

# Desenvolvimento de um padrão de linguagem formal para criação automatizada de diagramas de atividade para utilização médico-hospitalar

Emanuel Fagan Bissacotti  
Universidade Franciscana  
Ciência da Computação  
Santa Maria, RS  
Email: e.bissacotti@ufn.edu.br

Evandro Cunha dos Santos Junior  
Universidade Franciscana  
Ciência da Computação  
Santa Maria, RS  
Email: evandro.santos@ufn.edu.br

**Abstract**—Este artigo traz a criação de uma linguagem formal para a compreensão do chat GPT, para que o mesmo retorne um código com sintaxe PlantUML que cria diagrama de atividade para utilização médico-hospitalar, relatando seus testes realizados através do chat GPT, erros encontrados e como a aplicação funcionará.

## 1. Introdução

Este artigo fala sobre o desenvolvimento de uma linguagem formal para criação de diagramas de atividades. O sistema usará a API(Application Programming Interface) do GPT-4 e do PlantUML para a comunicação de forma integrada.

### 1.1. Objetivo

Facilitar os processos médico-hospitalar através de diagramas de atividades geradas automaticamente por um texto padronizado informado pelo usuário.

### 1.2. Estrutura

O artigo está estruturado da seguinte forma: iniciando com o resumo do mesmo, após, sua introdução trazendo seus objetivos, revisão bibliográfica mostrando dois trabalhos correlatos e citando ferramentas que serão utilizadas no seu desenvolvimento, a metodologia de desenvolvimento do artigo e a engenharia do projeto, resultados obtidos de como deve ser estabelecida a linguagem formal para evitar possíveis erros e a conclusão do artigo.

## 2. Revisão Bibliográfica

Esta seção trata exclusivamente de 2 trabalhos correlatos e da apresentação das ferramentas que serão utilizadas neste trabalho, dentre eles: diagrama de atividade, API, GPT-4 e PlantUML. Procurando dar uma breve explicada sobre cada uma delas.

### 2.1. Trabalhos Correlatos

O trabalho "Conceptual Modeling and Large Language Models: Impressions From First Experiments With Chat-GPT", desenvolvido por F. Hans-Georg, ET all. fala sobre a padronização dos retornos do chat GPT baseado no que o mesmo aprendeu à partir de informações dadas por usuários. Tal artigo define que são necessários milhares de dados para a geração de saída.

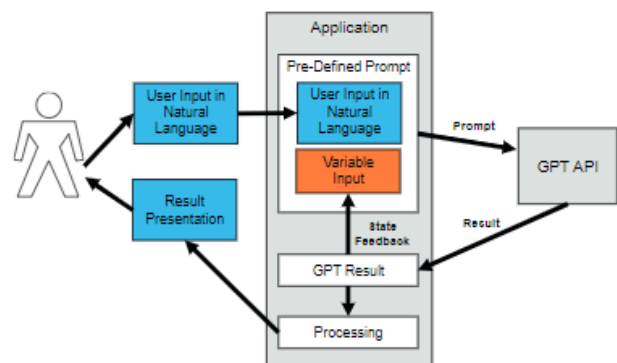


Figura 1 - Exemplo como sistema acima funciona[1].

O trabalho "On Codex Prompt Engineering for OCL Generation: An Empirical Study", desenvolvido por S. Abukhalaf, ET all. discute os avanços recentes em aprendizado profundo e processamento de linguagem natural e como eles levaram ao desenvolvimento de grandes modelos de linguagem destacando-o para a geração de código e explorando o uso desses modelos no contexto da Linguagem de Restrição de Objetos (OCL) para melhorar o desenvolvimento de software orientado a modelos.

### 2.2. Ferramentas

Descrição das ferramentas ou siglas que auxiliaram no projeto.

**2.2.1. Diagrama de atividade.** Um diagrama de atividade fornece uma visualização do comportamento de um fluxo

de atividade descrevendo a sequência de ações em um processo.[3]

**2.2.2. API.** Application Programming Interface que, traduzida para o português, pode ser compreendida como uma interface de programação de aplicação. Ou seja, API é um conjunto de normas que possibilita a comunicação entre plataformas por meio de uma série de padrões e protocolos.[4]

APIs são mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos. Por exemplo, o sistema de software do instituto meteorológico contém dados meteorológicos diários. A aplicação para a previsão do tempo em seu telefone “fala” com esse sistema por meio de APIs e mostra atualizações meteorológicas diárias no telefone.[7]

**2.2.3. GPT-4.** É uma tecnologia avançada de inteligência artificial de conversação desenvolvida pela OpenAI, disponível para assinantes do ChatGPT Plus. É a próxima geração de modelos de linguagem GPT (Transformador pré-treinado generativo) e promete levar as conversas homem-máquina a um nível totalmente novo.[5]

**2.2.4. PlantUML.** PlantUML é uma linguagem e ferramenta de modelagem de código aberto que permite aos usuários criar diagramas a partir de uma linguagem de texto simples.[6]

### 3. Metodologia

Para a criação deste artigo foi utilizado o método SCRUM de desenvolvimento. Onde os processos seguem detalhados:

**Backlog:** Foram realizadas reuniões semanais para desenvolver novas ideias inserindo em uma lista de backlog com prazos para desenvolvimento.

**Sprint backlog:** Para uma maior assertividade no desenvolvimento completo da ideia de backlog, a mesma foi dividida em pequenas tarefas, fazendo com que cada membro do grupo pudesse se responsabilizar por uma parte, cada uma possuindo um prazo pré-determinado.

**Incremento:** À partir da conclusão das pequenas tarefas, as mesmas foram incorporadas como parte de um todo, ou seja, a divisão em pequenas tarefas serviu como incremento da ideia maior de backlog.

**Definição de pronto:** Quando finalizada a ideia de backlog, foi analisado o desempenho de cada integrante em determinadas partes do projeto e se os objetivos foram concluídos da forma programada anteriormente pela equipe.

### 3.1. Engenharia do projeto

Apresentação dos dois diagramas criados para o melhor entendimento do projeto ao solicitante e ajudar o programador na codificação.

**3.1.1. Diagrama de atividade.** Este diagrama representa o comportamento ideal do início ao fim do software, sem que aconteça qualquer imprevisto durante a sua execução, trazendo o que cada agente fará dentro do mesmo.

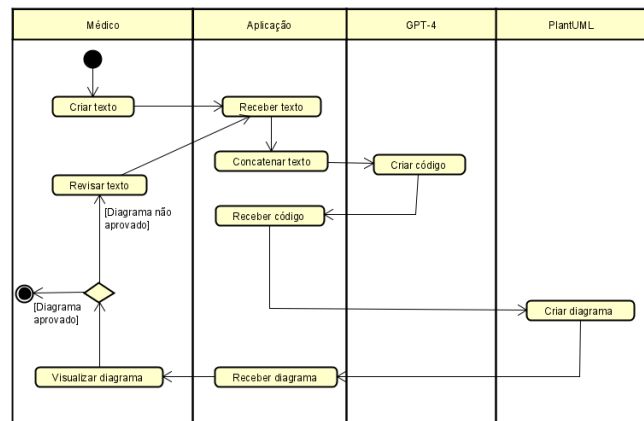


Figura 2 - Diagrama de atividade do projeto.

**3.1.2. Diagrama de casos de uso.** Este diagrama apresenta as funcionalidades do sistema e seu usuário.

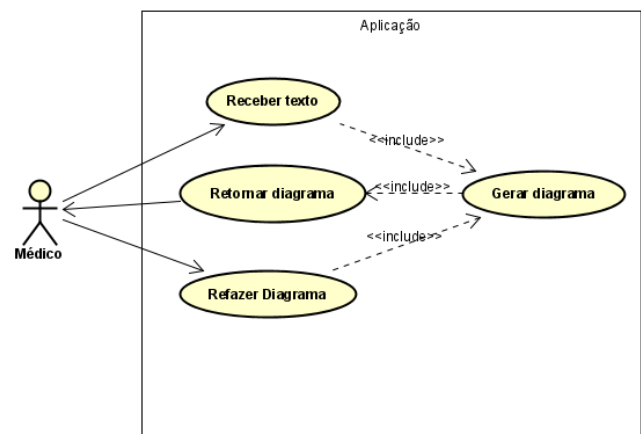


Figura 3 - Diagrama de casos de uso do projeto.

## 4. Resultados

À partir de testes realizados obteve-se como retorno do sistema, Resultados e problemas identificados para o desenvolvimento.

Foram testadas duas formas de comunicação com o chat GPT a primeira através de uma linguagem formal em forma contínua ou seja um texto corrido e a segunda uma linguagem formal em forma de frases ou seja através de frases.

### 4.1. Linguagem formal de forma contínua

O texto inserido pelo usuário deve estar bem escrito e entendível, descrito passo a passo sobre como deve ser desenvolvido o diagrama.

Exemplo de linguagem formal de forma continua que o usuário deve informar: "Gere um código PlantUML do diagrama de atividade que começa com a chegada de um paciente, que espera a chegada de um enfermeiro para que realize a triagem, após encaminha o paciente para a emergência se estiver em condição crítica ou para a sala de espera caso contrário. Em seguida, chama o paciente para a consulta com o médico, que realiza exames se necessário, dá o diagnóstico, agenda cirurgia se necessário ou prescreve medicamentos, e finalmente o paciente recebe tratamento e a consulta é finalizada. Sendo o médico, paciente e o enfermeiro partições individuais."

É de extrema importância palavras como "se necessário" ou "caso contrário" para que o GPT entenda que é uma condição a ser executada.

Com esta maneira de linguagem formal, o chat GPT não consegue identificar possíveis agentes para as partições, uma solução seria ao final do texto descrever quem são os agentes.

## 4.2. Linguagem formal em forma de frases

O texto em forma de frases que foi desenvolvido segue o seguinte padrão: Primeira palavra é o agente e restante é sua ação que pode apresentar condicionais. Ex: "Paciente chega ao hospital." Paciente é o agente e o restante é a ação do paciente.

Exemplo de texto em forma de frases criada: "O texto em forma de frases a seguir segue o seguinte padrão: A primeira palavra é o agente do diagrama. Lembrando que o agente é uma partição do diagrama. Sabendo disso gere o código PlantUML do diagrama de atividade a seguir:

Paciente chega ao hospital

Paciente espera a chegada de um enfermeiro

Enfermeiro realiza a triagem.

Enfermeiro encaminha o paciente para a emergência se estiver em condição crítica se não para a sala de espera.

Médico chama o paciente para a consulta.

Médico realiza exames se necessário.

Médico dá o diagnóstico, agenda cirurgia se necessário ou prescreve medicamentos.

Paciente recebe tratamento e a consulta é finalizada."

Texto começa dando o padrão em que o texto segue para que o chat GPT consiga entender o padrão e possa compreender melhor. E diz a ele que esse agente no diagrama deve ser uma partição individual.

Já com esta maneira de linguagem formal o chat GPT não conseguiu identificar possíveis condicionais no texto mesmo sendo explicado no texto de introdução do padrão que a linguagem segue.

## 5. Conclusão

À partir de estudos realizados com o chat GPT para desenvolver uma linguagem padrão foram identificados problemas ao gerar o código do diagrama, o mesmo não conseguiu identificar as partições informadas através do texto sem que fosse preciso especificar essa informação também apresentou problemas com sintaxe errada em alguns testes.

## 5.1. Diagramas de atividade criados

O Diagrama de atividade à seguir foi criado para que seja feita uma comparação entre os diagramas desenvolvidos pelo chat GPT.

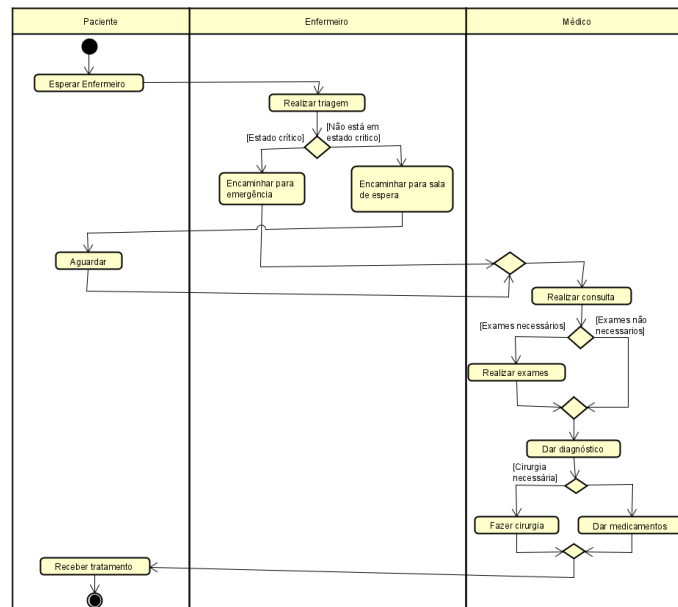


Figura 4 - Formato esperado do diagrama de atividade.

O diagrama de atividade com a linguagem de forma continua, apresenta uma grande similaridade entre o diagrama esperado e a saída do chat GPT. É possível ver os nós de decisões de uma forma muito próximas e as ações dos agentes quase iguais.

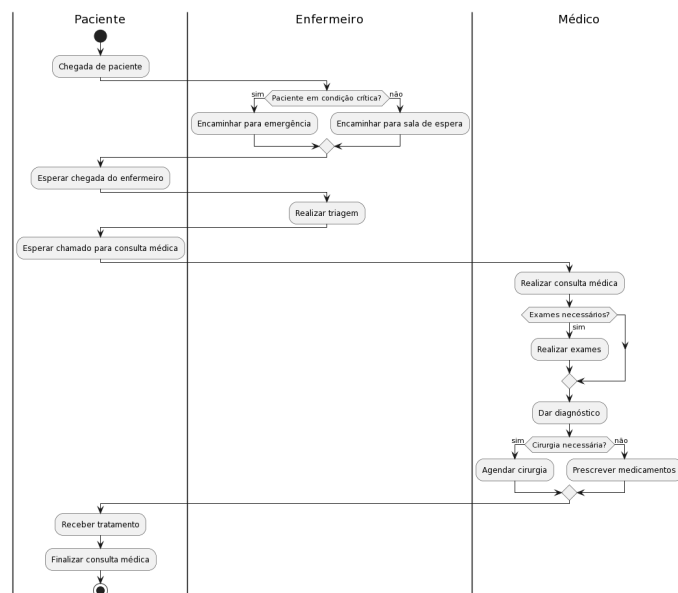


Figura 5 - Formato com linguagem formal de forma continua.

Já o diagrama com a linguagem em forma de frases, possui uma dessemelhança muito grande, já que ele não

conseguiu entender quando devia colocar nós de decisões e as ações ficaram com o nome do agente apresentando uma redundância.

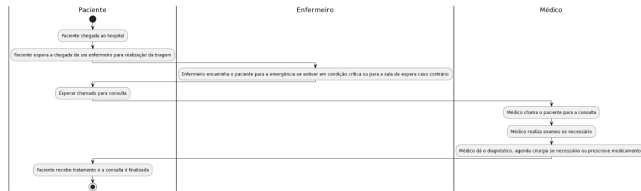


Figura 6 - Formato com linguagem formal em forma de frases, gerada pelo chat GPT.

## References

- [1] F. Hans-Georg, F. Peter and K. Julius. Conceptual Modeling and Large Language Models: Impressions From First Experiments With ChatGPT. Acessado dia 19 de junho de 2023. Disponível em: <https://folia.unifr.ch/global/documents/324646>
- [2] S. Abukhalaf, M. Hamdaqa and F. Khomh. On Codex Prompt Engineering for OCL Generation: An Empirical Study. Acessado dia 19 de junho de 2023. Disponível em: <https://arxiv.org/abs/2303.16244>
- [3] IBM. Diagramas de Atividades. Acessado dia 19 de junho de 2023. Disponível em: <https://www.ibm.com/docs/pt-br/rational-software/9.7.0?topic=diagrams-activity>
- [4] C. Fabro. O que é API e para que serve? Cinco perguntas e respostas. Acessado dia 19 de junho de 2023. Disponível em: <https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghtml>
- [5] F. Leite. O que é Chat GPT-4: nova versão da OpenAI. Acessado dia 19 de junho de 2023. Disponível em: <https://www.agenciaimma.com.br/o-que-e-chat-gpt-4/>
- [6] Open Risk Manual. PlantUML. Acessado dia 22 de junho de 2023. Disponível em: <https://www.openriskmanual.org/wiki/PlantUML>
- [7] AWS. O que é uma API?. Acessado dia 26 de junho de 2023. Disponível em: <https://aws.amazon.com/pt/what-is/api/>