TricIA: A Multi-Agent System to Facilitate Interaction with Business Intelligence Dashboards

Bruno Raviolo[1], **Joan Lucas M. S. Chaves**[1], **Raimir Holanda Filho**[1], **Carlos Caminha**[1,2]

[1]Programa de Pós-Graduação em Informática Aplicada - PPGIA -UNIFOR
P.O. Box 1258 – 60811-341 – Fortaleza – CE – Brasil

[2]Kunumi Lab - Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil

Abstract. *This paper presents TricIA, a multi-agent system based on an open-source language model with 12 billion parameters, designed to enhance the usability of the Monitor Fiscal — a business intelligence platform from the Court of Accounts of the State of Ceará. TricIA acts as a chatbot that assists users in navigating and understanding the available data, both through natural language responses and by performing direct actions on the interface, such as page redirections and filter applications. To ensure efficiency even on modest infrastructures, an architecture composed of three specialized agents was adopted: the orchestrator, the guardrail, and the function-calling agent. This structure distributes responsibilities and allows smaller models to overcome capacity limitations. The paper describes the architecture, integration techniques with the dashboards, and presents a functional demonstration of the system, highlighting its potential to democratize access to complex fiscal information.*

1. Introduction

Business Intelligence (BI) dashboards have established themselves as powerful tools for data analysis by offering interactive and comprehensive visualizations of large volumes of information. However, as these platforms grow in complexity—with multiple visualizations, filters, and thematic panels—many users, especially those unfamiliar with the underlying databases, struggle to fully explore the available information. This situation undermines the potential impact of such tools on transparency and the social oversight of public policies.

The *Monitor Fiscal*, developed by the Court of Accounts of the State of Ceará (TCE/CE), is a prominent example of this challenge. The platform consolidates public and audited financial data from both state and municipal governments, organized into several thematic dashboards, aiming to anticipate potential fiscal deterioration in government entities [Tribunal de Contas do Estado do Ceará 2025]. However, the wealth of available data often goes underutilized by users unfamiliar with the structure of the dashboards or the platform's underlying logic.

With the recent advances in Large Language Models (LLMs), new opportunities have emerged to address this issue. LLM-based chatbots have demonstrated capabilities in understanding natural language, executing contextual reasoning, and even invoking external functions, expanding their role beyond reactive interactions [Qin et al. 2024, Tran et al. 2025]. Successful applications include time series forecasting [Bastos et al. 2025], information extraction from digitized documents

[Almeida and Caminha 2024], and synthetic data generation [Karl et al. 2024], which highlights their potential in complex and specialized contexts.

Nevertheless, adopting LLMs in sensitive environments, such as public institutions, requires caution. Concerns regarding data privacy, high operational costs, and reliance on proprietary models—whose training processes and usage criteria are not always transparent—motivate the search for alternatives based on open-source models. Although open-source LLMs typically demand robust computational infrastructure, compression techniques such as quantization and low-rank adaptation have made it feasible to run them in more modest environments.

In this work, we present TricIA, a virtual assistant developed using an open-source language model with only 12 billion parameters, designed to operate as a chatbot within the Monitor Fiscal. To overcome the limitations imposed by the model's relatively small capacity—which would render a monolithic approach with long prompts and multiple instructions unfeasible—we adopted a multi-agent architecture. This divide and conquer strategy distributes complexity among specialized agents, optimizing system performance.

The system consists of three main agents: (i) the *Orchestrator* Agent, responsible for direct interaction with the user; (ii) the *Guardrail* Agent, tasked with keeping conversations within the Monitor Fiscal scope by rejecting queries unrelated to the domain; and (iii) the *Function Caller* Agent, which interacts with the platform environment by redirecting navigation to specific pages and applying filters when necessary. Together, these agents collaborate to ensure accessibility, accuracy, and the usefulness of the information available on Monitor Fiscal, democratizing access to public financial analysis through an intelligent and efficient conversational interface.

2. TricIA

TricIA is a multi-agent system (MAS) designed for question-and-answer interactions and function calling, supported by the Gemma3 language model as the reasoning engine for its agents. The choice of this model is based on the fact that it is open-source, ensuring that the system can be implemented on local infrastructure, thereby providing greater control over data, system usage, and operational costs.

In the architecture design presented in Figure 1, three specialized agents perform predefined roles in a cooperative relationship. User messages are initially evaluated by the *Guardrail* agent, which ensures the responsible use of the AI system by verifying whether the user's input falls within established limits, thereby preventing system misuse [Lu et al. 2024]. Next, the *Orchestrator* agent is activated to determine whether the user's request should be forwarded to a specialized agent, acting as a coordinator [Lu et al. 2024], or if it can directly generate a text response to be returned to the user. When the user requests site navigation assistance from TricIA, the *Function Caller* agent interprets the request and, using introspective reasoning, selects the appropriate tools to control the dashboard's visualization.

2.1. Agents

In order for each agent to perform a specialized role within a common system, different techniques are applied, along with specific system prompts. Techniques shared by all agents

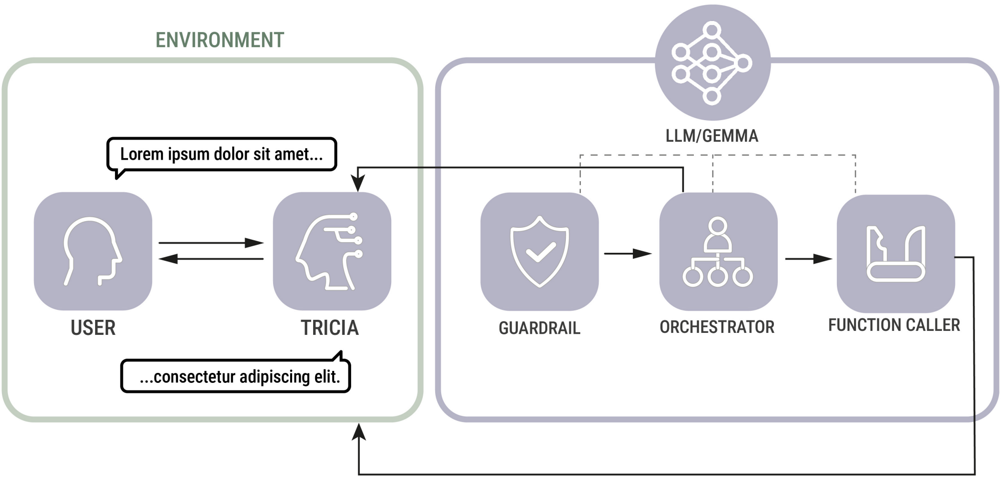
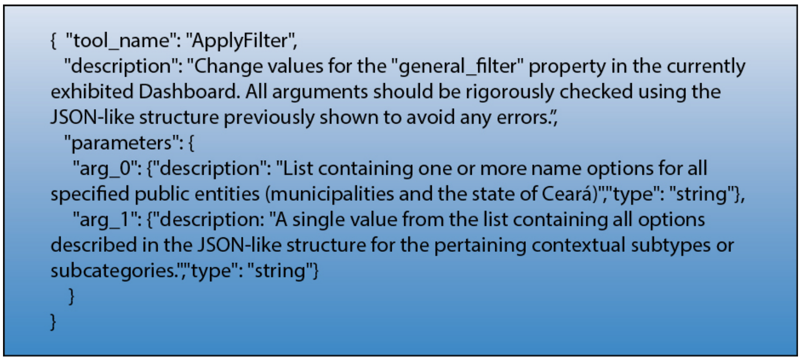


Figura 1. TricIA's MAS architecture diagram

include few-shot examples. To inform the *Orchestrator* and *Function Caller* agents about the structure of the Monitor Fiscal, a mapping of its components is provided, formatted in a JSON-like schema, identifying sections, available data types, descriptions, and applicable filters. To classify messages as either allowed or prohibited, the *Guardrail* agent uses a blacklist containing types of forbidden actions directly embedded in its prompt, returning a structured response to the system's backend.

2.2. Tools

A set of tools was developed to enable the *Function Caller* agent to operate within the *Monitor Fiscal* environment, providing the ability to redirect the user through the dashboards of different sections, as well as modify the properties of the objects contained within them, such as chart types and specific filters. Upon receiving a command from the *Orchestrator*, this agent interprets the request and performs function calls, passing the necessary arguments to each tool. To ensure the agent can correctly use these functions, its prompt includes a structured description of the three different tools, along with an explanation of their purpose and the required arguments, with an example shown in Figure 2.



```
{  "tool_name": "ApplyFilter",
   "description": "Change values for the "general_filter" property in the currently
exhibited Dashboard. All arguments should be rigorously checked using the
JSON-like structure previously shown to avoid any errors.",
   "parameters": {
     "arg_0": {"description": "List containing one or more name options for all
specified public entities (municipalities and the state of Ceará)","type": "string"},
     "arg_1": {"description: "A single value from the list containing all options
described in the JSON-like structure for the pertaining contextual subtypes or
subcategories.","type": "string"}
    }
}
```

Figura 2. Example of a tool description for *Function Caller* agent

The task of performing function calls is considered complex, as it involves the ability to decompose a problem into subcomponents, identify the appropriate tools in the correct sequence, and define the relevant arguments based on their descriptions provided in its prompt, while simultaneously interpreting the JSON structure — without access to intermediate results [Basu et al. 2025, Qin et al. 2024].

3. Tool Demonstration

TricIA was implemented in Python 3, along with the libraries Flask, requests, openai, and lmstudio. Initially, the *gemma3-12b-it-qat* model was used within LM Studio, running on an NVIDIA GeForce RTX 3060 12GB GPU. A demonstration video of TricIA[1] and a GitHub repository[2] were made available for this work.

The initial access page of the Monitor Fiscal dashboard presents a set of navigation elements, as shown in Figure 3 (a), including:

1. A vertical sidebar with buttons for all sections of the Monitor Fiscal dashboard, providing data on Revenues, ICMS, Social Security, Expenses, among others;
2. A checkbox filter containing all public entities — municipalities and the state of Ceará;
3. Charts related to the data types of each section being viewed, each with their own specific filters;
4. A dialogue box for interacting with TricIA.

When accessing the Monitor Fiscal main dashboard, user interaction TricIA begins with a dialog window in which it offers assistance. Upon responding, the user's message is initially sent to the *Guardrail* agent, which classifies the message as allowed or not. If allowed, and if the message pertains to a relevant *Monitor Fiscal* topic, the system proceeds to forward it to the *Orchestrator*. For general questions about covered *Monitor Fiscal* topics, such Social Security or Public Debt, the *Orchestrator* can directly generate an answer to explain it using its knowledge about Monitor Fiscal's structure. Although if it interprets the message as a request to locate a specific section, the system then activates the *Function Caller* agent, which makes a function call to change the dashboard view, guiding the user to the specified section.

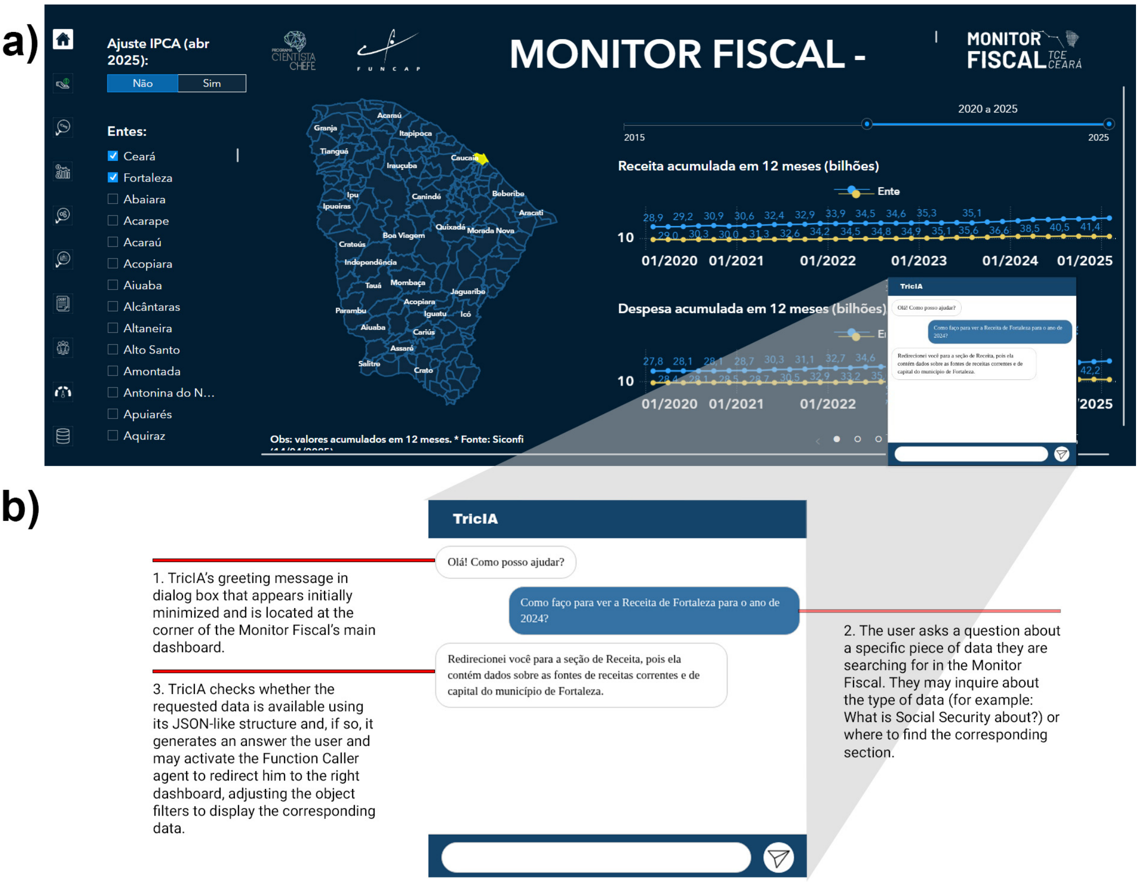
3.1. Experimental Evaluation

The system underwent a preliminary evaluation through unstructured testing with a group of three experts familiar with the Monitor Fiscal platform and with backgrounds in information technology and accounting. The objective was to assess the quality of user interaction with TricIA, functioning as a chatbot, through a series of questions formulated by the experts during the evaluation, covering different levels of difficulty, with each participant posing between 10 and 15 questions.

Its general explanatory capacity about Monitor Fiscal and its different sections was tested through questions regarding, for example, the meaning of acronyms and terminology, as well as the types of information presented in the dashboards, thereby emulating easier questions from the general public. Its ability to solve harder questions from experts was also tested. These questions included conceptual ambiguities or contradictions, as well as queries about how to locate specific data using particular visualization modes.

[1] https://www.youtube.com/watch?v=yT8Ljs7E1eM

[2] https://github.com/JLMSC/TCE_TricIA_SBBD_2025



Ajuste IPCA (abr 2025): [✓ Não] Sim

Entes:
☑ Ceará
☑ Fortaleza
□ Abaiara
□ Acarape
□ Acaraú
□ Acopiara
□ Aiuaba
□ Alcântaras
□ Altaneira
□ Alto Santo
□ Amontada
□ Antonina do N...
□ Apuiarés
□ Aquiraz

Obs: valores acumulados em 12 meses. * Fonte: Siconfi

TricIA

Olá! Como posso ajudar?

Como faço para ver a Receita de Fortaleza para o ano de 2024?

Redirecionei você para a seção de Receita, pois ela contém dados sobre as fontes de receitas correntes e de capital do município de Fortaleza.

1. TricIA's greeting message in dialog box that appears initially minimized and is located at the corner of the Monitor Fiscal's main dashboard.

2. The user asks a question about a specific piece of data they are searching for in the Monitor Fiscal. They may inquire about the type of data (for example: What is Social Security about?) or where to find the corresponding section.

3. TricIA checks whether the requested data is available using its JSON-like structure and, if so, it generates an answer the user and may activate the Function Caller agent to redirect him to the right dashboard, adjusting the object filters to display the corresponding data.

Figura 3. Environment of the Monitor Fiscal with TricIA integration (a), and interaction box between TricIA and users (b).

During the evaluation, compliance with explicit prompt behavioral constraints was monitored, along with the observation of any potential hallucinations.

The collected responses indicated overall satisfactory performance in the task of answering both general and expert-specific questions, as well as explaining concepts related to the data presented in the dashboards. The structured description of Monitor Fiscal in a JSON-like format covering the various dashboards, including all objects, data, filters, and visualization modes, effectively informed the model, resulting in the inference of consistent responses.

4. Conclusion

In this demonstration, we have presented TricIA, a virtual assistant that acts as a chatbot to enhance usability and navigation within the Monitor Fiscal dashboards. This multi-agent system features an architecture designed to address user queries and perform function calls for direct interaction with the dashboards, utilizing small-sized open-source language models compatible with affordable GPUs. This approach ensures low operational costs and greater control over the system, meeting the specific requirements of the TCE/CE as a public institution.

For future work, the implementation of additional agents is planned to incorporate more complex data analysis tasks, thereby expanding TricIA's capabilities; the creation of a specific benchmark for Monitor Fiscal, with a dataset containing questions and sequences of function calls to evaluate the *Function Caller* agent, enabling direct comparison with other open-source language models and measuring the benefits of prompt enginerring techniques such as Tree-of-Thought and Chain-of-Thought to improve agents accuracy.

5. Acknowledgments

The authors thank the Ceará Foundation for Support to Scientific and Technological Development (FUNCAP), which, through its Chief Scientist (Cientista Chefe) program, enabled the development of this research.

Referências

Almeida, F. C. and Caminha, C. (2024). Evaluation of entry-level open-source large language models for information extraction from digitized documents. In *Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)*, pages 25–32. SBC.

Bastos, Z., Freitas, J. D., Franco, J. W., and Caminha, C. (2025). Prompt-driven time series forecasting with large language models. In *Proceedings of the 27th International Conference on Enterprise Information Systems*, pages 309–316.

Basu, K., Abdelaziz, I., Kate, K., Agarwal, M., Crouse, M., Rizk, Y., Bradford, K., Munawar, A., Kumaravel, S., Goyal, S., Wang, X., Lastras, L. A., and Kapanipathi, P. (2025). Nestful: A benchmark for evaluating llms on nested sequences of api calls.

Karl, A. L., Fernandes, G. S., Pires, L. A., Serpa, Y. R., and Caminha, C. (2024). Synthetic ai data pipeline for domain-specific speech-to-text solutions. In *Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL)*, pages 37–47. SBC.

Lu, Q., Zhu, L., Xu, X., Xing, Z., Harrer, S., and Whittle, J. (2024). Towards responsible generative ai: A reference architecture for designing foundation model based agents. In *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, pages 119–126.

Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Zhou, X., Huang, Y., Xiao, C., Han, C., Fung, Y. R., Su, Y., Wang, H., Qian, C., Tian, R., Zhu, K., Liang, S., Shen, X., Xu, B., Zhang, Z., Ye, Y., Li, B., Tang, Z., Yi, J., Zhu, Y., Dai, Z., Yan, L., Cong, X., Lu, Y., Zhao, W., Huang, Y., Yan, J., Han, X., Sun, X., Li, D., Phang, J., Yang, C., Wu, T., Ji, H., Li, G., Liu, Z., and Sun, M. (2024). Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Tran, K.-T., Dao, D., Nguyen, M.-D., Pham, Q.-V., O'Sullivan, B., and Nguyen, H. D. (2025). Multi-agent collaboration mechanisms: A survey of llms.

Tribunal de Contas do Estado do Ceará (2025). Monitor fiscal - tce-ce. `https://monitorfiscal.tce.ce.gov.br/`. Acesso em: 02 jun. 2025.