NOVA IMS Information Management School

Master's Degree Program in

MDSAA

Data Science and Advanced Analytics

Business Cases with Data Science

Case 4: AI-Powered Chatbot

Ana Caleiro
Duarte Marques
Moeko Mitani
Oumaima Ben Hfaiedh
Sarah Leuthner

Group Q

1. INTRODUCTION

In today's rapidly evolving digital environment, companies are increasingly leveraging artificial intelligence (AI) to improve operational efficiency, empower employees, and increase customer engagement. The insurance industry, in particular, is under pressure to deliver personalized, value-driven experiences while maintaining compliance and optimizing internal workflows.

Fidelidade, as the leader of the market in Portugal, is committed to this transformation. As part of its innovation strategy, the company aims to support its sales force with the **AI-powered chatbot assistant** that can increase agent productivity, simplify product communication, and promote customer financial literacy.

Here, we present the design, development, and evaluation of a proof-of-concept (POC) chatbot assistant developed using **Azure OpenAI** and **Flowise.** The solution is tailored to assist agents in facilitating sales of two investment-focused financial products, **Fidelidade Savings** and **PPR Evoluir.** Through instant access to product information, customer-friendly explanations, and intelligent handling of frequently asked questions, the chatbot enables agents to serve customers more effectively and confidently.

Focusing on practical integration and sustainable impact, the initiative shows how the strategic implementation of AI can support human agents, drive digital transformation, and deliver measurable value across the organization.

2. BACKGROUND

Fidelidade, one of the oldest insurers with over 200 years of history in the world, is a leading insurance company focused on providing secure and smart financial solutions. According to the European Commission (2023), only 24% of the citizens were rated as high finance knowledgeable people. That means it was discovered that the EU needs targeted financial education overall. As part of their digital innovation efforts, the company is developing Fidelidade Savings and PPR Evoluir, savings and investment solutions, and they need their insurance agents to be “educators and trusted guides, not just sellers”.

To improve efficiency and client satisfaction, Fidelidade aims to introduce an AI-powered chatbot that will assist insurance agents by answering product-related questions, providing comparisons, and addressing common concerns. The goal is to empower agents with digital tools while maintaining high-quality personalized support.

2.1. BUSINESS OBJECTIVES

Here are four business objectives in this project:

- **Support Agents with AI Assistance**: Equip insurance agents with a chatbot capable of providing instant and accurate product details and comparisons to better assist clients.
- **Improve Financial Literacy**: Help agents ensure that clients understand savings products and financial terms through quick, friendly, and clear explanations.
- **Enhance Operational Efficiency:** Let the chatbot handle basic queries so agents can focus on high-value interactions, leading to increased conversion and client satisfaction.

- **Enable Smooth Adoption through Employee Training**: Develop and implement a training and deployment plan to ensure agents are well-prepared to use the chatbot effectively in their daily interactions with clients.

2.2. BUSINESS SUCCESS CRITERIA

The success of this project will be evaluated based on the following criteria:

1. **Information Accuracy**: The chatbot consistently provides correct and clear explanations of My Savings and PPR Evoluir.
2. **FAQ Coverage**: It effectively answers common agent and client questions on financial literacy and insurance products.
3. **Sales Support**: It boosts sales efficiency by allowing agents to prioritize personal engagement over administrative tasks.
4. **User Experience**: The chatbot maintains natural, polite conversation and handles follow-up questions smoothly.
5. **Escalation**: The chatbot recognizes its limits and appropriately escalates complex queries to human agents.

3. DATA UNDERSTANDING

We received a total of 19 files containing relevant information about Fidelidade's products and an overall view of the business. These files are extremely important, as they represent the primary source of knowledge used by the chatbot. Given the volume of files, the explanation is grouped according to the organization and structure in which they were originally provided.

3.1. QUESTIONS AND ANSWERS

This section consists of only one file, filled with important Q&As that will help us validate the quality of the answers given by our bot. We decided to not incorporate this into the training phase of the model, since we want the model to find the answers in the other documents we provide, making it adaptable to more information in the future.

3.2. MY SAVINGS DOCUMENTS

This section is also divided in three main sub-categories:

The competitor's information:
We have one file explaining how Fidelidade's products stand out from their competitors.

Internal Information:
This group of files contains information about how Fidelidade and its products work internally. The available data ranges from the operations manual to a complete guide on how employees should answer specific questions about the My Savings product. The chatbot will be able to understand all the details about how the My Savings works, and how to explain their functionalities to the Fidelidade employees so they correctly explain it to the clients.

PPR Public Information:

This sub-section is composed of two files, one about the pre-contractual information and the other about the general conditions of My Savings. Together, they form the legal requirements and informational foundation of the product. As result of it they provide a great summary of the product's purpose, conditions, fees, risk factors, and client rights. It will enable the chatbot to provide informed, regulation-aligned responses.

3.3. PPR Evoluir Documents

PPR Evoluir documents are organized similarly to the My Savings documents, but instead of focusing on My Savings, it covers information about competitors, legal details, and the operation of PPR Evoluir, another product we are training the chatbot on.

4. DEVELOPMENT OF THE CHATBOT

As there is no Data Preparation necessary in this project, the next step is Modelling - meaning the development of the chatbot.

4.1. Setup

4.1.1. Choice of Tools

Building a chatbot from scratch through traditional coding requires significant time, specialized development skills, and infrastructure setup. For a POC, speed, flexibility and accessibility are paramount. The five tools shown in the table below - *Flowise, Langflow, LlamaIndex, Botpress,* and *Haystack* - are among the current market leaders for building AI-powered chatbots. Platforms like *Flowise* and *Langflow* offer visual interfaces and pre-built functionalities, drastically reducing development time and making them accessible even to non-technical stakeholders.

These platforms enable fast prototyping and iteration, which is ideal for validating ideas, gathering feedback, and demonstrating business value before investing in a full-scale, custom-coded solution. They also support features like Retrieval-Augmented Generation (RAG) and integrations, which are often essential in modern AI chatbots.

Table 1: Market Landscape Analysis for Tools

Feature	Flowise	Langflow	LlamaIndex	Botpress	Haystack
Visual Interface (UI)	✓	✓	✗	✓	✗
No-code friendly	✓	✓	✗	✓	✗
Speed to Build	★★★	★★★	★★	★★★	★★
Built in RAG*	✓	✓	✗	✗	✓
Easy Deployment	✓	✓	✗	✓	✗
Best for	Non-dev teams, fast bots	Devs who like visual flows	Power users & big data	UI-first business bots	Researchers & enterprises

*RAG = Retrieval-Augmented Generation

Flowise is well-suited for chatbot POC development where rapid prototyping and ease of use are essential. As a fully no-code platform, it enables users, regardless of technical background, to design conversational flows via an intuitive visual interface. This lowers the barrier to entry for cross-functional teams and accelerates development cycles.

The platform includes native support for RAG, facilitating dynamic access to external data sources. Its straightforward deployment process further supports fast iteration and testing. *Flowise* is particularly appropriate when the POC's primary aim is to evaluate user interaction and functional viability, rather than implement complex custom logic.

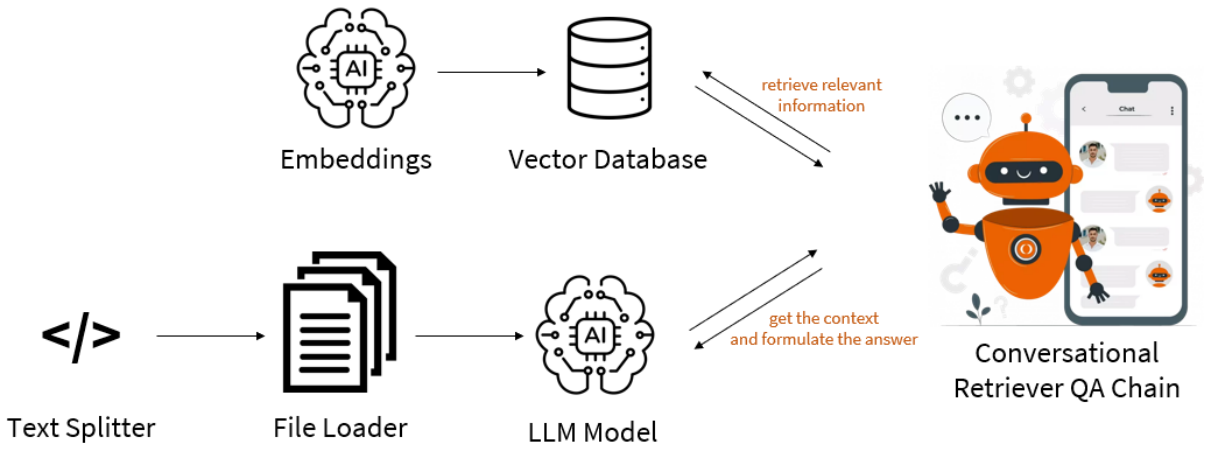
For these reasons this tool was chosen for this use case.

4.1.2. Workflow

The chatbot is implemented using a RAG architecture, which combines retrieval of relevant documents with the generative capabilities of a Large Language Model (LLM). The figure below outlines the key components and their interaction in the system:

- **Text Splitter**: Breaks input documents into chunks to optimize retrieval and model input
- **File Loader**: Loads the split text into the system for further processing
- **Embeddings**: Converts each text chunk into high-dimensional vectors using an embedding model
- **Vector Database**: Stores these embeddings, enabling fast similarity search during retrieval
- **LLM Model**: Pre-trained large language model that generates responses based on contextual input
- **Conversational Retriever QA Chain**: Coordinates the end-to-end flow from user input to response generation.

Figure 1: Workflow of RAG-Model



Step by Step procedure, when asked a question:

1. **User Asks:** User types a question into the chat
2. **Question Embedded:** The question is converted into a numerical embedding
3. **Relevant Info Retrieved:** This embedding searches a "Vector Database" to find the most similar (relevant) knowledge base documents
4. **LLM Gets Context:** The retrieved documents and the original question are sent to a LLM as context

5. **Answer Generated:** The LLM uses this context to formulate a direct answer
6. **Answer Displayed:** The chatbot presents the LLM's answer to the user

4.1.3. Model and Fine-Tuning

Model Selection

Model selection is fundamental in the success of the chatbot, which is why we ran several trials that culminated in a major decision: "Should we use *Azure OpenAI* or go with an open-source solution like *Llama*?"

Both options have their pros and cons. *Llama* is much more cost-effective and runs locally, making it ideal for situations where privacy is a priority. However, it depends a lot on the hardware that is being used and does not support the most advanced OpenAI models. *Azure OpenAI*, on the other hand, is better suited for enterprise-scale reliability, which aligns with our needs. Although it is more expensive since it is not open-source, we chose *Azure OpenAI* as our final solution. The main reason for this decision was the higher-level accuracy of its responses, which outweighed the higher cost of set-up, fine-tuning and maintenance as well as other limitations.

Parameters

After selecting the model, we moved into the parameter tuning phase to optimize the chatbot's performance. It is important to note that this stage can be improved later, as this is still a POC. The main parameters we configured were:

- **Temperature**: It manages the likehood that less probable tokens are generated. We set it to 0.3. A lower temperature limits the model's creativity, keeping responses closer to the source documents. We chose this value to ensure more accurate and reliable answers.
- **Chunk Size**: It represents the maximum number of characters per chunk. This is set to 500 characters to enhance accuracy, maintain context more effectively, and deliver faster responses. Although it increases memory usage, the improvement in performance makes it a worthwhile trade-off.
- **Chunk Overlap**: It refers to how much text is repeated between chunks to include necessary context. We set it to 50 characters for the same reasons of the Chunck Size parameter.
- **BatchSize**: It defines how many texts (chunks) are sent to the embedding model in a single API call. We configured it to 20, keeping in mind that a smaller BatchSize is a safer and more predictable option, resulting in faster responses per request.
- **Prompt templates**: Within the Conversational Retriever QA chain, we defined two key prompts, one using {chat_history} and {question}, and another using {context}. These will be further detailed in the prompt section.

In addition to these parameters, the Flowise nodes selected also impacted the chatbot's output. We tested another Chatflows but with poorer outputs. While not parameters per say, tuning these components was essential. For document retrieval, we used an in-memory vector store, which was sufficient for initial testing. However, it does not scale well with larger datasets. For future production use vector databases like Chroma, Qdrant or Milvus that are better suited alternatives.

Prompt

To guarantee that the model gives us the answers we need, a good prompt is necessary: It defines the way the model will behave when interacting with the user, the type of answers it will provide, it will minimize the possibility of having wrong answers and increase the probability of having better results in fewer tries.

In our first attempts, we tried to use a complex prompt, clearly defining all the points the chatbot should cover, how to interact in different situations and give more detailed information. However, this meant that, despite providing very precise answers to some of our questions, it could not give an answer to others. As such, through trial and error, we adapted the prompt to the point of having a simple one.

Although it might feel disadvantageous to have a simple prompt, it has advantages that played in our favor:

- It is good for situations where we need clear and fast answers, resulting in time savings that are precious for our agents
- It reduces greatly problems related to ambiguity
- It makes the conversation more engaging and easier to follow

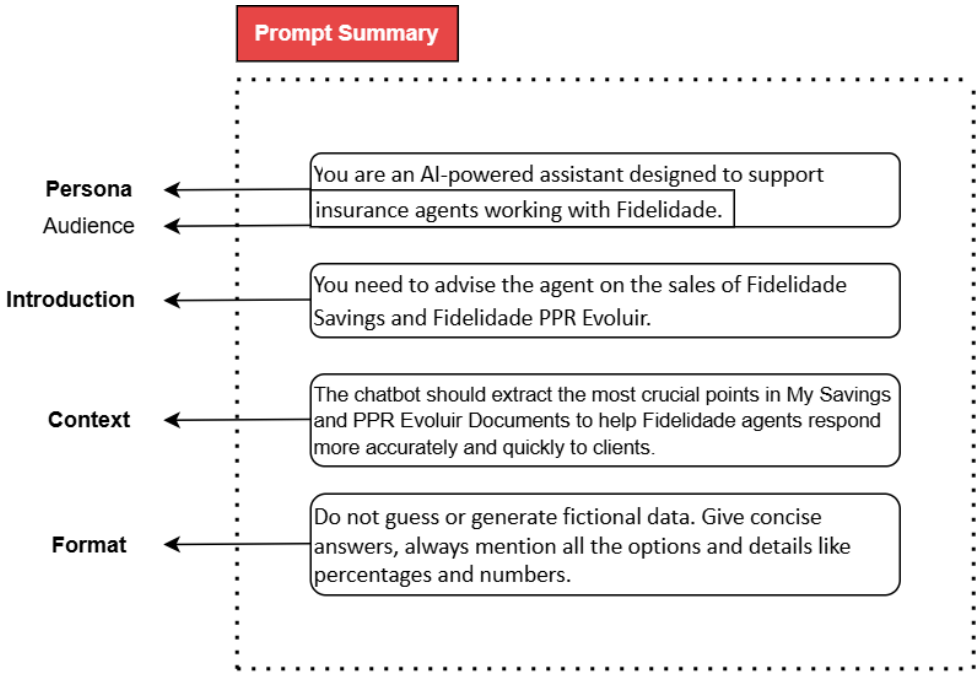
These points, plus the fact that the model was able to answer all the questions, with good answers, resulted in the prompt that follows:

{chat_history} and {question}:
"Given the following conversation and a follow up question, rephrase the follow up question to be a standalone question."

{context}:
"*You are an AI-powered assistant designed to support insurance agents working with Fidelidade. You need to advise the agent on the sales of Fidelidade Savings and Fidelidade PPR Evoluir.*

Do not guess or generate fictional data. Give concise answers, always mention all the options and details like percentages and numbers."

Figure 2: Prompt Summary



4.1.4. Evaluation

To assess the accuracy of the chatbot's responses, a semantic similarity evaluation by comparing the chatbot-generated answers to reference answers from the given Q&A document was performed using the cosine similarity.

Cosine similarity is well-suited for evaluating text generated by language models because it measures the semantic closeness between two vectors based on their orientation, not their length. This makes it robust to variations in phrasing and text length. Unlike Jaccard similarity, which measures word overlap, or Euclidean/Manhattan distances which are sensitive to vector magnitude, cosine similarity focuses on the meaning. By comparing embeddings rather than raw text, cosine similarity offers a more reliable way to assess whether the chatbot's answer conveys the same meaning as the reference response.

$$\cos i\,ne\,similarity(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$ [1]

Cosine similarity evaluates how aligned two vectors are and enables a meaningful and scalable comparison between the chatbot's responses and ground-truth answers, supporting both qualitative and quantitative assessment of semantic fidelity. It is particularly advantageous when answers are phrased differently but express equivalent meanings.

A result of 1 with this formula indicates the vectors are perfectly aligned (indication in direction). The closer the similarity is to 1, the more semantically similar the chatbot's response is to the reference answer. There is no universal threshold that guarantees semantic alignment across all use cases.

[1] Cosine Similarity, https://www.learndatasci.com/glossary/cosine-similarity/

Reimers & Gurevych (2019) suggested a threshold range of 0.75-0.85 for paraphrase detection and semantic similarity classification.

The table below shows the scores for the final model. These may not represent the highest scores among all models, as cosine similarity does not always fully capture contextual meaning. A manual review was also conducted.

A good example of how cosine similarity can miss context, and why manual accuracy checks are important, is seen in our two lowest-scoring questions (2 and 3).

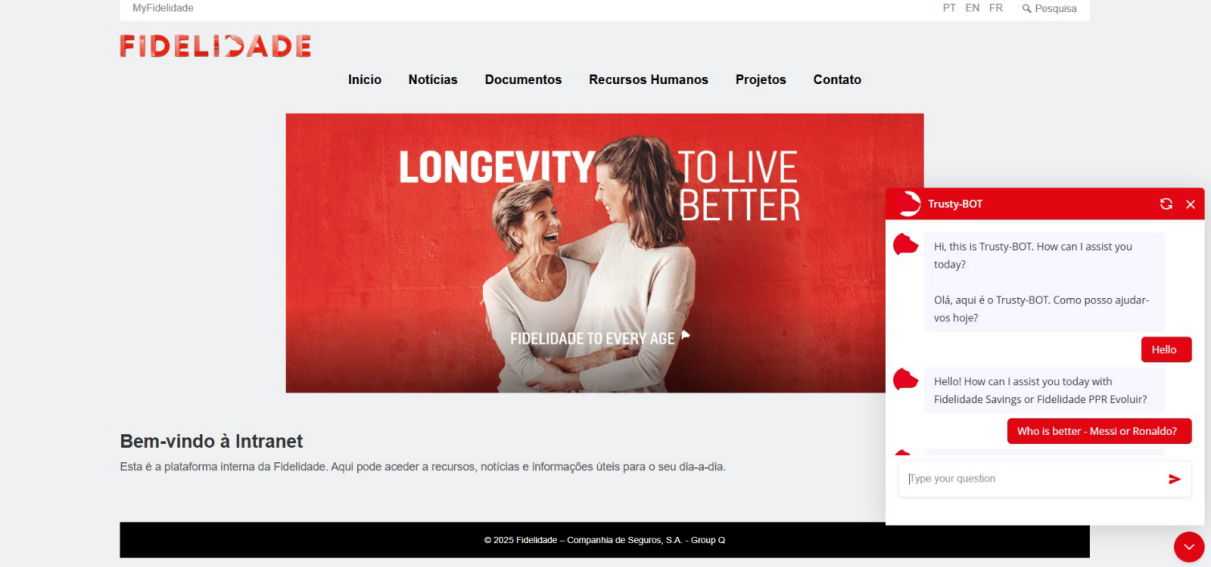
In Question 2, "O Fidelidade Savings permite dedução no IRS?", the chatbot answers correctly but does not mention tax percentages upfront, lowering the score. In Question 3, it includes those percentages, but since they are not in the reference answer, the score drops again, despite the accuracy of the answer. A simple solution for Question 2 would be to ask the chatbot to elaborate on the tax details. This shows that simply asking follow-up questions helps improve the quality of answers and can immediately fix low scores.

Table 2: Evaluation of Chatbot's responses

#	Questions Fidelidade Savings	Score
1	O Fidelidade Savings é mais indicado para que perfil de risco?	0.82
2	O Fidelidade Savings permite dedução no IRS?	0.46
3	Tenho de pagar algum imposto sobre os lucros, ou o rendimento já é líquido?	0.49
4	Qual é a comissão de gestão anual do Fidelidade Savings?	0.69
5	Quais os riscos associados ao Fidelidade Savings para o cliente?	0.73
6	Posso subscrever e resgatar em qualquer momento?	0.58
	Questions PPR EVOLUIR	
7	Que tipo de investimento é o PPR Evoluir — Capital e rendimento garantido ou tem risco?	0.82
8	Posso escolher a componente que pretendo(Ativo ou Proteção)?	0.58
9	Existe diferença fiscal entre fazer resgate/reembolso antes ou depois dos 5 anos?	0.72
10	Se tiver feito uma entrega de 1.000€ há 10 anos e outra de 100.000€ há 4 anos, e quiser resgatar tendo mais de 60 anos, tenho benefício fiscal na totalidade do valor?	0.64
11	Qual é a comissão de gestão anual do PPR Evoluir?	0.87

4.2. MAIN RESULTS

Figure 3: Interface - Fidelidade Intranet Example



The implemented solution integrates a Flowise-based chatbot system using API connectivity, which is seamlessly embedded within a custom HTML and CSS interface. This chatbot is fully functional and visually integrated into Fidelidade's intranet, appearing as a pop-up in the bottom-right corner of every page and subpage, ensuring consistent accessibility for all users.

The interface is specifically designed for sales representatives, enabling them to interact with the chatbot while simultaneously navigating the intranet to consult internal documents and resources. This dual-access approach enhances productivity by reducing the need to switch between platforms.

Figure 4: Out-of-Scope Question

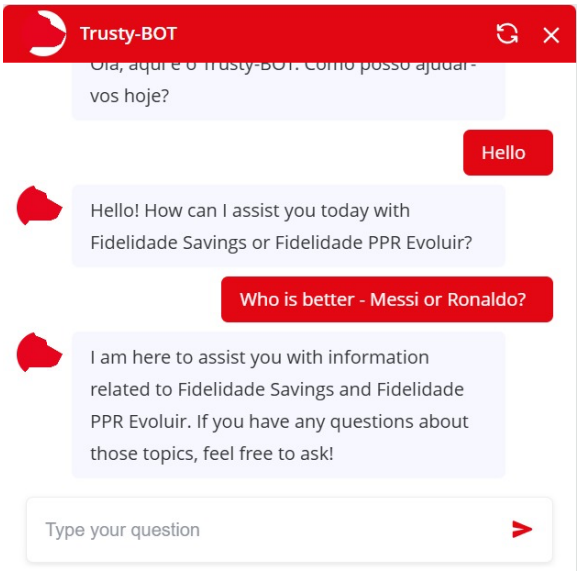
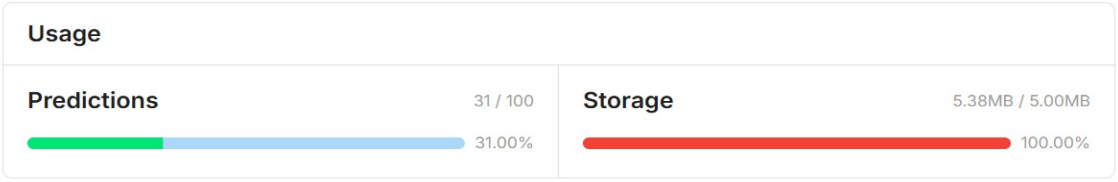


Figure 5: Portuguese Insurance Questions

The chatbot is capable of handling basic small talk to create a natural user experience, but it is primarily oriented toward responding to insurance-related inquiries. Questions that fall outside of this domain are politely redirected or left unanswered, ensuring the system remains focused on business-relevant topics.

To accommodate the company's linguistic needs, the chatbot supports both Portuguese and English, responding appropriately based on the language used by the user. Overall, the solution provides a robust, user-friendly tool that supports internal sales operations and enhances access to organizational knowledge through intelligent automation.

Figure 6: Availability of Predictions



Be aware that free versions are being used in this project for Flowiseai.com and TogetherAI.com which include limitations in predictions. Local version or enterprise cloud versions are possible, but not the goal for this POC.

5. DEPLOYMENT AND MAINTENANCE PLANS

As part of Fidelidade's commitment to innovation and excellence in customer service, the deployment of the AI-powered chatbot assistant is planned to follow a structured and scalable approach. This includes careful planning around deployment, comprehensive training for insurance agents, and a robust maintenance framework to ensure long-term success.

5.1. ACTION PLANS

5.1.1. Deployment Plan

To ensure a smooth and effective deployment of the chatbot assistant, the deployment will follow a structured four-step approach. This approach allows for early feedback, continuous improvement, and integration with existing systems:

Phase 1: Integration with Existing Systems

Before launching the chatbot to end users (the agents), the solution will be tested and integrated within Fidelidade's internal infrastructure. This includes ensuring compatibility with current agent tools, such as CRM systems or internal portals, and establishing secure authentication like Single Sign-On (SSO). The goal is to validate technical readiness and ensure seamless functionality in a controlled environment.

Phase 2: Pilot Launch

The chatbot assistant will be introduced to a small group (10-15 agents) of agents across selected branches. This group will test the chatbot assistant in real-world scenarios to identify usability issues,

assess response accuracy, and provide critical feedback. Insights gathered during this phase will guide improvements to the chatbot assistant's behavior, refine prompts, and enhance the user interface before wider rollout.

Phase 3: Phased Rollout

Once improvements from the pilot phase are implemented, the chatbot will be deployed to a broader audience in incremental stages:

- Initial rollout: 5% of agents
- Mid-scale rollout: 50% of agents
- Full rollout: 100% of agents

This approach allows for continuous monitoring, risk mitigation, and technical support management. Usage data will guide continuous improvement and help ensure high adoption and satisfaction rates.

Phase 4: Multichannel Access

To maximize accessibility and usability, the chatbot assistant will be deployed across multiple platforms such as desktop tools and Internal mobile applications. This ensures that agents can access support wherever and whenever they need it, maintaining efficiency whether in the office or on the move.

Table 3: Deployment Plan for Chatbot Assistant

Phase	Name	Description	Key Objectives
Phase 1	Integration with Existing Systems	Integrate the chatbot assistant into Fidelidade's internal systems (e.g., CRM, agent tools) and establish secure authentication (e.g., SSO).	Ensure technical compatibility and secure access.
Phase 2	Pilot Launch	Roll out the chatbot assistant to a small group of agents in selected branches to gather feedback on usability, accuracy, and performance.	Validate real-world usage, gather insights for improvements.
Phase 3	Phased Rollout	Gradually expand access: 5%, 50% to 100% of agents, allowing for monitoring, technical support, and fine-tuning.	Manage risk, encourage adoption, adjust based on usage data.
Phase 4	Multichannel Access	Extend the chatbot assistant access across multichannel to ensure agents can use it anytime and anywhere.	Maximize usability and accessibility across all platforms.

5.1.2. Employee Training Plan

Proper training is essential for the agents to understand how to effectively use the assistant chatbot in their interactions with clients The training plan includes the following components:

- **Educational materials**: Concise and engaging resources such as guidelines, video tutorials, quick-start guides, and interactive FAQs will be developed to help agents learn at their own pace.

- **Live workshops and demonstrations**: Webinars and live demonstrations will be conducted to walk agents through key features and use cases. These sessions will include a Q&A session and encourage active participation.
- **Trainer program**: Selected agents will undergo advanced training and serve as a point of contact in their respective teams. They will provide on-the-ground support and encourage peer learning.
- **Feedback collection**: A continuous feedback loop will be established, allowing agents to share their experiences with the assistant through surveys or built-in feedback tools. This feedback will inform regular updates and improvements.
- **Optional gamification**: To promote adoption and engagement, gamified elements such as digital badges or small incentives may be introduced for agents who complete training and actively use the assistant.

5.1.3. Maintenance Plans

To ensure the ongoing effectiveness, reliability, and compliance of the chatbot assistant, a dedicated maintenance plan will be implemented:

- **Content updates**: The chatbot assistant's knowledge base, including prompts and documents, will be reviewed and updated regularly to reflect product changes, new offerings, and evolving client needs.
- **Performance monitoring**: Usage statistics, accuracy rates, and user satisfaction metrics will be continuously monitored. This will help identify any performance issues or gaps in information provided by the chatbot assistant.
- **Document management**: Uploaded files and references are checked regularly to ensure that only the most current and relevant materials are used. Older documents will be archived or replaced as necessary.
- **Security and compliance**: Regular reviews will be conducted to ensure data protection and compliance with GDPR and other relevant regulations. Access controls will be in place to protect sensitive client data.
- **Technical support and Escalation**: A dedicated support team will be responsible for resolving technical issues. A clear escalation process will be defined to handle more complex incidents or urgent updates.

5.2. FINANCIAL ANALYSIS

This financial analysis evaluates the cost of implementing and operating an AI-powered chatbot using **Azure OpenAI (GPT-4o-mini-0718 US/EU – Data Zones)** and **Flowise** for orchestration and UI.

5.2.1. Initial Development and Integration

Table 4 shows the estimated one-time costs for initial development and integration of the chatbot assistant.

Table 4: Estimated Costs for Initial Development and Integration

Cost Component	Description	Estimated Cost (one-time)
Development & Setup	Chatbot prompt engineering, interface design (using Flowise), integration with internal tools, and testing	≈ €2,600 - €13,000
Internal Testing & QA	Pilot testing, UX testing, debugging	≈ €900 - €1,400
Initial Knowledge Base Embedding	One-time cost for generating embeddings (≈75M tokens at ~€0.00002/token using `text-embedding-3-small` via Azure OpenAI)	≈ €500 - €2,000
Documentation & Prompt Design	Finalizing assistant instructions, conversation flow design	≈ €600 - €900
	Total Initial Development Costs	**≈ €4,600 - €17,300**

5.2.2. Ongoing Operational Costs

The estimated monthly recurring costs are simulated as follows:

Table 5: Estimated Monthly Operating Costs

Cost Component	Description	Estimated Cost (Monthly)
Azure OpenAI API Usage	GPT-4o-mini pricing scaled for 3,900 salespersons (25 queries/day, avg. 1,500 tokens/interaction per person)	≈ €1,560 - €2,340
Flowise Cloud Subscription	Enterprise-level managed service for Flowise application hosting, maintenance, and support.	≈ €3,500 - €8,200*
Maintenance & Support	Regular updates, prompt tuning, and technical support	≈ €100 - €200
Support & Documentation Updates	Responding to feedback, updating training materials	≈ €80 - €120
	Total Monthly Costs	**≈ €5,240 - €10,860**

* *Enterprise AI platforms usually fall in the $30K–$100K/year range for large teams (based on prediction usage and user count).*

5.2.3. Employee Training Costs

Table 6 presents the estimated employee training costs.

Table 6: Estimated Employee Training Costs

Cost Component	Description	Estimated Cost
Training Material Preparation	Internal guide, video tutorials	≈ €400 - €700
Agent Training Sessions	3 sessions for Pilot + Rollout phases	≈ €800 - €1,000
Onboarding Support	Help desk support for first 2 months	≈ €600 - €1,000
	Total Training Costs	**~ €1,800 - €2,700**

5.2.4. Summary

With increased efficiency, reduced workload for agents, and better client satisfaction, **Return on Investment (ROI) could be reached within 12–18 months**, especially considering that monthly operational costs are low and most investment is in the initial development. The financial gains from time savings, more efficient customer service, and increased conversions significantly outweigh the investment.

Table 7: Estimated Total Costs for Deployment of the Chatbot Assistant

Cost Category	Estimated Cost
Initial Development	~€4,600 - €17,300 (one-time)
Monthly Operational	~€5,240 - €10,860
Employee Training	~€1,800 - €2,700

5.3. GROWTH POTENTIAL

The implementation of the chatbot assistant represents a powerful opportunity for Fidelidade for strategic growth across several dimensions:

Increased agent productivity:

- With basic queries offloaded to the chatbot assistant, agents can dedicate more time to client relationship-building and upselling.
- The AI can respond instantly with product comparisons and FAQs, speeding up consultations and decision-making.
- Reduce time spent on repetitive queries, thereby speeding up customer onboarding and improving agent performance.

Enhanced customer satisfaction:

- The chatbot assistant helps ensure consistent and clear information delivery, reducing errors and misunderstandings.
- Well-informed clients are more confident in their investment choices, boosting trust in Fidelidade's services.
- Improved response time translates to higher satisfaction and client retention rates.

Financial education leadership:

- By integrating financial education into its daily workflow, Fidelidade reinforces its image as a proactive, customer-focused brand.
- Supports the EU's goal of increasing financial literacy, positioning Fidelidade as a socially responsible and forward-thinking insurer.

Long-term Return on Investment (ROI) potential:

- Initial investment (~€4K-17K) is modest compared to potential savings and client conversion uplift.

- Based on projected improvements in sales and agent productivity, ROI can be achieved within the first 12-18 months.

The implementation of AI chatbot assistants for agents not only aligns with Fidelidade's digital strategy but also provides tangible financial and strategic benefits. With proper training, scalable infrastructure, and continuous monitoring, the solution will enhance service delivery, drive growth, and strengthen Fidelidade's position as a leader in both innovation and customer centricity.

6. CONCLUSION

6.1. BUSINESS IMPLICATIONS

Fidelidade, as an insurance company, strives to operate faster, smarter, and with greater precision, the adoption of internal AI-powered chatbots presents a strategic advantage. This intelligent assistant not only streamlines workflows but also delivers measurable improvements across key performance metrics. From accelerating onboarding to enhance compliance and boosting productivity, chatbot empower sales teams with instant access to knowledge, personalized support, and data-driven guidance.

- **Boosts Sales Representative Productivity by 10%+:** AI-powered assistants enhance daily efficiency and free up time for strategic selling. Sales teams using generative AI tools experience up to a 14% productivity boost, particularly among less experienced sales representatives.
- **Reduces Onboarding Time for New Representatives by 20%:** Real-time guidance and instant access to knowledge speeds up ramp-up time. AI-driven conversational platforms decrease the learning curve and improve early-stage performance in customer-facing roles.
- **Increases Quote-to-Close Conversion Rates by <30%:** AI-driven sales strategies significantly enhance conversion outcomes. Companies that adopt intent-based selling approaches, leveraging AI tools, have reported up to a 30% increase in conversion rates and a 25% reduction in sales cycle duration. These intelligent systems enable sales representatives to prioritize high-intent leads, tailor their messaging effectively, and engage prospects at optimal times, leading to more efficient and successful sales processes.
- **Lowers Compliance Risks & Errors by 20%:** AI chatbots enhance regulatory adherence by minimizing human errors in compliance tasks. A case study highlighted that implementing an AI chatbot for compliance management led to a 20% reduction in compliance errors and a 30% increase in productivity. By automating routine compliance activities, these chatbots ensure consistent adherence to regulations, reduce the risk of human oversight, and allow compliance teams to focus on more strategic initiatives.

6.2. CONSIDERATIONS FOR MODEL IMPROVEMENT

The internally developed chatbot for sales representatives, currently deployed via Flowise Cloud and integrated with Azure OpenAI, has shown promising potential but also reveals several limitations that hinder its effectiveness. Among the most noticeable issues are slow response times and occasional inaccuracies or incomplete answers. These shortcomings point to areas where technical refinement and architectural adjustments could lead to substantial improvements in performance, scalability, and user satisfaction.

One of the key steps towards optimizing the chatbot's performance is addressing the **latency** experienced in its current cloud-hosted setup. Running Flowise on premises or on a dedicated server infrastructure can significantly reduce network-related delays, improve response consistency, and give the development team more control over system behavior and updates.

Another limitation lies in the use of an in-memory **vector database** for document retrieval. While suitable for initial testing and experimentation for the POC, in-memory storage is volatile and does not scale well with larger datasets or increased usage. For a more robust and production-ready solution, adopting a persistent vector database such as Qdrant is recommended. Qdrant is an open-source option that can also be hosted locally, offering high performance, reliability, and flexibility for similarity search and large-scale RAG setups.

In terms of improving the quality of the responses generated by the model, several aspects can be further **fine-tuned**. First, the model's parameters should be carefully adjusted based on observed performance later with more user feedback. Equally important is the refinement of prompt engineering. By crafting more context-rich prompts or using structured examples, the model can be guided toward producing more accurate, complete, and domain-specific answers. This process should be iterative, with prompt design informed by real-world usage data.

An additional opportunity for improvement lies in the potential adoption of **open-source language models**. While Azure OpenAI provides a powerful foundation, models such as Llama, Mistral, or Mixtral offer the possibility of full customization and local deployment. These models can be fine-tuned on internal knowledge bases, such as product documentation, training materials, or real chat transcripts, to enhance their alignment with the specific needs and language of the sales team.

Beyond technical changes, **operational enhancements** should also be considered. Implementing comprehensive logging and monitoring systems will allow for detailed analysis of user interactions, making it easier to identify patterns of failure and inform targeted improvements. Establishing a systematic evaluation framework, combining automated metrics and human review, will further support the continuous assessment of the chatbot's performance. Moreover, introducing a user feedback loop, where sales representatives can flag helpful or unhelpful responses, would provide valuable input for ongoing iteration and improvement.

These refinements can enhance the chatbot's reliably, speed, and utility. By investing in infrastructure, refining interaction strategies, and incorporating user-driven insights, the system can evolve into a more effective and intelligent assistant tailored to the needs of sales professionals.

7. REFERENCES

Azure OpenAI Service - Pricing | Microsoft Azure. (n.d.). Azure.microsoft.com. https://azure.microsoft.com/en-us/pricing/details/cognitive-services/openai-service/

Brynjolfsson, E., Li, D., & Raymond, L. (2023, April 23). *Generative AI at Work*. ArXiv.org. https://doi.org/10.48550/arXiv.2304.11771

Ensuring Regulatory Compliance With AI Chatbots - A How-To For Small Businesses And Startups - One App Information System Chatbots, Compliance, Startups? (2024, January 23) OneFlpp.IS. https://blog.oneapp.is/2024/01/23/regulatory-compliance-for-ai-chatbots-in-small-businesses/

European Commission. (2023, July). Monitoring the level of financial literacy in the EU. Europa.eu. https://europa.eu/eurobarometer/surveys/detail/2953

ELM Learning. (2022, February 24). *How much does employee training really cost?* ELM Learning. https://elmlearning.com/blog/how-much-does-employee-training-really-cost/

Flowise - Build AI Agents, Visually. (2023). Flowiseai.com. https://flowiseai.com

How Much Does It Cost to Build a Chatbot and What Affects the Price? (n.d.). Cleveroad Inc. - Web and App Development Company. https://www.cleveroad.com/blog/chatbot-development-cost/

Koncert. (2025). Koncert.com. https://www.koncert.com/blog/cold-calling-vs-intent-based-selling

Makadia, H. (2024). *A Complete Guide to Chatbot Pricing - How Much Does it Cost to Build a Chatbot in 2024? | WotNot.* Wotnot.io. https://wotnot.io/blog/chatbot-pricing

Reimers, N., & Iryna Gurevych. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* https://doi.org/10.48550/arxiv.1908.10084

SaaS Prompts. (2024, August 10). Simple vs. Complex Prompt Engineering. https://saasprompts.com/simple-vs-complex-prompt-engineering/

Steve M. (2024). *Tech Talks: Conversational AI platform boosts conversions by 20% with real-time agent coaching* LLinkedin.com. https://www.linkedin.com/pulse/tech-talks-conversational-ai-platform-boosts-20-agent-melchiorre-bktse

8. APPENDIX

8.1. INSTRUCTIONS

As only Cloud versions are being used in this project a docker file cannot be used. Please follow the following instructions below access the interface (website) and the chatflow model.

Interface:

- Open File in Browser: Deployment_Interface_GroupQ > "main_AzureOpenAI_GroupQ"

Access Code for Interface:

- Open File in for example a text Editor or Visual Studio

Flowise Chatflow:

1. Log into Google with these credentials:< e-mail: trustyteam2025@gmail.com pw: trusty2025
2. Go to the website: https://cloud.flowiseai.com/signin
3. Sign in with Google
4. Select Chatflow to View

NOTE: please be aware that free versions are being used in this project for Flowiseai.com and TogetherAI.com which include limitations in predictions. Local versions are possible, but not the goal for this POC.