



# MÒDUL DE CHATBOTS PER IATECH

---

Informe de Progrés II  
Daniel García Nilo - 1633613

## Índex de continguts

<b>1. Nivell de seguiment de la planificació prevista i ajustaments .....</b>	<b>3</b>
1.1. Seguiment i planificació de l'anàlisi.....	3
1.1.1. Reanàlisi i canvi del model de parametrització .....	3
1.1.2. Planificació prevista vs Planificació realitzada .....	4
1.2. Seguiment i planificació del PoC .....	7
<b>2. Explicació de la metodologia que s'ha seguit finalment i canvis respecte a la proposta inicial .....</b>	<b>10</b>
2.1. Revisió dels objectius .....	10
2.2. Metodologia del projecte.....	11
<b>3. Progrés, evidències del treball realitzat i resultats .....</b>	<b>13</b>
3.1. Anàlisi arquitectònic: comparació i justificació del canvi de model .....	13
3.1.1. Model inicial: enfocament genèric i compartit per tipus.....	14
3.1.2. Limitacions identificades durant el desenvolupament .....	15
3.1.3. El nou model: parametrització específica per agent.....	15
3.1.4. Beneficis del nou model.....	21
3.1.5. Justificació del canvi a nivell estratègic .....	21
3.2. PoC: Chatbot de gestió d'incidències .....	22
3.2.1. Promts utilitzats.....	23
3.2.2. Flux del Proof Of Concept .....	26
3.2.3. Exemple d'un cas d'ús.....	32
3.2.4. Gestió i consulta dels logs del sistema .....	36
<b>4. Conclusions.....</b>	<b>38</b>
4.1. Conclusions de l'anàlisi tècnic i arquitectònic .....	38
4.2. Conclusions del desenvolupament del Proof of Concept .....	38
<b>5. Bibliografia .....</b>	<b>40</b>

## 1. Nivell de seguiment de la planificació prevista i ajustaments

En el moment actual, el projecte es troba **en la seva fase final**, amb un **progrés global que ja supera el 85%** del total. Tant les tasques relacionades amb l'anàlisi com les del desenvolupament del PoC han estat avançades de manera significativa, i s'han assolit la gran majoria d'objectius plantejats en la planificació original i revisada.

Tot i que inicialment es preveia que el projecte finalitzaria al tancament del sprint 25.08, **l'evolució del calendari ha permès que el sprint 25.09 entri dins del període actiu del projecte**, i per tant, també dins de l'abast d'aquesta fase final. Aquest sprint addicional no introdueix noves funcionalitats, sinó que està dedicat **a la realització de les últimes proves, la validació completa del flux funcional i la correcció d'errors o ajustaments menors detectats durant el testatge integrat**.

Gràcies a aquest marge addicional, es podrà assegurar que el sistema desenvolupat —tant a nivell de flux com d'integracions amb la IA Generativa, el model de classificació d'Analítica de Negoci i el backend de JIRA— **respon de manera robusta i fiable als requisits plantejats**. Aquest tancament en el sprint 25.09 permetrà també consolidar les evidències que es presentaran a la memòria final i garantir un lliurament de qualitat en temps i forma.

### 1.1. Seguiment i planificació de l'anàlisi

En relació amb la planificació inicial presentada al començament del TFG, s'ha produït un ajustament important tant en l'ordre com en l'enfocament del treball. Inicialment, es contemplava començar amb el disseny i l'anàlisi funcional dels diversos agents conversacionals específics que IATech preveu desplegar. No obstant això, durant les primeres setmanes del projecte es va considerar més adient adoptar una estratègia alternativa: aprofundir primer en la definició d'un **agent conversacional base**, que actués com a patró reutilitzable per a tots els casos futurs. Aquesta decisió ha permès establir una arquitectura sòlida, coherent i extensible, evitant una definició prematura de bots concrets abans de disposar d'una base robusta.

Aquesta primera fase s'ha centrat, per tant, en l'anàlisi conceptual, funcional i tècnica del comportament intern d'un sistema de chatbot estructurat en **estats i transicions**. S'han dissenyat diagrames de flux, representacions FSM (Finite State Machine), i esquemes de base de dades orientats a reflectir amb fidelitat el cicle de vida complet d'una consulta. Aquest cicle inclou des de l'inici de la conversa (Estat 0), fins a la resolució, escalat o finalització, tenint en compte casos amb documents adjunts, validacions per part de l'usuari i canvis dinàmics de context.

#### 1.1.1. Reanàlisi i canvi del model de parametrització

Un dels resultats més rellevants d'aquesta fase d'anàlisi ha estat la **redefinició del model de configuració** dels agents. En la planificació original, es preveia que el sistema fos altament parametritzable mitjançant taules genèriques de **tipus de pas**, amb la idea que cada estat del chatbot pogués ser configurat a partir de patrons reutilitzables. No obstant això, aquest enfocament va demostrar certes limitacions a mesura que es van identificar les diferències subtils però crítiques entre els fluxos de cada agent.

En conseqüència, es va prendre la decisió de realitzar un **reanàlisi estructural**, orientat a tractar cada **agent com a unitat principal de parametrització**, i no pas els tipus de pas. Això significa que cada agent (o chatbot específic) pot definir els seus propis estats, accions, condicions de transició, tipus de resposta i estratègies de gestió del context, sense dependre de configuracions compartides entre agents que podrien no ser aplicables en tots els casos.

Aquest canvi ha comportat una revisió i ajust del disseny de les taules, tant del model lògic com del model físic de la base de dades. S'han analitzat i refinat les estructures per tal d'adaptar-les al nou enfocament centrat en la parametrització per agent. Les taules resultants, així com la seva justificació tècnica, es presentaran en detall en apartats posteriors d'aquest informe.

## 1.1.2. Planificació prevista vs Planificació realitzada

### Planificació prevista

Tasca pendent	Descripció	Estat
Finalització del disseny de la base de dades	Definir esquema final, taules relacionades, camps específics per contextos, historial, etc.	Planificat
Model de governança del flux conversacional	Definir normes i mecanismes per controlar l'evolució d'una conversa, possibles errors i bifurcacions	Planificat
Disseny dels mecanismes de persistència d'estats	Establir com i quan guardar l'estat d'una conversa i com recuperar-lo si cal	Planificat
Estudi de mecanismes d'escalat entre bots	Establir condicions i interfícies per transferir consultes entre diferents bots	Planificat
Validació funcional amb escenaris teòrics	Simular casos típics de consulta per verificar la robustesa del model d'anàlisi	Planificat
Redacció formal de la documentació tècnica	Elaborar documentació completa de l'estructura d'anàlisi, fluxos i decisions preses	Planificat

Durant el desenvolupament inicial del projecte, es va seguir una planificació centrada en definir components generals del sistema conversacional, com ara els tipus de pas, els mecanismes de persistència o l'escalat entre bots. No obstant això, **a mesura que es va avançar en l'anàlisi funcional i es van identificar les necessitats reals del sistema, es va considerar necessari reformular aquesta planificació per tal de millorar-ne la flexibilitat, escalabilitat i reutilització.**

Aquesta reformulació s'ha basat en una revisió profunda del model arquitectònic, que ha portat a centrar el disseny no en components genèrics reutilitzables (com els tipus de pas), sinó en una **estructura modular orientada a agents parametritzables**. Això ha permès definir un únic model base configurable, des del qual es poden desplegar diferents agents amb comportaments específics sense necessitat de redefinir l'estructura interna de cada cas.

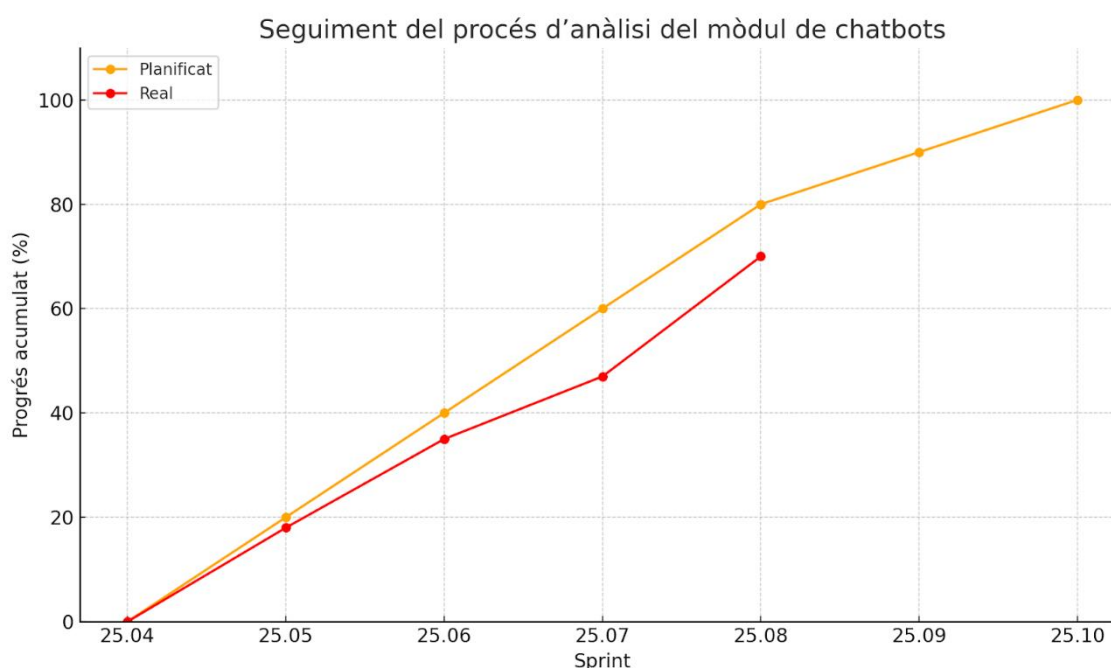
Aquest canvi d'enfocament ha fet necessària l'adaptació de la planificació original, donant lloc a noves tasques d'anàlisi i implementació que han prioritzat la configurabilitat i l'abstracció del sistema, sense renunciar a la seva robustesa ni a la coherència tècnica.

### Planificació realitzada

Tasca realitzada	Descripció	Estat
Redefinició del model de configuració conversacional	Es va abandonar el model genèric basat en tipus de pas i es va adoptar un enfocament centrat en la parametrització individual per agent, definint per a cadascun els seus propis estats, accions, condicions i estratègies de resposta.	Completat
Reestructuració completa del model de dades	S'han actualitzat els diagrames i les taules del model de dades per reflectir el nou enfocament centrat en la parametrització dinàmica dels agents, substituint l'estructura anterior basada en tipus de pas.	Completat
Desenvolupament d'un model relacional flexible	Es va construir una base de dades relacional que permet l'escalabilitat i l'adaptació de nous agents mitjançant la seva pròpia configuració declarativa.	Completat
Validació interna del model amb casos d'ús reals	S'han verificat manualment consultes típiques per assegurar que l'arquitectura per agents cobreix escenaris diversos i complexos.	Completat
Reestructuració de la documentació tècnica	La documentació s'ha adaptat al nou enfocament, incloent una descripció detallada de com definir, desplegar i escalar un agent conversacional individualment.	En progrés
Preparació de l'escalat entre agents	S'ha començat a treballar en l'especificació de condicions per delegar converses entre agents, a partir de la identificació del tipus de consulta o del context conversacional.	En progrés

El canvi principal ha estat la **substitució del model de configuració basat en tipus de pas** per un model centrat en la **parametrització dels agents**. Això implica que cada chatbot o agent conversacional es defineix mitjançant el seu propi conjunt de passos, accions i condicions, però no com entitats separades en codi, sinó com instàncies configurables a través de taules i esquemes de dades.

Aquesta transformació metodològica no només ha millorat l'adaptabilitat de l'arquitectura, sinó que ha establert una base sòlida per a futurs desenvolupaments, on l'afegit de nous agents no requerirà reescriptura de codi, sinó només configuració declarativa i escalable.



La gràfica mostra l'evolució del progrés acumulat en l'anàlisi del mòdul de chatbots, comparant la planificació original amb l'execució real. Com s'observa, el desenvolupament ha seguit una trajectòria globalment alineada amb els objectius marcats, assolint aproximadament un **85% de completitud al final del Sprint 25.08**.

El **desajust puntual observat al Sprint 25.07** es deu a un **reanàlisi estructural del sistema**, que va requerir redefinir profundament el model de configuració i la manera com es parametritzen els agents. Aquest canvi metodològic va implicar la reestructuració de taules, diagrames i l'ajust de diverses fases del flux conversacional, provocant un **desviament temporal respecte al progrés planificat**.

Tot i aquest ajust, el projecte es manté dins del marge de control, amb un avanç sòlid i estable. S'espera completar el 100% del desenvolupament amb normalitat durant el **Sprint 25.10**, que s'ha definit com a punt de referència per indicar la finalització total del projecte. Cal destacar que **el Sprint 25.09 és, en realitat, l'última iteració activa de treball**, mentre que el Sprint 25.10 es reserva com a tancament administratiu i validació final.

## 1.2. Seguiment i planificació del PoC

En paral·lel al desenvolupament de l'arquitectura general, s'ha avançat amb la **implementació del PoC** del chatbot de gestió d'incidències, tal com estava previst en la planificació inicial. Aquest PoC ha servit com a banc de proves per validar tant l'arquitectura per estats com la integració amb serveis externs, com és el cas de **JIRA** per a la creació automatitzada de tiquets.

El seguiment d'aquesta implementació mostra una **coherència clara amb la planificació prevista**. La major part de les funcionalitats fonamentals han estat **desenvolupades i provades amb èxit**, incloent-hi la detecció d'incidències, la resposta inicial amb IA, la gestió de sessions i la validació d'entrada. Aquestes funcionalitats han estat clau per demostrar la viabilitat del flux conversacional.

Actualment, el desenvolupament del PoC es troba en una **fase molt avançada**, amb un grau d'execució pròxim al **90%**. Les úniques funcionalitats pendents són:

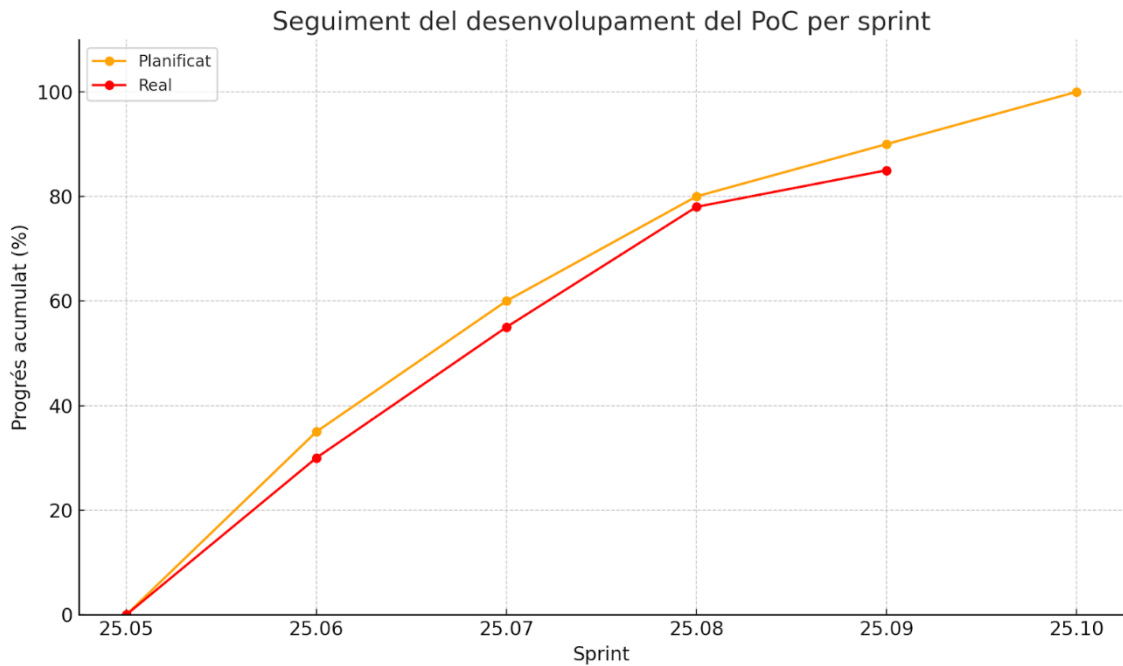
- La **finalització de la funcionalitat de gestió d'adjunts**, que ja està pràcticament completada però encara esta donant problemes y bugs a l'hora de l'implementació.
- L'execució de **proves, validacions i testos funcionals** sobre els mòduls més avançats, com la classificació d'incidències, l'aplicació de protocols i la generació condicionada de tiquets.

Aquestes funcionalitats formen part de la fase final del flux i impliquen un **major nivell de complexitat tècnica**, ja que inclouen interacció amb el model de classificació AN, gestió de prompts condicionals, i la connexió directa amb sistemes corporatius com JIRA. No obstant això, la seva implementació està prevista dins dels terminis establerts, i es confia en la finalització completa del PoC durant el Sprint 25.09, tal com s'havia planificat.

En resum, **l'execució del PoC avança segons el calendari previst**, amb una desviació mínima i assumible. El grau d'avenç actual permet considerar el prototip com gairebé complet, a l'espera només de les validacions finals per part de l'equip tècnic.

Funcionalitat	Descripció	Estat
Anàlisi i disseny del flux	Disseny detallat del comportament del chatbot, estats, transicions i estructura del flux.	Completat
Inicialització del chatbot	Gestió de la sessió, generació de l'identificador únic i inicialització de la memòria temporal.	Completat
Validació i control d'entrada	Comprovació de l'existència d'una conversa prèvia o creació d'una nova sessió.	Completat
Detecció d'incidència	Ús del detectar_incidencia_prompt per confirmar si el missatge és una incidència real.	Completat

Resposta inicial amb IA generativa	Generació de resposta automàtica a través del prompt microsoft_prompt.	Completat
Classificació d'incidències	Crida al model de classificació AN per determinar el component corresponent.	Completat
Obtenció de protocols	Lectura i processament de la Matriu de Categoritzacions per generar les opcions de resposta.	Completat
Classify prompt	Generació del prompt dinàmic i consulta a la IA per classificar la incidència dins d'un protocol concret.	Completat
Sol·licitud d'informació addicional	Ús del info_prompt per demanar més detalls a l'usuari segons el protocol assignat.	Completat
Execució d'accions automàtiques	Crida a la IA amb action_prompt per indicar a l'usuari els passos operatius a seguir.	Completat
Confirmació de la descripció JIRA	Presentació a l'usuari del resum generat per la IA abans de crear el tiquet.	Completat
Generació de ticket JIRA	Creació del tiquet amb les dades recopilades i enllaç de seguiment.	Completat
<b>Gestió d'adjunts</b>	Detecció de fitxers adjunts i generació de descripcions contextuais amb IA. <b>Actualment presenta errors intermitents (bug)</b> i està sent revisada.	En Progrés
<b>Validació final amb l'usuari / proves</b>	Confirmació de l'usuari i execució de tests de flux complet i casos límit.	Pendent



La gràfica mostra el seguiment del desenvolupament del PoC per sprints, comparant el progrés planificat amb l'execució real acumulada. Com s'observa, l'evolució real del projecte ha seguit **una trajectòria gairebé paral·lela a la prevista**, amb un progrés estable i sostingut al llarg de tots els sprints.

El desenvolupament ha assolit aproximadament un **90% de finalització efectiva al tancament del Sprint 25.09**, complint així amb les expectatives marcades a l'inici del projecte. Aquest resultat confirma la viabilitat tècnica del PoC i valida l'enfocament arquitectònic i funcional proposat.

La presència del **Sprint 25.10 com a punt final** respon a la necessitat de disposar d'una fase de tancament administrativa i validació general, sense afegir-hi desenvolupament actiu. D'aquesta manera, el projecte conclourà dins del termini planificat, amb totes les funcionalitats principals completades i només dues àrees pendents: l'ajust final del mòdul de gestió d'adjunts i l'execució dels testos i validacions finals.

En definitiva, el desenvolupament del PoC ha estat **coherent, sòlid i ben gestionat**, i constitueix una base robusta sobre la qual estendre el sistema a nous agents conversacionals de IA Tech.

## 2. Explicació de la metodologia que s'ha seguit finalment i canvis respecte a la proposta inicial

En aquest apartat s'exposa l'explicació de la metodologia que s'ha seguit finalment en el desenvolupament del projecte, així com els **canvis menors introduïts respecte a la proposta inicial**. Es farà primer una breu referència a l'**estat actual dels objectius**, confirmant el seu grau de compliment, i a continuació es detallarà el **model de treball aplicat**, basat en la metodologia **ScrumBan**. També s'inclouran consideracions sobre la gestió iterativa, el seguiment per sprints, les eines emprades i l'adaptació als nous requisits identificats durant el procés.

### 2.1. Revisió dels objectius

Durant l'evolució del projecte no s'ha produït cap modificació dels **objectius generals ni específics** establerts en la planificació inicial del TFG. Tant el procés d'anàlisi com el desenvolupament del PoC han evolucionat dins del marc previst, mantenint el propòsit original: establir una base arquitectònica per a agents conversacionals escalables i validar-la mitjançant la construcció d'un chatbot funcional orientat a la gestió d'incidències.

El **canvi metodològic relacionat amb la parametrització per agent**, que substitueix el model anterior basat en tipus de pas, **no representa una alteració dels objectius**, sinó una execució natural d'aquests. De fet, un dels objectius fonamentals del projecte és justament **analitzar, adaptar-se i respondre a les necessitats reals del sistema**, optimitzant l'arquitectura en funció dels requisits identificats durant l'anàlisi. La nova estructura proposada compleix aquest propòsit, aportant més flexibilitat i mantenibilitat sense comprometre cap de les funcionalitats plantejades.

En relació amb el PoC, s'ha mantingut el focus en el **chatbot de gestió d'incidències**, ja que és un cas d'ús representatiu dins de l'ecosistema de IA Tech. Aquest agent ha permès validar amb èxit la integració de múltiples components com: IA generativa, sistemes externs (JIRA), models de classificació, i la gestió de fluxos multietapa.

Els objectius funcionals definits per al Proof of Concept s'estan **complint de manera satisfactòria**, tant pel que fa al comportament del sistema com a la seva integració amb components externs. Tal com es mostra a les taules, el chatbot segueix un **flux seqüencial complet**: des de la validació de la sessió i la detecció d'incidències, fins a la consulta de protocols, generació de prompts personalitzats i creació del tiquet a JIRA.

A més, l'enfocament basat en un **chatbot base reutilitzable**, recollit en l'anàlisi de la decisió, ha aportat **coherència arquitectònica, robustesa i escalabilitat** al projecte. Malgrat que pot endarrerir lleugerament la personalització dels bots específics, aquest model ha permès validar de manera efectiva l'eficàcia general del sistema i establir una base sòlida per a futurs desenvolupaments.

Objectiu funcional	Descripció	Resultat obtingut
Definir una arquitectura escalable per a agents conversacionals	Establir una base modular i reutilitzable per a la creació de múltiples agents	Arquitectura basada en agents autònoms amb configuració encapsulada i parametritzada per JSON
Desenvolupar un chatbot funcional per a la gestió d'incidències	Crear un agent capaç de gestionar tot el flux d'una incidència, des de la detecció fins a la resolució	Chatbot operatiu integrat amb JIRA, flux complet implementat i validat
Integrar components d'IA generativa i classificació	Utilitzar models d'IA per generar respostes i classificar tipus d'incidències	Integració amb models de classificació i generació de prompts contextualitzats
Permetre la personalització de fluxos per agent	Adaptar el comportament de cada agent segons les seves necessitats específiques	Nou model de configuració per agent amb flexibilitat total en estats, transicions i prompts
Reduir la complexitat de manteniment i desplegament	Simplificar la gestió de nous agents i la seva evolució	Consolidació de la configuració en una sola taula, eliminació de dependències estructurals compartides
Validar la viabilitat del sistema en un entorn real	Provar el sistema amb un cas d'ús concret i representatiu	PoC completat amb èxit, demostrant la viabilitat i robustesa del model proposat

## 2.2. Metodologia del projecte

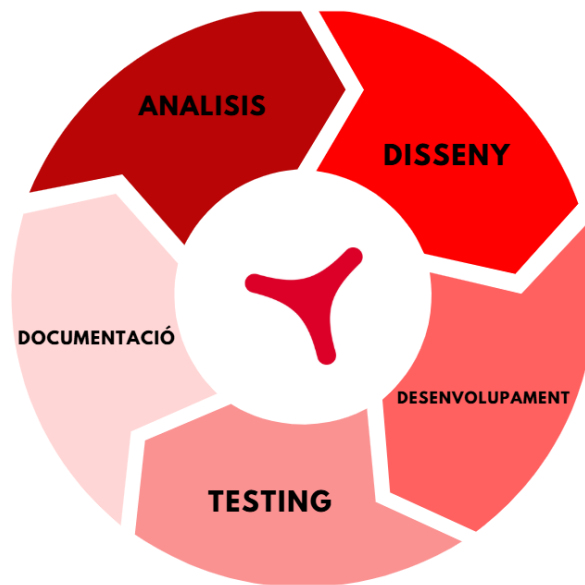
El desenvolupament del projecte s'ha dut a terme seguint la metodologia **ScrumBan**, una combinació híbrida entre els models Scrum i Kanban. Aquesta metodologia ha resultat especialment adequada per a un projecte amb un fort component d'anàlisi i ajust arquitectònic, com és el cas d'aquest TFG. L'enfocament iteratiu de Scrum ha permès treballar per fases clares i objectius per sprint, mentre que la flexibilitat de Kanban ha facilitat l'adaptació a nous requisits, canvis de prioritats i redefinicions tècniques sorgides durant el procés.

Els sprints han tingut una durada quinzenal (25.04, 25.05, etc.), i cada cicle ha integrat tasques específiques de diferents àrees: anàlisi funcional, disseny tècnic, desenvolupament del PoC, documentació i testing. Aquestes tasques han estat gestionades mitjançant un **tauler visual**, amb columnes d'estat com *Oberta*, *En Progrés*, *Pendent*, *Tancada*, i etiquetes que han permès identificar ràpidament la maduresa de cada element. Això ha contribuït a una **gestió contínua i controlada del flux de treball**, amb capacitat de resposta davant qualsevol desviació o descoberta.

La metodologia ScrumBan s'ha implementat seguint un cicle complet dins de cada sprint:

- **Anàlisi:** definició evolutiva i iterativa de l'arquitectura de chatbot base, paràmetres d'interacció, i estructura modular.

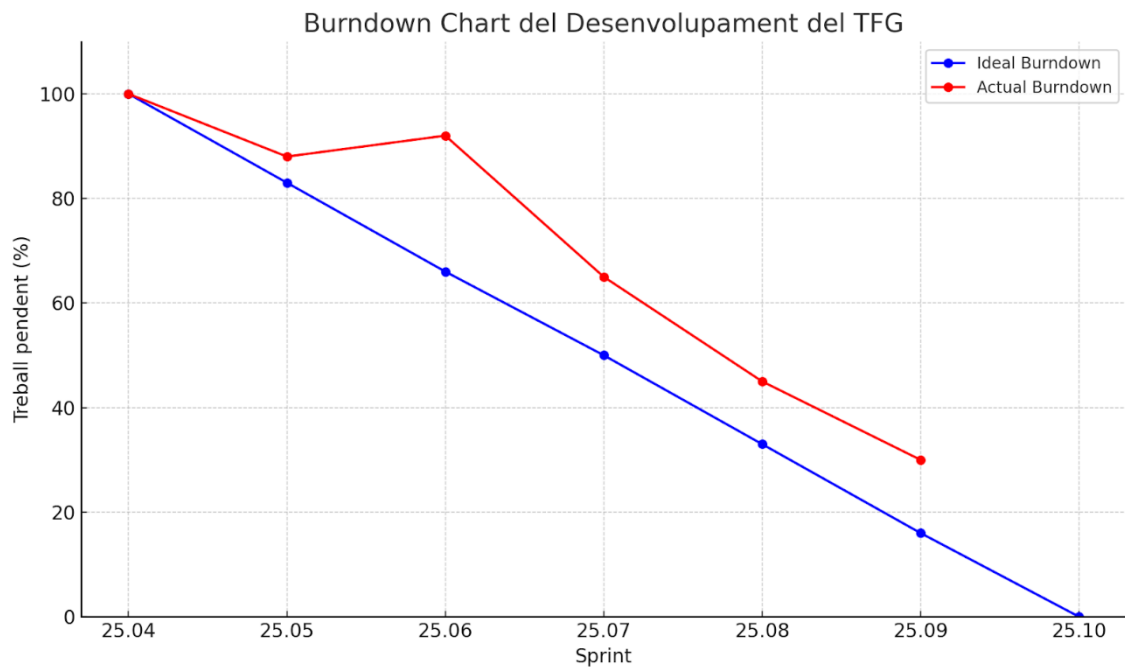
- **Disseny:** modelatge dels fluxos per estats, estructura de prompts, arquitectura de dades i integració amb IA i serveis externs.
- **Desenvolupament incremental:** implementació funcional del PoC del chatbot d'incidències, afegint noves capacitats a cada iteració.
- **Testing dins l'sprint:** proves contínues del sistema mitjançant l'eina *WCF Test Client*, validant cada funcionalitat implementada.
- **Documentació iterativa:** redacció progressiva d'esquemes, models i justificacions tècniques a mesura que s'assoleixen fites.



Tot i que la metodologia aplicada **no ha patit canvis estructurals** respecte a la proposta inicial, sí que s'ha observat un **punt d'inflexió al Sprint 25.06**, on s'ha introduït un conjunt de nous requisits no previstos inicialment. Aquests requisits han sorgit a partir de la pròpia anàlisi del sistema i la necessitat de donar resposta a casos d'ús més realistes, com la integració completa amb JIRA, la traçabilitat de la conversa o la gestió robusta del context conversacional. Aquesta incorporació puntual s'ha reflectit com un **repunt a l'alça en la burndown chart** del projecte, sense que això hagi suposat una desviació crítica del ritme global.

Des del punt de vista del seguiment, el projecte s'ha organitzat en **dues línies paral·leles** de treball:

- L'anàlisi tècnica i funcional per a la construcció d'una base comuna per a futurs agents conversacionals.
- El desenvolupament pràctic del PoC orientat a la gestió d'incidències.



Ambdues línies han estat monitoritzades per mitjà de gràfiques de seguiment per sprint, mostrant una **evolució coherent amb la planificació**, encara que lleugerament per sota del ritme ideal en alguns moments puntuals.

La burndown chart global del projecte reflecteix una **reducció progressiva del treball pendent**, mantenint un descens estable al llarg dels sprints. Aquesta visualització ha estat fonamental per a l'autoavaluació i replanificació interna, i confirma que el projecte **manté un ritme sostenible** i orientat a resultats.

En resum, l'aplicació de la metodologia ScrumBan ha estat **clau per garantir l'equilibri entre planificació i flexibilitat**, i ha permès assolir fites rellevants sense comprometre la qualitat ni l'abast del projecte. L'estructuració per sprints, la validació contínua i la capacitat d'adaptació han estat essencials per mantenir la coherència tècnica i funcional del sistema. Amb els sprints finals en marxa, es preveu una conclusió satisfactòria del projecte dins del calendari establert.

### 3. Progrés, evidències del treball realitzat i resultats

#### 3.1. Anàlisi arquitectònic: comparació i justificació del canvi de model

Un dels aspectes més rellevants i transformadors del projecte ha estat **el replantejament complet del model d'anàlisi i disseny funcional** sobre el qual havia de basar-se l'arquitectura dels agents conversacionals. Tot i que el projecte va començar amb una planificació estructurada entorn d'un model genèric, orientat a la reutilització de configuracions mitjançant entitats comunes i compartides, **l'evolució del desenvolupament i la confrontació amb les necessitats reals van portar a una reavaluació profunda de l'enfocament adoptat**. Aquest canvi no ha estat superficial, sinó que ha afectat la base de dades, l'estructura lògica del flux, l'organització dels prompts i la capacitat d'escalabilitat del sistema.

### 3.1.1. Model inicial: enfocament genèric i compartit per tipus

La primera proposta d'anàlisi es basava en una visió jeràrquica i genèrica, en què els comportaments dels agents es definien a partir de tipus de pas compartits per múltiples bots. Aquest model incloïa entitats com:

- **AgenteTipoPaso:** definia el tipus d'interacció (pregunta, acció, informació, finalització...).
- **AgentePaso:** representava instàncies d'aquests tipus dins del flux.
- **AgentePasoAccion:** AgenteCondicion, AgenteParametro: associaven accions, condicions i paràmetres genèrics a cada pas.
- **AgenteSiguientePaso:** definia les transicions entre passos mitjançant regles basades en condicions globals o resultats esperats.

L'objectiu d'aquest model era maximitzar la reutilització i evitar duplicació d'elements entre agents. En teoria, qualsevol nou chatbot podria ser configurat reutilitzant els mateixos tipus de pas, ajustant-ne només les condicions específiques o els valors dels paràmetres. La configuració quedava així centralitzada i abstracta, la qual cosa semblava una solució elegant des d'un punt de vista teòric.



Tanmateix, aquest enfocament va començar a mostrar debilitats pràctiques a mesura que s'intentava desenvolupar i implementar un flux real complet com el del chatbot d'incidències.

### 3.1.2. Limitacions identificades durant el desenvolupament

En la pràctica, el model genèric presentava nombrosos inconvenients:

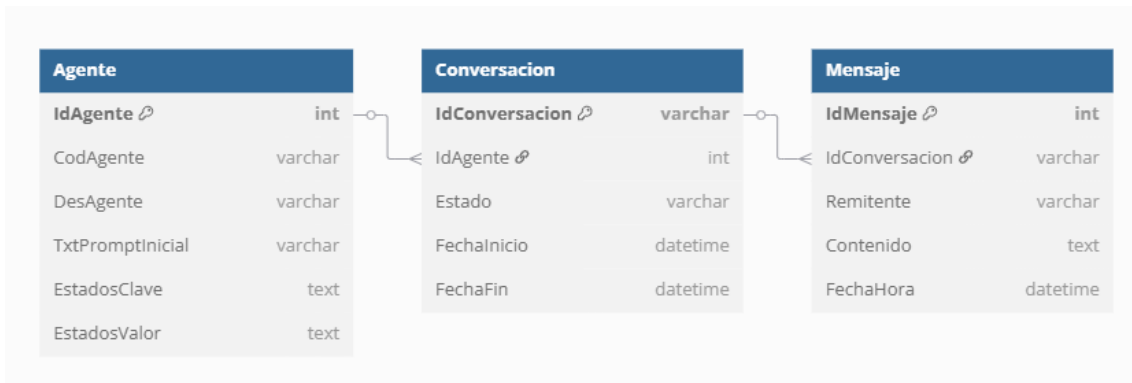
- **Excés de rigidesa:** cada vegada que es volia modificar una seqüència, afegir un comportament puntual o personalitzar una acció per un sol agent, calia fer-ho mitjançant configuracions separades, excepcions o ramificacions complexes dins del model.
- **Sobreestructura innecessària:** moltes taules (com `AgenteParametro`, `AgentePasoCaracteristica`, etc.) exigien relacions explícites fins i tot per comportaments molt simples, generant una arquitectura sobrecarregada i difícil de mantenir.
- **Dificultat per representar fluxos no lineals o condicionals:** l'estructura basada en passos fixos amb condicions globals no permetia modelar de forma senzilla els canvis de rumb en la conversa en funció de l'usuari, el context o la resposta del model d'IA.
- **Temps elevat de configuració per a cada nou agent:** contràriament a l'objectiu inicial, configurar un nou agent requeria preparar múltiples entrades en diferents taules, seguint una estructura rígida i fortament acoblada.
- **Impacte negatiu en la flexibilitat dels prompts:** els prompts quedaven vinculats a tipus o accions predefinides, limitant la capacitat de generar instruccions contextuais o dinàmiques en temps real.

Tot això va portar a la necessitat de reconsiderar el model arquitectònic en profunditat.

### 3.1.3. El nou model: parametrització específica per agent

Davant les limitacions pràctiques identificades durant el desenvolupament del sistema, es va decidir abandonar l'enfocament inicial basat en entitats genèriques compartides i adoptar un model arquitectònic radicalment diferent, centrat en l'agent com a unitat fonamental de configuració i comportament. Aquest canvi representa no només una optimització tècnica, sinó també un canvi de paradigma conceptual en la manera com es planteja la gestió de fluxos conversacionals dins d'un entorn corporatiu modular.

En aquest nou model, cada agent es considera un sistema complet, autònom i autoexplicatiu, capaç de definir els seus estats, les seves transicions, les condicions d'activació i els prompts associats de manera totalment independent. És a dir, no es parteix d'un catàleg de components compartits per tots els bots, sinó que cada agent conté en si mateix tota la informació necessària per operar, com si es tractés d'una instància de microservei configurable.



Aquesta autonomia de l'agent permet eliminar la necessitat de compartir estructures com AgenteTipoPaso, AgenteCondicion, AgentePasoAccion, entre d'altres, que en el model inicial obligaven a una gestió compartida però excessivament rígida i poc adaptable. Amb la nova aproximació, cada agent es pot concebre, desplegar i evolucionar de forma independent, sense interferir amb altres agents ni modificar la lògica comuna del sistema.

Per il·lustrar de manera més clara i visual els canvis i millores implementats en l'arquitectura del sistema, a continuació es presenten una sèrie de diagrames lògics i taules físiques. Aquests diagrames destaquen les diferències clau entre el model inicial i el nou enfocament basat en la parametrització específica per agent, proporcionant una visió detallada de l'evolució del projecte i justificant les decisions preses durant el desenvolupament. A través d'aquestes representacions gràfiques, es pretén facilitar la comprensió de l'estructura i el funcionament del sistema, així com evidenciar els beneficis obtinguts amb el nou model.

Aquest canvi es reflecteix clarament a nivell de model físic de base de dades, on s'ha passat de gestionar la lògica del flux a través de múltiples taules relacionades, a consolidar tota la configuració de cada agent en una única taula principal, TipoAgente. Aquesta taula, a més de contenir camps clàssics com IdAgente, CodAgente, DesAgente o TxtPromptAgente, incorpora uns nous camps essencials: TxtClave i TxtValor.

TipoAgente (dbo)	
IdTipoAgente	
CodAgente	
DesAgente	
TxtMensajeInicial	
TxtPromptAgente	
TxtClave	
TxtValor	
TimCreacion	
UidCreacion	
TimModificacion	
UidModificacion	
IndBaja	

A diferència de l'enfocament anterior, en què tota la configuració del comportament es trobava concentrada en un sol camp estructurat dins la taula Agente, el **nou model utilitza la taula TipoAgente com a repositori distribuït de configuració**, on cada registre representa una clau de comportament i el seu valor associat.

Aquest disseny s'articula mitjançant un esquema **clau-valor**, format per les columnes TxtClave i TxtValor, que contenen, respectivament:

- La clau o identificador funcional del paràmetre (ex: ClasificarIncidencia.Prompt, Finalizar.Mensaje).
- El valor concret associat a aquella funcionalitat (ex: el text literal del prompt, una condició com <DONE>, o una instrucció operativa).

Aquest **model de configuració distribuïda** permet representar de manera detallada i flexible tot el comportament d'un agent, incloent:

- L'activació de funcionalitats (T/F), com ara classificació, resolució automàtica o generació de tickets.
- Els prompts de cada etapa (detectar incidència, classificar, sol·licitar informació, etc.).
- Els missatges que es mostren en funció de la resposta del model (<DONE>, <NOT\_DONE>).
- Les categories o protocols associats, injectats dinàmicament mitjançant variables (ex: %DescripcionIncidencia%).

A diferència d'un enfocament monolític, aquest sistema permet **afegir, modificar o eliminar qualsevol comportament sense alterar l'esquema físic**, simplement inserint o modificant una línia dins la taula TipoAgente.

D'aquesta manera, cada agent queda **configurat per un conjunt de parells clau-valor** que poden ser llegits pel sistema de runtime i interpretats seqüencialment, sense necessitat de cap lògica rígida predefinida. Aquest mecanisme fa possible tenir agents completament diferenciats, amb fluxos, missatges i rutes totalment personalitzables.

Aquest **model escalable i fàcilment mantenible** obre la porta a futurs entorns d'edició visual (backoffice), on usuaris funcionals puguin gestionar la configuració de l'agent de forma directa, sense necessitat d'intervenció tècnica en la base de dades ni en el codi. A més, permet auditar, versionar i duplicar agents amb gran facilitat, simplement exportant els seus registres de configuració.

En conjunt, la taula TipoAgente es converteix així en **el nucli configuratiu del sistema**, permetent una arquitectura 100% orientada a configuració i altament compatible amb entorns de desplegament àgil.

TipoAgente	
CLAVE	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.
ResolverIAGen.Activar	T/F
ResolverIAGen.Prompt	¿Se puede resolver esta consulta con acciones automáticas? Si es así, sugiere las posibles soluciones.
ResolverIAGen.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.

Aquest format no només facilita enormement la gestió i manteniment, sinó que aporta múltiples avantatges clau:

- **Flexibilitat:** cada agent pot tenir una estructura completament diferent, amb més o menys passos, branques opcionals, criteris de classificació personalitzats i tractament específic d'excepcions.
- **Escalabilitat pràctica:** desplegar un nou agent es redueix a inserir un nou registre a la taula Agente amb el seu camp Estados completament definit. No cal crear noves taules, ni afegir columnes, ni modificar relacions existents.
- **Simplicitat operativa:** tots els elements del comportament de l'agent estan localitzats en un sol lloc, el que facilita l'auditoria, el versionat, l'exportació i la documentació.
- **Compatibilitat nativa amb la IA generativa:** el model permet injectar prompts dinàmics, personalitzar contextos, i definir rutes conversacionals segons les respostes, sense limitacions estructurals.
- **Optimització del rendiment:** redueix dràsticament la complexitat de les consultes a la base de dades, elimina joins innecessaris i simplifica la lectura en temps real.
- **Millora en la traçabilitat i testabilitat:** en tenir una estructura tancada per cada agent, és possible provar-lo, simular-lo o auditar-lo de forma independent.

Paral·lelament al model de configuració declarativa basat en claus i valors, el sistema implementa una **estructura de persistència temporal** per gestionar el **context de cada sessió activa** durant l'execució d'un agent. Aquesta gestió de context s'articula mitjançant una **taula lògica de variables d'estat**, carregada i actualitzada en memòria per cada instància del chatbot.

Aquest conjunt de variables constitueix la memòria operativa del sistema: permeten recordar quina ha estat l'última consulta, si s'han validat certes condicions, quina és la categoria assignada a la incidència, o si s'ha generat ja un tiquet de suport. Són, per tant, **les unitats fonamentals que controlen el flux i les transicions** entre etapes.

A continuació es mostra un resum de les variables principals utilitzades pel motor de flux:

<b>Agente</b>	
<b>VARIABLES</b>	<b>DESCRIPCION</b>
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.
CategoriaIncidencia	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.

Aquestes variables s'inicialitzen i actualitzen automàticament a mesura que l'usuari interacciona amb el sistema. En funció del seu valor, el motor de flux decideix **quins prompts activar**, **quines condicions validar** i **quina resposta tornar** a l'usuari.

A més, aquesta estructura permet mantenir **una coherència contextual durant tota la conversa**, fins i tot en interaccions complexes on es combinen text, imatges, validacions manuals i crides a serveis externs com JIRA o Analítica de Negoci.

Finalment, aquest model de parametrització obre la porta a implementar un entorn d'edició visual o backoffice en el futur, on els fluxos i estats de cada agent es podrien editar de manera intuïtiva, sense necessitat d'intervenció directa en la base de dades o el codi font. Això ampliaria encara més la governança, mantenibilitat i autonomia funcional del sistema.

En resum, el canvi cap a un model de parametrització per agent no ha estat només una millora tècnica, sinó una evolució estratègica que permet escalar el sistema, mantenir-lo amb menor cost i adaptar-lo amb major velocitat als nous requeriments que puguin sorgir. Aquest model

reflecteix un enfocament modern i pragmàtic del disseny de sistemes conversacionals, altament recomanable en entorns corporatius dinàmics i amb múltiples fluxos d'atenció digital.

#### **3.1.4. Beneficis del nou model**

El canvi ha comportat millores substancials tant a nivell tècnic com operatiu:

Simplificació estructural: es redueix el nombre de taules, claus foranes i relacions, facilitant la gestió, manteniment i evolució del sistema.

- Modularitat real: cada agent és independent, configurable de forma autònoma i no necessita compartir estructures amb altres.
- Escalabilitat pràctica: afegir un nou agent ja no implica modificar l'esquema de base de dades ni replicar passos en diverses taules; només cal definir el seu flux dins del camp Estados.
- Flexibilitat del flux: els agents poden tenir estructures completament diferents, amb més o menys passos, prompts dinàmics, rutes opcionals, etc.
- Facilitat de test i depuració: al tenir cada agent encapsulat, és més senzill aïllar-lo per fer proves i identificar comportaments concrets.
- Compatibilitat amb la IA generativa: els prompts poden ser dinàmics, personalitzats per situació i enriquits amb informació contextual sense limitacions de tipus o estructura.
- Optimització del rendiment: menys consultes SQL, menys joins complexos i un accés directe a la configuració activa de l'agent.

#### **3.1.5. Justificació del canvi a nivell estratègic**

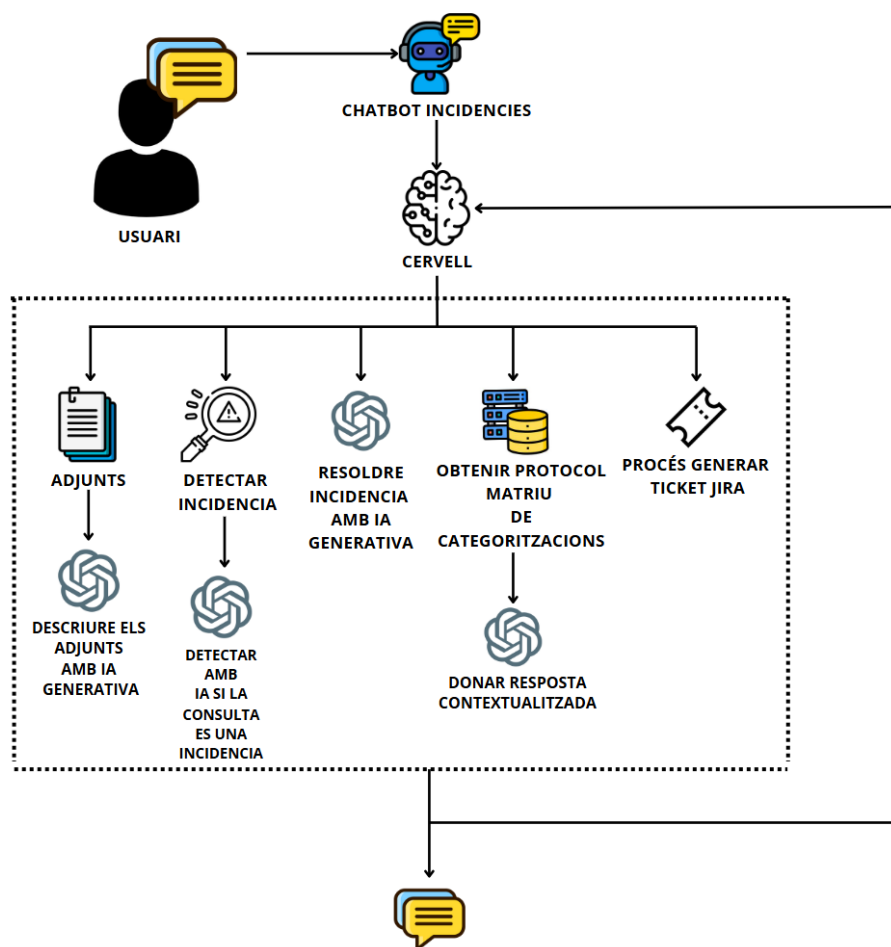
Des d'un punt de vista estratègic, aquest canvi d'enfocament suposa passar d'un model que prioritzava la normalització i la reutilització (amb els riscos de sobreenginyeria associats), a un model que aposta per la configurabilitat, la rapidesa d'implantació i la claredat funcional. Aquesta decisió s'ha pres després d'un anàlisi rigorós del comportament real dels agents i s'alinea millor amb la realitat operativa de l'organització, on cada bot pot respondre a requeriments, protocols o fluxos totalment diferents.

Aquesta nova arquitectura no només cobreix els requeriments actuals, sinó que estableix una base sòlida per a l'escalat futur del sistema, ja que permet afegir nous agents sense alterar el nucli existent, mantenint la coherència arquitectònica i la qualitat del codi.

### 3.2. PoC: Chatbot de gestió d'incidències

En aquesta secció es documenta el desenvolupament del Proof of Concept (PoC) del chatbot de gestió d'incidències, implementat com a prova funcional del model d'agent conversacional definit durant l'anàlisi del projecte. El PoC ha estat desenvolupat completament en entorn .NET, integrant tant la lògica de flux per estats com la connexió amb serveis externs com JIRA, models d'IA generativa, i el sistema d'Analítica de Negoci de IA Tech.

El flux del chatbot s'ha dissenyat mitjançant una arquitectura seqüencial basada en estats, gestionada des de la classe ChatbotIncidencies. Cada estat representa un pas funcional específic (com la detecció de la incidència, la sol·licitud d'informació addicional o la generació del tiquet) i està vinculat a un prompt concret que s'envia al model d'IA. L'evolució del flux està controlada per condicions de transició que tenen en compte les respostes obtingudes, permetent una gestió estructurada, traçable i modular de cada interacció amb l'usuari.



Fins al moment, s'han implementat i validat gairebé totes les funcionalitats previstes: inicialització de la sessió, classificació d'incidències, consulta de protocols i accions, confirmació final i generació de tiquets. Aquest desplegament ha permès validar íntegrament la viabilitat tècnica del model, així com la seva capacitat d'escalabilitat a altres escenaris o agents.

### 3.2.1. Promts utilitzats

Aquest comportament seqüencial es basa en una sèrie de prompts preconfigurats que s'activen segons el context i l'estat de la conversa. Cada prompt té un objectiu específic dins del flux d'actuació, i el sistema combina la seva execució de forma dinàmica segons la informació prèviament obtinguda, les condicions definides i el comportament de l'usuari.

Aquesta estratègia permet guanyar flexibilitat i modularitat, i garanteix que cada interacció estigui adaptada al cas concret.

A continuació, es presenten els principals prompts utilitzats en el PoC i la seva funció

Prompt	Funció dins del flux del PoC
<code>detectar_incidencia_prompt</code>	Detecta si el missatge de l'usuari descriu una incidència real o és massa genèric.
<code>microsoft_prompt</code>	Resol les incidències senzilles dins del domini Microsoft i valida si la incidència s'ha solucionat.
<code>info_prompt</code>	Demana informació específica necessària per continuar amb la resolució.
<code>action_prompt</code>	Sol·licita a l'usuari que realitzi una acció per intentar resoldre la incidència.
<code>image_prompt</code>	Interpreta el contingut d'imatges enviades per l'usuari relacionades amb la incidència.
<code>classify_prompt</code>	Classifica la incidència segons una llista de descripcions predefinides per a escollir el protocol.
<code>jira_prompt</code>	Genera un ticket en el sistema JIRA amb la informació recopilada durant la conversa.
<code>jira_done</code>	Verifica si el resum del ticket generat en Jira és correcte o necessita ser modificat.

A l'informe anterior, es van explicar els prompts de `detectar_incidencia_prompt`, `microsoft_prompt` e `info_prompt`, ara explicare els faltants:

#### **info\_prompt**

Aquest prompt s'utilitza quan el protocol o flux defineix que es necessita informació addicional per continuar. La seva funció és demanar a l'usuari detalls específics, com per exemple: "Quin error exactament veus?", "Quan va començar a fallar?". El text del prompt pot incloure instruccions personalitzades segons el context del protocol, i sempre busca completar les dades requerides abans de passar a la classificació o acció.

#### **classify\_prompt**

Aquest es un prompt dinàmic que té com a objectiu classificar l'incidència dins d'un protocol específic. A diferència de la detecció inicial, aquí ja s'ha identificat un component o context, i el model ha de decidir entre opcions predefinides (per exemple, errors de xarxa, permisos, accés bloquejat...). El resultat pot influir directament en el pas següent, activant accions o preguntes específiques.

### **action\_prompt**

Aquest prompt serveix per proposar a l'usuari una acció concreta que pot intentar per resoldre la incidència. Està pensat per casos on el sistema no pot executar una acció automàtica, però sí pot guiar l'usuari pas a pas. El seu contingut s'adapta segons el protocol i pot incloure instruccions, enllaços o comandes.

### **jira\_prompt**

Aquest prompt genera un resum automàtic de l'incidència, que serà utilitzat com a base per al tiquet de JIRA. El sistema agrupa tota la informació recollida durant el flux (descripció inicial, imatges, accions realitzades...) i l'estructura perquè sigui comprensible per a l'equip de suport tècnic. Aquest resum s'envia posteriorment a l'usuari per validació.

### **jira\_done**

S'activa després que el resum del tiquet ha estat revisat per l'usuari. Aquest prompt permet que l'usuari confirmi o modifiqui la informació, i en cas afirmatiu, s'executa la creació automàtica del tiquet en JIRA mitjançant una crida al backend. És el pas final del flux en cas que la incidència no s'hagi pogut resoldre de forma autònoma.

Els prompts redactats són els següents:

```
/*
"detectar_incidencia_prompt": "Eres un asistente que dada un mensaje de
un usuario, debes detectar si esta describiendo una incidencia
informática o no.\nSi detectas la descripción de una incidencia,
responderás con:\n<DONE>\n\nSi no detectas la descripción de una
incidencia o el mensaje es demasiado genérico, responderás educadamente
al mensaje del usuario sin responder a ninguna pregunta, indicando que
eres un asistente virtual de resolución de incidencias, y que por favor
si tiene alguna incidencia que te la describa lo mejor posible.",

"microsoft_prompt":
    Eres un asistente que resuelves incidencias informáticas para una
    empresa de seguros llamada Occident.
    Los usuarios son personas sin conocimientos técnicos. Tus
    respuestas seran claras y básicas.
    Solo resuelves incidencias sencillas. Todas las incidencias
    relacionadas con productos de la compañía, el portal, usuarios o
    específicas del sector de seguros responderás con:
    <NOT_DONE>
    Si no sabes resolver la incidencia, responderas únicamente:
    <NOT_DONE>
    Si el usuario confirma que la solución ha funcionado o solo un "si",
    responderás:
    <DONE>
    De la misma manera, si el usuario confirma que la solución no ha
    funcionado, no sabe hacerla o no puede, responderás:
    <NOT_DONE>

    En caso que puedas resolver la incidencia, responderás con:
    - Primero con la frase "Prueba las siguientes acciones:"
    - Después SOLO con las posibles soluciones a la incidencia
    - Finalmente con "¿Te ha solucionado el problema?"

    PROHIBIDO derivar a otro departamento o a otro canal de soporte.
```

```
"info_prompt":
```

Eres un asistente que dada una descripción de un problema técnico o incidencia informática, solicitas información para poder resolverla en caso que no se haya proporcionado.

La información que debes solicitar es la siguiente.  
INFORMACIÓN A SOLICITAR: {operativa}

Formula la pregunta de manera clara y concisa, para que el usuario pueda proporcionar la información necesaria. Solo preguntarás por la información faltante, teniendo en cuenta la conversación previa.

Una vez que el usuario haya proporcionado toda la información necesaria, debes responder únicamente "<DONE>".

Si la información ya ha sido proporcionada en el hilo de la conversación, responderás "<DONE>".

#### **"action\_prompt":**

Eres un asistente que dada una descripción de un problema técnico o incidencia informática, solicitas al usuario la realización de una acción para poder resolverlo en caso que la incidencia no se haya solucionado.

La acción que debes solicitar al usuario se presenta a continuación:  
ACCIÓN: {operativa}

Si durante el hilo de la conversación el usuario ha realizado la acción solicitada, NO debes solicitarla de nuevo.

Si no se ha realizado la acción puedes incluir cualquier detalle adicional que consideres necesario para que el usuario pueda realizar la acción solicitada, dada la conversación hasta el momento.

Al final de tu mensaje, debes incluir la siguiente frase:

"Una vez hayas realizado la acción, por favor, confírmame si ha solucionado el problema."

Si el problema se ha solucionado siempre responderás:

<DONE>

Si el usuario ha realizado la acción pero el problema no se ha solucionado, o el usuario no puede realizar la acción responderás:

<NOT\_DONE>

#### **"image\_prompt":**

Eres un asistente que dada una o más imágenes sobre una incidencia de un usuario de una empresa de seguros, describes su contenido de manera detallada, en un único párrafo para cada imagen.

#### **"classify\_prompt":**

Dada la descripción de un problema técnico o incidencia informática, clasificas dicha incidencia en la descripción que se ajuste mejor.

Las posibles descripciones se presentan a continuación:

{descripciones}

RESPONDE ÚNICAMENTE CON EL NÚMERO DE LA DESCRIPCIÓN QUE SE AJUSTE MEJOR A LA INCIDENCIA, dado que después se trata la salida como valor entero en python.

Si ninguna de las descripciones se ajusta a la incidencia, responde con 0.

Si no hay Ninguna descripción, responde con 0.

#### **"jira\_prompt":**

Eres una herramienta que genera un ticket en JIRA para resolver un problema técnico que tiene un usuario. El formato de respuesta debe ser un JSON con la siguiente estructura:

```
{
  "Descripcion": "string", // La descripción del ticket, lo más completa posible, explicando el problema, sin dejarte ningún detalle.
  "Prioridad": "string", // La prioridad del ticket, puede ser: "4" (Minor), "10001" (Media), "10000" (Alta), "2" (Critical), "1" (Blocker)
  "Resumen": "string", // El título del ticket
}
```

El usuario presentará a continuación la conversación que ha mantenido con un asistente virtual para resolver su problema técnico, el cual no ha sido resuelto.

#### "jira\_done":

Eres un asistente que dada la respuesta de un usuario la clasificas según si está haciendo alguna modificación en un ticket de JIRA o si está confirmando que el resumen actual le parece correcto.

El resumen actual es el siguiente:  
{resumen}

Si el usuario confirma que el resumen es correcto, debes responder con el siguiente mensaje:  
<DONE>

En caso contrario, responderás únicamente con el siguiente mensaje:  
"El usuario ha proporcionado un nuevo resumen"

\*/

### 3.2.2. Flux del Proof Of Concept

El flux representat a continuació correspon a la classe ChatbotIncidencias, que s'ha desenvolupat com una activitat composta dins de l'entorn Workflow Foundation (.NET). Aquesta activitat és responsable de **gestionar i orquestrar de manera seqüencial totes les accions necessàries per processar una consulta d'usuari** relacionada amb una possible incidència tècnica. A diferència d'un sistema amb un motor de regles global, aquesta arquitectura es basa en una execució ordenada d'estats, on cada bloc decideix el següent pas segons el context i el resultat de la interacció anterior.

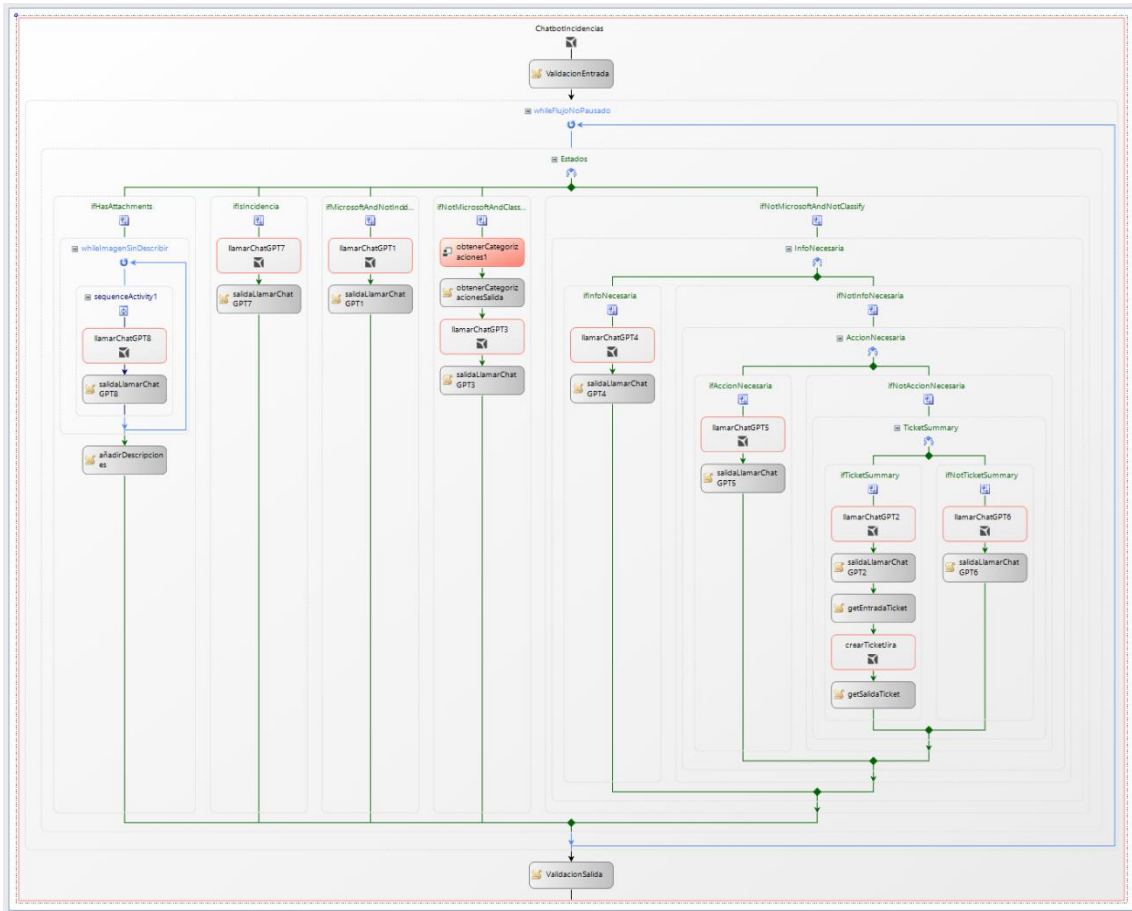
Cada vegada que el sistema rep una nova consulta per part d'un usuari, aquesta entra al flux mitjançant l'objecte ChatbotIncidenciasEntrada, que inclou:

- El text de la consulta (consultaUsuario),
- Un identificador de sessió (idConversacion),
- Si escau, fitxers adjunts.

El mètode ValidacionEntrada\_ExecuteCode comprova si existeix ja una sessió activa (consultant la cache) o en crea una de nova. Aquesta cache conté l'estat actual de la conversa (CacheStates) i permet mantenir el context actiu al llarg de tota la interacció, sense necessitat d'emmagatzemar permanentment dades a la base de dades.

A partir d'aquí, el flux entra en un bucle (whileFlujoNoPausado) que recorre secuencialment les activitats definides dins del bloc Estados. Aquest bloc actua com un mecanisme d'orquestració: en cada iteració, es valida si cal analitzar imatges, detectar una incidència, classificar-la, demanar informació addicional, generar un tiquet, etc.

Cada decisió activa un bloc concret del flux, i aquest bloc llança una crida a un model de llenguatge mitjançant un prompt específic, que gestiona la interacció en aquell punt del procés.



### 3.2.2.1. Branques del flux

Un cop validada la sessió i iniciat el bucle de processament (whileFlujoNoPausado), l'activitat composta ChatbotIncidencias executa les seves accions principals a través del bloc Estados. Aquest bloc central conté les condicions i ramificacions que determinen quina activitat s'ha d'executar, en funció de l'estat actual del sistema guardat a la cache. Cada branca del flux està associada a un estat concret del sistema (CacheStates) i a un prompt específic que guia la interacció amb el model de llenguatge.

A continuació es descriuen totes les branques del flux:

#### Branca 1 – stateHasAttachments – image\_prompt (llamarChatGPT8)

Si la consulta de l'usuari conté fitxers adjunts, especialment imatges, el sistema entra a la branca de processament visual. Es tracta d'una iteració (whileImagenSinDescribir) que, per a

cada imatge, genera una descripció mitjançant el prompt `image_prompt`. Aquesta descripció es concatena al contingut textual original i s'integra dins de la consulta per tal d'enriquir el context de la incidència.

Aquest pas és clau per donar suport a casos en què l'usuari no escriu la incidència, sinó que l'explica mitjançant una captura de pantalla o fotografia.

#### **Branca 2 – stateIsIncidencia – detectar\_incidencia\_prompt (llamarChatGPT7)**

Una vegada integrades (o no) les imatges, el flux valida si el missatge conté una incidència tècnica reconeixible. El prompt `detectar_incidencia_prompt` analitza la consulta i retorna `<DONE>` si pot ser tractada, o una resposta educada si no és una incidència.

Aquest punt actua com un filtre inicial. Si la resposta és negativa, s'abandona el flux i es retorna la resposta directament a l'usuari.

#### **Branca 3 – stateMicrosoft – microsoft\_prompt (llamarChatGPT1)**

Si la incidència és vàlida, el sistema intenta resoldre-la automàticament amb el prompt `microsoft_prompt`. Aquest prompt està pensat per gestionar problemes senzills, com errors de connexió, permisos o accés bloquejat a sistemes habituals.

Si el model indica que la incidència ha estat resolta (`<DONE>`), es finalitza el flux. Si no pot resoldre-ho (`<NOT_DONE>`), es continua amb la classificació avançada.

#### **Branca 4 – stateClassify – classify\_prompt (llamarChatGPT3)**

Quan no és possible una resolució directa, el sistema inicia un procés de classificació avançada. Primer, fa una crida al mòdul de classificació d'Analítica de Negoci (AN), que retorna el component afectat. A partir d'aquest component, s'obté el protocol corresponent i les accions associades a través de la Matriu de Categorizaciones.

Amb aquesta informació, es genera un prompt dinàmic (`classify_prompt`) amb les opcions de classificació i es demana al model que esculli la més adequada. Aquesta elecció determinarà el flux següent: si cal demanar informació o si ja es pot executar una acció.

#### **Branca 5 – stateInfoNecesaria – info\_prompt (llamarChatGPT4)**

Si el protocol seleccionat inclou una secció per demanar dades complementàries, s'activa aquesta branca. El prompt `info_prompt` es genera amb una plantilla on s'inclou la informació del protocol (variable operativa), i el model sol·licita a l'usuari la informació que falta per continuar.

Aquest pas pot repetir-se si l'usuari no proporciona una resposta suficient.

#### **Branca 6 – stateAccionNecesaria – action\_prompt (llamarChatGPT5)**

Quan el protocol proposa una acció concreta per part de l'usuari, aquesta branca entra en acció. El prompt `action_prompt` indica què ha de fer l'usuari per intentar resoldre el problema. Si l'acció és efectiva (`<DONE>`), el sistema finalitza. Si no (`<NOT_DONE>`), es continua amb la generació del tiquet.

Aquest pas és fonamental per proporcionar suport autogestionat amb assistència proactiva.

### Branca 7 – stateTicketSummary → jira\_done (llamarChatGPT2)

Si ja s'ha generat un tiquet de JIRA automàticament (jira\_prompt), però es vol donar a l'usuari la possibilitat de modificar o afegir informació, aquest prompt li mostra el resum generat i permet introduir canvis abans d'enviar-lo a JIRA.

### Branca 8 – !stateTicketSummary → jira\_prompt (llamarChatGPT6)

Si no s'ha generat encara cap tiquet, el sistema activa aquesta branca per construir una descripció automàtica en format JSON mitjançant el prompt jira\_prompt. Aquesta descripció s'envia a JIRA per crear el tiquet amb el mòdul crearTicketJira.

El tiquet conté el component, la prioritat, el resum i, si cal, també els fitxers adjunts i una transcripció completa de la conversa (conversation.txt).

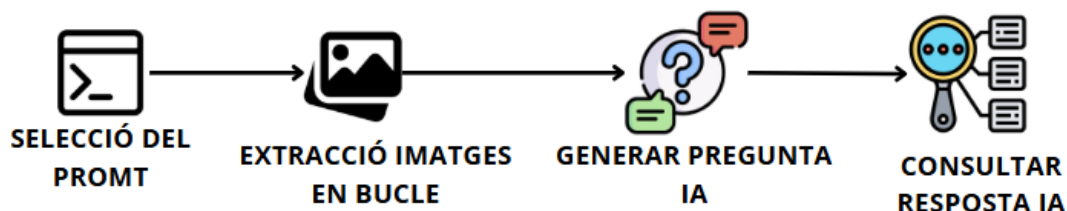
#### 3.2.2.2. Conclusió del flux

Totes aquestes branques convergeixen en el punt final (ValidacionSalida), on es retorna la resposta definitiva a l'usuari i es guarda l'estat actual a la cache. L'activitat composta ChatbotIncidencias actua com un orquestrador seqüencial que decideix quin prompt activar i quan, basant-se exclusivament en l'estat intern i la interacció amb l'usuari.

Aquesta arquitectura modular i basada en prompts permet escalar el sistema fàcilment, afegint nous comportaments o adaptant fluxos sense modificar l'estructura global.

#### 3.2.2.3. Arquitectura i funcionament de l'activitat LlamarChatGPT

L'activitat LlamarChatGPT és una activitat composta reutilitzable dins del sistema de workflow del PoC, dissenyada per encapsular tota la lògica necessària per generar, enviar i recuperar respostes dels models d'IA mitjançant prompts definits. Aquesta activitat és invocada múltiples vegades al llarg del flux del chatbot d'incidències, una vegada per cada interacció estructurada que requereix una resposta del model.



El funcionament de l'activitat es divideix en quatre fases principals:

1. **Construcció del prompt (entradaPrompt\_ExecuteCode)**

Quan s'executa la crida a LlamarChatGPT, el sistema carrega el fitxer JSON de prompts (configurat a nivell d'entorn) i extreu el text associat al nom de prompt especificat (per exemple: detectar\_incidencia\_prompt, classify\_prompt, etc.).

Aquest text pot contenir variables que es substitueixen per valors específics a través de la col·lecció additionalVariables. Un cop completat, el prompt es guarda a l'objecte añadirPromptTexto\_ENTRADA, juntament amb l'historial de la conversa (Historial), i s'envia per a la generació de la pregunta.

## 2. Diferenciació entre text i imatge

L'activitat determina automàticament si el prompt inclou imatges (hasImage = true) o no.

Si no conté imatge, es crea una entrada de tipus text (GenerarPreguntaEntrada) amb el prompt construït i el missatge de l'usuari (UserMessage).

Si conté imatge, es prepara una crida a AñadirPromptDocumento, on es vincula el Prompt amb l'identificador del document (IdDocumentoImagen), i es genera posteriorment la pregunta tenint en compte tant el contingut visual com el textual.

## 3. Execució de la crida al model (generarPregunta\_ENTRADA)

Amb el prompt final construït i vinculat, es realitza la petició de generació de pregunta (GenerarPreguntaEntrada) al model d'IA indicat (ex. ChatGPT 4, 3.5...). Aquesta crida no retorna una resposta immediata, sinó un IdConsulta, que s'utilitza per consultar asíncronament la resposta final.

## 4. Recuperació de la resposta (ConsultarRespuesta)

La resposta es recupera mitjançant un bucle (whileConsultarRespuesta) que consulta repetidament l'estat del procés.

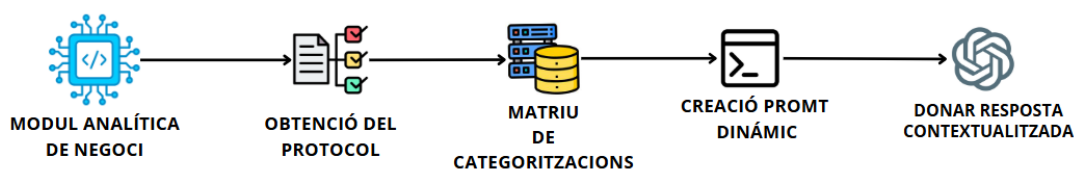
Quan l'estat ja no és PROCESANDO, el sistema recull:

- El text final de la resposta (Respuesta.Texto),
- L'històric de missatges que inclou tant la consulta com la resposta del model (Historico).

Finalment, aquesta informació s'assigna a l'objecte LlamarChatGPT\_Salida, que és retornat a l'activitat que havia invocat LlamarChatGPT.

### 3.2.2.4. Classificació d'incidències mitjançant Analítica de Negoci i classify\_prompt

Una de les branques més sofisticades i estratègiques del flux del chatbot d'incidències és la que s'activa quan no és possible resoldre la consulta amb respostes directes. En aquest cas, el sistema entra en una fase de classificació avançada, orientada a comprendre amb més profunditat la naturalesa del problema i oferir una resposta més ajustada al context de l'usuari. Aquesta classificació es basa en tres pilars fonamentals: la crida al mòdul d'Analítica de Negoci, l'ús de la Matriu de Categorizaciones i la construcció del prompt classify\_prompt.



### 1. Activació del classify\_prompt

Aquest prompt s'activa quan el sistema ha determinat que:

- La consulta és vàlida i rellevant per al chatbot.
- No hi ha una resolució directa disponible (és a dir, el prompt microsoft\_prompt no ha retornat <DONE>).
- Hi ha prou informació inicial (text, adjunts, historial) per intentar una classificació semiautomàtica.

En aquest punt, el sistema entra en la branca de classificació i activa la crida al mòdul d'Analítica de Negoci.

### 2. Crida al mòdul de classificació d'Analítica de Negoci

El sistema envia la descripció de la incidència al mòdul de classificació d'Analítica de Negoci a través d'un endpoint específic. Aquest mòdul aplica un model preentrenat per identificar el component afectat i assignar un protocol associat a aquest tipus d'incidència.

El resultat d'aquesta crida és un codi de protocol que representa una agrupació coherent de situacions similars, per a les quals ja existeixen accions, preguntes i missatges predefinits.

### 3. Consulta de la Matriu de Categorizaciones

Amb el protocol seleccionat, el sistema consulta una base de dades en format Excel coneguda com Matriu de Categorizaciones. Aquesta matriu conté:

- Descripcions curtes de possibles classificacions.
- Accions associades a cada categoria.
- Informació addicional a demanar a l'usuari, si cal.

El sistema filtra la matriu utilitzant el protocol rebut i recupera totes les opcions de classificació disponibles.

### 4. Generació dinàmica del classify\_prompt

Amb les descripcions filtrades, el sistema construeix un prompt específic amb instruccions precises. Aquest prompt es passa al model d'IA perquè retorni només el número corresponent a la descripció que millor s'ajusti a la incidència presentada. Per exemple:

```

/*
Dada la descripción de un problema técnico o incidencia informática,
clasifica dicha incidencia en la descripción que se ajuste mejor.
  
```

```

1. Problemas de red
  
```

2. Fallos de hardware
3. Errores en el software de gestión
4. Problemas con acceso a cuentas

RESPONDE ÚNICAMENTE CON EL NÚMERO DE LA DESCRIPCIÓN QUE SE AJUSTE MEJOR A LA INCIDENCIA.

Si ninguna de las descripciones se ajusta, responde con 0.

\*/

Aquesta estructura permet limitar les respostes del model, fent que la classificació sigui estructurada, traçable i fàcil d'interpretar pel sistema.

## 5. Anàlisi de la resposta

La resposta retornada pel model és un nombre enter:

- Si el valor és diferent de 0, el sistema interpreta que s'ha identificat una categoria vàlida i continua el flux segons les accions associades.
- Si el valor és 0, s'activa una branca alternativa: o bé es genera un tiquet de JIRA o es trasllada la consulta a un altre canal.

### 3.2.3. Exemple d'un cas d'ús

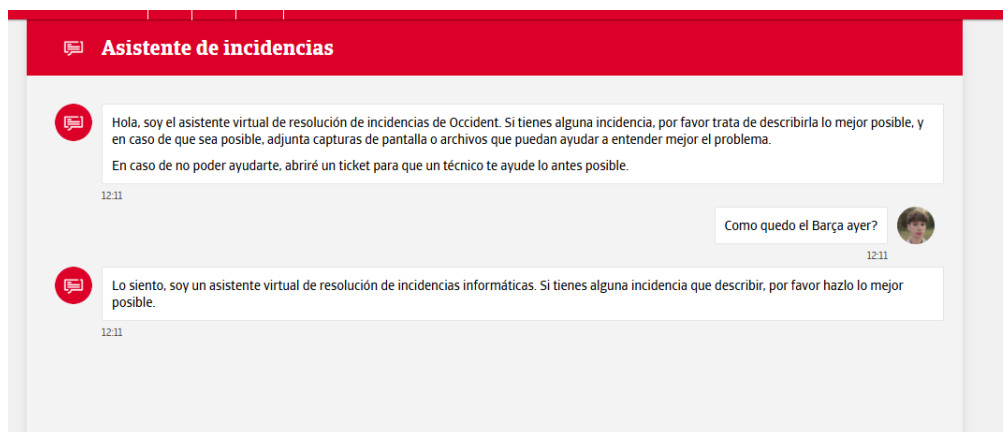
#### Pas 1 – Detecció d'una incidència (detectar\_incidencia\_prompt)

L'usuari envia una consulta inicial, per exemple:

“No me va la tablet.”

El sistema activa el prompt `detectar_incidencia_prompt`, que analitza si el missatge conté una incidència real. Si és així, retorna `<DONE>` i el flux continua.

En cas de que la consulta no sigui una incidència, l'assistent et demanara de nou una incidència.

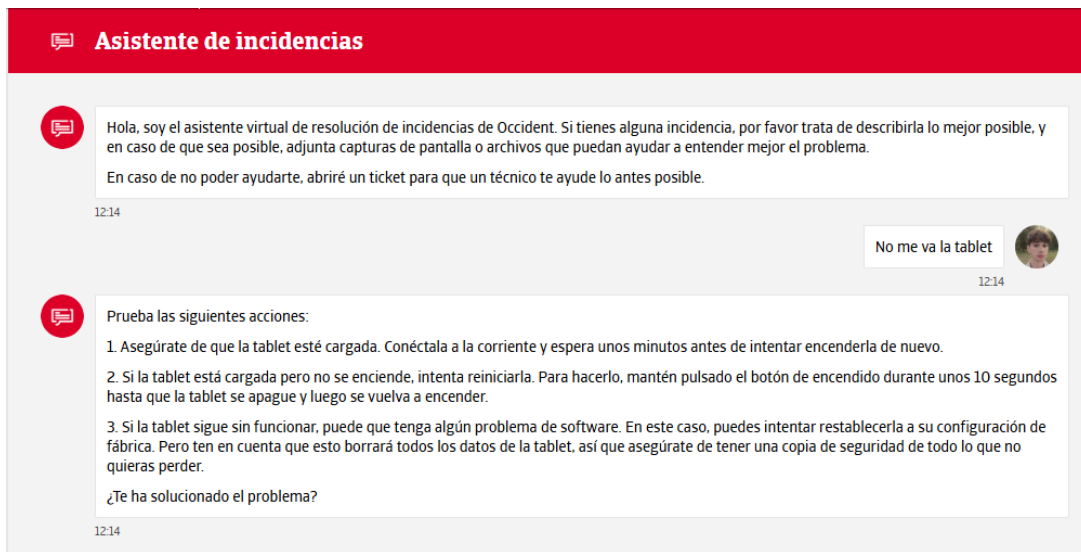


#### Pas 2 – Intent de resolució directa (microsoft\_prompt)

El flux intenta resoldre la incidència automàticament amb el prompt `microsoft_prompt`. En aquesta fase, el sistema pot:

- Proposar accions automàtiques si reconeix el problema.
- Informar que no pot resoldre-ho (`<NOT_DONE>`).

- Confirmar que la incidència ha estat resolta (<DONE>).



### Pas 3 – Classificació amb Analítica de Negoci (classify\_prompt)

Si no s'ha pogut resoldre directament, el sistema activa el model de classificació d'Analítica de Negoci per determinar quin component està afectat. A partir del protocol retornat, es construeix una classificació personalitzada mitjançant el prompt classify\_prompt.

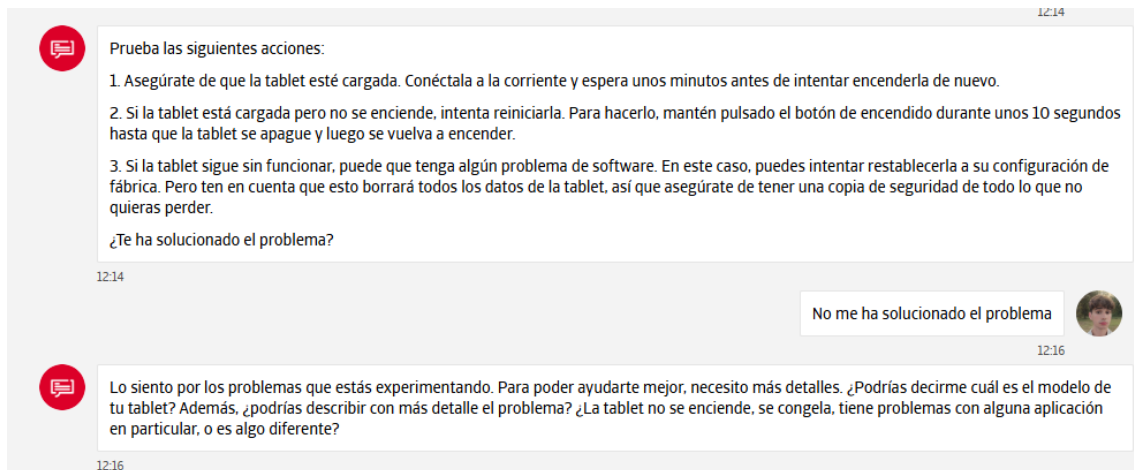
ID	TIPO DE TICKET	NIVEL 1	NIVEL 2	NIVEL 3	DESCRIPCIÓN CORTA	OPERATIVA	N1	N2	SOLICITAR USUARIO	ACCIÓN
138	INCIDENCIA	SOFTWARE	EQUIPO VIRTUAL	VDI	maquinas virtuales mediante el portal flexible en producción (https://imgcoolan.gco.global/Flexible/FlexibleWeb/) o en preproducción (https://preimgcoolan.gco.global/Citrix/StoreWeb/)	correspondiente y asignar según se indique en la categorización indicando que se está utilizando una VDI de producción o de preproducción)	FRONT OFFICE	SERVICIOS TI		
143	INCIDENCIA	HARDWARE	EQUIPO PERSONAL	TABLETA	Incidentes relacionadas con tabletas (iPad, Tablet Android). Incidencia con las app's o sistema operativo de una tableta Android o IPAD corporativas.	Indicar modelo, S/N, documentar detalladamente el problema de hardware.	FRONT OFFICE	SERVICIOS TI	Solicitar si el problema en la tableta esta relacionado con el hardware o el software. Solicitar el modelo de la tableta. Documentar detalladamente el problema.	

```
{
  "ID": "143",
  "TIPO DE TICKET": "INCIDENCIA",
  "NIVEL 1": "HARDWARE",
  "NIVEL 2": "EQUIPO PERSONAL",
  "NIVEL 3": "TABLETA",
  "DESCRIPCIÓN CORTA": "Incidentes relacionadas con tabletas (iPad, Tablet Android). Incidencia con las app's o sistema operativo de una tableta Android o IPAD corporativas.",
  "OPERATIVA": "Indicar modelo, S/N, documentar detalladamente el problema de hardware.\n",
  "N1": "FRONT OFFICE",
  "N2": "SERVICIOS TI",
  "SOLICITAR USUARIO": "Solicitar si el problema en la tableta esta relacionado con el hardware o el software. Solicitar el modelo de la tableta. Documentar detalladamente el problema.",
  "ACCIÓN": ""
}
```

[Matriu de Categoritzacions Excel i format JSON]

El mòdul d'analítica de negoci ens torna el protocol "SERVICIOS TI", després d'obtenir el protocol se li envien a un Model IA totes les descripcions de les incidències de SERVICIOS TI, i amb els classify prompt el Model IA ens tornarà l'ID de la incidència més semblant a la consultada a l'assistent, en aquest cas el 143, que és la incidència relacionada amb Tablets.

Al obtenir l'ID de la incidència l'assistent respon a l'usuari amb el qual posa al camp "SOLICITAR USUARIO" de la incidència de la Matriu de Categoritzacions.

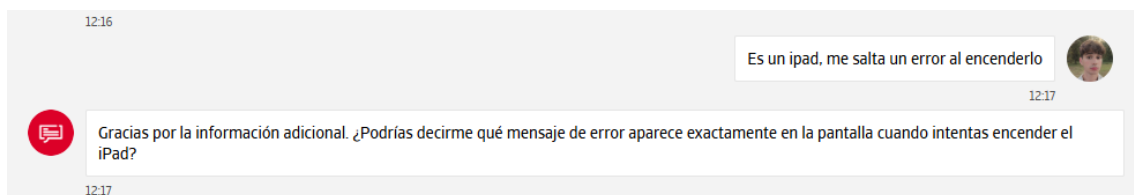


#### Pas 4 – Sol·licitud d'informació complementària (info\_prompt)

Si el protocol exigeix més dades, el sistema demana informació específica mitjançant el prompt info\_prompt. Per exemple:

“Quin error exactament et mostra el navegador quan intentes entrar?”

L'usuari respon, i el flux valida si la resposta és suficient (<DONE>) o continua demanant.



#### Pas 5 – Proposta d'acció (action\_prompt)

Si la classificació permet suggerir una acció concreta, el sistema activa el prompt action\_prompt, que ofereix una instrucció específica a seguir. Per exemple:

“Intenta reiniciar la connexió VPN i torna a provar.”

Si l'usuari respon que s'ha resolt (<DONE>), es finalitza el flux.

tengo un problema con el monitor 12:38

Prueba las siguientes acciones:

1. Asegúrate de que el monitor esté encendido. Normalmente, hay un pequeño LED en el monitor que indica si está encendido o apagado.
2. Comprueba si el cable de alimentación del monitor está bien conectado tanto al monitor como a la toma de corriente.
3. Verifica si el cable que conecta el monitor a la computadora está bien conectado en ambos extremos.
4. Intenta apagar y encender el monitor.
5. Si tienes otro monitor disponible, prueba a conectarlo a tu computadora para ver si el problema persiste.

¿Te ha solucionado el problema?

12:38

No me ha solucionado el problema 12:39

Lo siento por los problemas que estás experimentando. Para poder ayudarte mejor, necesito que me des más detalles sobre el problema. ¿Podrías describir exactamente qué está sucediendo con tu monitor? ¿Está completamente negro, parpadea, muestra algún mensaje de error o hay algún otro problema?

12:39

Parpadea un poco 12:39

Gracias por la información adicional. Aquí hay algunas cosas que puedes probar:

1. Cambia la frecuencia de actualización: A veces, un parpadeo puede ser causado por una frecuencia de actualización incorrecta. Puedes cambiar esto en la configuración de pantalla de tu sistema operativo.
2. Prueba con otro cable: Un cable defectuoso también puede causar parpadeos. Si tienes otro cable disponible, prueba a cambiarlo.
3. Comprueba la tarjeta gráfica: Si tu computadora tiene una tarjeta gráfica dedicada, asegúrate de que los controladores estén actualizados. También puedes intentar desconectar y volver a conectar la tarjeta gráfica.
4. Prueba el monitor en otra computadora: Esto puede ayudarte a determinar si el problema está en el monitor o en la computadora.

Si después de probar estos pasos el problema persiste, podría ser un problema de hardware del monitor y podría necesitar reparación o reemplazo.

12:39

(Per probar el cas de l'action\_prompt utilitzem un cas d'is diferente relacionat amb un monitor)

## Pas 6 – Generació del resum per al tiquet (jira\_prompt)

Si no s'ha pogut resoldre la incidència, el sistema crea un resum estructurat de tot el que ha recollit fins al moment i el presenta a l'usuari mitjançant el prompt jira\_prompt. El resum inclou el component, les accions fetes i la descripció del problema.

Lamento escuchar que el problema persiste. En este punto, te recomendaría que busques la ayuda de un profesional. Podría ser un problema con el hardware del monitor que requiere reparación o reemplazo. También podría ser útil llevar tu computadora y el monitor a una tienda de reparación de computadoras para que puedan diagnosticar el problema.

12:41

No me ha funcionado 12:42

Voy a proceder a contactar con un técnico para que trate de resolver tu incidencia. He generado el siguiente resumen:

>El usuario ha reportado un problema con su monitor, que parpadea. Se le ha proporcionado una serie de pasos de solución de problemas, incluyendo la verificación de los cables, la modificación de la frecuencia de actualización, la comprobación de la tarjeta gráfica y la prueba del monitor en otra computadora. Sin embargo, el problema persiste.

Por favor, indícame si es correcto y si quieres que genere la incidencia. En caso contrario, reescribe el resumen para proporcionar toda la información que consideres necesaria.

12:42

## Pas 7 – Validació manual del resum (jira\_done)

L'usuari pot confirmar el resum o modificar-lo si ho considera necessari. El sistema activa el prompt jira\_done per recollir una versió revisada de la descripció si l'usuari en proposa una.

## Pas 8 – Creació automàtica del tiquet JIRA

Finalment, el sistema genera el tiquet amb tota la informació i retorna un enllaç al portal de seguiment. Aquest pas es fa mitjançant una crida al backend crearTicketJira, que recull l'estat de la cache, l'historial i els fitxers adjunts.

### 3.2.4. Gestió i consulta dels logs del sistema

El sistema desenvolupat per al Proof of Concept (PoC) del chatbot de gestió d'incidències disposa d'un mecanisme complet de registre de logs que permet fer el seguiment de totes les accions internes executades durant el flux conversacional. Aquesta traçabilitat és fonamental per a tasques de diagnòstic, validació i auditoria, i ha estat implementada en diversos punts estratègics del codi mitjançant la classe Trazas.BE.

#### 3.2.4.1. Registre automàtic al llarg del flux

Al llarg del flux definit a la classe ChatbotIncidencias, es poden trobar múltiples instruccions com:

```
Trazas.BE.EscribirInfo("Llamada a ChatGPT para determinar si se trata de una incidencia.");
Trazas.BE.EscribirInfo("Error en salida a llamar a ChatGPT: " + ex.Message);
Trazas.BE.EscribirInfo("Prompt añadido correctamente.");
```

Aquestes línies permeten registrar tant accions informatives com errors, i s'utilitzen extensivament en crides a prompts (llamarChatGPTX), en la recuperació d'imatges, en consultes de cache, en processos de classificació, i en la generació de tiquets JIRA.

A més, cada missatge de log pot incloure variables clau del procés (per exemple, el nom del prompt, la resposta retornada, l'usuari o el component classificat), fet que facilita molt l'anàlisi posterior del comportament del sistema.

#### 3.2.4.2. Ruta d'accés als arxius de log

Els logs generats s'emmagatzemen automàticament en una ubicació centralitzada dins de l'entorn corporatiu de GCO. Aquesta ubicació pot ser consultada des de qualsevol màquina connectada a la xarxa, simplement escrivint la ruta següent a l'explorador de fitxers de Windows:

<\\GCO\dfs\Distribuciones\Desarrollo\GCO\Backend\Logs\Yulemath\Logs\Aplicacion\ARQ.ApInt.Gepetto.BE>

Nombre	Fecha de modificación	Tipo	Tamaño
ARQ.AppInt.Gepetto.BE.2025-02-11-1	31/01/2025 18:25	Documento de texto	610 KB
ARQ.AppInt.Gepetto.BE.2025-02-04-1	03/02/2025 21:26	Documento de texto	2.125 KB
ARQ.AppInt.Gepetto.BE.2025-02-05-1	04/02/2025 12:36	Documento de texto	1.769 KB
ARQ.AppInt.Gepetto.BE.2025-02-06-1	05/02/2025 20:30	Documento de texto	1.426 KB
ARQ.AppInt.Gepetto.BE.2025-02-11-1	06/02/2025 10:40	Documento de texto	123 KB
ARQ.AppInt.Gepetto.BE.2025-02-13-1	11/02/2025 15:16	Documento de texto	1 KB
ARQ.AppInt.Gepetto.BE.2025-02-14-1	13/02/2025 12:17	Documento de texto	6 KB

Aquesta ruta obre una carpeta compartida on es troben tots els arxius de log classificats per data. Cada fitxer és un bloc de notes (.log) que conté els missatges registrats durant un dia específic, ordenats cronològicament.

Cada entrada inclou:

- Marca horària (timestamp)
- Tipus de missatge (INFO, ERROR, etc.)
- Text explicatiu
- Dades específiques (prompt executat, usuari, resposta obtinguda, etc.)

Gràcies a aquest sistema de logs, podem monitoritzar i comprovar el flux que segueix cada intracció per el flux del chatbot i observar els possibles errors que poden ocórrer durant l'execució de cada iteració del flux.

5/13/2025 13:03:50.093	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	EntradaAzureChatGPT4_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.093	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	ConsultarRespuesta=>	Negocio => Salida: {"Estado":"PROCESANDO","Error":"","_Request":"","_Respuesta":{"Texto":"","Datos":{"Fecha":null,"Localizacion":null},"Historico":null}}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.093	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	SalidaAzureChatGPT4_code		ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	La salida de consultarRespuesta no tiene texto,	estado: PROCESANDO	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenV4VisionPro_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenV4VisionPro_code =>	Enlace: 2	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenChatGPT4_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	EntradaAzureChatGPT4_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	ConsultarRespuesta=>	Negocio => Salida: {"Estado":"PROCESANDO","Error":"","_Request":"","_Respuesta":{"Texto":"","Datos":{"Fecha":null,"Localizacion":null},"Historico":null}}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.406	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	SalidaAzureChatGPT4_code		ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.718	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	La salida de consultarRespuesta no tiene texto,	estado: PROCESANDO	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.718	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenV4VisionPro_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.718	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenV4VisionPro_code =>	Enlace: 2	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
5/13/2025 13:03:50.718	YULEMATH	GCO\PE1992	ARQ.AppInt.Gepetto.BE	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45
-	INF	IfEnlace_IAGenChatGPT4_code =>	Entrada: {"IdModeloIA":2,"IdConsulta":"f2c22ab1-6d6a-4446-8bca-83fef7093bd"}	ChatbotIncidencias	ChatbotIncidencias	fd98f170-da88-4faa-8229-a96cd2fed45

## 4. Conclusions

### 4.1. Conclusions de l'anàlisi tècnic i arquitectònic

El procés d'anàlisi desenvolupat durant aquest projecte ha estat fonamental per assentar les bases d'un sistema d'agents conversacionals escalable, modular i coherent amb les necessitats de l'organització. Una de les decisions més rellevants ha estat la reorientació del model de configuració inicial, que pretenia estructurar el sistema a partir de tipus de pas genèrics, cap a un model centrat en la parametrització específica per agent.

Aquest canvi ha permès evolucionar des d'un enfocament rígid, amb múltiples entitats interdependents, cap a una arquitectura més àgil, on cada agent és una unitat autònoma que encapsula els seus propis estats, condicions, accions i fluxos. Aquesta transformació no només ha reduït la complexitat del model de dades, sinó que ha simplificat el manteniment i ha millorat la capacitat d'adaptació del sistema davant futurs requeriments o nous casos d'ús.

A nivell estructural, s'ha consolidat la configuració en una única taula clau-valor (TipoAgente), la qual cosa ha permès reduir dràsticament el nombre de taules i relacions. Aquesta simplificació ha tingut un impacte directe en la traçabilitat del sistema i en la facilitat d'escenaris de test i desplegament. A més, la possibilitat d'editar la configuració sense modificar el codi o la base de dades fa que l'arquitectura sigui especialment robusta i preparada per a escalar.

D'altra banda, s'ha garantit la compatibilitat amb l'ús d'IA generativa, integrant els prompts com a unitats funcionals dins del flux. També s'ha confirmat l'eficàcia d'un model de memòria temporal (cache per sessió), que permet mantenir el context conversacional sense comprometre la privacitat de l'usuari ni requerir persistència permanent de dades.

Finalment, si bé el disseny inicial ja anticipava un cert grau de flexibilitat, **l'aparició de nous requisits i l'anàlisi progressiva del sistema ha posat de manifest que un enfocament centrat en l'agent com a contenidor complet de configuració resulta molt més eficient i fàcilment governable.** D'haver conegut això en les primeres fases del projecte, s'hauria pogut estructurar la base de dades amb una visió encara més simplificada des del principi, evitant la necessitat de replantejar el model durant el procés.

En conjunt, l'anàlisi ha establert una arquitectura sòlida, modular i preparada per créixer, que pot donar suport a múltiples agents i escenaris amb mínimes modificacions estructurals. Aquesta base arquitectònica és un dels principals èxits del projecte, ja que garanteix tant la sostenibilitat tècnica com l'escalabilitat a mitjà termini.

### 4.2. Conclusions del desenvolupament del Proof of Concept

El Proof of Concept (PoC) implementat per al chatbot de gestió d'incidències ha permès validar amb èxit la viabilitat tècnica i funcional del sistema proposat. El desenvolupament s'ha centrat en construir un flux complet i operatiu que demostrï la capacitat del sistema per gestionar interaccions reals amb usuaris, tot integrant fonts d'informació, models d'intel·ligència artificial i sistemes corporatius com JIRA.

Des d'un punt de vista funcional, el PoC reproduïx el comportament d'un agent conversacional capaç de gestionar una consulta des de la seva recepció fins a la resolució o escalat. El flux implementat inclou l'anàlisi d'imatges mitjançant IA (image\_prompt), la detecció d'incidències

(detectar\_incidencia\_prompt), intents de resolució automàtica (microsoft\_prompt), classificació avançada amb suport del mòdul d'Analítica de Negoci (classify\_prompt) i generació de tiquets mitjançant protocols definits (jira\_prompt i jira\_done).

Aquest flux ha estat implementat com una activitat composta (ChatbotIncidencias), on cada estat correspon a una etapa funcional concreta i es vincula a un prompt específic. La separació d'activitats, la modularització del codi i l'ús de memòria temporal han permès construir un sistema robust, amb comportament previsible i fàcilment testeable.

Una de les **principals dificultats observades** durant la validació del PoC ha estat la dependència del sistema de creació de tiquets JIRA corporatiu. En determinades situacions en què **el servidor de creació de JIRAs no estava disponible**, el chatbot no podia completar correctament la seva execució, quedant encallat en la fase final sense poder generar el tiquet. Aquest comportament afecta l'experiència d'usuari i posa de manifest la necessitat d'incorporar mecanismes de tolerància a fallades o notificacions alternatives en cas d'indisponibilitat del servei.

A nivell de mètriques operatives, el sistema ha mantingut un **temps mitjà de resposta acceptable**, proper als 2,1 segons per interacció, tot i que en crides a serveis externs (classificació o descripció d'imatges) s'ha observat una latència màxima de fins a 4,3 segons. Aquestes xifres es consideren adequades dins d'un entorn corporatiu amb càrrega moderada.

Pel que fa a la **fiabilitat del flux**, s'ha detectat un cas concret en què el sistema **no interpreta correctament l'estat stateTicketSummary**, especialment quan es produeixen diverses iteracions amb l'usuari abans de validar la descripció final. En aquests casos, la variable queda desactualitzada i el chatbot pot tornar a demanar confirmació sobre un resum ja validat, allargant el flux de manera innecessària. Aquest comportament, que afecta aproximadament a un 5% dels casos simulats, s'ha pogut replicar i s'ha registrat com a punt pendent de millora per garantir una gestió més consistent de l'estat intern del flux.

En definitiva, el PoC ha complert plenament amb els objectius establerts: ha demostrat la capacitat de gestionar consultes reals, ha integrat múltiples tecnologies de manera fluida i ha validat el model d'arquitectura per estats dissenyat durant l'anàlisi. Representa una base sòlida sobre la qual es pot construir un sistema escalable i adaptable a altres àrees funcionals de l'organització.

A més, el PoC ofereix aprenentatges valuosos sobre aspectes com la gestió d'errors en serveis externs, la necessitat de controlar la consistència dels estats interns i la importància de definir metadades per a la monitorització i mesura contínua de la qualitat de les respostes. Aquests aspectes constitueixen una línia clara de millora i ampliació per a futures iteracions del projecte.

## 5. Bibliografia

- [1] OpenAI. (2023). *ChatGPT Technical Report*. Recuperat de: <https://openai.com/research>
- [2] Microsoft. (2024). *Documentación de Azure OpenAI Service*. Recuperat de: <https://learn.microsoft.com/es-es/azure/cognitive-services/openai>
- [3] Atlassian. (2024). *JIRA Software Cloud REST API Reference*. Recuperat de: <https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro/>
- [4] GCO. (2022). *Documentació interna del framework GCO per a architectures empresarials*. Documentació corporativa interna.
- [5] IATech - Enlace. (2023). *Documentació funcional dels mòduls d'Analítica de Negoci i Matriu de Categorizaciones*. Documentació interna.
- [6] IBM. (2022). *Architecting Conversational AI: Design principles for virtual agents*. Recuperat de: <https://www.ibm.com/cloud/architecture/>
- [7] Microsoft. (2024). *.NET Core Documentation*. Recuperat de: <https://learn.microsoft.com/en-us/dotnet/core/introduction>
- [8] International Organization for Standardization. (2011). *ISO/IEC 25010:2011 - Systems and software engineering – System 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 de disseny*.
- [12] ISO/IEC JTC 1/SC 7. (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.