORM (Object-Relational Mapping) que et permet interactuar amb les dades utilitzant codi Python en comptes de SQL. És important mencionar que aquesta ORM està complement optimitzada per bases de dades relacionals, no permet de manera explícita treballar amb bases de dades no relacionals, com MongoDB.

Una característica destacada de Django és el seu sistema d'administrar automàticament la interfície d'administració. Amb només unes poques línies de codi, pots obtenir una interfície d'administració completa i personalitzable per gestionar les entitats del teu sistema. Aquesta funcionalitat estalvia temps de desenvolupament i permet als administradors gestionar el contingut de la plataforma.

La possibilitat de desenvolupar ràpidament una REST API, la sintaxi de python i la prèvia experiència que tenia amb aquest framework, han sigut les principals raons per fer servir Django en la part del backend.

4.2.2.4. Djoser

Djoser [\[17\]](#) és una biblioteca Python que s'utilitza com a extensió de Django per a la gestió de l'autenticació d'usuaris a través d'API REST. Si bé Djoser no és un framework en si mateix, és una eina poderosa per simplificar i agilitzar el desenvolupament d'aplicacions web que utilitzen Django per a la REST API.

Djoser proporciona funcionalitats com la gestió de registre d'usuaris, inici de sessió, restabliment de contrasenyes i altres accions relacionades amb l'autenticació d'usuaris. Amb només uns pocs passos de configuració, pots integrar Djoser a la teva aplicació Django i obtenir rutes i vistes predefinides per gestionar les operacions d'autenticació.

A més, Djoser permet personalitzar les respostes de l'API i els correus electrònics enviats als usuaris durant el procés d'autenticació. Això et permet adaptar l'experiència de l'usuari i proporcionar missatges i informació personalitzada.

Aquesta llibreria complementada amb redux permet fer un setup d'autenticació end-to-end de manera molt estable i el més important: segura.

4.2.2.5. Tailwind

Tailwind CSS [\[1\]](#) és un framework de CSS utilitzat per desenvolupar interfícies d'usuari modernes i atractives.

Una de les principals característiques de Tailwind és el seu enfocament utilitari. En lloc d'utilitzar classes predefinides per estil i disposició, Tailwind proporciona un conjunt de classes utilitats que permeten aplicar estils específics a elements HTML. Això significa que pots construir de forma modular el teu estil combinant les classes utilitats de Tailwind segons les teves necessitats.

Amb Tailwind, tens accés a una àmplia gamma de classes utilitats per estil, mida, espaiat, colors, tipografia i altres propietats. Això et permet personalitzar l'aparença de cada element de la teva plataforma sense necessitat d'escriure CSS personalitzat. Tailwind fomenta la reutilització de classes i facilita la creació d'interfícies coherents i estandaritzades.

4.2.3 Plataformes

En el següent subapartat, explicaré les diferents plataformes que han sigut d'utilitat per aquest projecte.

4.2.3.1. Github

GitHub és una plataforma de desenvolupament de programari basada en web que permet als desenvolupadors col·laborar en projectes utilitzant el sistema de control de versions Git. Durant el procés de desenvolupament d'un projecte, s'efectuen innumerables modificacions al codi font mentre es desplega, el control de versions Git permet mantenir un registre de tots aquests canvis. Cada canvi és enregistrat mitjançant un commit, el qual rep un identificador únic que identifica una alteració o un conjunt de modificacions. Aquest registre de commits proporciona als desenvolupadors la capacitat de revisar el codi font en un estat anterior a un commit determinat, i fins i tot de restaurar-lo si és necessari. Això resulta útil per al seguiment i la gestió del codi durant el desenvolupament del projecte, permetent una col·laboració eficient i facilitant la recuperació de versions anteriors del codi font en cas de necessitat.

GitHub, també incorpora una sèrie d'eines a part del control de codi com per exemple github projects, que entre altres coses, permet incorporar una taula Kanban.

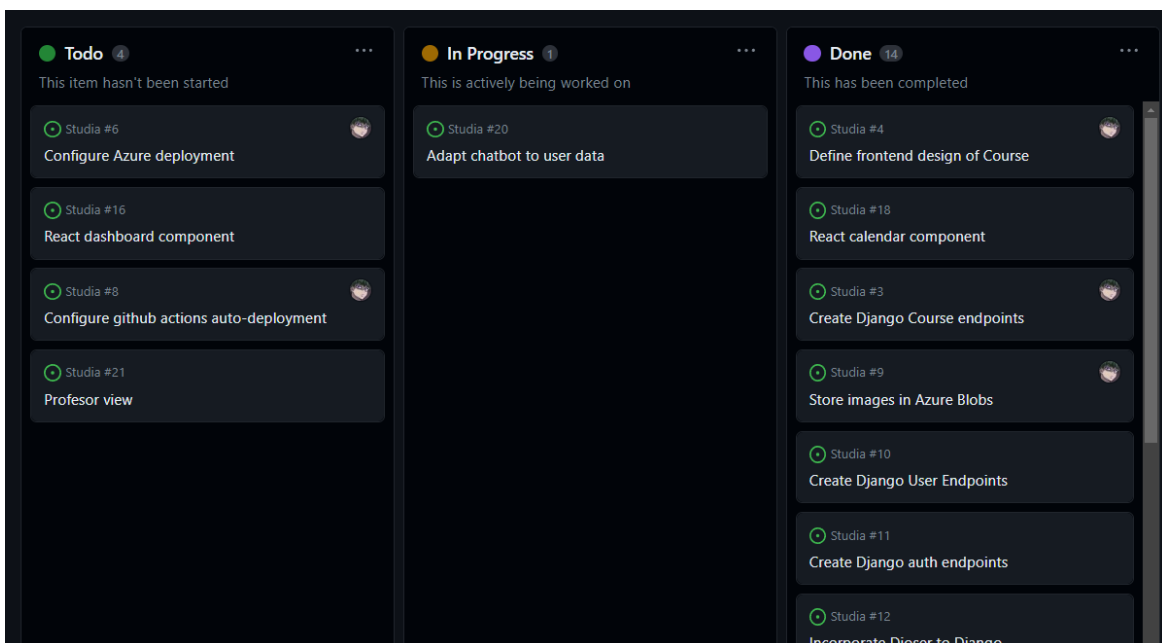


Fig. 11. Kanban de Studia.

En la **Fig. 11**, podem observar un overview de la feina general, tenim algunes tasques encara pendents, algunes en progrés i finalment les tasques que ja hem completat.

GitHub també incorpora l'opció de fer CI/CD (Continuous Deployment and Continuous Integration), de la qual parlarem més en el següent apartat.

Finalment, per realitzar un correcte ús de les branches de git, he utilitzat Gitflow adaptat al scope del meu treball. En el projecte, existeix una branca 'master' on només es pot fer un push al

final d'una entrega i una branca 'development' que s'actualitza constantment amb les noves features. Com que en aquest projecte he estat treballant sol, no té gaire sentit crear una branca per cada feature, així que he estat treballant directament sota la branca de 'development', el flux es pot visualitzar en la **Fig. 12**

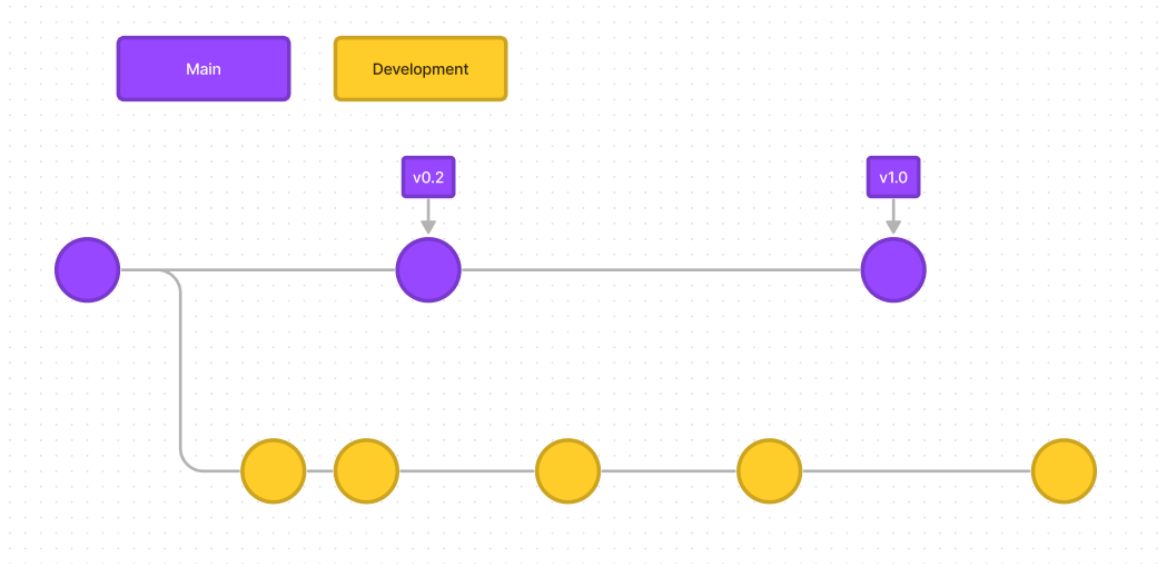


Fig. 12. Gitflow utilitzant dos branques.

4.2.3.2. Figma

Per l'apartat de dissenyar diagrames, wireframes, prototips i tot en general, he utilitzat la plataforma de Figma. Aquesta és un editor de gràfics vectorials que permet de forma senzilla fer des de wireframes fins a prototips d'alta fidelitat.

Anteriorment ja l'he utilitzat per a altres projectes i és una eina molt potent, per això l'he utilitzat també per aquest projecte.

4.3. Wireframes

Els wireframes són representacions visuals esquemàtiques que capturen l'estructura, la disposició i les funcionalitats clau de la pàgina sense entrar en detalls de disseny visual. Aquesta fase inicial del procés de disseny permet establir les bases per a la posterior implementació i millorar la comunicació entre els membres de l'equip de desenvolupament i els stakeholders.

En aquest apartat, exploraré com els wireframes han contribuït a l'elaboració del meu disseny de pàgina web, analitzant la seva importància en l'organització del contingut, l'accessibilitat i la interacció de l'usuari amb la pàgina.

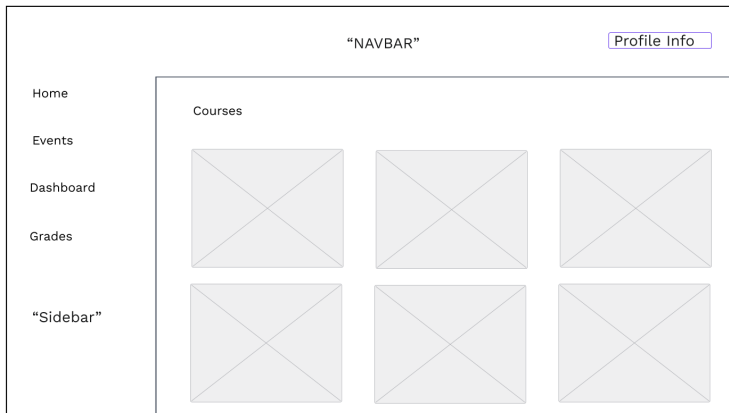


Fig. 13.1 Home Page

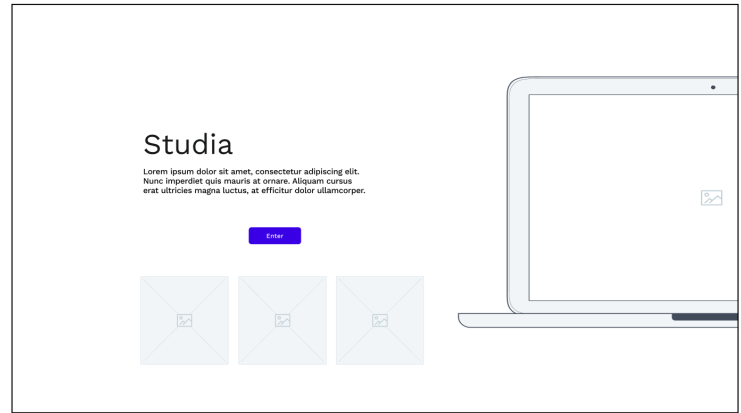


Fig. 13.2 Landing Page

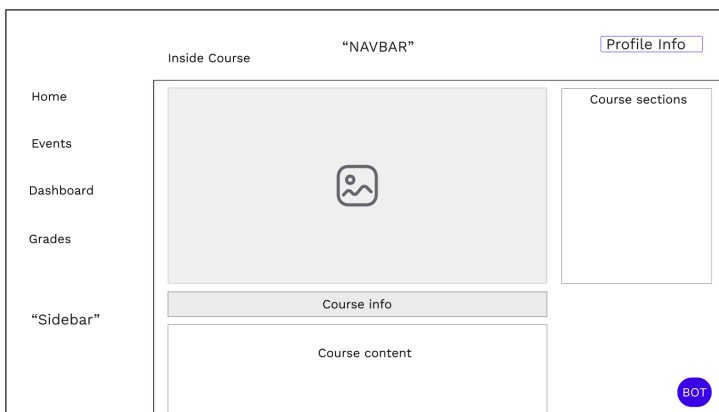


Fig. 13.3 Course Page

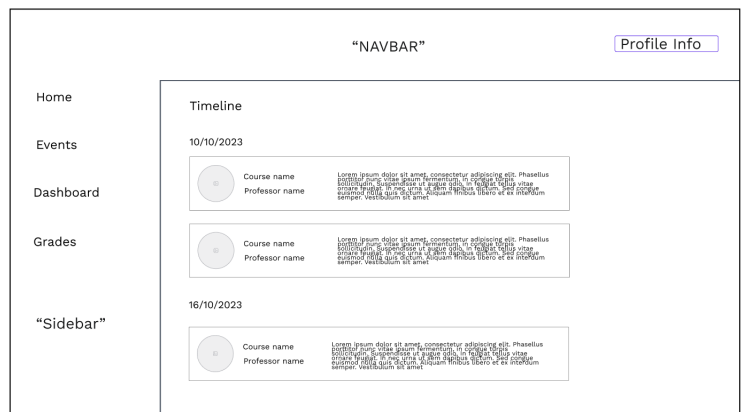


Fig. 13.4 Events Page

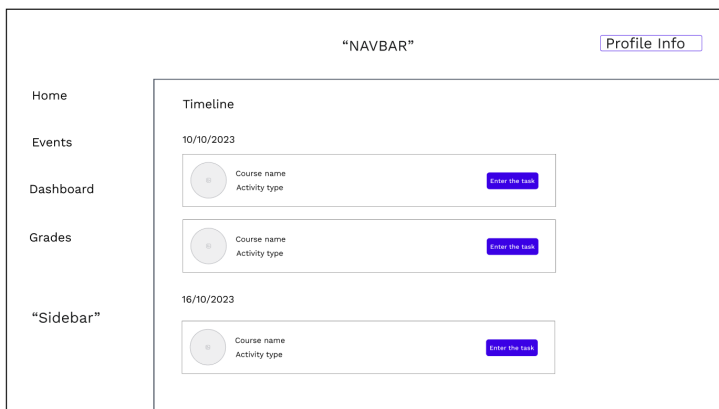


Fig. 13.5 Timeline Page

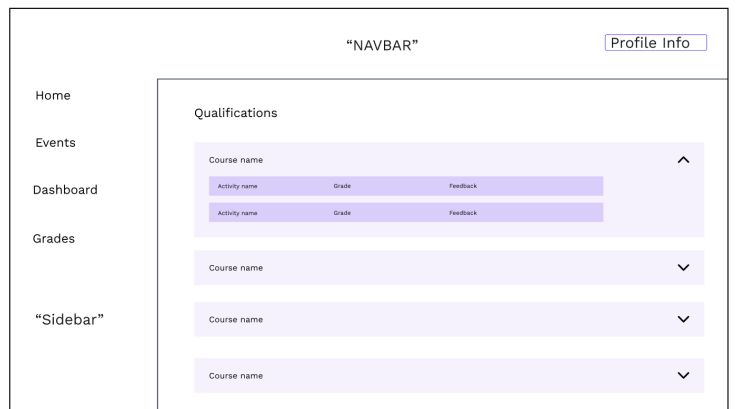


Fig. 13.6 Qualifications Page

En les **Fig. 13.***, podem veure la primera iteració sobre els wireframes de la aplicació web, pel que fa a les pàgines principals he optat per un disseny central format per un layout amb una sidebar a la part esquerra, com es pot veure en la **Fig.13.1**. Aquesta sidebar permet accedir a les diferents pàgines principals de la pàgina "Home" (**Fig. 13.1**), "Events" (**Fig. 13.4** i **Fig. 13.5**), "Dashboard" (no definit en aquesta primera iteració de wireframes), i Grades (**Fig. 13.6**).

Les decisions de disseny en les que m'he basat són les següents:

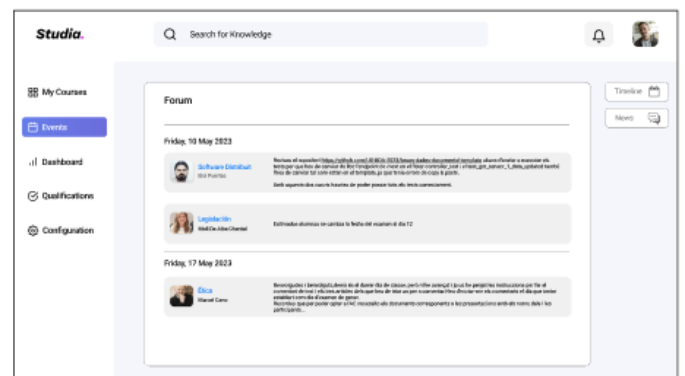
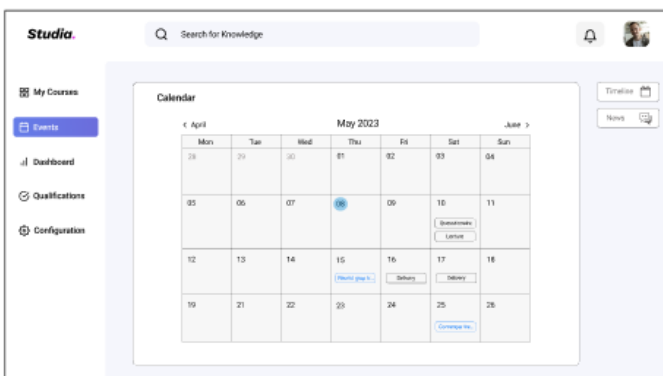
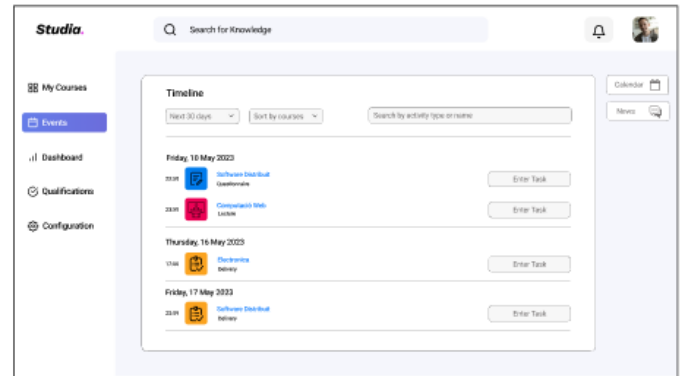
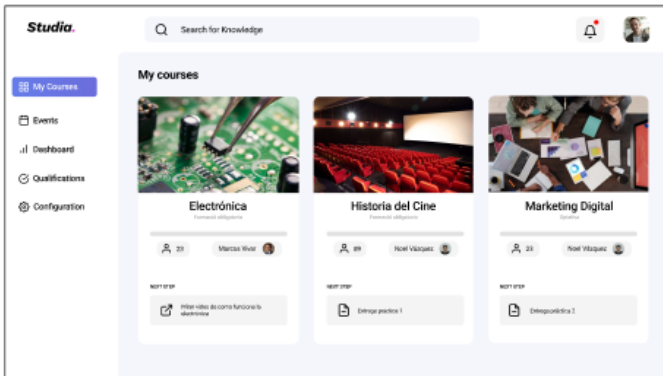
- **Disseny centrat en la usabilitat:** La inclusió d'un sidebar lateral i una navbar pot facilitar la navegació per als usuaris, ja que els proporciona un accés ràpid i visible a les pàgines importants i la informació rellevant.
- **Jerarquia de contingut:** El sidebar lateral s'utilitza comunament per mostrar una jerarquia clara de contingut. En agrupar les pàgines importants en el sidebar, es pot donar prioritat a aquestes pàgines i permetre als usuaris navegar ràpidament a través d'elles.
- **Disseny responsive:** És important tenir en compte que el disseny de la pàgina web s'ha d'adaptar a diferents dispositius i mides de pantalla. Un sidebar lateral pot ser una solució eficaç per mostrar opcions de navegació en pantalles més grans, mentre que en pantalles més petites, es pot transformar en un menú desplegable o es pot ocultar i mostrar segons sigui necessari.
- **Identitat de marca:** La navbar amb el logo i la informació de l'usuari pot ajudar a reforçar la identitat de marca de la pàgina web. En col·locar el logo a la navbar, es pot crear una associació visual instantània amb la marca, la qual cosa ajuda a reforçar el reconeixement i la coherència en tot el lloc.
- **Estètica i equilibri visual:** Un sidebar lateral i una navbar ben dissenyats poden ajudar a equilibrar visualment la pàgina web i afegir un component estètic atractiu. En triar colors, tipografies i espaiat adequats, es pot aconseguir un disseny harmoniós i agradable a la vista com podem observar en el següent apartat de prototips.

4.4. Prototips

Els prototips d'alta fidelitat són representacions visuals detallades de la nostra pàgina web, capturant el disseny i l'estructura del producte final. Tot i que no són interactius, són una eina valuosa

per comunicar i avaluar el disseny abans de la implementació. Ens permeten prendre decisions informades, presentar una representació atractiva i obtenir feedback dels stakeholders, també serveixen com a guia per als desenvolupadors i establen una base visual per al desenvolupament.

En aquest apartat visualitzarem els diferents prototips que hem realitzat per la nostra aplicació.



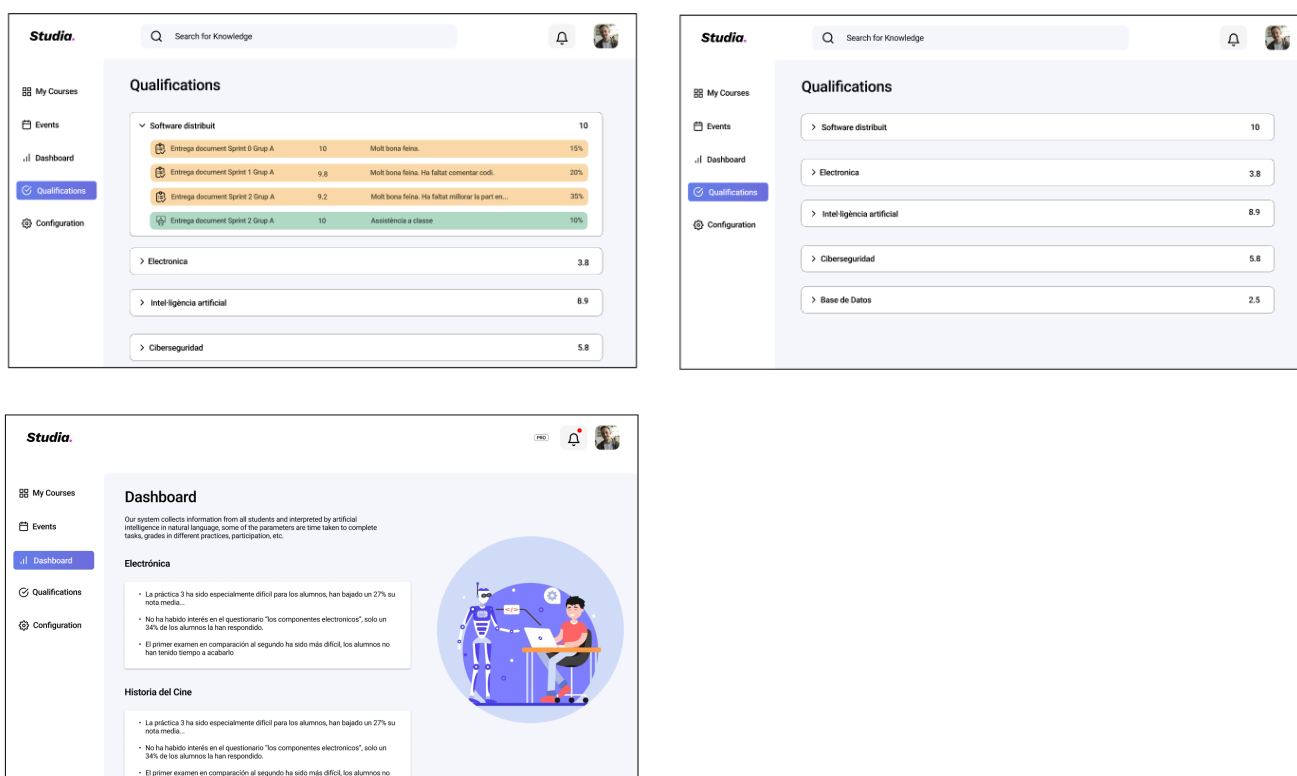


Fig. 14. Prototips d'alta fidelitat des de la vista de un estudiant

En la **Fig. 14** podem veure el conjunt de tots els prototips de la vista d'estudiant en la nostra pàgina, observem que tots els dissenys giren al voltant d'una paleta de colors violeta, aquest és sovint associat amb la creativitat, la imaginació i la intuïció. Utilitzar aquest color com a color principal en una pàgina de e-learning pot transmetre una sensació de pensament creatiu i inspiració, que poden ser valors importants en un entorn d'aprenentatge. Pel que fa a les icones de l'aplicació, hem utilitzat la llibreria de react-icons [7] que ens permet inserir-los en la nostra aplicació de React de manera senzilla amb una gran varietat d'icones.

Hem mantingut el mateix layout mencionat en els wireframes, però a més a més trobem el logo a la part superior esquerra en el navbar, acompanyat del perfil d'usuari, notificaciones i una search bar per filtrar els cursos.

Per la part dels cursos, també mantenim layout dels wireframes, on trobem en el centre tota la informació dels apartats incloent les activitats que el conformen, i a la dreta una navigation bar en forma de desplegable per on podem anar als diferents apartats de forma ràpida, és important mencionar que també podem observar a quin tipus de fase de l'aprenentatge autoregulat es troba cada apartat simulant una seqüència lineal. Els tres tipus de fase han sigut representats amb els colors; verd (Forethought), groc (Performance) i finalment vermell (Self-Reflection), així l'estudiant en

tot moment és conscient de l'objectiu de l'apartat. En aquests frames dels cursos, l'estudiant també tindrà accés al chatbot amb un popup que es troba a la part inferior dreta.

Finalment, el Dashboard tindrà informació resumida de com l'estudiant està avançant en els diferents cursos, en un principi l'objectiu és traduir les dades que recopilem a la pàgina a llenguatge natural mitjançant la implementació de la mateixa API que utilitzem en el chatbot.

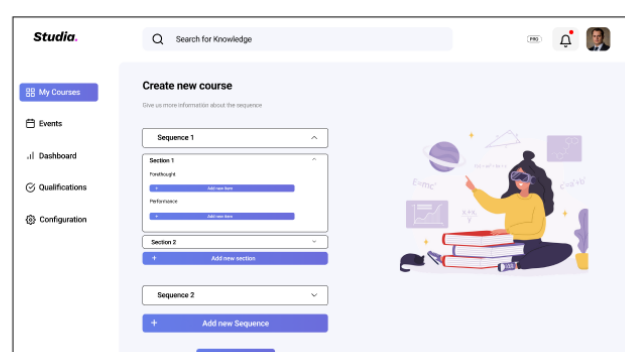
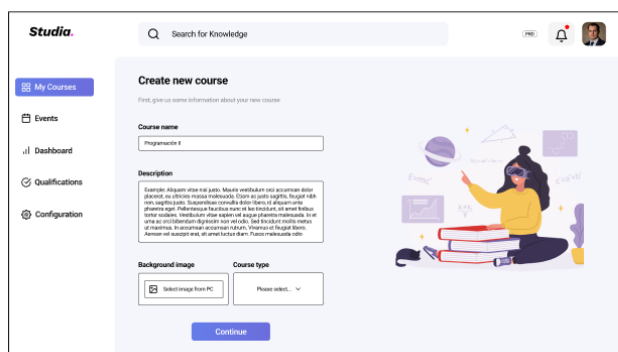
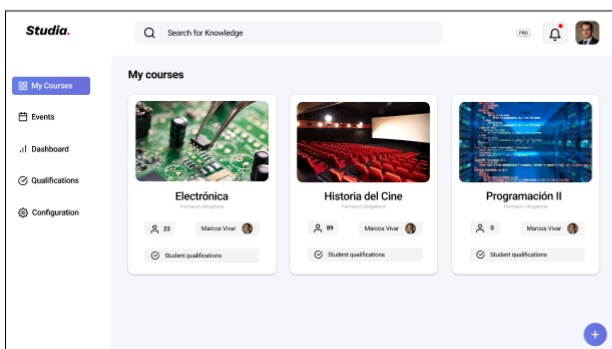
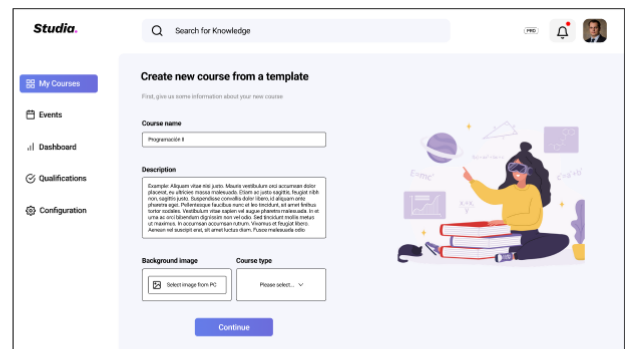
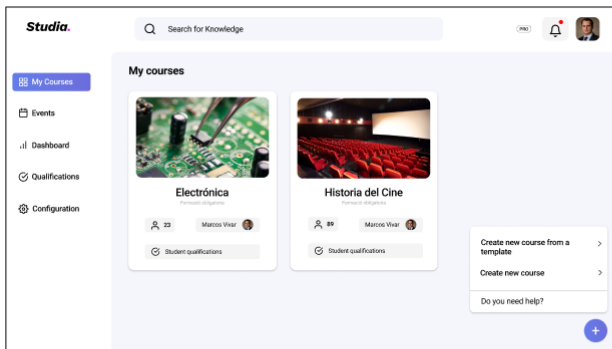


Fig. 15. Prototips d'alta fidelitat des de la vista d'un administrador

Com ja hem comentat anteriorment, l'administrador és l'encarregat de crear els cursos que poden visualitzar tant els estudiants com els professors en la nostra aplicació, en la **Fig. 15**, veiem com podria fer un administrador per crear un curs, tenint l'opció tant des d'una template com des de zero, com hem fet en els anteriors dissenys busquem sempre que sigui intuïtiu i fàcil per qualsevol usuari.

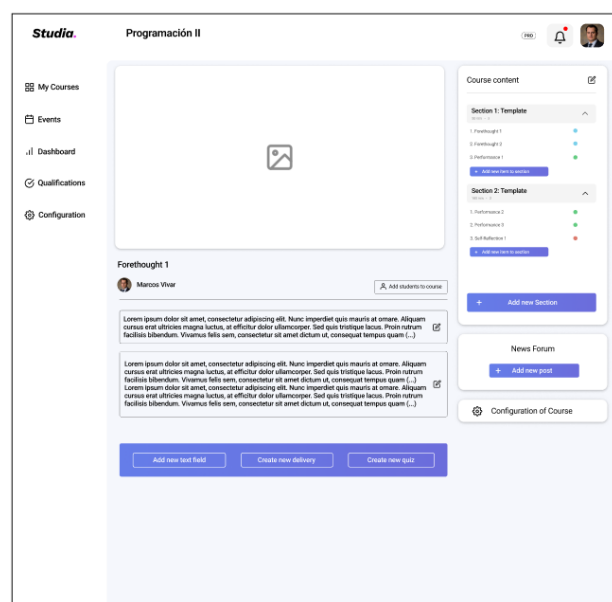
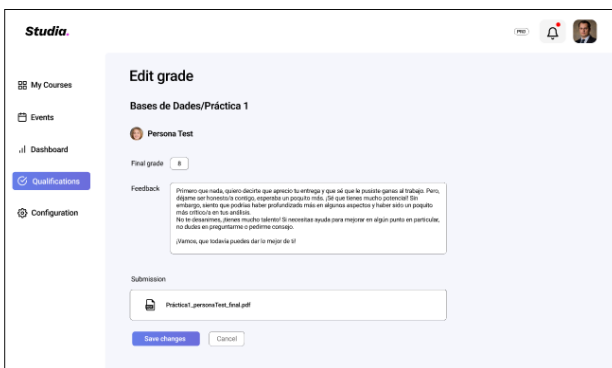
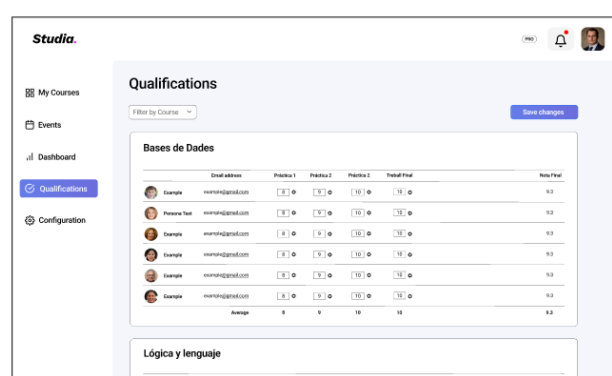
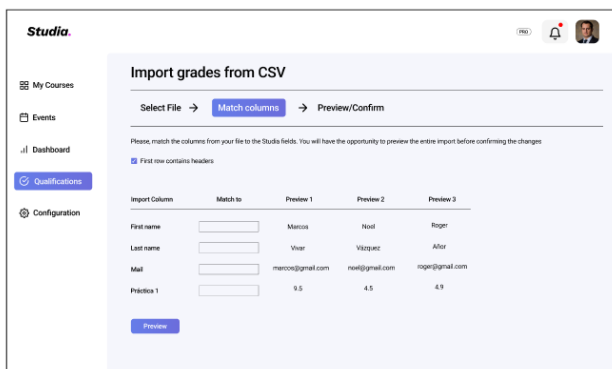
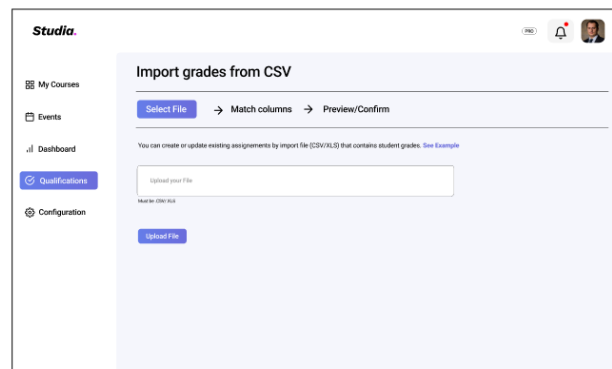
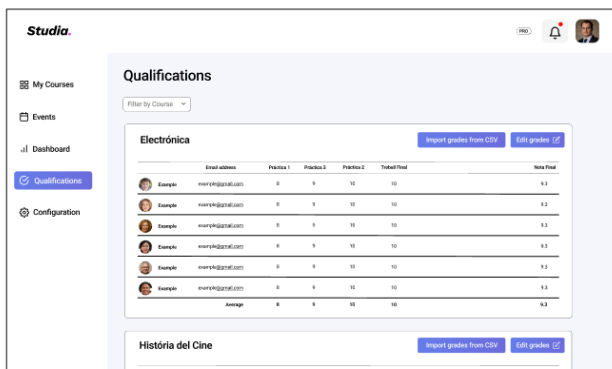


Fig. 16. Prototips d'alta fidelitat des del professor

Un dels requisits com a professor és que existeixi un sistema per pujar automàticament les notes a partir d'un fitxer excel, aquest sistema el podem veure representat en la Fig. 16, on podem trobar també com podria un professor consultar les qualificacions dels seus cursos, editar les notes i

finalment editar tota la informació del curs. En el mateix feed d'editar un curs, el professor podrà agregar també un post de notícia com els que hem vist i mencionat anteriorment.

5. Implementació i resultats

En aquest apartat explorarem la implementació d'alguns dels diferents requeriments o mòduls que podem trobar en l'aplicació web, també profunditzarem en els serveis de tercers que hem utilitzat i finalitzarem amb com hem estructurat el deployment de Studia.

5.1. Crides a API des de React

Abans de veure com consumir APIs, és important entendre que consumir APIs en React és molt diferent de com es fa en JavaScript. Això és perquè aquestes peticions ara es fan dins d'un Component de React, en el nostre cas, farem servir components funcionals, el que significa que necessitem utilitzar dos dels principals React Hooks [\[2\]](#) [\[14\]](#):

- **useEffect Hook:** A React, realitzem les peticions a l'API dins del hook `useEffect()`. Aquest es renderitza immediatament quan l'aplicació es construeix o després d'arribar a un estat específic.
- **useState Hook:** Quan fem una petició de dades, hem de preparar un estat on les dades es guardaran quan es retornin. Podem guardar-les en una eina de gestió d'estats com Redux o en un objecte de context.

A continuació, es presenta un exemple de com consumir una REST API utilitzant aquestes eines:

```
function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetch('https://api.example.com/data');
        const jsonData = await response.json();
        setData(jsonData);
      } catch (error) {
        console.error('Error en obtenir les dades:', error);
      }
    };

    fetchData();
  });
}
```

```
}, []);  
}
```

Fig. 17. Pseudocodi per consumir una API en React

En l'anterior **Fig. 17** s'utilitza el hook `useEffect` per a realitzar una crida a l'API quan el component s'ha creat. Dins de la funció `fetchData`, s'usa la funció `fetch()` per a fer una crida GET a l'URL de l'API desitjada ('<https://api.example.com/data>' en aquest cas). Un cop s'obté la resposta, s'utilitza la funció `json()` per convertir-la en format JSON. Les dades obtingudes es guarden a l'estat local utilitzant la funció `setData`.

5.2. Serveis externs

Les APIs de tercers són desenvolupades per les empreses o proveïdors per permetre la integració amb altres aplicacions o serveis, aquestes proporcionen un conjunt d'instruccions i funcions que permeten als desenvolupadors accedir, interactuar i utilitzar les funcionalitats d'aquest servei o aplicació externa. Poden incloure accions com enviament i recepció de dades, l'accés a bases de dades, l'autenticació d'usuaris o altres funcionalitats específiques proporcionades per aquest servei en particular.

En aquest apartat, comentarem quines d'aquestes eines hem aprofitat, com les hem implementat i el resultat final.

5.2.1. OpenAI API

Per realitzar la implementació de Chat GPT [8] dins de Studia, hem tingut que utilitzar lògicament la API que ens proporciona OpenAI que es la empresa que està darrere de Chat GPT.

OpenAI ofereix diferents models de chatGPT, incloent GPT 4, GPT 3, GPT 3.5, GPT 3.5 Turbo, sent aquest últim el que hem utilitzat en la nostra aplicació. Tots aquests serveis no son gratuïts, pero el seu cost és exageradament baix, prenent de referència que 1000 tokens amb GPT 3.5 Turbo tenen un cost aproximat de 0.001 euros en un ús normal d'un estudiant probablement mai arribi a utilitzar ni dos cèntims d'euro.

La seva implantació és molt senzilla [18], només hem d'enviar un POST a aquest URL: <https://api.openai.com/v1/chat/completions> amb un body format pel model que volem utilitzar, el missatge enviat i l'historial de missatges com podem veure en la **Fig. 18**.

```
const apiRequestBody = {  
  "model": "gpt-3.5-turbo",  
  "messages": [  
    {  
      "role": "user",  
      "content": "Hello!"  
    }  
  ]  
}
```

```

    systemMessage, // cua de missatges
    ...apiMessages // missatges que envien els estudiants
]

```

Fig. 18. RequestBody per la API de OpenAI

Aquesta request, ens retorna un JSON amb diferent informació, el més interessant és veure quants tokens hem tingut que utilitzar i la llista "choices" que conté la resposta que ens ha donat el BOT com podem veure en la **Fig. 19**.

```

{
  "id": "chatcmpl-abc123",
  "object": "chat.completion",
  "created": 1677858242,
  "model": "gpt-3.5-turbo-0301",
  "usage": {
    "prompt_tokens": 13,
    "completion_tokens": 7,
    "total_tokens": 20
  },
  "choices": [
    {
      "message": {
        "role": "assistant",
        "content": "\n\nThis is a test!"
      },
      "finish_reason": "stop",
      "index": 0
    }
  ]
}

```

Fig. 19. Response donat per la API de OpenAI

5.2.2. Google Forms API

En aquest apartat, es descriurà com s'ha utilitzat l'API de Google Forms per a la creació dels qüestionaris en la nostra aplicació. A través de d'aquesta s'ha pogut implementar una funcionalitat que permet als professors generar i personalitzar qüestionaris de forma dinàmica per als estudiants.

Per començar a utilitzar l'API de Google Forms en el nostre projecte, s'ha requerit realitzar una configuració prèvia per a l'accés a l'API a través de la Consola de Desenvolupadors de Google,

aquesta configuració inclou la creació d'un compte de servei i l'obtenció d'un fitxer de credencials en format JSON.

Un cop s'ha realitzat la configuració, s'ha implementat el següent codi per a la creació dels qüestionaris utilitzant l'API de Google Forms:

```
credentials = service_account.Credentials.from_service_account_file(
    'credencials.json',    scopes=['https://www.googleapis.com/auth/forms']
)
# Autenticació amb les credencials
credentials.refresh(Request())

# Creació d'una instància del client de l'API de Google Forms
service = build('forms', 'v1', credentials=credentials)
# Creació del qüestionari
form = service.forms().create(
    body={
        'title': 'Títol del qüestionari',
        'description': 'Descripció del qüestionari',
        'questions': [
            {
                'type': 'multipleChoice',
                'question': 'Pregunta 1',
                'choices': [
                    {'text': 'Opció 1'},
                    {'text': 'Opció 2'},
                ]
            },
            {
                'type': 'text',
                'question': 'Pregunta 2'
            }
        ]
    }
).execute()
# Obtenció de l'enllaç per accedir al qüestionari
form_url = form['response']['formUrl']

# Utilització de l'enllaç per mostrar el qüestionari als estudiants
print(f'Enllaç del qüestionari: {form_url}')
```

Fig. 20. Pseudocodi creació d'un qüestionari

En la **Fig. 20**, podem veure un pseudocodi on s'han configurat les credencials del compte de servei i s'ha establert la connexió amb l'API de Google Forms. A continuació, s'ha creat un qüestionari amb un títol i una descripció específics, així com preguntes personalitzades, un cop el qüestionari ha estat creat, s'ha obtingut l'enllaç que permet als estudiants accedir-hi. Com podem veure, és bastant senzill utilitzar aquesta API i el més important és que ens facilita molt la feina de cara al backend.

5.3. Devops Deployment

En aquest apartat, es descriurà el procés de desplegament de l'aplicació utilitzant Docker i Docker Compose, així com la seva possible posterior implementació en un servei de contenidors com Azure.

5.3.1 Docker

Docker és una plataforma de virtualització a nivell d'aplicació que permet empaçar i distribuir aplicacions juntament amb les seves dependències en forma de contenidors. Un contenidor Docker és una unitat lleugera i independent que conté tot el necessari per executar una aplicació, inclòs el codi, les llibreries i les configuracions. Aquesta tecnologia ofereix una manera eficient i reproduïble de distribuir i executar aplicacions, eliminant els problemes d'incompatibilitat i assegurant una alta portabilitat entre diferents entorns.

Per generar aquests contenidors, primer definim un Dockerfile, que és simplement un fitxer de text senzill amb un conjunt d'ordres o instruccions. Aquestes ordres/instruccions s'executen successivament per realitzar accions sobre la imatge base i crear una nova imatge de Docker. Fent un build d'aquest Dockerfile no hem generat encara el nostre contenidor, sinó que tenim una imatge d'aquest, finalment per poder crear el contenidor haurem d'utilitzar la comanda 'run' a la imatge.

En el nostre cas hem creat dos dockerfiles [\[3\]](#), que podrien representar dos dels nostres serveis, una imatge pel frontend **Fig. 21** i una imatge pel backend **Fig. 22**.

```
FROM python:3.10

RUN mkdir app

WORKDIR /app

RUN pip install --upgrade pip
```

```

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

COPY . /app

EXPOSE 8000

CMD python manage.py makemigrations && python manage.py migrate && python
manage.py runserver 0.0.0.0:8000

```

Fig. 21. Dockerfile del backend.

El Dockerfile del backend comença establint la imatge base com a Python 3.10. A continuació, es crea un directori anomenat "app" dins del contenidor i se'n fa el directori de treball actual.

Per assegurar-se de tenir la versió més recent de pip, s'executa la comanda `pip install --upgrade pip`. A continuació, es copia el fitxer `requirements.txt` al directori de treball del contenidor i s'instal·len les dependències necessàries per al projecte amb la comanda `pip install --no-cache-dir -r requirements.txt`.

Seguidament, es copia tot el contingut del backend al directori de treball del contenidor, aquest pas és important perquè el backend pugui funcionar correctament dins del contenidor. S'obre el port 8000 per permetre les connexions al servidor de Django, això permet que el backend sigui accessible des de l'exterior.

Finalment, s'executen els següents comandaments per a realitzar les migracions, establir la base de dades i engagar el servidor de Django: `python manage.py makemigrations`, `python manage.py migrate` i `python manage.py runserver 0.0.0.0:8000`.

```

FROM node:16
RUN mkdir app

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . /app

# Construir la aplicació de React

```

```
RUN npm run build  
  
# Especificar el comando de inicio para el contenedor  
CMD ["npm", "start"]
```

Fig. 22. Dockerfile del frontend.

Aquest docker configurat per la part de frontend comença establint una imatge de "node:16", que proporciona un entorn de treball Node.js preconfigurat. Aquesta imatge ja inclou tot el necessari per executar aplicacions Node.js, i tal com hem fet en l'anterior docker, hem creat un directori 'app' dins del nostre contenidor, aquest sera el directori de treball per tots els passos següents.

Els fitxers "package.json" i "package-lock.json" del projecte es copien al directori de treball del contenidor. Aquests fitxers contenen la informació de les dependències i els scripts necessaris per tal que l'aplicació funcioni correctament.

Un cop copiats els fitxers, s'executa la comanda "npm install" per instal·lar les dependències del projecte. Això assegura que totes les llibreries requerides estiguin disponibles per a l'aplicació, tot el contingut de l'aplicació, incloent-hi els fitxers del codi font, es copia al directori de treball del contenidor. A continuació, s'executa la comanda "npm run build" per construir l'aplicació de React i generar els fitxers del build.

Finalment, s'estableix el comandament de "npm start" com a comanda d'inici per al contenidor, això indicarà al contenidor que s'ha d'iniciar l'aplicació utilitzant el servidor web proporcionat per l'aplicació.

Amb aquests dos serveis de contenidors ja disponibles pel seu ús, només ens farà falta aixecar-los a l'hora i vincular-los amb la nostra bbdd relacional per poder començar a desplegar la nostra aplicació, per realitzar aquesta feina, utilitzarem la eina de Docker Compose.

5.3.2 Docker Compose

Docker Compose és una eina que permet definir i gestionar aplicacions multi-contenidor amb Docker. Amb Docker Compose, és possible definir tota l'arquitectura de l'aplicació utilitzant un fitxer YAML, que descriu els contenidors, les xarxes i altres configuracions relacionades. Això permet simplificar la gestió de múltiples contenidors i facilita el desplegament i l'execució de l'aplicació en diferents entorns.


```

version: '3'
services:
  db:
    image: postgres
    volumes:
      - db_data:/var/lib/postgresql/data
    expose:
      - 5432
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgresadmin
      POSTGRES_DB: postgres
    ports:
      - 5432:5432

  frontend:
    build:
      context: /studia_fe
      dockerfile: Dockerfile
    ports:
      - 3000:3000

  backend:
    build:
      context: /studia_be
      dockerfile: Dockerfile
    ports:
      - 8000:8000
    depends_on:
      - db

volumes:
  db_data:

```

Fig. 23. Docker-compose de Studia.

En aquest docker-compose de la **Fig. 23** hem definit tres serveis:

- **Servei "db"**: Per desplegar la base de dades PostgreSQL, s'utilitza la imatge de Docker "postgres". Aquesta imatge serveix com a base per crear un contenidor que contindrà la base de dades, en aquesta s'assigna un nom al volum anomenat "db_data" que s'utilitzarà per

emmagatzemar les dades de la base de dades, a més, es configura el contenidor per exposar el port 5432, que permetrà la connexió a la base de dades des de l'exterior, per a això, es configuren les variables d'entorn necessàries, com ara el nom d'usuari, la contrasenya i el nom de la base de dades de PostgreSQL.

- **Servei "frontend"**: Per desplegar l'aplicació frontend, es crea un contenidor utilitzant el fitxer Dockerfile de la Figura 14. Aquest fitxer conté les instruccions necessàries per construir l'entorn d'execució de l'aplicació. Una vegada creat el contenidor, es realitza un enllaç entre el port 3000 del contenidor i el port 3000 de l'usuari. Això permet accedir a l'aplicació frontend des del navegador utilitzant la direcció IP de l'amfitrió i el port especificat. D'aquesta manera, es facilita l'accés i la interacció amb l'aplicació frontend des de l'exterior.
- **Servei "backend"**: Per desplegar l'aplicació backend, es crea un contenidor utilitzant el fitxer Dockerfile de la Figura 13. Aquest fitxer conté les instruccions necessàries per construir l'entorn d'execució de l'aplicació backend. Una vegada creat el contenidor, es realitza un enllaç entre el port 8000 del contenidor i el port 8000 de l'usuari. A més, el servei backend depèn del servei "db", el qual fa que s'asseguri que el servei de la base de dades estigui en funcionament abans d'iniciar el backend, això garanteix que l'aplicació backend tingui accés a la base de dades i pugui realitzar les operacions necessàries.

Finalment, es defineix el volum "db_data" per emmagatzemar les dades de la base de dades PostgreSQL, en la següent figura podem veure com ha quedat l'esquema dels nostres contenidors, podem veure l'esquema final en la **Fig. 24**.

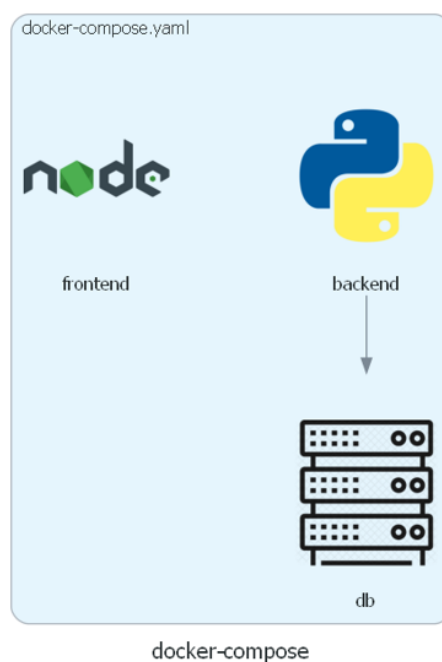


Fig. 24. Esquema del docker-compose de Studia.

Per fer el deployment de l'aplicació podríem utilitzar diversos serveis de núvol o proveïdors de serveis d'aplicacions, com per exemple Heroku, Digital Ocean o Azure. En aquest cas no ho he fet perquè crec que no forma part del scope d'aquest treball de fi de grau i tampoc tinc els recursos per finançar aquests serveis, però hem dut a terme tots els passos per poder desplegar l'aplicació en cas que es necessités.

5.4. Resultats Finals

Com hem comentat no hem fet un deployment amb un servei de cloud, però sí que hem fet un deployment funcional local mitjançant els dockers comentats en l'anterior apartat. En la [Fig. 1. Annexos](#) podem veure un conjunt d'imatges amb els resultats finals de la pàgina. Podem observar, han sigut bastant fidels als prototips finals que hem vist anteriorment, però hem adaptat algunes de les pantalles per fer un layout més responsive.

A més a més, trobem la documentació de la API final feta amb Django en la [Fig. 2. Annexos](#), [Fig. 3. Annexos](#) i [Fig. 4. Annexos](#), en aquest ordre sent la documentació de la API de les accounts, de cursos i finalment la de auth, on es guarda tota la informació de autenticació.

6. Conclusions i treball futur

L'objectiu principal d'aquest treball ha sigut dissenyar una aplicació web de principi a fi, des del disseny de l'arquitectura, fins al deployment de l'aplicació, en aquest sentit em puc sentir orgullós, ja que he aconseguit tot el que hem proposat amb el meu tutor al principi de tot el projecte.

Per la part del backend considero que he desenvolupat una REST API fàcilment escalable i dissenyada des d'un principi perquè sigui consistent i segura, aplicant totes les tècniques que he après en altres assignatures de la carrera, com bases de dades avançades. Gràcies a aquesta assignatura també, he pogut fer el disseny de l'arquitectura, que d'igual manera he tingut en compte que sigui escalable i fàcil de mantenir per pròxims treballs.

Pel que fa al front-end, s'ha dissenyat una interfície intuïtiva i moderna que s'aproxima molt a el que com a estudiants, ens agradaria tenir com a plataforma.

D'altra banda, no s'ha complert tots els requisits detallats en el backlog d'aquesta memòria i de cara a un futur treball s'haurien de desenvolupar els següents ítems:

- Frontend per afegir un estudiant a un curs com a rol d'administrador.
- Vista de professor per visualitzar, modificar i crear un curs.
- Sistema d'avaluació entre iguals.
- Entrenar el model del chatbot amb dades de l'usuari i del sistema.
- Millores de seguretat per accés de endpoints.
- Backend i frontend del dashboards pels estudiants i professors.
- Modificar arquitectura per guardar dashboards.
- Actualització a temps real de l'estat d'un curs per un estudiant.
- Frontend per la recuperació de contrasenya.
- Sistema per introduir notes automàticament com a rol de professor.
- Sistema per introduir afegir alumnes a un curs automàticament com a rol de professor.
- Frontend de l'administrador per eliminar cursos, crear cursos, editar cursos, etc.
- Frontend i backend per visualitzar els perfils d'usuari.
- Sistema d'entregues com a rol d'estudiant.
- Sistema de notificacions.

Finalment, per concloure aquest treball ha sigut molt interessant i sobretot divertir aprendre tant en aquesta àrea de web development, des d'un principi vaig tindre interès en aquest àmbit i amb aquest treball he après molt. A més, és realment satisfactori poder aplicar tota la teoria o coneixements previs que s'han vist en altres assignatures. Esperem que aquest projecte continuï endavant per acabar de complir totes les user stories i marquem un final en aquesta aplicació.

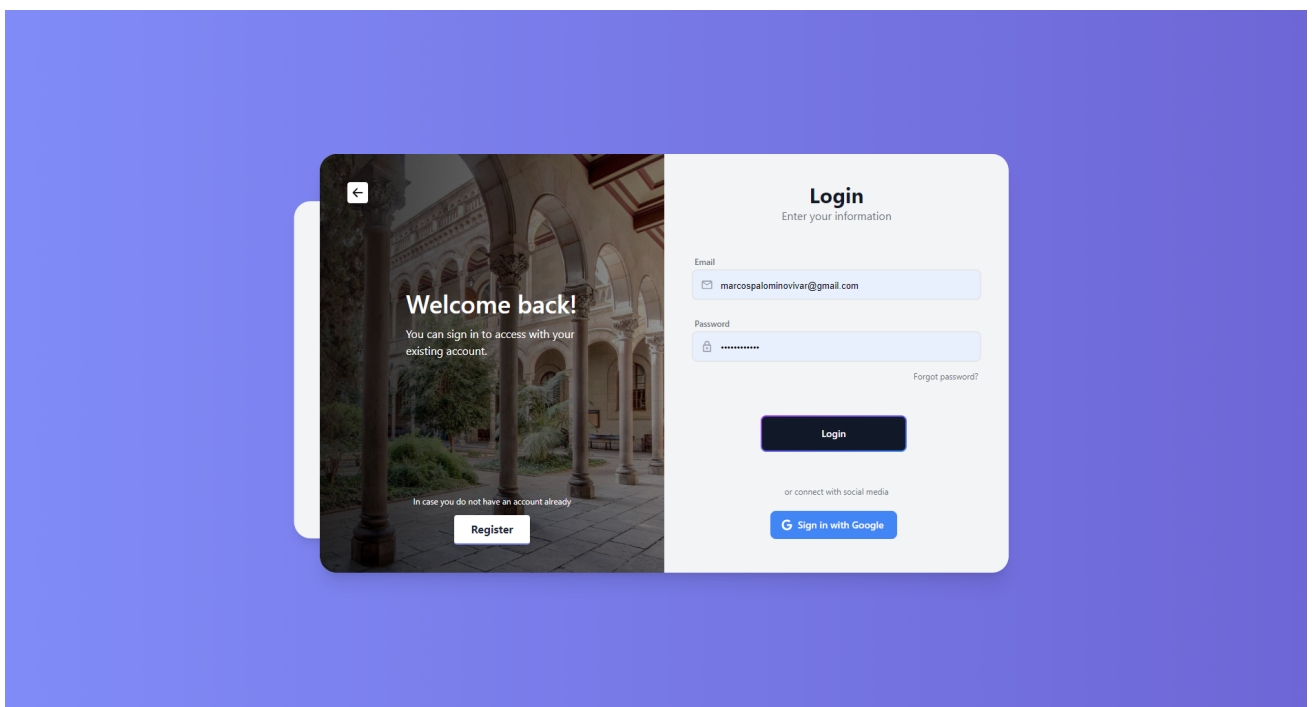
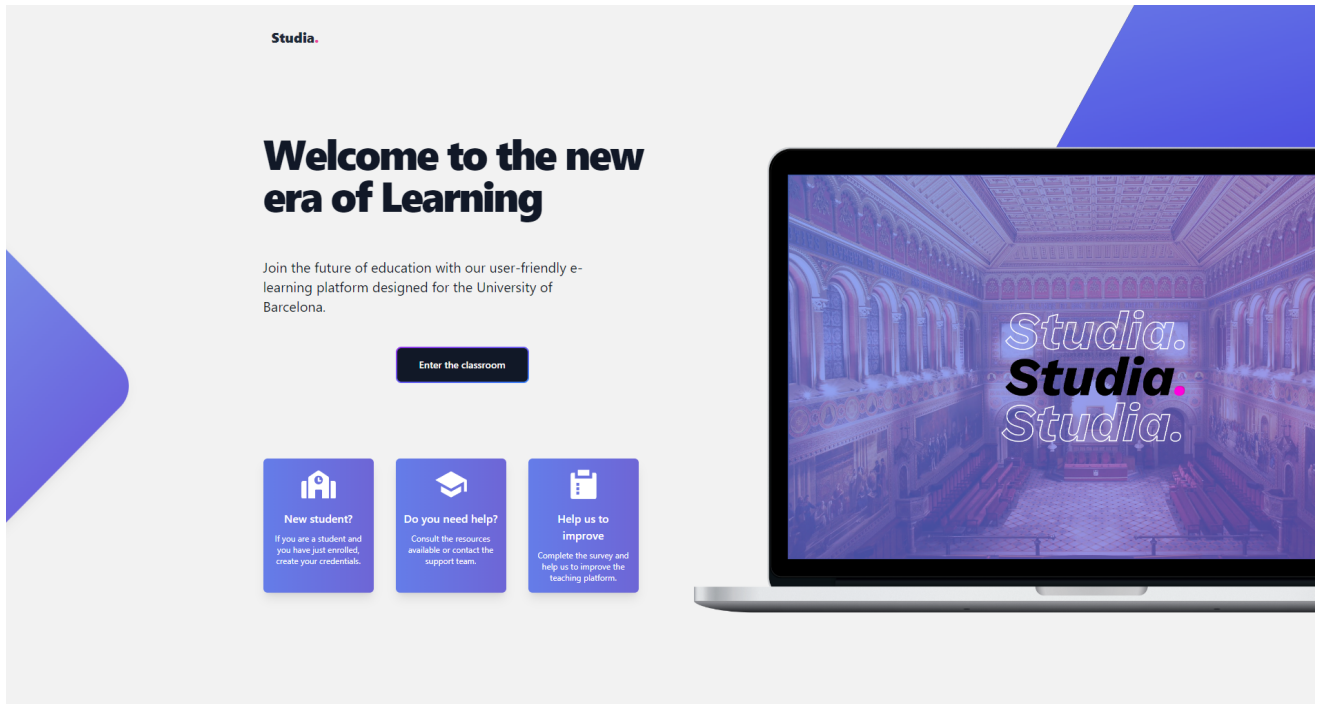
7. Bibliografia

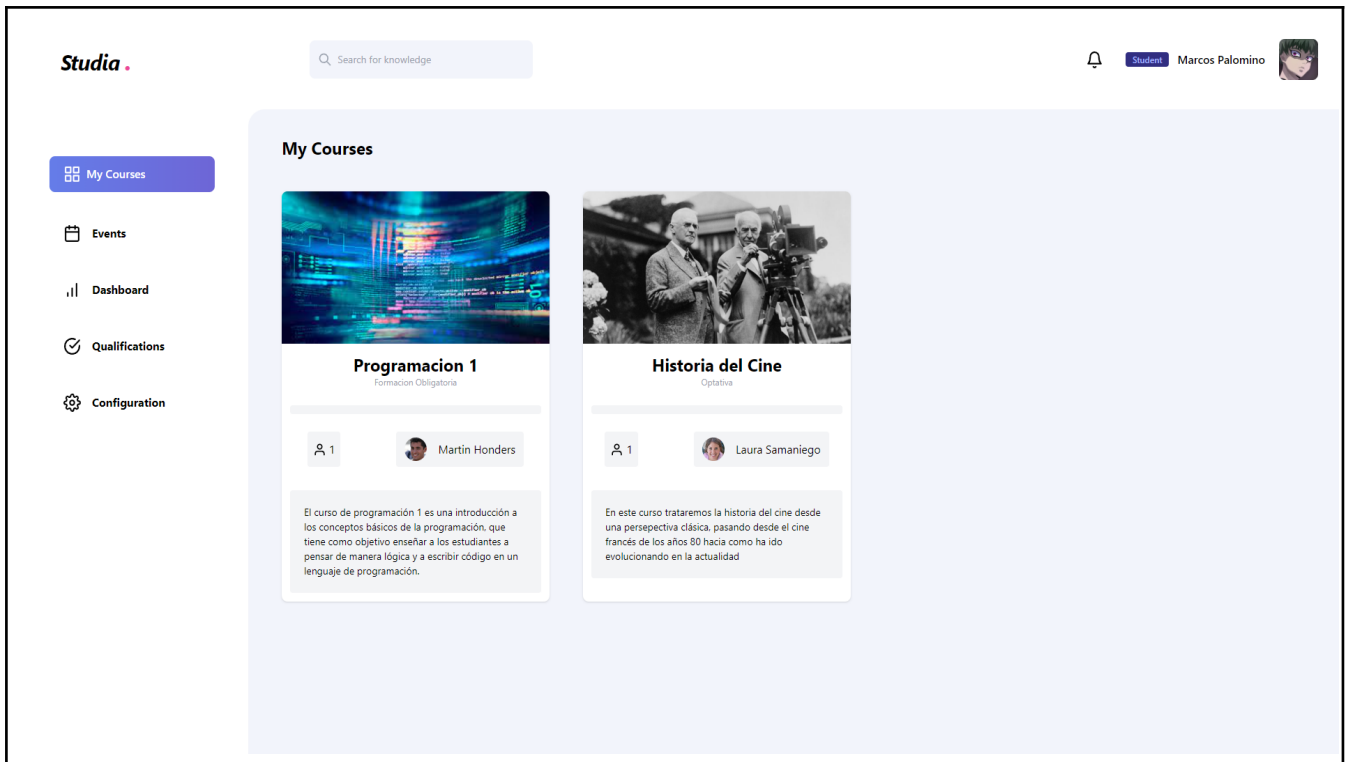
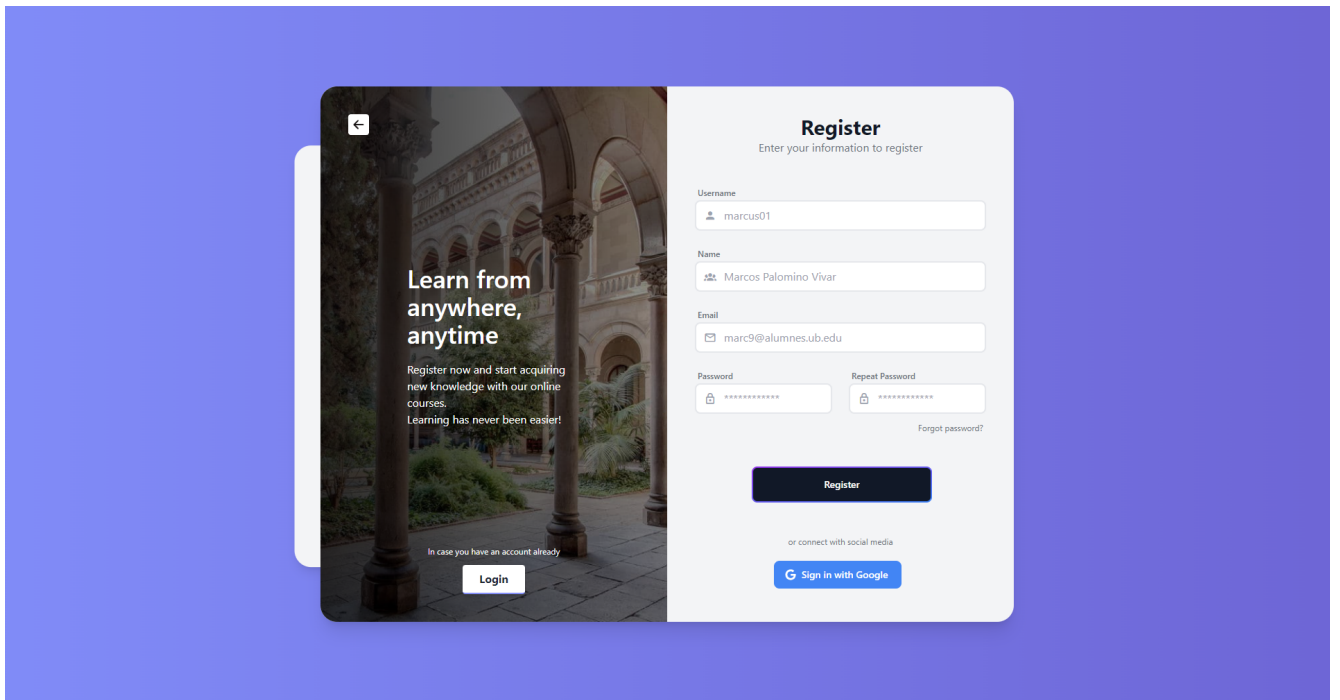
- [1] "Documentation - tailwind CSS," *Documentation - Tailwind CSS*. [Online]. Available: <https://v2.tailwindcss.com/docs>
- [2] G. Singhal, "Handle Ajax with React," *Pluralsight*, 05-Feb-2020. [Online]. Available: <https://www.pluralsight.com/guides/how-to-handle-ajax-with-react>
- [3] "Dockerizing a python django web application - semaphore tutorial," *Semaphore*, 29-Aug-2022. [Online]. Available: <https://semaphoreci.com/community/tutorials/dockerizing-a-python-django-web-application>
- [4] N. Sebastian, "Getting started with postgres in your react app," *LogRocket Blog*, 29-Oct-2021. [Online]. Available: <https://blog.logrocket.com/getting-started-with-postgres-in-your-react-app/>
- [5] "MongoDB documentation" "<https://www.mongodb.com/docs/>," *The MongoDB Team*.
- [6] *React*. [Online]. Available: <https://react.dev/>
- [7] *React icons*. [Online]. Available: <https://react-icons.github.io/react-icons/icons>
- [8] J. Kemper, "CHATGPT Guide: 10 prompt strategies for better output," *THE DECODER*, 08-Apr-2023. [Online]. Available: <https://the-decoder.com/chatgpt-guide-prompt-strategies/>
- [9] MozDevNet, "JavaScript," *MDN*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [10] MozDevNet, "CSS: Cascading style sheets," *MDN*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [11] MozDevNet, "HTML: Hypertext markup language," *MDN*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [12] "Welcome to Python.org," *Python.org*. [Online]. Available: <https://www.python.org/>.
- [13] D. Song, "Self-regulated learning. 3 phases," *Einbrain*, 11-Nov-2022. [Online]. Available: <https://einbrain.com/1773>.
- [14] "Built-in react hooks," *React*. [Online]. Available: <https://react.dev/reference/react>.
- [15] "Django," *Django Project*. [Online]. Available: <https://docs.djangoproject.com/en/4.2/>


- [16] “Redux Fundamentals, part 2: Concepts and data flow,” *Redux*, 30-Apr-2023. [Online]. Available: <https://redux.js.org/tutorials/fundamentals/part-2-concepts-data-flow>
- [17] “Welcome to Djoser’s documentation!¶,” *Welcome to djoser’s documentation! - djoser 2.2.0 documentation*. [Online]. Available: <https://djoser.readthedocs.io/en/latest/>
- [18] “Build a chatbot with the CHATGPT API in react (GPT-3.5-turbo tutorial),” *YouTube*, 02-Mar-2023. [Online]. Available: https://www.youtube.com/watch?v=Lag9Pj_33hM&ab_channel=CooperCodes
- [19] Lluch Molins, L., & Cano García, E. (2023). “How to Embed SRL in Online Learning Settings? Design Through Learning Analytics and Personalized Learning Design in Moodle”. *Journal of New Approaches in Educational, Research*, 12(1), 120-138. doi: 10.7821/naer.2023.1.1127
- [20] Héctor Lamas Rojas “Aprendizaje autorregulado, motivación y rendimiento académico” http://www.scielo.org.pe/scielo.php?script=sci_arttext&pid=S1729-48272008000100003
- [21] Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice*, 41(2), 64-70. doi:10.1207/s15430421tip4102_2
- [22] Elena Cano “Plataforma E-Learning para la Autorregulación del Aprendizaje” PDC2022-133248-100 Subdivisión de Planificación y Gestión Administrativa, Agencia Estatal de Investigación

8. Annexos

Resultats finals de la pàgina





Studia Search for knowledge Student Marcos Palomino 


- My Courses
- Events
- Dashboard
- Qualifications
- Configuration

Timeline

Next 90 days
Sort by Courses

Search by activity type or name →


10-07-2023



Historia del Cine

Planifica el trabajo 'efectos de camara'


21-07-2023



Programacion 1

Primera entrega tarea 2


23-07-2023



Historia del Cine


Primera entrega 'efectos de camara'


28-07-2023




Programacion 1

peer review geometria en el espacio

Calendar 

News 

Studia Search for knowledge Student Marcos Palomino 


- My Courses
- Events
- Dashboard
- Qualifications
- Configuration


Calendar

Add event

< May 2023 >
Today

Su	Mo	Tu	We	Th	Fr	Sa
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Timeline 

News 

The screenshot shows the Studia interface with a sidebar on the left containing 'My Courses', 'Events', 'Dashboard', 'Qualifications', and 'Configuration'. The main area displays a calendar for May 2023. An 'Add Event' modal is open, showing fields for 'Title' and 'Date' (with a date picker icon and a placeholder 'dd/mm/aaaa'). The modal has 'Cancel' and 'Add Event' buttons. On the right side of the calendar, there are buttons for 'Add event', 'Timeline', and 'News'.

The screenshot shows the Studia interface with the 'News' section active. The sidebar on the left is the same as in the first screenshot. The main area displays a 'News' section with four items, each with a profile picture, a title, and a subtitle:

- Programacion 1** (Martin Honders): El examen de hoy será en la aula B6.
- Programacion 1** (Martin Honders): Recordad hacer todas las actividades antes de ir al examen.
- Historia del Cine** (Laura Samaniego): Se cancela la clase de hoy debido a temas personales, mis disculpas.
- Historia del Cine** (Laura Samaniego): Recordad que el trabajo que tenéis que entregar tiene que tener un formato de pdf!

On the right side of the news section, there are buttons for 'Calendar' and 'Timeline'.

Studia Search for knowledge Student Marcos Palomino

- My Courses
- Events
- Dashboard
- Qualifications**
- Configuration

Qualifications

^ Programacion 1


Trabajo de Producto Cartesiano	6.5	Buena presentación, falta profundidad en al;	25%
Peer Review trabajo Producto Cartesiano	8	Muy buen trabajo	15%

^ Historia del Cine

Trabajo de Producto Cartesiano	9	Felicidades, muy buen trabajo	15%
Trabajo de Producto Cartesiano	4	Esperaba más de ti, mejorable	15%

Studia Programacion 1 Student Marcos Palomino

- My Courses**
- Events
- Dashboard
- Qualifications
- Configuration



Course content

- Sección 1: Algebra Relacional
- Sección 2: Geometria

MSLQ or DLSQ self-regulation questionnaire

[Participants](#)

El objetivo de esta encuesta es comprender mejor cómo afronta usted su trabajo de curso y qué factores le motivan para aprender y tener éxito académicamente.

El cuestionario se basa en el Motivated Strategies for Learning Questionnaire (MSLQ) o en el Deep Learning Survey Questionnaire (DLSQ), ambos ampliamente utilizados en la investigación educativa para evaluar las estrategias de aprendizaje y la motivación de los estudiantes.

Su participación es voluntaria y anónima, y sus respuestas serán confidenciales. Por favor, responda a cada pregunta honestamente y lo mejor que pueda.

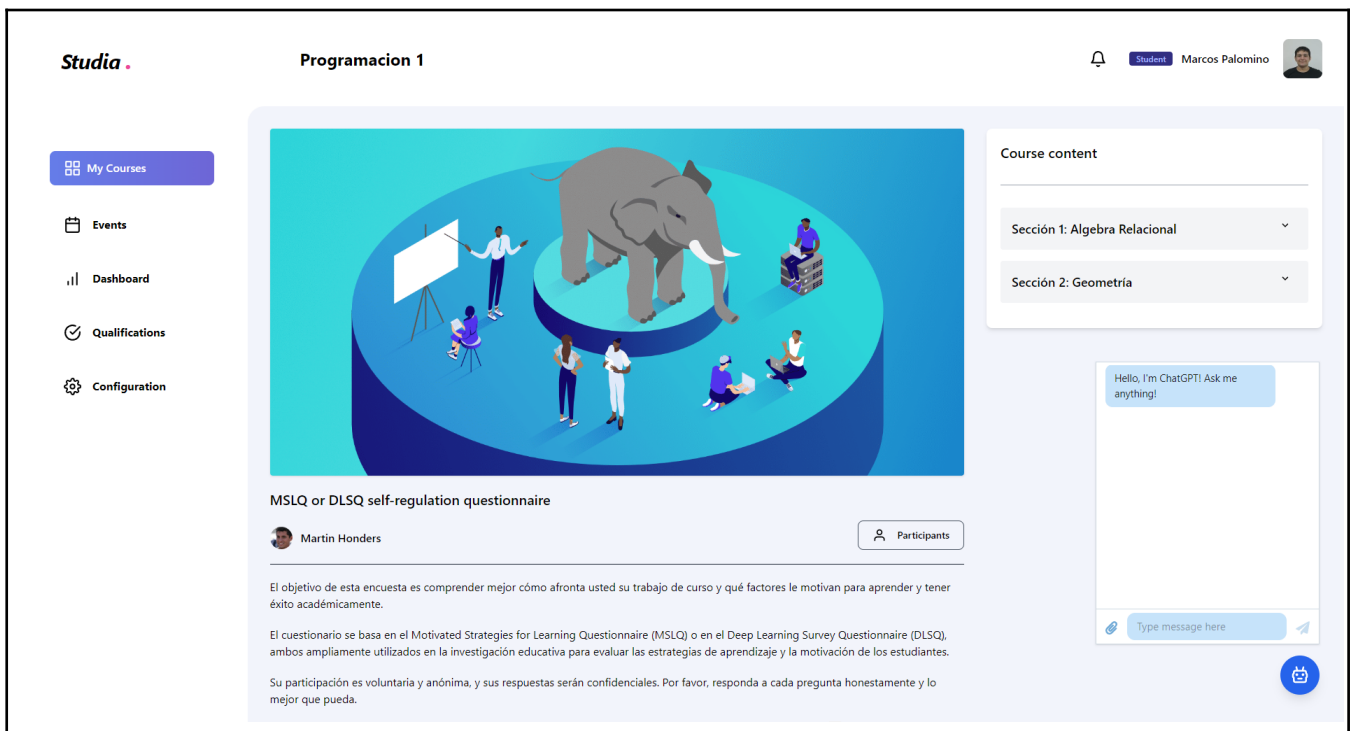
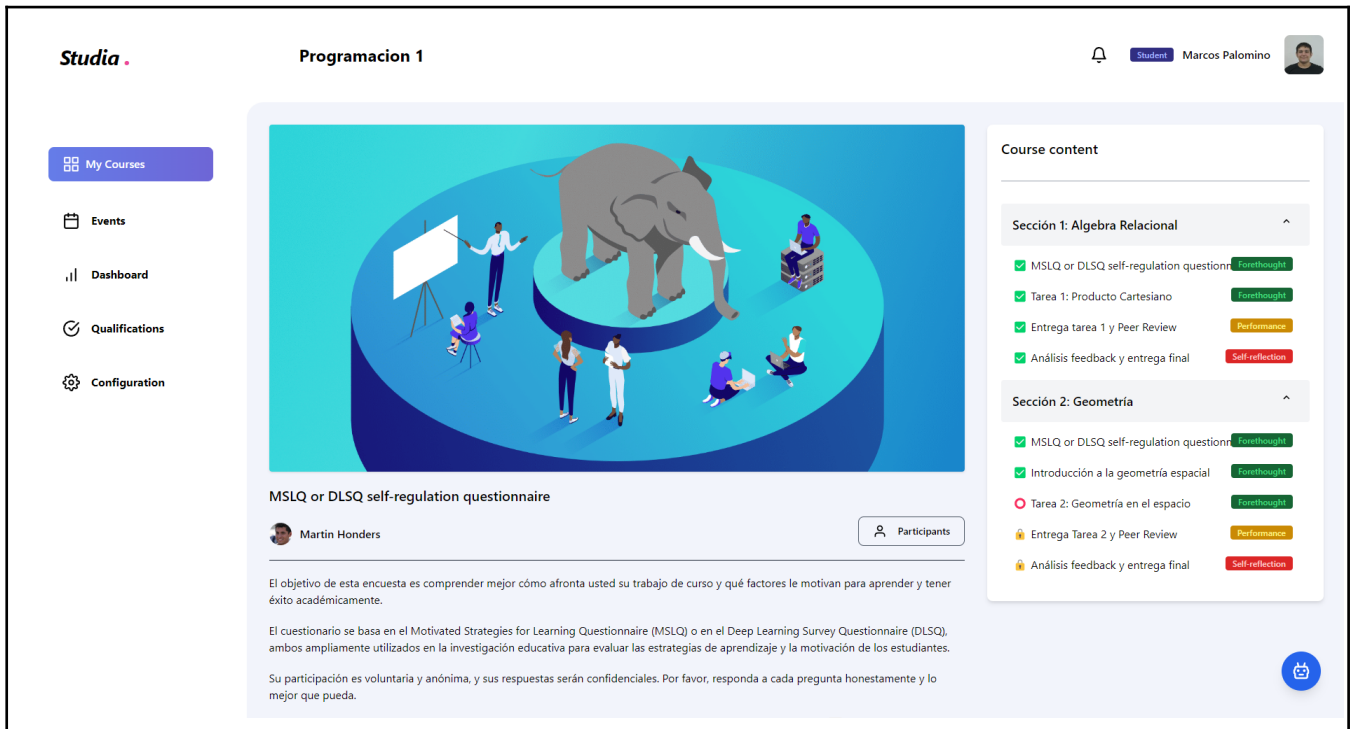


Fig. Anexos 1. Imatges finals de Studia

Documentació de la API

Accounts

Endpoint	Descripció	Params	Mètode
api/accounts/users/{pk}/	Obté els detalls d'un usuari específic a partir del seu ID (pk).	{id del usuari:pk}	GET
api/accounts/users/	Llista tots els usuaris.	-	GET
api/accounts/users/{pk}/courses/	Obté la llista de cursos d'un usuari específic a partir del seu ID.	{id del usuari:pk}	GET
api/accounts/users/{pk}/courses/timeline/	Obté una cronologia de les activitats d'un curs d'un usuari específic.	{id del usuari:pk}	GET
api/accounts/users/{pk}/qualifications/	Obté les qualificacions d'un estudiant específic a partir del seu ID.	{id del usuari:pk}	GET

api/accounts/users/{pk}/courses/new/	Obté les novetats dels cursos d'un usuari específic a partir del seu ID.	{id del usuari:pk}	GET
api/accounts/users/{pk}/	Modifica les dades d'un usuari específic a partir del seu ID.	{id del usuari:pk, dades de l'usuari}	PUT
api/accounts/users/{pk}/	Modifica les dades d'un usuari específic a partir del seu ID.	{id del usuari:pk}	DELETE

Fig. Anexos 2. API del servei de Accounts

Courses

Endpoint	Descripció	Params	Mètode
api/courses/	Llista tots els cursos.	-	GET
api/courses/create/	Crea un nou curs.	Dades del curs	POST
api/courses/{pk}/add_student/{user_id}/	Afegeix un estudiant a un curs específic.	{pk: ID del curs, user_id: ID de l'estudiant}	POST

api/courses/{pk}/	Obté els detalls d'un curs específic.	{pk: ID del curs}	GET
api/courses/{pk}/activities/	Obté els detalls de les activitats d'un curs específic.	{pk: ID del curs}	GET
api/courses/{pk}/news	Obté les novetats d'un curs específic.	{pk: ID del curs}	GET
api/courses/{pk}/update/	Actualitza les dades d'un curs específic.	{pk: ID del curs, dades del curs}	PUT
api/courses/{pk}/delete/	Elimina un curs específic.	{pk: ID del curs}	DELETE
api/courses/{pk}/students/	Obté la llista d'estudiants inscrits en un curs específic.	{pk: ID del curs}	GET
api/courses/{pk}/enroll/	Inscriu un estudiant en un curs específic.	{pk: ID del curs, user_id: ID de l'estudiant, dades d'inscripció}	POST
api/courses/{pk}/remove_student/{user_id}/	Elimina un estudiant d'un curs específic.	{pk: ID del curs, user_id: ID de l'estudiant}	DELETE
api/courses/{pk}/activities/create/	Crea una nova activitat per a un curs específic.	{pk: ID del curs, dades de l'activitat}	POST

api/courses/{pk}/activities/{activity_id}/	Obté els detalls d'una activitat específica d'un curs.	{pk: ID del curs, activity_id: ID de l'activitat}	GET
api/courses/{pk}/activities/{activity_id}/update/	Actualitza les dades d'una activitat específica d'un curs.	{pk: ID del curs, activity_id: ID de l'activitat, dades de l'activitat}	PUT
api/courses/{pk}/activities/{activity_id}/delete/	Elimina una activitat específica d'un curs.	{pk: ID del curs, activity_id: ID de l'activitat}	DELETE
api/courses/{pk}/news/create/	Crea una nova notícia per a un curs específic.	{pk: ID del curs, dades de la notícia}	POST
api/courses/{pk}/news/{news_id}/	Obté els detalls d'una notícia específica d'un curs.	{pk: ID del curs, news_id: ID de la notícia}	GET
api/courses/{pk}/news/{news_id}/update/	Actualitza les dades d'una notícia específica d'un curs.	{pk: ID del curs, news_id: ID de la notícia, dades de la notícia}	PUT
api/courses/{pk}/news/{news_id}/delete/	Elimina una notícia específica d'un curs.	{pk: ID del curs, news_id: ID de la notícia}	DELETE

Fig. Anexos 3. API del servei de Courses

Auth System (Djoser)

Endpoint	Descripció	Params	Mètode
/auth/users/	Crea un nou usuari.	{ "email", "password" }	POST
/auth/users/me/	Obté els detalls de l'usuari autènticat.	-	GET
/auth/users/{id}/	Obté els detalls d'un usuari específic.	{id: ID de l'usuari}	GET
/auth/users/{id}/	Actualitza els detalls d'un usuari específic.	{id: ID de l'usuari}	PUT
/auth/jwt/create/	Obté un token JWT per autènticar l'usuari.	{ "email", "password" }	POST
/auth/jwt/refresh/	Actualitza el token JWT per estendre el seu temps de validesa.	{ "refresh" }	POST
/auth/jwt/verify/	Verifica la validesa d'un token JWT.	{ "token" }	POST
/auth/password/reset/	Envia un correu electrònic amb instruccions per restablir la contrasenya.	{ "email" }	POST
/auth/password/reset/confirm/	Restableix la contrasenya després de rebre el correu electrònic de restabliment.	{ "uid", "token", "new_password" }	POST
/auth/users/{id}/delete/	Elimina un usuari específic.	{id: ID de l'usuari}	DELETE
/auth/users/set_password/	Canvia la contrasenya de l'usuari autènticat.	{ "new_password", "re_new_password" }	POST

/auth/users/set_username/	Canvia el nom d'usuari de l'usuari autenticat.	{ "new_username" }	POST
/auth/token/login/	Obté un token d'autenticació per l'usuari.	{ "username", "password" }	POST
/auth/token/logout/	Invalida el token d'autenticació actual.	-	POST
/auth/users/activation/	Activa el compte de l'usuari després de rebre el correu electrònic d'activació.	{ "uid", "token" }	POST
/auth/users/resend_activation/	Torna a enviar el correu electrònic d'activació a l'usuari.	{ "email" }	POST

Fig. Anexos 4. API del servei de Authentication

Fig. Anexos 5. Model Relacional de PostgreSQL

