

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GIUSEPE TESSARI LOPES DE OLIVEIRA

Aplicação de LLMs na Modelagem de Requisitos: Melhorando a Criação de Diagramas UML e Estimulando a Criatividade em Novos Casos de Uso

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr^a. Leticia dos Santos Machado

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profª Marcia Cristina Bernardes Barbosa

Vice-Reitor: Prof. Pedro de Almeida Costa

Pró-Reitora de Graduação: Profª Nádyá Pesce da Silveira

Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary

Coordenador do Curso de Ciência da Computação: Prof. Marcelo Walter

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, por todo amor, cuidado e suporte, e por terem me proporcionado as melhores condições para que eu pudesse me dedicar aos meus estudos. À minha irmã que sempre me dá muito apoio, e que é uma enorme fonte de alegria e inspiração na minha vida. Também agradeço aos meus amigos e demais familiares, por compartilharem a vida comigo. A todos os professores que tive ao longo dessa jornada, em especial a professora Leticia, por ter aceitado orientar esse trabalho e dado todo suporte no planejamento e execução dele. Finalmente, agradeço a todos os técnicos administrativos e demais funcionários da UFRGS pelo ambiente de ensino que nos propiciam.

RESUMO

Avanços recentes nos *Large Language Models* (LLMs) têm atraído a atenção de pesquisadores e profissionais para a aplicação dessa tecnologia em diversas áreas. No contexto da engenharia de software, estudos têm focado em investigar a utilização dos LLMs nas etapas de desenvolvimento e testes, entretanto pouco se tem discutido sobre seu emprego na Engenharia de Requisitos (ER). O presente trabalho apresenta um estudo de caso que explora o uso de LLMs em atividades de ER, mais especificamente para criação de diagramas seguindo o padrão *Unified Modeling Language* (UML), e para modelagem de casos de uso criativos. Para isso conduziu-se um experimento com 24 desenvolvedores, onde solicitou-se que utilizassem uma ferramenta baseada em LLM para geração de um diagrama UML de atividade, e em seguida trabalhassem em conjunto com um modelo de LLM para elaboração de um novo caso de uso. Os resultados revelaram que o uso dessa tecnologia para geração de diagramas UML traz impacto positivo na produtividade de desenvolvedores, e que os LLMs são boas ferramentas para trazer inspiração durante a modelagem de requisitos. Entretanto, que não são totalmente confiáveis, muitas vezes demandando intervenção humana em seus resultados.

Palavras-chave: Engenharia de Software. Engenharia de Requisitos. Modelagem de Software. *Large Language Models*. UML. Criatividade.

Application of LLMs in Requirements Modeling: Improving UML Diagram Creation and Stimulating Creativity in New Use Cases

ABSTRACT

Recent advances in Large Language Models (LLMs) have attracted the attention of researchers and professionals to the application of this technology in several areas. In the context of software engineering, studies have focused on investigating the use of LLMs in the development and testing stages, however little has been discussed about their use in Requirements Engineering (RE). The present work presents a case study that explores the use of LLMs in RE activities, more specifically for creating diagrams following the Unified Modeling Language (UML) standard, and for modeling creative use cases. To this end, an experiment was conducted with 24 developers, where they were asked to use a LLM-based tool to generate an UML activity diagram, and then work together with a LLM model to develop a new use case. The results revealed that the use of this technology to generate UML diagrams has a positive impact on developer productivity, and that LLMs are good tools for providing inspiration during requirements modeling. However, they are not completely reliable, often requiring human intervention in their results.

Keywords: Software Engineering. Requirements Engineering. Software Modeling. Large Language Models. UML. Creativity.

LISTA DE FIGURAS

Figura 2.1 – Tendências no número de pré-impressões do arXiv	15
Figura 2.2 – Evolução da engenharia de software.....	18
Figura 2.3 – Pontuação média no DAT e intervalos de confiança de 95%	24
Figura 3.1 – Atividades realizadas durante o estudo de caso	31
Figura 4.1 – Frequência do uso de LLMs em atividades de engenharia de software	36
Figura 4.2 – Diagrama resultante obtido através da Lucid GPT	37
Figura 4.3 – Conformidade do diagrama gerado com a notação UML.....	39
Figura 4.4 – Precisão da representação dos elementos de software.....	39
Figura 4.5 – Completude e legibilidade dos diagramas gerados.....	40
Figura 4.6 – Resultado da inclusão de raias pelo Lucid GPT	41
Figura 4.7 – Disposição das raias no “gabarito” do cenário	41
Figura 4.8 – Aumento da produtividade e poucas interações para obter o diagrama.....	43
Figura 4.9 – Satisfação com o diagrama e recomendação a outros desenvolvedores	44
Figura 4.10 – Comparação entre o diagrama manual e o gerado pela Lucid GPT.....	44
Figura 4.11 – Avaliação do desempenho criativo apresentado pelo ChatGPT.....	48
Figura 4.12 – Prompt e fluxo principal do caso de uso 4 gerado pelo LLM.....	50
Figura 4.13 – Prompt e fluxo principal do caso de uso 10 gerado pelo LLM.....	51
Figura 4.14 – LLM potencializando a criatividade e gerando ideias viáveis.....	52

LISTA DE TABELAS

Tabela 3.1 – Características das ferramentas analisadas	30
Tabela 3.2 – Perguntas da primeira fase do estudo	32
Tabela 3.3 – Perguntas da segunda fase do estudo	33
Tabela 4.1 – Lista de temas e códigos identificados para as respostas abertas	37
Tabela 4.2 – Ideias de novos casos de uso sugeridas pelo LLM	48

LISTA DE ABREVIATURAS E SIGLAS

CASE	<i>Computer Aided Software Engineering</i>
DAT	<i>Divergent Association Task</i>
ES	Engenharia de Software
ER	Engenharia de Requisitos
GENAI	<i>Generative Artificial Intelligence</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	Inteligência Artificial
IOT	<i>Internet of Things</i>
LLM	<i>Large Language Model</i>
ML	Modelo de Linguagem
PP	Pergunta de Pesquisa
UFRGS	Universidade Federal do Rio Grande do Sul
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1 INTRODUÇÃO	9
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 Definição e Contextualização de LLM	11
2.2 LLM no Contexto de Engenharia de Software	14
2.2.1 Perspectiva sobre o Futuro do Processo de Engenharia de Software Assistido por IA	17
2.2.2 Problemas em Aberto sobre a Utilização de LLMs	18
2.3 Uso de LLMs em Atividades de Engenharia de Requisitos	19
2.4 Criatividade no Contexto de Engenharia de Requisitos	21
2.5 Avaliação da Criatividade de LLMs	23
2.5 Emprego de LLMs para Geração de Diagramas UML e Casos de Uso Mais Criativos	24
3 METODOLOGIA	26
3.1 Avaliação de Ferramentas Baseadas em LLM para Geração de Diagramas UML	27
3.1.1 Miro	27
3.1.2 Lucidchart e Lucid GPT	28
3.1.3 Eraser	29
3.1.4 Diagramming AI	29
3.1.5 Escolha da Ferramenta	29
3.2 Primeira Fase: Estudo de Caso	30
3.3 Segunda Fase: LLM para Modelagem de Novos Casos de Uso	33
4 ANÁLISE DE RESULTADOS	35
4.1 Demografia dos Participantes	35
4.2 Resultados da Primeira Fase	36
4.2.1 Avaliação da Estrutura do Diagrama Gerado	38
4.2.2 Satisfação com o Uso do LLM	42
4.2.3 Retomando a Pergunta de Pesquisa 1	46
4.3 Resultados da Segunda Fase	47
4.3.1 Retomando a Pergunta de Pesquisa 2	53
4.4 Limitações do Estudo	54
5 CONCLUSÃO	55
5.1 Disponibilidade dos Dados	56
REFERÊNCIAS	57
ANEXO A - DESCRIÇÃO DO CASO DE USO	61

1 INTRODUÇÃO

Estamos em um mundo que demanda que soluções de software cheguem cada vez mais rapidamente ao mercado, resultando na busca por formas de otimizar o processo de produção de software sem que seja preciso pular etapas, conforme os trabalhos de Beck et al. (2001) e Kim *et al.* (2016), que discutem a importância de processos ágeis que entregam software rapidamente sem abrir mão da qualidade. Além disso, em um cenário de crescente digitalização, onde a tecnologia evolui diariamente, *stakeholders* têm buscado não só que seus produtos contêm os requisitos que idealizaram, mas também que apresentem diferenciais quando comparados a seus concorrentes, como pode-se observar em Maiden, Gizikis e Robertson (2004) e Santos *et al.* (2022). Avanços recentes no campo da inteligência artificial popularizaram os grandes modelos de linguagem (*Large Language Models*, ou LLM), que por sua vez têm atraído pesquisadores e profissionais de diversas áreas que buscam avaliar aplicações dessa tecnologia em contextos variados, como é o caso de Wan *et al.* (2024), que investigaram o uso de LLMs auxiliando em atividades de escrita criativa.

Fan *et al.* (2023) mapearam o progresso de pesquisas sobre a utilização de LLMs auxiliando nas diversas etapas do processo de engenharia de software. Nesse trabalho, os autores identificaram que os maiores avanços apresentam-se nas etapas de desenvolvimento e testes, e que ainda há um número limitado de pesquisas sobre a aplicação dessa tecnologia para engenharia de requisitos.

Nesse contexto, o presente trabalho tem por objetivo investigar como os LLMs podem auxiliar na etapa de engenharia de requisitos, olhando mais especificamente para modelagem de requisitos através de diagramas UML¹ (*Unified Modeling Language*). Além disso, também busca-se avaliar a influência do uso dos LLMs no processo criativo para concepção de novos casos de uso.

Para isso, conduziu-se um estudo de caso com estudantes de graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul (UFRGS). Na primeira etapa desse estudo, foi necessário selecionar uma ferramenta CASE (*Computer Aided Software Engineering*) baseada em LLM para geração de diagramas UML. Então realizou-se o

¹UML. **Endereço da especificação.** Disponível em: <https://www.omg.org/spec/UML/>. Acesso em: 29 dez. 2024.

experimento, onde uma descrição de caso de uso foi fornecida aos estudantes, e lhes foi solicitado que criassem manualmente um diagrama UML de atividades para esse caso de uso, gerassem um diagrama do mesmo tipo utilizando a ferramenta CASE baseada em LLM escolhida, e comparassem os resultados obtidos. Também lhes foi requisitado que trabalhassem em conjunto com um LLM popular (ChatGPT²) para modelarem um novo caso de uso que integrasse um sistema de forma a torná-lo inovador. Ao final do experimento os participantes responderam a um questionário, onde avaliaram os resultados obtidos bem como sua percepção sobre benefícios e desafios da utilização do LLM na execução da atividade.

O texto aqui apresentado foi organizado em 5 capítulos, sendo este o capítulo inicial de introdução, seguido pelo Capítulo 2 onde é apresentada a fundamentação teórica necessária para o entendimento do texto como um todo, contextualizando sobre o que são os *Large Language Models* e quais são seus diferentes tipos, descrevendo diferentes aplicações de LLMs em atividades de engenharia de software e de requisitos, discutindo o uso de criatividade no contexto de engenharia de requisitos e apresentando uma avaliação do nível criativo de diferentes modelos LLM. Em seguida, o Capítulo 3 detalha a escolha da ferramenta CASE baseada em LLM para geração de diagramas UML, bem como a metodologia e a coleta de dados do presente trabalho. Ao longo do Capítulo 4, discutem-se os resultados obtidos com o experimento, bem como as limitações do presente estudo. Finalmente, o Capítulo 5 apresenta conclusões e contribuições desse trabalho, assim como trabalhos futuros.

²CHATGPT. **Endereço do LLM.** Disponível em: <https://chatgpt.com/>. Acesso em: 29 dez. 2024.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo tem por objetivo apresentar o que são os *Large Language Models* (LLMs), como esses modelos estão se popularizando e ganhando espaço em diversas atividades do processo de engenharia de software e qual é a força de seu uso para tarefas criativas, bem como discorrer sobre a forma com que a criatividade tem sido utilizada para engenharia de requisitos. Os conceitos aqui apresentados formam a base necessária para a compreensão da proposta de trabalho, sua importância e relevância nos dias atuais.

2.1 Definição e Contextualização de LLM

Para que seja possível entender o que é um LLM, é preciso primeiramente entender o que é um modelo de linguagem (ML). Segundo Russell e Norvig (2010), ML são modelos que prevêem a distribuição de probabilidade de expressões de linguagem. Linguagens formais como as de programação têm modelos precisamente definidos. Já linguagens naturais, como Português e Inglês, não podem ser caracterizadas como um conjunto definitivo de sentenças. Portanto, é mais produtivo definir um modelo de linguagem natural como uma distribuição de probabilidade sobre sentenças, em vez de um conjunto definitivo. Linguagens naturais também são ambíguas, podendo existir mais de um significado para uma sentença, o que reforça o uso de uma distribuição de probabilidade sobre os possíveis significados.

De forma semelhante, Developers (2024) define ML como um modelo que visa prever e gerar uma linguagem plausível, trabalhando a partir da estimativa de probabilidade de um *token* ou sequência de *tokens* ocorrer em uma sequência ainda maior. *Tokens*, por sua vez, são unidades atômicas que os ML utilizam para representar sua entrada ou saída. São enumerações que podem representar palavras, caracteres, subpalavras ou outros segmentos de texto e/ou código (Fan *et al.*, 2023). A seguinte sentença representa um exemplo: “Quando sinto cheiro de flores, sinto vontade de _”. Um ML poderia determinar as seguintes probabilidades para palavras ou sequências de palavras para substituir o símbolo *underscore*: “cantar 9.6%; fazer um piquenique 8.3%; ir ao médico 4.1%; carro 2.2%”.

Conhecendo o conceito de ML, é possível então definir o que é um LLM. Grande modelo de linguagem (*Large Language Model*, ou LLM) refere-se a um modelo de

inteligência artificial que foi treinado com grandes quantidades de dados e é capaz de gerar texto de maneira semelhante à humana. São tipicamente baseados em técnicas de aprendizado profundo, como transformadores, e conseguem gerar resultados linguísticos úteis, sendo considerados capazes de realizar uma vasta gama de tarefas relacionadas a linguagem, incluindo geração de texto, resposta a perguntas, tradução, entre outras (Fan *et al.*, 2023).

Modelar a linguagem humana em escala é uma tarefa complexa e demanda bastante recurso. Conforme os modelos crescem, sua complexidade e eficácia aumentam, sendo capazes de prever a probabilidade de parágrafos ou até mesmo documentos inteiros. Segundo Developers (2024), um desenvolvimento chave em modelagem de linguagens ocorreu em 2017 com a introdução da arquitetura de transformadores, projetada em torno da ideia de autoatenção. Ela habilitou o processamento de sequências mais longas, focando apenas nas partes mais importantes da entrada, solucionando problemas de memória enfrentados até então. O mecanismo de autoatenção captura o relacionamento entre cada *token* da entrada, e no lugar de cada um deles pergunta, “Quanto cada um dos outros *tokens* é relevante para mim?”. Conforme Fan *et al.* (2023), transformadores consistem de um codificador, que mapeia uma entrada de tipo específico (como texto, imagem, áudio) em uma representação intermediária, e um decodificador, que recebe essa representação como entrada e constrói dados do tipo original.

Ainda de acordo com Fan *et al.* (2023), existem três categorias de LLM. São eles:

1. **Modelo apenas codificador** – Consiste de uma rede codificadora mas não possui uma rede decodificadora separada. Recebe uma sequência de entrada e a mapeia para uma representação de menor dimensão. Seu propósito é aprender uma codificação dos dados de entrada. Exemplos de LLMs dessa categoria são BERT do Google, RoBERTa da Meta, e DeBERTa da Microsoft.

2. **Modelo codificador-decodificador** – Adicionalmente à rede codificadora, também possui uma rede decodificadora que produz uma sequência de saída gerando iterativamente *tokens* ou símbolos com base no vetor de contexto e *tokens* gerados anteriormente. Podem ser adotados para tarefas como tradução ou geração

de texto. São exemplos de LLMs dessa categoria o T5 do Google e o BART da Meta.

3. Modelo apenas decodificador – Diferentemente das outras duas categorias, nesse modelo as LLMs não possuem uma componente codificadora para processar dados de entrada, mas apenas uma componente decodificadora que gera diretamente uma sequência de saída baseada num dado contexto ou entrada. Cada *token* gerado pelo decodificador é condicionado ao contexto e aos *tokens* gerados anteriormente. Exemplos de LLMs dessa categoria são a série GPT (*Generative Pre-trained Transformer*) desenvolvida pelo OpenAI, LLaMA da Meta, Claude da Anthropic, e PaLM e Gemini do Google.

Os LLMs podem ser questionados quanto a sua dimensão, sendo comumente mensurada a partir de seu número de parâmetros. Conforme descrito em Developers (2024), parâmetros são os pesos que o modelo aprendeu durante o processo de treinamento, e são usados para prever qual o próximo *token* na sequência. Ainda segundo o autor, BERT (110 milhões de parâmetros) e PaLM (até 340 bilhões de parâmetros) foram descritos como grandes. Adicionalmente, GPT-3 é treinado com 45 terabytes de dados textuais e possui 175 bilhões de parâmetros, e LLaMA é treinado com 1.4 trilhões de tokens, com tamanhos de modelo variando de 7 bilhões a 65 bilhões de parâmetros.

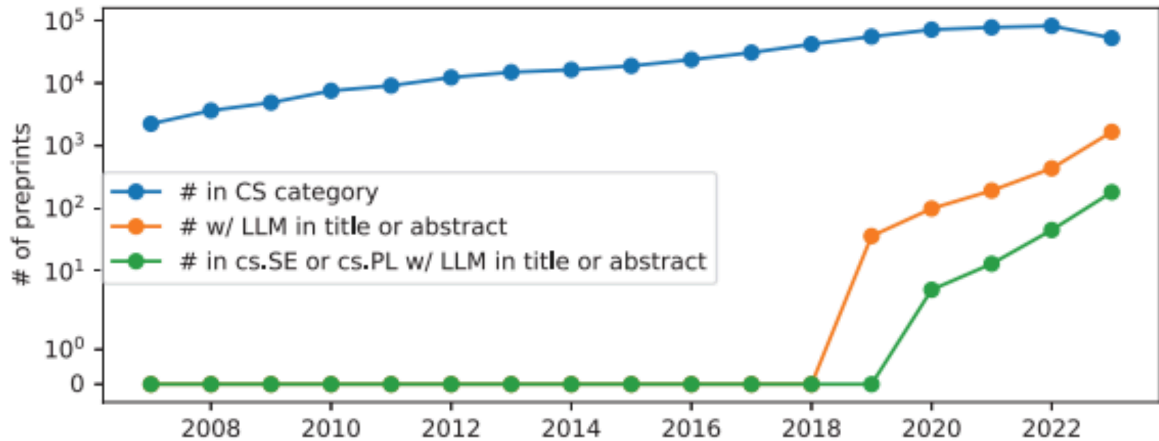
Na literatura referente a LLMs, frequentemente observa-se esse tópico sendo tratado em conjunto com inteligência artificial generativa (GenAI). Ferrara (2024) apresenta uma diferenciação entre esses dois conceitos, que têm recebido muita atenção tanto da academia quanto da indústria. Segundo o autor, GenAI refere-se a sistemas de inteligência artificial (IA) que são capazes de gerar novos conteúdos, incluindo texto, imagens e áudio, baseado em dados existentes. Diferente da IA tradicional, que busca reconhecer padrões ou fazer previsões, GenAI cria novas saídas a partir de algoritmos e modelos complexos que aprendem com grandes conjuntos de dados, reconhecem estruturas subjacentes e as emula de formas únicas. LLMs são um subconjunto desses sistemas, que lidam especificamente com dados textuais, sendo treinados com uma coletânea extensiva de textos, aprendendo padrões de linguagem, sintaxe e contexto.

2.2 LLM no Contexto de Engenharia de Software

Os recentes avanços em performance e precisão e o aumento da publicidade dos LLMs, bem como suas capacidades, têm atraído pesquisadores e profissionais de diversas áreas que buscam avaliar aplicações desses sistemas em contextos variados. Wan *et al.* (2024) descrevem um estudo sobre como usuários interagem com LLMs para suporte de atividades de pré-escrita, sendo essa a atividade mental que ocorre antes que as ideias de escrita estejam prontas para palavras ou nas páginas. Já Ferrara (2024) discute os riscos do uso indevido desses sistemas, desde seu potencial papel na intensificação de campanhas de desinformação até à criação de fraudes direcionadas ou álibis personalizados, e destaca a urgência de estratégias robustas de mitigação, diretrizes éticas e monitoramento contínuo para sistemas de GenAI e LLM.

Buscando entender as tendências de crescimento na área de engenharia de software (ES) baseada em LLM, Fan *et al.* (2023) analisaram manualmente dados referentes ao número de publicações no arXiv, um repositório de acesso aberto de pré-impressões e pós-impressões eletrônicas de artigos científicos. A Figura 2.1 mostra os resultados do crescimento de publicações desde o ano de 2008, onde a linha azul denota o total de pré-impressões categorizadas em Ciência da Computação, a linha laranja representa quantas delas possuíam ocorrência de LLM no título ou resumo, e a linha verde denota o número de pré-impressões nas categorias de ES ou linguagem de programação cujo título ou resumo continha LLM, no período entre 2007 e 27 de Julho de 2023. É possível perceber que a proporção de artigos sobre ES baseada em LLM tem aumentado drasticamente desde 2019, com mais de 10% de todas as publicações relacionadas a LLMs tratando desse assunto.

Figura 2.1 – Tendências no número de pré-impressões do arXiv



Fonte: Fan *et al.* (2023)

Em Fevereiro de 2023, Zhao (2023) reportou que o GitHub Copilot³, uma ferramenta baseada em LLM que fornece conclusão de código para usuários, estava por trás de uma média de 46% do código dos desenvolvedores, colocando em perspectiva todas as linguagens de programação. Ao comparar com a média de 27% reportada no ano de 2022, pode-se perceber uma clara tendência dos desenvolvedores adotarem essas ferramentas durante a realização de suas tarefas de programação.

Peng *et al.* (2023) apresentam um ensaio controlado do GitHub Copilot, conduzido com 95 programadores profissionais, que foram solicitados a implementarem o mais rápido possível um servidor HTTP em Javascript. O grupo de tratamento tinha acesso ao GitHub Copilot e assistiu a um breve vídeo sobre como utilizar a ferramenta. Já o grupo de controle estava livre para realizar buscas na internet para completar a tarefa, entretanto não possuía acesso ao GitHub Copilot. A diferença de performance observada foi estatisticamente significativa, onde o grupo de tratamento completou com sucesso a tarefa 55.8% mais rápido, evidenciando um impacto positivo no desempenho dos desenvolvedores que trabalharam em conjunto com a ferramenta baseada em LLM.

Trabalhos nesse sentido têm sido conduzidos para examinar a aplicação prática de LLMs no ensino superior de ES. Dirin e Laine (2024) realizaram um experimento com 36 estudantes de Engenharia de Software no quarto semestre, dividindo-os em times para desenvolverem projetos de software a partir de requisitos de alto nível fornecidos

³GITHUB COPILOT. **Endereço da ferramenta.** Disponível em: <https://github.com/features/copilot>. Acesso em: 19 dez. 2024.

previamente. Um subconjunto desses times foi instruído a utilizar livremente ferramentas de IA (como Copilot e ChatGPT) no desenvolvimento do projeto, outro precisaria de aprovação para realizar qualquer atividade com essas ferramentas, e o último subconjunto deveria construir o projeto sem qualquer auxílio de IA. Os resultados obtidos demonstraram que o uso de LLMs facilitou a implementação das tarefas e funcionalidades definidas inicialmente, com os times que tiveram acesso total ou parcial a ferramentas de IA conseguindo entregar um número maior de tarefas, enquanto os grupos sem esse acesso tiveram mais tarefas adiadas para futuras iterações do produto. Nesse contexto, Kozov, Ivanova e Atanasova (2024) publicaram um trabalho que trata do desenvolvimento de aplicações com auxílio de LLMs por estudantes de Engenharia de Software de nível superior. Os autores constataram que estudantes com o mesmo antecedente universitário foram bem sucedidos em projetos que são mais complexos, completos e aprofundados que os de anos anteriores. Ainda, que a melhoria mais significativa ocorreu naqueles com menor desempenho, visto que o acesso a ferramentas tão poderosas elevou seu nível de habilidade suficientemente para que se sentissem motivados a completar um projeto de software grande.

Apesar de geração e conclusão de código ser a área mais explorada até o momento para aplicação de LLMs na ES, Fan *et al.* (2023) também revisam o progresso de estudos sobre o uso dessas ferramentas nas etapas de teste de software (para geração, avaliação da adequação e minimização de testes), manutenção e evolução (para depuração, melhoria de performance e refatoração), geração de documentação, entre outras. Entretanto, ainda segundo os autores, a disciplina de engenharia de requisitos tem recebido até agora menos atenção da literatura emergente sobre ES baseada em LLM. Diferente das outras atividades de software, não foram encontrados muitos trabalhos sobre ER baseada em LLM, apesar desses sistemas também terem potencial significativo para apoiar atividades nessa área, graças às suas capacidades de processamento de linguagem natural. Essa baixa representatividade, bem como o potencial apresentado pelas LLMs, servem como uma das motivações do presente trabalho.

2.2.1 Perspectiva sobre o Futuro do Processo de Engenharia de Software Assistido por IA

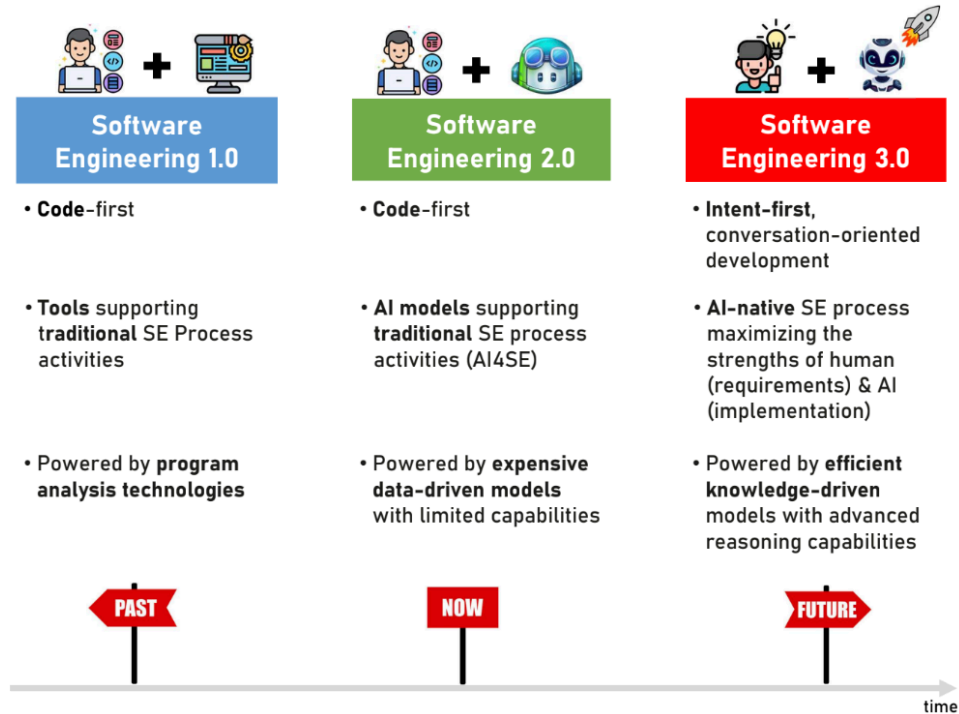
Com a crescente adoção de sistemas de IA na ES, começam a surgir questionamentos sobre a eficiência da abordagem que tem sido utilizada para integração dessas ferramentas no processo de produção de software. Em seu trabalho, Hassan *et al.* (2024) apresentam o termo Engenharia de Software 3.0, uma evolução da era atual. Conforme descrito no artigo, a Engenharia de Software 1.0 era enraizada em uma abordagem que priorizava o código, onde as ferramentas primariamente suportavam atividades do processo de ES tradicional. No contexto de programação, essas ferramentas eram utilizadas para auxiliar no gerenciamento da complexidade do código, garantindo correção e otimizando o desempenho. De forma geral, os humanos estavam no centro e dirigiam todo o processo.

Na era da Engenharia de Software 2.0, o código continua sendo priorizado, entretanto os modelos de IA são incorporados para melhorar as atividades de ES tradicional, passando a ser descrito como um processo assistido por IA. Os autores listam algumas limitações dessa forma de produzir software, entre elas alta sobrecarga cognitiva nos humanos e qualidade do código gerado pelas ferramentas baseadas em LLM abaixo do ideal. Os desenvolvedores ainda são responsáveis por conduzir a criação do código, e portanto precisam decompor um problema complexo, planejar possíveis soluções e explorá-las, entretanto frequentemente se encontram adaptando ou corrigindo o código produzido por IA para adequá-lo ao projeto.

Sendo assim, os autores apresentam a Engenharia de Software 3.0 como a alternativa para o futuro da integração de IA no processo de desenvolvimento de software. ES 3.0 é nativo de IA e busca aproveitar da forma mais eficiente possível os pontos fortes e complementares de desenvolvedores humanos (como refletir sobre as necessidades do negócio) e sistemas sofisticados de inteligência artificial (como por exemplo a habilidade de buscar múltiplas soluções para um problema de forma muito rápida). Ela marca uma mudança de paradigma em direção a uma abordagem que prioriza a intenção, de forma que o desenvolvimento passa a ser dirigido por intenções expressas através de conversas entre desenvolvedores e a IA. Nesse paradigma de desenvolvimento orientado a conversação, a inteligência artificial conduz a criação de código sintetizando as intenções em software

executável. A Figura 2.2 ilustra essa evolução da ES, bem como as principais características de cada uma das fases descritas.

Figura 2.2 – Evolução da engenharia de software



Fonte: Hassan *et al.* (2024)

2.2.2 Problemas em Aberto sobre a Utilização de LLMs

Apesar do uso de LLMs proporcionar diversos benefícios, conforme descrito através desse capítulo, ainda existem diversos problemas em aberto sobre o uso dessa tecnologia. No contexto da engenharia de software, de maneira geral a qualidade do artefato produzido por essas ferramentas está abaixo do ideal, o que torna necessário que os desenvolvedores façam uma avaliação desses artefatos gerados antes que possam ser utilizados. Encontram-se referências a isso nos trabalhos de Fan *et al.* (2023), Dirin e Laine (2024) e Hassan *et al.* (2024). Recorrentemente, trabalhos como Ferrari, Abualhaija e Arora (2024) e Fan *et al.* (2023) têm destacado alucinação como um problema que também ocorre com LLMs, o que representa um risco para o uso dessa tecnologia em ES. De forma similar ao que ocorre com humanos, alucinação significa que esses modelos podem criar saídas que são coerentes, entretanto fictícias. Para engenharia de software, isso implica que apesar de parecerem plausíveis, os artefatos gerados podem estar incorretos (Fan *et al.*, 2023).

Fora do contexto de ES, o uso de LLMs também tem levantado discussões sobre questões como privacidade, segurança cibernética e direitos autorais. Ferrara (2024) descreve diversas formas pelas quais um indivíduo mal intencionado pode prejudicar outros através do uso de LLMs, entre as quais o autor cita identidades sintéticas, assédio direcionado, desordem informacional e campanhas de influência. Esses e outros problemas evidenciam a urgência de políticas regulatórias para o uso dessas ferramentas.

2.3 Uso de LLMs em Atividades de Engenharia de Requisitos

O amplo espectro de tarefas e técnicas que levam ao entendimento dos requisitos de um sistema é chamado de engenharia de requisitos (ER) (PRESSMAN, 2009). Do ponto de vista do processo de software, é uma grande ação que começa durante a atividade de comunicação e continua na atividade de modelagem, conectando o projeto e a construção do produto. Conforme Pressman (2009), ER fornece o mecanismo apropriado para entender os desejos do cliente, analisar a necessidade, avaliar a viabilidade, negociar uma solução razoável, especificar essa solução de maneira não ambígua, validar a especificação e gerenciar os requisitos conforme eles são transformados em um sistema em operação. O autor ainda descreve as tarefas abrangidas por essa atividade, sendo elas a comunicação inicial entre os interessados na solução, seguido pelas etapas de elicitação, elaboração, negociação, especificação, validação e gerenciamento dos requisitos do sistema.

Em seu trabalho, Liu, Li e Peng (2010) realizaram uma pesquisa com indivíduos atuantes da área de engenharia de software na China, buscando entender o quanto ER está presente no contexto de trabalho desses indivíduos. Dos dados coletados a partir de 377 respostas obtidas, as atividades de ER mais praticadas foram aquisição, descrição e especificação, e modelagem e análise. Entre as técnicas de representação de requisitos, 57% dos participantes afirmaram que utilizam modelos UML em seus projetos. Fernández *et al.* (2017) questionaram atuantes da área de engenharia de software sobre suas percepções com relação ao processo de engenharia de requisitos, levantando seu ponto de vista com relação aos problemas enfrentados através desse processo. Representantes de 228 organizações espalhadas por dez países diferentes completaram essa pesquisa e falta de tempo para

execução das atividades de ER apareceu entre os principais problemas citados, tendo sido mencionado por 32% dos participantes.

Nesse contexto, os LLMs surgem como ferramentas com potencial de facilitar muito o trabalho de engenheiros de requisitos e agilizar os processos de ER. Arora, Grundy e Abdelrazek (2023) exploraram o uso de LLMs no processo de ER, focando nas atividades de elicitação, análise, especificação e validação de requisitos, e descobriram que os LLMs podem aprimorar diversas tarefas envolvidas nessa etapa, possibilitando automação, simplificação e aumento das capacidades humanas.

Requisitos são tipicamente escritos em linguagem natural, e complementá-los com modelos requer tempo e esforço manual significativo. Conforme Ferrari, Abualhaija e Arora (2024), trabalhos existentes sobre engenharia de requisitos para geração automática de diagramas UML a partir de requisitos tipicamente utilizam abordagens de processamento de linguagem natural baseadas em regras heurísticas. Tais abordagens possuem diversas limitações, incluindo esforço manual significativo para construção e manutenção, e difícil adaptação para diferentes contextos. Os LLMs podem superar algumas dessas limitações por apresentarem entendimento contextual aceitável, serem tipicamente pré-treinados e prontos pra uso, assim reduzindo o esforço humano na construção de ferramentas de geração de modelos.

Em seu estudo, Ferrari, Abualhaija e Arora (2024) investigaram a capacidade do ChatGPT de gerar diagramas de sequência UML no formato Planttext a partir de requisitos especificados em linguagem natural. Os resultados mostraram que, de forma geral, os diagramas estavam de acordo com os padrões e exibiram boa compreensibilidade. Entretanto, também ficou claro que a correção e completude dos artefatos produzidos são muitas vezes um desafio, manifestado particularmente na presença de características indesejadas como ambiguidade e inconsistência. Além disso, foi observado que múltiplas iterações de *prompt* contribuíram para melhorar o entendimento do LLM sobre os requisitos, resultando em um aumento na qualidade do diagrama produzido.

Já Wang *et al.* (2024) conduziram um experimento com estudantes de ensino superior cursando Engenharia de Software, onde exploraram o desempenho de LLMs auxiliando na criação de diagramas UML de caso de uso, de classe e de sequência para um dado estudo de caso. A maior parte dos participantes utilizou ChatGPT na atividade, e houveram diferenças

no formato dos diagramas entregues. Alguns solicitaram que a ferramenta gerasse os diagramas diretamente, o que resultou em uma aparência rudimentar e com grande chance de perda de informação. Uma segunda parcela dos participantes solicitou explicitamente que o ChatGPT gerasse o código dos diagramas no formato PlantUML, para que pudessem inseri-lo como entrada em uma ferramenta específica para gerar automaticamente os diagramas UML de forma gráfica. O restante dos participantes fez uso do LLM para gerar uma saída bruta dos diagramas, que então poderiam melhorar manualmente ou modificar os *prompts* buscando otimizar esse artefato, para então desenhar os diagramas a partir do resultado obtido utilizando ferramentas específicas para modelagem UML. Os resultados obtidos mostraram que LLMs podem desempenhar um papel auxiliando na análise e modelagem de requisitos. No entanto, também foram identificadas falhas e instabilidades, uma delas sendo a pouca habilidade apresentada pelo LLM em analisar e extrair relacionamentos entre os objetos.

2.4 Criatividade no Contexto de Engenharia de Requisitos

A criatividade está presente na área de pesquisa em engenharia de requisitos tendo como uma das principais motivações o surgimento e avanço de novas tecnologias de informação e comunicação, que por sua vez têm mudado drasticamente a maneira como as pessoas trabalham, interagem e vivem. Conforme afirmado por Maiden e Gizikis (2001), para construir produtos criativos e imaginativos que evitam reimplementar o óbvio, deve-se tornar a criatividade parte do processo de requisitos. Sendo assim, o pensamento criativo em ER é peça chave para expandir a visão e descobrir novos requisitos que possibilitem a criação de sistemas inovadores. Em sua publicação, Santos *et al.* (2022) reiteram que organizações obtêm benefícios naturais da criatividade por reorganizar de forma inovadora seus processos, projetos e produtos. Dessa forma, o foco em inovação e criatividade é reconhecido como fator importante para agilidade e competitividade dessas empresas.

Em Sternberg e Lubart (1995), criatividade é definida como a habilidade de produzir um trabalho que seja ao mesmo tempo novo (original, inesperado) e apropriado (ou seja, útil, adaptativo em relação a tarefas e restrições). Adicionalmente, Boden (2003) apresenta a descrição dos conceitos de criatividade psicológica (ou criatividade P), que pode ser analisada a nível de indivíduo, e criatividade histórica (ou criatividade H), que requer uma análise mais

ampla e transcultural. Sosa e Gero (2003) estendem essa idéia introduzindo o conceito de criatividade situada (ou criatividade S). Segundo os autores, soluções criativas são produzidas para satisfazer as necessidades de um grupo social específico. Dessa forma, é possível avaliar o grau de novidade de uma ideia através dos conceitos apresentados. Ideias *P-novel* parecem novas para o indivíduo que as criou, *S-novel* são reconhecidas como novas e originais pelos criadores e grupos sociais envolvidos, e *H-novel* são ideias que parecem originais para todo mundo. Esses conceitos são utilizados em trabalhos como Nguyen e Shanks (2009), que reitera que na engenharia de requisitos as ideias ocorrem em níveis intermediários no contínuo entre criatividade psicológica e histórica, como resultado de esforço coletivo e sujeito a julgamento de um time, da organização, da comunidade do domínio ou da sociedade.

Pode-se encontrar na literatura uma série de estudos sobre a aplicação de técnicas criativas para realização de atividades de ER. Santos *et al.* (2022) apresentam uma metodologia para introduzir criatividade e inovação no processo de engenharia de requisitos. A aplicação desse método no contexto real de uma instituição social possibilitou a identificação de novas oportunidades que permitiram que a organização passasse a fornecer serviços que são mais adequados às necessidades das pessoas.

Já Giunta *et al.* (2022) propõem uma ferramenta leve, que demanda menos tempo e envolvimento de recursos, para suporte de criatividade em ER, e avaliam o desempenho dessa ferramenta considerando a habilidade de revelar numerosas ideias, a qualidade das ideias produzidas e a conveniência de seu uso. A ferramenta se provou valiosa nos três pontos avaliados, com destaque para sua habilidade de estruturar e guiar discussões, dando caminhos para o pensamento e ajudando a superar a síndrome da página em branco.

Visando gerar requisitos através de criatividade exploratória, que consiste em explorar o espaço de possíveis ideias para assim criar novas, e criatividade combinatória, baseada na combinação e síntese de ideias existentes para criação de novas, Maiden, Gizikis e Robertson (2004) conduziram quatro *workshops* com um time de profissionais envolvidos no desenvolvimento de uma aplicação no contexto de gerenciamento de tráfego aéreo. Como resultado desses *workshops*, 201 ideias foram geradas. Analisando retrospectivamente, os autores dessas ideias determinaram que conseguiriam descrever apenas 54 delas como requisitos de *stakeholder* (propriedades desejáveis e que independem da solução do sistema futuro). As 147 restantes encapsularam diferentes graus de projeto e conhecimento do

processo. Também constatou-se que as ideias geradas forneceram uma base para escrita de casos de uso mais precisos.

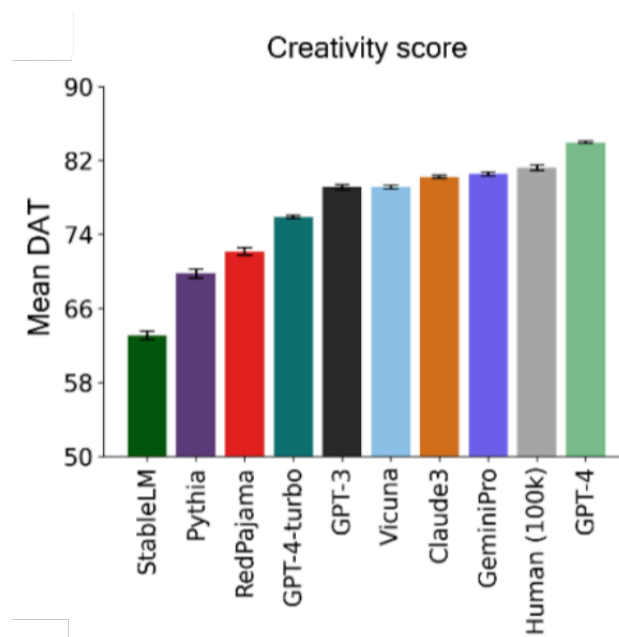
2.5 Avaliação da Criatividade de LLMs

O potencial do uso de LLMs para suporte criativo tem recebido bastante foco na literatura acadêmica. Estudos têm explorado como usuários interagem com os modelos para realização de atividades criativas, e avaliado o desempenho criativo dos LLMs. Em Wan *et al.* (2024), foram analisadas estratégias adotadas por usuários enquanto trabalhavam colaborativamente com um LLM para realização de tarefas criativas no contexto de pré-escrita. Foram identificadas duas formas principais utilizadas pelos participantes do estudo, sendo elas ideação explícita, em que perguntas eram feitas diretamente ao LLM na esperança que gerasse palavras chave, e ideação iterativa, em que os usuários incrementavam seus *prompts* com mais contexto afim de obter melhores resultados. Foi possível verificar que a aleatoriedade do LLM e resultados inesperados, causados ou por ambiguidades no *prompt* ou limitações do modelo, foram as fontes de inspiração mais comuns. As capacidades do LLM foram frequentemente observadas pelos participantes como surpreendentes, onde *prompts* com conceitos vagos foram interpretados para ideias concretas e fascinantes.

Em Zhao *et al.* (2024), os autores avaliaram a criatividade de seis LLMs distintos durante a execução de tarefas variadas, utilizando como critérios fluência (refere-se a quantidade de ideias), flexibilidade (habilidade de abordar um dado problema de diferente ângulos), originalidade e elaboração (capacidade de adicionar detalhes, desenvolver nuances). Os modelos avaliados nesse experimento foram GPT-3.5, LLaMA-2-13b, LLaMA-2-70b, Qwen, Vicuna-13b, Vicuna-7b. A partir dos resultados, foi possível observar que GPT-3.5 apresenta o nível mais elevado de criatividade, seguido pelos modelos da arquitetura LLaMA-2. O trabalho também comparou diferentes tipos de *prompts* em termos da qualidade da resposta produzida, e determinaram que os do tipo instrutivos, que fornecem descrição detalhada da resposta esperada, e cadeia de pensamento, técnica que permite raciocínios complexos através de etapas de raciocínio intermediárias, aumentaram significativamente o nível criativo dos modelos.

Com objetivo de avaliar o desempenho de uma ampla gama de LLMs em um teste de criatividade simples e validado, Bellemare-Pepin *et al.* (2024) compararam as pontuações obtidas por esses modelos com uma amostra de 100 mil respostas humanas. O teste escolhido para aplicação é chamado *Divergent Association Task* (ou DAT), e consiste em solicitar aos participantes que gerem uma lista de 10 palavras que são o mais semanticamente distantes possível. Indivíduos que são mais criativos tendem a cobrir um repertório semântico maior, resultando numa semântica média maior entre as palavras. O modelo GPT-4 demonstrou o melhor desempenho, superando até mesmo a pontuação humana, já GeminiPro e GPT-3 ficaram levemente abaixo. A Figura 2.3 mostra um gráfico com essas e as demais pontuações obtidas no experimento. Os autores também investigaram o desempenho dos três modelos que melhor performaram no DAT em tarefas de escrita criativa, e apesar de terem apresentado resultados positivos, os LLMs não foram capazes de superar as amostras humanas em termos de criatividade nessa atribuição.

Figura 2.3 – Pontuação média no DAT e intervalos de confiança de 95%



Fonte: Bellemare-Pepin *et al.* (2024)

2.5 Emprego de LLMs para Geração de Diagramas UML e Casos de Uso Mais Criativos

Conforme apresentado nos subcapítulos anteriores, pode-se perceber que os LLMs têm grande potencial criativo, e podem melhorar a engenharia de requisitos em diversas etapas, contribuindo para agilizar o processo de ER e para o projeto de sistemas competitivos e

inovadores. Dessa forma, o estudo aqui apresentado tem como objetivos entender, a partir de um *use case* previamente definido, os benefícios e desafios do emprego de LLMs para geração de diagramas UML de atividades, e avaliar a utilização desses modelos como parceiros criativos, com intuito de gerar casos de uso novos e criativos.

3 METODOLOGIA

O presente estudo foi conduzido em duas fases. Primeiramente, um estudo de caso foi realizado para avaliar a utilização de uma ferramenta CASE baseada em LLM por estudantes de graduação em Ciência da Computação, para geração de um diagrama UML de atividades. Já na segunda fase, o mesmo conjunto de participantes fez uso de um LLM como parceiro de trabalho, buscando obter ideias criativas para produção de um novo caso de uso.

A escolha da realização de um estudo de caso para esse trabalho se dá por ele ser um método de pesquisa particularmente adequado para examinação exploratória de fenômenos que precisam ser investigados no ambiente em que ocorrem, conforme Yin (2001). A aplicação desse método é indicada quando precisa-se gerar novas teorias suportadas na prática, entender a natureza e complexidade de processos conforme ocorrem, e trazer novos fatores e informações evidenciados durante a execução do processo estudado (YIN, 2013).

Uma primeira versão desse estudo de caso foi realizada durante o primeiro semestre de 2024, pela professora Leticia Machado, orientadora do presente trabalho, em conjunto com outros pesquisadores da UFRGS. O estudo aqui apresentado é uma modificação dessa versão anterior, com uma reestruturação das perguntas do questionário de acompanhamento a partir do *feedback* dos participantes do piloto, e com a adição de uma segunda etapa que buscou avaliar os LLMs no contexto da criatividade na elaboração de casos de uso. Dessa forma, as seguintes perguntas de pesquisa (PP) foram utilizadas para nortear o presente estudo:

- **PP1.** Qual é o potencial das ferramentas baseadas em LLM para auxiliar na elaboração de diagramas UML em projetos de modelagem de software?

- **PP2.** Como os LLMs contribuem no processo criativo para concepção de novos casos de uso em modelagem de software?

Com a primeira pergunta, buscou-se entender a percepção dos estudantes sobre o uso da ferramenta CASE baseada em LLM para geração do diagrama UML, e comparar a modelagem resultante com um diagrama criado manualmente pelos participantes, em termos de clareza, completude, estrutura e precisão. Já com a segunda, o objetivo foi analisar de que

forma os estudantes interagiram com o LLM buscando obter ideias criativas, e avaliar a percepção dos participantes sobre a habilidade criativa apresentada pelo modelo, conforme critérios de avaliação descritos em Zhao et al. (2024), e a experiência de trabalhar em conjunto com o LLM para realização da tarefa proposta.

O estudo foi conduzido como uma atividade da disciplina de Engenharia de Software, permitindo que os estudantes colocassem em prática os conceitos abordados em aula. Esses estudantes já tinham familiaridade com o contexto utilizado para modelagem nas duas etapas desse estudo, de uma aplicação para clínica veterinária, visto que haviam realizado outros exercícios para construção manual de diagramas UML de classes, caso de uso, sequência e diagrama de transição de estado baseados no mesmo contexto. No restante desse capítulo serão detalhadas as etapas envolvidas em cada uma das fases do experimento.

3.1 Avaliação de Ferramentas Baseadas em LLM para Geração de Diagramas UML

Para realização da primeira fase do experimento, foi necessário primeiramente selecionar uma ferramenta CASE baseada em LLM para geração de diagramas UML. Durante esse subcapítulo, serão apresentadas diferentes ferramentas disponíveis que se encaixam nesse perfil, entre elas a escolhida para utilização nesse estudo.

3.1.1 Miro

Miro⁴ é uma plataforma de colaboração digital para gerenciamento de projetos e comunicação distribuída de equipes. A plataforma oferece ferramentas que permitem a construção de artefatos para acompanhamento de projetos, como Kanban, e modelagem de processos e software, como fluxogramas e mapas da jornada do cliente.

Diagramas UML estão entre os modelos que podem ser construídos na plataforma. Além da criação manual de diagramas de atividades, caso de uso, classes, sequência, entre outros, a plataforma também possui um LLM integrado para geração de diagramas. A empresa anuncia essa ferramenta como Miro AI, entretanto não foi encontrada documentação

⁴MIRO. **Endereço da plataforma.** Disponível em: <https://miro.com/pt/>. Acesso em: 20 nov. 2024.

que detalha as características do modelo utilizado. Os tipos de diagramas que podem ser gerados por esse modelo são fluxograma, mapa mental, diagrama entidade relacionamento, diagrama de sequência UML e diagrama de classe UML.

A plataforma trabalha com um sistema de créditos para uso da Miro AI. Cada ação realizada pelo usuário com a IA consome um crédito. No plano gratuito são oferecidos 10 créditos por mês, enquanto nas assinaturas esse número varia entre 25 a 100 créditos por mês.

3.1.2 Lucidchart e Lucid GPT

Lucidchart⁵ é uma aplicação web de diagramação que permite aos usuários colaborarem visualmente na criação de gráficos e diagramas. Entre os diversos tipos de diagramas que podem ser criados através da plataforma estão diagrama entidade relacionamento, diagramas UML de atividades, classes e sequência, e fluxo de jornada do usuário.

A Lucidchart também oferece geração automática de diagramas diretamente na plataforma a partir de *prompts* descrevendo o cenário que está sendo modelado. Atualmente, fluxogramas, diagramas de atividades, classes e sequência e diagramas entidade relacionamento podem ser obtidos através da ferramenta. Conforme descrito em Lucid (2024), as funcionalidades de IA disponíveis na aplicação utilizam Microsoft Azure OpenAI Service, um serviço fornecido pela Microsoft que oferece acesso aos modelos de linguagem do OpenAI, para que esses possam ser adaptados para utilização em tarefas específicas. Ainda segundo a empresa, os recursos da Lucid AI estão atualmente disponíveis a todos os usuários, mas podem estar sujeitos a um modelo de assinatura pago no futuro.

Além disso, a Lucid também oferece uma integração com ChatGPT para geração de diagramas. Chamada de Lucid GPT, essa integração funciona como uma versão personalizada do modelo LLM do OpenAI e pode ser acessada diretamente da página principal do ChatGPT. Com base em uma descrição textual fornecida, a Lucid GPT gera uma visualização do diagrama solicitado em forma de imagem, e fornece um link que permite edição e compartilhamento desse diagrama através da Lucidchart.

⁵LUCIDCHART. **Endereço da plataforma.** Disponível em: <https://www.lucidchart.com/pages>. Acesso em: 21 nov. 2024.

3.1.3 Eraser

Eraser⁶ é um copiloto de IA para projeto técnico, capaz de gerar diagramas de arquitetura, fluxo de dados, árvore de decisão, entre outros. A plataforma trabalha com um modelo de planos de assinatura, e limita o uso da IA pelo número de diagramas gerados. No plano gratuito, os usuários podem gerar um máximo de 10 diagramas por conta. Já nos planos pagos, o total de diagramas varia de 20 mensais na assinatura mais básica, a ilimitados no plano empresarial.

A plataforma também disponibiliza de forma gratuita o DiagramGPT, uma ferramenta baseada em LLM para geração de diagramas a partir de descrição textual. De acordo com Eraser (2024), a ferramenta aproveita-se do GPT-4 para execução de suas tarefas. Atualmente, os tipos de diagramas disponíveis para geração através do DiagramGPT são entidade relacionamento, sequência, fluxograma e arquitetura *cloud*.

3.1.4 Diagramming AI

Diagramming AI⁷ é uma ferramenta para geração e edição de diagramas que usa como modelo o GPT-4 do OpenAI. Sua utilização também é baseada em créditos, onde cada crédito representa uma ação com a IA, e os planos variam de 10 créditos na versão gratuita a 2500 créditos no plano profissional. A ferramenta é capaz de gerar uma grande variedade de diagramas, incluindo fluxograma, UML de classes, UML de estados, UML de sequência, entidade relacionamento, jornada de usuário, entre outros.

3.1.5 Escolha da Ferramenta

Para seleção da ferramenta utilizada na primeira etapa do estudo reportado nesse trabalho, foram considerados como critérios a capacidade de gerar diagramas UML de atividades, visto que é o tipo sob análise no presente trabalho, e a possibilidade de interagir

⁶ERASER. **Endereço da plataforma.** Disponível em: <https://www.eraser.io/>. Acesso em: 21 nov. 2024.

⁷DIAGRAMMING AI. **Endereço da plataforma.** Disponível em: <https://diagrammingai.com/>. Acesso em: 24 nov. 2024.

com o LLM de forma gratuita e sem limitações, permitindo que os participantes pudessem, quando desejado, iterar sobre os resultados obtidos.

A partir dos critérios listados, as duas primeiras opções desconsideradas foram Miro e Diagramming AI. Por permitirem a geração de apenas 10 diagramas em suas versões gratuitas, essas ferramentas poderiam potencialmente limitar a execução do experimento. Em seguida, DiagramGPT foi removida da disputa considerando os diagramas UML que é capaz de gerar no presente momento, que limita-se a um tipo e que não cumpre com o que está sendo analisado nesse estudo. Dessa forma, Lucid GPT foi a ferramenta de geração de diagramas baseada em LLM escolhida para utilização na primeira etapa do experimento. A Tabela 3.1 sintetiza as informações apresentadas nesse subcapítulo sobre cada uma das ferramentas analisadas.

Tabela 3.1 – Características das ferramentas analisadas

	<i>Modelo utilizado</i>	<i>Tipos de diagrama UML</i>	<i>Limitação versão gratuita</i>
Miro	Não listado	UML classes, UML sequência	Sim. 10 diagramas por mês
Lucid GPT	ChatGPT (versão atualizada do modelo GPT)	UML atividades, UML classes, UML sequência	Não
DiagramGPT	GPT-4	UML sequência	Não
Diagramming AI	GPT-4	UML caso de uso, UML classes, UML estado, UML sequência	Sim. 10 diagramas por mês

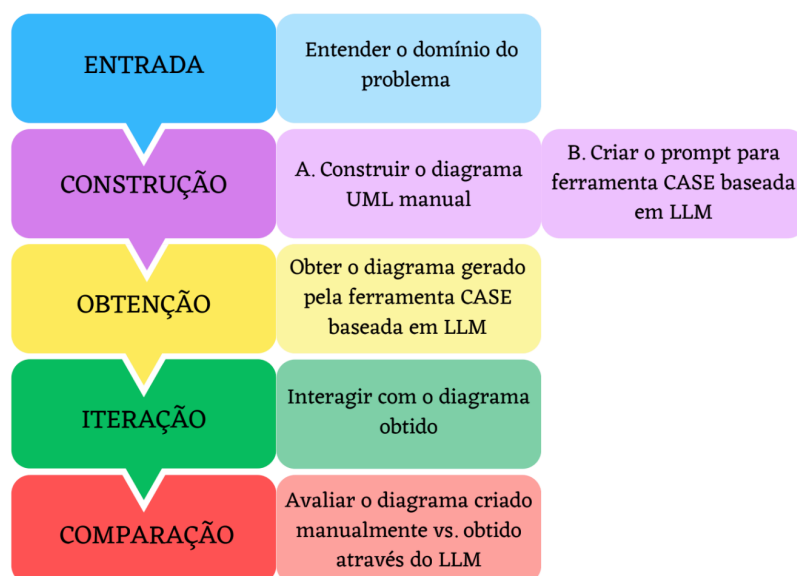
Fonte: O Autor.

3.2 Primeira Fase: Estudo de Caso

Nessa etapa, investigou-se o desempenho da ferramenta Lucid GPT para geração de diagramas UML de atividades, comparando o resultado obtido através do LLM com versões produzidas manualmente para a mesma descrição de caso de uso. Para isso, conduziu-se um estudo onde 24 alunos de graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul foram solicitados a criar diagramas UML de atividades para modelar um

caso de uso, e então utilizar a mesma definição de caso de uso para gerar diagramas UML de atividades através da Lucid GPT. Além disso, foi requisitado que esses estudantes avaliassem quão completo e preciso foram os diagramas obtidos através da ferramenta em comparação com os que desenharam manualmente. A Figura 3.1 ilustra as atividades realizadas durante essa etapa do estudo.

Figura 3.1 – Atividades realizadas durante o estudo de caso



Fonte: O Autor.

Para avaliar o desempenho do modelo LLM na realização da tarefa, foram utilizados como critérios a correspondência de elementos do diagrama (como atividade, transição e decisão), se esses elementos estavam de acordo com as convenções de notação do padrão UML, se ações e transições importantes do sistema estavam presentes no diagrama e nenhuma parte importante do fluxo descrito no caso de uso foi omitida. Essa avaliação foi feita pelos estudantes respondendo a um questionário contendo 10 perguntas, dos tipos aberta, comparativa e escala Likert de 5 pontos (1=discordo totalmente, 3=neutro, 5=concordo totalmente). A Tabela 3.2 lista as perguntas respondidas nessa etapa do estudo.

Tabela 3.2 – Perguntas da primeira fase do estudo

<i>Pergunta</i>	<i>Tipo</i>
O uso do LLM para gerar diagramas UML aumenta minha produtividade em projetos de software (em relação a velocidade, novos insights, etc.).	Escala Likert
A partir de poucas interações para refinar o prompt com a ferramenta CASE, pude obter o diagrama UML.	Escala Likert
O diagrama UML gerado pela ferramenta é correto em termos de notação e sintaxe, conforme as convenções do padrão UML.	Escala Likert
As representações dos elementos de software (como atividade, transição e decisão) são precisas e corretas no diagrama de atividades gerado pela ferramenta.	Escala Likert
O diagrama UML gerado é completo e não omite informações importantes sobre o caso de uso, cobrindo todos os aspectos relevantes do sistema modelado.	Escala Likert
O diagrama gerado pela ferramenta Lucid GPT é organizado, legível e fácil de entender.	Escala Likert
Estou satisfeito com a qualidade do diagrama UML gerado pela ferramenta Lucid GPT.	Escala Likert
Eu recomendaria para outros desenvolvedores o uso do Lucid GPT para geração de diagramas UML.	Escala Likert
Na sua opinião o diagrama manual está mais, menos ou igualmente completo e preciso em comparação ao diagrama gerado pelo Lucid GPT?	Comparativa
Você poderia contar como foi a sua experiência considerando dificuldades e oportunidades na elaboração dos diagramas manual e do gerado automaticamente com o modelo de LLM?	Aberta

Fonte: O Autor.

A análise das respostas à pergunta aberta foi feita da seguinte forma, conforme descrito em Strauss e Corbin (2014): 1) identificaram-se códigos relevantes no contexto do uso do LLM para geração de diagramas UML (*initial coding*), 2) analisou-se o relacionamento entre esses códigos e 3) identificaram-se temas para agrupar os códigos encontrados (*focused coding*). A síntese desses é uma lista dos benefícios e dificuldades reportados pelos participantes sobre o uso da ferramenta CASE baseada em LLM para geração de diagramas UML.

3.3 Segunda Fase: LLM para Modelagem de Novos Casos de Uso

Nessa segunda fase, o mesmo grupo de estudantes trabalhou em conjunto com um LLM para modelarem um novo caso de uso, no contexto de uma aplicação para clínica veterinária. Foi-lhes solicitado a gerarem um caso de uso que integrasse o sistema projetado de forma a torná-lo inovador, buscando incentivar o uso da criatividade para realização dessa tarefa e com intuito de avaliar a eficiência do LLM em produzir ideias criativas, auxiliando no processo de geração desse novo caso de uso.

O modelo LLM escolhido para utilização nessa etapa foi o ChatGPT 4, considerando os resultados de Bellemare-Pepin *et al.* (2024), onde apresentou-se como tendo o maior potencial criativo entre os modelos verificados, contribuindo diretamente com o objetivo dessa fase do estudo.

Para realização dessa etapa, os estudantes descreveram o contexto para o LLM, solicitaram que produzisse ideias e iteraram sobre os resultados até sentirem-se satisfeitos com o caso de uso obtido. Por fim responderam a um questionário de 6 perguntas, onde avaliaram sua percepção do impacto em sua criatividade produzido pelo uso do ChatGPT, bem como a qualidade, do ponto de vista criativo, das ideias produzidas pelo modelo LLM. A Tabela 3.3 lista as perguntas respondidas no questionário dessa etapa, onde a escala Likert utilizada foi de 5 pontos.

Tabela 3.3 – Perguntas da segunda fase do estudo

<i>Pergunta</i>	<i>Tipo</i>
O uso do LLM como parceiro no processo de modelagem potencializou minha criatividade sugerindo novas abordagens na especificação dos requisitos (considere criatividade como a habilidade de produzir ideias novas e apropriadas).	Escala Likert
O modelo de LLM apresentou diferentes alternativas, abordando o problema de diferentes ângulos.	Escala Likert
As ideias geradas pelo modelo de LLM são únicas, não convencionais.	Escala Likert
O modelo de LLM conseguiu refinar ideias, adicionando detalhes sobre o estudo de caso modelado tornando-o mais elaborado.	Escala Likert

As ideias geradas pelo modelo de LLM são viáveis e úteis para utilizar no contexto do problema apresentado.	Escala Likert
Você poderia descrever qual foi sua abordagem de prompts para obtenção de uma resposta considerada razoável e aceitável para o caso de uso modelado (e.g. número de interações para gerar o “melhor” prompt, adicionou e/ou detalhou com outras informações, etc.)?	Aberta

Fonte: O Autor.

4 ANÁLISE DE RESULTADOS

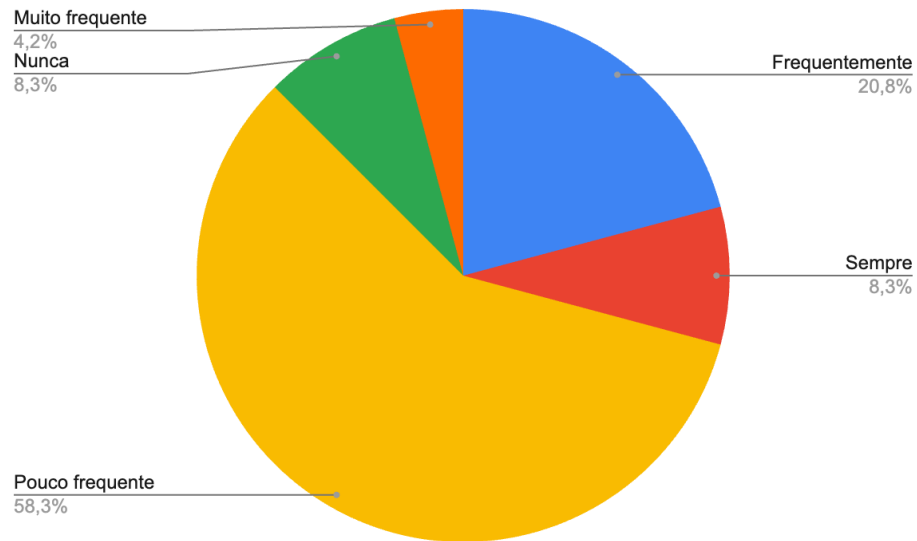
Nesse capítulo são apresentados os resultados do experimento conduzido. Primeiramente, é feita uma descrição demográfica dos participantes do estudo. Em seguida, são examinados os resultados que tratam da PP1. Finalmente, são analisados os dados coletados para responder a PP2.

4.1 Demografia dos Participantes

O estudo contou com 24 participantes, estudantes de graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul, compreendendo 21 autoidentificados homens e 3 autoidentificadas mulheres. Quando perguntados sobre trabalharem ou terem trabalhado em uma função que envolve etapas do processo de desenvolvimento de software, 87,5% responderam que já haviam tido contato e trabalhado em alguma das etapas da engenharia de software (analista, *product owner*, programador, arquiteto, testador, etc.). Entretanto, mais da metade dos participantes (54,2%) afirmou não ter experiência prévia com atividades de Engenharia de Requisitos.

Com relação a frequência com que utilizam LLMs para suporte em atividades específicas da ES, como naquelas envolvidas na etapa de engenharia de requisitos, 58,3% dos estudantes consideram que usam com pouca frequência, já 20,8% afirmaram utilizar frequentemente. A Figura 4.1 mostra a distribuição completa de respostas para essa pergunta, onde pode-se observar que o uso dos LLMs ainda é baixo considerando a prevalência dos resultados dada pelos participantes que utilizam de forma pouco frequente, o que indica uma baixa adesão ou crença dos participantes quanto ao possível suporte e agilidade oferecidos pelos LLMs durante a modelagem de requisitos. Uma razão para esses resultados possivelmente é o não conhecimento dos participantes de que existem ferramentas CASE que já incorporam LLMs para auxiliar na construção de modelos e diagramas UML durante as atividades da engenharia de requisitos. Essa baixa adesão demonstrada corrobora com Fan *et al.* (2023) e Wang *et al.* (2024), sobre a pouca exploração do uso de LLMs para uma das fases cruciais no desenvolvimento de software - análise, especificação e validação de requisitos.

Figura 4.1 – Frequência do uso de LLMs em atividades de engenharia de software



Fonte: O Autor.

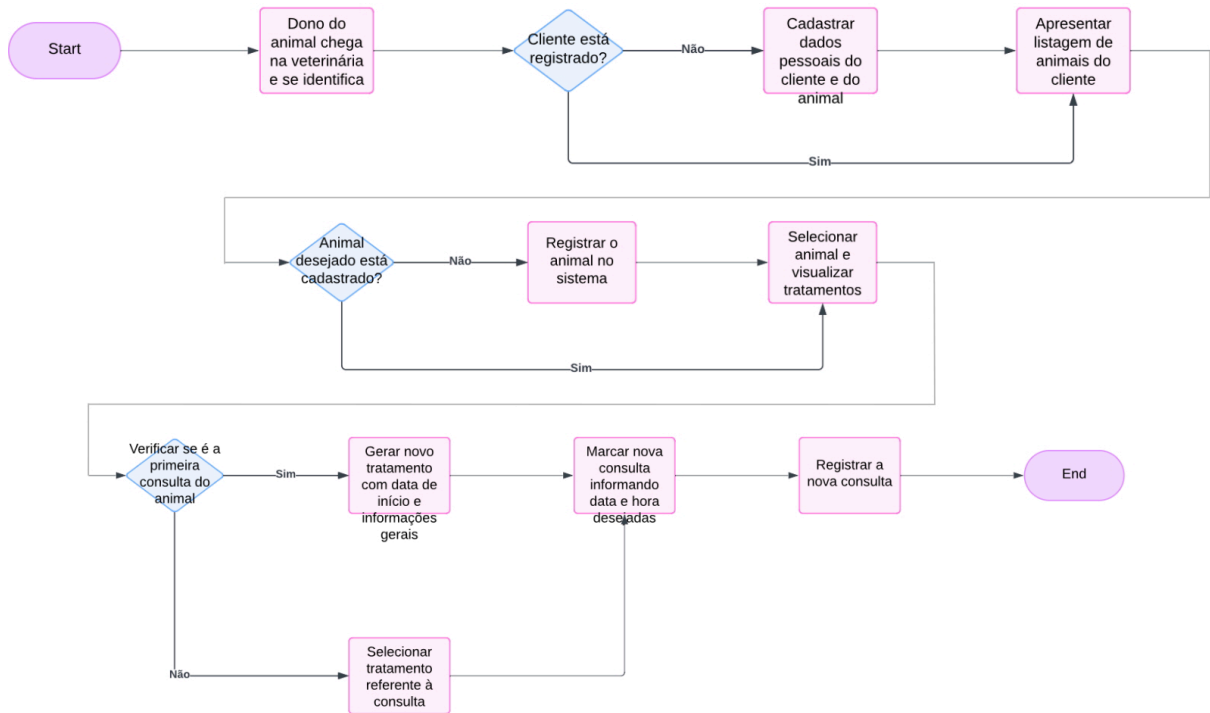
4.2 Resultados da Primeira Fase

Nessa subseção será feita a análise dos resultados da primeira fase do estudo, sobre o uso da ferramenta Lucid GPT para geração de diagramas UML de atividades. São apresentados os dados sobre a avaliação da estrutura dos diagramas gerados através da ferramenta, bem como a satisfação dos estudantes sobre esses diagramas, e também são discutidos temas levantados nas respostas da pergunta aberta presente no questionário.

A Figura 4.2 apresenta um diagrama resultante obtido durante o experimento através da Lucid GPT, para modelagem do caso de uso cuja descrição encontra-se no Anexo A do presente trabalho. As duplas de participantes obtiveram um resultado similar a esse (sendo um deles a própria figura em questão), com pequenas variações visuais, e em média foram necessárias 2 iterações de *prompt* para alcançar o resultado final.

Considerando que podem haver múltiplas formas para modelar o contexto central do caso de uso - clínica veterinária, especificamente dentro do escopo de marcação de consultas e gerenciamento de usuários, um “gabarito” foi disponibilizado para os participantes compararem suas respostas.

Figura 4.2 – Diagrama resultante obtido através da Lucid GPT



Fonte: O Autor.

A Tabela 4.1 a seguir mostra os códigos e temas identificados a partir da análise temática das respostas abertas fornecidas pelos participantes nessa etapa do estudo, e que serão discutidos nos subcapítulos que seguem.

Tabela 4.1 – Lista de temas e códigos identificados para as respostas abertas

<i>Temas</i>	<i>Códigos</i>
Rapidez na geração automática do diagrama	Facilidade no uso da ferramenta Lucid GPT para geração de diagramas UML de atividades.
	Otimização do processo de modelagem de software.
	Agilidade na elaboração de diagramas.
	Praticidade na criação de diagramas, comparado ao uso de ferramentas CASE ou desenho manual.
Necessidade de revisão dos resultados por humanos	Ajustes no prompt para obtenção de um diagrama de qualidade.
	Uso do diagrama gerado como referência ou versão inicial, sobre a qual ajustes e melhorias podem ser feitos.
	Impossibilidade de confiar completamente no LLM para desenvolvimento do diagrama.
	Dificuldade do LLM em reconhecer escolhas de modelagem erradas.
Contraste entre criação manual e automatizada do diagrama	Dificuldade em identificar ordem das ações e decisões na diagramação manual, simplificada pela geração automática.
	Apesar de exigir mais tempo para criação, os diagramas manuais são mais precisos e elaborados.

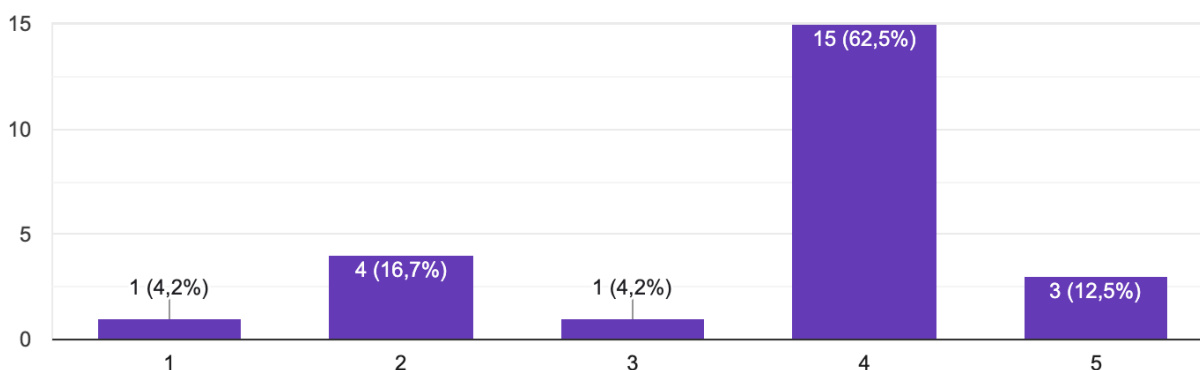
Conformidade com padrões de notação UML	O LLM apresentou dificuldades em seguir o padrão de notação para diagramas de atividades.
	Quando solicitado a organizar o diagrama utilizando swimlanes, o resultado obtido foi de baixa qualidade.
Impacto dos prompts no resultado	O diagrama gerado é limitado pela qualidade do prompt utilizado.
	Múltiplas iterações melhorando o prompt foram necessárias para alcançar um diagrama de qualidade.
	Para obter bons diagramas, é necessário um prompt grande e bastante descritivo.
Literalidade do diagrama	O LLM utiliza de maneira direta os passos da descrição de caso de uso para gerar o diagrama.
	Atividades dos diagramas gerados automaticamente menos detalhadas que as dos criados manualmente.
Diferentes pontos de vista do diagrama	Oportunidade de ter uma nova perspectiva das ações e decisões do diagrama de atividades modelado.
	Possibilidade de gerar rapidamente diferentes configurações de diagrama, até obter aquela que parece melhor modelar o caso de uso.

Fonte: O Autor.

4.2.1 Avaliação da Estrutura do Diagrama Gerado

Avaliando os resultados referentes a clareza, completude, estrutura e precisão dos diagramas gerados através da Lucid GPT, considerando a escala Likert utilizada e mencionada no subcapítulo 3.2, onde os valores 4 e 5 representam respectivamente concordo e concordo totalmente, os participantes concordaram em maioria (75%) que o diagrama gerado pela ferramenta é correto em termos de notação do padrão UML, conforme mostrado na Figura 4.3.

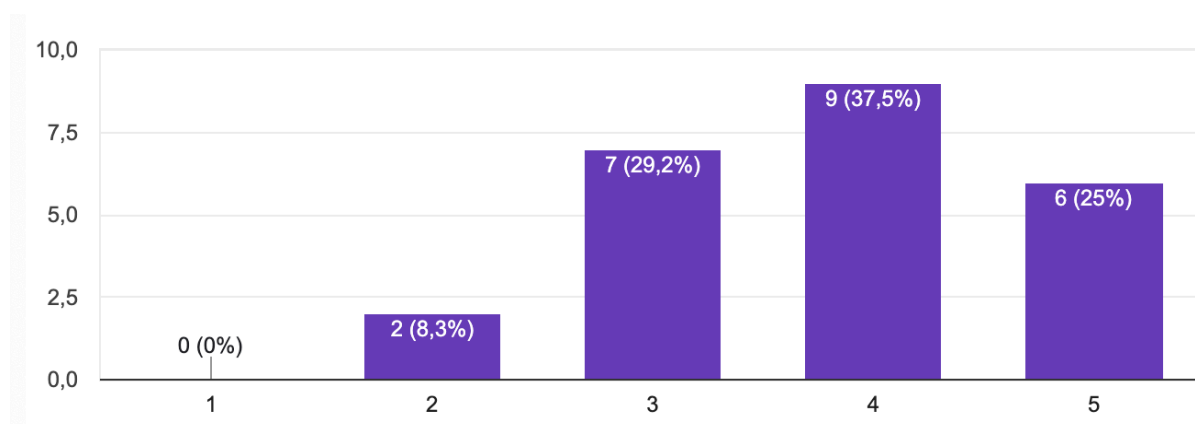
Figura 4.3 – Conformidade do diagrama gerado com a notação UML.



Fonte: O Autor.

Com relação aos elementos de software presentes no diagrama, como atividade, transição e decisão, 37,5% dos participantes concordaram que os diagramas gerados pela Lucid GPT representam de forma precisa e correta esses elementos, enquanto 29,2% permaneceram neutros nessa pergunta. A figura 4.4 mostra a distribuição de respostas para essa pergunta.

Figura 4.4 – Precisão da representação dos elementos de software

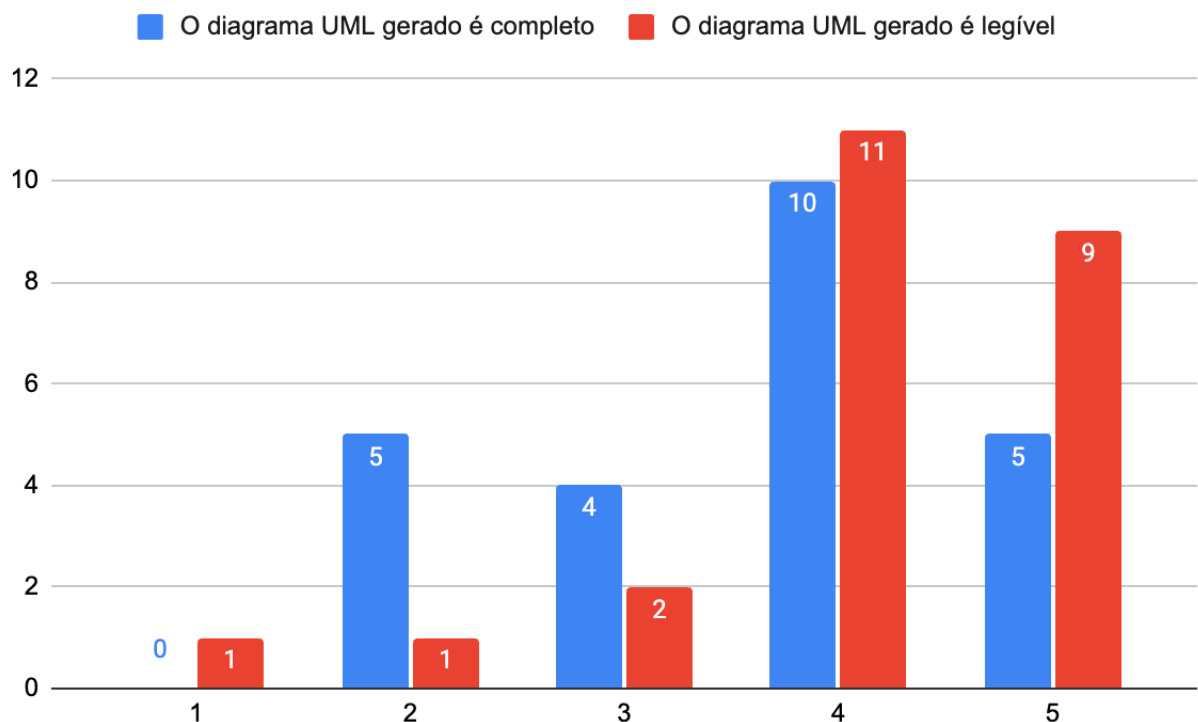


Fonte: O Autor.

Tratando-se de completude, 41,7% dos participantes concordaram que o diagrama gerado é completo e não omite informações importantes sobre o caso de uso, e 20,8% concordaram totalmente com essa afirmação. Já sobre organização e legibilidade, 83,3% dos estudantes responderam que concordam parcialmente ou totalmente que o diagrama gerado pela Lucid GPT é organizado, legível e fácil de entender. A Figura 4.5 mostra os resultados obtidos para essas duas perguntas. É válido destacar que, embora o diagrama UML de atividades possa ser considerado um diagrama simples para estruturar comportamentos sequenciais ou paralelos dentro de um sistema, também é bastante útil para facilitar a

compreensão do fluxo de processos entre *stakeholders*, desenvolvedores e analistas de requisitos. Analisando os dados obtidos através do questionário, a alta prevalência dos participantes em concordar parcialmente ou totalmente quanto a acurácia da ferramenta CASE baseada em LLM pode se dar pela facilidade do próprio contexto do caso de uso utilizado para modelagem, bem como pela baixa complexidade em realizar seu mapeamento automaticamente.

Figura 4.5 – Completude e legibilidade dos diagramas gerados

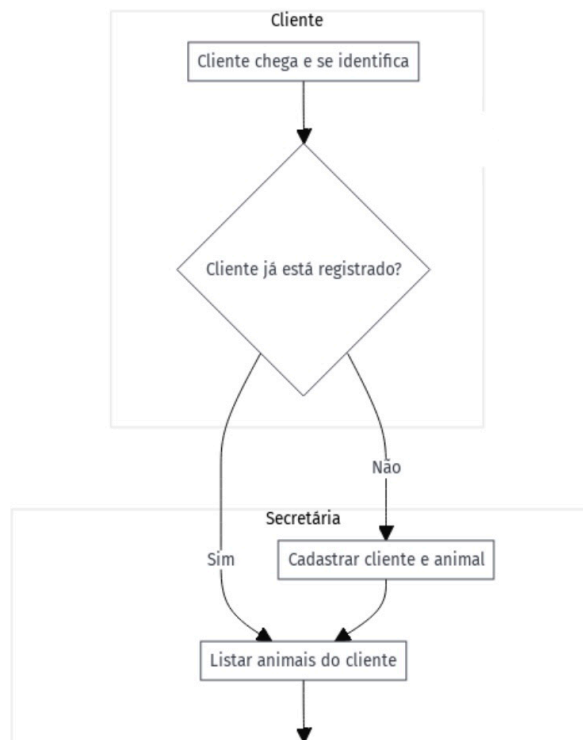


Fonte: O Autor.

Analisando as respostas à pergunta aberta, onde os participantes foram solicitados a refletir e compartilhar sobre a experiência ao comparar o processo de modelagem apoiado pelo uso do LLM e sem o seu uso, e reforçando relatos coletados durante a execução do experimento, os participantes ficaram insatisfeitos com o resultado apresentado quando solicitaram à ferramenta que incluísse raias (ou *swimlanes*) no diagrama, estrutura essa que faz parte da notação UML para diagrama de atividades. O seguinte trecho da resposta do participante 23 (P23) menciona esse tópico: “Quando tentamos dar um *prompt* mais detalhado para a criação das “raias”, o resultado não foi muito positivo, pois ele criou raias em um formato diferente do que estávamos esperando.”. Um dos resultados apresentados pela Lucid GPT quando solicitada a incluir *swimlanes* pode ser observado na Figura 4.6. Nela, pode-se perceber que apesar de haver uma tentativa de separação lógica entre “Cliente” e “Secretária”,

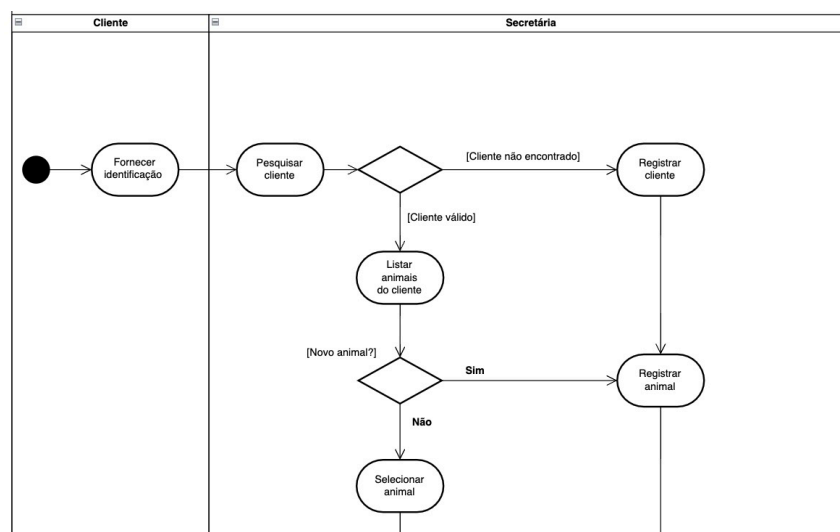
com retângulos delimitando as atividades de cada ator, essa separação não está correta, visto que a decisão “Cliente já está registrado?” deveria fazer parte do fluxo da “Secretária”. Além disso, a formatação das raias não está de acordo com a utilizada pelo padrão UML, onde o nome dos atores são apresentados no topo ou na extremidade esquerda do diagrama, para melhor legibilidade. A Figura 4.7 mostra como as raias foram utilizadas para modelar as mesmas atividades do caso de uso no “gabarito” do cenário.

Figura 4.6 – Resultado da inclusão de raias pelo Lucid GPT



Fonte: O Autor.

Figura 4.7 – Disposição das raias no “gabarito” do cenário



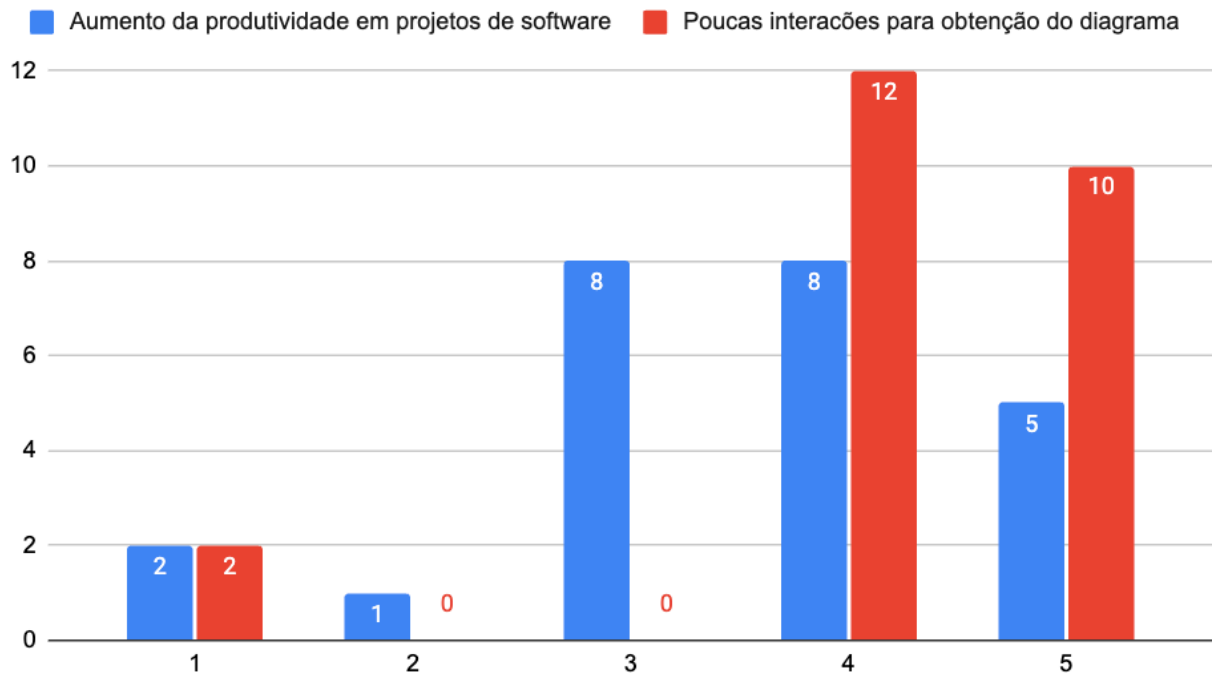
Fonte: O Autor.

Sendo assim, os resultados quantitativos obtidos mostram que a ferramenta Lucid GPT apresentou bom desempenho em termos de organização, estrutura e representação de elementos de software na geração de diagramas UML de atividades. Entretanto, ainda existem pontos de melhoria no aspecto de conformidade com a notação UML. Conforme observado nesse subcapítulo, a inserção de raias no diagrama mostrou-se um ponto fraco da ferramenta.

4.2.2 Satisfação com o Uso do LLM

Sobre o uso do LLM para gerar diagramas UML aumentar a produtividade em projetos de software, obteve-se 4 como mediana das respostas à essa pergunta, o que demonstra que a maioria dos participantes concordam que o uso da ferramenta tem impacto positivo em sua produtividade, tendo ainda 20,8% concordado totalmente com essa afirmação. Esse resultado está alinhado com o obtido por Peng *et al.* (2023), que também determinou um impacto positivo na produtividade em atividades do processo de produção de software a partir do uso de LLMs. Além disso, os participantes concordaram de forma quase unânime (91,7%) que poucas interações no ajuste dos *prompts* com a ferramenta foram necessárias para obtenção do diagrama UML que representasse de forma adequada o cenário modelado. Lembrando que um possível “gabarito” foi apresentado aos participantes para que essa análise crítica frente aos resultados gerados pelo LLM pudesse ser realizada. A Figura 4.8 mostra a distribuição de respostas para ambas essas perguntas.

Figura 4.8 – Aumento da produtividade e poucas interações para obter o diagrama

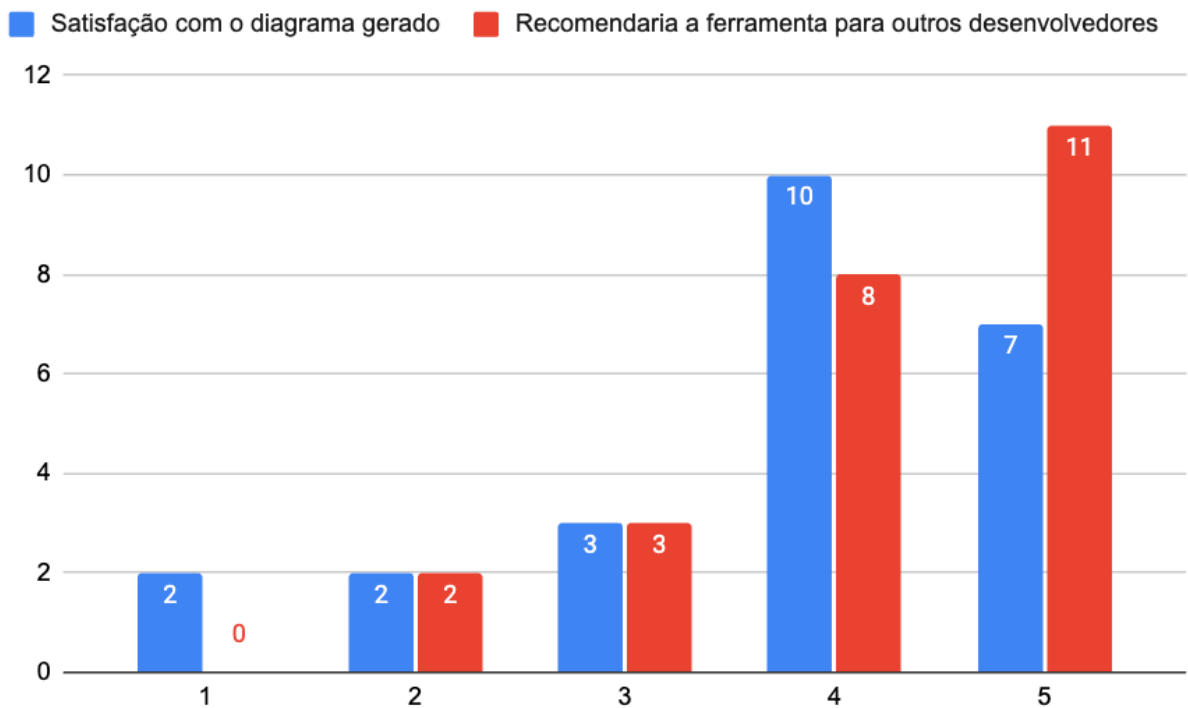


Fonte: O Autor.

As repostas à pergunta aberta corroboram com os resultados quantitativos obtidos. O seguinte relato do participante 18 (P18) demonstra a facilidade no uso da ferramenta Lucid GPT para geração de diagramas, e sua capacidade de impactar positivamente a produtividade: “A ferramenta é de fácil uso e pode otimizar bastante o tempo de desenvolvimento de software.”. Seguindo essa linha de pensamento, o seguinte trecho da resposta fornecida pelo P20 também destaca a praticidade na criação de diagramas através da ferramenta CASE baseada em LLM, quando comparada ao uso de ferramentas de modelagem tradicionais ou ao desenho manual: “O uso de uma LLM acabou sendo mais prático e fácil, assim que muito tempo é perdido apenas utilizando a ferramenta case (ou mesmo escrevendo em papel).”.

Em termos de satisfação, 70,9% dos participantes concordaram parcialmente ou totalmente com estarem satisfeitos com a qualidade do diagrama gerado pela Lucid GPT. Além disso, responderam em maioria (79,1%) que recomendariam o uso da ferramenta a outros desenvolvedores para geração de diagramas UML. A Figura 4.9 mostra os resultados obtidos para essas duas perguntas.

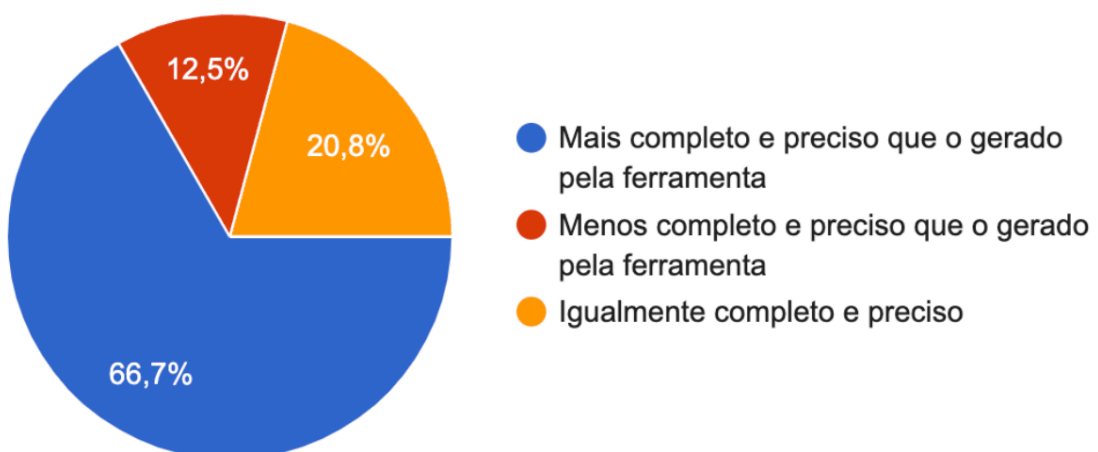
Figura 4.9 – Satisfação com o diagrama e recomendação a outros desenvolvedores



Fonte: O Autor.

Entretanto, quando solicitados a compararem em termos de completude e precisão o diagrama que elaboraram manualmente durante o estudo com o que geraram através da Lucid GPT, 66,7% dos participantes opinaram que o diagrama manual estava mais completo e preciso que o gerado pela ferramenta, conforme mostra a Figura 4.10.

Figura 4.10 – Comparação entre o diagrama manual e o gerado pela Lucid GPT



Fonte: O Autor.

O fator que mais contribuiu para esse resultado certamente foi a literalidade dos diagramas gerados pela ferramenta CASE baseada em LLM, com pouco detalhamento das atividades que modelam o caso de uso. Isso é reforçado pelo seguinte trecho da resposta

fornecida pelo P14: “Os passos do diagrama de atividades também precisam estar bem definidos, visto que a LLM os utilizou de maneira direta no diagrama plotado”. A resposta do P17 reitera esse fato: “O gerado pelo LLM foi feito rapidamente, porém com menos detalhes que o manual”. Essa questão também foi identificada por Ferrari, Abualhaija e Arora (2024), que avaliaram a geração de diagramas UML por LLMs e reportaram atividades excessivamente sumarizadas nos diagramas obtidos. Esse resultado aponta para uma possível dificuldade dos modelos LLM em elaborar pequenos passos que compõem a execução de uma grande ação.

Outro tópico discutido nas respostas à pergunta aberta foi o impacto do *prompt* no diagrama resultante. Participantes destacaram que o diagrama gerado é limitado pela qualidade do *prompt* utilizado, e que para se obter bons diagramas, é necessário um *prompt* extenso e bastante descritivo. Isso está diretamente relacionado com a questão discutida no parágrafo anterior. Considerando que o LLM utiliza a entrada fornecida de forma direta para elaboração das atividades, quanto mais descritivo for esse *prompt*, mais detalhado será o diagrama gerado. O seguinte trecho da resposta fornecida pelo P14 aborda esse tema: “O modelo gerado por LLMs é bastante limitado pela qualidade do prompt utilizado. Atingir a divisão por swimlanes, por exemplo, levou alguns passos. Para alcançar o diagrama necessário, é necessário um prompt grande e bastante descritivo.”.

Analisando os *prompts* fornecidos ao final do estudo, pode-se observar que alguns participantes elaboraram uma descrição textual baseada no caso de uso para fornecer como entrada para o LLM. Entretanto, a maioria das duplas (66,6%) utilizou como primeiro *prompt* a descrição exatamente como fornecida no enunciado da atividade (presente no Anexo A), e a partir do resultado inicial continuaram interagindo com o modelo LLM solicitando ajustes até que obtivessem o resultado esperado. Nesse contexto, alguns participantes mencionaram a necessidade de revisão humana dos resultados obtidos. No trecho a seguir, adaptado da resposta fornecida pelo P20, observa-se que apesar de ser uma boa ferramenta auxiliar, não é possível confiar completamente no LLM para construção do diagrama: “Eu não confiaria na LLM completamente para desenvolver o diagrama, mas acho que pode ser uma boa ferramenta de auxílio.”. O P21 também menciona em sua resposta uma dificuldade do LLM em reconhecer escolhas de modelagem erradas: “É razoável notar que muitas vezes situações

que parecem obviamente erradas para humanos não são tão facilmente reconhecidas pela LLM utilizada.”.

Apesar disso, os estudantes apontaram o uso do LLM como uma boa opção para obtenção de uma versão inicial do diagrama, tanto para ser utilizada como referência quanto para ser refinada, manualmente ou através de novos *prompts*, para alcançar uma versão final. O uso de LLMs como ferramenta de suporte para geração de diagramas UML também foi apresentado por Wang *et al.* (2024) como uma das implicações de seu trabalho. O P20 reforça esse ponto em sua resposta: “A LLM permite uma visão geral do diagrama em poucos segundos. Mesmo que o resultado não seja ideal, é possível analisar o diagrama adquirido e fazer ajustes manualmente, salvando tempo e trabalho.”. A resposta fornecida pelo P8 também segue essa ideia: “Mesmo se faltar algo no diagrama criado pelo GPT, podemos realizar apenas alguns ajustes manualmente ao invés de todo o diagrama do zero”. É válido pontuar que o fato dos participantes terem primeiramente refletido sobre o caso de uso e construído o diagrama manualmente certamente contribuiu para que chegassem a essa conclusão.

Finalmente, outro ponto positivo relatado pelos participantes sobre o uso do LLM nesse contexto foi a simplificação do processo de identificação da ordem das ações e decisões, conforme pode-se observar na seguinte resposta do P15: “O diagrama manual tem por maior dificuldade identificar qual a ordem em que as ações devem ser feitas e posicionamento correto das decisões. O diagrama gerado automaticamente simplifica esse processo”. Além disso, o uso da ferramenta oferece a oportunidade de ter uma nova perspectiva das ações e decisões do diagrama modelado. Isso é particularmente valioso para indivíduos no processo de aprendizagem dos conceitos de modelagem de requisitos, visto que podem rapidamente obter um novo diagrama com outro ponto de vista sobre o caso de uso, comparar com o diagrama que construíram e então refletir sobre qual a melhor forma de modelá-lo. O P16 destaca esse ponto em sua resposta: “Oportunidade de pensar de forma diferente no diagrama de atividades, ter uma nova perspectiva sobre os termos e ações”.

4.2.3 Retomando a Pergunta de Pesquisa 1

A partir dessa análise de resultados, pode-se então elaborar uma resposta para a PP1 desse estudo - “Qual é o potencial das ferramentas baseadas em LLM para auxiliar na

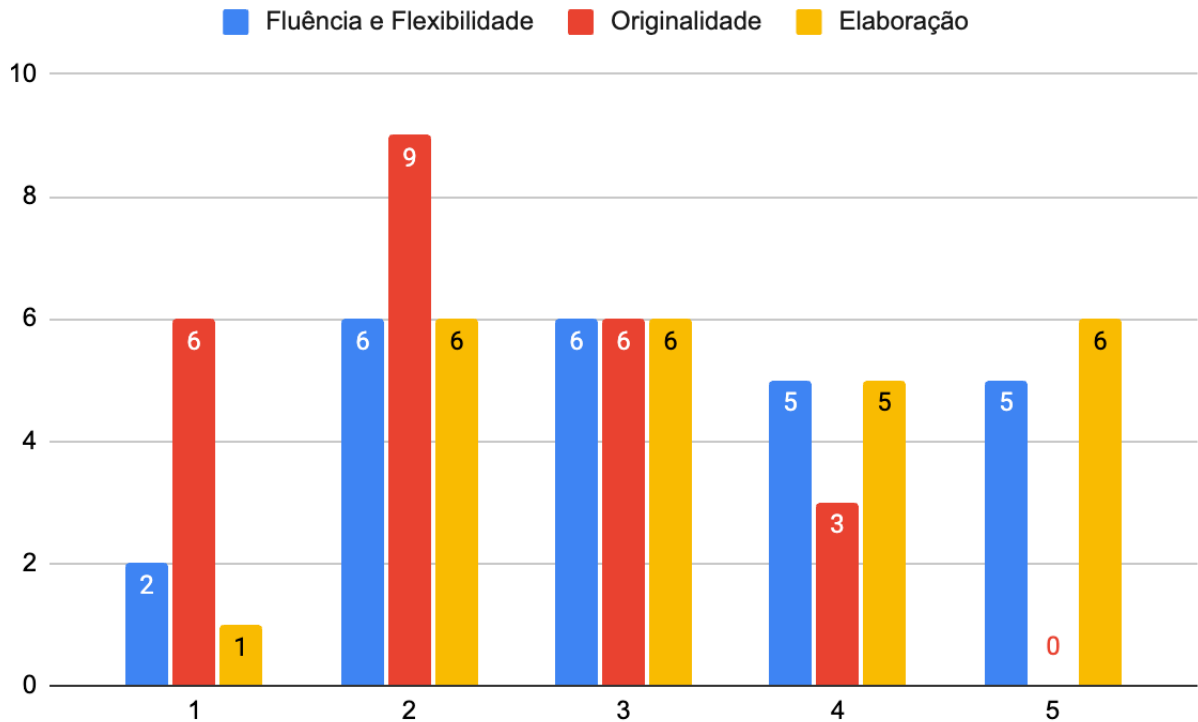
elaboração de diagramas UML em projetos de modelagem de software?”. O uso de ferramentas baseadas em LLM apresentou bom potencial para geração de versões iniciais de diagramas UML rapidamente, que podem ser refinadas buscando alcançar a modelagem definitiva. A adoção dessas ferramentas na etapa de modelagem da ER certamente impactaria positivamente a produtividade de desenvolvedores, principalmente quando comparado ao uso de ferramentas de diagramação tradicionais. Entretanto, também constatou-se que ainda não é possível confiar completamente nessas ferramentas para criação de diagramas. Os participantes reportaram incoerências nas escolhas de modelagem da Lucid GPT, ferramenta utilizada nessa etapa do estudo, que demandaram intervenção humana para correção. Ainda no contexto da Lucid GPT, tratando-se de diagramas UML de atividades, também foram identificados pontos de melhoria com relação a conformidade com a notação do padrão UML. Finalmente, determinou-se que o *prompt* fornecido à essas ferramentas tem impacto direto na qualidade do diagrama obtido.

4.3 Resultados da Segunda Fase

Através dos dados coletados, é possível fazer uma avaliação do desempenho criativo apresentado pelo ChatGPT durante a realização dessa segunda etapa, segundo os critérios utilizados por Zhao et al. (2024), que analisaram a criatividade de LLMs durante a execução de tarefas variadas. Esses critérios são fluência (refere-se a quantidade de ideias), flexibilidade (habilidade de abordar um dado problema de diferente ângulos), originalidade e elaboração (capacidade de adicionar detalhes, desenvolver nuances).

A Figura 4.11 apresenta os resultados das perguntas construídas para avaliação desses critérios. Sobre fluência e flexibilidade obteve-se mediana 3, com mais participantes avaliando positivamente (41,6%) do que negativamente (33,3%) o desempenho do LLM em termos dessas características. Com relação a originalidade das ideias a mediana obtida foi 2, tendo a maioria (62,5%) discordado que as ideias geradas pelo modelo foram únicas e não convencionais. Já na pergunta sobre a capacidade de elaboração do LLM obteve-se mediana 3, novamente com uma tendência mais positiva (45,8%) do que negativa (29,2%) nas respostas.

Figura 4.11 – Avaliação do desempenho criativo apresentado pelo ChatGPT



Fonte: O Autor.

Esses resultados podem ser percebidos nos casos de uso submetidos pelos participantes nos entregáveis da atividade. Apesar do modelo LLM ter apresentado um desempenho aceitável fornecendo e detalhando diferentes ideias dentro do contexto de aplicação para uma clínica veterinária, de forma geral essas ideias não foram originais ou inovadoras. Conforme pode-se observar na Tabela 4.2, a maior parte das ideias obtidas são casos de uso que incrementariam o sistema, mas ainda assim são usuais para uma aplicação de gerenciamento de clínica.

Tabela 4.2 – Ideias de novos casos de uso sugeridas pelo LLM

<i>Identificador</i>	<i>Ideias sugeridas pelo ChatGPT</i>
1	Agendamento de teleconsulta veterinária
2	Sistema automatizado de lembrete de vacinação
3	Consultas veterinárias sustentáveis e colaborativas
4	Monitoramento remoto da saúde do animal
5	Reagendamento de consultas

6	Acompanhamento do <i>status</i> da consulta
7	Gerenciamento de lista de espera
8	Agendamento de consultas de emergência
9	Geração de relatórios sobre consultas
10	Consulta virtual assistida por IA
11	Gestão de histórico de consultas do animal

Fonte: O Autor.

Ainda assim é possível identificar ideias criativas entre essas sugestões, como é o caso do monitoramento remoto da saúde do animal (identificador 4 da Tabela 4.2). A proposta seria aproveitar-se de dispositivos IoT (*Internet of Things*, ou internet das coisas) e de IA para fornecer um gerenciamento proativo da saúde dos animais de estimação, aprimorando a oferta de serviços veterinários e dando tranquilidade aos tutores. A Figura 4.12 mostra o *prompt* utilizado para obtenção desse caso de uso, onde em azul destaca-se o trecho que potencialmente incentivou que o LLM fosse criativo para execução da tarefa, bem como a descrição do fluxo principal fornecida pelo ChatGPT, sendo destacada em verde a inovação inserida pelo LLM no caso de uso gerado.

Figura 4.12 – *Prompt* e fluxo principal do caso de uso 4 gerado pelo LLM

Prompt: “In the following scenario: a customer contacts the vet to schedule an appointment for his pet. First the customer identifies themselves and then their pet. If the customer or the pet are not registered, the vet should register them immediately. After that, the vet should be able to see ongoing treatments for that pet. If no treatment exists, the vet creates one. Finally, the vet should ask the customer for a date and time, and then schedule an appointment on that date and time. Generate an **innovative use case** for this scenario.”

Main Success Scenario (Basic Flow):

1. The customer **connects their pet’s health monitoring device to the veterinary system** via a mobile app or web portal.
2. The **device sends periodic health data** (e.g., heart rate, activity levels, temperature) **to the veterinary system**.
3. The **system analyzes the incoming data for anomalies** using pre-set thresholds or **AI algorithms**.
 - **If the data is normal:** No action is required, and the system continues monitoring.
 - **If the data indicates an anomaly:** Proceeds to Step 4.
4. The system alerts the veterinarian with details of the anomaly (e.g., elevated heart rate, unusual inactivity).
5. The veterinarian reviews the anomaly and contacts the customer to discuss observations and recommend next steps (e.g., an in-person visit, teleconsultation, or lifestyle changes).
6. The customer and vet decide on follow-up actions, which may include scheduling an appointment.

Fonte: O Autor.

Outro resultado criativo obtido foi o caso de uso 10 da Tabela 4.2, que sugere a implementação de consultas virtuais assistidas por IA, permitindo que o cliente realize consultas preliminares com base nos sintomas do animal de forma remota. O *prompt* e o fluxo principal desse caso de uso são apresentados na Figura 4.13, onde faz-se uso do mesmo código de cores utilizado na Figura 4.12 para destaque de trechos.

Figura 4.13 – *Prompt* e fluxo principal do caso de uso 10 gerado pelo LLM

Prompt fornecido:

Faça: Etapa 3 5. Novo Caso de Uso: ○ Utilize o ChatGPT como parceiro para modelar um novo caso de uso, seguindo o contexto da clínica veterinária. ○ O novo caso de uso deverá integrar o sistema projetado de forma a torná-lo inovador. ○ Você pode basear esse novo caso de uso em critérios como acessibilidade, sustentabilidade, portabilidade, imprevisibilidade, envolvimento do usuário, conforto, etc.

Fluxo Principal:

1. O cliente acessa o sistema da clínica e seleciona "Consulta Virtual Assistida".
2. O sistema solicita a identificação do cliente e a seleção de um animal.
3. O cliente insere os sintomas observados no animal (ex.: tosse, falta de apetite).
4. A IA analisa os sintomas com base em um banco de dados veterinário integrado e sugere possíveis condições.
5. O sistema recomenda ações imediatas, como marcar uma consulta presencial, aplicar primeiros socorros, ou monitorar sintomas.
6. Se necessário, o cliente pode agendar diretamente uma consulta presencial com base na recomendação da IA.

Fonte: O Autor.

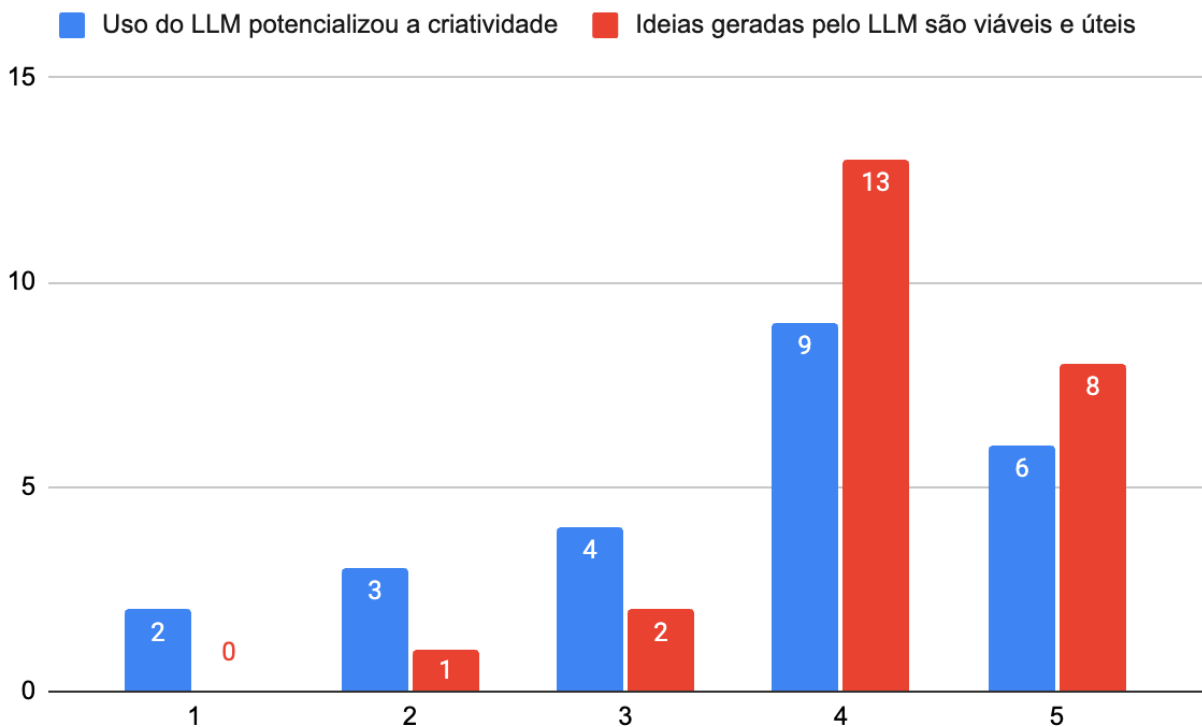
Nota-se que o *prompt* utilizado para geração do caso de uso apresentado na Figura 4.12 está em inglês. Isso se deve a alguns estudantes terem reportado que obtiveram melhores resultados nessa etapa interagindo em inglês com o ChatGPT. O seguinte trecho da resposta fornecida pelo P6 à pergunta aberta, que pedia que os participantes descrevessem a abordagem adotada para obtenção do caso de uso, destaca esse fato: “Inicialmente percebemos que o *prompt* sendo colocado em inglês trouxe uma eficiência muito maior na hora de gerar boas ideias de caso de uso”. Esse certamente é um bom ponto de investigação para trabalhos futuros, mas indícios nos resultados obtidos sugerem que isso pode ter ocorrido em casos isolados, e que o idioma utilizado na interação não foi o fator determinante para a qualidade do resultado, visto que a própria Figura 4.13 apresenta um bom resultado gerado em português.

Em contrapartida, um fator que certamente teve grande impacto no grau de criatividade dos casos de uso gerados foi o nível de clareza e detalhamento dos *prompts*. Os melhores resultados foram obtidos pelos participantes que contextualizaram o ChatGPT com a descrição utilizada na etapa 1 desse estudo, e então detalharam o que esperavam do novo caso de uso solicitado, em alguns casos mencionando até critérios a serem utilizados como base para essa elaboração. Isso também foi constatado por Zhao et al. (2024), que reportaram que o uso de *prompts* do tipo instrutivos, que fornecem descrição detalhada do que é esperado, leva a obtenção de resultados mais criativos por parte dos LLMs. Respostas à

pergunta aberta reforçam essa ideia, como é o caso do seguinte relato do P7: “Para gerar o caso de uso foi construído uma frase mais elaborada que trouxesse novas ideias para o sistema”. O P21 também segue essa linha em sua resposta: “Para o caso de uso, escolhi anexar a imagem da descrição textual do contexto do problema. A partir desta imagem, meus prompts envolveram indicar a tarefa a ser feita baseada no contexto passado.”.

Sobre o uso do LLM como parceiro no processo de modelagem, os participantes concordaram em maioria (62,5%) que isso potencializou sua criatividade. Além disso, a grande maioria (87,5%) também concordou que as ideias geradas pelo modelo LLM eram viáveis e úteis no contexto do problema apresentado. A distribuição de respostas para essas duas perguntas pode ser observada na Figura 4.14.

Figura 4.14 – LLM potencializando a criatividade e gerando ideias viáveis



Fonte: O Autor.

Refletindo sobre os resultados obtidos durante essa etapa pode-se concluir que, mesmo não apresentando o melhor desempenho criativo gerando ideias para novos casos de uso de forma direta, o LLM apresentou-se como um bom catalisador para construção dessas ideias. Trabalhando em conjunto com essas ferramentas, os desenvolvedores podem se aproveitar das ideias fornecidas pelo LLM para sair da inércia criativa e guiar seu pensamento em direção a novas ideias, de forma mais rápida que utilizando ferramentas criativas tradicionais. Esse tópico é abordado por Wan *et al.* (2024) ao apresentarem suas descobertas sobre o papel de

humanos e LLMs enquanto colaboram para execução de tarefas criativas. Os autores identificaram que quando os participantes de seu estudo não tinham nenhuma ideia, utilizaram o LLM para geração de novas ideias. Nesse cenário, o LLM tem a iniciativa e lidera o processo criativo. Tendo uma ideia em mente, esses indivíduos passaram a preferir utilizar o LLM como ferramenta para organizar e resumir seus pensamentos, passando assim a liderar o processo criativo.

Finalmente, é válido mencionar que o ChatGPT apresentou bom desempenho estruturando os casos de uso gerados. Todos eles apresentam descrição, pré-condições, pós-condições, atores primários e fluxo principal, bem como fluxos alternativos quando aplicáveis, seguindo a estrutura de caso de uso descrita por Booch, Rumbaugh e Jacobson (2006). Isso indica um potencial dos LLMs de serem integrados em processos de modelagem de requisitos visando simplificar e padronizar a documentação dessa etapa do processo de ER.

4.3.1 Retomando a Pergunta de Pesquisa 2

Feita a análise dos resultados da segunda etapa, pode-se então retomar a PP2 - “Como os LLMs contribuem no processo criativo para concepção de novos casos de uso em modelagem de software?”. Os resultados apresentados durante esse subcapítulo mostram que houveram boas ideias criativas para casos de uso geradas pelo LLM, entretanto não foram maioria. As respostas indicaram um baixo desempenho criativo do ChatGPT durante a execução da tarefa, com destaque para pouca originalidade das ideias produzidas. Também identificou-se que o *prompt* utilizado têm grande influência na qualidade criativa da resposta gerada pelo LLM, tendo *prompts* descritivos, que explicitam o contexto e detalham o que se espera que seja produzido, apresentando os melhores resultados.

Ainda assim, os participantes reportaram ter sido uma boa experiência trabalhar em conjunto com o ChatGPT para elaboração de um caso de uso, e que o LLM teve impacto positivo em sua criatividade sugerindo novas abordagens, através de ideias úteis e viáveis, para a modelagem do novo caso de uso. A partir disso, constatou-se que o uso de LLMs nesse contexto é mais proveitoso auxiliando na busca por inspiração, onde junta-se material para construção de novas ideias e diferentes pontos de vista sobre o problema são analisados.

4.4 Limitações do Estudo

O presente estudo certamente possui limitações. Uma delas é a possibilidade da amostra de participantes não representar completamente a população de desenvolvedores, considerando o tamanho dessa amostra e o fato dos participantes possuírem perfil e experiência similares. Considerando que os indivíduos que fizeram parte do estudo estavam passando pelo processo de aprendizagem de conceitos analisados no experimento, isso também pode ter impactado os resultados obtidos.

Além disso, o uso de uma única ferramenta CASE baseada em LLM na etapa 1, bem como o uso de um único modelo LLM na etapa 2, pode ter afetado a experiência dos usuários durante a execução do estudo, consequentemente influenciando nos resultados obtidos. Conforme descrito no Capítulo 3, existem outras ferramentas LLM para geração de diagramas UML, que possuem diferentes funcionalidades e que podem apresentar resultados diferentes. Adicionalmente, a análise da geração de um único tipo de diagrama UML também apresenta-se como uma limitação desse estudo, visto que é possível que ferramentas baseadas em LLM apresentem desempenhos diferentes a depender do tipo de diagrama sendo gerado. Finalmente, a simplicidade do contexto utilizado para modelagem nas duas etapas do estudo também pode ter impactado nos resultados encontrados, em especial na segunda etapa, onde avaliou-se o uso da criatividade para modelagem de novos casos de uso.

5 CONCLUSÃO

Ao longo desse trabalho, examinou-se o impacto do uso de LLMs em atividades do processo de engenharia de requisitos. Investigou-se como desenvolvedores avaliam o uso de ferramentas baseadas em LLM para criação de diagramas UML. Também, como esses indivíduos percebem a utilização de LLMs auxiliando na produção de ideias criativas para modelagem de novos casos de uso.

Foi realizada uma análise qualitativa das respostas fornecidas pelos participantes sobre sua experiência utilizando uma ferramenta CASE baseada em LLM para geração de diagramas UML de atividades. Os resultados mostraram que os participantes reconhecem um impacto positivo em sua produtividade causado pelo uso da ferramenta, entretanto ressaltaram a importância da ação humana corrigindo escolhas de modelagem incorretas e desenvolvendo *prompts* eficazes e que geram melhores resultados.

Os resultados da segunda etapa do estudo indicaram que o uso de LLMs como parceiros criativos na elaboração de casos de uso é mais eficiente quando aplicado para fornecer ideias que servem de inspiração para os desenvolvedores durante a modelagem. Também foi constatado um grande impacto do *prompt* utilizado no desempenho criativo do LLM, onde aqueles do tipo descritivos, que detalham o que se espera do resultado, apresentaram-se como os mais eficazes. Isso está alinhado com o que foi encontrado na etapa 1 desse estudo com relação a *prompts*, e reforça a ideia que existem tipos de *prompt* mais adequados a depender da tarefa que deseja-se executar.

As contribuições do presente trabalho passam pela exploração do campo emergente que é a aplicação de LLMs na engenharia de software. Como mostra o seguinte relato do participante 13 dessa pesquisa, os LLMs são capazes de surpreender com seus resultados, o que por sua vez indica que ainda há muito potencial a ser explorado a partir de sua utilização: “Eu nunca uso/usei LLMs para nada, então achei surpreendentemente indolor fazer ela fazer os diagramas.”. Além disso, contribuí estendendo a discussão sobre o impacto do uso de LLMs em atividades de engenharia de requisitos, etapa do processo de ES que tem recebido menos atenção no contexto do uso dessa tecnologia, principalmente quando comparada as etapas de desenvolvimento e testes.

Como trabalhos futuros, planeja-se diversificar o grupo de participantes e incluir desenvolvedores com diferentes níveis de experiência e de diferentes regiões do Brasil e do mundo, buscando analisar como a vivência profissional e as diferenças culturais impactam os resultados. A utilização de LLMs para geração de outros tipos de diagramas UML, ou como parceiro criativo em outras etapas da engenharia de requisitos (como por exemplo na elicitação de requisitos) também são possibilidades de extensão do presente trabalho. Finalmente, também pretende-se utilizar a engenharia de *prompt* como objeto de estudo, com propósito de aprofundar o conhecimento sobre as características de um bom *prompt* para geração de diagramas UML através de ferramentas baseadas em LLM.

5.1 Disponibilidade dos Dados

Os artefatos gerados no experimento conduzido e analisados através desse estudo estão publicamente disponíveis em Oliveira (2024).

REFERÊNCIAS

BECK, Kent *et al.* **Manifesto para Desenvolvimento Ágil de Software**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 19 dez. 2024.

KIM, Gene; HUMBLE, Jez; DEBOIS, Patrick; WILLIS, John. **The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations**. [S. L.]: It Revolution Press, 2016.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River: Prentice-Hall, 2010.

DEVELOPERS, Google For. **Introduction to Large Language Models**. Disponível em: <https://developers.google.com/machine-learning/resources/intro-llms>. Acesso em: 27 out. 2024.

FAN, Angela; GOKKAYA, Beliz; HARMAN, Mark; LYUBARSKIY, Mitya; SENGUPTA, Shubho; YOO, Shin; ZHANG, Jie M.. Large Language Models for Software Engineering: Survey and Open Problems. In: 2023 IEEE/ACM INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: FUTURE OF SOFTWARE ENGINEERING (ICSE-FOSE), 2023., 2023, Melbourne. **Proceedings [...]** . [S.L.]: Ieee, 2023. p. 31-53. Disponível em: <https://ieeexplore.ieee.org/document/10449667>. Acesso em: 27 out. 2024.

FERRARA, Emilio. GenAI against humanity: nefarious applications of generative artificial intelligence and large language models. **Journal Of Computational Social Science**. Los Angeles, p. 549-569. abr. 2024. Disponível em: <https://link.springer.com/article/10.1007/s42001-024-00250-1>. Acesso em: 27 out. 2024.

WAN, Qian; HU, Siying; ZHANG, Yu; WANG, Piaohong; WEN, Bo; LU, Zhicong. "It Felt Like Having a Second Mind": Investigating Human-AI Co-creativity in Prewriting with Large Language Models. **Proceedings Of The Acm On Human-Computer Interaction**, [S. L.], v. 8, n. 1, p. 1-26, abr. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2307.10811>. Acesso em: 28 out. 2024.

ZHAO, Shuyin. **GitHub Copilot now has a better AI model and new capabilities**. 2023. Disponível em: <https://github.blog/ai-and-ml/github-copilot/github-copilot-now-has-a-better-ai-model-and-new-capabilities/>. Acesso em: 28 out. 2024.

PENG, Sida; KALLIAMVAKOU, Eirini; CIHON, Peter; DEMIRER, Mert. **The Impact of AI on Developer Productivity: Evidence from GitHub Copilot**. 2023. Disponível em: <https://doi.org/10.48550/arXiv.2302.06590>. Acesso em: 28 out. 2024.

DIRIN, Amir; LAINE, Teemu. Examining the Utilization of Artificial Intelligence Tools by Students in Software Engineering Projects. In: INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED EDUCATION, 16., 2024, Angers. **Proceedings [...]** . [S. L.]:

Scitepress, 2024. p. 286-293. Disponível em: <https://www.scitepress.org/Documents/2024/127294/>. Acesso em: 29 out. 2024.

KOZOV, Vasil; IVANOVA, Galina; ATANASOVA, Desislava. Practical Application of AI and Large Language Models in Software Engineering Education. **International Journal Of Advanced Computer Science And Applications (Ijacs)**, [S. L.], v. 15, n. 1, p. 690-696, jan. 2024. Disponível em: <https://thesai.org/Publications/ViewPaper?Volume=15&Issue=1&Code=IJACSA&SerialNo=68>. Acesso em: 29 out. 2024.

HASSAN, Ahmed E.; OLIVA, Gustavo A.; LIN, Dayi; CHEN, Boyuan; JIANG, Zhen Ming. **Towards AI-Native Software Engineering (SE 3.0): A Vision and a Challenge Roadmap**. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2410.06107>. Acesso em: 29 out. 2024.

FERRARI, Alessio; ABUALHAIJA, Sallam; ARORA, Chetan. **Model Generation with LLMs: From Requirements to UML Sequence Diagrams**. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2404.06371>. Acesso em: 02 nov. 2024.

PRESSMAN, Roger S.. **Software Engineering: A Practitioner's Approach**. 7. ed. [S. L.]: McGraw-Hill Education, 2009.

LIU, Lin; LI, Tong; PENG, Fei. Why Requirements Engineering Fails: A Survey Report from China. In: IEEE INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE, 18., 2010, Sydney. **Proceedings [...]**. [S. L.]: Ieee, 2010. p. 317-322. Disponível em: <https://ieeexplore.ieee.org/document/5636549>. Acesso em: 03 nov. 2024.

FERNÁNDEZ, Daniel Méndez *et al.* Naming the pain in requirements engineering. **Empirical Software Engineering**, [S. L.], v. 22, n. 5, p. 2298-2338, out. 2017. Bimestral. Disponível em: <https://doi.org/10.1007/s10664-016-9451-7>. Acesso em: 03 nov. 2024.

ARORA, Chetan; GRUNDY, John; ABDELRAZEK, Mohamed. **Advancing Requirements Engineering through Generative AI: Assessing the Role of LLMs**. 2023. Disponível em: <https://doi.org/10.48550/arXiv.2310.13976>. Acesso em: 03 nov. 2024.

WANG, Beian; WANG, Chong; LIANG, Peng; LI, Bing; ZENG, Cheng. **How LLMs Aid in UML Modeling: An Exploratory Study with Novice Analysts**. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2404.17739>. Acesso em: 03 nov. 2024.

MAIDEN, Neil; GIZIKIS, Lexis. Where Do Requirements Come From? **Ieee Software**, [S. L.], v. 18, n. 5, p. 10-12, set. 2001. Bimestral. Disponível em: <https://ieeexplore.ieee.org/document/951486>. Acesso em: 04 nov. 2024.

SANTOS, Vitor; MAMEDE, Henrique; SILVEIRA, Clara; REIS, Leonilde. Methodology for Introducing Creativity in Requirements Engineering. **Procedia Computer Science**, [S. L.], v. 196, n. 1, p. 27-35, jan. 2022. Mensal. Disponível em: <https://doi.org/10.1016/j.procs.2021.11.069>. Acesso em: 04 nov. 2024.

STERNBERG, Robert J.; LUBART, Todd I. **Defying the Crowd: Cultivating Creativity in a Culture of Conformity**. [S. L.]: Free Press, 1995.

BODEN, Margaret A. **The Creative Mind: Myths and Mechanisms**. 2. ed. [S. L.]: Routledge, 2003.

SOSA, Ricardo; GERO, John S. Design and change: A model of situated creativity. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 18., 2003, Acapulco. **Approaches to creativity in artificial intelligence and cognitive science**. [S. L.]: International Joint Conferences On Artificial Intelligence, 2003. p. 25-34.

NGUYEN, Lemai; SHANKS, Graeme. A framework for understanding creativity in requirements engineering. **Information And Software Technology**, [S. L.], v. 51, n. 3, p. 655-662, maio 2009. Mensal. Disponível em: <https://doi.org/10.1016/j.infsof.2008.09.002>. Acesso em: 05 nov. 2024.

GIUNTA, Benito; BURNAY, Corentin; MAIDEN, Neil; FAULKNER, Stéphane. Creativity Triggers: Extension and empirical evaluation of their effectiveness during requirements elicitation,. **Journal Of Systems And Software**, [S. L.], v. 191, n. , p. 1-15, set. 2022. Mensal. Disponível em: <https://doi.org/10.1016/j.jss.2022.111365>. Acesso em: 05 nov. 2024.

MAIDEN, Neil; GIZIKIS, Alexis; ROBERTSON, Suzanne. Provoking creativity: imagine what your requirements could be like. **Ieee Software**, [S. L.], v. 21, n. 5, p. 68-75, set. 2004. Bimestral. Disponível em: <https://doi.org/10.1109/MS.2004.1331305>. Acesso em: 05 nov. 2024.

ZHAO, Yunpu; ZHANG, Rui; LI, Wenyi; HUANG, Di; GUO, Jiaming; PENG, Shaohui; HAO, Yifan; WEN, Yuanbo; HU, Xing; DU, Zidong; GUO, Qi; LI, Ling; CHEN, Yunji. **Assessing and Understanding Creativity in Large Language Models**. 2024. Disponível em: <https://arxiv.org/abs/2401.12491>. Acesso em: 06 nov. 2024.

BELLEMARE-PEPIN, Antoine; LESPINASSE, François; THÖLKE, Philipp; HAREL, Yann; MATHEWSON, Kory; OLSON, Jay A.; BENGIO, Yoshua; JERBI, Karim. **Divergent Creativity in Humans and Large Language Models**. 2024. Disponível em: <https://doi.org/10.48550/arXiv.2405.13012>. Acesso em: 06 nov. 2024.

LUCID. **AI in Lucid: Supercharge your workflow**. Disponível em: <https://lucid.co/platform/ai>. Acesso em: 21 nov. 2024.

ERASER. **DiagramGPT - AI diagram generator**. Disponível em: <https://www.eraser.io/diagramgpt>. Acesso em: 24 nov. 2024.

YIN, Robert K.. **Estudo de Caso: Planejamento e métodos**. 2. ed. [S. L.]: Bookman editora, 2001.

YIN, Robert K.. **Case Study Research: Design and Methods**. 5. ed. [S. L.]: Sage Publications, Inc, 2013.

STRAUSS, Anselm; CORBIN, Juliet. **Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory**. 4. ed. [S. L.]: Sage Publications, Inc, 2014.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **Uml - Guia do Usuário**. 2. ed. [S. L.]: Gen Ltc, 2006.

OLIVEIRA, Giuseppe. **Material Suplementar ao TCC**. Disponível em: https://drive.google.com/file/d/1nEqgI_Seo_OYZeF4f1xV4NrT-CNuls-c/view. Acesso em: 29 dez. 2024.

ANEXO A - DESCRIÇÃO DO CASO DE USO

O presente anexo apresenta a descrição do caso de uso utilizada pelos participantes do estudo para a etapa 1 do trabalho. Tanto para modelagem manual, quanto através da LLM, essa foi a descrição utilizada.

Crie um **Diagrama de Atividades** referente ao processo de geração de uma nova consulta de um determinado animal do sistema da clínica veterinária, levando em consideração as seguintes características:

- Ao chegar a veterinária, o dono do animal deve se identificar. Caso este não esteja registrado, a secretária deverá cadastrar os dados pessoais do cliente, bem como de seu animal.
- Ao consultar um cliente, uma listagem de todos os animais por ele possuídos será apresentada juntamente com seus dados pessoais. Caso o animal para o qual o cliente deseja marcar a consulta não esteja cadastrado, a secretária deverá registrá-lo.
- Ao selecionar um animal, a secretária pode visualizar todos os tratamentos do animal, já realizados ou ainda em andamento.
- A partir da listagem dos tratamentos, a secretária pode selecionar um destes, o que fará surgir juntamente com as informações gerais do tratamento, todas as consultas já realizadas durante o mesmo, permitindo também que outra consulta possa ser marcada, caso o tratamento em questão não esteja concluído.
- Se for a primeira consulta do animal, primeiramente um novo tratamento deve ser gerado, contendo a data de início do tratamento e identificando o animal e seu dono, bem como os sintomas gerais do animal. Caso não seja a primeira consulta deve-se selecionar o tratamento referente à consulta para depois marcá-la.
- Após selecionar o cliente, o animal para o qual será marcada a consulta e o seu tratamento, a secretária poderá marcar a consulta. Sempre que for registrar uma nova consulta, a secretária deve informar a data e a hora desejadas pelo cliente.