
This is the **published version** of the bachelor thesis:

Purti Ramirez, Gerard. *Assistència conversacional privada per a empreses : arquitectura, desplegament local i integració amb ERP mitjançant RAG*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334408>

under the terms of the  license.

Assistència conversacional privada per a empreses: arquitectura, desplegament local i integració amb ERP mitjançant RAG

Gerard Purí Ramírez

Resum — Les empreses concentren gran part de la seva informació en sistemes ERP, sovint difícils de consultar per a usuaris no tècnics. Aquest treball dissenya, implementa i avalua un assistent conversacional privat que permet consultar aquesta informació en llenguatge natural, executable en local per garantir-ne la confidencialitat. La solució combina una arquitectura de Retrieval-Augmented Generation (RAG) —que fonamenta les respostes en les dades reals de l'empresa per evitar les al·lucinacions— amb un control d'accés basat en rols (RBAC) que limita la informació segons el perfil de l'usuari. Es comparen quatre models de llenguatge de codi obert (LLaMA 2, Mistral, Phi-2 i DeepSeek), amb i sense RAG, per identificar la millor opció en una instal·lació local amb recursos limitats. Els resultats mostren que, sense RAG, cap model respon correctament les consultes empresarials, mentre que amb RAG la qualitat millora substancialment i LLaMA 2 assoleix el millor equilibri entre fiabilitat i velocitat. El sistema es valida sobre dades representatives d'un ERP i se n'analitza la viabilitat com a servei SaaS.

Paraules clau—Assistents conversacionals, intel·ligència artificial, ERP, RAG, sistemes d'informació empresarials, GPT, RBAC, SaaS

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

L'aparició dels models de llenguatge de gran escala, LLMs (Large Language Models), ha canviat la forma en què les persones interactuen avui amb els sistemes d'informació. Aquests models, basats en l'arquitectura Transformer proposada per Vaswani et al. [1], poden entendre i generar text en llenguatge natural amb un grau de fluïdesa que fa uns anys semblava impossible. Els primers models de la família GPT (Generative Pre-trained Transformer) [2] van obrir el camí, i avui trobem versions molt més potents com GPT-4 [3] juntament amb alternatives de codi obert com LLaMA [4], Mistral [5] o DeepSeek [14]. El ventall de possibilitats ha augmentat notablement en pocs anys.

En general, el rendiment d'aquests models va estretament lligat a la seva mida, és a dir, al seu nombre de paràmetres [11]. Com més gran és un model, millors solen ser les seves respostes, però també requereix més recursos computacionals per executar-se. Aquesta característica condiona el seu ús en entorns locals: quan una empresa vol desplegar un model en la seva pròpia infraestructura per garantir la seguretat de les dades i evitar compartir informació sensible amb serveis externs, sovint es veu obligada a utilitzar models més petits, tot i que aquests presenten més limitacions. En aquest context, models com Mistral 7B [5] o Phi-2 [6] resulten especialment atractius, ja que estan dissenyats per funcionar amb recursos limitats sense renunciar del tot a la qualitat. En canvi, el model LLaMA 2 70B [4] ofereix un rendiment superior però necessita infraestructura molt més potent, cosa que el fa menys viable per a una PIME.

Tot i les capacitats d'aquests models, el seu ús directe presenta una limitació important: les respostes es generen únicament a partir del coneixement adquirit durant l'entrenament, sense accés a informació actualitzada o

específica d'un domini concret. Per superar aquesta limitació, Lewis et al. [7] van proposar l'arquitectura de generació augmentada amb recuperació d'informació (RAG, Retrieval-Augmented Generation), que combina un model generatiu amb un sistema de cerca que recupera documents rellevants per contextualitzar la resposta. Un dels elements clau d'un sistema RAG és la indexació dels documents, que permet estimar correctament el contingut de cadascun i recuperar els més rellevants per a cada consulta. Gao et al. [8] presenten una revisió completa de les variants d'aquesta arquitectura, que inclouen des del RAG bàsic fins a configuracions avançades amb reranking, recuperació iterativa o fragmentació adaptativa dels documents.

En l'àmbit empresarial, una part significativa de la informació operativa es concentra en sistemes ERP (Enterprise Resource Planning), un programari de gestió empresarial que integra i centralitza dades de diferents àrees funcionals com comptabilitat, vendes, inventari, compres o recursos humans en una única base de dades. L'accés a aquesta informació sovint requereix coneixements tècnics o la formulació de consultes complexes que no tots els treballadors dominen. Un assistent conversacional que permeti accedir a aquestes dades mitjançant llenguatge natural pot facilitar l'accés a la informació per a usuaris no tècnics i simplificar les operacions diàries de l'empresa.

En aquest projecte es desenvolupa un sistema funcional que es pot desplegar en una infraestructura local, garantint la confidencialitat de les dades i fent possible la seva integració amb altres sistemes interns de l'empresa, com l'ERP. Durant el projecte es comparen diferents models de llenguatge de codi obert (LLaMA, Mistral, Phi i DeepSeek) amb l'objectiu de trobar la configuració que ofereixi els millors resultats en termes de qualitat de resposta, velocitat

d'inferència i consum de recursos quan es combina amb un sistema RAG. El sistema incorpora també un control d'accés basat en rols (RBAC) que garanteix que cada perfil d'usuari només pugui consultar la informació que li correspon segons les seves responsabilitats dins l'empresa.

1.1 Objectius

L'objectiu general d'aquest treball és dissenyar, implementar i avaluar una arquitectura d'assistència conversacional privada per a entorns empresarials, comparant diferents models de llenguatge de codi obert combinats amb un sistema RAG i integrats amb les dades d'un ERP.

OE1. Dissenyar i implementar un sistema RAG que indexi i recuperi informació de l'ERP i la incorpori com a context per a la generació de respostes, comparant diferents estratègies de fragmentació i recuperació de documents.

OE2. Analitzar i comparar el rendiment de diferents models de llenguatge de codi obert (LLaMA, Mistral, Phi i DeepSeek) en tasques conversacionals dins d'un context empresarial, avaluant la qualitat de les respostes, el temps d'inferència i els requisits de memòria de cada model.

OE3. Dissenyar i implementar un control d'accés basat en rols (RBAC) que garanteixi que cada perfil d'usuari només pugui consultar la informació autoritzada, aplicant la restricció a la fase de recuperació del sistema.

OE4. Validar el funcionament del sistema complet —interfície, RAG i RBAC— sobre un conjunt de dades representatiu de l'entorn empresarial i deixar la solució preparada per al desplegament en una infraestructura real. Per motius de confidencialitat de les dades de l'empresa, la validació es realitza sobre dades anonimitzades que reproduïen les categories d'informació d'un ERP (facturació, comandes, clients i catàleg de serveis).

2 ESTAT DE L'ART I TREBALLS RELACIONATS

L'aplicació de models de llenguatge a la consulta d'informació empresarial s'emmarca en dues línies de recerca consolidades: els models de llenguatge de gran escala de codi obert i les arquitectures de recuperació augmentada (RAG).

Pel que fa als models, l'alliberament de famílies com LLaMA [4], Mistral [5] o Phi [6] ha fet possible desplegar capacitats de generació de text en infraestructura pròpia, sense dependre de serveis externs. Aquesta disponibilitat és la que fa viable un assistent privat com el que es planteja a l'OE2, on es comparen precisament aquests models en un entorn local.

Pel que fa a la recuperació, Lewis et al. [7] van introduir el paradigma RAG, i Gao et al. [8] en recullen les variants actuals. Dins d'aquesta línia, treballs com Dense Passage Retrieval [12] han demostrat l'eficàcia de representar consultes i documents en un espai vectorial comú per a la cerca

semàntica, principi que fonamenta directament el mòdul de recuperació de l'OE1 (Sentence-BERT i FAISS). Així mateix, l'estudi de Liu et al. [13] sobre com els models utilitzen el context mostra que la quantitat i la posició de la informació dins del prompt n'afecten la qualitat, un factor que condiciona el disseny de la fase de recuperació avaluada en aquest treball.

A diferència dels sistemes RAG genèrics, l'aportació d'aquest treball és la combinació d'aquest paradigma amb un control d'accés basat en rols aplicat a la capa de recuperació i orientat a un domini empresarial concret (dades d'ERP), un aspecte poc tractat a la literatura general de RAG i que constitueix el nucli de l'OE3.

3 PLANIFICACIÓ

Per organitzar el desenvolupament del treball s'han agrupat les tasques en quatre paquets de treball (work packages) que cobreixen tot el cicle del projecte. La distribució temporal es recull al diagrama de Gantt (Figura 1).

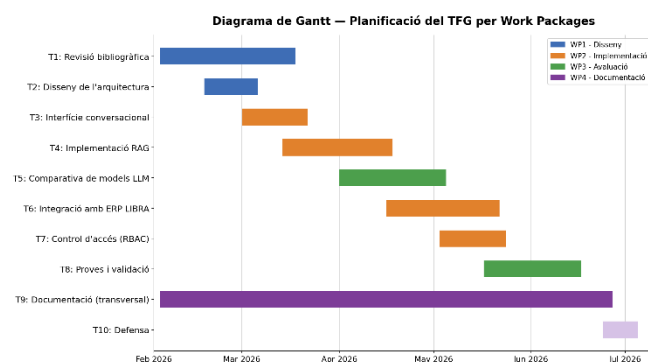


Figura 1. Diagrama de Gantt amb la planificació temporal del projecte.

WP1 — Disseny. Inclou la revisió bibliogràfica i l'estudi de l'estat de l'art (T1), així com el disseny de l'arquitectura del sistema (T2). Aquest paquet ha servit per fonamentar les decisions tècniques posteriors i definir els components principals del sistema.

WP2 — Implementació. Agrupa el desenvolupament de la interfície conversacional (T3), la implementació del sistema RAG amb chunking, embeddings i base de dades vectorial (T4), la integració amb les dades de l'ERP (T6) i el sistema de control d'accés basat en rols (T7).

WP3 — Avaluació. Inclou la comparativa sistemàtica dels models de llenguatge candidats (T5) i les proves de validació del sistema complet (T8). En aquest paquet es recullen les mètriques de qualitat (rellevància, correcció) i de rendiment (latència, ús de memòria).

WP4 — Documentació. A diferència dels altres paquets, aquest és transversal a tot el projecte. Inclou la redacció dels informes de seguiment al llarg del semestre, l'elaboració de l'informe final (T9) i la preparació de la defensa pública (T10).

4 METODOLOGIA

El projecte avança amb una metodologia iterativa i incremental basada en el marc de treball Scrum, adaptada a les particularitats d'un treball individual. S'han definit sprints de dues setmanes. Cada sprint acaba amb una revisió del progrés assolit, que porta a una planificació de les tasques del següent sprint. Les reunions de seguiment amb el tutor del projecte actuen com a sprint review, ajuden a resoldre dubtes tècnics i són útils per detectar millores.

La metodologia es tria per diferents raons. En primer lloc, pel caràcter d'exploració del projecte, que compara diversos models i configuracions; això fa recomanable incorporar els aprenentatges de cada iteració en les decisions posteriors. En segon lloc, perquè el sistema té parts independents — interfície, RAG, integració amb l'ERP i control d'accés —, cosa que fa útil construir-lo peça a peça.

5 MÈTODES

5.1 Arquitectura general

El sistema s'estructura en quatre components principals que es comuniquen entre si (Figura 2). Aquesta arquitectura modular permet substituir el model de llenguatge sense haver de modificar la resta del sistema, cosa que facilita la comparativa entre models prevista a OE2.

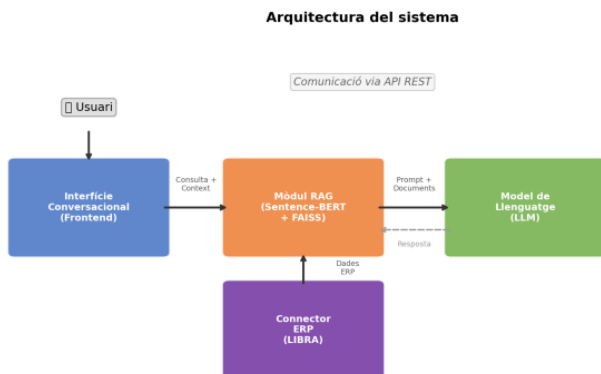


Figura 2. Arquitectura del sistema amb els quatre components principals.

El primer component és la interfície conversacional, que gestiona la interacció amb l'usuari i manté l'historial de la conversa per preservar el context entre missatges consecutius. El segon component és el mòdul de recuperació d'informació basat en RAG, que indexa els documents, genera embeddings amb Sentence-BERT [9] i els emmagatzema en una base de dades vectorial FAISS [10] per fer cerques per similitud. El tercer component és el model de llenguatge, que rep la consulta de l'usuari juntament amb els fragments rellevants recuperats i genera la resposta final. El quart component és el connector de dades, que transforma la informació de l'ERP en documents indexables pel mòdul RAG.

5.2 Recursos computacionals

Les proves de la comparativa s'han executat al clúster de càlcul DCC de la UAB, gestionat amb el sistema de cues SLURM. Cada model s'ha avaluat en un treball independent amb un límit de 8 GB de memòria. Aquesta restricció és determinant: els models de 7B paràmetres en precisió float16 ocupen aproximadament 14 GB i no es podien carregar dins del límit. Per superar-ho, Mistral i DeepSeek es van executar amb versions quantitzades en format GGUF (4 bits, ~4 GB) amb la llibreria llama-cpp-python, mentre que LLaMA 2 i Phi-2 es van poder executar en float16 amb Transformers. A més, una incompatibilitat entre la versió de PyTorch disponible per a Python 3.12 i l'arquitectura de la GPU del node (compute capability 6.1, no suportada per les versions recents de PyTorch) va obligar a executar la comparativa en CPU. Aquesta limitació afecta els temps d'inferència absoluts però no invalida la comparació relativa entre models, ja que tots s'han executat en les mateixes condicions de maquinari. Finalment, la comparativa es va reproduir en les mateixes condicions (CPU, float16) per a LLaMA 2 i Phi-2, obtenint temps consistents amb la primera execució, fet que confirma la reproductibilitat dels resultats.

5.3 Interfície conversacional

S'ha desenvolupat una interfície conversacional funcional (Figura 3) que permet als usuaris formular consultes en llenguatge natural i visualitzar les respostes de l'assistent en format de xat. La interfície incorpora una pantalla d'inici de sessió on l'usuari selecciona el seu rol (gerència, comercial o fiscal), que determina a quins documents podrà accedir. Manté l'historial de conversa i mostra, per a cada resposta, els documents concrets dels quals s'ha extret la informació, fet que permet verificar el comportament del control d'accés.

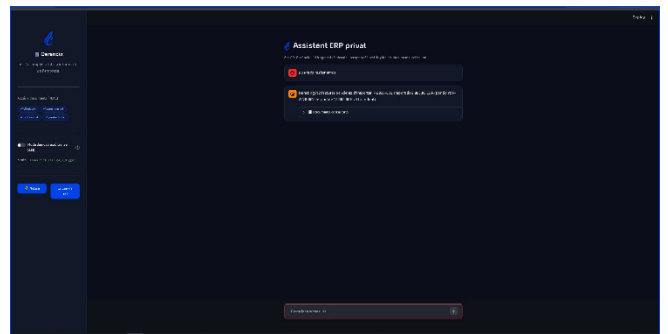


Figura 3. Interfície conversacional amb selecció de rol i resposta generada pel sistema (LLaMA 2) a partir de les dades recuperades.

5.4 Mòdul de recuperació (RAG)

El mòdul de recuperació d'informació és el component que permet al sistema accedir a dades internes de l'empresa per contextualitzar les respostes del model de llenguatge. El pipeline RAG (Figura 4) segueix tres etapes:

fragmentació dels documents (chunking), generació de representacions vectorials (embeddings) i cerca per similitud.

En primer lloc, els documents textuais extrets de l’ERP es divideixen en fragments de mida controlada per facilitar-ne la indexació i la recuperació precisa. A continuació, cada fragment es transforma en un vector numèric mitjançant el model Sentence-BERT [9], que genera embeddings que capturen el significat semàntic del text. Aquests vectors s’emmagatzemen en una base de dades vectorial FAISS [10], que permet fer cerques eficients per similitud.

Quan l’usuari formula una consulta, el sistema genera l’embedding de la consulta amb el mateix model Sentence-BERT, cerca els fragments més similars a la base de dades FAISS i els incorpora com a context dins del prompt que s’envia al model de llenguatge. D’aquesta manera, el model genera la resposta a partir de les dades reals de l’empresa i no únicament del seu coneixement preentrenat.

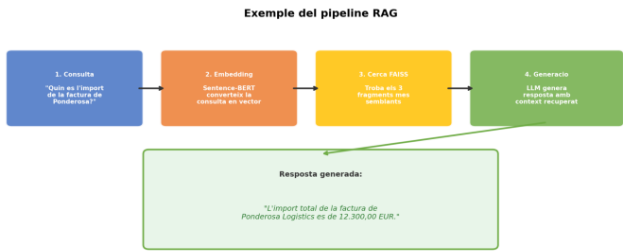


Figura 4. Exemple del pipeline RAG: des de la consulta de l’usuari fins a la resposta generada amb context recuperat.

Pel que fa a la configuració concreta del pipeline, els documents s’han fragmentat per paràgrafs (separació per doble salt de línia), estratègia que preserva la coherència semàntica de cada fragment sense imposar una mida fixa artificial; el conjunt de dades de quatre fitxers ha generat un total de 14 fragments indexats. Per a la generació d’embeddings s’ha emprat el model paraphrase-multilingual-MiniLM-L12-v2 de Sentence-BERT, que produeix vectors de 384 dimensions i suporta nativament el català i el castellà, cosa especialment rellevant per a consultes empresarials en entorns multilingües. Els vectors s’indexen amb FAISS IndexFlatL2 (cerca exacta per distància euclídea), i per a cada consulta es recuperen els 3 fragments més similars, que s’incorporen com a context dins del prompt. La generació s’ha configurat en mode determinista (do_sample=False) amb temperatura 1.0 i un límit de 256 tokens per resposta, garantint la reproductibilitat dels resultats.

5.5 Models de llenguatge

Per a la comparativa prevista a OE2 s’han identificat quatre models candidats: LLaMA 2 7B, Mistral 7B, Phi-2 i DeepSeek. Tots quatre són models de codi obert que poden executar-se en infraestructura local. La comparativa avalua cada model en dues condicions, amb RAG i sense RAG, per quantificar l’aportació del sistema de recuperació. Les mètriques d’avaluació inclouen la qualitat de les respostes

(coherència, rellevància i correcció factual), el temps d’inferència mitjà per consulta i el consum de memòria. Les proves s’han dut a terme amb el mateix maquinari i el mateix conjunt de consultes per garantir que els resultats siguin comparables.

A causa de les restriccions de memòria del clúster (8 GB per job), els models Mistral7B i DeepSeek 7B no han pogut ser carregats en precisió float16 (~14 GB). Per superar aquesta limitació, s’han emprat versions quantitzades en format GGUF amb precisió de 4 bits, que redueixen el consum de memòria a aproximadament 4 GB per model. Aquesta tècnica de quantització comprimeix els pesos del model amb una pèrdua de qualitat mínima i és habitual en desplegaments amb recursos limitats. LLaMA 2 i Phi-2 s’han executat en float16 amb la llibreria Transformers, mentre que Mistral i DeepSeek s’han executat amb llama-cpp-python.

5.6 Conjunt de dades i control d’accés

Per motius de confidencialitat, el sistema s’ha validat sobre un conjunt de dades representatiu i anonimitzat que reproduïx les categories d’informació d’un ERP real: factures, comandes, fitxes de clients i catàleg de serveis. Sobre aquestes dades s’aplica el control d’accés basat en rols, amb tres perfils diferenciats — gerència, comercial i fiscal —, cadascun amb un subconjunt específic d’informació accessible segons les seves necessitats operatives. El disseny modular del connector permet substituir aquest conjunt per l’extracció directa d’un ERP en producció (per exemple LIBRA) sense modificar la resta del sistema.

Cal remarcar que, de manera deliberada, el sistema no s’ha connectat a les dades reals de producció de l’empresa col·laboradora. La confidencialitat de la informació interna — que és precisament la motivació central d’aquest projecte — impedeix incloure o exposar dades reals en el marc d’un treball acadèmic públic. Per aquest motiu, el desenvolupament i la validació s’han dut a terme sobre un conjunt de dades anonimitzat que reproduïx fidelment l’estructura i les categories d’un ERP real (factures, comandes, clients i catàleg de serveis). El connector de dades i la interfície s’han dissenyat de manera que la substitució d’aquest conjunt per la connexió directa a l’ERP LIBRA en un entorn de producció no requereixi cap modificació de la resta del sistema: només cal redirigir la font de dades cap als mòduls corresponents de l’ERP. D’aquesta manera, la solució queda plenament preparada per al desplegament real, mantenint alhora la confidencialitat exigida.

Rol	Documents accessibles
Gerència	factures, comandes, clients, productes (accés complet)
Comercial	comandes, clients,

	productes (sense factures)
Fiscal	factures, clients (sense comandes ni productes)

Taula 1. Assignació de documents accessibles per rol (RBAC).

La restricció s'implementa a la fase de recuperació i no com a filtre posterior: per a cada rol es construeix un índex que conté únicament els fragments dels documents permesos, de manera que una consulta feta des d'un rol determinat mai pot recuperar informació de documents als quals aquell perfil no té accés. Aquest disseny garanteix que la segregació de la informació sigui efectiva a nivell de recuperació, abans que les dades arribin al model de llenguatge.

5.7 Desplegament i interfície d'usuari

La solució s'ha empaquetat com una aplicació web lleugera desenvolupada amb Streamlit, executable de manera local sense dependència de serveis externs, en coherència amb l'objectiu de confidencialitat. Per al desplegament en màquines amb recursos limitats, el model de llenguatge s'integra en format quantitzat GGUF mitjançant la llibreria llama-cpp-python, cosa que permet executar LLaMA 2 sobre CPU sense necessitat de GPU. L'aplicació gestiona la sessió de l'usuari, el rol actiu i l'historial de conversa, i mostra per a cada resposta els documents consultats, fet que fa transparent el funcionament del control d'accés.

5.8 Avaluació del sistema

L'avaluació del sistema s'ha dissenyat per mesurar dues dimensions independents: la contribució del sistema RAG a la qualitat de les respostes, i el rendiment computacional de cada model en condicions equivalents.

Per avaluar la qualitat de les respostes s'ha definit un conjunt de 5 consultes representatives que cobreixen les categories d'informació principals d'un ERP: facturació, gestió de comandes, dades de clients i catàleg de serveis (vegeu Annex A). Cada consulta s'ha executat en dues condicions: sense RAG, on el model genera la resposta únicament a partir del seu coneixement preentrenat, i amb RAG, on el model rep com a context els 3 fragments més rellevants recuperats de l'índex. Aquesta comparació parellada permet aïllar i quantificar l'aportació del sistema de recuperació.

La correcció de cada resposta s'ha verificat manualment comparant-la amb les dades contingudes als documents indexats. Una resposta es classifica com a correcta si proporciona la informació sol·licitada sense errors factuais ni invencions, i com a incorrecta en cas contrari. Aquesta metodologia d'avaluació manual és habitual en estudis preliminars amb conjunts reduïts de consultes [8], i permet detectar tant errors d'omissió com al·lucinacions, que són

especialment rellevants en un context empresarial on la fiabilitat de la informació és crítica. Per avaluar el rendiment computacional s'ha mesurat el temps d'inferència mitjà per consulta de cada model en les dues condicions, executant totes les proves en el mateix maquinari (CPU, clúster DCC de la UAB) i amb el mateix conjunt de consultes, per garantir la comparabilitat dels resultats. Per als models que no es podien carregar en precisió float16 per restriccions de memòria (Mistral i DeepSeek, ~14 GB), s'han emprat versions quantitzades GGUF 4-bit, fet que introdueix una asimetria en les condicions d'execució que cal tenir en compte en la interpretació dels temps comparats. Les mètriques automàtiques (BLEU, ROUGE) no s'han aplicat perquè requereixen respostes de referència gold-standard que no estaven disponibles.

6 RESULTATS

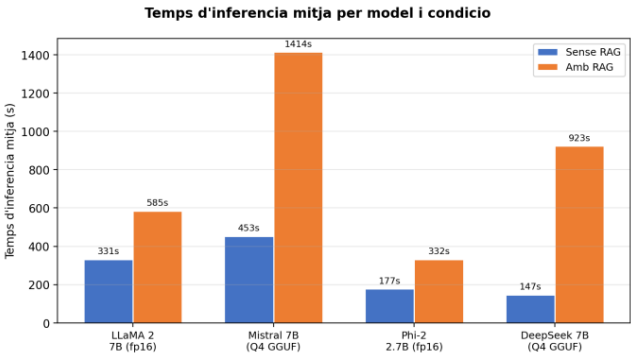
En aquest apartat es presenten els resultats obtinguts de la comparativa entre els quatre models avaluats: LLaMA 2 7B i Phi-2 2.7B (executats en float16) i Mistral 7B i DeepSeek 7B (executats en format quantitzat GGUF 4-bit). Tots quatre models s'han provat amb i sense el sistema RAG sobre el mateix conjunt de 5 consultes empresarials.

Model	Mida	Format	Temps sense RAG (s)	Temps amb RAG (s)	Correctes sense RAG	Correctes amb RAG
LLaMA 2 7B	7B	float16	331	585	0/5	4/5
Mistral 7B	7B	GGUF 4-bit	453	1414	0/5	3/5
Phi-2	2,7B	float16	177	332	0/5	2/5
DeepSeek 7B	7B	GGUF 4-bit	147	923	0/5	2/5

Taula 2. Resum de la comparativa: temps d'inferència mitjà per consulta i respostes correctes (sobre 5) per model i condició.

6.1 Temps d'inferència

La Gràfica 1 recull els temps d'inferència mitjans per consulta de cada model, amb i sense RAG.



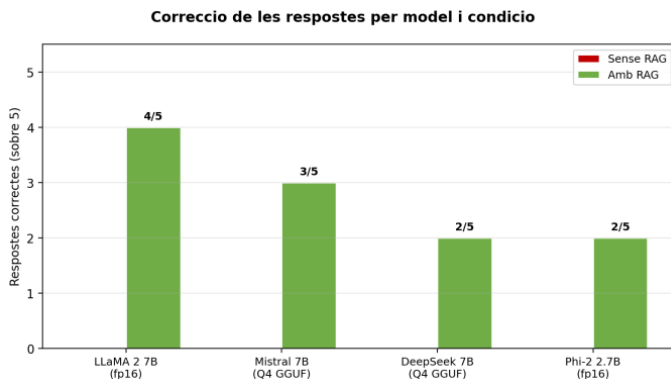
Gràfica 1. Temps d'inferència mitjà per consulta (en segons, execució en CPU).

Phi-2 és el model més ràpid en les dues condicions, cosa coherent amb la seva mida inferior (2,7B paràmetres). Entre els models de 7B, els temps varien significativament segons el format d'execució: LLaMA 2 en float16 resulta més ràpid que Mistral i DeepSeek en GGUF 4-bit, cosa que suggereix que el format quantitzat via llama-cpp-python no està optimitzat per a CPU de la mateixa manera que Transformers amb float16. Cal tenir present que els temps absoluts corresponen a execució en CPU i no representen el rendiment esperable amb GPU.

L'increment de temps amb RAG oscil·la entre un 77% (LLaMA 2) i un 528% (DeepSeek), depenent de la longitud de la resposta generada i del context addicional al prompt.

6.2 Correcció de les respostes

La Gràfica 2 mostra el nombre de respostes correctes per model sobre les 5 consultes.



Gràfica 2. Respostes correctes per model i condició (sobre 5 consultes).

Sense RAG, cap dels quatre models és capaç de respondre correctament cap consulta sobre dades internes de l'empresa. Alguns models (LLaMA 2, Mistral) generen respostes aparentment detallades però completament inventades, cosa que els fa especialment perillosos en un entorn empresarial sense mecanismes de verificació. DeepSeek, en canvi, retorna respostes buides en 3 de les 5 consultes, mostrant una tendència més conservadora davant la manca d'informació.

Amb RAG, la millora és substancial en tots els models. LLaMA 2 és el més fiable amb 4 de 5 respostes correctes, proporcionant respostes precises i directes. Mistral encerta 3 de 5 però tendeix a generar contingut addicional no sol·licitat (tradueix respostes a altres idiomes, inventa preguntes complementàries). DeepSeek encerta 2 de 5 amb respostes ben contextualitzades quan acerta, però inventa dades inexistents en algunes consultes. Phi-2 també encerta 2 de 5 però presenta problemes de repetició en bucle i barreja d'idiomes.

6.3 Integració final: interfície, RAG i RBAC

A partir dels resultats de la comparativa, el sistema complet s'ha integrat utilitzant LLaMA 2 com a model de generació, en ser el de millor equilibri entre qualitat i velocitat. La interfície conversacional, el mòdul RAG i el control d'accés per rols funcionen de manera conjunta: l'usuari inicia sessió amb un rol, formula una consulta en llenguatge natural i el sistema recupera únicament els fragments dels documents als quals aquell rol té accés, els passa a LLaMA 2 i retorna una resposta redactada i fonamentada en les dades. Les proves de validació confirmen que un mateix interrogant produeix resultats diferents segons el rol: quan un perfil no té accés al document que conté la informació, el sistema no recupera cap fragment d'aquell document, de manera que el control d'accés actua a la capa de recuperació i no únicament com a filtre de presentació.

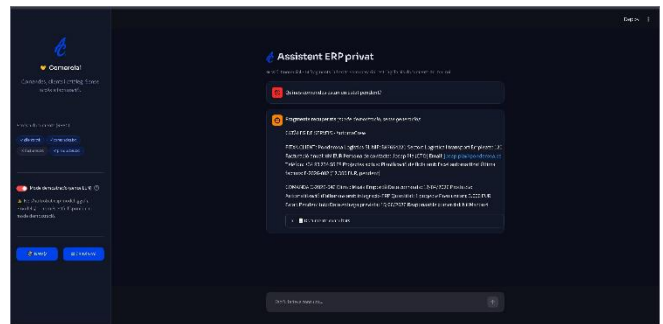


Figura 5. Resultat per al rol comercial davant la consulta «Quines comandes estan en estat pendent?»: el sistema recupera fragments del document de comandes, al qual aquest rol té accés.

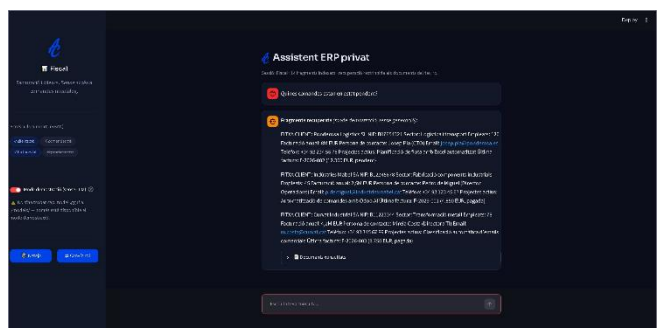


Figura 6. Resultat per al rol fiscal davant la mateixa consulta: en no tenir accés al document de comandes, el sistema només recupera informació dels documents permisos (factures i clients).

Com a exemple tenim la consulta «Quines comandes estan en estat pendent?», un usuari amb rol comercial obté els fragments del document de comandes, mentre que un usuari amb rol fiscal —que no hi té accés— rep únicament

informació dels documents permesos (factures i clients) i en cap cas fragments del document de comandes (Figures 5 i 6). Aquest comportament confirma que la restricció s'aplica a la fase de recuperació i no com a simple filtre de presentació.

Per a aquesta demostració, el sistema s'ha executat en mode de recuperació sense generació, de manera que es mostren directament els fragments recuperats per a cada rol; això permet evidenciar que el control d'accés opera sobre la recuperació de documents, amb independència del model de llenguatge que els processa posteriorment.

7 DISCUSSIÓ

7.1 Aportació del sistema RAG

Els resultats confirmen que el sistema RAG aporta un valor diferencial clar en els quatre models avaluats: transforma models que generen al·lucinacions en assistents capaços de proporcionar respostes fonamentades en dades reals. Sense RAG, cap model és útil per a consultes empresarials; amb RAG, LLaMA 2 assoleix un nivell de correcció del 80%. Aquest resultat és coherent amb Lewis et al. [7], que van demostrar que l'arquitectura RAG millora la precisió dels models en tasques que requereixen coneixement específic, un patró que es confirma en tots els models avaluats.

7.2 Comparació entre models

El rànquing de qualitat amb RAG és: LLaMA 2 (4/5) > Mistral (3/5) > DeepSeek = Phi-2 (2/5). LLaMA 2 destaca per la concisió i precisió de les respostes; Mistral ofereix respostes correctes però verboses; DeepSeek inventa dades addicionals; i Phi-2 pateix de repetició en bucle. Aquests resultats es poden posar en relació amb la literatura sobre escala de models: Kaplan et al. [11] estableixen que el rendiment segueix lleis d'escala predictibles segons el nombre de paràmetres, cosa que explica per què LLaMA 2 (7B) supera Phi-2 (2,7B) en correcció. Tot i això, Mistral [5] matisa aquesta relació: tot i tenir el mateix nombre de paràmetres que LLaMA 2, obté una puntuació inferior, fet que suggereix que l'arquitectura i les dades d'entrenament també hi juguen un paper rellevant.

7.3 Rendiment i recursos

Pel que fa a la velocitat, Phi-2 és el model més ràpid per la seva mida inferior. La comparació de temps entre models de 7B en float16 (LLaMA 2) i en GGUF 4-bit (Mistral, DeepSeek) no és directament equiparable, ja que utilitzen llibreries d'inferència diferents. Tot i això, es pot concloure que la quantització GGUF permet executar models de 7B dins de les restriccions de memòria del clúster (8 GB) a costa d'un temps d'inferència superior. En conjunt, per a un desplegament empresarial real amb les restriccions avaluades, LLaMA 2 7B amb RAG és el model recomanat: ofereix el millor equilibri entre qualitat de resposta i velocitat

d'inferència.

7.4 Limitacions

Aquest treball presenta diverses limitacions que cal tenir en compte. En primer lloc, el conjunt d'avaluació és reduït (5 consultes), de manera que els resultats tenen un caràcter indicatiu i no estadísticament concloent. En segon lloc, la correcció s'ha valorat manualment per un únic avaluador, fet que introdueix subjectivitat; l'ús de mètriques automàtiques o de diversos avaluadors en milloraria la fiabilitat. En tercer lloc, els temps s'han mesurat en CPU per les restriccions del clúster, de manera que no reflecteixen el rendiment esperable en producció amb GPU. Finalment, la comparativa combina models en float16 i en GGUF 4-bit per limitacions de memòria, una asimetria que condiciona la comparació directa de temps. Tot i això, els resultats permeten extreure conclusions clares sobre l'aportació del RAG i el comportament relatiu dels models.

8 ANÀLISI DE VIABILITAT COM A SERVEI SAAS

Un dels interrogants plantejats al títol i als objectius del treball és si l'arquitectura desenvolupada es podria transformar en un servei SaaS (Software as a Service) multiempresa. A partir de l'experiència d'aquest projecte, es considera que aquesta evolució és tècnicament viable, tot i que presenta reptes d'enginyeria concrets que cal abordar.

El nucli del sistema —la indexació amb embeddings, la base de dades vectorial, la recuperació i la generació amb el LLM— és independent de l'origen concret de les dades: opera sobre documents textuals i no sobre l'estructura interna d'un ERP determinat. Aquesta independència fa que el mateix motor pugui servir empreses diferents sense modificacions. A més, el control d'accés basat en rols ja proporciona la base per a la segregació de la informació, un requisit imprescindible en un entorn multiempresa on les dades de cada client han de quedar completament aïllades. En un model SaaS, aquesta segregació s'estendria a nivell de client (tenant): cada empresa disposaria del seu propi índex vectorial i del seu conjunt de rols, garantint que cap consulta pugui accedir a dades d'una altra organització.

La principal dificultat per a aquesta generalització rau en el connector de dades. Cada sistema ERP del mercat (LIBRA, SAP, Microsoft Dynamics, Odoo, etc.) utilitza un model de dades, una estructura i uns mecanismes d'accés diferents, de manera que el connector desenvolupat per a un ERP concret no és directament reutilitzable. Una solució SaaS requeriria una capa d'integració genèrica capaç de connectar-se a múltiples ERPs —mitjançant connectors específics per a cada sistema o una capa d'abstracció comuna— i de transformar dades heterogènies en el format textual que el mòdul RAG indexa. Aquest és l'esforç d'enginyeria més rellevant per a l'escalabilitat del producte, però és un problema acotat i abordable: es tracta de desenvolupar i mantenir un catàleg de connectors, no de redisenyar el nucli del sistema.

Altres aspectes a considerar serien l'escalabilitat de la inferència —el cost computacional dels models de llenguatge amb molts usuaris concurrents, que podria requerir infraestructura amb GPU o models optimitzats—, el manteniment dels connectors davant l'evolució dels ERPs, i les garanties de privacitat i compliment normatiu (per exemple, el RGPD) en allotjar dades de múltiples empreses. En conjunt, però, es conclou que la transformació en SaaS és viable: l'arquitectura modular i la separació entre el nucli i la font de dades fan que el repte principal sigui la generallització de la connexió amb els diferents ERPs, mentre que la resta de components són reaprofitables. Aquesta línia es proposa com a evolució natural del treball.

9 CONCLUSIONS

S'han assolit els quatre objectius específics plantejats. L'OE1 (disseny i implementació del sistema RAG) està completat: el pipeline de chunking, generació d'embeddings amb Sentence-BERT i indexació amb FAISS funciona correctament i recupera documents rellevants per a les consultes. L'OE2 (comparativa de models) s'ha completat amb els quatre models previstos (LLaMA 2, Mistral, Phi-2 i DeepSeek), obtenint resultats que demostren la contribució del RAG i identifiquen LLaMA 2 com el model més fiable. L'OE3 (implementació del control d'accés per rols) s'ha assolit: el RBAC s'aplica a la fase de recuperació, de manera que cada rol només accedeix als documents autoritzats. L'OE4 (validació del sistema complet) s'ha assolit integrant la interfície conversacional amb el RAG i el RBAC i validant-ne el funcionament sobre un conjunt de dades representatiu. Per motius de confidencialitat, la connexió amb les dades reals de l'ERP no s'ha realitzat dins l'abast d'aquest treball, però l'arquitectura modular del sistema deixa la solució preparada per a aquesta integració en un entorn de producció sense canvis estructurals.

Les principals dificultats trobades han estat d'infraestructura: la incompatibilitat de la GPU del clúster amb PyTorch per a Python 3.12 i les restriccions de memòria (8 GB per job), que van requerir l'ús de models quantitzats (GGUF 4-bit) per a Mistral i DeepSeek, introduint una asimetria en les condicions d'avaluació que cal tenir en compte en la interpretació dels resultats.

Com a línies de treball futur es proposa: la integració directa amb l'ERP LIBRA en un entorn de producció, l'ampliació del conjunt de consultes d'avaluació, l'ús de mètriques automàtiques amb respostes de referència, i el desenvolupament d'una capa de connectors genèrica per a múltiples ERPs com a pas previ a una oferta SaaS multiempresa.

BIBLIOGRAFIA

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, i I. Polosukhin, "Attention is all you need," a Proc. Advances

in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[2] A. Radford, K. Narasimhan, T. Salimans, i I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018.

[3] OpenAI, "GPT-4 technical report," arXiv:2303.08774, 2023.

[4] H. Touvron, L. Martin, K. Stone et al., "LLaMA2: Open foundation and fine-tuned chat models," arXiv:2307.09288, 2023.

[5] A. Q. Jiang, A. Sablayrolles, A. Mensch et al., "Mistral 7B," arXiv:2310.06825, 2023.

[6] Y. Li, S. Bubeck, R. Eldan, A. Del Giorno, S. Gunasekar, i Y. T. Lee, "Textbooks are all you need II: Phi-1.5 technical report," arXiv:2309.05463, 2023.

[7] P. Lewis, E. Perez, A. Piktus et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," a Proc. NeurIPS, 2020, pp. 9459–9474.

[8] Y. Gao, Y. Xiong, X. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.

[9] N. Reimers i I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," a Proc. EMNLP, 2019, pp. 3982–3992.

[10] J. Johnson, M. Douze, i H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[11] J. Kaplan, S. McCandlish, T. Henighan et al., "Scaling laws for neural language models," arXiv:2001.08361, 2020.

[12] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," a Proc. EMNLP, 2020, pp. 6769–6781.

[13] N. F. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," Trans. of the ACL, vol. 12, 2024.

[14] DeepSeek-AI, "DeepSeek LLM: Scaling Open-Source Language Models with Longtermism," arXiv:2401.02954, 2024.

APÈNDIX

A1. Consultes d'avaluació

Les 5 consultes utilitzades per avaluar els models de llenguatge en les dues condicions (amb RAG i sense RAG) són les següents:

1. “Quin és l'import total de la factura de Ponderosa Logistics?”
2. “Quins serveis ofereix AutomaCrew i quin és el seu preu?”
3. “Qui és la persona de contacte de Curvat Industrial?”
4. “Quines comandes estan en estat pendent?”
5. “Quina és la data de venciment de la factura F-2026-003?”

Aquestes consultes cobreixen les principals categories d'informació d'un ERP: facturació (consultes 1 i 5), catàleg de serveis (consulta 2), dades de clients (consulta 3) i gestió de comandes (consulta 4).