

CHATBOT PARA A DOCUMENTACIÓN E NORMATIVA DA UDC

Marcelo Ferreiro Sánchez

Portavoz do grupo

marcelo.fsanchez – marcelo.fsanchez@udc.es

Marcos Grobas Martínez

marcos.grobas – marcos.grobas@udc.es

José Romero Conde

j.rconde – j.rconde@udc.es

Resumo

O obxectivo do proxecto é desenvolver un chatbot capaz de solventar dúbihdas acerca do funcionamiento dos procesos burocráticos e de documentación da Universidade da Coruña (UDC). Para conseguilo optamos por unha arquitectura xenerativa aumentada por recuperación (RAG), técnica moi empleada para mellorar o desempeño de chatbots baseados en LLM cando buscan información específica dun dominio sobre o que o modelo de linguaxe orixinal non foi entrenado.

Índice xeral

1. Introdución	1
2. Solución proposta	2
2.1. A Arquitectura	2
2.2. O <i>Crawler</i>	2
2.2.1. Selección de páxinas relevantes	3
2.2.2. Procesado dos contidos de cada páxina	3
2.2.3. Funcionamento interno do <i>Crawler</i>	5
2.3. Análise dos datos obtidos mediante o <i>Crawler</i>	6
2.4. O <i>RAGsystem</i>	8
2.4.1. Construcción da base de vectores	8
2.4.2. O sistema de recuperación	9
2.5. Os modelos	10
2.6. Ferramentas usadas	11
3. Instalación e uso	12
4. Validación e resultados	13
4.1. Validación do sistema RAG	13
4.1.1. Complexidade da validación	13
4.1.2. Conxunto de validación	13
4.1.3. Evaluación do módulo de <i>retrieval</i>	14
4.1.4. Evaluación da xeración	14
4.1.5. Uso do mecanismo de avaliación como métrica	15
4.1.6. Consideracións sobre o corpus reducido	17
4.1.7. Limitacións	17
5. Conclusions e traballos futuros	18
5.1. Extensiós arquitectónicas e metodolóxicas	18
5.2. Extensiós da validación	19
A. Máis información acerca do <i>crawler</i>	22
A.1. <i>Keywords</i> utilizadas polo crawler	22
A.2. Saída OCR da táboa de exemplo	24
A.3. Progresión de páxinas <i>retrieveadas</i> ao longo do crawleo	24
B. Análise do Corpus	26

B.1. Distribución de tipos de ficheiros	26
B.2. Palabras más frecuentes	27
B.3. Palabras menos frecuentes	28
B.4. Resultados validación RAGsystem	28
B.4.1. <i>Heatmaps</i> de varias métricas	28
B.4.2. <i>Surface plots</i> de varias métricas	29
B.4.3. Comparación de métricas para diferentes valores de perso de BM25 co mellor <i>chunk size</i> (2048)	30

1. Introducción



burocracia de calquera campo pode chegar a ser moi complexa e pode chegar a consumir moito tempo e recursos ás persoas que teñen que lidiar con ela. Os procesos universitarios non son unha excepción e moitas das persoas involucradas neles (sexan, alumnos, profesores ou persoal administrativo) os poden atopar inabarcables ou imposibles de navegar sen axuda.

Neste contexto, aprecioouse como podería ser de gran utilidade o desenvolvemento dun chatbot capaz de responder a preguntas relacionadas coa documentación e normativa da Universidade seguindo a gran tendencia da actualidade de empregar chatbots para diversas tarefas.

O obxectivo deste proxecto é desenvolver un chatbot que poida simplificar e explicar **referindo sempre ás fontes burocráticas oficiais** os procesos burocráticos e de documentación da Universidade da Coruña (UDC).

O sistema é resultado da unión de compoñentes e técnicas xa ben coñecidas, sempre tendo en mente o obxectivo a cumplir. Polo tanto, tratouse máis ben dunha tarefa de aplicación e adaptación de técnicas e ferramentas xa existentes nun caso particular, antes que de deseño ou resolución de novos problemas.

Se ben este traballo está circunscrito no contexto da asignatura de Técnicas Avanzadas de Procesamiento de Linguaxe Natural (TAPLN), debido á súa natureza e obxectivo, gran parte do tempo invertido no proxecto dedicouse á tarefa de recolección de información para o RAG.

Non existindo unha 'base de datos' oficial da UDC sobre a que un usuario poda descargar toda a documentación relativa ao centro, senón que atópase esta repartida nas páxinas web dos seus diferentes centros, foi necesario deseñar unha solución que nos permita extraela a partir dos seus portais oficiais.

Este tipo de tarefas non son novas no mundo da informática, de feito pertencen a unha área más que consolidada chamada Recuperación de Información (IR) e tales métodos que buscan, extraen e organizan información disposta en webs html son coñecidos como *crawlers*, parte esencial do traballo final pois para asegurar que o sistema sexa capaz de responder á maioría de dúbidas dos usuarios, é necesario ter un corpus de toda información mínimamente relevante acerca do funcionamento interno da universidade.

2. Solución proposta



ostrarase nesta sección a arquitectura xeral do sistema proposto e, a continuación, describirase en detalle cada un dos seus componentes principais, sendo estes o *Crawler* e o *RAG system*. O obxectivo desta arquitectura é combinar un mecanismo de adquisición automática de información con un modelo de linguaxe capaz de xerar respuestas fundamentadas en contido externo e actualizado.

2.1 A Arquitectura

De maneira xeral, o sistema compõe de dúas partes funcionalmente diferenciadas pero fortemente interconectadas. Por unha banda, o sistema de *crawling*, encargado de explorar e percorrer de forma automatizada os distintos portais e páxinas web da Universidade da Coruña (UDC), co fin de descargar e almacenar aqueles documentos que potencialmente conteñen información relevante. Por outra banda, o *RAG system*, concibido como unha pipeline de procesamento da información que integra mecanismos de recuperación e xeración, é o responsable de atender as consultas dos usuarios e producir respuestas en linguaxe natural apoiadas en contido externo recuperado.

Nas siguientes subseccións explicaranse en detalle o funcionamento interno de cada parte e cómo interactúan entre elas. Optouse por unha orde de exposición que siga o fluxo de execución do sistema, comezando polo *Crawler* e continuando polo *RAG system*. Aínda que se considerou presentar o desenvolvemento seguindo a orde cronolóxica da implementación, descartouse esta opción ao introducir unha carga expositiva innecesaria e dificultar a comprensión global da arquitectura. De calquera forma, explicarase a evolución das partes que máis cambios sufriron ao longo da súa implementación.

2.2 O *Crawler*

Considerouse que a posesión dun bo volume de datos sería crucial para resolver o problema a tratar, na maioría dos casos as dúbidas burocráticas resólvense encontrando o documento adecuado, e de non telo, o sistema veríase obrigado a comunicarlle ao usuario de que non dispón da información solicitada, limitando significativamente a súa capacidade para ofrecer respuestas útiles. É por iso que a Recuperación de Información xoga un papel crucial para o RAG.

2.2.1 Selección de páxinas relevantes

Sendo así, o primeiro problema a abordar é o deseño dun criterio axeitado para determinar que se considera un **documento relevante** no contexto da aplicación. Trátase dunha cuestión que pode complicarse *ad infinitum* e que continúa a ser unha área de estudio activa na literatura recente [1]. Porén, no marco deste traballo optouse por un enfoque moito máis sinxelo e pragmático.

Esta decisión baséase na premisa de que a tarefa non require o manexo de conxuntos masivos de datos nin a exploración exhaustiva da web. Abonda con navegar un número limitado de URLs, principalmente pertencentes aos directorios raíz das distintas facultades e servizos da UDC, polo que resulta aceptable descargar páxinas e documentos potencialmente pouco relevantes. Ademais, o volume total de información a recoller presenta un límite de tamaño finito e razoable. Mesmo nun escenario extremo no que se descargasen todas as URLs asociadas á UDC, o espazo de almacenamento necesario estaría moi lonxe das ordes de magnitude propias dun *crawler* de propósito xeral, onde si se require xestionar volumes de datos a escala de terabytes. Esta abordabilidade é posible grazas a que, unha vez descargadas, as páxinas e documentos son procesados de forma estruturada, permitindo unha xestión eficiente do contido.

Deste xeito, a solución proposta concíbese como un método sinxelo pero funcional, baseado nunha asignación de relevancia binaria (*relevante ou non relevante*) a partir de *keywords*. A pesar da súa simplicidade, este enfoque está estreitamente relacionado cos métodos pioneiros de *focused crawling* [2], que sentaron as bases para a recuperación de información temática na web.

As palabras clave seleccionadas son fixas e específicas do ámbito académico e do vocabulario burocrático da universidade, aparecendo en galego, castelán e inglés. Para consultar a lista completa, véxase o Apéndice A.1.

2.2.2 Procesado dos contidos de cada páxina

Unha vez unha páxina se considera relevante, é necesario procesar o seu contido para presentalo, sempre que sexa posible, en texto plano, evitando decoradores e elementos innecesarios –como menús, cabeceiras, pés de páxina, publicidade ou elementos visuais que non aportan contido relevante–. Para iso, o *crawler* encárgase de *parsear* a estrutura da páxina, extraendo o contido textual relevante de HTML e outros elementos web. No caso de uso deste traballo, moita información burocrática útil para o usuario final atópase en documentos PDF ligados ao URL; por iso, o *crawler* tamén se encarga de extraer, parsear e converter estes ficheiros a texto plano. Dispoñer do contido en texto plano permite posteriormente dividir os documentos en *chunks*, xerar *embeddings* e construír *vector stores*, permitiendo así a recuperación e xeración eficiente de respuestas no sistema RAG.

Non obstante, este enfoque presenta unha limitación importante: calquera información que apareza exclusivamente en formato de imaxe (capturas de pantalla, carteis

dixitalizados ou documentos escaneados) permanece invisible para o *crawler*. Esta debilidade fixose evidente nas primeiras fases de probas do *chatbot*, cando o sistema non foi capaz de responder a unha pregunta básica para estudiantes: “Cantos libros pódense emprestar?”. Tras investigar o caso e verificar que a **páxina correspondente** fóra efectivamente analizada polo *crawler*, comprobouse que, aínda que a información figura nunha táboa dentro da páxina, esta estaba en formato imaxe (véxase 2.1). Este exemplo evidencia que é necesario aproveitar mellor a información dispoñible en cada páxina para cumplir os obxectivos establecidos.

Tipos de usuarios	Nº de documentos en préstamo	Días de préstamo	Renovaciones	Reservas
GRUPO 1				
Estudiantado de Grado de centros propios y adscritos				
Estudiantado de programa de movilidad (Erasmus, Sigue-Séneca)				
Estudiantado de doble Grado, de simultaneidad				
Estudiantado de grados interuniversitarios				
Estudiantado da Universidad Sénior				
GRUPO 2				
Estudiantado de MÁSTERES e posgrados propios, incluyendo los de la Fundación Universidad de Coruña				
Estudiantado de Trabajos de fin de grado				
Personal de administración en servicio (PTGAS)				
GRUPO 3				
Estudiantado de Doctorado	35	30 días	Indefinidas	Límite 35 docs.
Personal investigador visitante: visitante senior/ visitante predoctoral o postdoctoral	35	30 días	Indefinidas	Límite 35 docs.
GRUPO 4				
Becarios/as de investigación				
Personal contratado investigador	35	Curso académico (cada biblioteca podrá excluir de este tipo de préstamo documentos por razón de uso y disponibilidad)	Indefinidas	Límite 35 docs.
GRUPO 5				
PDI da UDC (incluyendo centros adscritos), de la Fundación UDC				
Profesorado emérito, jubilado incentivado y honorario				
Profesorado visitante				
Lectores/as	100	Curso académico (cada biblioteca podrá excluir de este tipo de préstamo documentos por razón de uso y disponibilidad)	Indefinidas	Límite 100 docs.
GRUPO 6				
Cualquier persona ajena a la comunidad universitaria de la UDC que sea autorizada por la Biblioteca Universitaria	6	10 días	Indefinidas	Límite 6 docs.

Figura 2.1 Táboa de préstamos

Procesado de imaxes mediante OCR

O recoñecemento óptico de caracteres (OCR, polas súas siglas en inglés) é unha técnica que permite extraer texto a partir de imaxes dixitais. A súa aplicación máis habitual é a dixitalización de documentos físicos escaneados, mais tamén pode empregarse en calquera imaxe que conteña caracteres, como é o caso neste traballo. Para este fin, implementouse unha interface en *Python* empregando a librería *pytesseract*, que actúa como envoltorio do motor OCR Tesseract [3], inicialmente creado por HP e actualmente mantido por Google.

Desde o punto de vista computacional, o procesado mediante OCR resulta asumible. A maioría das imaxes presentes nos documentos son pequenas e irrelevantes, polo que a súa análise non introduce unha latencia significativa no sistema. A modo ilustrativo, o procesado OCR dunha imaxe tabular real (Figura 2.1) presentou un tempo de execución de 0,6280 segundos, o que confirma a viabilidade práctica desta aproximación.

Durante o desenvolvemento do sistema observouse que unha parte relevante da información aparece en forma de táboas embebidas como imaxes. Nunha primeira

aproximación, estas imaxes procesábanse mediante OCR xeral, obtendo un texto plano sen estrutura explícita. Porén, este enfoque presentaba unha limitación importante: ao fragmentar posteriormente o texto en *chunks*, a estrutura lóxica da táboa perdíase, facendo que os fragmentos resultantes non contivesen contexto suficiente para unha interpretación correcta.

Para mitigar este problema, adoptouse un enfoque de OCR en dúas modalidades. Por unha banda, incorpórarse un OCR especializado na detección de estruturas tabulares, capaz de identificar filas, columnas e relacións internas. Por outra banda, mantense un OCR xeral para aquelas imaxes que non presentan unha estrutura tabular clara, como infografías ou imaxes con texto libre. Este deseño permite extraer información textual de forma robusta e adaptada á natureza de cada imaxe, preservando a estrutura cando esta existe e contribuíndo a eficiencia global do sistema.

2.2.3 Funcionamento interno do *Crawler*

A implementación proposta usa da librería *Python BeautifulSoup* para navegar a rede e extraer información das páxinas. O proceso comeza cunha lista de URLs semente que representa os portais principais da universidade (concretamente a *homepage* de cada facultade). Para cada URL, o sistema descarga o contido HTML, extrae todos os enlaces e imaxes presentes na páxina, e identifica documentos PDF que conteñan termos relacionados coa burocracia universitaria segundo a lista de *keywords* antes exposta. Os recursos descargados almacénanse de forma organizada no sistema de ficheiros local, mentres que un sistema de metadatos rexistra información sobre cada recurso para optimizar futuras execucións, concretamente garda o *etag*, a data de última modificación de cada páxina, a data de descarga e o hash do contido.

En tanto ás imaxes, aplícase OCR a todas as presentes en cada documento. O proceso realiza en dúas fases consecutivas: primeiro procésase coa libraría *img2table* (deseñada para extraer texto de táboas en imaxes) e, se esta primeira fase non extrae un número mínimo de caracteres, entón aplícase un segundo OCR utilizando *pytesseract* de xeito estándar. O texto extraído por OCR intégrase directamente no contido textual da páxina á que pertence a imaxe (sempr e cando alcance un número mínimo de caracteres definido): engádese ao ficheiro de texto correspondente cun *header* que indica a súa orixe ([TÁBOA OCR] ou [IMAXE OCR]), permitindo así conservar tanto o texto HTML como o contido visual nun único documento estruturado.

Todo este procesamento realiza ao final de *crawlear* cada páxina: primeiro extráense as hiperreferencias a outras páxinas e documentos PDF, e despois procesanse as imaxes encoladas para OCR. Esta estratexia minimiza o cuello de botella que se produciría se o OCR se realizase de forma síncrona mentres se descargan outros recursos.

Para acelerar o proceso de *crawling* aplicáronse técnicas de programación concurrente. É posible establecer un número de traballadores (*workers*) aos cales se lles asigna unha URL como punto de partida, permitindo múltiples crawlers simultáneamente navegando a rede. Para evitar problemas de escritura nos ficheiros de metadatos ou

rexistro de páxinas visitadas, implementáronse *locks*, garantindo que dous crawlers non poidan escribir á vez nun mesmo recurso.

Antes de procesar cada documento, comprobanse os metadatos para determinar se o contido cambiou desde o último crawleo. Primeiro verifica os campos `last_modified` e `etag`; se se detecta unha modificación, extráese o texto completo e calcúlase un hash do contido. Se este hash é diferente do hash previo, a variable booleana `needs_embeddings` actívase, indicando que o documento debe ser vectorizado no seguinte paso. Deste xeito, evítase xerar embeddings para documentos que non cambiaron, optimizando tanto o tempo de procesamento como o uso de recursos.

En conxunto, esta combinación de crawling distribuído, OCR en dúas fases, xestión eficiente de imaxes e metadatos, e comprobación de cambios mediante hash garante que o crawler funcione de forma rápida, scalable e eficiente, mantendo a información relevante organizada e lista para o procesamento posterior, como se mostra nas gráficas do apéndice A.1.

2.3 Análise dos datos obtidos mediante o *Crawler*

Tras facer un *crawl* sobre as *homepages* das facultades e escolas da UDC, obtiveronse un total de 4.966 documentos, dos cales 2.668 son páxinas *HTML* e 2.298 documentos *PDF* (véxase B.1).

Debido á relación coas técnicas clásicas de Recuperación da Información, realizouse unha análise exploratoria do vocabulario e da lonxitude dos documentos:

Cadro 2.1 Estadísticas del corpus

Métrica	Valor
Arquivos procesados	4.966
Total de caracteres	75.088.461
Total de palabras	10.164.265
Tamaño del vocabulario	73.640
Palabras únicas (<i>hapax legomena</i>)	15.889

Atopáronse 10.164.265 palabras cun vocabulario de 73.640 termos, das cales 15.889 aparecen só unha vez (*hapax legomena*), un 20 % do vocabulario. A maioría non son errores: só 230 son faltas ortográficas e 30 xeradas polo OCR. Este fenómeno de vocabulario de uso desigual segue a lei de Zipf, onde poucas palabras son moi frecuentes e moitas son raras, como se mostra nas Figuras 2.2 e 2.3.

Tamén se observa o principio de Pareto: aproximadamente o 20 % das palabras más frecuentes representan arredor do 80 % das ocorrencias, aínda que neste corpus a relación é máis extrema: o 2,1 % do vocabulario alcanza o 80 % das ocorrencias e o 20 % cubre o 97,5 %.

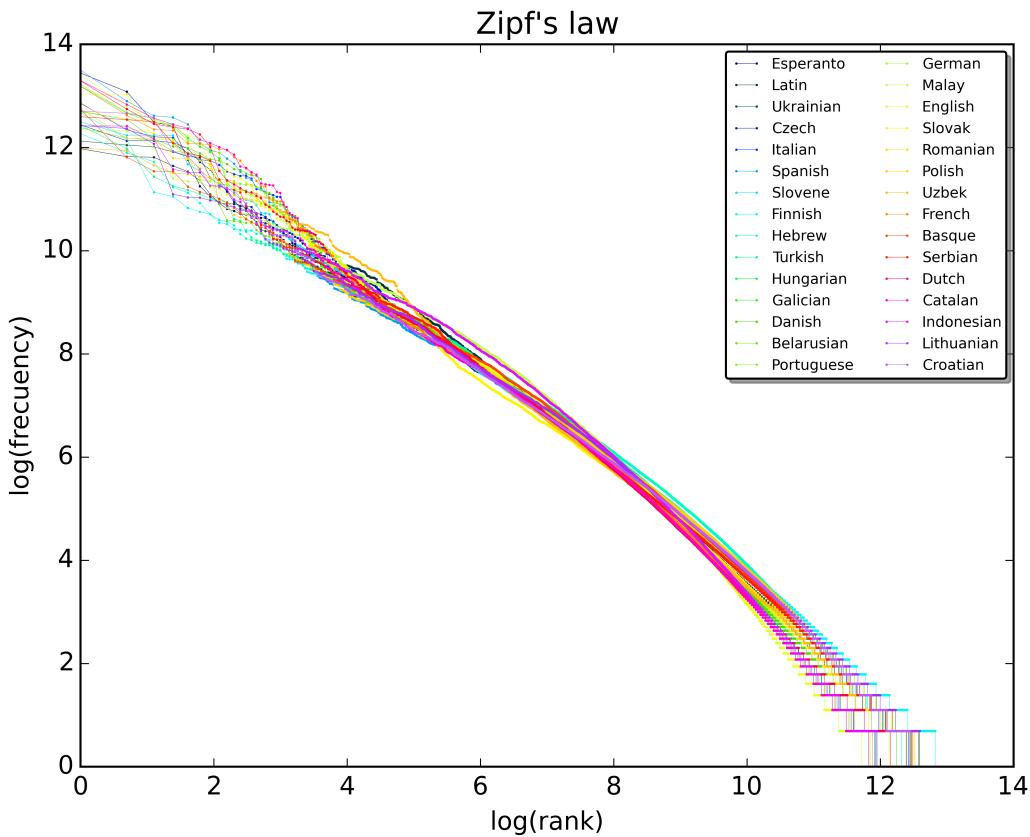


Figura 2.2Frecuencia de termos en diversas lingüas [4]

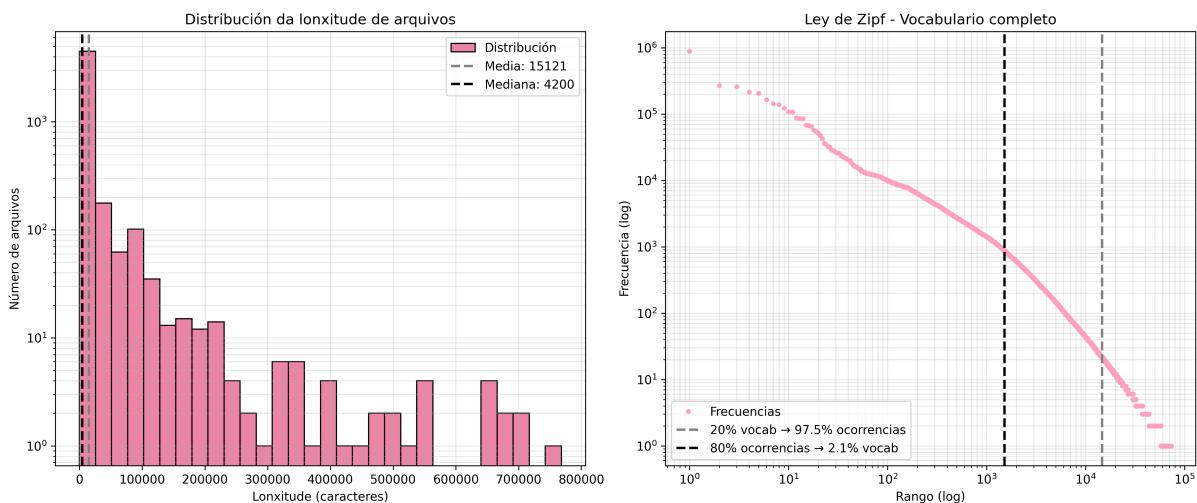


Figura 2.3Gráficos de distribución de lonxitude de documentos e frecuencia de termos

A distribución das lonxitudes dos documentos (Figura 2.3, esquerda) mostra moitos documentos curtos: mediana de 4.200 caracteres e media de 15.121, indicando asimetría debido a documentos longos que elevan a media. Moitas páxinas web teñen contido limitado ou só mostran ligazóns ou imaxes. No *crawleo* detectáronse 115 páxinas baleiras (0 caracteres), ben por conter só imaxes sen texto extraíble por OCR, ben por ser redireccións ou navegación pura.

2.4 O RAGsystem

O sistema RAG (Retrieval-Augmented Generation) implementado constitúe o núcleo do chatbot, combinando técnicas de recuperación de información con modelos de linguaxe xerativos. O proceso de resposta a unha consulta desenvólvese nun fluxo secuencial e optimizado:

Inicialmente, a consulta do usuario pasa por un módulo de *clasificación* realizado por un modelo de linguaxe lixeiro (Claude Haiku) para determinar se require recuperación de documentación. Cando é necesaria, a mesma consulta é *reescrita* (*query rewriting*) para mellorar a súa formulación tanto para a busca semántica como léxica.

Para a recuperación, empregase un sistema *híbrido* que combina busca densa (mediante embeddings semánticos e similitude coseno en FAISS) e busca léxica (usando BM25, unha evolución do TF-IDF). As puntuacións de ambos métodos fusiónnanse con pesos determinados empiricamente, seleccionando os k fragmentos de texto más relevantes da base de documentos da universidade.

Os documentos recuperados, xunto coa consulta orixinal e o historial de conversación (de lonxitude configurable), intégranse nun prompt estruturado. Este pasa a un modelo de linguaxe grande (Claude Sonnet) escollido pola súa capacidade de razonamento e, especialmente, pola súa efectividade na comunicación entre axentes e o seguimento de instrucións complexas [5]. O modelo xera así unha resposta precisa fundamentada na documentación oficial.

O sistema enriquece cada documento recuperado con metadatos que inclúen as súas procedencias (hipervínculos), datas (de actualización da páxina), puntuacións de relevancia tanto da busca densa como da léxica, garantindo total trazabilidade e permitindo a verificación das fontes utilizadas en cada resposta.

2.4.1 Construcción da base de vectores

[6] A base de vectores (*vectorstore*) constrúese a partir dos documentos recollidos e previamente convertidos a texto plano (.txt) polo *crawler*. Cada documento é fragmentado en varios *chunks* de tamaño configurable, establecéndose tamén un *overlap* que especifica cantos *tokens* comparten os fragmentos consecutivos. A utilidade deste solapamento é impedir que referencias cruzadas ou ideas contidas nun texto queden cortadas arbitrariamente ao final dun *chunk*, xa que nese caso o LLM non podería comprender o contexto completo.

Cada fragmento de texto (*chunk*) xera, ademais do seu contenido, un conxunto de metadatos que inclúen: o hipervínculo de orixe da páxina, a data da última actualización do documento orixinal (obtida durante o *crawling*) e outros campos auxiliares para a trazabilidade. Estes metadatos almacénanse xunto co vector resultante e son recuperados co texto durante a busca, o que permite ao sistema referenciar a fonte exacta e avaliar a vixencia da información.

Para o almacenamento e indexación eficiente dos *embeddings*, emprégase FAISS (*Facebook AI Similarity Search*) [7], unha biblioteca optimizada para a busca de similaridade en espazos vectoriais de alta dimensionalidade. FAISS implementa algoritmos de busca aproximada de veciños más próximos (ANN, *Approximate Nearest Neighbors*), que permiten realizar consultas en millóns de vectores de forma eficiente. O proceso funciona do seguinte xeito: cada *chunk* de texto convértese nun vector denso (*embedding*) mediante un modelo de linguaxe especializado; estes vectores almacénanse, xunto cos seus metadatos, nunha estrutura de índice optimizada; e cando se realiza unha consulta, FAISS calcula rapidamente os vectores más similares utilizando medidas de distancia como a similaridade coseno ou a distancia euclidiana [8].

2.4.2 O sistema de recuperación

O sistema polo que o RAG recibe os documentos relevantes para cada consulta recibe múltiples iteracións e cambios tras varias fases de testeo e replanteamento teórico. Trátase dun compoñente clave do sistema e non ten unha solución trivial, pois é un área de investigación moi activa [9]. Poderíase dicir que, áinda que a área de RI (Recuperación de Información) parecía estar algo estancada nos últimos anos debido aos bons resultados que alcanzaron os buscadores web, agora volve a verse moi activa grazas ao xurdimento dos sistemas RAG.

Inicialmente, o sistema empregou un método de recuperación baseado exclusivamente na similitude coseno sobre vectores FAISS. Porén, durante as primeiras probas comprobouse que esta aproximación podía non ofrecer os resultados esperados, xa que daba a mesma importancia a todas as palabras da consulta, sen ter en conta cal podería ser a palabra clave máis relevante. A maioría das consultas, especialmente aquelas de carácter burocrático, conteñen moitas palabras comúns e repetidas ao longo da colección de documentos (como *proceso*, *documento* ou *normativa*), o que podía facer que o sistema recuperase documentos pouco relevantes.

Como primeira mellora, implementouse MMR (Maximal Marginal Relevance) en lugar da similitude coseno. A idea subxacente era que, deste xeito, non se recuperarían documentos redundantes entre si, xa que se existisen varios documentos moi similares, a súa información adicional para a consulta sería limitada. Non obstante, durante a experimentación observouse que esta suposición non se cumplía no noso dominio. É moi común atopar diferentes documentos oficiais que repiten a mesma información, ou varias versións e revisións dun proceso en distintos documentos. Neses casos, o sistema tendía a devolver só un deles, perdéndose información relevante ou, incluso, indicando un proceso xa obsoleto, ao priorizar a diversidade fronte á exhaustividade [10]. Esta problemática, relacionada coa vixencia temporal dos documentos, constitúe un dos maiores retos deste proxecto (e de calquera sistema RAG que manipule información cambiante ao longo do tempo [11]) e será tratada con maior detalle máis adiante, xa que require modificacións en diversas partes do sistema.

Unha vez comprobado que o MMR non era axeitado para o noso dominio, retomouse

o enfoque orixinal, pero buscando unha maneira de discriminar mellor entre os termos relevantes e irrelevantes da consulta. Para iso, decidiuse explorar métodos clásicos de recuperación, como o TF-IDF (Term Frequency – Inverse Document Frequency) [12]. A primeira aplicación consistiu en usalo para reordenar (*reranking*) os documentos xa recuperados por FAISS. Con todo, axiña se observou que esta estratexia non ofrecía resultados óptimos, xa que simplemente cambiaba a orde dos documentos recuperados, sen engadir novos documentos relevantes que puidesen ser ignorados pola similitude coseno.

A solución final adoptada foi un sistema híbrido que combina as puntuacións da similitude coseno (recuperación densa) e do BM25 (unha mellora sobre o TF-IDF para recuperación léxica ou dispersa), mediante pesos configurables. Despois dunha avaliación da literatura, no inicio, estableceuse un balance de 50/50 entre ambos métodos, un enfoque común en sistemas RAG actuais [13], [14], máis unha vez deseñado un *pipeline* para realizar validación, optouse por usar tal experimento para determinar empíricamente o mellor valor de peso, ademais do mellor *chunk size*.

Ademais, o fluxo de recuperación intégrase dentro dunha canle más ampla de procesamento da consulta. Antes de realizar a búsqueda, lévase a cabo unha *clasificación da consulta* (mediante un LLM lixeiro, como Claude Haiku) para determinar se é necesario recuperar documentos. En caso afirmativo, aplícase un *reescrito da consulta (query rewriting)*, co obxectivo de optimizar a súa formulación para a busca tanto no índice denso (FAISS) como no léxico (BM25). Os documentos recuperados por este sistema híbrido, xunto coa consulta orixinal, pásanlle finalmente a un modelo de linguaxe máis grande (como Claude Sonnet) para xerar a resposta final.

2.5 Os modelos

A elección dos modelos de linguaxe no sistema RAG realizouse atendendo a criterios de eficiencia, latencia e rendemento específico para cada tarefa do fluxo. O sistema emprega de xeito estratégico modelos de diferentes capacidades segundo a súa función dentro da arquitectura, optimizando así o balance entre precisión, tempo de resposta e custo.

Para as tarefas iniciais de procesamento da consulta (clasificación para determinar a necesidade de recuperación de documentos e reescritura (*query rewriting*) para mellorar a formulación da busca) utilízase **Claude Haiku**. Este modelo, sendo o máis rápido e económico da familia Claude de Anthropic, resulta adecuado para tarefas lixeiras de comprensión e transformación textual que non requieren razonamento profundo pero si baixa latencia.

Para a fase central de xeración da resposta empégase **Claude Sonnet**. Este modelo ofrece un equilibrio óptimo entre capacidades de razonamento, seguimento de instruccions complexas e eficiencia, sendo especialmente adecuado para sintetizar información de múltiples documentos e producir respuestas coherentes, precisas e ben fundamentadas. A súa capacidade para traballar con *prompts* estruturados que inclúen contexto

recuperado, historial de conversación e instrucións específicas de formato resulta clave para a calidade final do sistema.

A separación de responsabilidades entre modelos lixeiros para tarefas previas e modelos más capaces para a xeración final segue unha práctica común en arquitecturas de sistemas RAG, destinada a minimizar a latencia e o custo computacional sen comprometer a calidade das respostas [15], [16]. Neste proxecto, a elección de Claude Haiku e Sonnet implementa esta estratexia adaptándoa aos recursos e limitacións do noso entorno.

2.6 Ferramentas usadas

3. Instalación e uso

4. Validación e resultados

4.1 Validación do sistema RAG

4.1.1 Complexidade da validación

A validación dun sistema Retrieval-Augmented Generation (RAG) é intrinsecamente complexa debido á súa natureza híbrida, xa que combina dous subsistemas conceptualmente distintos pero fortemente acoplados: o módulo de *retrieval*, encargado de seleccionar a información relevante, e o módulo de *generation*, responsable de producir a resposta final condicionada polo contexto recuperado. Avaliar correctamente un sistema RAG implica analizar de forma separada e conxunta ambos compoñentes, así como a súa interacción, incrementando notablemente a dificultade do proceso de validación [17].

Ademais, unha avaliación rigorosa require a disposición de conxuntos de datos anotados e métricas específicas para cada etapa do pipeline. A literatura sinala que a definición de *ground truth* fiable —cun conxunto completo de documentos relevantes e respuestas correctas— é custosa en tempo e recursos, e que as métricas existentes poden verse afectadas pola variabilidade na anotación humana [17], [18].

4.1.2 Conxunto de validación

Para realizar unha avaliación práctica e reproducible dentro das limitacións do proxecto, construíuse un conxunto de validación reducido composto por **25 preguntas**. Cada pregunta está asociada a unha resposta esperada e ao documento concreto do corpus onde se atopa a información necesaria para resolvela. Na maioría dos casos, a información relevante localízase de forma explícita nun único documento.

É importante destacar que esta avaliación realizase sobre un **corpus reducido e controlado de documentos**. Este corpus representa só unha fracción do conxunto total de documentos que o sistema RAG manexará nun contexto real, onde se espera que haxa moitos máis documentos, algúns dos cales poderían recuperarse sen ser necesarios para responder a consulta concreta. Traballar cun corpus acotado permite, con todo, illar variables e analizar de forma precisa o comportamento do sistema. En particular, pódese estudar o impacto de parámetros como estratexias de recuperación, métricas de similitude, tamaño de chunk ou número de documentos recuperados, aspectos relevantes no deseño práctico de sistemas RAG [19].

Esta aproximación debe considerarse **subestimada e conservadora**, xa que en escenarios reais poderían existir múltiples documentos relevantes por consulta non cubertos neste estudo. As métricas obtidas non buscan medir o rendemento absoluto do sistema en producción, senón servir como base experimental para estudar o impacto relativo

das distintas decisións de deseño do RAG.

4.1.3 Evaluación do módulo de *retrieval*

O módulo de recuperación avalíase de forma independente mediante métricas clásicas de *Information Retrieval*, adaptadas ao contexto dun sistema RAG:

- **Recall@10:** proporción de documentos relevantes que aparecen entre los 10 primeiros resultados recuperados. Calculase como a fracción de documentos relevantes presentes no top-10 do ranking. Aínda que presentamos principalmente o **recall fraccional**, convén aclarar que neste conxunto de validación cada pregunta adoita ter un único documento relevante. Polo tanto, nesta situación o recall fraccional é praticamente equivalente ao **recall binario**, que indica se polo menos un documento relevante aparece entre os top-10. Esta métrica reflicte a capacidade do sistema de garantir que a información esencial chega ao modelo xerativo.
- **MRR (Mean Reciprocal Rank):** penaliza a aparición tardía do primeiro documento relevante no ranking. Calculase como o recíproco da posición do primeiro documento relevante atopado (1 se é o primeiro, 0.5 se é o segundo, etc.). Esta métrica reflicte a rapidez coa que o sistema proporciona información útil ao LLM.
- **Precision@10:** proporción de *chunks* recuperados que pertencen a documentos relevantes dentro dos 10 primeiros resultados. Avalía a capacidade do sistema para minimizar a recuperación de información irrelevante, especialmente crítico en corpus grandes ou ruidosos. Esta métrica é a nivel de chunk, reflectindo o nivel de “ruído” que chega ao LLM.

No noso sistema, o **recall** e o **MRR** calcúlase a nivel de documento, considerando cada documento como unha unidade discreta de relevancia. Isto simplifica a avaliación e garante que se mida a efectividade do sistema en levar información esencial ao modelo xerativo, independentemente de como os documentos estean fragmentados en chunks para a súa utilización polo LLM.

Pola súa banda, a **precision** calcúlase a nivel de chunk, como proporción de fragments pertencentes a documentos relevantes entre os recuperados. Este enfoque mixto (recall a nivel documento, precision a nivel chunk) proporciona unha visión útil para o desempeño do sistema RAG: asegura que a información clave está disponible para o LLM, ao mesmo tempo que avalía a calidade do material recuperado.

4.1.4 Evaluación da xeración

A avaliación do módulo de xeración céntrase en dúas dimensíons complementarias:

- **Answer Relevance:** mide se a resposta aborda correctamente a pregunta e evita información irrelevante. Esta métrica non xulga a veracidade, só a pertinencia.

- **Answer Faithfulness:** avalia se as afirmacións contidas na resposta están efectivamente respaldadas polo contexto recuperado polo módulo de *retrieval*, detectando posibles alucinacións ou información non soportada.

Mecanismo de avaliación: *LLM-as-judge*

Para automatizar a avaliación empregouse un modelo de linguaxe avanzado (**GPT-5.2**) como avaliador (*LLM-as-judge*), garantindo capacidade suficiente para xular de forma coherente e consistente as respostas do xerador [18]. Este enfoque permite unha avaliación máis detallada que as métricas clásicas, incorporando aspectos de completitude, coherencia e consistencia co contexto.

Faithfulness O LLM analiza cada resposta dividíndoa en afirmacións atómicas e comproba se cada unha está efectivamente respaldada polo contexto recuperado. O score final, entre 0 e 1, representa a proporción de afirmacións verificadas. Ademais, xérase unha explicación detallada que identifica que afirmacións fallaron e por que, permitindo unha maior transparencia na avaliación e a detección de posibles alucinacións ou inclusión de información externa ao contexto.

Relevance A relevancia mide se a resposta aborda todos os puntos da pregunta de forma concisa e útil, evitando información irrelevante ou redundante. Tamén retorna un score entre 0 e 1 e unha explicación que describe posibles omisións ou exceso de información.

Beneficios do enfoque O uso de *LLM-as-judge* permite:

- Avaliar fidelidade e pertinencia de respostas en linguaxe natural de forma consistente e reproducible.
- Detectar matices de completitude, coherencia e relevancia que métricas de IR tradicionais non capturan.
- Reducir a dependencia do etiquetado humano para a avaliación detallada de respostas.

4.1.5 Uso do mecanismo de avaliación como métrica

Anteriormente fixose referencia a como empregariamos o *pipeline* de avaliación para atopar un máximo no espacio de hiperparámetros do modelo, sendo estes o *chunk size* e o peso entre a busca densa e a dispersa.

Tal experimentación requiriu dunha *grid search* que mediou cinco métricas para cada combinación de valores: tres propias de Recuperación de Información (*Recall*, *Precision* e *MRR*) e dúas propias da avaliación con *LLM-as-judge* (*Faithfulness* e *Relevance*). Tras

promediar os valores para cada pregunta, puidéronse construir *heatmaps* para cada unha das medicións (ver Figura B.2).

métricas para valores máis altos do peso asignado a BM25 (busca léxica) fronte á busca densa (FAISS). A puntuación final de cada documento no sistema híbrido calcúlase mediante a combinación lineal:

$$Score_{final} = Score_{denso} + \alpha \cdot Score_{disperso} \quad (4.1)$$

onde α representa o peso relativo de BM25. Cando $\alpha > 1$, como no valor óptimo de 1.6 atopado, a busca léxica ten maior influencia na selección final. Isto indica que, para o dominio específico da documentación burocrática universitaria, a recuperación baseada en coincidencia de termos clave é especialmente efectiva. As consultas neste contexto adoitan incluír nomes propios de formularios, códigos de procedementos ou termos técnicos moi específicos, onde a recuperación léxica supera á semántica pura. Este fenómeno é consistente coa literatura que destaca a superioridade de BM25 en dominios con vocabulario especializado e baixa variación lexical [20]. Ademais, a natureza formal e estándar da linguaxe administrativa fai que os documentos relevantes compartan termos exactos coas consultas, situación onde os métodos de recuperación tradicional baseados en termos áinda ofrecen vantaxes sobre os enfoques puramente semánticos.

Debido a que é necesario elixir un par de valores para os hiperparámetros, optouse por aqueles que maximizan a métrica de *Relevance* obtida mediante *LLM-as-judge*. Esta decisión baséase na premisa de que a relevancia percibida da resposta final é o criterio de calidad más alineado coa experiencia do usuario final nun sistema de preguntas e respostas. Mientras que métricas tradicionais de RI como *Precision* e *Recall* miden a eficacia da recuperación de documentos, a *Relevance* evalúa directamente se o contido xerado responde de maneira útil e adecuada á intención da consulta orixinal. Maximizar esta métrica prioriza que o sistema produza respostas substanciais e directamente aplicables, por riba doutras consideracións como a exhaustivididade bruta (*Recall*) ou a minimización de información irrelevante (*Precision*) no contexto recuperado. Esta aproximación está avalada por investigacións recentes que suxiren que, en sistemas RAG interactivos, a relevancia da resposta xerada é o predictor máis forte da satisfacción do usuario [21].

Polo tanto, tras facer un *surface plot* (Figura B.5) da *Relevance* (e tamén da *Faithfulness*), optouse polo par de valores no seu cumio (que o maximizan), sendo este un peso de 1.6 para BM25 e un *chunk size* de 2048 *tokens*. Tamén se realizou un *heatmap* de todas estas métricas promediadas (Figura B.3), onde o par de valores con mellor puntuación segue sendo o mesmo.

Non é de estrañar que o *chunk size* óptimo para este dominio sexa relativamente grande. A documentación burocrática e administrativa universitaria adoita estar estruturada en seccións longas e autocontidas (como procedementos completos, regulamentos ou guías), onde o contexto local é crucial para comprender requisitos, excepcións ou pasos

interrelacionados. Fragmentos demasiado pequenos poden separar información conceptualmente unida, dificultando que o modelo de linguaxe xere respuestas coherentes e exhaustivas. Este resultado é consistente coa literatura sobre recuperación en dominios técnicos e legais, onde se recomiendan tamaños de fragmento maiores para preservar a integridade semántica dos documentos [22].

4.1.6 Consideracións sobre o corpus reducido

O estudo realizouse sobre un **corpus reducido e controlado de documentos**. Isto permite illar variables e analizar o impacto de parámetros como: as estratexias de recuperación (TF-IDF, híbridas, reranking) ou o tamaño de chunk e superposición.

En escenarios de producción, con moitos más documentos dispoñibles, é altamente probable que o sistema recupere unha maior cantidade de documentos irrelevantes, o que afectará especialmente á **precision@10**. O estudo realizado neste corpus reducido proporciona unha visión inicial sobre a capacidade do RAG para filtrar información relevante en presenza de ruído documental, e serve como base experimental para estudar o impacto relativo das decisións de deseño.

4.1.7 Limitacións

O conxunto reducido introduce certas limitacións que afectan a toda a validación:

- Non reflicte a heteroxeneidade nin a escala real do corpus.
- Subrepresenta casos que requieren integración multi-documento.
- Posibles omisións no etiquetado humano.
- As métricas non son directamente extrapolables a producción.

A avaliación proposta proporciona unha visión relativa do impacto das decisións de deseño do RAG, e non debe interpretarse como unha medida directa do seu rendemento absoluto en escenarios de producción [17].

5. Conclusions e traballos futuros

5.1 Extensíons arquitectónicas e metodolóxicas

Ademais da ampliación das dimensíons de avaliación, existen varias liñas de mellora de arquitectura que poderían reforzar o rendemento e a robustez do sistema RAG analizado neste traballo. En particular, a literatura recente sinala que decisións relativas á segmentación do texto, ao reranking dos documentos recuperados e á selección dos conxuntos de avaliación teñen un impacto significativo na calidade final das respostas xeradas.

Unha primeira extensión relevante sería a adopción dunha estratexia de *small-to-big chunking*. Este enfoque consiste en empregar fragmentos de tamaño reducido durante a fase de recuperación, co obxectivo de maximizar a precisión e a discriminación semántica do retriever, e posteriormente expandir eses fragmentos a unidades de contexto más amplas na fase de xeración. Deste modo, o modelo xerador dispón dun contexto más rico e cohesionado, preservando ao mesmo tempo a relevancia local identificada na recuperación inicial. Estudos recentes sinalan que este tipo de estratexias axudan a mitigar a perda de contexto e melloran a coherencia factual das respostas, especialmente en tarefas de preguntas complexas [22], [23].

Outra liña de traballo futuro consiste na incorporación dunha fase explícita de *reranking* baseada en modelos neuronais especializados, como TILDEv2. Este tipo de modelos combina sinais léxicos e semánticos para refinar a orde dos documentos recuperados, permitindo unha selección más precisa do contexto más informativo antes da xeración. A integración dun reranker deste tipo resulta especialmente relevante en escenarios con vocabulario técnico ou especializado, nos que a recuperación puramente semántica pode resultar insuficiente [20]. Ademais, análises empíricas recentes mostran que a inclusión dunha etapa de reranking pode producir melloras consistentes en métricas de recuperación e na calidade percibida das respostas finais, mesmo cando se empregan retrievers competitivos de base [23]. A avaliación comparativa entre unha pipeline sen reranking e outra que incorpore TILDEv2 permitiría cuantificar o impacto desta capa adicional sobre diferentes métricas.

Finalmente, no ámbito da avaliación, unha extensión natural deste traballo sería o uso de datasets estandarizados de *benchmarking*, en particular aqueles orientados a tarefas de *fact checking* e *reasoning*. Este tipo de conxuntos de datos permiten avaliar non só a corrección factual das respostas, senón tamén a capacidade do sistema para razonar sobre evidencias múltiples e detectar inconsistencias ou información incorrecta nos documentos recuperados. A súa incorporación facilitaría a comparación directa con outros sistemas RAG descritos na literatura e permitiría situar os resultados obtidos nun contexto más amplio e representativo do estado da arte, seguindo recomendacións metodolóxicas recentes para a avaliación sistemática de arquitecturas RAG [17], [23].

En conxunto, estas liñas de traballo futuro apuntan cara a unha evolución do sistema avaliado tanto a nivel arquitectónico como metodolóxico, reforzando a súa robustez, capacidade de xeneralización e adecuación a escenarios reais de uso. A súa exploración constitúe unha continuidade natural deste estudio e abre a porta a avaliaciós más completas e comparables cos sistemas RAG actuais descritos na literatura científica [15], [19].

5.2 Extensíóns da validación

A avaliación presentada neste traballo céntrase en métricas clásicas de recuperación e en criterios básicos de calidade da xeración, o cal resulta adecuado para un estudio controlado cun conxunto de validación construído manualmente. Esta elección responde a unha decisión metodolóxica orientada a garantir reproducibilidade, trazabilidade e consistencia na avaliación, máis que a unha limitación conceptual do enfoque.

A literatura recente sinala que o comportamento dun sistema RAG en escenarios reais depende tamén dun conxunto de capacidades más avanzadas (*required abilities*), cuxa avaliación resulta máis complexa e require condicións experimentais específicas, como datasets deseñados ad hoc e un maior volume de anotación humana [17].

Como liña de traballo futuro, resulta recomendable ampliar a validación cara á avaliación destas capacidades, entre as que se inclúen:

- **Noise robustness:** capacidade de xestionar documentos ruidosos semanticamente relacionados coa pregunta pero sen información útil para a resposta.
- **Negative rejection:** habilidade do sistema para recoñecer contextos insuficientes e evitar a xeración de respostas especulativas.
- **Information integration:** capacidade para combinar información procedente de múltiples documentos relevantes en preguntas complexas.
- **Counterfactual robustness:** aptitude para detectar e ignorar información incorrecta ou contraditoria presente nos documentos recuperados.

A incorporación destas dimensíóns permitiría unha avaliación más completa do sistema RAG, pero implicaría a ampliación do conxunto de validación actual ou a creación de novos datasets específicos, así como un maior investimento en anotación humana e deseño experimental. Estas extensiós constitúen, por tanto, unha continuidade natural deste traballo, orientada a aproximar a avaliación ás condicións reais de uso e a analizar o comportamento do sistema en escenarios más complexos e variados [19].

Bibliografía

- [1] F. Pezzuti, S. MacAvaney e N. Tonellotto, "Neural Prioritisation for Web Crawling," en *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, ser. ICTIR '25, Collection: ICTIR '25, ACM, xul. de 2025, pp. 307–314. DOI: [10.1145/3731120.3744597](https://doi.org/10.1145/3731120.3744597) URL: <http://dx.doi.org/10.1145/3731120.3744597>
- [2] S. Chakrabarti, M. Van den Berg e B. Dom, "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks*, vol. 31, n.º 11-16, pp. 1623–1640, 1999. DOI: [10.1016/S1389-1286\(99\)00052-3](https://doi.org/10.1016/S1389-1286(99)00052-3)
- [3] T. T. O. D. Team, *Tesseract OCR Engine*, <https://github.com/tesseract-ocr/tesseract>, versión 5.5.1, Licensed under Apache License 2.0. Accessed: 2025-12-19, 2025.
- [4] Wikipedia. "Ley de Zipf," Wikipedia, La enciclopedia libre, accedido en 15 de dec. de 2025. URL: https://es.wikipedia.org/wiki/Ley_de_Zipf
- [5] Anthropic, *Claude for Multi-Agent Systems: Technical Report on Coordination and Communication Capabilities*, <https://www.anthropic.com/research/clause-multi-agent-systems>, Accessed: 2024-11-30, 2024.
- [6] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [7] J. Johnson, M. Douze e H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, n.º 3, pp. 535–547, 2019.
- [8] M. Douze et al., *The Faiss library*, <https://github.com/facebookresearch/faiss>, Accessed: 2024-12-15, 2024.
- [9] A. Author e B. Author, "A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions," *arXiv preprint*, vol. arXiv:2507.18910, xul. de 2025, Consultado: Diciembre 2025.
- [10] J. Carbonell e J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," en *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '98, New York, NY, USA: ACM, 1998, pp. 335–336.
- [11] M. Grofsky, "Solving Freshness in RAG: A Simple Recency Prior and the Limits of Heuristic Trend Detection," *arXiv preprint*, vol. arXiv:2509.19376, set. de 2025, Consultado: Diciembre 2025.
- [12] K. Spärck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation*, vol. 28, n.º 1, pp. 11–21, 1972. DOI: [10.1108/eb026526](https://doi.org/10.1108/eb026526)

- [13] A. Author et al., "A Hybrid Approach to Information Retrieval and Answer Generation for Regulatory Texts," *arXiv preprint*, vol. arXiv:2502.16767, feb. de 2025, Consultado: Diciembre 2025.
- [14] A. Sharma et al., "Domain-specific Question Answering with Hybrid Search," *arXiv preprint*, vol. arXiv:2412.03736, dec. de 2024, Consultado: Diciembre 2025.
- [15] V. Lakshmanan, "RAG Architecture Design Patterns," *arXiv preprint arXiv:2403.03187*, 2024.
- [16] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," en *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023, pp. –.
- [17] "LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods," 2024. arXiv: [2412.05579](https://arxiv.org/abs/2412.05579).
- [18] "LLM-based NLG Evaluation: Current Status and Challenges," 2024. arXiv: [2402.01383](https://arxiv.org/abs/2402.01383).
- [19] "A Survey on Retrieval-Augmented Generation," 2023. arXiv: [2312.10997](https://arxiv.org/abs/2312.10997).
- [20] J. Lin e X. Ma, "In Defense of Lexical Retrieval: How Specialized Vocabularies Challenge Semantic Search," en *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1926–1930.
- [21] F. Saad et al., "Evaluating the User Perception of Retrieval-Augmented Generation Systems," en *Proceedings of the 2024 Conference on Human Factors in Computing Systems*, 2024, pp. 1–15.
- [22] Y. Wang et al., "Chunking Strategies for Retrieval-Augmented Generation in Legal and Administrative Domains," en *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 10 587–10 602.
- [23] X. Wang et al., "Searching for Best Practices in Retrieval-Augmented Generation," en *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal e Y.-N. Chen, eds., Miami, Florida, USA: Association for Computational Linguistics, nov. de 2024, pp. 17716–17736. DOI: [10.18653/v1/2024.emnlp-main.981](https://doi.org/10.18653/v1/2024.emnlp-main.981) URL: <https://aclanthology.org/2024.emnlp-main.981>

A. Más información acerca do *crawler*

A.1 *Keywords* utilizadas polo *crawler*

A continuación móstrase o conxunto completo de palabras clave utilizadas polo *crawler* para determinar a relevancia das páxinas:

- | | | |
|------------------|-----------------|-----------------|
| ■ regulation | ■ calendario | ■ resolución |
| ■ reglamento | ■ syllabus | ■ circular |
| ■ normativa | ■ programa | ■ instruccións |
| ■ procedure | ■ requirements | ■ instrucciones |
| ■ procedimiento | ■ requisitos | ■ bases |
| ■ proceso | ■ regulamento | ■ anexo |
| ■ form | ■ regulación | ■ catalog |
| ■ formulario | ■ procedemento | ■ catalogo |
| ■ solicitud | ■ solicitude | ■ catálogo |
| ■ guideline | ■ guía | ■ library |
| ■ guia | ■ docente | ■ biblioteca |
| ■ manual | ■ asignatura | ■ collection |
| ■ policy | ■ política | ■ colección |
| ■ politica | ■ matrícula | ■ colección |
| ■ norma | ■ inscripción | ■ acquisition |
| ■ enrollment | ■ académico | ■ adquisicion |
| ■ matricula | ■ convocatoria | ■ adquisición |
| ■ inscripcion | ■ prazo | ■ loan |
| ■ administrative | ■ prazos | ■ prestamo |
| ■ administrativo | ■ documentación | ■ préstamo |
| ■ academic | ■ tramite | ■ reserve |
| ■ academico | ■ trámite | ■ reserva |
| ■ calendar | ■ ordenanza | ■ interlibrary |

- interbibliotecario ■ bibliographic ■ autorizacion
- reference ■ bibliografico ■ autorización
- referencia ■ bibliográfico ■ notification
- circulation ■ holdings ■ notificacion
- circulacion ■ fondos ■ notificación
- circulación ■ serials ■ registration
- periodical ■ publicacions seriadas ■ registro
- periodico ■ special collections ■ protocol
- periódico ■ coleccions especiais ■ protocolo
- journal ■ reading room ■ statute
- revista ■ sala de lectura ■ estatuto
- archive ■ stacks ■ ordinance
- archivo ■ depósito ■ decree
- arquivos ■ microfilm ■ decreto
- repository ■ microficha ■ resolution
- repositorio ■ digital library ■ resolucion
- classification ■ biblioteca dixital ■ official
- clasificacion ■ opac ■ oficial
- clasificación ■ marc ■ office
- indexing ■ application ■ oficina
- indexacion ■ deadline ■ department
- indexación ■ plazo ■ departamento
- cataloging ■ documentation ■ service
- catalogacion ■ documentacion ■ servicio
- catalogación ■ certification ■ servizo
- dewey ■ certificado ■ unit
- isbn ■ certificación ■ unidad
- issn ■ authorization ■ unidade

A.2 Saída OCR da táboa de exemplo

Texto OCR

[Tipos de usuarios
Nº de documentos en préstamo
Días de préstamo
Renovaciones
Reservas
GRUPO 1
Estudiantado de Grado de centros propios y adscritos
Estudiantado de programa de movilidad (Erasmus, Sicue-Séneca)
Estudiantado de doble Grado, de simultaneidad 10 10 días Indefinidas | Límite 10 docs.
Estudiantado de grados interuniversitarios Estudiantado da Universidad Séntor GRUPO 2
Estudiantado de MASTERES e posgrados propios, incluyendo los dela Fundación Universidade da Coruña . , 15 21 días Indefinidas | Límite 15 docs. Estudiantado de Trabajos de fin de grado. Personal de administración en servicio (PTGAS) GRUPO 3 Estudiantado de Doctorado 35 30 días Indefinidas | Límite 35 does. Personal investigador visitante; visitant tant 'ersonal investigador visitante: visitante senior] visitante 35 30 días indefinidas | Límite 38 docs predoctoral o postdactoral GRUPO 4 Becarios /as de investigación Curso académico Personal contratado investigador (cada biblioteca podrá excluir 35 de este tipo de préstamo Indefinidas | Límite 35 docs. documentos par razón de uso y disponibilidad) GRUPO 5 PDI da UDC (incluyendo centros adscritos), de la Fundación UDC Curso académico Profesorado emérito, jubilado incentivado y honorario (cada biblioteca podrá excluir Profesorado visitante 100 de este tipo de préstamo Indefiridas | Límite 100 docs. Lectores /as documentos par razón de uso y disponibilidad) GRUPO 6 Cual ! idad taria de la UDC cualquier persona ajena a la comunidad universitaria dela 6 10 días indefinidas | Límite 6 docs que sea autorizada por la Biblioteca Universitaria

A.3 Progresión de páxinas *retrieveadas* ao longo do crawleo

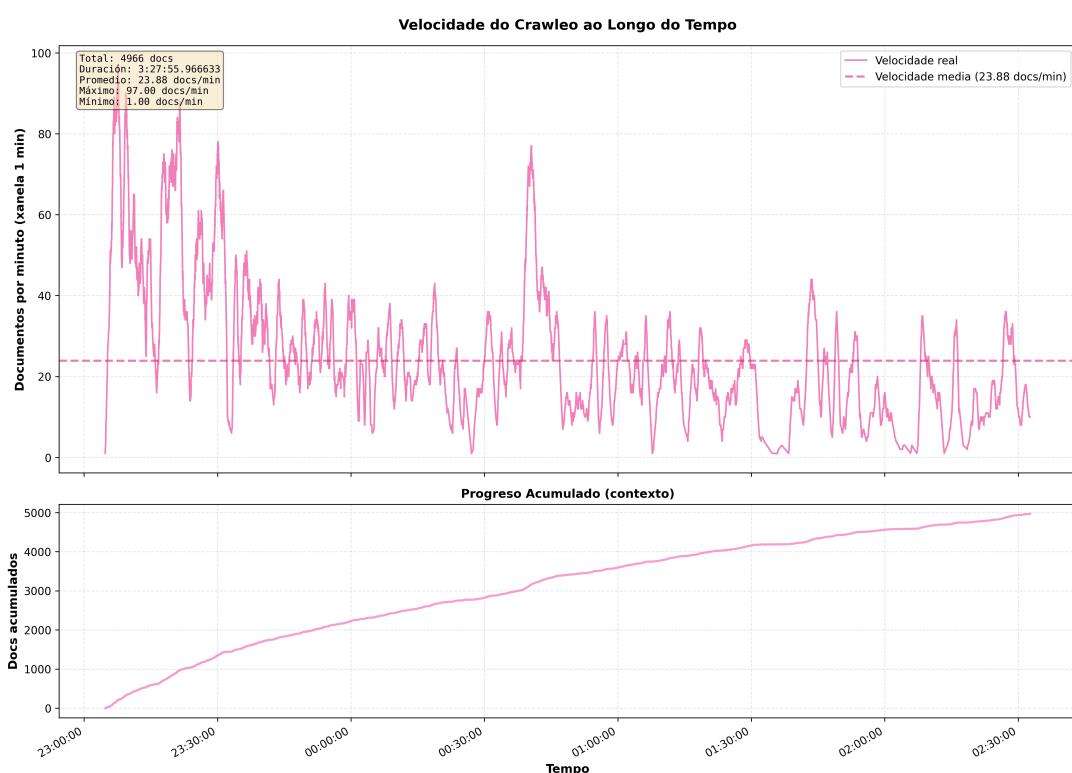


Figura A.1 Progresión da eficiencia do crawler ao longo do tempo

B. Análise do Corpus

B.1 Distribución de tipos de ficheiros

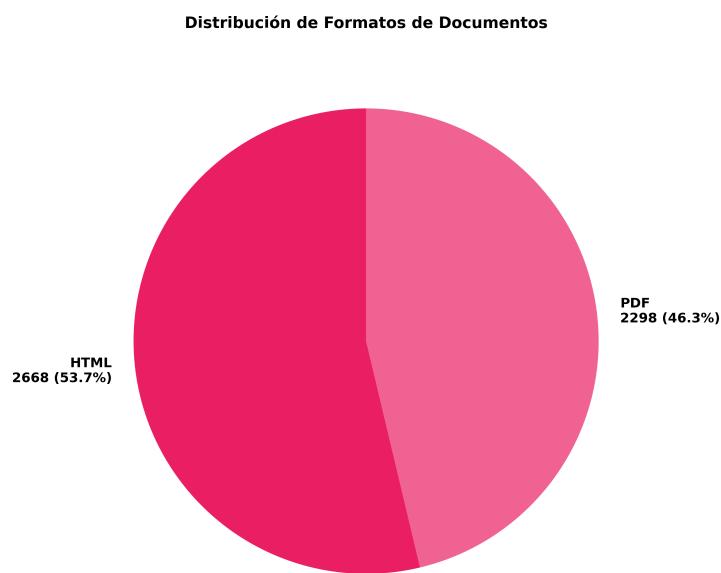


Figura B.1 Distribución de tipos de ficheiros no corpus

B.2 Palabras más frecuentes

Pos.	Palabra	Frecuencia
1	de	385,703
2	a	113,823
3	en	102,598
4	la	102,014
5	e	90,196
6	y	77,587
7	que	73,069
8	o	57,323
9	da	49,975
10	el	48,951
11	para	43,204
12	do	40,254
13	se	35,387
14	los	35,200
15	del	33,793
16	las	29,507
17	por	27,315
18	no	25,683
19	con	25,326
20	udc	25,182

Cadro B.1Top 20 palabras más frecuentes no corpus

B.3 Palabras menos frecuentes

Pos.	Palabra	Frecuencia
1	ricondo	- 1
2	acompañamiento	- 1
3	recomendaciéns	- 1
4	aproximacién	- 1
5	nosum	- 1
6	climent	- 1
7	vengut	- 1
8	empar	- 1
9	retransmision	- 1
10	toxicoloxia	- 1
11	landeira	- 1
12	angelines	- 1
13	psicoloxicos	- 1
14	mase	- 1
15	lameiras	- 1
16	cartelixornadasumisionquimicaevs	- 1
17	conciliacíons	- 1
18	conciliaciones	- 1
19	operatoria	- 1
20	extranjeos	- 1

Cadro B.2Top 20 palabras menos frecuentes no corpus

B.4 Resultados validación RAGsystem

B.4.1 Heatmaps de varias métricas

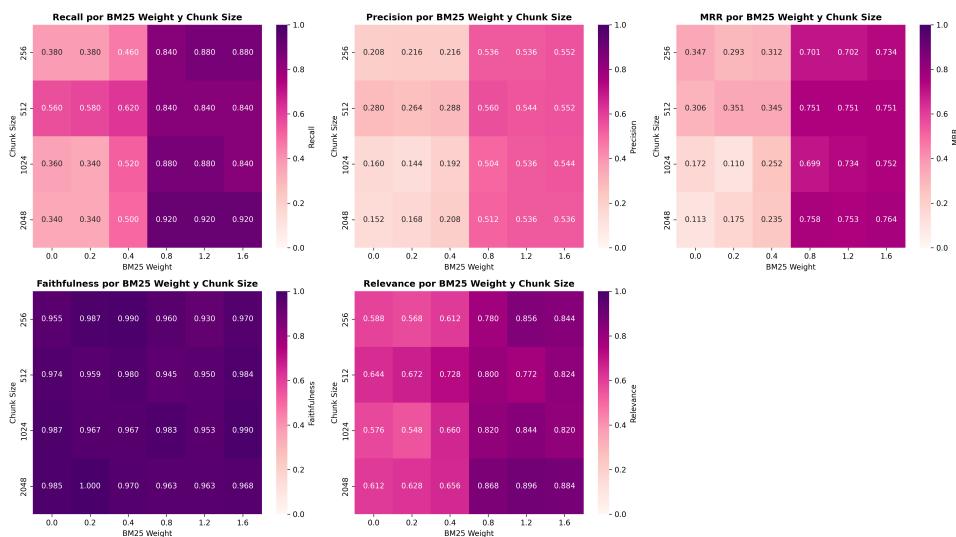


Figura B.2Heatmaps das métricas de validación

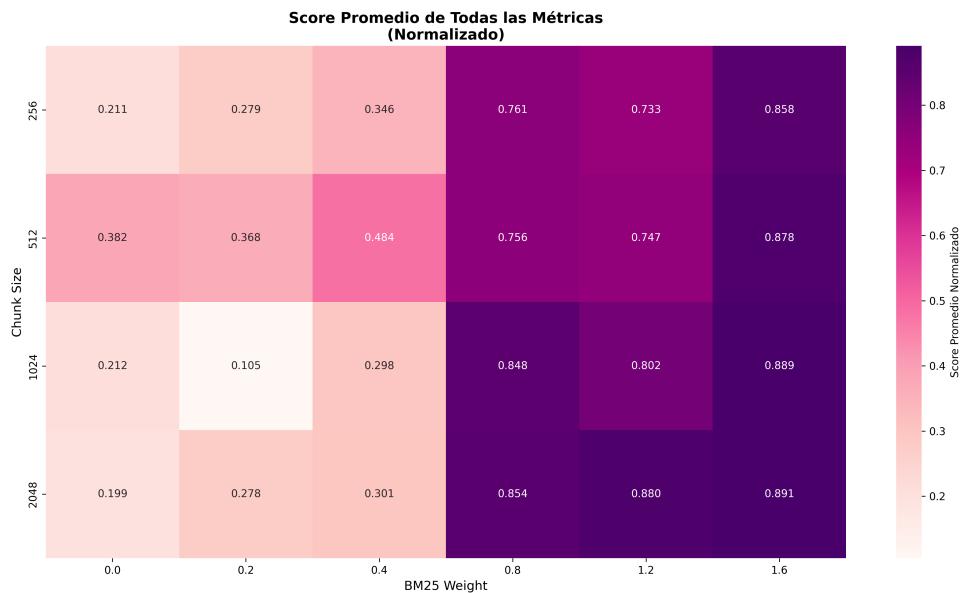


Figura B.3 Heatmap combinado de todas las métricas normalizadas

B.4.2 Surface plots de varias métricas

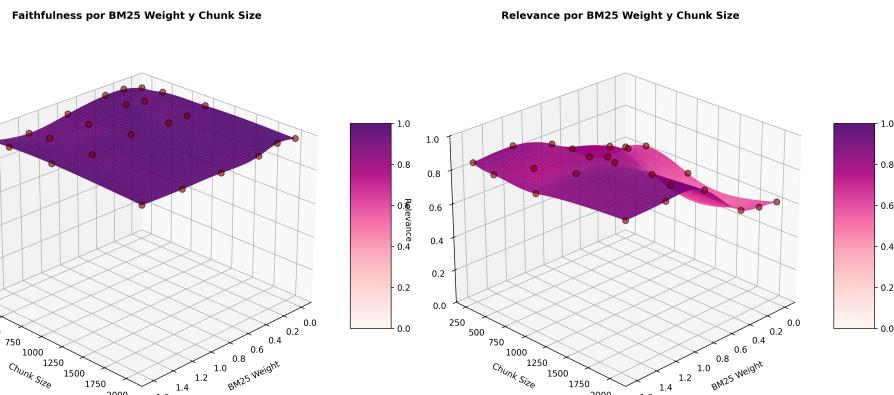


Figura B.4 Surface plots de Faithfulness y Relevance

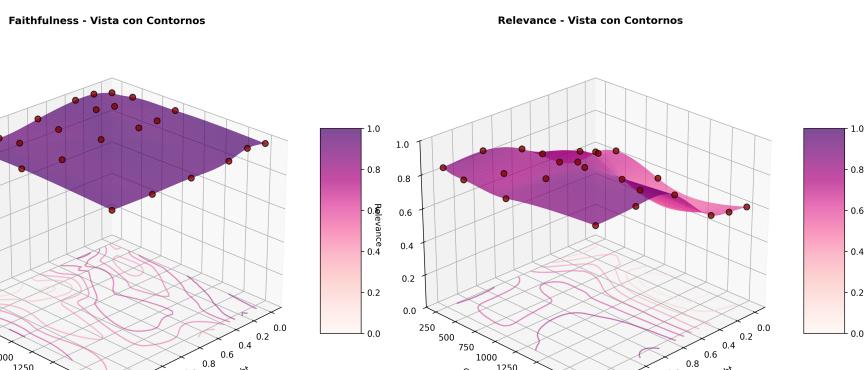


Figura B.5 Surface plots con contornos de Faithfulness y Relevance

B.4.3 Comparación de métricas para diferentes valores de peso de BM25 co mellor *chunk size* (2048)

Cadro B.3 Faithfulness según BM25 Weight (Chunk Size = 2048)

BM25 Weight	Faithfulness
0.0	0.985
0.2	1.000
0.4	0.970
0.8	0.963
1.2	0.963
1.6	0.968

Cadro B.4 Relevance según BM25 Weight (Chunk Size = 2048)

BM25 Weight	Relevance
0.0	0.612
0.2	0.628
0.4	0.656
0.8	0.868
1.2	0.896
1.6	0.884