

# Divination: Applying Retrieval-Augmented Generation with LLMs to generate content for D&D

Luiz Carlos Costa da Silva<sup>1</sup> and Renato Cordeiro Ferreira<sup>1</sup>

<sup>1</sup> Institute of Mathematics and Statistics - University of São Paulo

## Introduction

The Divination project provides an application that uses the fantasy tabletop role-playing game Dungeons and Dragons (D&D) sourcebooks to help DMs manage players' adventures.

The system is divided into a front-end and a back-end. The front-end uses the Vue3js framework to provide a user chat interface, and the back-end uses Python+FASTAPI powered by ChatGPT-4o LLM.

To answer the questions as faithfully as possible, avoiding LLM **Hallucinations**, the back-end uses Retrieval-Augmented Generation (RAG) as a structure; this structure reads external data from the D&D Free Rules (2024) page and stores it on a Vector Store Database. When a query is received in the back-end, the RAG retrieves the most relevant chunks from the vector store and sends them to the rest of the system to use as context to give the most accurate answer to the user [2].

## Motivation & Objectives

The use of LLMs like chatGPT and LLaMMA often exhibits content that diverges from the user's questions or fabricated answers; this behavior is known as hallucinations. This phenomenon can be mitigated using RAG structure to provide better quality and accuracy in AI responses [3].

The Divination project have the following objectives:

- Provide a user chat interface
- Reduce LLM hallucinations by using RAG structure
- Answer D&D questions concisely

## Interface

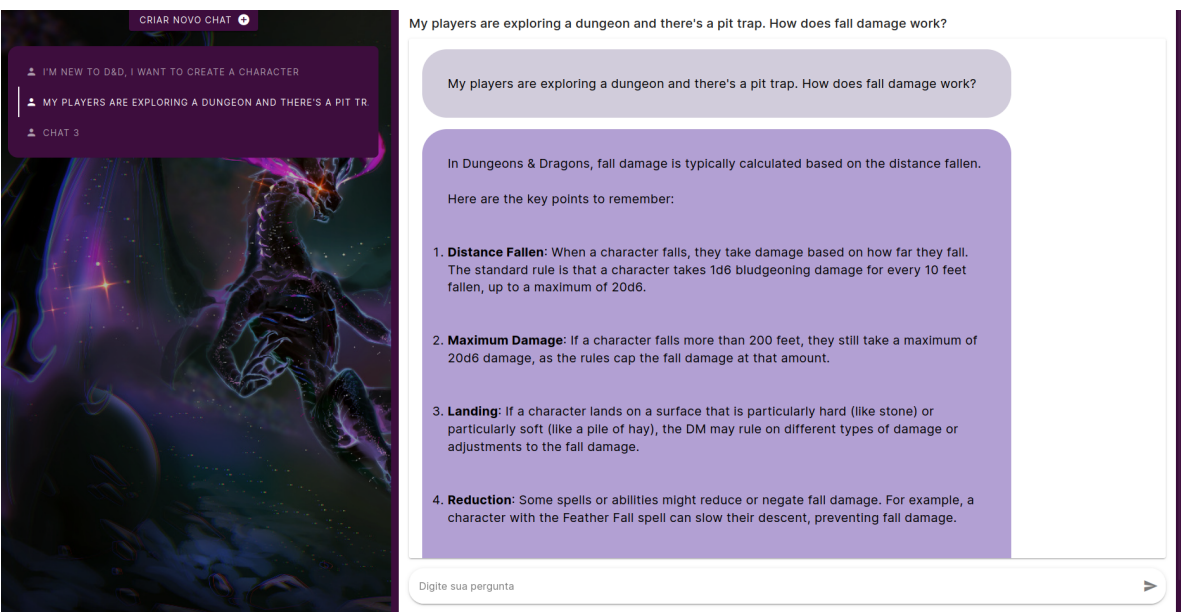


Figure 1: *Web Application Interface*: The interface shows Divination functionalities and a basic user interaction

## Architecture

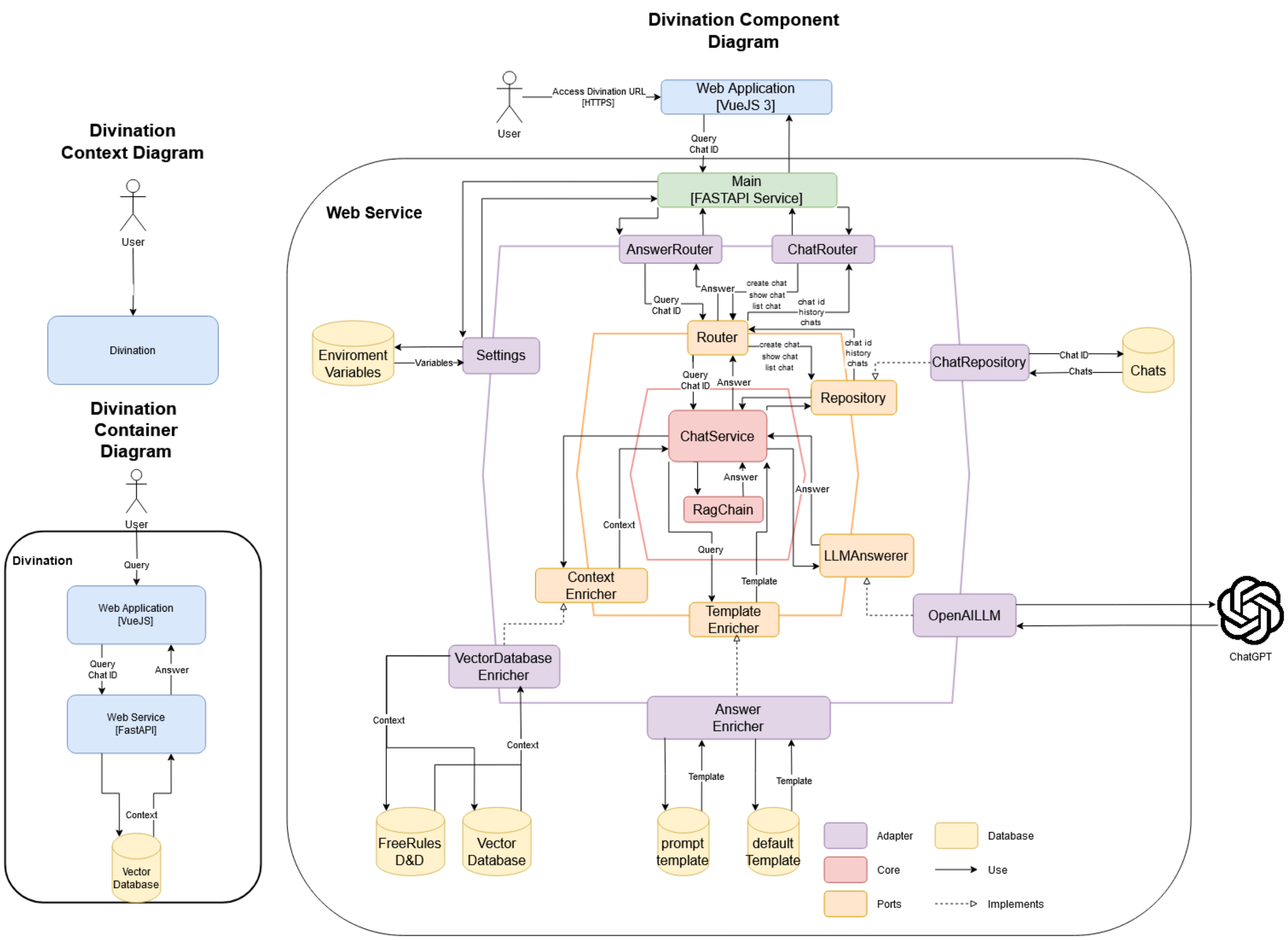


Figure 2: *Software Architecture*: The C4 model representation showing the first 3 levels of details. The **Context Diagram** shows the interaction between the software and the world around it. The **Container Diagram** zooms into the software system and shows the application inside it. The **Component Diagram** zooms into the web service and shows its internal structure, made with a **Hexagonal architecture pattern** [1].

## Personas

Two **Personas** were created to test the project with real users:

- Player
- Dungeon Master

The Player persona receives scenarios around the initial character creation, choosing his race, class, and origin.

The Dungeon Master receives scenarios about the dungeon exploration systems, jump rules, and trap damage calculation.

## Conclusion

After performing tests with the two Personas, the results gave the feedback that the system, compared to standard ChatGPT-4o, gave more concise information and made fewer hallucinations. The interface will be improved for future improvements, considering user feedback and the single-user system will be turned into a multi-user system.

## References

- [1] Alistair Cockburn. *Hexagonal Architecture Explained*. Humans e Technology Inc, 2024.
- [2] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. 2024. arXiv: 2312.10997 [cs.CL]. url: <https://arxiv.org/abs/2312.10997>.
- [3] Yue Zhang et al. *Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models*. 2023. arXiv: 2309.01219 [cs.CL]. url: <https://arxiv.org/abs/2309.01219>.

For further information, see <https://linux.ime.usp.br/~luizcarlosdk/mac0499/> or submit an email to [luizcarlosdk@usp.br](mailto:luizcarlosdk@usp.br)