Luis Miguel Soriano Frutos

DESENVOLUPAMENT D'UNA EINA DE SUPORT A L'AVALUACIÓ D'EXÀMENS MITJANÇANT OCR I LLMs

TREBALL DE FI DE GRAU

dirigit per Dr. Edgar Batista de Frutos

Grau d'Enginyeria Informàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2025

Resum.

Aquest treball té com a objectiu el desenvolupament d'una eina de suport per a la correcció d'exàmens manuscrits mitjançant la combinació de models de reconeixement òptic de caràcters i Large Language Models. Per aconseguir-ho, s'implementen diferents tècniques d'intel·ligència artificial per a la transcripció de text manuscrit i la seva posterior revisió i avaluació assistida. A més, es realitza una anàlisi comparativa entre els models de reconeixement òptic de caràcters per determinar quins ofereixen millors resultats en termes de precisió. Aquest sistema busca facilitar i optimitzar el procés de correcció i reduir la càrrega de treball dels docents, millorant així l'eficiència en l'avaluació acadèmica. Aquest enfocament permet assistir una part significativa del procés educatiu sense substituir la intervenció del docent, i sense comprometre la qualitat ni la personalització de l'avaluació.

Paraules clau:

Reconeixement òptic de caràcters, Xarxes neuronals, Model de llenguatge gran, Intel·ligència artificial, Suport a l'avaluació d'exàmens, Processament de llenguatge natural.

Resumen.

Este trabajo tiene como objetivo el desarrollo de una herramienta de apoyo para la evaluación de exámenes manuscritos mediante la combinación de modelos de reconocimiento óptico de caracteres y modelos de lenguaje grande. Para lograrlo, se implementan diferentes técnicas de inteligencia artificial para la transcripción de texto manuscrito y su posterior revisión y evaluación asistida. Además, se realiza un análisis comparativo entre modelos de reconocimiento óptico de caracteres para determinar cuáles ofrecen mejores resultados en términos de precisión. Esta herramienta busca asistir y optimizar el proceso de corrección y reducir la carga de trabajo de los docentes, mejorando así la eficiencia en la evaluación académica. Este enfoque permite apoyar una parte significativa del proceso educativo sin sustituir la intervención docente, y sin comprometer la calidad ni la personalización de la evaluación.

Palabras clave:

Reconocimiento óptico de caracteres, Redes neuronales, Modelo de lenguaje grande, Inteligencia artificial, Soporte en la evaluación de exámenes, Procesamiento de lenguaje natural.

Abstract.

This project aims to develop a support tool for grading handwritten exams by combining optical character recognition models and Large Language Models. To achieve this, different artificial intelligence techniques are implemented for handwritten text transcription and its subsequent assisted review and evaluation. Additionally, a comparative analysis between optical character recognition models is conducted to determine which offers better results in terms of accuracy and robustness. This system aims to assist and optimize the grading process and reduce the workload of teachers, thereby improving efficiency in academic assessment. This approach allows for supporting a significant part of the educational process without replacing the teacher's role,

and without compromising the quality, fairness, or personalization of assessment.

Keywords:

Optical character recognition, Neural networks, Large language models, Artificial intelligence, Support for exam grading, Natural language processing.

Índex

1 INTRODUCCIÓ 5
1.1 DESCRIPCIÓ GENERAL DEL PROJECTE 5
1.2 NECESSITATS 5
1.3 PREVISIÓ D'ÚS 6
1.4 OBJECTIUS DEL TFG 6
2 ESTAT DE L'ART 8
2.1 RECONEIXEMENT ÒPTIC DE CARÀCTERS 8
2.1.1 Tècniques d'OCR 8
2.2 MODELS DE LLENGUATGE GRAN 10
3 PLANIFICACIÓ 11
4 REQUISITS 13
4.1 REQUISITS FUNCIONALS 13
4.1.1 Casos d'ús principals 13
4.2 REQUISITS NO FUNCIONALS 15
5 ANÀLISI DELS REQUISITS FUNCIONALS 17
5.1 DIAGRAMA DE CLASSES 17
5.2 DIAGRAMA DE SEQÜÈNCIES 19
6 DISSENY 24
6.1 ARQUITECTURA DE L'APLICACIÓ 24
6.2 DISSENY DE LA INTERFÍCIE GRÀFICA 25
6.3 DISSENY DE LA PERSISTÈNCIA DE DADES 25
7 IMPLEMENTACIÓ 27
7.1 ARQUITECTURA 27
7.2 ESTRUCTURA I FUNCIONALITAT DELS COMPONENTS 27
7.2.1 app_main.py i app_main_console.py 27
7.2.2 app_layout.py 28
7.2.3 image_processing_module.py 29
7.2.4 ocr_module.py 31
7.2.5 llm_module.py 31
7.2.6 ollama_server.py 32
8 AVALUACIÓ 33
8.1 SELECCIÓ DE L'OCR: PROVES PRÈVIES AL DESENVOLUPAMENT DEL SISTEMA 33
8.1.1 Resultats obtinguts en les proves 34
8.1.2 Conclusió de les proves de selecció d'OCR 37
8.2 VALIDACIÓ POST-IMPLEMENTACIÓ: SEGMENTACIÓ I CORRECCIÓ LLM 38
8.2.1 Validació de la segmentació de les imatges i reconeixement OCR 38
8.2.2 Validació de la correcció automàtica amb el model de llenguatge 39
8.3 AVALUACIÓ DEL SISTEMA EN RELACIÓ ALS REQUISITS FUNCIONALS I NO FUNCIONALS 41
9 CONCLUSIONS 42
10 APLICACIÓ DELS PRINCIPIS ÈTICS I RESPONSABILITAT SOCIAL 44
10.1 IGUALTAT 44
10.2 MEDI AMBIENT 44
10.3 RESPONSABILITAT SOCIAL 44
10.4 ÈTICA 44
11 RECURSOS UTILITZATS 45
11.1 LLICÈNCIES DE PROGRAMARI USAT 45
11.2 DATASETS UTILITZATS 46

11.3 PAQUETS I LLIBRERIES UTILITZADES 46
CITES 48

Índex de taules

TAULA 1. TAULA RESUM DE LES MÈTRIQUES AVALUADES SOBRE ELS MODELS OCR AMB IAM HANDWRITING LINES. 34
TAULA 2. TAULA RESUM DE LES MÈTRIQUES AVALUADES SOBRE ELS MODELS OCR AMB IAM HANDWRITING SENTENCES. 36
TAULA 3. TAULA DE LLICÈNCIES USADES PER A AQUEST PROJECTE. 45

Índex de figures

FIGURA 1. TASQUES DEL TREBALL DEFINIDES AL MSPROJECT I DIAGRAMA DE GANTT. 12
FIGURA 2. DIAGRAMA DE CLASSES DE L'APLICACIÓ. 17
FIGURA 3. DIAGRAMA DE SEQÜENCIES DEL CD01. 19
FIGURA 4. DIAGRAMA DE SEQÜENCIES DEL CD02. 20
FIGURA 5. DIAGRAMA DE SEQÜENCIES DEL CD03. 21
FIGURA 6. DIAGRAMA DE SEQÜENCIES DEL CD04. 22
FIGURA 7. DIAGRAMA DE SEQÜENCIES DEL CD05. 23
FIGURA 8. DIAGRAMA DEL FLUX D'EXECUCIÓ 25
FIGURA 9. INTERFÍCIE GRÀFICA DE L'APLICACIÓ 28
FIGURA 10. IMATGE PROCESSADA A ESCALA DE GRISOS. 29
FIGURA 11. IMATGE BINARITZADA. 29
FIGURA 12. REGIONS DETECTADES I AGRUPADES EN BLOCS CONTINUS 30
FIGURA 13. SECCIÓ DETECTADA CONTENIDORA DE L'ENUNCIAT 30
FIGURA 14. SECCIÓ DETECTADA CONTENIDORA DE LA RESPOSTA MANUSCRITA. 30
FIGURA 15. GRÀFIC DE LA PRECISIÓ MITJANA OBTINGUDA PER A IAM HANDWRITING LINES. 35
FIGURA 16. GRÀFIC DEL TEMPS D'EXECUCIÓ REQUERIT PELS MODELS OCR EN PROCESSAR 100 IMATGES DE IAM HANDWRITING LINES 35
FIGURA 17. GRÀFIC DE LA PRECISIÓ MITJANA OBTINGUDA PER A IAM HANDWRITING SENTENCES. 36
FIGURA 18. GRÀFIC DEL TEMPS D'EXECUCIÓ REQUERIT PELS MODELS OCR EN PROCESSAR 100 IMATGES DE IAM HANDWRITING SENTENCES 37
FIGURA 19. GRÀFIC DEL PERCENTATGE D'ENCERTS I ERRORS DEL MODEL LLAMA2 40

1 Introducció

El reconeixement òptic de caràcters o OCR[1] és una eina que ha anat evolucionant significativament els últims anys gràcies als avenços en intel·ligència artificial, xarxes neuronals i visió per computador. Aquesta tecnologia permet convertir textos manuscrits o impresos en formats digitals per tal de ser processats i analitzats automàticament, obrint la porta a noves formes d'automatització i eficiència en àmbits tan diversos com l'educació, l'administració o la indústria. L'OCR, combinat amb Large Language Model o LLM[2], ofereix un gran potencial per a l'automatització de tasques com la correcció i avaluació de textos i respostes en exàmens, permetent reduir costos i temps, alhora que s'augmenta la precisió i l'objectivitat. A mesura que els models d'OCR milloren, poden reconèixer amb més precisió escriptura manuscrita, fins i tot amb cal·ligrafies difícils, oferint així una eina potent per a la digitalització i avaluació de textos escrits a mà. El principal repte, i l'objectiu d'aquest treball, és aconseguir una integració òptima entre el reconeixement de caràcters i l'ús d'LLM per garantir una avaluació d'exàmens precisa, eficient i automatitzada, contribuint a transformar i modernitzar els processos educatius tradicionals.

1.1 Descripció general del projecte

La idea principal d'aquest treball és desenvolupar una aplicació innovadora que, a partir d'OCR i LLM, pugui reconèixer respostes manuscrites en exàmens de resposta llarga i aplicar-hi una avaluació automàtica. Aquesta solució tecnològica vol oferir un sistema fiable i accessible que pugui ser aplicat en diferents contextos educatius, facilitant la feina dels docents.

Per aconseguir-ho, també es durà a terme una avaluació de la precisió en el reconeixement dels diferents models d'OCR i una avaluació de la qualitat de l'avaluació realitzada en les diferents respostes dels exàmens, per garantir la màxima qualitat del sistema.

1.2 Necessitats

L'avenç de les tecnologies d'intel·ligència artificial ha permès millorar notablement el reconeixement i el processament de text, però la integració efectiva entre OCR i LLM encara suposa reptes significatius, especialment en l'àmbit de la correcció i l'avaluació de respostes manuscrites en exàmens. Aquest projecte neix de la necessitat de combinar models OCR amb LLM per tal de poder automatitzar i optimitzar el procés de correcció, maximitzant tant la precisió del reconeixement de text com la coherència de l'avaluació.
Les principals necessitats del projecte són:

- La integració de models OCR amb LLM per explorar com es poden complementar per al procés d'avaluació.
- El suport i simplificació del procés de correcció d'exàmens, reduint considerablement la càrrega de treball dels docents i assegurant una avaluació més fiable i objectiva.

[1] OCR: Reconeixement òptic de caràcters o *Optical Character Recognition.*

[2] LLM: Model de llenguatge gran o *Large Language Model.*

- Avaluació de la compatibilitat entre diferents models OCR i LLM, identificant quines combinacions ofereixen millors resultats en termes de precisió en la transcripció i qualitat en l'avaluació de les respostes.
- Anàlisi de la coherència de les avaluacions generades pels LLM.

Aquest projecte busca proporcionar una eina útil per a l'educació, ajudant a optimitzar l'avaluació d'exàmens escrit.

1.3 Previsió d'ús

L'aplicació desenvolupada en aquest projecte tindrà una àmplia aplicabilitat en l'àmbit educatiu, especialment en el suport de la correcció d'exàmens i tasques escrites a mà, contribuint a fer el procés més eficient i accessible. Els seus usos potencials inclouen:

- Institucions educatives: Centres de secundària, universitats i escoles de formació professional podran utilitzar aquest sistema per agilitzar el procés de correcció d'exàmens de resposta llarga, reduint la càrrega de treball dels professors i millorant l'objectivitat de l'avaluació.
- Plataformes d'aprenentatge en línia: Sistemes d'e-learning poden integrar aquesta eina per permetre als estudiants enviar respostes manuscrites digitalitzades i rebre una correcció automatitzada amb retroalimentació quasi immediata, millorant l'experiència d'aprenentatge digital.
- Anàlisi de models OCR i LLM: Investigadors i desenvolupadors d'intel·ligència artificial podran utilitzar aquest projecte per avaluar quines combinacions d'OCR i LLM ofereixen millors resultats en reconeixement de text i coherència en l'avaluació de respostes.

Aquest projecte, per tant, no només representa una millora tecnològica en l'OCR, sinó que també suposa una eina amb impacte directe en l'educació i la investigació en intel·ligència artificial.

1.4 Objectius del TFG

En aquest treball s'estableixen diversos objectius, sent l'objectiu principal el desenvolupament d'una aplicació orientada a la integració de diferents models d'OCR i de LLM, amb la finalitat de realitzar avaluacions de respostes manuscrites a exàmens de diversos nivells de formació.

No obstant, per garantir que la solució sigui eficient i efectiva, és necessari abordar diversos aspectes tècnics al llarg del desenvolupament de l'aplicació. Un dels primers objectius serà l'avaluació de la precisió de cadascun dels models d'OCR utilitzats, per poder categoritzar-los i inferir en quins casos d'ús un model és més eficient i precís respecte els altres. Això implica realitzar una comparativa entre els models més populars, identificant els punts forts i febles de cadascun en funció de les característiques específiques dels exàmens. Aquesta anàlisi permetrà determinar quins models ofereixen millors resultats en funció de l'entorn d'ús concret, ajudant a triar la millor opció per a cada cas.

D'altra banda, un dels objectius més crítics serà avaluar la precisió de la correcció i avaluació dels exàmens a partir dels LLM. En aquest cas, l'objectiu és determinar com els models poden interpretar les respostes manuscrites i aplicar les normes de correcció establertes, amb l'objectiu d'assegurar-se que l'avaluació sigui precisa i objectiva. Aquesta tasca inclourà l'anàlisi de la capacitat dels models per identificar errors conceptuals,

incoherències o respostes incompletes, tot tenint en compte el context i la interpretació que s'esperaria en una avaluació humana.

En resum, els objectius d'aquest treball inclouen el desenvolupament d'una aplicació integrada per a l'avaluació de respostes manuscrites i l'optimització dels processos de correcció i avaluació d'exàmens.

2 Estat de l'art

En aquest apartat es presenta una visió general de les tecnologies que fan possible aquest projecte, centrant-se principalment en l'OCR i els LLM. L'objectiu és entendre com funcionen aquestes eines i quines tècniques hi ha darrere. Es fa un repàs als sistemes OCR més comuns i com han evolucionat gràcies a l'ús de xarxes neuronals, així com als models que permeten interpretar el contingut i generar respostes coherents. Aquesta base ajudarà a entendre de manera més clara els diferents components del sistema desenvolupat i quines decisions tecnològiques s'han pres durant el desenvolupament.

2.1 Reconeixement òptic de caràcters

Aquest apartat presenta els conceptes fonamentals que engloben els aspectes tècnics generals dels models d'OCR, destacant les tecnologies clau utilitzades en la conversió de text present en una imatge a text digital. Per a això, s'exposaran les diferents tècniques emprades actualment.

Els sistemes d'OCR es poden classificar segons diversos criteris:

- Segons el tipus d'execució:
  - OCR Offline: Processa imatges estàtiques de documents escanejats o fotografiats.
  - OCR Online: Requereix connexió a internet per processar les imatges mitjançant serveis externs.
- Segons la metodologia d'anàlisi:
  - Mètodes tradicionals, basats en regles heurístiques i tècniques estadístiques, que inclouen aproximacions com la coincidència de plantilles, l'extracció manual de característiques i models probabilístics.
  - Mètodes basats en Deep Learning, que aprofiten xarxes neuronals profundes per aconseguir un reconeixement més robust i adaptatiu, especialment en textos manuscrits i documents degradats.

Si bé els enfocaments tradicionals van ser fonamentals en les primeres etapes del desenvolupament de l'OCR, presentaven limitacions significatives en la seva capacitat de generalització, especialment davant de variacions tipogràfiques i escriptures manuscrites. Amb l'avenç de la visió per computador i el Deep Learning, els sistemes moderns han evolucionat cap a arquitectures més sofisticades, capaces de processar text de manera més precisa i eficient. La visió per computador ha permès extreure característiques rellevants de les imatges, detectant i processant patrons textuals fins i tot en condicions adverses, com il·luminació variable, fons sorollosos o distorsionats [1].

Aquest treball se centra principalment en l'OCR de text manuscrit (Handwritten OCR), una de les àrees més complexes dins d'aquest camp. A diferència del text imprès, el text manuscrit presenta una gran variabilitat en l'escriptura, incloent-hi diferències en l'estil de cada individu, la inclinació de les lletres i possibles distorsions, fet que requereix l'ús de tècniques avançades per garantir una correcta identificació i transcripció del contingut.

2.1.1 Tècniques d'OCR

Els sistemes d'OCR moderns es basen en xarxes neuronals profundes per millorar la precisió en el reconeixement de text. A continuació, es descriuen els models principals utilitzats:

1. **Xarxes Neuronals Convolucionals (CNN - Convolutional Neural Network):** Són essencials en sistemes OCR, ja que permeten extreure característiques visuals d'una imatge mitjançant filtres que identifiquen patrons com vores, formes i lletres. Aquestes característiques es detecten de manera progressiva a través de capes, des d'elements bàsics fins a estructures complexes com caràcters complets. Les capes de pooling redueixen la mida de la imatge tot mantenint la informació rellevant, mentre que les capes totalment connectades (FC) permeten classificar els caràcters detectats. Aquest procés fa que les CNN siguin molt eficients per al reconeixement de text, especialment en manuscrits o fonts degradades [1].
2. **Xarxes Neuronals Recurrents (RNN - Recurrent Neural Network):** Són utilitzades per processar seqüències de text, mantenint una relació entre caràcters consecutius. Aquestes xarxes són útils en OCR, ja que permeten millorar el reconeixement de paraules i frases senceres, tenint en compte el context de la informació. A diferència de les xarxes neuronals convencionals, les RNN tenen una memòria interna que els permet emmagatzemar informació referent a iteracions anteriors, fent-les especialment adequades per a tasques de seqüències com el reconeixement de text manuscrit. No obstant, poden patir el que s'anomena *vanishing gradient* que fa referència a que la informació es va dissipant a mesura que es retro-alimenta a través de les capes de la xarxa, factor que influeix negativament en l'aprenentatge en cas de seqüències llargues [2].
3. **Xarxes Neuronals LSTM (Long Short-Term Memory):** Són una variant avançada de les RNNs, les quals solucionen el problema de pèrdua de memòria a llarg termini. Aquest problema es resol mitjançant una arquitectura basada en cel·les de memòria i portes adaptatives. Aquestes portes inclouen la porta d'entrada (input gate), la porta d'oblit (forget gate) i la porta de sortida (output gate), que controlen el flux d'informació dins de la xarxa. La porta d'oblit (forget gate) és de gran rellevància tenint en compte que decideix quina informació es conserva i quina es descarta, permetent que la xarxa recordi la informació més rellevant durant el transcurs dels períodes més llargs. En OCR, aquestes xarxes s'utilitzen per reconèixer text manuscrit i cursiu, ja que poden mantenir la informació sobre caràcters anteriors i millorar la interpretació del text. Aquest enfocament és especialment valuós per a idiomes amb cal·ligrafies complexes o escriptures cursives, on la connexió entre caràcters és fonamental per aconseguir una transcripció precisa [2].
4. **Xarxes CRNN (Convolutional Recurrent Neural Networks):** Són una combinació entre CNNs (per a l'extracció de característiques) i RNNs o LSTMs (per a modelar la seqüència de text). Aquesta combinació és especialment efectiva en OCR per a documents escanejats, manuscrits i textos complexos, ja que permet capturar tant la informació visual com la relació seqüencial entre caràcters [4].
5. **Transformers aplicats a OCR:** Són un model d'intel·ligència artificial basat en mecanismes d'atenció que permet el reconeixement òptic de caràcters analitzant una imatge de text en la seva totalitat. A diferència de les CNNs, que processen característiques locals, els Transformers capten relacions de llarg abast entre caràcters i paraules, millorant el reconeixement en textos desordenats o manuscrits. El procés inclou una etapa d'extracció de característiques, on un transformador divideix la imatge en segments i extreu informació rellevant de forma jeràrquica. A continuació, el model utilitza el mecanisme self-attention per analitzar les relacions entre diferents parts del

document. Això li permet entendre el context global i no només fragments aïllats, millorant així la precisió en la identificació del text. Gràcies a aquesta capacitat, els transformers poden processar documents amb estructures complexes o poc convencionals sense perdre informació rellevant [3].

2.2 Models de llenguatge gran

Els LLM són models d'intel·ligència artificial entrenats amb grans quantitats de text per comprendre, generar i analitzar llenguatge natural amb un grau alt de precisió. En el context de l'OCR i en aquest treball, els LLM s'utilitzen per interpretar, corregir i avaluar textos reconeguts automàticament.

A continuació, es presenten els principals enfocaments funcionals dins dels models de llenguatge gran, tots ells basats en l'arquitectura Transformer:

- **Models encoder-decoder:** Segueixen una estructura en dues fases (codificació i generació) i són habituals en tasques com la traducció, el resum i la reescriptura.
- **Models autoregressius:** Generen text de manera seqüencial, predint la següent paraula en funció de les anteriors. Són especialment útils en la creació de contingut, la resposta a preguntes i la generació de text coherent i contextualitzat.
- **Models adaptatius:** Poden ser especialitzats mitjançant fine-tuning, per adaptar-se en camps específics, ajustant els seus pesos per comprendre millor termes tècnics, llenguatges especialitzats o contextos concrets.

Encara que aquests enfocaments poden variar en el seu ús o aplicació, comparteixen una base comuna: el model Transformer, que permet representar relacions entre paraules de manera contextual, independentment de la seva posició en la seqüència [14].

3 Planificació

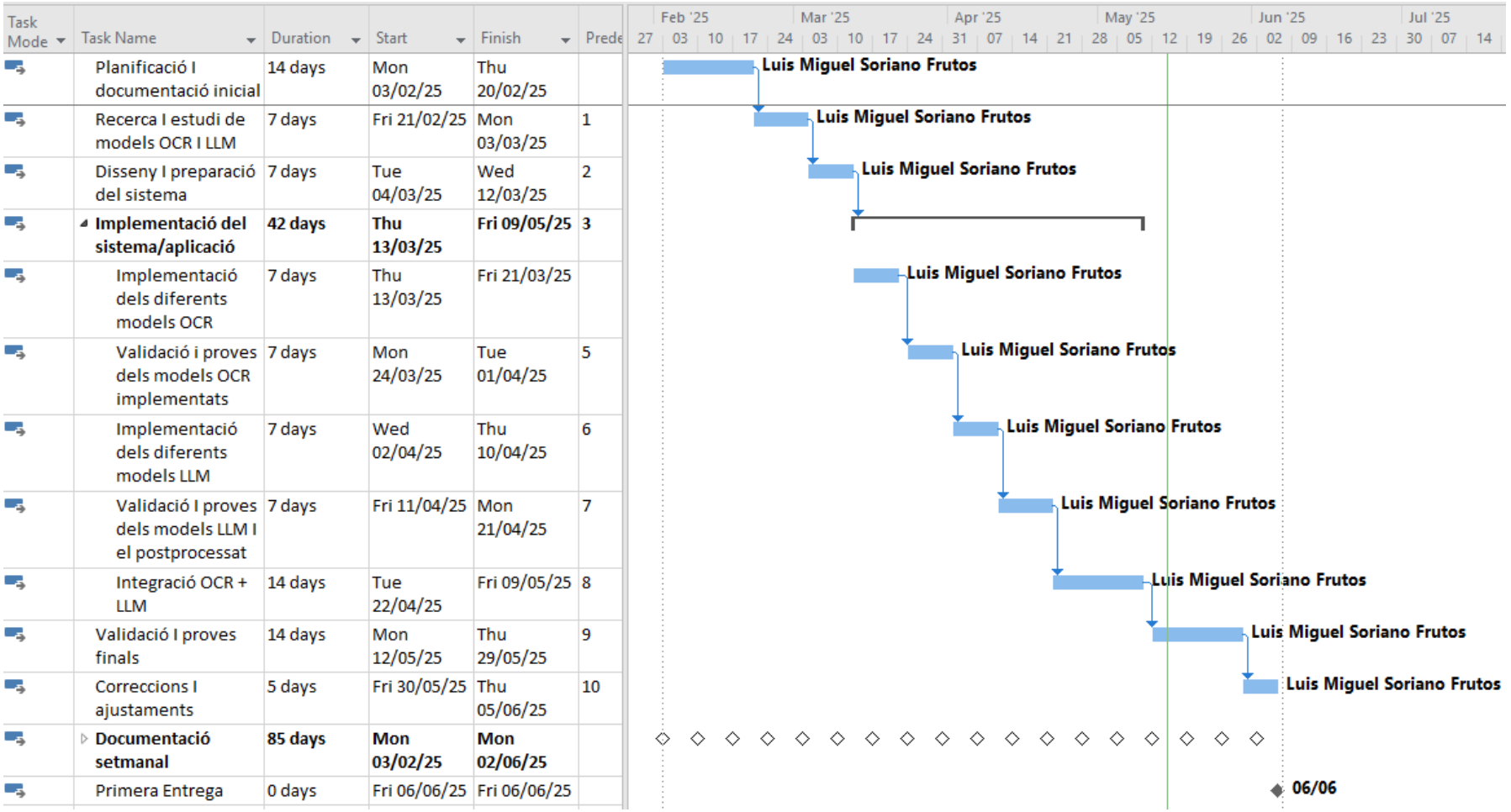
Per tal d'assegurar i poder realitzar un desenvolupament estructurat i eficient, s'ha establert una planificació del treball basada en un diagrama de Gantt, elaborat amb l'eina informàtica MS Project. Aquest diagrama defineix les diferents fases del treball, incloent-hi la planificació, recerca, disseny, implementació, validació, proves i documentació, així com les dependències entre les diferents tasques.

El treball s'ha estructurat en les següents fases:

- **Fase 1: Planificació i documentació inicial**
  Fase dedicada a establir els objectius del treball, definir la metodologia a seguir i elaborar una planificació temporal del treball. A més inclou la redacció dels primers apartats de la documentació, com la introducció, els objectius i les paraules clau.
- **Fase 2: Recerca i estudi de models OCR i LLM**
  Investigació sobre les diferents tecnologies OCR i LLM per determinar les millors opcions per al treball.
- **Fase 3: Disseny i preparació del sistema**
  Es defineix l'estructura i arquitectura de l'aplicació i a més es preparen els conjunts de dades per a poder realitzar les proves pertinents
- **Fase 4: Implementació del sistema**
  Durant aquesta fase es desenvolupa el sistema complet, començant per la configuració i implementació dels diferents models d'OCR a utilitzar, seguit de la seva validació per assegurar-ne l'eficiència en el reconeixement de text. Posteriorment, es procedeix a la implementació i configuració dels diferents models d'LLM a utilitzar. Finalment, es realitza la integració dels models OCR i LLM per obtenir un sistema funcional capaç de transcriure i avaluar textos manuscrits de manera automatitzada.
- **Fase 5: Validació i proves finals**
  Proves exhaustives per avaluar la precisió global del sistema i corregir possibles errors.
- **Fase 6: Correccions i ajustaments**
  Millores finals segons els resultats de les proves i preparació per a l'entrega.

A més, s'ha establert una tasca iterativa orientativa per a representar el procés de documentació continu durant tot el desenvolupament del treball. L'objectiu d'aquesta tasca es plasmar i garantir el registre continu dels avenços del projecte.

En la següent imatge es mostra tant les diferents fases amb les seves tasques, com la seva representació en el diagrama de Gantt.



Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Planificació I documentació inicial	14 days	Mon 03/02/25	Thu 20/02/25	
	Recerca I estudi de models OCR I LLM	7 days	Fri 21/02/25	Mon 03/03/25	1
	Disseny I preparació del sistema	7 days	Tue 04/03/25	Wed 12/03/25	2
	Implementació del sistema/aplicació	**42 days**	**Thu 13/03/25**	**Fri 09/05/25**	**3**
	Implementació dels diferents models OCR	7 days	Thu 13/03/25	Fri 21/03/25	
	Validació i proves dels models OCR implementats	7 days	Mon 24/03/25	Tue 01/04/25	5
	Implementació dels diferents models LLM	7 days	Wed 02/04/25	Thu 10/04/25	6
	Validació I proves dels models LLM I el postprocessat	7 days	Fri 11/04/25	Mon 21/04/25	7
	Integració OCR + LLM	14 days	Tue 22/04/25	Fri 09/05/25	8
	Validació I proves finals	14 days	Mon 12/05/25	Thu 29/05/25	9
	Correccions I ajustaments	5 days	Fri 30/05/25	Thu 05/06/25	10
	Documentació setmanal	**85 days**	**Mon 03/02/25**	**Mon 02/06/25**	
	Primera Entrega	0 days	Fri 06/06/25	Fri 06/06/25	

Figura 1. Tasques del treball definides al MSProject i diagrama de Gantt.

4 Requisits

Aquest apartat descriu els requisits que ha de complir l'aplicació desenvolupada per tal d'assolir els objectius definits en aquest treball. Aquests es divideixen en dues categories principals: funcionals i no funcionals.

La correcta definició i implementació d'aquests requisits és essencial per garantir que l'aplicació compleixi amb els objectius del projecte, oferint una eina eficaç, segura i útil per al suport en la correcció d'exàmens manuscrits.

4.1 Requisits funcionals

Els requisits funcionals especifiquen les funcionalitats bàsiques i necessàries que el sistema ha de poder realitzar correctament.

- **Càrrega d'imatges:** El sistema ha de permetre a l'usuari carregar imatges d'exàmens manuscrits en formats comuns com PNG i JPG.
- **Extracció d'enunciats i respostes:** Un cop carregada la imatge, l'aplicació ha de ser capaç d'identificar i extreure tant les preguntes com les respostes.
- **Segmentació del contingut:** El text extret ha de ser segmentat per tal de distingir clarament entre enunciats i respostes.
- **Generació del prompt per a l'LLM:** El sistema ha de construir un prompt estructurat, que inclogui tant l'enunciat de la pregunta com la resposta manuscrita de l'estudiant. A més, el prompt ha d'indicar explícitament el rol de l'LLM com a corrector d'exàmens, sol·licitant una valoració clara i objectiva de la resposta, així com una justificació o correcció si escau.
- **Correcció automàtica:** El sistema ha de poder valorar i corregir la resposta mitjançant un LLM, en funció dels criteris definits.
- **Generació de retroalimentació:** L'aplicació ha de proporcionar una valoració (qualitativa i/o quantitativa) de la resposta, amb comentaris que ajudin a l'estudiant a entendre els seus errors.

4.1.1 Casos d'ús principals

A continuació es descriuen els casos d'ús que cobreixen les funcionalitats bàsiques del sistema:

- **CU01: Carregar una imatge**
  - **Resum funcionalitat:** L'usuari carrega una imatge de l'examen manuscrit des del seu dispositiu.
  - **Actors:** Usuari.
  - **Precondició:** L'usuari té l'arxiu de l'examen preparat en format PNG o JPG.
  - **Postcondició:** El sistema rep i emmagatzema temporalment la imatge per al processament.
  - **Flux Principal:**
    1. L'usuari inicia l'aplicació.
    2. Fa clic sobre el botó "Select Image".

3. Selecciona un fitxer PNG o JPG del seu ordinador.
4. El sistema mostra una previsualització de la imatge carregada.

- **Alternatives de procés i excepcions:** Si el format no és vàlid, no es pot seleccionar.

- **CU02: Extracció del contingut d'una imatge**
  - **Resum funcionalitat:** El sistema processa la imatge carregada per extreure el text manuscrit, diferenciant preguntes i respostes.
  - **Actors:** Sistema.
  - **Precondició:** Hi ha una imatge vàlida carregada.
  - **Postcondició:** El text ha estat extret i segmentat en enunciat i resposta.
  - **Flux Principal:**
    1. El sistema rep la imatge carregada.
    2. Segmenta la imatge en pregunta i resposta.
    3. Aplica OCR sobre els segments per reconèixer el text.
    4. Processa el text reconegut.
    5. Desa el contingut per generar el prompt.
  - **Alternatives de procés i excepcions:** Cap.
- **CU03: Generació del prompt per a l'LLM**
  - **Resum funcionalitat:** El sistema construeix un prompt que inclou la pregunta, la resposta i la instrucció per a l'LLM.
  - **Actors:** Sistema.
  - **Precondició:** La pregunta i la resposta han d'estar prèviament extretes i segmentades.
  - **Postcondició:** El prompt està preparat per ser enviat al model LLM.
  - **Flux Principal:**
    1. El sistema accedeix a les dades extretes respecte la imatge (enunciat i resposta).
    2. Crea un prompt estructurat amb la pregunta i la resposta.
    3. Afegeix instruccions específiques perquè el model actuï com a corrector d'exàmens.
    4. Desa el prompt per a la següent etapa.
  - **Alternatives de procés i excepcions:** Cap.
- **CU04: Correcció de la resposta**
  - **Resum funcionalitat:** El sistema envia el prompt a l'LLM i rep la correcció.
  - **Actors:** Sistema

- **Precondició:** El prompt està generat correctament.
- **Postcondició:** El sistema rep i mostra la correcció amb la seva justificació o comentari corresponent.
- **Flux Principal:**
    1. El prompt es passa a l'LLM.
    2. El model processa la informació i genera una resposta.
    3. El sistema rep la correcció amb justificació/comentari.
    4. Emmagatzema temporalment la correcció per a poder-la mostrar després.
- **Alternatives de procés i excepcions:** Cap.

- **CU05: Presentació de resultats**
    - **Resum funcionalitat:** L'usuari visualitza la resposta corregida i els comentaris del model.
    - **Actors:** Usuari.
    - **Precondició:** El sistema ha rebut i emmagatzemat la correcció.
    - **Postcondició:** L'usuari ha pogut veure els resultats.
    - **Flux Principal:**
        1. El sistema mostra la resposta corregida.
        2. El sistema mostra la nota total obtinguda en la pregunta.
        3. El sistema mostra el comentari generat per l'LLM.
    - **Alternatives de procés i excepcions:** Cap.

4.2 Requisits no funcionals

Els requisits no funcionals, en canvi, fan referència a les condicions generals del sistema que no afecten directament a la funcionalitat, però si a la qualitat del servei ofert.

- **Usabilitat:** La interfície ha de ser intuïtiva i senzilla, pensada per a usuaris sense coneixements tècnics. L'objectiu és que qualsevol persona pugui carregar una imatge i obtenir una correcció sense dificultat.
- **Privacitat i confidencialitat:** El sistema ha de garantir que les dades utilitzades (les imatges d'exàmens) es tractin de forma segura i privada. No s'hauria d'emmagatzemar informació de manera permanent ni compartir-la amb serveis externs no autoritzats.
- **Temps de resposta:** El sistema ha de ser capaç de completar el procés de correcció en un temps raonable, de manera que l'usuari no hagi d'esperar més del necessari per obtenir els resultats. Idealment inferior a 2 minuts.
- **Compatibilitat:** El sistema hauria de funcionar en les plataformes i els sistemes operatius més habituals i a més, en diferents entorns de treball, per tal d'assegurar l'accessibilitat i la portabilitat del programari.

- **Escalabilitat:** Tot i estar dissenyat inicialment per a un ús individual o amb un volum de dades limitat, el sistema hauria de permetre adaptacions futures per gestionar una major quantitat de dades o usuaris.
- **Mantenibilitat:** L'arquitectura del sistema ha de facilitar-ne el manteniment i l'evolució, de manera que sigui possible afegir noves funcionalitats o actualitzar components amb un impacte mínim.

5 Anàlisi dels requisits funcionals

Aquest apartat té com a objectiu analitzar els requisits funcionals del sistema desenvolupat. Per fer-ho, s'utilitzen representacions UML com diagrames de classes i els diagrames de seqüència corresponents als principals casos d'ús. Aquestes eines permeten visualitzar l'estructura interna del sistema, així com el comportament dinàmic de les interaccions entre els diferents components i actors.

5.1 Diagrama de classes

En aquest subapartat es mostra l'estructura del sistema mitjançant un diagrama UML de classes. Aquest mostra les relacions entre els diferents components de l'aplicació, reflectint com s'organitzen les responsabilitats i les dependències per satisfer els requisits funcionals.

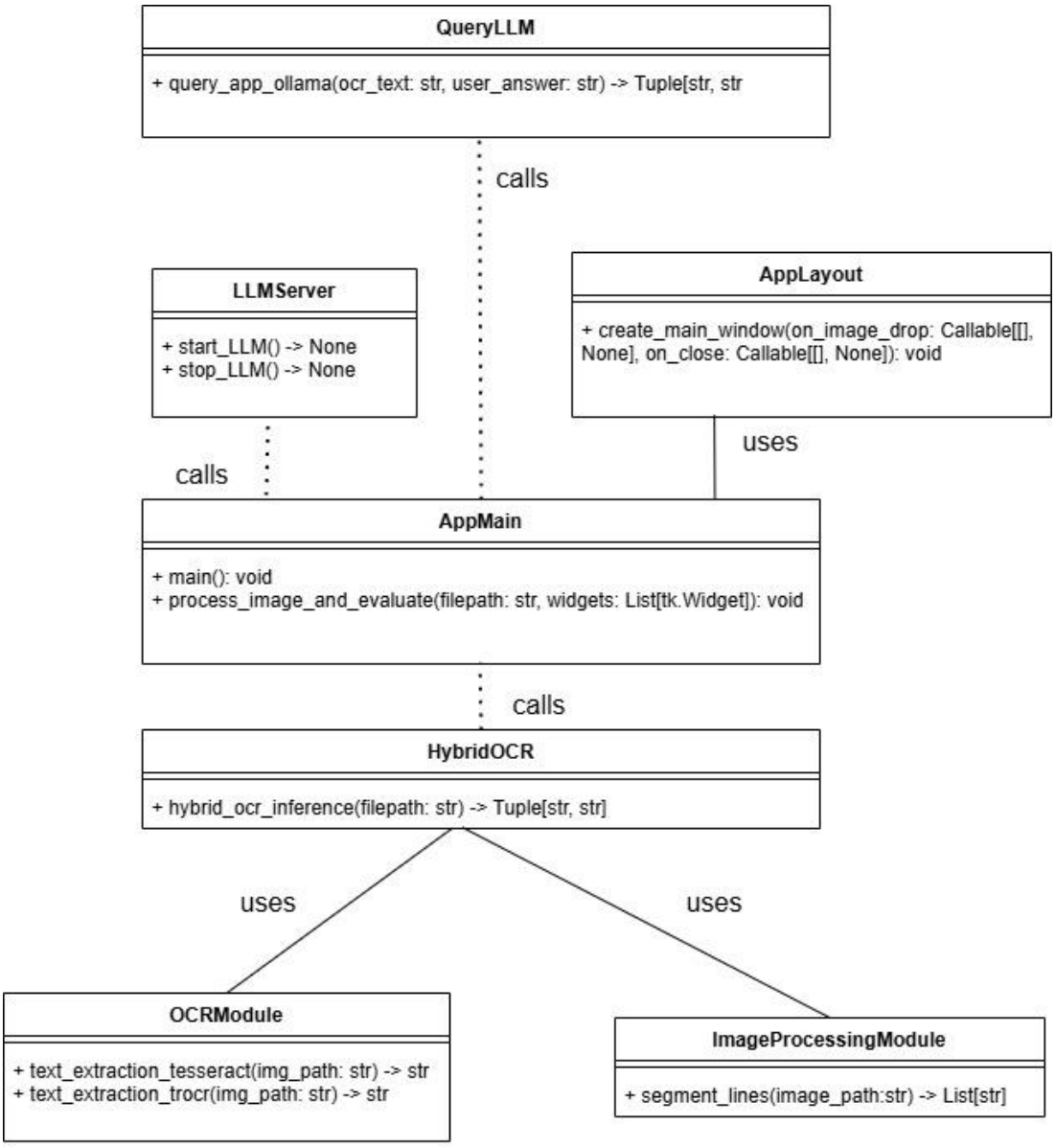


Figura 2. Diagrama de classes de l'aplicació.

A través d'aquest model visual, es pot observar com la classe principal AppMain actua com a coordinadora del sistema, interaccionant amb diversos mòduls especialitzats com HybridOCR per al reconeixement òptic de caràcters, QueryLLM per a consultes als models LLM i LLMServer per a la gestió del servidor LLM. També es mostra l'ús de mòduls auxiliars com OCRModule i ImageProcessingModule, que proporcionen funcionalitats específiques dins del procés d'anàlisi d'imatges.

Aquestes relacions estan representades mitjançant diferents tipus de línies segons l'estàndard UML: línies contínues per a associacions i línies de punts per a dependències. Això permet entendre clarament quins components depenen dels altres i quins comparteixen una relació més estructural o col·laborativa.

La visualització de les classes i les seves relacions contribueixen a validar que el disseny de l'aplicació compleix amb els requisits funcionals, facilitant la seva traçabilitat i posteriorment manteniment.

A continuació es descriuen breument les principals classes representades al diagrama i les relacions que mantenen entre elles:

- **AppMain:** Classe principal que gestiona el flux de l'aplicació. Té associació amb AppLayout (estructura visual) i dependència directa de HybridOCR, QueryLLM i LLMServer, als quals delega funcionalitats específiques com l'anàlisi d'imatge, la consulta a models LLM i la gestió del servidor respectivament.
- **AppLayout:** Component encarregat de la interfície d'usuari. És usat per AppMain per mostrar i organitzar els resultats.
- **HybridOCR:** Mòdul especialitzat en reconeixement òptic de caràcters i en la separació de l'enunciat i la resposta d'una mateixa imatge. S'encarrega principalment de processar imatges i extreure text. Manté una associació amb OCRModule (que realitza les inferències OCR) i amb ImageProcessingModule (que segmenta i prepara les imatges).
- **QueryLLM:** Classe responsable d'enviar consultes textuals a l'LLM. És invocada per AppMain quan es vol generar una valoració sobre el text reconegut a la imatge.
- **LLMServer:** S'encarrega d'iniciar i aturar el servidor del model LLM, sent cridat per AppMain quan cal gestionar el backend.
- **OCRModule i ImageProcessingModule:** Són mòduls auxiliars que encapsulen funcionalitats concretes dins del procés d'anàlisi d'imatges, i són ambdós usats per HybridOCR.

5.2 Diagrama de seqüències

A continuació es presenten els diagrames de seqüència corresponents als casos d'ús identificats en l'apartat 4.1.1, mostrant la comunicació entre els diferents components principals del sistema per realitzar cadascuna de les funcionalitats de l'aplicació.

CD01: Carregar una imatge

El diagrama de seqüència per al cas d'ús "Carregar una imatge" mostra el procés mitjançant el qual l'usuari carrega una imatge des del seu propi dispositiu. El flux comença a partir de l'usuari en el moment que fa clic al botó de selecció d'imatge "Select Image", el qual invoca una finestra per seleccionar la imatge de la pregunta i resposta que es vulguin corregir. Un cop seleccionada la imatge, el sistema mostra una previsualització d'aquesta. En aquest cas les interaccions es realitzen entre l'usuari, l'AppLayout i AppMain, representant com el sistema rep, emmagatzema temporalment i mostra la imatge carregada per analitzar-la posteriorment.

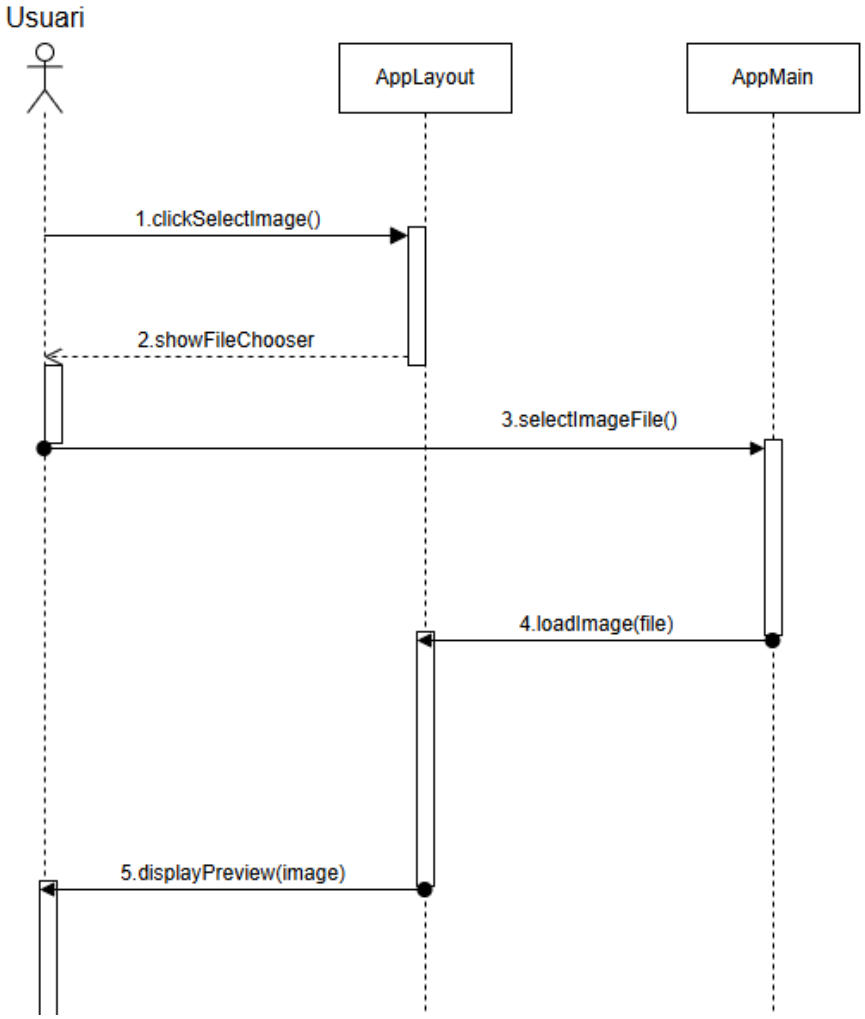


Figura 3. Diagrama de seqüencies del CD01.

CD02: Extracció del contingut d'una imatge

El diagrama de seqüència per al cas d'ús "Extracció del contingut d'una imatge" mostra com el sistema processa la imatge carregada per extreure el text manuscrit i l'enunciat. El procés comença quan l'AppMain rep la imatge carregada, que seguidament es segmenta en enunciat i resposta. A continuació, el sistema aplica l'OCR per extraure el text. El diagrama mostra el flux de missatges entre AppMain, HybridOCR, ImageProcessingModule i OCRModule per a realitzar aquestes operacions. Finalment, el sistema desa temporalment la informació extreta per a la seva posterior utilització.

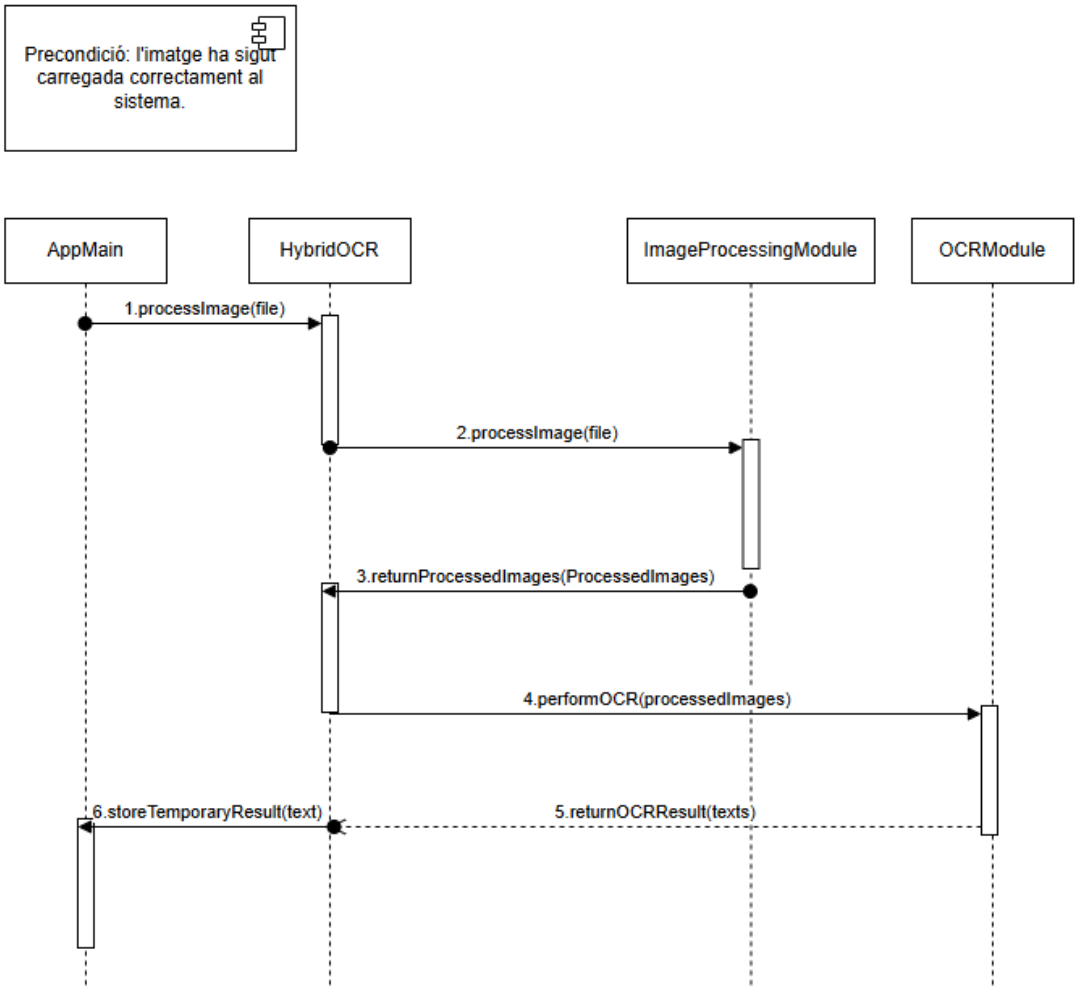


Figura 4. Diagrama de seqüencies del CD02.

CU03: Generació del prompt per a l'LLM

El cas d'ús "Generació del prompt per a l'LLM" explica com el sistema construeix un prompt que inclou la pregunta, la resposta i instruccions específiques per a l'LLM. Un cop el text ha estat extret i classificat (enunciat i resposta), AppMain accedeix a les dades i demana a QueryLLM crear un prompt estructurat a partir de la informació extreta al cas d'ús anterior. Aquest prompt es prepara per ser enviat a l'LLM i s'emmagatzema temporalment.

El diagrama de seqüències mostra les comunicacions entre AppMain, HybridOCR i QueryLLM, així com la creació del prompt que serà enviat posteriorment al model LLM.

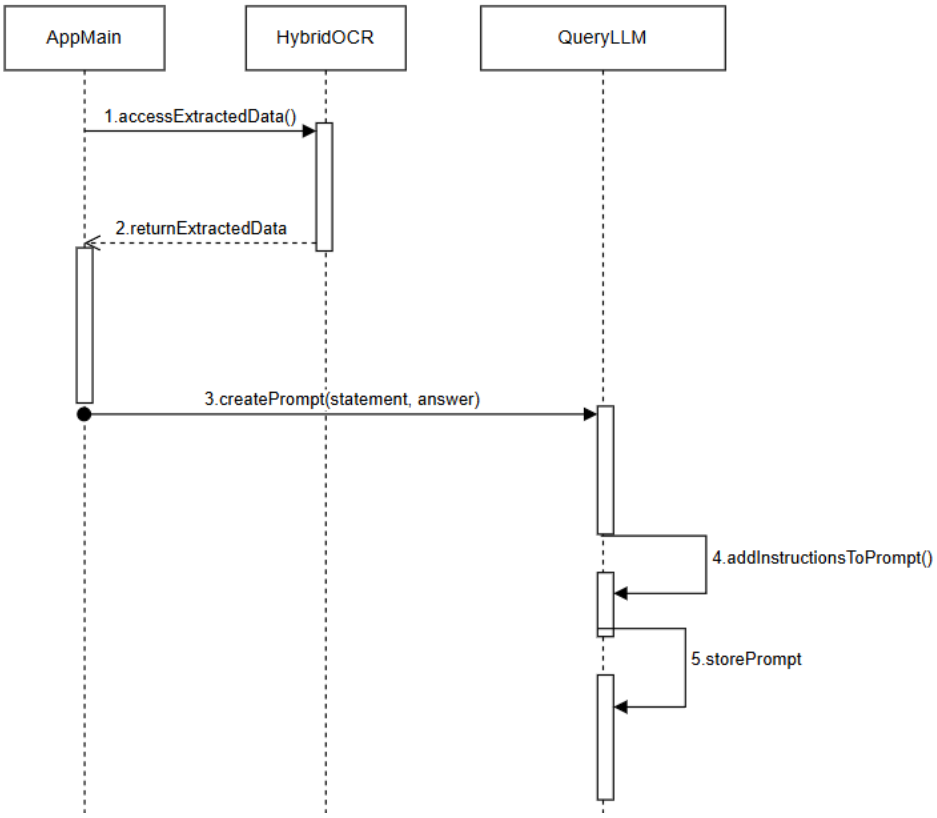


Figura 5. Diagrama de seqüencies del CD03.

CU04: Correcció de la resposta

El diagrama de seqüències per al cas d'ús "Correcció de la resposta" mostra com el sistema envia el prompt a l'LLM i rep la correcció. Un cop el prompt ha estat generat correctament, QueryLLM envia la consulta a l'LLM (gestionat per LLMServer). El model processa la informació, genera la correcció amb la justificació, i el sistema rep i emmagatzema temporalment aquesta correcció per a poder-la mostrar més tard. Aquest flux és gestionat per AppMain, QueryLLM, LLMServer, i el diagrama mostra la interacció entre aquests components per dur a terme la correcció.

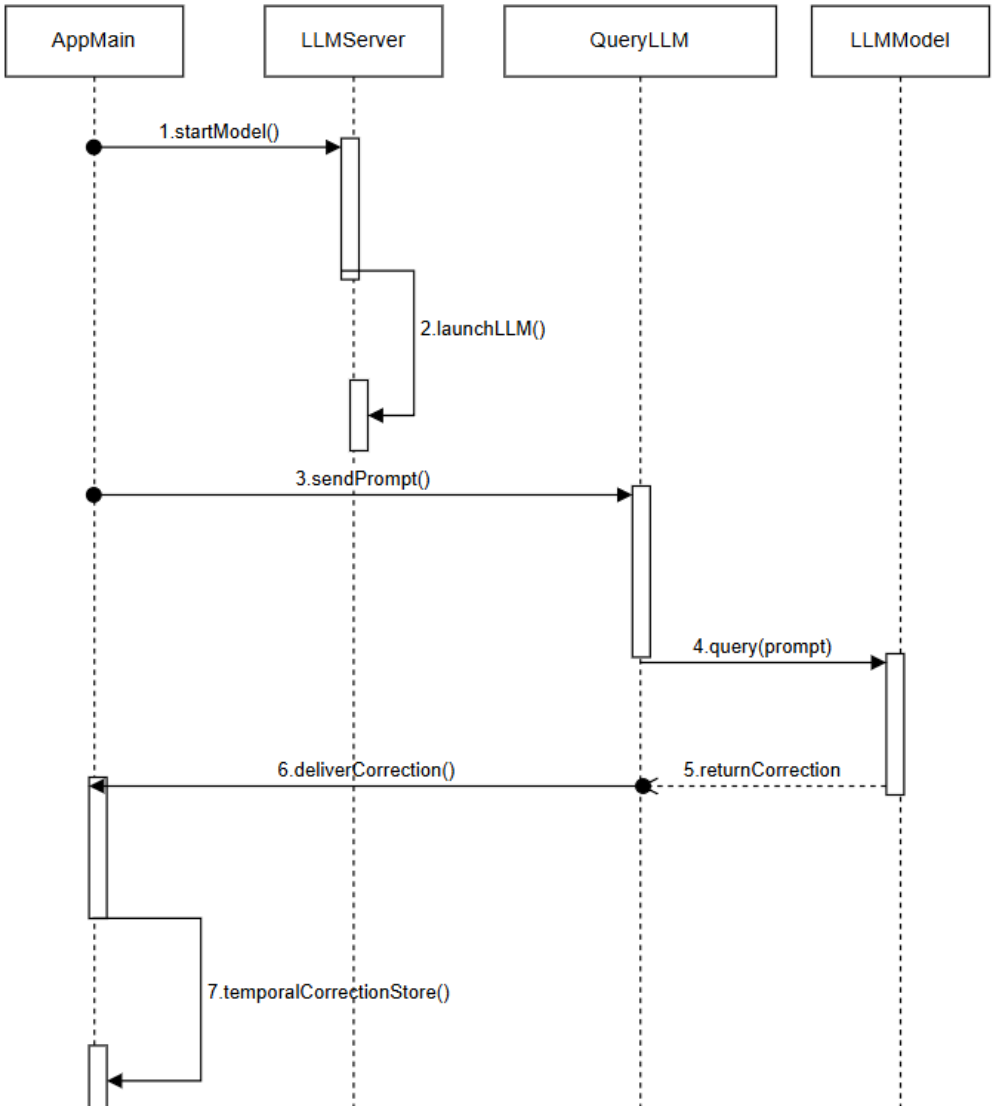


Figura 6. Diagrama de seqüencies del CD04.

CU05: Presentació de resultats

Finalment, el cas d'ús "Presentació dels resultats" mostra com l'usuari visualitza la correcció i els comentaris del model. Un cop el sistema ha rebut i emmagatzemat la correcció, AppMain la passa a AppLayout per a la seva presentació a l'usuari. AppLayout mostra la resposta corregida, la nota total de la pregunta i el comentari generat per l'LLM. El flux mostra com el sistema es comunica amb l'usuari per visualitzar els resultats finals del procés de correcció.

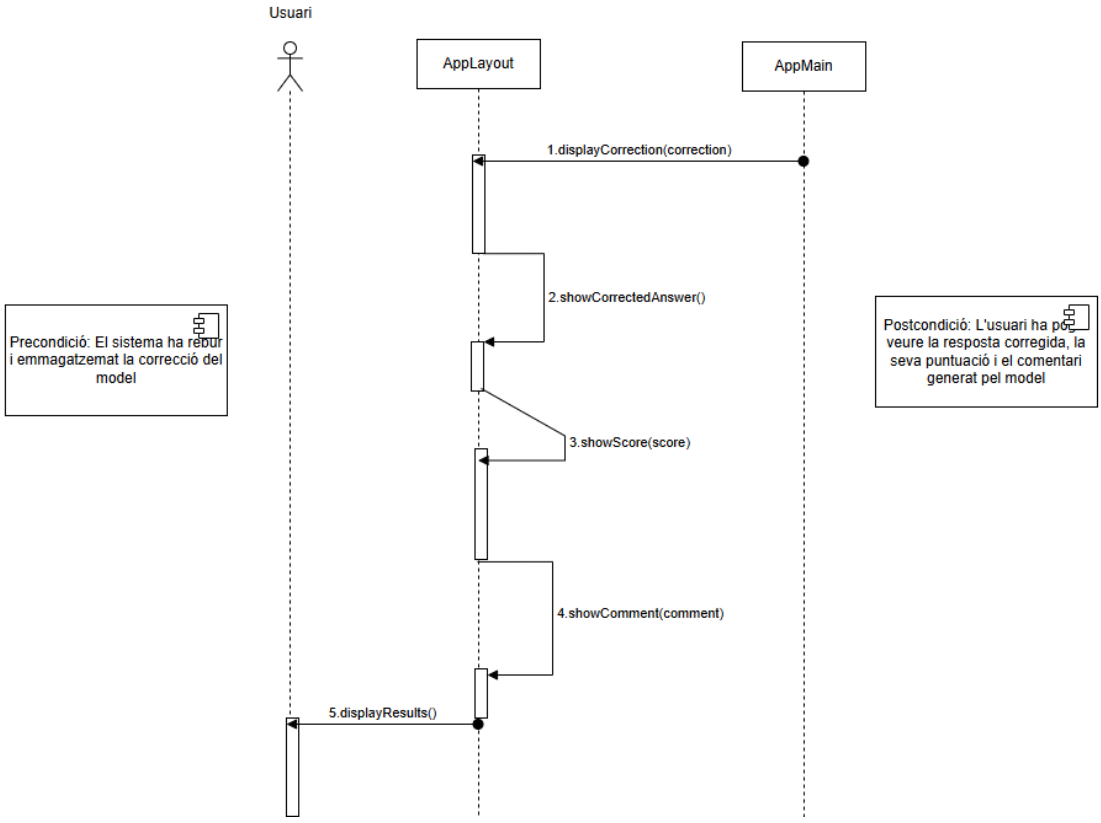


Figura 7. Diagrama de seqüencies del CD05.

6 Disseny

En aquest apartat s'explica el disseny de l'aplicació desenvolupada, abordant la seva arquitectura, l'estructura de la interfície gràfica d'usuari, entre altres components. Aquest disseny s'ha elaborat a partir dels requisits funcionals i no funcionals especificats en l'apartat de "Requisits", i es basa en els diagrames UML i l'anàlisi realitzada en l'apartat de "Anàlisi dels requisits funcionals".

6.1 Arquitectura de l'aplicació

L'aplicació ha estat dissenyada per tal de seguir una arquitectura modular basada en la separació de responsabilitats. Cada component o mòdul s'encarrega d'una funcionalitat o conjunt de funcionalitats del mateix àmbit específiques dins del sistema. S'ha realitzat amb aquesta metodologia amb l'objectiu de garantir la claredat de responsabilitats, la mantenibilitat del codi i l'escalabilitat del projecte.

Els principals blocs de l'arquitectura són els següents:

- **AppMain:** Mòdul principal que coordina tot el sistema, dirigint les interaccions entre els components i gestionant l'estat de l'aplicació durant el cicle d'ús i execució.
- **AppLayout:** Component encarregat de la interfície gràfica d'usuari mostrant la informació visual de l'aplicació i recollint les interaccions de l'usuari. I també responsable de fer visible el progrés de la correcció, mostrar el resultat d'aquesta i de la gestió dels components de la interfície.
- **HybridOCR:** Mòdul responsable de processar les imatges carregades, segmentant-les per separar l'enunciat de la resposta i encarregat, també, de realitzar el reconeixement de caràcters a cadascun dels dos blocs (enunciat i resposta).
- **QueryLLM:** Mòdul que construeix el prompt a partir de l'enunciat i resposta reconegudes a la imatge, i que posteriorment envia al model LLM.
- **LLMServer:** Component dedicat a la posada en marxa i aturada del servidor local de l'LLM.
- **OCRModule:** Component auxiliar que utilitza HybridOCR per tal d'utilitzar funcions específiques de models OCR diferents.
- **ImageProcessingModule:** Component auxiliar que utilitza HybridOCR per tal de poder realitzar el processament d'imatges i la seva segmentació.

Tots els mòduls es comuniquen mitjançant crides directes entre aquests, mantenint l'acoblament al mínim. A continuació es mostra una imatge representativa del flux de funcionament d'aquest sistema.

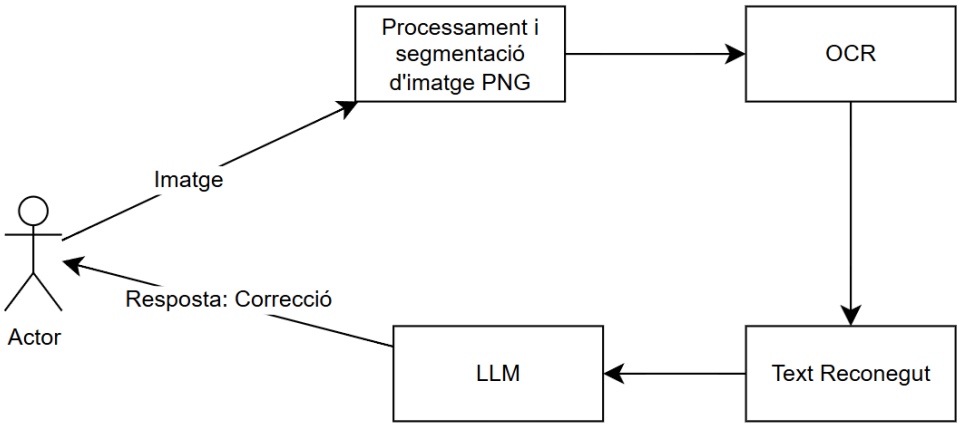


Figura 8. Diagrama del flux d'execució

6.2 Disseny de la interfície gràfica

La interfície gràfica d'usuari (GUI) ha estat dissenyada per proporcionar una experiència intuïtiva i senzilla. L'objectiu és garantir que l'usuari pugui interactuar de la forma més eficient possible amb el sistema.

Els principals components de la GUI són:

- **Botó de càrrega d'imatge:** Element que permet a l'usuari seleccionar la imatge a analitzar des del seu dispositiu. Aquest botó activa el procés de càrrega i prepara la imatge per al processament.
- **Àrea de previsualització de la imatge:** Element que mostra una vista prèvia de la imatge carregada per l'usuari.
- **Secció dels resultats OCR:** Aquesta secció mostra el text extret de la imatge mitjançant els models OCR. L'usuari pot veure l'enunciat i la resposta reconeguda.
- **Secció de la nota obtinguda:** Aquesta secció, un cop el sistema ha completat l'anàlisi i la correcció mostra la nota obtinguda per l'LLM sobre la resposta reconeguda. La nota s'expressa de manera fàcil i senzilla per a millor comprensió de l'usuari.
- **Secció de la correcció LLM:** Aquesta secció mostra la correcció realitzada pel LLM sobre la resposta. Aquesta correcció inclou, a més, comentaris justificatius que expliquen les raons darrere de la valoració, proporcionant així una explicació més detallada de com s'ha arribat a la valoració final.
- **Indicadors de progrés:** Diferents etiquetes en les altres seccions que indicaran el progrés de cadascun dels processos.

6.3 Disseny de la persistència de dades

L'aplicació no ha estat dissenyada per tal de tenir persistència de dades degut a que l'enfocament que s'ha seguit és orientat a una sessió d'usuari de manera efímera. Tot i això, s'utilitzen diferents mecanismes d'emmagatzematge temporal d'informació necessària per a realitzar les inferències dels models OCR i LLM.

Aquestes dades emmagatzemades temporalment són les següents:

- **Imatges carregades:** Les imatges seleccionades per l'usuari es desaran temporalment per poder-les processar.
- **Text extret:** El text identificat de la imatge (enunciat i resposta) es guarda temporalment per a posteriorment ser utilitzades en la construcció del prompt per a la query de correcció.
- **Prompt generat:** El sistema desa el prompt creat per enviar-lo a l'LLM i així poder rebre una correcció.
- **Respostes generades:** La resposta corregida per part de l'LLM es desarà per a la seva visualització a la GUI.

En cas de necessitar una persistència més permanent en el futur (per exemple, per a la gestió d'històrics o emmagatzemament de dades de diverses sessions), la infraestructura dissenyada permetria una integració senzilla d'una base de dades sense afectar el funcionament principal.

7 Implementació

La implementació del projecte s'ha construït sobre diferents mòduls amb funcionalitats independents, amb l'objectiu d'assegurar la mantenibilitat, l'escalabilitat i la reutilització del codi. Tot el sistema s'ha desenvolupat utilitzant el llenguatge de programació Python, gràcies a la seva versatilitat i la disponibilitat de llibreries especialitzades per a visió per computador i OCR. El sistema està composat per tecnologies d'OCR, visió per computador, processament de llenguatge natural (NLP[3]), una interfície d'usuari per consola i una altra de gràfica.

7.1 Arquitectura

El sistema es composa de les següents capes de lògica principals:

- **Capa de processament:** Aquesta capa té la responsabilitat de processar i manipular les imatges per al seu ús correcte i l'aplicació de tècniques OCR per a l'extracció de text.
- **Capa d'entrada/sortida:** Aquesta capa gestiona les interaccions amb l'usuari utilitzant tant la interfície per consola com la interfície gràfica.
- **Capa semàntica:** Aquesta capa s'encarrega de la comprensió del text obtingut mitjançant l'ús de tècniques LLM.

7.2 Estructura i funcionalitat dels components

En aquest subapartat s'enumeren i expliquen tots els components i funcionalitats d'aquests que s'han fet servir per a la construcció del sistema de correcció automàtic de preguntes d'exàmens:

7.2.1 app_main.py i app_main_console.py

Aquest dos fitxers són els coordinadors principals de la lògica de l'aplicació, a més són els dos únics punts d'entrada al sistema.

El primer, app_main.py constitueix la versió de l'aplicació amb interfície gràfica, implementada a partir de la llibreria Tkinter, proporcionant funcionalitats visuals còmodes per a la càrrega d'imatges i visualització dels resultats.

Ambdós s'encarreguen d'iniciar i aturar el servei LLM per tal de gestionar-ne el seu ús. També són els responsables d'invocar al mòdul image_processing_module.py per fer el processament de la imatge, segmentant-la i fent la inferència OCR de manera immediata.

Un dels components clau de app_main.py és la funció process_image_and_evaluate(), la qual gestiona el flux de treball de l'aplicació:

- Mostra la imatge seleccionada en la interfície gràfica.
- Invoca la funció hybrid_ocr_inference del mòdul imatge_processing_module.py per obtenir el text de l'enunciat i la resposta a partir de la imatge.
- Realitza la consulta a Ollama per obtenir l'avaluació de l'LLM, que inclou la puntuació i comentaris.

[3] NLP: Processament de llenguatge natural o *Natural Language Processing*

- Actualitza la interfície gràfica amb els resultats obtinguts: el text de l'enunciat, la resposta, la puntuació i els comentaris de l'avaluació.

7.2.2 app_layout.py

Aquest fitxer defineix l'estructura i la disposició de la interfície gràfica. Controla com es mostren cadascun dels elements de la finestra principal de l'aplicació: Selecció i visualització de la imatge, camps de text on es mostren les inferències OCR tant de l'enunciat com de la resposta i per últim la visualització dels resultats obtinguts en la correcció de l'LLM (Nota sobre 10, Comentari avaluatiu i etiqueta de correcte o incorrecte).

Per a la seva implementació s'utilitza la llibreria Tkinter, amb els següents components destacats:

- **Tk(), Label, Button, Frame, Text**: Components usats per crear la finestra principal i organitzar els elements de manera jeràrquica.
- **Canvas i Scrollbar**: Components que gestionen el desplaçament vertical en interfícies de gran longitud, millorant l'experiència d'usuari.
- **Filedialog.askopenfilename:** Component per permetre l'usuari seleccionar una imatge del seu dispositiu.
- **Protocol("WM_DELETE_WINDOW", ...):** Component per a la gestió controlada del tancament de l'aplicació.

Aquest enfocament amb Tkinter proporciona una interfície senzilla i funcional per a la interacció amb el sistema OCR+LLM.

A continuació es mostra la disposició de la interfície final amb un exemple de correcció realitzada:

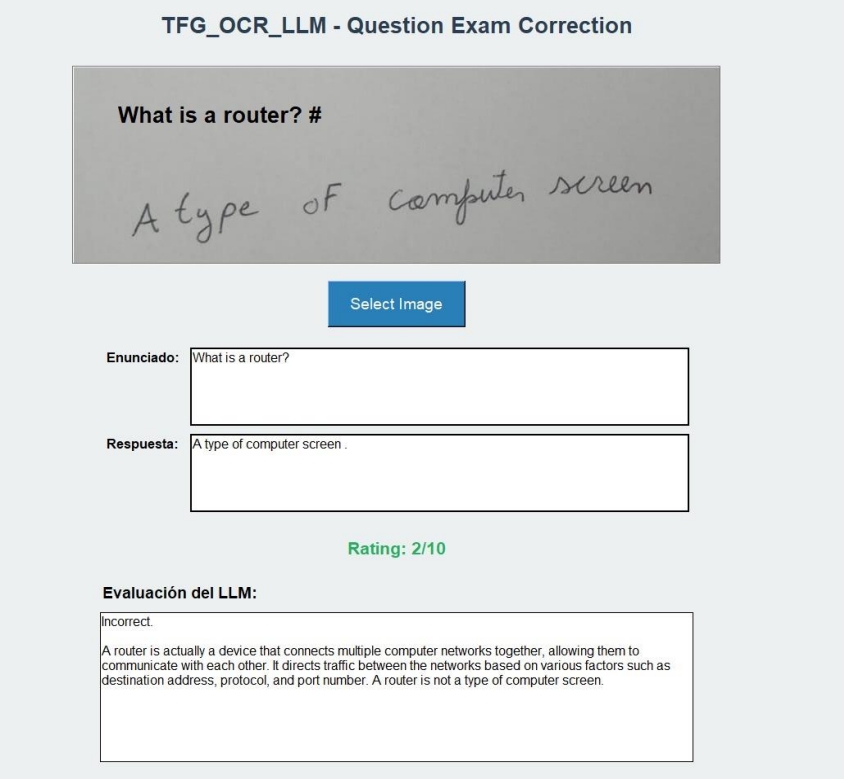


Figura 9. Interfície gràfica de l'aplicació.

7.2.3 image_processing_module.py

Aquest mòdul conté tota la lògica relacionada amb el preprocessament d'imatges, la segmentació de línies i l'execució híbrida de l'OCR, mitjançant Tesseract i TrOCR.

Funcionalitats principals:

- Conversió i preprocessament de la imatge (escala de grisos, binarització, morfologia).
- Segmentació de línies mitjançant contorns.
- Aplicació d'OCR híbrid: Tesseract per al text imprès (enunciat) i TrOCR per al text manuscrit (resposta).

A continuació, es descriu el funcionament detallat de les funcions principals de mòdul:

Segmentació de línies: segment_lines(image_path)

Aquesta funció s'encarrega de dividir la imatge d'un examen en fragments corresponents a línies de text individuals. Aquest pas és essencial per dos motius, el primer és per poder separar l'enunciat de la resposta i el segon és perquè el model de TrOCR (base-handwritten) treballa òptimament a nivell de línia.

El procés segueix les etapes següents:

- **Conversió a escala de grisos:** Permet simplificar la imatge reduint-la a tons de gris, facilitant el processament posterior.

Figura 10. Imatge processada a escala de grisos.

- **Binarització:** Mitjançant la tècnica de llindar adaptatiu d'Otsu, transforma la imatge en blanc i negre invertits, destacant les zones amb text.

Figura 11. Imatge Binaritzada.

- **Operacions morfològiques:** L'aplicació de dilatació i erosió permet unir zones de text i eliminar petits sorolls.

Figura 12. Regions detectades i agrupades en blocs continus.

- **Detecció de contorns:** Es detecten les àrees més significatives (línies) i s'ordenen de dalt a baix per preservar l'ordre lògic del text.
- **Extracció i emmagatzematge temporal:** Cada línia detectada es desa com a fitxer d'imatge temporal per al seu processament posterior.

Figura 13. Secció detectada contenidora de l'enunciat.

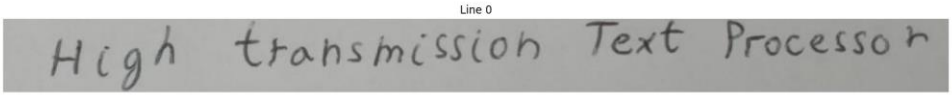


Figura 14. Secció detectada contenidora de la resposta manuscrita.

Extracció de text híbrida: hybrid_ocr_inference(imatge_path)

Aquesta funció és la responsable de combinar les dues tècniques d'OCR per extreure el contingut complet d'una imatge d'una pregunta d'examen. El seu funcionament es divideix en els següents passos:

- **Segmentació prèvia de línies:** Primer, la imatge es passa pel mètode segment_lines() explicat prèviament, que genera imatges temporals per a cada línia detectada.
- **Extracció de l'enunciat:** El text de l'enunciat s'extreu directament de la imatge original utilitzant Tesseract, ja que normalment es tracta de text imprès, més fàcil de reconèixer amb aquest motor. El text es talla just abans del primer símbol clar de separació "#".
- **Detecció del punt de canvi a resposta:** A mesura que es recorren les línies segmentades, es comprova si es conté el caràcter "#". Quan es troba, s'interpreta que a partir d'aquella línia comença la resposta manuscrita.
- **Extracció de la resposta amb TrOCR:** Les línies posteriors al punt de canvi es processen una a una amb TrOCR (base-handwritten), un model d'aprenentatge profund especialment entrenat per reconèixer escriptura manual. El text reconegut de cada línia s'afegeix a la resposta completa.
- **Neteja i eliminació de fitxers temporals:** Un cop extret tot el text, la funció s'encarrega d'esborrar les imatges temporals que s'han creat per tal d'optimitzar l'ús de recursos.

- **Retorn de resultats:** Finalment, la funció retorna dues cadenes de text: l'enunciat (obtingut amb tesseract) i la resposta (obtinguda amb TrOCR), llestes per ser analitzades i/o visualitzades.

7.2.4 ocr_module.py

Aquest mòdul encapsula diverses estratègies d'extracció de text emprant diferents motors i biblioteques de reconeixement. L'objectiu d'aquest mòdul és oferir una interfície unificada i flexible per extreure text de documents escanejats o fotografiats, tant si és imprès com manuscrit.

Funcionalitats principals:

- **Tesseract OCR:** Utilitzat principalment per reconèixer text imprès. S'hi inclouen múltiples variants:
  - **text_extraction_tesseract():** Aplicació bàsica amb paràmetres predefinits. És la funció que s'empra efectivament en l'extracció del text de l'enunciat.
  - **text_extraction_tesseract_config():** Aplicació que permet especificar opcions personalitzades de configuració del motor.
  - **text_extraction_tesseract_preproc():** Aplicació que incorpora un pas previ de preprocessament en cas de que fos necessari.
- **TrOCR (Transformers OCR):** El motor principal per a reconèixer text manuscrit.
  - **text_extraction_trocr():** Funció que redimensiona la imatge a 384x384 píxels, genera les prediccions amb VisionEncoderDecoderModel i decodifica el resultat. Aquesta eina s'ha integrat en el pipeline híbrid per tal de millorar l'eficàcia amb respostes escrites a mà.
- **OCR alternatius (comentats o disponibles com a opció):** Tot i que no s'utilitzen en la versió final del sistema, el mòdul també inclou la inicialització de diverses biblioteques OCR:
  - **EasyOCR**
  - **PaddleOCR**
  - **docTR**

Aquests mètodes es van considerar durant la fase d'experimentació per comparar el rendiment de diferents estratègies d'extracció de text. Finalment, es van escollir Tesseract (per al text imprès) i TrOCR (per al text manuscrit) com les opcions més adequades per a l'aplicació. Aquesta decisió es justifica en detall a l'apartat de l'avaluació.

7.2.5 llm_module.py

Aquest mòdul gestiona la interacció amb un LLM a través d'Ollama, amb l'objectiu d'avaluar respostes manuscrites donades a enunciats extrets mitjançant OCR. Tot i que es mantenen algunes funcions amb GPT-2 per a proves, la versió final del sistema fa ús exclusiu del model LlaMa2 d'Ollama.

Funcionalitats principals:

- **Plantilla de prompt (PROMPT_TEMPLATE_TEST):** Es defineix una única plantilla per construir els missatges enviats al model. Aquesta plantilla contextualitza la tasca com si l'LLM fos un corrector d'exàmens, i estructura la resposta en el format: "[Correct|Incorrect] – Score: x/10 – justificació breu"

  L'enunciat i la resposta extretes s'inclouen dins del prompt per garantir una avaluació precisa basada en el contingut.

- **query_app_ollama():** Funció encarregada d'enviar la sol·licitud al model LlaMa2 a través d'Ollama, utilitzant la plantilla anterior. Aquesta funció rep com a entrada l'enunciat i la resposta de l'usuari, i retorna una valoració textual que inclou l'etiqueta de correcció, puntuació i justificació.

Aquest enfocament permet delegar la correcció a l'LLM, oferint una avaluació coherent i estructurada que s'adapta al context de cada exercici.

7.2.6 ollama_server.py

Aquest mòdul s'encarrega de gestionar l'execució del servidor local d'Ollama, assegurant-se que estigui en funcionament quan l'aplicació ho necessita i detenint-lo correctament quan finalitza l'execució.

Funcionalitats principals:

- **is_ollama_running():** Comprova si el servidor d'Ollama està actiu a la màquina local (per defecte al port 11434).
- **start_ollama():** Inicia el servidor d'Ollama mitjançant una crida a subprocess, si no s'està executant en el moment. Inclou una petita espera per garantir que el servei estigui llest abans de continuar.
- **stop_ollama():** Finalitza el procés del servidor si ha estat iniciat per aquest script. Aquesta funció s'enregistra automàticament perquè s'executi en sortir del programa, utilitzant atexit.register.

Aquest mòdul és especialment útil en entorns locals on es vol garantir que Ollama estigui disponible de manera controlada i automatitzada durant l'execució del sistema, evitant la necessitat d'iniciar-lo manualment.

8 Avaluació

En aquest apartat es descriuen els processos d'avaluació seguits per tal de validar i garantir el correcte funcionament del sistema corrector automàtic. L'avaluació està dividida en dues fases principals:

Una primera fase prèvia al desenvolupament del sistema, centrada en l'anàlisi comparativa de diversos motors OCR per seleccionar el més adequat per al reconeixement de text manuscrit.

I una segona fase posterior a la implementació, dedicada a validar la segmentació de les imatges de les preguntes d'exàmens i la capacitat de l'LLM per corregir respostes manuscrites, utilitzant un conjunt de dades propi basat en preguntes d'examen.

Aquest procés d'avaluació ha sigut clau per assegurar que cada component del sistema (OCR, segmentació i correcció) funciona amb la precisió necessària per als casos d'ús plantejats, així com per identificar possibles limitacions i àrees de millora.

8.1 Selecció de l'OCR: Proves prèvies al desenvolupament del sistema

Aquest apartat realitza una avaluació comparativa entre diferents motors OCR: PaddleOCR, Tesseract, EasyOCR, docTR i TrOCR, utilitzant el dataset IAM Handwriting en diferents formats (línies i frases). Aquestes proves estan orientades a determinar el motor amb millor rendiment en la tasca d'extracció de text manuscrit.

Motors analitzats:

- **PaddleOCR:** Toolkit multilingüe d'OCR basat en deep learning, dissenyat per oferir eines pràctiques per a la formació i aplicació de models OCR.
- **Tesseract:** Motor OCR de codi obert desenvolupat originalment per Hewlett-Packard i actualment mantingut per Google. Utilitza una arquitectura basada en LSTM per al reconeixement de text.
- **EasyOCR:** Llibreria OCR basada en Python que admet més de 80 idiomes incloent-hi llatí, xinès, entre d'altres.
- **docTR:** Llibreria OCR basada en deep learning que proporciona una solució d'extrem a extrem per al reconeixement de text en documents, utilitzant arquitectura de transformers.
- **TrOCR:** Model OCR basat en transformers que combina models preentrenats de visió per computador i processament de llenguatge natural per al reconeixement de text manuscrit.

Per a cada motor OCR, es va processar un conjunt de 100 línies i 100 frases extretes del data set IAM Handwriting. Les mètriques analitzades inclouen:

- Total de línies processades.
- Errors detectats.
- Presició mitjana (%): Proporció de caràcters reconeguts correctament.

- WER[4] (Word Error Rate): Proporció d'errors a nivell de paraula.
- CER[5] (Character Error Rate): Proporció d'errors a nivell de caràcter.
- Temps d'execució.
- Taxa de prediccions buides (%): proporció de línies sense sortida OCR.

8.1.1 Resultats obtinguts en les proves

En aquesta secció es presenten els resultats obtinguts a partir de les proves realitzades amb els diversos motors OCR utilitzats per a l'extracció de text manuscrit, utilitzant el dataset IAM Handwriting en els formats de línies i frases. Els resultats s'han analitzat en funció de les mètriques clau, com la precisió mitjana, el Word Error Rate, el Character Error Rate, el temps d'execució i la taxa de prediccions buides, amb l'objectiu de determinar quin motor OCR ofereix el millor rendiment per a la tasca específica de reconeixement de text manuscrit.

A continuació, es mostren les mètriques obtingudes per cada motor OCR per als dos formats de dades: línies manuscrites i frases manuscrites. Els resultats permeten comparar l'eficàcia de cada motor en termes de precisió, velocitat i capacitat per a generar prediccions vàlides, proporcionant uns fonaments per a la selecció del motor OCR més adequat per a l'aplicació final.

IAM Handwriting lines

A la Taula 1 de resultats per a les línies manuscrites es pot observar que TrOCR és el motor que ha obtingut la millor precisió mitjana, amb un 96.89%, seguit per EasyOCR amb un 43.18% de precisió mitjana. En termes de WER i CER, TrOCR també sobresurt amb els valors més baixos (15.2% i 24.47%, respectivament), indicant una millor capacitat per reconèixer text manuscrit de forma precisa.

OCR	Imatges processades	Precisió mitjana (%)	WER mitjà (%)	CER Mitjà(%)	Temps d'execució (min)	Taxa Prediccions buides (%)
EasyOCR	100	43.18	112.64	69.37	4.1	0
PaddleOCR	100	38.77	97.31	71.34	0.52	9
TrOCR	100	96.89	15.2	24.47	24.12	0
docTR	100	33.22	102.1	75.16	5.64	3
Tesseract	100	33.65	112.95	86.94	0.28	20

Taula 1. Taula resum de les mètriques avaluades sobre els models OCR amb IAM Handwriting lines.

[4] WER: *Word Error Rate.*

[5] CER: *Character error Rate.*

A la Figura 15, es pot veure gràficament aquesta distribució de precisió mitjana obtinguda pels motors OCR per a IAM Handwriting lines.

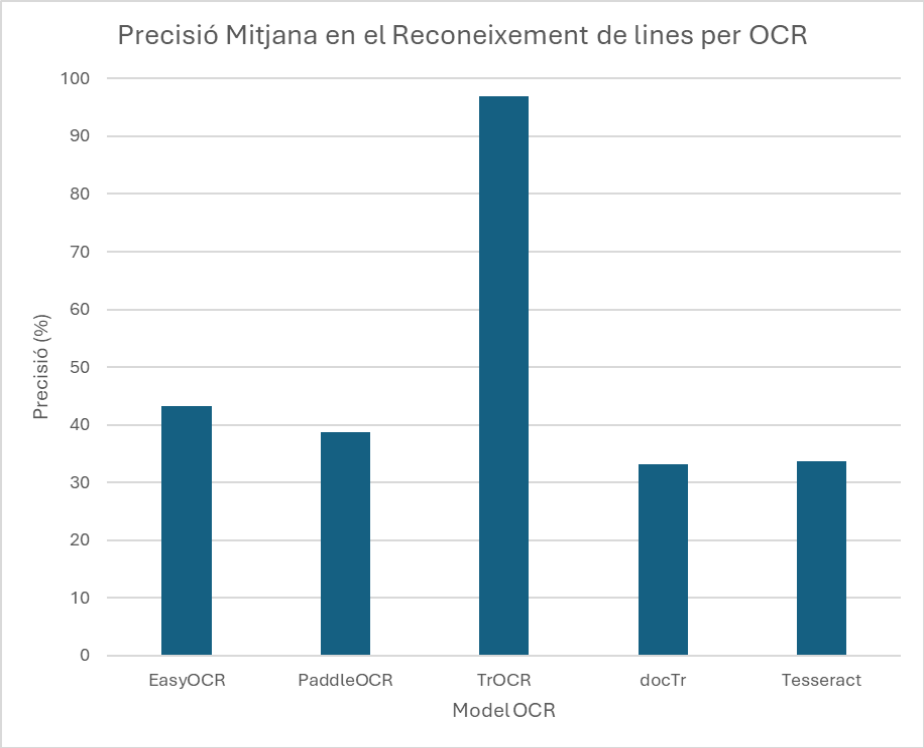


Figura 15. Gràfic de la precisió mitjana obtinguda per a IAM Handwriting lines.

Els resultats del temps d'execució per al processament de les imatges de línies manuscrites es presenten en la Figura 16. Com es pot veure, TrOCR és el motor amb un temps d'execució més alt (24.12 minuts), seguit per docTR (5.64 minuts) i EasyOCR (4.1 minuts). Això reflecteix el compromís entre la precisió i la velocitat de processament, amb TrOCR oferint una excel·lent precisió a costa d'un temps de processament més llarg.

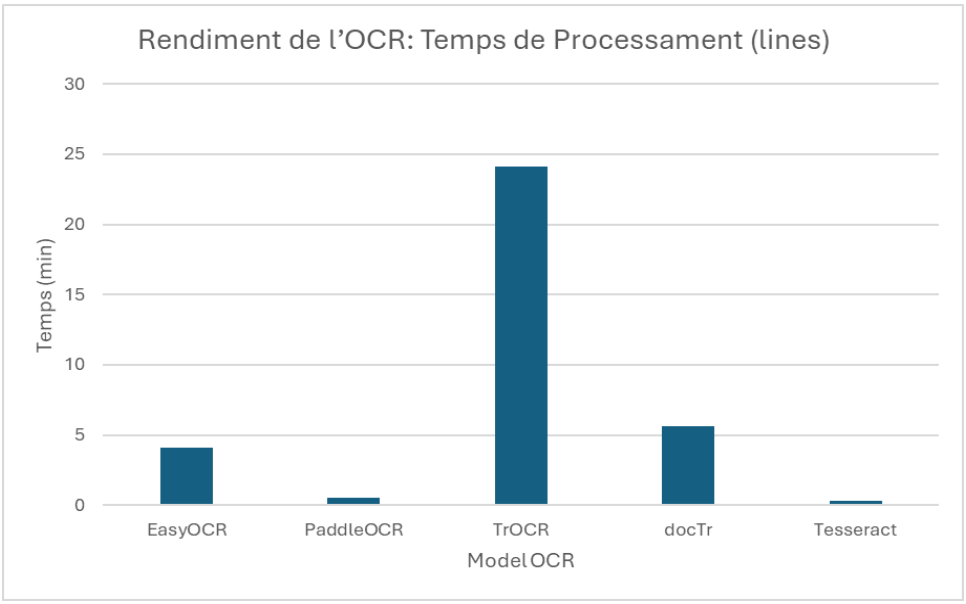


Figura 16. Gràfic del temps d'execució requerit pels models OCR en processar 100 imatges de IAM Handwriting lines

IAM Handwriting sentences

Pel que fa a les frases manuscrites, els resultats segueixen una tendència similar, amb TrOCR destacant-se novament com el millor motor en termes de precisió mitjana (95.7%), amb una significativa millora respecte els altres motors.

OCR	Imatges processades	Precisió mitjana (%)	WER mitjà (%)	CER Mitjà(%)	Temps d'execució (min)	Taxa Prediccions buides (%)
EasyOCR	100	41.36	111.95	71.61	3.41	2
PaddleOCR	100	33.08	97.85	75.84	0.47	18
TrOCR	100	95.7	18.28	26.73	20.71	0
docTR	100	31.27	105.49	77.6	5.64	5
Tesseract	100	34.49	114.68	86.38	0.28	18

Taula 2. Taula resum de les mètriques avaluades sobre els models OCR amb IAM Handwriting sentences.

La Figura 17 mostra aquest comportament de manera visual. D'altra banda, EasyOCR presenta un rendiment moderat en precisió (41.36%), amb una taxa de prediccions buides mínima, igual que a les línies manuscrites.

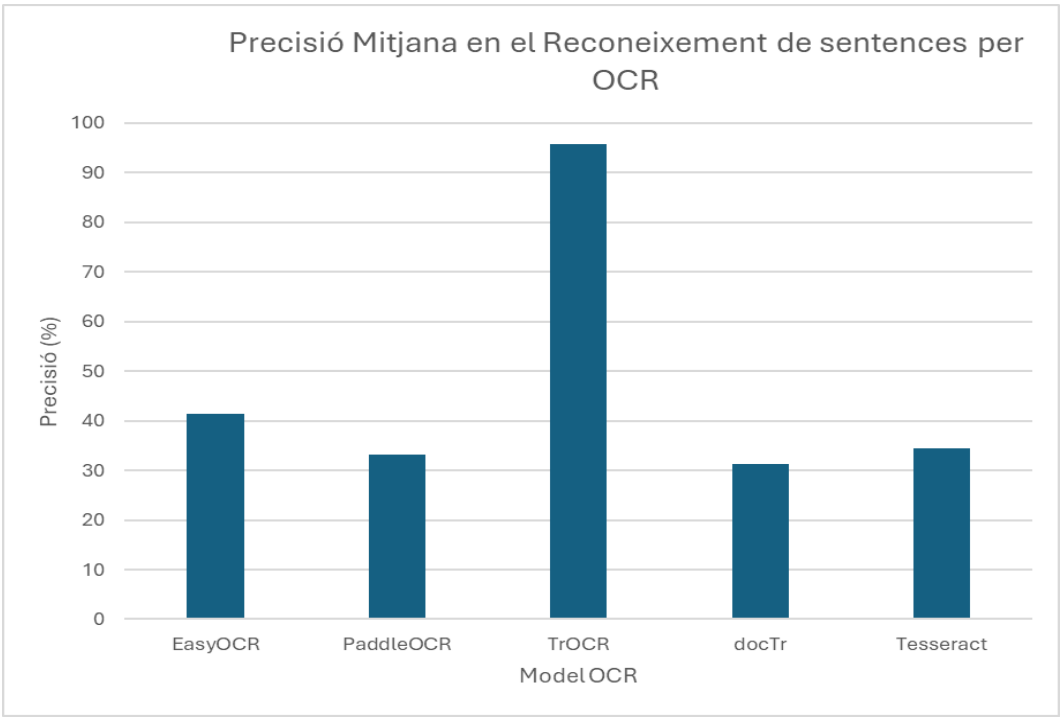


Figura 17. Gràfic de la precisió mitjana obtinguda per a IAM Handwriting sentences.

Els resultats del temps d'execució per al processament de les imatges de frases manuscrites es mostren a la Figura 18. Com en el cas de les línies manuscrites, TrOCR requereix més temps per processar les imatges (20.71 minuts), mentre que altres motors com PaddleOCR i Tesseract presenten temps d'execució més curts, sent Tesseract el més ràpid (0.28 minuts).

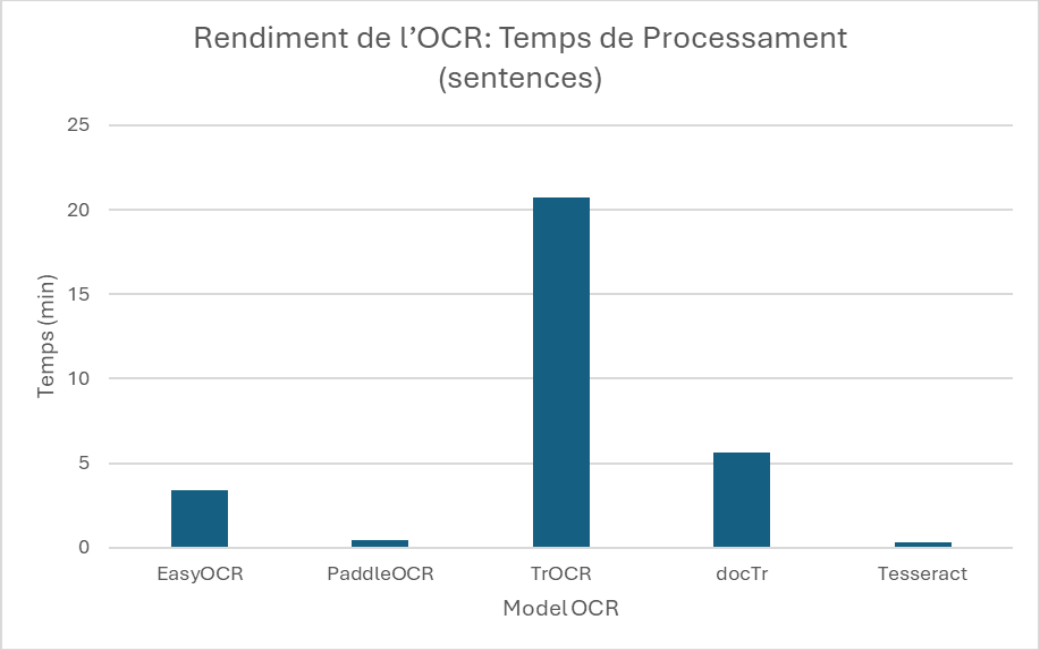


Figura 18. Gràfic del temps d'execució requerit pels models OCR en processar 100 imatges de IAM Handwriting sentences.

8.1.2 Conclusió de les proves de selecció d'OCR

En resum, TrOCR es presenta com el motor més precís en ambdós formats (línies i frases manuscrites), tot i que la seva superioritat en termes de precisió s'acompanya d'un temps d'execució més elevat. Els altres motors, com EasyOCR i PaddleOCR, mostren un bon rendiment en termes de velocitat, però no arriben als nivells de precisió de TrOCR. Les figures presentades ajuden a visualitzar clarament aquestes diferències i proporcionen una base per a la selecció del millor motor OCR en funció de les necessitats específiques del projecte.

8.2 Validació post-implementació: Segmentació i correcció LLM

Aquest apartat descriu els processos de validació post-implementació realitzats per a garantir el correcte funcionament del sistema. En aquesta fase els test es centren en dos aspectes principals de l'aplicació: la segmentació de les imatges (enunciat i resposta) i la correcció mitjançant l'LLM LlaMa2.

Per a la validació del sistema, s'utilitza un conjunt de dades de prova personalitzat, compost per 63 imatges de preguntes impreses i respostes manuscrites. El conjunt està estructurat en format XML i defineix, per a cada imatge, els següents camps: la ruta al fitxer de la imatge, l'enunciat de la pregunta, la transcripció de la resposta i una etiqueta de validació (Correct o Incorrect) que indica si la resposta es considera adequada respecte l'enunciat.

El conjunt de dades incorpora una diversitat considerable de preguntes i respostes, tant en contingut com en qualitat. Inclou preguntes teòriques, com definicions o acrònims tècnics. Les respostes manuscrites cobreixen des d'enunciats correctes i vàlids fins a errors conceptuals, respostes fora de context, incompletes o deliberadament absurdes. Aquesta heterogeneïtat permet avaluar els sistema en escenaris variats, incloent situacions òptimes i casos límit, i garanteix una validació robusta tant de la segmentació OCR com del procés de correcció basada en l'LLM.

Tant les preguntes com les respostes estan redactades en anglès, amb l'objectiu de maximitzar el rendiment del sistema, especialment en la fase d'avaluació mitjançant l'LLM.

8.2.1 Validació de la segmentació de les imatges i reconeixement OCR

La segmentació de les imatges és un pas clau en el sistema, ja que permet dividir les respostes manuscrites en parts més petites i manipulables per a facilitar el seu posterior reconeixement i correcció. Aquesta segmentació és especialment rellevant quan es treballa amb imatges de respostes d'examen, on la precisió en la localització de cada fragment de la imatge incideix directament en la qualitat, precisió i coherència del reconeixement OCR.

Per realitzar la validació de la segmentació i el reconeixement OCR, s'ha realitzat un script automatitzat, anomenat test_segmentation.py, dissenyat per treballar amb el conjunt d'imatges explicat a la introducció d'aquest apartat. Aquest script permet avaluar la qualitat tant de la segmentació com del reconeixement OCR.

El funcionament de l'script consisteix a llegir el conjunt de proves des d'un fitxer XML, localitzar les imatges corresponents i aplicar els mòduls de processament d'imatges i OCR del sistema per obtenir automàticament l'enunciat i la resposta reconeguts. A continuació, compara les prediccions OCR amb les transcripcions de referència mitjançant tres mètriques de qualitat habituals: Word Error Rate, Character Error Rate i la precisió. Finalment, enregistra tots els resultats en un fitxer CSV, incloent-hi la ruta amb el nom de la imatge, els textos esperats i reconeguts, i els valors obtinguts per a cadascuna de les mètriques en l'enunciat i la resposta

A partir d'aquest registre, es calcula també una mitjana global per a cada mètrica, separant els resultats segons el tipus de text (enunciat o resposta manuscrita). Aquesta acció permet analitzar el rendiment del sistema de forma específica i detectar possibles fonts d'error associades a la segmentació o al reconeixement. A continuació es mostren els resultats de precisió en la separació i reconeixement dels enunciats i respostes:

- **Precisió mitjana en enunciats: 98.71%**

- **Precisió mitjana en respostes manuscrites: 87.52%**

Aquestes xifres reflecteixen l'eficàcia global del sistema en la segmentació i reconeixement OCR per a les dues parts dels texts. Tot i que els enunciats mostren una alta precisió, la precisió en les respostes manuscrites és lleugerament inferior. Aquest resultat és normal, tenint en compte que la lletra manuscrita és molt més variable que la lletra impresa. La variabilitat en l'escriptura manuscrita és inherent a les diferents formes de lletres que cada individu pot utilitzar, i fins i tot en una sola persona, l'escriptura pot presentar una gran diversitat estilística, tant en la mida, la inclinació com la forma de les lletres i altres factors. Per tant, la disminució de la precisió en el reconeixement de respostes manuscrites és un fenomen esperable i comprensible en sistemes OCR.

En relació amb la qualitat de les imatges utilitzades en aquest procés, cal destacar que el processament i la resolució de les imatges també poden tenir un impacte significatiu en els resultats obtinguts. Imatges de baixa qualitat, amb soroll, baixa resolució o il·luminació inadequada poden influir negativament la segmentació i el reconeixement del text.

8.2.2 Validació de la correcció automàtica amb el model de llenguatge

Un cop avaluats tant la segmentació com el reconeixement del enunciat i la resposta, el següent pas consisteix a aplicar la correcció mitjançant l'LLM LlaMa2 d'Ollama. Aquest model s'encarrega d'analitzar si la resposta manuscrita reconeguda respon adequadament a l'enunciat de la pregunta, hi ho fa seguint unes instruccions precises a través d'un prompt explicat a l'apartat 7.2.5, dissenyat específicament per simular el comportament d'un corrector d'exàmens.

Per validar la capacitat de l'LLM en aquesta tasca de correcció, s'ha implementat un sistema automatitzat que segueix un conjunt de passos seqüencials. En primer lloc, es realitza el carregament de dades, mitjançant la lectura de les imatges de prova des d'un fitxer XML que conté la ruta de les imatges, l'enunciat, la resposta de referència i l'etiqueta esperada (correcte o incorrecte). A continuació, es realitza la segmentació i el reconeixement OCR per extreure automàticament el text de l'enunciat i la resposta. Un cop obtinguts els texts, s'envia al model LlaMa2 a través del prompt per a poder prosseguir amb l'avaluació. El model retorna una valoració estructurada, que inclou una classificació (Correcte/Incorrecte), una puntuació i un comentari. Seguidament, es fa la comparació amb l'etiqueta esperada per comprovar si la classificació del model coincideix amb l'etiqueta estipulada al XML. Per últim, es realitza l'emmagatzematge dels resultats en un fitxer CSV, on es recullen la ruta de la imatge, l'etiqueta de referència, la resposta generada pel model i si hi ha hagut coincidència o no en la classificació.

Aquest procediment ha permès avaluar de manera sistemàtica el rendiment de l'LLM en la tasca de correcció de respostes escrites a mà. A més, gràcies al format estructurat del prompt, el sistema no només determina si una resposta és vàlida, sinó que també aporta una justificació textual, cosa que pot ser útil per a posteriors revisions o millores del sistema.

Els resultats obtinguts es fan servir per calcular el percentatge d'encerts del model (és a dir la proporció de casos en què la seva valoració coincideix amb l'etiqueta esperada), cosa que proporciona una mesura clara de la fiabilitat del sistema de correcció. Aquesta validació és clau per determinar fins a quin punt el model pot servir de suport fiable a la correcció manual en entorns reals d'avaluació.

En total, s'han avaluat els 63 casos del conjunt de dades, dels quals en 55 casos el model va coincidir amb l'etiqueta esperada, mentre que en 8 casos la seva classificació va ser diferent a l'esperada. Aquestes dades representen un percentatge global d'encert del 87,3%, una xifra que indica un comportament generalment fiable del model en la tasca de correcció. Per tal de visualitzar aquesta proporció de manera més clara, s'ha inclòs un gràfic de percentatges que mostra la distribució dels encerts i desacords entre el model i les etiquetes de referència.

Figura 19. Gràfic del percentatge d'encerts i errors del model LlaMa2.

Anàlisi dels errors:

Els 8 casos en què el model no va coincidir amb l'etiqueta esperada es poden classificar en les següents categories:

- **Falta de reconeixement OCR (1 cas):**

  En aquest cas, el model no va poder reconèixer text degut a un error en el reconeixement OCR. Aquest error es pot atribuir a una fallada en el procés de segmentació, on el sistema no va ser capaç de seleccionar de manera precisa el contorn de la imatge a extreure del text. A vegades, la segmentació no identifica correctament les àrees que contenen el text, cosa que pot provocar que la imatge es redimensioni i esdevingui, per tant, deformada afectant així la qualitat del reconeixement. Així, la dificultat no recau únicament en la qualitat de la imatge, sinó en la precisió del procés de segmentació, el qual pot influir directament en la capacitat del model OCR per identificar correctament la resposta. Aquesta

situació manifesta la dependència del model en la fiabilitat dels processos prèvis, com la segmentació, per tal de garantir una avaluació precisa.

- **Nota sobrevalorada amb comentari correcte (1 cas):**

  En un cas, el model ha puntuat una nota més alta de la que corresponia, tot i que el comentari proporcionat és adequat i correcte. Això podria ser el resultat d'un error en el càlcul de la puntuació o d'una inconsistència en el procés de valoració, que no va reflectir adequadament el grau de precisió de la resposta.

- **Mala interpretació del text reconegut (6 casos):**

  En aquests casos, tot i que el text reconegut és correcte, el model ha interpretat malament la informació. Això pot haver passat per diverses raons, com ara la falta de context adequat o la complexitat de la pregunta, que ha portat a una resposta incorrecta o confusa. Tot i així, els comentaris proporcionats pel model eren correctes en relació al contingut, però la interpretació no va ser encertada en el context de la pregunta.

8.3 Avaluació del sistema en relació als requisits funcionals i no funcionals

El sistema dissenyat compleix de manera efectiva amb els requisits funcionals i no funcionals establerts. Permet la càrrega d'imatges d'exàmens en format PNG, extreu i segmenta amb precisió els enunciats i respostes, genera un prompt estructurat per a la correcció mitjançant un LLM i ofereix retroalimentació detallada per a l'usuari. La interfície es intuïtiva i accessible, garantint una usabilitat òptima per als usuaris sense coneixements tècnics. A més, la privacitat de les dades es garanteix mitjançant el processament local, ja que el sistema no necessita serveis o APIs externes, utilitzant models locals com el d'Ollama i el d'OCR (Tesseract i TrOCR). El sistema també, degut a la seva estructura modular, es escalable i mantenible permetent l'evolució i integració futura sense dificultat.

9 Conclusions

Aquest Treball de Fi de Grau ha tingut com a objectiu principal el desenvolupament d'una aplicació orientada al suport en l'avaluació de respostes manuscrites, integrant tecnologies avançades com models d'OCR i models LLM. La implementació i validació d'aquesta solució han permès assolir amb èxit els objectius definits inicialment, tot i que el procés ha comportat diversos desafiaments tècnics que han presentat una font d'aprenentatge.

Els resultats obtinguts mostren que el sistema és capaç d'assolir una taxa de precisió superior al 80% en el reconeixement i correcció de respostes manuscrites, la qual cosa demostra la seva viabilitat dins del context proposat. Aquesta eficiència ha estat validada mitjançant eines d'avaluació com Jiwer per tal de parametritzar el funcionament del sistema o components d'aquest. A més, s'ha aconseguit una integració efectiva entre els models OCR i LLM, destacant especialment l'ús de TrOCR per al reconeixement de text manuscrit.

Malgrat aquests èxits, s'han identificat algunes limitacions en el sistema. L'aplicació presenta una càrrega computacional elevada, derivada de l'ús simultani de diversos models d'IA, fet que condiciona la seva escalabilitat en entorns amb recursos limitats. Així mateix, la interfície gràfica desenvolupada, tot i ser funcional, podria millorar-se per optimitzar l'experiència d'usuari.

Seguidament, una de les proves realitzades mostra que el sistema ha trigat aproximadament 46 minuts per processar i corregir 63 imatges, cadascuna composta per un enunciat imprès i una resposta manuscrita. Aquest temps elevat és principalment degut a les limitacions de maquinari on s'ha executat l'LLM. En canvi, en entorns amb més capacitat computacional, com un servidor dedicat o al núvol, aquest problema es reduiria considerablement. Aquesta dada evidencia una limitació temporal important, especialment en entorns educatius amb altes càrregues de treball o necessitats de resposta immediates. Per això, la integració de sistemes més eficients o l'ús d'estratègies d'optimització serien línies de millora prioritàries per optimitzar el temps de resposta.

Cal recalcar que aquesta eina s'ha concebut com un suport per a la docència, no com una substitució dels processos educatius tradicionals ni dels criteris pedagògics establerts. L'objectiu és alleugerir la càrrega dels docents en tasques repetitives i proporcionar una base objectiva sobre la qual fer una revisió final, però sempre amb supervisió humana per garantir l'equitat i la qualitat educativa.

A més, un possible enfocament a futur seria avaluar la viabilitat d'eliminar la dependència dels models OCR dedicats, aprofitant les capacitats emergents dels LLM multimodals que ja poden interpretar text dins d'imatges amb una precisió notable. Aquest enfocament podria simplificar l'arquitectura i reduir els temps de processament, encara que caldria evaluar-ne amb detall la precisió en contextos educatius.

Una línia d'evolució rellevant seria la incorporació de tècniques Retrieval-Augmented Generation (RAG, generació augmentada amb recuperació d'informació), una arquitectura que combina l'accés a bases de dades externes amb la generació de respostes per part del model. En aquest context, la incorporació de RAG podria permetre al sistema consultar recursos docents, bancs de preguntes, contingut específic i rúbriques d'avaluació per millorar la qualitat i coherència de les correccions. Aquesta millora permetria adaptar l'eina a diferents matèries i nivells educatius de manera més precisa i escalable [13].

Com a possibles línies de futur, es proposa una evolució cap a una arquitectura distribuïda, on la càrrega computacional es pugui delegar a serveis externs mitjançant API, permetent una experiència més fluïda i amb una menor dependència dels recursos locals. Això inclouria també la incorporació de models més potents i refinats, entrenats amb volums majors de dades.

Des d'un punt de vista formatiu, aquest TFG ha estat una experiència productiva. M'ha permès profunditzar en camps tecnològics que desconeixia abans d'iniciar el projecte, com ara són els sistemes OCR i els LLM, així com en metodologies d'avaluació i experimentació amb dades. He après a gestionar un projecte tecnològic complex de manera autònoma, i a adaptar-me a les limitacions de recursos, prioritzant solucions open-source. A nivell pràctic, he adquirit competències en integració de sistemes, tractament i validació de dades textuals, i ús d'eines per a la mesura de rendiment i qualitat dels resultats.

En conclusió, aquest projecte ha estat una experiència clau per aplicar de manera pràctica tot el que he après durant el grau. M'ha permès afrontar reptes reals, prendre decisions tècniques amb criteri i entendre millor com integrar diferents tecnologies per resoldre un problema concret. Ha estat una bona oportunitat per guanyar habilitats útils tant a nivell tècnic com de gestió de projecte, que segur m'ajudaran en el futur professional.

10 Aplicació dels principis ètics i responsabilitat social

Aquest apartat està orientat a la reflexió sobre els aspectes ètics, socials i ambientals relacionats amb el projecte desenvolupat, consistint aquest en una eina de suport a l'avaluació d'exàmens manuscrits. Es tindran en compte la igualtat, el medi ambient, la responsabilitat social i l'ètica.

10.1 Igualtat

En el cas d'aquest projecte, no es recull ni es té cap dada relacionada amb el gènere de la persona avaluada, ja que l'aplicació es basa exclusivament en l'anàlisi de la resposta escrita a mà. Per tant, el sistema no té cap accés a informació personal que pugui condicionar l'avaluació de les respostes en funció del gènere. Aquesta absència d'informació sensible actua com a mesura protectora davant possibles discriminacions, assegurant que el procés d'avaluació es centra únicament en el contingut i la qualitat de la resposta escrita i reconeguda a partir de l'OCR. Això contribueix a una avaluació imparcial, on totes les persones són tractades de manera igualitària.

10.2 Medi ambient

El projecte utilitza com a entrada imatges d'exàmens manuscrits, generalment fets en paper. Tot i que l'aplicació no manté cap registre ni permet la reutilització del contingut, pot contribuir a reduir l'ús del paper en algunes fases del procés educatiu, especialment si s'utilitza conjuntament amb suports digitals. Aquest factor pot afavorir una transició progressiva cap a entorns més digitals i menys dependents del paper, especialment en contextos on el plantejament de l'enunciat i la resposta pugui digitalitzar-se.

10.3 Responsabilitat social

Aquest projecte ofereix una eina de suport per al procés d'avaluació, amb la finalitat de proporcionar orientacions i suggeriments en la correcció de respostes manuscrites. No es tracta d'un sistema automatitzat ni substitutiu, sinó d'un recurs complementari que ajuda al professorat a guiar el seu criteri i millorar la coherència en la valoració de respostes. Aquesta assistència pot resultar especialment útil en contextos amb una elevada càrrega docent o amb recursos educatius limitats, afavorint així una distribució més eficient del temps i una millora general en la qualitat del procés d'avaluació. Amb aquest enfocament, el projecte contribueix a fer l'educació més equitativa i accessible, mantenint sempre la importància del docent en la presa de decisions.

10.4 Ètica

El projecte s'ha desenvolupat tenint en compte els principis ètics relacionats amb la privacitat, la responsabilitat i l'ús just de la tecnologia. L'aplicació no recull dades personals ni emmagatzema informació un cop finalitzada l'execució, fet que garanteix la confidencialitat de les respostes i protegeix els drets dels usuaris.

A més, l'aplicació es planteja com un suport i no com un sistema de decisió automàtica. Aquesta aproximació reforça el compromís amb un ús responsable de la tecnologia, mantenint sempre el criteri professional en el procés d'avaluació.

11 Recursos Utilitzats

Per a realitzar aquest projecte s'han utilitzat diversos recursos tant de programari com de maquinari, així com biblioteques, entorns de desenvolupament i eines d'avaluació. Aquests recursos han estat seleccionats en funció dels requisits tècnics del sistema, la seva compatibilitat amb tecnologies d'intel·ligència artificial i reconeixement òptic de caràcters, i la seva disponibilitat com a solucions open-source o d'ús lliure per a entorns acadèmics.

11.1 Llicències de programari usat

Aquest projecte ha fet ús de diverses eines i recursos de codi obert. A continuació, es detallen les llicències associades a cadascun d'ells per garantir el compliment de les condicions d'ús establertes.

Llibreria Programari	**Llicència**	**Descripció**
Tesseract OCR	Apache 2.0	Permet utilitzar, modificar i distribuir el programari de manera gratuïta, amb la condició que s'inclogui l'avís de drets d'autor. No permet l'ús de marques registrades sense permís [5].
Tr-OCR	MIT License	Permet utilitzar, copiar, modificar, fusionar, publicar i distribuir el programari. Es proporciona "tal qual", sense garanties [10] [11] [12].
EasyOCR	Apache 2.0	Permet utilitzar, modificar i distribuir el programari, amb la condició que s'inclogui l'avís de drets d'autor de la llicència [8].
PaddleOCR	Apache 2.0	Permet l'ús gratuït, la modificació i la distribució del programari, amb la condició de mantenir els avisos de drets d'autor i llicències [7].
Doctr	MIT License	Permet l'ús, modificació i distribució gratuïta del programari, amb la condició que s'inclogui l'avís de drets d'autor [9].
Ollama	MIT License	Permet l'ús, modificació i distribució gratuïta del programari, amb la condició que s'inclogui l'avís de drets d'autor [6].
LlaMa 2	Community License Agreement	Permet l'ús no comercial, sense modificacions. No permet la comercialització ni la millora del model per altres LLM [6].

Taula 3. Taula de llicències usades per a aquest projecte.

Tot el software utilitzat en aquest projecte és open-source i d'accés gratuït. A continuació es detallen les eines i llibreries emprades:

- **Python 3.11:** Llenguatge de programació utilitzat per al desenvolupament del sistema.
- **Tesseract OCR:** Motor OCR open-source utilitzat per a la conversió d'imatges de text imprès o digital a text extret.
- **TrOCR (base-handwritten):** Model de preentrenament utilitzat per al reconeixement de text manuscrit basat en Transformer.
- **EasyOCR:** Eina OCR utilitzada per a fer proves de selecció d'OCRs.
- **PaddleOCR:** Framework OCR utilitzat per fer proves de selecció d'OCRs.
- **Python-doctr:** Llibreria utilitzada per al processament de documents i imatges amb xarxes neuronals, optimitzada per a OCR. S'ha usat en aquest treball per tal de fer proves de selecció d'OCRs.
- **PyCharm:** IDE utilitzat per al desenvolupament del codi amb suport per a Python.
- **Ollama (LlaMa2):** Servei d'integració de models LlaMa2 per al processament de llenguatge natural.
- **Git i GitHub:** Eines per al control de versions del codi font i la gestió del projecte. Git permet gestionar l'historial de canvis.

11.2 Datasets utilitzats

- **IAM Handwriting Database:** Base de dades utilitzada per a la prova del sistema OCR en textos manuscrits (inclou paraules, línies i frases) [15].
- **ICDAR 2003:** Conjunt de dades utilitzat per a l'avaluació de reconeixement de caràcters i paraules de texts [16].
- **Dataset propi de proves:** Conjunt d'imatges creat manualment per a la validació de l'aplicació desenvolupada. Aquest dataset inclou fotografies de preguntes i respostes manuscrites, acompanyades de fitxers XML que contenen la transcripció estructurada de cada mostra. S'ha utilitzat per provar el funcionament global de l'eina i avaluar la precisió del reconeixement i processament de text en escenaris reals.

11.3 Paquets i llibreries utilitzades

- **OCR i processament d'imatges:**
  - opencv-python, pytesseract, easyocr, paddleocr, doctr[torch], Pillow, numpy, matplotlib.
- **Models prèviament entrenats i NLP:**
  - Transformers, torch, jiwer, ollama
- **GUI:**
  - Tkinter
- **Utilitats del sistema i gestió de dades:**
  - Tqdm, pandas

- **Processament XML:**
  - Lxml

Cites

[1] H. Zhang, D. Liu, and Z. Xiong, "CNN-Based Text Image Super-Resolution Tailored for OCR," Proceedings of the Visual Communications and Image Processing (VCIP), St. Petersburg, USA, Dec. 2017.

[2] A. Naseer and K. Zafar, "Meta features-based scale invariant OCR decision making using LSTM-RNN," *Computational and Mathematical Organization Theory*, vol. 24, no. 3, pp. 1-20, 2018.

[3] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, i S. Park, "OCR-free Document Understanding Transformer," arXiv preprint arXiv:2111.15664, Nov. 2021.

[4] J. Memon et al., "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," IEEE Access, vol. 8, pp. 142642-142660, 2020.

[5] Tesseract OCR, "Tesseract: Open Source OCR Engine", GitHub. [Enlace]. Disponible a: https://github.com/tesseract-ocr/tesseract.

[6] Ollama, "Ollama: Llama 2", GitHub. [Enllaç]. Disponible a: https://github.com/ollama/ollama/tree/main.

[7] PaddlePaddle, "PaddleOCR", GitHub. [Enllaç]. Disponible a: https://github.com/PaddlePaddle/PaddleOCR.

[8] JaidedAI, "EasyOCR", GitHub. [Enllaç]. Disponible a: https://github.com/JaidedAI/EasyOCR.

[9] Mindee, "Doctr", GitHub. [Enllaç]. Disponible a: https://github.com/mindee/doctr.

[10] Hugging Face, "TrOCR: Transformer-based Optical Character Recognition", Hugging Face. [Enllaç]. Disponible a: https://huggingface.co/docs/transformers/model_doc/trocr.

[11] Microsoft, "TrOCR: Transformer-based Optical Character Recognition", GitHub. [Enllaç]. Disponible a: https://github.com/microsoft/unilm/tree/master/trocr.

[12] Microsoft, "TrOCR base-handwritten - Transformer-based OCR model for handwritten text", Hugging Face. [Enllaç]. Disponible a: https://huggingface.co/microsoft/trocr-base-handwritten

[13] J. Wang, W. Ding y X. Zhu, "Financial Analysis: Intelligent Financial Data Analysis System Based on LLM-RAG," Proceedings of [Conferencia o Revista], 2024.

[14] W. Ansar, S. Goswami, and A. Chakrabarti, "A Survey on Transformers in NLP With Focus on Efficiency," arXiv preprint arXiv:2406.16893, 2024.

[15] U. Marti and H. Bunke, "The IAM-database: An English sentence database for off-line handwriting recognition," International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp. 39-46, 2002.

[16] S. M. Lucas et al., "ICDAR 2003 Robust Reading Competitions: Entries, Results and Future Directions," International Journal on Document Analysis and Recognition, vol. 7, no. 2-3, pp. 105–122, 2005.