

Mòdul de Chatbots per IA Tech

Daniel García Nilo

Resum - Aquest projecte té com a objectiu principal dissenyar i analitzar un mòdul escalable per a la gestió d'agents conversacionals dins l'entorn corporatiu d'IA Tech. A partir d'un enfocament basat en agents, s'ha desenvolupat una proposta funcional i tècnica capaç de gestionar múltiples fluxos conversacionals de manera flexible i autònoma. L'anàlisi inicial va partir d'un model altament estructurat, compost per múltiples entitats compartides, però durant el procés es va optar per una reformulació arquitectònica centrada en la parametrització individualitzada de cada agent.

El nou model permet encapsular tota la lògica dins d'un únic registre en base de dades, utilitzant estructures clau-valor per definir estats, transicions i accions. Aquesta simplificació facilita el manteniment, millora l'escalabilitat i permet una millor integració amb tecnologies d'intel·ligència artificial generativa.

Per validar aquest enfocament, s'ha implementat un primer Proof of Concept orientat a la gestió d'incidències, confirmant la viabilitat del disseny i la seva capacitat per gestionar fluxos de manera eficient i configurable.

Paraules clau - Agents conversacionals, fluxos configurables, intel·ligència artificial generativa, arquitectura modular, parametrització, clau-valor, escalabilitat, manteniment, Proof of Concept, gestió d'incidències.

Abstract - The main objective of this project is to design and analyze a scalable module for managing conversational agents within IA Tech's corporate environment. Based on an agent-oriented approach, a functional and technical proposal has been developed to manage multiple conversational flows in a flexible and autonomous way. The initial analysis began with a highly structured model composed of multiple shared entities, but during the process, an architectural redesign was adopted, centered on the individualized parameterization of each agent.

The new model allows all the logic to be encapsulated within a single database record, using key-value structures to define states, transitions, and actions. This simplification facilitates maintenance, improves scalability, and enables better integration with generative artificial intelligence technologies.

To validate this approach, an initial Proof of Concept focused on incident management has been implemented, confirming the design's viability and its ability to handle flows efficiently and in a configurable manner.

Index Terms - Conversational agents, configurable flows, generative artificial intelligence, modular architecture, parameterization, key-value, scalability, maintenance, Proof of Concept, incident management.



1 INTRODUCCIÓ – CONTEXT DEL TREBALL

1.1 IDENTIFICACIÓ DEL PROBLEMA

En un entorn empresarial cada vegada més digital i interconnectat, la gestió eficient de la informació i l'automatització de processos esdevenen factors clau per a la competitivitat i sostenibilitat de les organitzacions. El volum creixent de dades i documentació, sumat a la necessitat d'oferir respostes ràpides i contextualitzades, genera una càrrega operativa significativa en tasques de suport, atenció al client o recuperació d'informació.

Aquest projecte sorgeix com a resposta a una problemàtica recurrent dins l'àmbit corporatiu: la dificultat per gestionar múltiples processos conversacionals de manera coherent, flexible i escalable. Sovint, les empreses disposen de canals d'atenció distribuïts, eines aïllades o sistemes manuals que no permeten una gestió eficient del coneixement intern. Aquesta realitat fa evident la necessitat d'un sistema que centralitzi la lògica de decisió, permeti personalitzar fluxos i aprofiti les capacitats de la intel·ligència artificial per millorar l'experiència d'usuari i optimitzar els recursos disponibles.

- *E-mail de contacte:* 1633613@uab.cat
- *Menció realitzada:* Enginyeria del Software
- *Treball tutoritzat per:* Joan Protasio Ramírez (Departament de Ciències de la Computacion)
- *Curs* 2024/25

Dins d'aquest context, el departament de Transformació Digital de l'organització ha impulsat el desenvolupament d'un mòdul genèric de gestió d'agents conversacionals. Aquesta iniciativa té com a objectiu establir una base funcional sòlida per dissenyar i desplegar chatbots especialitzats, capaços d'automatitzar processos, interactuar amb usuaris, i gestionar fluxos conversacionals amb autonomia i precisió.

La proposta s'integra dins de l'ecosistema IA Tech, una plataforma d'automatització corporativa que utilitza un bus d'esdeveniments (Kafka) i un conjunt d'accions backend per gestionar sol·licituds en temps real. El desenvolupament d'un mòdul de chatbots dins d'aquest entorn permetrà escalar aquestes capacitats cap a la interacció directa amb l'usuari i l'automatització de consultes complexes.

Aquest projecte es planteja com una primera fase d'anàlisi i disseny funcional del sistema, amb l'objectiu d'assentar una arquitectura flexible basada en agents autònoms, configurables i reutilitzables. Per validar aquest enfocament, s'ha desenvolupat un primer prototip funcional centrat en la gestió d'incidències informàtiques, que ha servit com a prova inicial del model dissenyat.

1.2 ESTAT DE L'ART

L'anàlisi de l'estat de l'art permet identificar les solucions actuals per a la gestió d'informació i l'automatització de processos, així com les limitacions que aquest projecte pretén resoldre.

Solucions existents:

- Sistemes de gestió documental (SharePoint, Alfresco, OpenText): centralitzen la informació, però depenen de classificació manual i no proporcionen respostes contextuals immediates.

- Plataformes de desenvolupament de chatbots (Dialogflow, Azure Bot Framework): ofereixen funcionalitats avançades i integracions, però estan fortament vinculades a ecosistemes tancats, limitant la personalització i augmentant la dependència tecnològica.

- Eines d'automatització de processos (UiPath, Automation Anywhere): permeten automatitzar tasques repetitives, però no estan orientades a la gestió intel·ligent del diàleg ni a la interpretació de context.

Aportació diferencial del projecte:

Aquest projecte proposa una arquitectura pròpia, modular i independent, que no depèn de plataformes externes. A diferència de les solucions tancades, permet adaptar completament el comportament dels agents a les

necessitats específiques de l'empresa, mantenint el control total sobre la lògica, les dades i el desplegament.

El sistema introdueix un canvi de paradigma, on cada agent es configura com una unitat autònoma mitjançant parells clau-valor, amb definició pròpia de fluxos, estats i respostes. Aquesta estructura afavoreix l'adaptació a entorns corporatius diversos, la flexibilitat en l'evolució i l'escalabilitat funcional.

2 OBJECTIUS

Aquest projecte s'ha centrat en dissenyar una arquitectura versàtil per a agents conversacionals dins de l'entorn corporatiu d'IA Tech. Més enllà de resoldre un cas d'ús concret, s'ha buscat construir una base comuna sobre la qual es puguin desplegar múltiples agents amb lògiques pròpies, però amb una estructura coherent i fàcilment gestionable.

Durant les primeres fases del projecte, l'anàlisi funcional va evidenciar que una aproximació basada en agents independents generava una complexitat excessiva. Això va motivar una reorientació estratègica cap a un model base configurable, que ha demostrat ser més adequat tant a nivell tècnic com operatiu.

La viabilitat d'aquest model es va contrastar mitjançant un prototip funcional aplicat a un cas real. Aquesta prova ha servit per validar no només el funcionament tècnic de l'arquitectura, sinó també l'encert de les decisions preses durant el procés d'anàlisi.

Aquesta evolució metodològica ha consolidat una base robusta per al desenvolupament futur d'agents conversacionals, amb una orientació clara cap a la reutilització, la flexibilitat i la sostenibilitat a llarg termini.

Aspecte	Descripció
Objectiu del projecte	Desenvolupar una arquitectura general per a múltiples agents reutilitzables
Enfocament inicial	Desenvolupament de chatbots independents per àrees funcionals
Problema detectat	Excés de complexitat i càrrega de manteniment elevada
Solució adoptada	Disseny d'un model base parametrizable per a agents autònoms
Avantatges del nou model	Flexibilitat, escalabilitat, mantenibilitat i coherència estructural
Validació del model	Proof of Concept aplicat a la gestió d'incidències

Aspecte	Descripció
Resultat del procés	Model sòlid i adaptat a les necessitats reals per a futurs desplegaments

3 METODOLOGIA

El desenvolupament d'aquest projecte s'ha basat en la metodologia ScrumBan, una estratègia híbrida que combina les millors pràctiques de dos enfocaments àgils àmpliament reconeguts: Scrum i Kanban. Aquesta elecció metodològica ha estat especialment adequada per a un projecte amb una forta càrrega d'anàlisi conceptual, disseny arquitectònic i desenvolupament exploratori, on la capacitat d'adaptació i replanificació contínua ha estat clau per assolir els objectius plantejats.

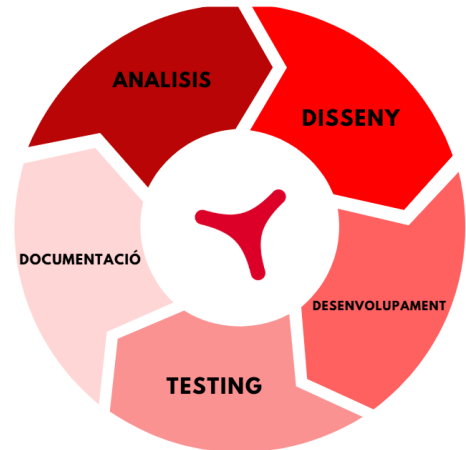
La part estructurada de Scrum ha proporcionat una base sòlida per organitzar el projecte en sprints quinzenals (25.04, 25.05, 25.06...), cadascun dels quals ha incorporat activitats específiques d'anàlisi funcional, disseny de fluxos, desenvolupament, proves i documentació. A l'inici de cada sprint s'han definit objectius concrets i, al final, s'ha realitzat una revisió per avaluar el grau d'assoliment i identificar possibles millores, mitjançant les clàssiques cerimònies de Sprint Review i Retrospective. Això ha permès mantenir un ritme iteratiu estable i una avaluació constant del progrés.

D'altra banda, la flexibilitat pròpia del model Kanban ha facilitat una gestió més fluida i visual del flux de treball diari. Mitjançant un tauler digital amb columnes com "Oberta", "Preparada", "En Progrés", "Pendent" i "Tancada", s'ha pogut visualitzar l'estat de cada tasca i prioritzar-les segons la seva urgència o dependències tècniques. Aquest sistema ha resultat especialment útil per gestionar tasques contínues com la revisió de codi, la integració amb serveis externs com JIRA i l'Anàlisi de Negoci, o l'ajust progressiu del model base.

La implementació de ScrumBan també ha permès incorporar canvis metodològics derivats de les necessitats detectades durant el desenvolupament. Per exemple, a partir del Sprint 25.06 es van identificar nous requisits funcionals com la necessitat d'un control més robust del context conversacional o una millor traçabilitat de les interaccions. Gràcies a l'enfocament flexible, aquests canvis es van poder incorporar sense interrompre el ritme general del projecte ni comprometre l'estabilitat de les funcionalitats ja implementades.

A cada sprint s'ha seguit un cicle complet de desenvolupament àgil, amb les següents fases:

- Anàlisi
- Disseny
- Desenvolupament incremental
- Testing integrat dins de l'sprint:
- Documentació iterativa



[Figura 1: Metodologies del desenvolupament]

La gestió del projecte s'ha reforçat amb l'ús de gràfiques de seguiment, com la burndown chart global, que ha permès mesurar la progressió acumulada del treball pendent. Aquesta gràfica reflecteix un descens constant, tot i que lleugerament inferior al ritme ideal previst, degut en gran part a la complexitat inicial de l'anàlisi arquitectònic. Malgrat aquesta desviació, la reorganització interna ha permès afegir un sprint addicional (25.09), amb l'objectiu de tancar la fase de test i completar el desenvolupament funcional del PoC sense comprometre la qualitat ni els objectius marcats.

Finalment, cal destacar que la metodologia ScrumBan ha afavorit una visió global del projecte, mantenint l'equilibri entre estructura i flexibilitat, i ha resultat determinant per afrontar amb èxit els reptes tècnics i organitzatius que han anat sorgint. La seva aplicació ha garantit una evolució coherent i sostenible del projecte, tant en la seva vessant analítica com en la implementació pràctica, establint les bases per a la continuïtat futura del sistema d'agents conversacionals.

4 PLANIFICACIÓ

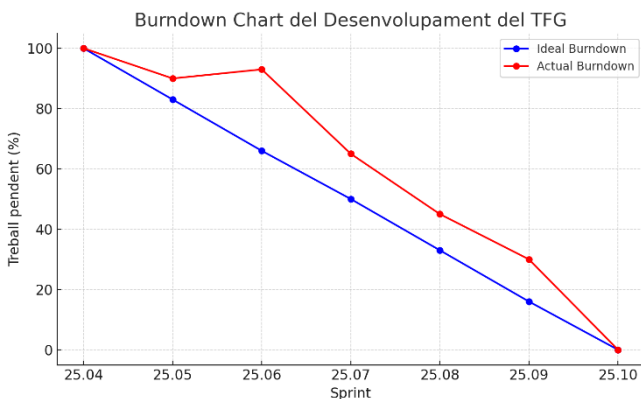
La planificació del projecte s'ha concebut des del principi com un procés iteratiu i flexible, capaç d'adaptar-se als aprenentatges i a les necessitats emergents que han anat apareixent durant el desenvolupament. Tot i que inicialment es va definir una estructura clàssica en fases (anàlisi, disseny, desenvolupament i validació), la naturalesa exploratòria del projecte, centrada en la definició d'una arquitectura per a agents conversacionals, ha exigut revisions estratègiques tant en l'ordre com en el pes relatiu de cada etapa.

Inicialment es contemplava treballar en paral·lel en dues línies: l'anàlisi funcional dels primers agents previstos per IA Tech, i el desenvolupament simultani d'un primer Proof of Concept (PoC). Però ja en les primeres setmanes es va decidir prioritzar una fase d'anàlisi en profunditat, orientada a construir un model base reutilitzable que permetés crear futurs agents amb un cost tècnic mínim. Aquesta decisió va comportar una reorganització substancial respecte a la planificació original, però sense alterar els objectius globals, sinó adaptant-ne l'execució segons el coneixement adquirit.

Aquesta reestructuració va derivar en tres etapes diferenciades:

- Una primera fase d'anàlisi conceptual i arquitectònic, centrada en definir el model de dades i el comportament per estats.
- Una segona fase de validació mitjançant un PoC funcional, per comprovar la viabilitat del model proposat.
- I finalment, una fase de reestructuració tècnica i documental, per adaptar el sistema complet al nou model per agents independents.

Aquest enfocament ha resultat més òptim que el previst inicialment, ja que ha permès validar la solidesa de la proposta abans d'entrar en etapes més costoses de desenvolupament, evitant futures reestructuracions i assegurant la coherència global del sistema.



Cal destacar també que, durant el procés, s'han afegit nous requisits i funcionalitats no previstos en la planificació inicial. Això es reflecteix clarament en el pic visible a la burndown chart, que indica un increment temporal de la càrrega de treball. Tanmateix, aquest augment no s'ha interpretat com una desviació negativa, sinó com un reflex directe de l'evolució natural del projecte, que ha sabut incorporar nous elements de valor i adaptar-ne l'estratègia per reforçar la qualitat i la sostenibilitat de la solució final.

5 DESENVOLUPAMENT DEL ANÀLISI

5.1 KICK-OFF DEL PROJECTE I ANÀLISI PRELIMINAR

El punt de partida del projecte va consistir en una fase inicial d'alineació estratègica i anàlisi funcional centrada en comprendre l'estat actual del sistema de gestió d'incidències mitjançant chatbot, així com les necessitats reals de l'organització. Aquesta fase de kick-off va incloure la revisió de documentació existent, l'estudi de tecnologies relacionades i diverses reunions amb l'equip d'Analítica de Negoci, actors clau en la implantació de solucions d'IA conversacional dins l'empresa.

En primer lloc, es va dur a terme una revisió exhaustiva de la documentació sobre el chatbot d'incidències ja desplegat, amb l'objectiu de conèixer tant la seva arquitectura actual com els escenaris d'ús contemplats. Aquest procés va incloure l'anàlisi d'informes previs, la identificació de les tecnologies implicades i l'examen de diagrames estructurals. Paral·lelament, es van establir diverses reunions amb el departament d'arquitectura i sistemes, que van permetre aclarir dubtes tècnics i obtenir una visió més detallada sobre el comportament real del sistema i les seves limitacions.

Per afavorir la comprensió global del flux de funcionament del chatbot, es va elaborar un diagrama conceptual que descrivia de forma gràfica totes les etapes del procés conversacional: des de l'activació inicial per part de l'usuari fins a la resolució (o derivació) de la incidència. Aquest diagrama incloïa també els punts de decisió clau i les possibles bifurcacions del flux, servint de base per a futures iteracions i refinaments.

Dins d'aquest marc d'anàlisi, es va dedicar especial atenció a l'estudi del sistema RAG (Retrieval-Augmented Generation), que constitueix un component essencial del model actual de resposta automatitzada. Es va observar com el sistema fragmenta la informació documental en blocs coherents per facilitar la cerca semàntica, i com el model d'IA genera respostes precises mitjançant la construcció de prompts dinàmics adaptats al context de la consulta. Aquest procés es basa en un flux seqüencial que inclou: recepció i preprocesament de la pregunta, filtratge i classificació, recuperació de la informació rellevant, generació i validació de la resposta, i finalment, presentació interactiva a l'usuari.

[Figura 2: Diagrama sistema RAG d'Analítica de negoci]

Finalment, a través de l'anàlisi del model de chatbot utilitzat a Analítica de Negoci —conegut internament com a *Geppeto*—, es van extreure les funcionalitats essencials que un agent conversacional havia d'incorporar per ser efectiu: capacitat de comprensió contextual, gestió de l'ambigüitat, interacció multicanal, i recuperació eficient d'informació estructurada. Aquest coneixement va ser fonamental per orientar el disseny d'un sistema flexible, reutilitzable i

preparat per escalar en entorns empresarials reals.

5.2 ENFOCAMENT INICIAL I PRIMERA APROXIMACIÓ FUNCIONAL

Amb l'objectiu d'abordar el comportament dels agents conversacionals de manera estructurada, el projecte va iniciar-se amb una anàlisi progressiva que partia d'una visió externa del sistema, sovint coneguda com a enfocament de "caixa negra". Aquesta aproximació es basava en observar les converses des del punt de vista de l'usuari, sense entrar inicialment en els mecanismes interns del sistema.

En aquesta fase inicial, la conversa es conceptualitzava com una seqüència iterativa de preguntes i respostes: l'usuari formulava una consulta i el bot generava una resposta. Aquesta representació simplificada, tot i que útil per establir una primera comprensió funcional, resultava insuficient per capturar la complexitat real de les interaccions. No totes les respostes són simples ni totes les consultes són directes; sovint cal interpretar contextos, formular preguntes intermèdies o consultar fonts d'informació addicionals.

Per aquest motiu, es va considerar necessari realitzar un zoom analític cap a l'interior del sistema, per tal d'identificar i entendre els components interns que intervenen en la generació de respostes i en l'evolució del flux conversacional.



[Figura 3: Caixa negra]

Aquest procés d'aprofundiment va permetre definir amb precisió el concepte de "pas" dins del sistema. Cada pas representa un punt específic del flux on el bot executa una acció concreta: mostrar un missatge, formular una pregunta, esperar una resposta, accedir a una font externa (com un sistema RAG o una matriu de categorització), o transferir la conversa a un altre agent especialitzat. Aquests passos, lluny de ser simples blocs de text, inclouen tota la configuració necessària per determinar el comportament del bot i les condicions per avançar cap al següent estat.

A més, aquest enfocament permet que cada interacció pugui adoptar formes diferents segons el context: una pregunta oberta, una selecció múltiple, una resposta generada per IA, o fins i tot un missatge de finalització. Això fa possible una arquitectura conversacional flexible i intel·ligent, capaç d'adaptar-se a escenaris diversos.

El resultat d'aquest zoom analític va ser el reconeixement que la conversa no segueix un flux lineal, sinó que constitueix una xarxa d'interaccions condicionals, amb múltiples camins possibles. Aquesta comprensió ha estat

clau per establir les bases d'una arquitectura modular, on cada pas pot ser dissenyat i reutilitzat de manera independent, i on el flux global pot adaptar-se dinàmicament sense haver de redissenyar el sistema complet.

5.3 EVOLUCIÓ FUNCIONAL I MODEL INTERN DEL SISTEMA CONVERSACIONAL

5.3.1. IDENTIFICACIÓ DE FUNCIONALITATS I COMPORTAMENT INTERN

Després de l'anàlisi inicial des d'una perspectiva externa —on la conversa es representava com una successió lineal de missatges entre l'usuari i el chatbot—, es va iniciar un procés de profundització progressiva que va permetre descobrir la complexitat real del sistema conversacional. Aquest enfocament de *zoom* analític va ser clau per descompondre la interacció en components funcionals més específics i comprensibles, deixant enrere la visió simplificada de consulta-resposta.

Amb l'ampliació del nivell de detall, es va identificar que un agent conversacional no només ha de ser capaç de respondre preguntes, sinó que també ha de poder prendre decisions, gestionar el context, recuperar informació externa, validar respostes i, en alguns casos, executar accions específiques. Aquesta descoberta va marcar un punt d'inflexió en l'anàlisi, permetent passar d'un model rígid i seqüencial a una arquitectura funcional més rica, flexible i modular.

Les funcionalitats essencials que el sistema havia d'incorporar no es van formalitzar inicialment com a mètodes tancats, sinó com a capacitats operatives, orientades a garantir l'adaptabilitat davant diferents escenaris. Algunes d'aquestes capacitats inclouen: decidir quin agent ha de gestionar una consulta, formular preguntes clares en situacions d'ambigüitat, recuperar dades externes rellevants, classificar automàticament les consultes segons la seva naturalesa, i validar la coherència d'una resposta abans d'enviar-la a l'usuari. Aquesta estratègia conceptual es va escollir per evitar imposar estructures rígides en una fase exploratòria i per mantenir la flexibilitat necessària en les etapes següents del disseny.

Com a primer pas per representar el comportament del sistema de manera estructurada, es va proposar un model conceptual conegut com a *Thread View* (model de fil), que observava la conversa com una **seqüència temporal d'interaccions** entre usuari i bot. Aquesta representació permetia visualitzar clarament els torns i servia com a base per identificar patrons i bifurcacions en el flux conversacional.

[Figura 4: Diagrama sistema "Model Fil"]

Tanmateix, aquest model lineal no capturava encara els processos interns que intervenen en la generació de respostes, per la qual cosa es va evolucionar cap a una visió més profunda del funcionament intern.

En aquest nou enfocament, es va conceptualitzar cada interacció com una unitat tancada: comença amb la consulta de l'usuari i finalitza amb la resposta del bot. Aquesta estructura modular permet analitzar i reutilitzar cada interacció de forma independent. A partir d'aquí, es va introduir el concepte del *cervell* del chatbot — un nucli funcional que, tot i no formar part directa del flux de passos visibles, resulta essencial per orquestrar-lo. Aquest component actua com un **sistema de control intel·ligent** que analitza el missatge rebut, interpreta el context i l'historial de la conversa, i selecciona l'estratègia més adequada per generar una resposta.

[Figura 5: Diagrama sistema “cervell”]

Entre les accions que aquest nucli pot dur a terme s'inclouen: generar una resposta mitjançant IA generativa, formular preguntes aclaridores, consultar sistemes externs com bases de dades o matrius de categorització, o bé transferir la conversa a un altre agent especialitzat si escau. A més, aquest cervell no només determina *què* cal fer, sinó també *com* fer-ho, escollint l'eina, el mètode i l'ordre d'execució segons la situació.

Durant el disseny d'aquest model, es va establir també una **jerarquia d'accions** que permet al sistema evitar reintents innecessaris i optimitzar la presa de decisions. Per exemple, si una resposta generada per IA no és satisfactòria, el sistema pot escalar automàticament a una altra font d'informació, sense repetir intents previs. Aquest mecanisme incrementa l'eficiència del flux i evita bucles improductius.

Una altra capacitat rellevant identificada en aquesta fase és la **delegació**: si el sistema detecta que la consulta excedeix el domini del chatbot actual, pot transferir-la de forma intel·ligent a un altre agent especialitzat, obrint una nova sessió en un context diferent. Aquest enfocament introdueix una lògica de col·laboració entre agents i contribueix a l'escalabilitat i especialització de la plataforma.

En conjunt, aquest model basat en interaccions completes i en un nucli de decisió centralitzat representa una evolució significativa en la comprensió del comportament dels chatbots. Permet dissenyar un sistema **modular, flexible i escalable**, on cada component té una funció clara i on el flux s'adapta de forma dinàmica a les necessitats de l'usuari i a la complexitat de la consulta.

5.3.2 PRIMER DISSENY DE LA ESTRUCTURA DE DADES

Un cop establert el comportament intern del sistema i el rol central del nucli de decisió, es va iniciar una fase de formalització del model mitjançant estructures de dades que permetessin representar el flux conversacional del chatbot de manera editable i reutilitzable. Aquesta etapa va resultar clau per validar les idees conceptuals anteriors i assentar les bases d'una futura implementació robusta.

Amb aquest objectiu, es va construir una primera versió del model de configuració emprant un conjunt de taules dissenyades en format Excel. Aquestes taules no només permetien representar els elements del sistema de manera clara i estructurada, sinó que servien com a eina de disseny semiformal i com a prototip per una futura translació a un sistema de base de dades relacional.

La taula de passos fou el punt de partida d'aquesta formalització. Cada fila representava un estat dins del flux conversacional, amb informació detallada sobre la seva funció, el tipus de missatge, si requeria resposta de l'usuari i el mètode de generació de la resposta. Aquest esquema permetia visualitzar el sistema com una màquina d'estats parametrizable, on cada pas tenia un rol específic i ben definit dins la conversa.

[Figura 6: Taula excel “Paso”]

Els passos es van classificar funcionalment en quatre categories:

- Inicialització: punts d'entrada al sistema.
- Decisió: passos interns que determinen el següent estat.
- Interacció amb l'usuari: missatges que esperen resposta.
- Acció final: passos que tanquen la conversa o executen una acció.

Per modelar el moviment entre passos, es va crear la taula de transicions, que defineix els enllaços possibles entre passos d'origen i destinació, indicant el tipus de transició (automàtica, per regla, per interacció, etc.).

[Figura 7: Taula excel “Transiciones”]

Aquesta es completava amb la taula de condicions, que detallava les regles específiques que calia complir per activar una transició concreta. Les condicions podien estar basades en la resposta de l'usuari, en el resultat d'un model d'IA o en regles internes del sistema.

[Figura 8: Taula excel “Condiciones”]

Aquest conjunt de taules va permetre construir una primera versió funcional del model de configuració, facilitant la validació de fluxos, la detecció d'errors lògics i la simulació de comportaments abans de la seva implementació. A més, va assentar la base per a la seva conversió futura a una base de dades relacional, garantint traçabilitat, modularitat i escalabilitat.

Amb la voluntat de donar suport tècnic a aquesta arquitectura conversacional, es va dissenyar una base de dades estructurada en tres nivells: conceptual, lògic i físic. El model conceptual es va representar mitjançant un diagrama Entitat-Relació (E-R), dividit en dues zones principals: configuració i execució.

- A la zona de configuració, s'hi troben les entitats

Chatbot, Pas i PasSiguiende, que defineixen el comportament estàtic del sistema.

- A la zona d'execució, s'hi defineixen Sessió i Conversació, que registren l'activitat dinàmica del sistema en temps real.

[Figura 9: Diagrama E-R]

El model lògic relacional tradueix aquest esquema conceptual en una sèrie de taules concretes:

- Bústia de configuració:
 - chatbot: identificació de cada bot.
 - paso: definició de cada pas amb els seus atributs.
 - paso_siguiende: transicions entre passos i condicions associades.
- Bústia d'execució:
 - chatbot_session: inici d'una sessió de conversa amb un agent concret.
 - sesion: fragment d'interacció dins d'una sessió.
 - conversacion: registre detallat dels missatges intercanviats.

[Figura 10: Primera versió diagrama Lògic]

Aquest model permet configurar el comportament del sistema de forma flexible, mantenint la traçabilitat de les converses i garantint la seva correcta evolució. Finalment, es va elaborar el model físic, desenvolupant les taules reals que s'emprarien al sistema per a gestionar els fluxos conversacionals de manera totalment parametrizada.

Aquestes taules no només representen dades, sinó que actuen com a infraestructura de configuració dinàmica. Cada pas, condició o acció del sistema pot ser definida i modificada des de base de dades, sense necessitat de reescriure codi, afavorint així la reutilització i la flexibilitat.

[Inserir aquí: exemple real d'una taula física usada en el sistema, com una vista de TipoPaso, PasoConfigurable o similars]

En resum, aquest primer disseny de la infraestructura de dades representa un avenç decisiu en la construcció d'un sistema conversacional configurable per disseny, on el flux d'interacció es construeix a partir de dades estructurades i no de lògica embeguda al codi. Aquest enfocament facilita la mantenibilitat, l'escalabilitat i la capacitat de personalització del sistema, preparant-lo per a futurs entorns de gestió visual o integració amb agents autònoms.

[Figura 11: Primer diagrama de taules físiques]

5.4 VALIDACIÓ DEL MODEL INICIAL I TRANSICIÓ CAP A UNA ARQUITECTURA PER AGENTS

Un cop completat el primer disseny del model conversacional, juntament amb la seva estructura de dades i les taules de configuració, es va dur a terme una validació

pràctica amb l'objectiu de comprovar la seva viabilitat i detectar possibles limitacions abans de comprometre's amb una implementació definitiva.

Per fer aquesta validació, es va construir un Proof Of Concept funcional del sistema, utilitzant les taules definides en la fase anterior. Es van simular diversos escenaris conversacionals, configurant diferents passos, condicions i transicions per tal d'observar el comportament del sistema davant consultes de naturalesa diversa. Aquestes proves tenien com a propòsit avaluar:

- La capacitat del sistema per gestionar correctament els passos definits.
- La coherència en les transicions entre estats.
- La flexibilitat per adaptar-se a diversos tipus d'agents i fluxos.

Els resultats d'aquestes simulacions van confirmar que el model era tècnicament funcional, però també van posar de manifest una sèrie de limitacions pràctiques que comprometen la seva escalabilitat, mantenibilitat i usabilitat en contextos reals.

Entre les limitacions detectades, destaquen les següents:

- Rigidesa estructural: qualsevol modificació puntual requeria ajustos en múltiples taules, fet que dificultava la personalització àgil de nous agents.
- Sobrecàrrega en la definició de casos: la necessitat de definir explícitament cada característica, acció i condició generava una arquitectura excessivament complexa.
- Dificultat per representar fluxos no lineals: el model inicial no permetia modelar fàcilment bifurcacions, bucles o decisions dinàmiques basades en el context de la conversa.
- Temps elevat de configuració: la creació d'un nou agent suposava una inversió considerable de temps, contrària als objectius de reutilització i iteració ràpida.
- Limitacions en la generació dinàmica de prompts: els missatges quedaven massa acoblats a estructures predefinides, fet que dificultava l'ús de tècniques avançades com la IA generativa.

Aquest conjunt de dificultats va portar a qüestionar la validesa de l'enfocament inicial i a plantejar un canvi de paradigma en l'arquitectura general del sistema.

Com a resultat d'aquest anàlisi crític, es va decidir abandonar el model basat en la parametrizació genèrica de passos i adoptar un enfocament més flexible i funcional: la configuració directa per agent. En aquest nou model, cada agent es concep com una unitat autònoma, capaç de definir el seu comportament, els seus estats, transicions i accions específiques, sense dependre d'una estructura comuna ni compartida.

Aquest canvi suposa una evolució conceptual rellevant: es deixa enrere la idea de dissenyar chatbots genèrics per

abraçar la construcció d'agents intel·ligents especialitzats, amb capacitat per executar accions concretes com ara generar tickets en sistemes externs, enviar notificacions per correu electrònic o interactuar amb serveis addicionals.

La nova arquitectura permet que cada agent sigui autoexplicatiu, modular i independent, la qual cosa facilita el seu desplegament, manteniment i evolució. A més, elimina els efectes col·laterals que es podrien produir quan es comparteix lògica entre agents diferents, i permet adaptar-se amb agilitat a les necessitats reals i canviants de cada cas d'ús.

En definitiva, aquest replantejament arquitectònic marca l'inici d'una nova etapa del projecte, orientada a la creació de microserveis conversacionals configurables, amb un nivell d'escalabilitat i flexibilitat significativament superior a la proposta inicial.

6 NOVA ARQUITECTURA CONFIGURABLE PER AGENT I GESTIÓ DINÀMICA DEL CONTEXT

Per superar les limitacions del model anterior i assolir una arquitectura realment flexible, escalable i mantenible, es va iniciar una nova fase d'anàlisi centrada en la parametrització específica per agent. Aquest nou enfocament redefineix completament la manera en què es configura i executa el comportament dels agents conversacionals dins del sistema.

El canvi de paradigma consisteix a deixar enrere l'arquitectura estructurada mitjançant múltiples taules interrelacionades (per passos, condicions, accions, etc.), i consolidar tota la configuració en una única taula principal anomenada *TipoAgente*. Aquesta taula actua com a repositori centralitzat de configuració, on cada fila representa una clau funcional i el seu valor associat, seguint un esquema simple de parells clau-valor. D'aquesta manera, és possible encapsular tot el comportament d'un agent dins d'un conjunt de registres fàcilment editables i exportables.

[Figura 12: Taula excel de configuració "TipoAgente"]

L'estructura de la taula *TipoAgente* inclou, a més dels camps identificadors clàssics (*IdAgente*, *CodAgente*, *DesAgente*), dos camps fonamentals:

- *TxtClave*: defineix el nom funcional d'un paràmetre, com ara *ClasificarIncidencia.Prompt* o *Finalizar.Mensaje*.
- *TxtValor*: conté el valor corresponent, com pot ser un text, una condició (<DONE>) o una instrucció operativa.

Aquest model permet representar de manera flexible:

- L'activació o desactivació de funcionalitats (mitjançant valors com T/F).
- Els textos de prompt per a cada etapa del flux.

- Respostes condicionals segons el resultat d'una classificació o acció.
- Categories, protocols i variables injectades dinàmicament.

[Figura 13: Taula excel d'execució "Agente"]

Aquest enfocament ofereix una sèrie d'avantatges substancials respecte del model anterior:

- **Flexibilitat total**: cada agent pot tenir una estructura i comportament completament personalitzats.
- **Escalabilitat pràctica**: afegir un nou agent només implica crear nous parells clau-valor, sense modificar l'esquema físic.
- **Simplicitat operativa**: la centralització de la configuració facilita l'auditoria, el control de versions i la gestió funcional.
- **Compatibilitat amb IA generativa**: es poden injectar prompts dinàmics i adaptar el comportament segons la resposta del model.
- **Millora de la traçabilitat**: cada agent es pot entendre com una unitat tancada, documentada i testeable de forma independent.
- **Optimització del rendiment**: elimina l'ús de joins complexos i simplifica les consultes en temps real.

Aquest model declaratiu prepara també el sistema per a una evolució cap a un *backoffice* visual, on perfils no tècnics puguin gestionar agents mitjançant interfícies gràfiques, sense necessitat de modificar codi.

De forma paral·lela a la configuració per clau-valor, el sistema incorpora un mecanisme de persistència temporal de variables de sessió, que actua com a memòria operativa per cada instància activa del chatbot. Aquestes variables s'emmagatzemen i actualitzen dinàmicament, i permeten recordar informació essencial durant el desenvolupament de la conversa.

Algunes de les variables de sessió més representatives són:

- *PromptUsuario*: últim missatge rebut de l'usuari.
- *ContextoCorrecto*: indica si la consulta es troba dins del domini de l'agent.
- *TextoIncidencia*: descripció de la incidència proporcionada.
- *CategoriaIncidencia*: resultat de la classificació automàtica.
- *AccionSugerida* i *AccionResultado*: acció recomanada i el seu resultat (<DONE>, <NOT_DONE>).
- *CodTicketJira*: codi de seguiment del ticket generat.
- *EstadoFlujo*: estat actual del flux (p. ex. *Detectant*, *Classificant*, *Finalitzat*).
- *ConversacionActiva*: si la conversa està encara oberta.

[Figura 14: Taula física TipoAgente]

Aquestes variables condicionen el comportament del sistema, ja que permeten adaptar les respostes, decidir transicions o repetir passos segons l'estat actual de la sessió. Això és especialment rellevant en casos on el flux inclou imatges, múltiples intents o dependència d'una acció anterior, com ara la creació d'un ticket.

Aquest model obert i autoconfigurable no només millora la gestió tècnica del sistema, sinó que representa també una evolució estratègica. Passa d'un model basat en la reutilització rígida i la normalització (amb el risc de sobreenginyeria), a un enfocament que prioritza la claredat funcional, la personalització i la implantació àgil. A més, permet donar resposta a requeriments heterogenis entre agents, ja que cadascun pot tenir el seu propi protocol, estructura i flux de treball, sense afectar la resta del sistema.

Finalment, aquest nou disseny estableix una base sòlida per al creixement futur, ja que permet incorporar nous agents amb autonomia total, mantenint la coherència arquitectònica i la qualitat del sistema.

7 PROOF OF CONCEPT

Amb l'objectiu de validar la viabilitat de l'arquitectura inicial proposada i comprovar el seu funcionament en un entorn controlat, es va desenvolupar una prova de concepte (PoC) centrada en un cas d'ús real: la gestió d'incidències informàtiques. Aquesta prova no només va permetre confirmar que la primera versió del model era funcional, sinó que també va posar de manifest les seves limitacions estructurals, les quals van motivar el replantejament arquitectònic que es descriu als apartats anteriors.

Inicialment, la PoC es va construir sobre el model basat en taules relacionals, amb passos, condicions i transicions explícitament definides. Es va desenvolupar un mock funcional, en què es configuraven diversos escenaris conversacionals, per tal de comprovar si el sistema:

- Gestionava correctament els passos previstos.
- Seguía les transicions definides de manera coherent.
- S'adaptava amb flexibilitat a diferents tipus de consultes.

Aquestes primeres proves van confirmar que el model era viable tècnicament, però també van posar en evidència limitacions pràctiques relacionades amb la rigidesa del disseny, el temps de configuració, i la dificultat d'adaptació a escenaris no lineals o personalitzats.

A partir d'aquesta anàlisi crítica i gràcies a l'evidència recollida durant la PoC, es va iniciar una segona fase d'anàlisi, orientada a redefinir completament l'arquitectura del sistema, deixant enrere el model estructurat per

passos i transicions, i adoptant un model més àgil, configurable per agent.

Les funcionalitats del flux del PoC són les següents:

- Validació del context de la consulta.
- Captura i validació de la descripció de la incidència.
- Anàlisi d'imatges i generació de descripcions automàtiques.
- Resolució automàtica mitjançant IA generativa.
- Classificació segons protocol definit.
- Suggestiment d'una acció i confirmació d'execució.
- Derivació a ticket en JIRA si cal.
- Missatge de tancament i opcions de continuació.

[Figura 15: Diagrama funcionament PoC]

Tot el comportament es va definir amb parells clau-valor i variables de sessió com PromptUsuario, CategoriaIncidencia, CodTicketJira, etc., les quals van demostrar ser essencials per mantenir el context actiu i coherent al llarg de la conversa.

Els resultats de la nova PoC van ser especialment rellevants:

- Validació que el nou model és plenament funcional i flexible.
- Simplicitat operativa: tot el comportament es configura des d'una sola taula.
- Escalabilitat real: es poden definir nous agents sense modificar la base de dades física.
- Compatibilitat natural amb IA generativa i serveis externs com JIRA.
- Preparació per a futurs entorns d'edició visual (backoffice).

En conclusió, aquesta prova de concepte ha complert una doble funció crítica en el desenvolupament del projecte: primer, com a eina de validació del model inicial, i després com a instrument decisiu que va motivar la transformació arquitectònica cap al model per agents autònoms, confirmant així la direcció estratègica adoptada per a la fase final del sistema.

8 CONCLUSIÓ

El projecte ha permès definir una **arquitectura sòlida, escalable i orientada a la reutilització**, adequada per a la gestió d'agents conversacionals dins l'entorn corporatiu d'IATech. Aquesta arquitectura ha estat el resultat d'un procés evolutiu que ha sabut adaptar-se a partir de l'anàlisi funcional i les limitacions detectades en models previs.

El pas d'un model genèric compartit a una configuració **autònoma per agent** no només respon a criteris tècnics, sinó que s'alinea amb l'objectiu central del treball: identificar una forma òptima d'estructurar, desplegar i escalar fluxos conversacionals dins d'una organització complexa.

Aspecte clau	Conclusió
Enfocament arquitectònic	Disseny modular i configurable per agent, orientat a la reutilització i escalabilitat
Evolució del model	D'un enfocament genèric a una estructura pragmàtica i autònoma per agent
Innovació destacada	Ús de parells clau-valor per definir estats, prompts i accions sense duplicar lògica
Validació pràctica	PoC funcional que confirma l'aplicabilitat del model i la seva robustesa tècnica
Preparació per al futur	Arquitectura pensada per créixer: nous agents, integracions i adaptacions funcionals
Impacte potencial	Model replicable en altres entorns corporatius amb requeriments semblants

Un dels elements més destacats ha estat la **consolidació del comportament de cada agent** en un sol registre de base de dades, mitjançant parells clau-valor. Aquesta configuració encapsulada millora la governança, facilita el manteniment i permet desplegar nous agents sense afectar l'estructura general del sistema.

Tot i que el **Proof of Concept** desenvolupat no era l'objectiu principal, ha servit com a eina de validació per confirmar la robustesa del model i la seva aplicabilitat real. El sistema ha demostrat ser operatiu, estable i preparat per a escenaris reals.

Amb la base actual, el projecte està llest per a una **implementació progressiva en entorn corporatiu**, amb capacitat per adaptar-se a nous tipus d'agents, casos d'ús i integracions. Com és natural en sistemes vius, s'espera que es puguin introduir ajustos futurs en funció de les necessitats que vagin apareixent.

En resum, el treball ha assolit plenament els seus objectius, establint un **marc arquitectònic potent, flexible i extensible**, preparat per escalar i replicar-se en altres entorns similars.

9 AGRAÏMENTS

Vull expressar el meu agraïment més sincer a totes les persones i equips que han fet possible aquest Treball de Final de Grau.

En l'àmbit acadèmic, agraeixo al meu tutor Joan Protasio, per la seva orientació constant, la disponibilitat i els seus comentaris rigorosos, que han estat clau durant tot el procés.

A nivell professional, dono les gràcies al Departament de Transformació Digital per oferir-me l'oportunitat de dur a

terme aquest projecte en un entorn real i col·laboratiu. De manera especial, agraeixo a Andrei Fernando, pel seu lideratge estratègic com a Product Owner; a Jordi García Baquero, pel seu suport constant; i a Xavier Bermejo, Òscar Martínez i Alberto Madruelo, per les seves aportacions durant tot el desenvolupament.

També vull reconèixer el suport rebut des del Departament d'Arquitectura, especialment per part de Raül Villar i Eloi Milego, les seves aportacions tècniques van ser determinants en moments clau de l'anàlisi.

Finalment, gràcies a la meva família i amics pel seu suport emocional, paciència i ànims incondicionals. Sense vosaltres, aquest projecte no hauria estat possible.

10 BIBLIOGRAFIA

- [1] OpenAI. (2023). *ChatGPT Technical Report*. <https://openai.com/research>
- [2] Microsoft. (2024). *Azure OpenAI Service Documentation*. <https://learn.microsoft.com/es-es/azure/cognitive-services/openai>
- [3] Atlassian. (2024). *JIRA Software Cloud REST API Reference*. <https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro>
- [4] GCO. (2022). *Documentació interna del framework GCO per a arquitectures empresarials*. Documentació corporativa interna.
- [5] IATech. (2023). *Documentació funcional dels mòduls d'Anàlisi de Negoci i Matriu de Categorizaciones*. Documentació interna.
- [6] IBM. (2022). *Architecting Conversational AI: Design Principles for Virtual Agents*. <https://www.ibm.com/cloud/architecture/>
- [7] Microsoft. (2024). *.NET Core Documentation*. <https://learn.microsoft.com/en-us/dotnet/core/introduction>
- [8] ISO/IEC. (2011). *ISO/IEC 25010:2011 – Systems and Software Quality Models*.
- [9] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [10] Workflow Foundation Team. (2023). *Windows Workflow Foundation Documentation*. Microsoft Docs.
- [11] Stack Overflow, GitHub, Medium. (2023–2024). Consultes tècniques puntuals a comunitats de desenvolupadors per resolució de problemes i bones pràctiques.
- [12] ISO/IEC/IEEE. (2018). *Software and Systems Engineering – Lifecycle Processes – Requirements Engineering*. ISO/IEC/IEEE 29148:2018.
- [13] Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures (REST)*. University of California, Irvine.
- [14] Van Lamsweerde, A. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley.
- [15] Ceri, S., Fraternali, P., & Bongio, A. (2000). *Web Modeling Language (WebML): A Modeling Language for Designing Web Sites*. ACM Transactions on Internet Technology.

APÈNDIX

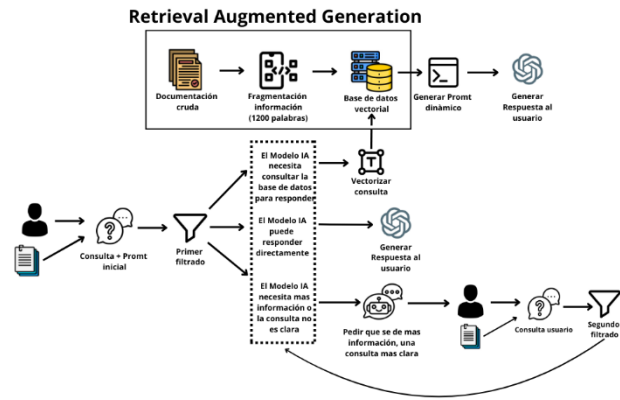


Figura 2: Diagrama sistema RAG d'Anàlisi de negoci

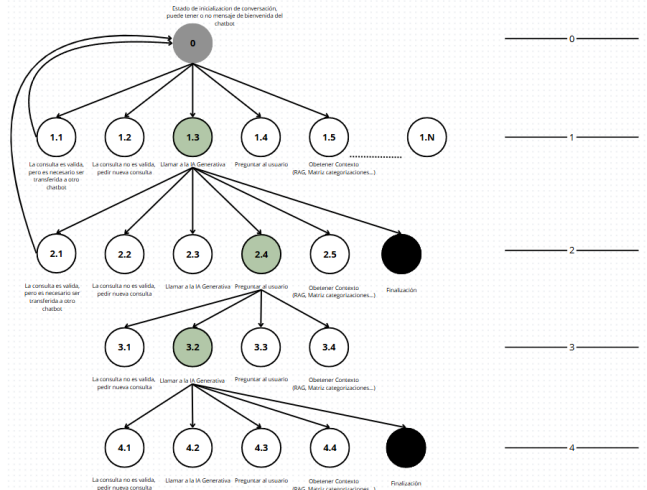


Figura 4: Diagrama sistema "Model Fil"

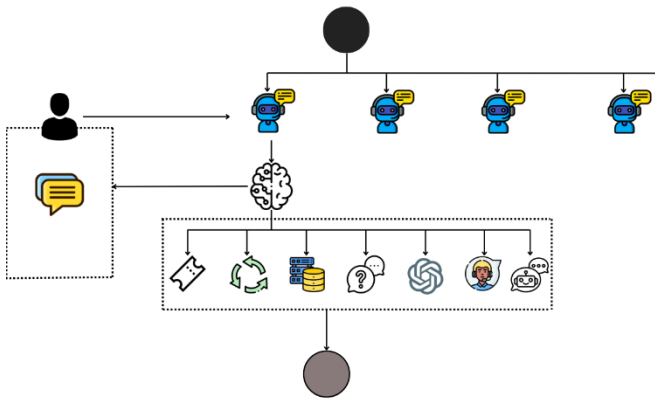


Figura 5: Diagrama sistema "cervell"

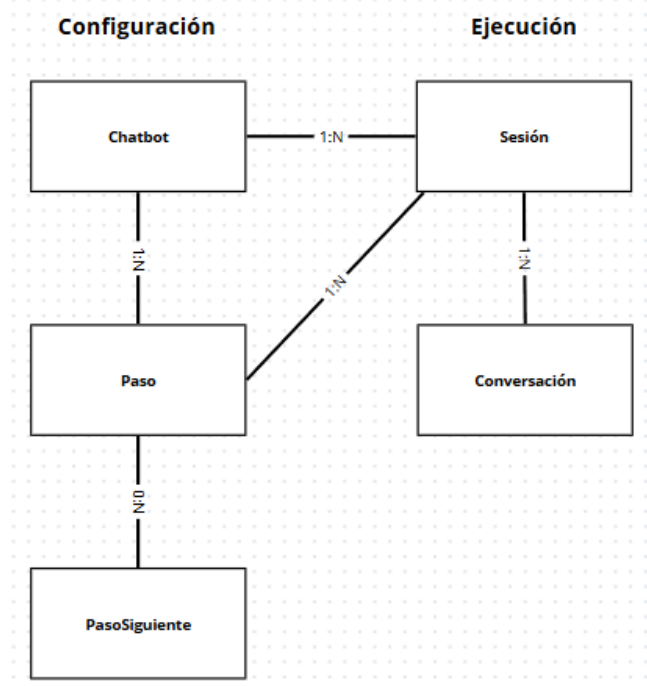


Figura 9: Diagrama E-R

PASOS						
PASO	MENSAJE	TIPO DE MENSAJE	TIPO DE PASO	REQUIERE RESPUESTA	PASO INICIAL	PASO FINAL
Inicializar Conversación	Hola, ¿en qué puedo ayudarte respecto a x? // * *	Predefinido	Inicialización	True	True	False
Declarar acción	False	NO	Decisión	False	True	False
Pedir nueva consulta	Lo siento, no puedo responder a eso. ¿Puedes reformular tu pregunta?	Predefinido	Interacción con el usuario	True	False	True
Derivar a otro chatbot	Voy a transferirte al chatbot correspondiente...	Predefinido	Acción final	False	True	True
Consultar RAG	False	NO	Obtener Contexto	False	False	False
Responder con una IA Gen	Respuesta de la IA Generativa	IA Generativa	Interacción con el usuario	True	False	True
Mostrar categorizaciones	False	NO	Obtener Contexto	False	False	False
Pregunta de SI/No	¿Es correcto que te refieres a un problema de acceso? (SI/No)	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Abierta	¿Puedes darme más detalles sobre tu problema?	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Multiselección	Selecciona una de las opciones: 1, 2, 3...	IA Generativa	Interacción con el usuario	True	False	True
Pregunta Intervalo	En qué rango de valores se encuentra tu problema?	IA Generativa	Interacción con el usuario	True	False	True
Paso Finalización	Gracias por usar el chatbot, ¡hasta luego!	Predefinido	Acción final	False	False	True
Contactar con agente humano	Te conectaré con un agente humano para continuar con tu solicitud.	Predefinido	Acción final	True	True	True
Generar Ticket de JIRA	Se ha generado un ticket en JIRA para hacer seguimiento a tu incidencia.	Predefinido	Acción final	True	False	True
Consultar base de datos interna	False	NO	Obtener Contexto	False	False	False

Figura 6: Taula excel "Paso"

TRANSICIONES		
PASO	PASO SIGUIENTE	TIPO DE TRANSICIÓN
Inicializar Conversación	Decidir acción	Interactuar con el usuario
Decidir acción	Pedir nueva consulta	Regla predefinida
Decidir acción	Derivar a otro chatbot	Regla predefinida
Decidir acción	Consultar RAG	Sistema
Decidir acción	Responder con una IA Gen	Sistema
Decidir acción	Matriz Gatorizaciones	Sistema
Decidir acción	Pregunta de Si/No	Sistema
Decidir acción	Pregunta Abierta	Sistema
Decidir acción	Pregunta Multiselección	Sistema
Decidir acción	Pregunta Intervalo	Sistema
Pedir nueva consulta	Decidir acción	Sistema
Derivar a otro chatbot	Inicializar Conversación	Sistema
Responder con una IA Gen	Paso Finalización	Sistema Intención
Responder con una IA Gen	Decidir acción	Interactuar con el usuario
Pregunta de Si/No	Decidir acción	Interactuar con el usuario
Pregunta Abierta	Decidir acción	Interactuar con el usuario
Pregunta Multiselección	Decidir acción	Interactuar con el usuario
Pregunta Intervalo	Decidir acción	Interactuar con el usuario
Matriz Gatorizaciones	Responder con una IA Gen	Sistema
Decidir acción	Contactar con agente humano	Interactuar con el usuario
Decidir acción	Generar Ticket de JIRA	Interactuar con el usuario
Decidir acción	Consultar base de datos interna	Sistema
Consultar RAG	Responder con una IA Gen	Sistema
Consultar base de datos interna	Responder con una IA Gen	Sistema

Figura 7: Taula excel "Transiciones"

CONDICIONES			
PASO INICIAL	CONDICIÓN	FUENTE DE LA CONDICIÓN	PASO SIGUIENTE
Inicializar Conversación	Se recibe la consulta del usuario.	Interactuar con el usuario	Decidir acción
Decidir acción	Si la consulta no puede ser respondida y el usuario debe reformularla.	Regla predefinida	Pedir nueva consulta
Decidir acción	Si la consulta pertenece a otro chatbot especializado.	Regla predefinida	Derivar a otro chatbot
Decidir acción	Si el sistema determina que no tiene suficiente información y se necesita contexto externo.	Sistema	Consultar RAG
Decidir acción	Si el sistema determina que la consulta es clara y puede ser respondida por la IA Generativa.	Sistema	Responder
Decidir acción	Si el sistema determina que la respuesta requiere definir un rango de valores.	Sistema	Pregunta Intervalo
Decidir acción	Si el sistema determina que la IA necesita una confirmación binaria para continuar.	Sistema	Pregunta de Si/No
Decidir acción	Si el sistema determina que hay opciones predefinidas y el usuario debe elegir una.	Sistema	Pregunta Multiselección
Decidir acción	Si el sistema determina que la consulta es abierta y necesita más detalles del usuario.	Sistema	Pregunta Abierta
Pedir nueva consulta	Si el usuario ha reformulado la consulta.	Interactuar con el usuario	Decidir acción
Derivar a otro chatbot	Si el chatbot de destino está disponible.	False	Finalizar ciclo
Responder con una IA Gen	Si el usuario no hace más consultas o ya está satisfecho.	Interactuar con el usuario	Paso Finalización
Responder con una IA Gen	Si el usuario hace otra consulta.	Interactuar con el usuario	Decidir acción
Pregunta de Si/No	Si el usuario responde con 'Sí' o 'No', se evalúa la transición.	Interactuar con el usuario	Decidir acción
Pregunta Abierta	Si el usuario proporciona una respuesta que permite clasificar la consulta.	Interactuar con el usuario	Decidir acción
Pregunta Multiselección	Si el usuario selecciona una opción válida.	Interactuar con el usuario	Decidir acción
Pregunta Intervalo	Si el usuario proporciona un valor dentro del intervalo esperado.	Interactuar con el usuario	Decidir acción
Matriz Gatorizaciones	Si ya se han recuperado los datos necesarios para dar una respuesta con la IA Generativa.	Sistema	Responder con una IA Gen
Consultar base de datos interna	Si se recuperan los datos necesarios de la base interna, se pasa a la IA Generativa.	Sistema	Responder con una IA Gen
Consultar RAG	Si se recuperan los fragmentos necesarios para responder con la IA Generativa.	Sistema	Responder con una IA Gen
Decidir acción	Si el sistema decide que necesita recuperar el protocolo de la Matriz de Gatorizaciones.	Sistema	Matriz Gatorizaciones
Decidir acción	Si el sistema determina que la consulta requiere asistencia humana.	Sistema	Contactar con agente humano
Decidir acción	Si el sistema determina que la consulta requiere generar una incidencia en JIRA.	Sistema	Generar Ticket de JIRA
Decidir acción	Si el sistema determina que la información debe obtenerse de la base de datos interna.	Sistema	Consultar base de datos interna

Figura 8: Taula excel "Condiciones"

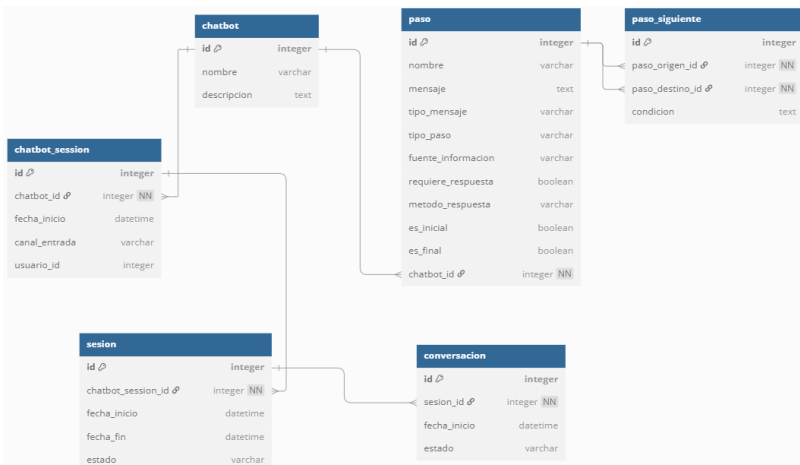


Figura 10: Primera versió diagrama Lògic

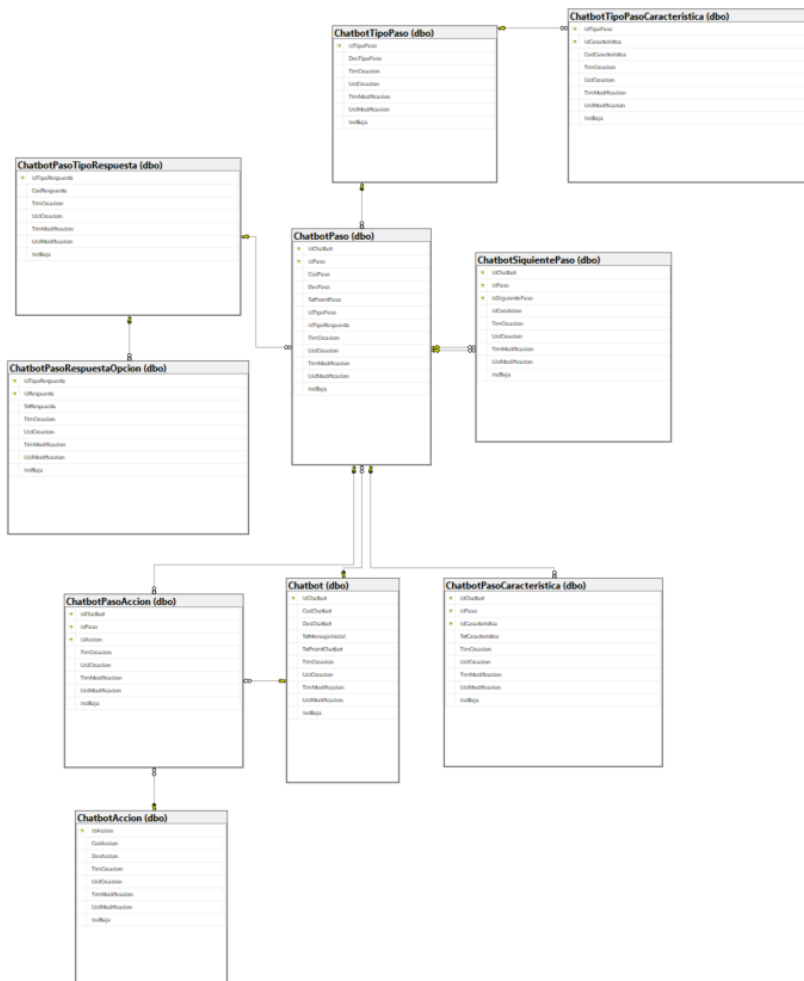


Figura 11: Primer diagrama de taules físiques

Clave		TipoAgente	Valor
Inicio.Descripcion			La temática de este chatbot es la gestión de incidencias.
ValidarContexto.Activar			T/F
ValidarContexto.Prompt			El siguiente prompt: %PromptUsuario% tiene relación con incidencias informáticas? Si el contexto es adecuado devuelve <DONE>, si no <NOT_DONE>.
ValidarContexto.Respuesta.NOT_DONE			Esta consulta no pertenece al ámbito de este chatbot. ¿Deseas que te redirija a otro asistente?
IdentificarIncidencia.Activar			T/F
IdentificarIncidencia.Prompt			¿Puedes describir brevemente tu incidencia informática?
IdentificarIncidencia.Validacion			Si se detecta una incidencia clara, devuelve <DONE>; si no, <NOT_DONE>.
IdentificarIncidencia.Respuesta.NOT_DONE			No hemos podido identificar la incidencia. ¿Podrías proporcionar más detalles?
AnalizarImagen.Activar			T/F
AnalizarImagen.Prompt			Genera una descripción detallada de esta imagen relacionada con la incidencia. Formato: párrafo único.
ResolverConocimiento.Activar			T/F
ResolverConocimiento.Prompt			¿Se puede resolver esta consulta con acciones automáticas? Si es así, sugiere las posibles soluciones.
ResolverConocimiento.Respuesta.DONE			El problema ha sido resuelto. ¿Necesitas ayuda con algo más?
ResolverConocimiento.Respuesta.NOT_DONE			No podemos resolverlo automáticamente. Vamos a intentar clasificar la consulta.
ClasificarIncidencia.Activar			T/F
ClasificarIncidencia.Prompt			Clasifica esta incidencia: %DescripcionIncidencia% según las opciones disponibles del protocolo asignado.
ClasificarIncidencia.Sistema			Filtra el Excel de categorizaciones usando el protocolo retornado por el módulo de Analítica de Negocio.
ClasificarIncidencia.Categorias			Correo, Conectividad, Software, Hardware, Otros
PedirInfoExtra.Activar			T/F
PedirInfoExtra.Prompt			Añade cualquier otra información que creas relevante para entender mejor el problema.
SugerirAccion.Activar			T/F
SugerirAccion.Prompt			Realiza esta acción para intentar resolver la incidencia: %AccionSugerida%
GenerarTicketJira.Activar			T/F
GenerarTicketJira.Prompt			No hemos podido resolver tu incidencia. Vamos a generar un ticket. Este es el resumen: %ResumenIncidencia%
GenerarTicketJira.Confirmacion			¿Confirmas que el resumen es correcto para crear el ticket?
GenerarTicketJira.Modificacion			Si deseas modificar la descripción, puedes escribir una versión mejorada.
GenerarTicketJira.Envio			Ticket creado. Código de seguimiento: %CodTicket%
Finalizar.ConfirmacionUsuario			Gracias. Tu incidencia ha sido registrada correctamente. ¿Deseas hacer otra consulta?
Finalizar.Mensaje			Gracias por contactar con el servicio de soporte. Si necesitas más ayuda, no dudes en escribir de nuevo.
FinalPrompt.Activar			T/F
FinalPrompt.Missatge			Gràcies per contactar amb el servei de suport. Si necessites més ajuda, torna a escriure.

Figura 12: Taula excel de configuració "TipoAgente"

Agente	
VARIABLES	DESCRIPCION
PromptUsuario	Último mensaje recibido del usuario.
ContextoCorrecto	T/F - Indica si el mensaje está dentro del ámbito del chatbot.
ListaAttachments	Lista de archivos enviados.
ListaAttachmentsDescripciones	Descripciones generadas automáticamente de cada imagen.
TextoIncidencia	Descripción original de la incidencia dada por el usuario.
IncidenciaValidada	T/F - Si se ha validado que la descripción representa una incidencia clara.
CategorialIncidencia	Resultado de la clasificación automática (ej. Software, Correo...).
InfoExtra	Texto adicional aportado por el usuario.
AccionSugerida	Acción técnica sugerida por el sistema.
AccionResultado	Resultado de aplicar la acción sugerida (<DONE>, <NOT_DONE>).
ResumenIncidencia	Texto completo que se usará para crear el ticket.
ConfirmacionResumen	T/F - Si el usuario ha confirmado el resumen antes de crear el ticket.
CodTicketJira	Código de seguimiento generado por Jira.
EstadoFlujo	Estado actual del flujo (Detectando, Clasificando, Finalizado, etc.).
ConversacionActiva	T/F - Si la conversación está abierta o cerrada.

Figura 13: Taula excel d'execució "Agente"

TipoAgente (dbo)	
IdTipoAgente	
CodAgente	
DesAgente	
TxtMensajelnicial	
TxtPromtAgente	
TxtClave	
TxtValor	
TimCreacion	
UidCreacion	
TimModificacion	
UidModificacion	
IndBaja	

Figura 14: Taula física TipoAgente

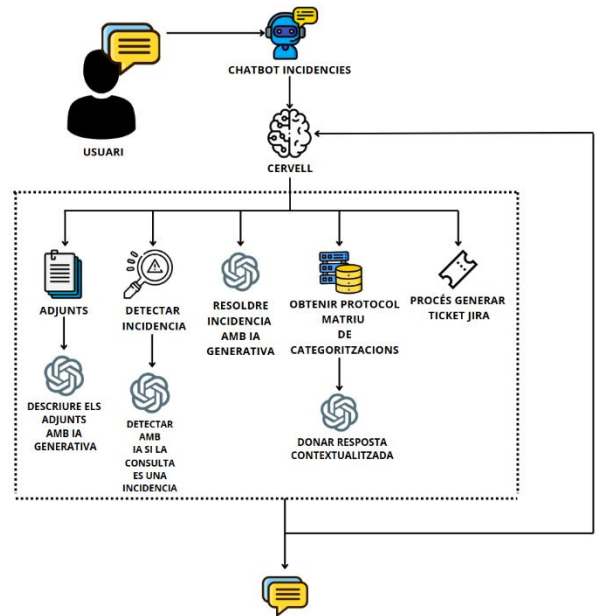


Figura 15: Diagrama funcionament PoC