UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO - CIN

CAROLINA CARNEIRO REIS E SILVA

VALIDAÇÃO DE CRITÉRIOS DE ACEITAÇÃO USANDO LLM: UMA ANÁLISE BASEADA EM GUIDELINES DE UX

Projeto de Mestrado

ORIENTADOR: PROFª. DRª JÉSSYKA FLAVYANNE FERREIRA VILELA

Recife, PE - Brasil

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Silva, Carolina Carneiro Reis e.
Validação de critérios de aceitação usando LLM: uma análise baseada em Guidelines de UX / Carolina Carneiro Reis e Silva. - Recife, 2025.
155f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2025.
Orientação: Jéssyka Flavyanne Ferreira Vilela.
Inclui referências e apêndices.

1. Experiência do Usuário; 2. Histórias de Usuário; 3. Critérios de Aceitação; 4. Grandes Modelos de Linguagem; 5. Engenharia de Requisitos. I. Vilela, Jéssyka Flavyanne Ferreira. II. Título.

UFPE-Biblioteca Central

CAROLINA CARNEIRO REIS E SILVA

VALIDAÇÃO DE CRITÉRIOS DE ACEITAÇÃO
USANDO LLM:
UMA ANÁLISE BASEADA EM GUIDELINES DE UX

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciências da Computação, na área de concentração: Engenharia de Software e Linguagens de Programação.

Aprovado em: 31/07/2025.

BANCA EXAMINADORA

_____

Profa. Dra. Carla Taciana Lima Lourenço Silva Schuenemann
Centro de Informática / UFPE

_____

Profa. Dra. Tayana Uchôa Conte
Instituto de Computação / UFAM

_____

Profa. Dra. Jéssyka Flavyanne Ferreira Vilela
Centro de Informática/UFPE
(**orientadora**)

Este trabalho é dedicado primordialmente a Deus, que me presenteia todos os dias com a energia da vida, à minha mãe, por todo esforço, incentivo e amor, ao meu pai, por seu apoio, resiliência e exemplo de ser, e à minha irmã, pela reciprocidade, suporte e carinho.

AGRADECIMENTOS

Agradeço,

Aos meus pais, pelo apoio incondicional em todas as etapas da minha trajetória acadêmica e pela confiança depositada em cada uma de minhas escolhas.

À minha irmã, pela parceria, pelo incentivo e pela presença acolhedora nos momentos mais desafiadores.

Ao meu namorado, pelo companheirismo, paciência e compreensão durante os períodos de maior dedicação a este trabalho.

À minha orientadora, pela escuta atenta, pelas orientações técnicas e por conduzir este processo com seriedade, respeito e incentivo à autonomia.

À banca avaliadora, pelas contribuições valiosas durante a qualificação e defesa, que enriqueceram significativamente este estudo.

Aos amigos, do grupo de pesquisa DARE, pelo ambiente colaborativo, pelas trocas e pela inspiração gerada nos encontros e debates compartilhados.

A todos que, direta ou indiretamente contribuíram com este objetivo, deixo meu sincero agradecimento.

"A melhor maneira de prever o futuro é criá-lo."
Peter Drucker

RESUMO

[Contexto] A crescente complexidade dos projetos digitais têm motivado o uso de recursos autônomos baseados em Inteligência Artificial Generativa para apoiar equipes de tecnologia, sobretudo em atividades de Engenharia de Requisitos. [Problema] Nesse contexto, os aspectos de Experiência do Usuário (UX) permanecem como uma dimensão crítica e ainda pouco explorada na documentação ágil de requisitos, especialmente na redação de *Acceptance Criteria* (AC). [Método] Essa pesquisa investiga se *Large Language Models* (LLMs) podem apoiar a revisão de ACs com base em diretrizes de UX e, assim, contribuir para a validação de requisitos por meio de tutoria automatizada. Para isso, foi conduzido um estudo empírico em duas fases: exploratória, composta por *survey* com 31 profissionais de TI sobre a influência de *Definition of Ready* (DoR) em requisitos endereçados a UX, que serviu de inspiração para, além de uma análise de conteúdo de 20 ACs escritos no formato *Behavior-Driven Development* (BDD) por alunos de graduação e analisados por um avaliador humano, elaborar a fase seguinte, uma análise comparativa entre dois LLMs, ChatGPT-4o e Gemini 2.5 Flash, configurados por meio de um modelo de *prompting* instrucional denominado ACUX Tutor 1.0. As respostas geradas pelos LLMs foram avaliadas por Concordância (alinhamento com o avaliador humano), Precisão Técnica (recomendações de *guidelines* de UX tecnicamente corretas e aplicáveis ao contexto do AC) e Explicabilidade (clareza e fundamentação das justificativas sobre a *guideline* recomendada). [Resultados] Ambos os LLMs foram capazes de identificar oportunidades de aprimoramento de UX nos trechos dos ACs, com padrões distintos de comportamento. O ChatGPT-4o apresentou maior precisão técnica (89,29%), alinhamento pontual às avaliações humanas, e aderência às restrições do *prompt*, enquanto o Gemini 2.5 Flash destacou-se pela explicabilidade (78,95%) e concordância com o avaliador humano (58,60%), demonstrando amplo repertório semântico sobre elementos de UX, embora, por vezes, extrapolasse recomendações além do contexto imediato dos ACs. [Conclusão] Conclui-se que abordagens baseadas em LLMs, quando orientadas por *prompts* estruturados e guiadas por diretrizes de domínio, possuem potencial para complementar a análise humana no apoio à validação de requisitos, promovendo especificações mais consistentes, rastreáveis e orientadas à UX. Contudo, não substituem a validação humana. A supervisão sobre as respostas dos LLMs permanece indispensável para garantir a profundidade interpretativa e o julgamento contextual necessários em projetos reais, devendo ser consideradas, portanto, como ferramentas assistivas.

Palavras-chave: Experiência do Usuário, Histórias de Usuário, Critérios de Aceitação, Grandes Modelos de Linguagem, Engenharia de Requisitos.

ABSTRACT

[Context] The increasing complexity of digital projects has driven the adoption of autonomous resources based on Generative Artificial Intelligence to support technology teams, particularly in activities related to Requirements Engineering. [Problem] In this context, User Experience (UX) aspects remain a critical yet underexplored dimension within agile requirements documentation, especially in the writing of Acceptance Criteria (AC). [Method] This research investigates whether Large Language Models (LLMs) can support the revision of ACs based on UX guidelines and, consequently, contribute to requirements validation through automated tutoring. To this end, an empirical study was conducted in two phases: an exploratory phase, consisting of a survey with 31 IT professionals about the influence of Definition of Ready (DoR) on UX-related requirements, which inspired, in addition to a content analysis of 20 ACs written in Behavior-Driven Development (BDD) format by undergraduate students and analyzed by a human evaluator, the design of the following phase, a comparative analysis between two LLMs, ChatGPT-4o and Gemini 2.5 Flash, configured through an instructional prompting model called ACUX Tutor 1.0. The responses generated by the LLMs were evaluated based on Agreement (alignment with the human evaluator), Technical Accuracy (technically correct UX guideline recommendations applicable to the AC context), and Explainability (clarity and justification of the recommended guideline). [Results] Both LLMs were able to identify opportunities for UX improvement in the AC excerpts, exhibiting distinct behavioral patterns. ChatGPT-4o demonstrated higher technical accuracy (89.29%), punctual alignment with human evaluations, and strong adherence to prompt constraints, while Gemini 2.5 Flash stood out for its explainability (78.95%) and concordance with the human evaluator (58.60%), displaying a broader semantic repertoire on UX elements, although it occasionally extended recommendations beyond the immediate context of the ACs. [Conclusion] The study concludes that LLM-based approaches, when guided by structured prompts and domain-specific guidelines, have the potential to complement human analysis in supporting requirements validation, fostering more consistent, traceable, and UX-oriented specifications. However, they do not replace human validation. Oversight of LLM-generated responses remains essential to ensure the interpretative depth and contextual judgment required in real-world projects; thus, such models should be regarded as assistive tools.

***Keywords**: User Experience, User Stories, Acceptance Criteria, Large Language Models, Requirements Engineering.*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 Introdução

1.1 Contexto

A busca por técnicas que garantam qualidade, eficiência e inovação para desenvolver soluções competitivas tem se intensificado de forma significativa nas últimas décadas. O mundo está cada vez mais acelerado, ditando, de certa forma, novas visões sobre como conduzir o trabalho, como tornar entregas mais eficientes. A exemplo disso, se percebe a “crise do software” com equipes de *software* enfrentando desafios em suas atividades e fluxo de trabalho, sobretudo no cumprimento das expectativas associadas aos projetos, como entregas no prazo, dentro do orçamento e que atendam às necessidades do cliente, apesar das conquistas tecnológicas (Pressman & Maxim, 2016, 2020). Em vista da complexidade das soluções digitais, somada à pressão por entregas ágeis, vemos a importância de adotar práticas estruturantes que ajudem a amenizar tal realidade. Segundo Pressman & Maxim (2020), a crescente demanda por sistemas mais sofisticados, com ciclos de desenvolvimento reduzidos e requisitos de alta qualidade, reforça a necessidade de metodologias organizadas e processos bem definidos, que sejam capazes de equilibrar flexibilidade e controle no ambiente de Engenharia de *Software* (ES).

É fundamental manter as equipes melhor preparadas de forma que determinem e implementem com confiança as características do sistema, alinhando com segurança às necessidades dos usuários e demais interessados (Pressman & Maxim, 2016, 2020; Mergulhão et al., 2019; Norman & Nielsen, 1998). Nesse ponto, a Engenharia de Requisitos (ER) se destaca como um dos processos contínuos essenciais entre as atividades gerenciais que estruturam o ciclo de vida de um projeto de *software* (Mergulhão et al., 2019). A área também exalta a importância da colaboração direta de desenvolvedores, clientes e outros *stakeholders* na definição e documentação dos requisitos necessários para que o produto atenda adequadamente às demandas estabelecidas (Sommerville, 2011). Em contextos ágeis, a prioridade é colaborar com a ER expandindo esse processo de descoberta por meio da comunicação contínua sobre os “indivíduos e interações mais que processos e ferramentas”, assim como a “colaboração com o cliente mais que

negociação de contratos” (Beck et al., 2001). Tais alinhamentos permitem identificar, portanto, os requisitos que definem o que o sistema deve realizar (requisitos funcionais), e sob quais restrições (requisitos não-funcionais) (Sommerville, 2011). A falta de requisitos bem definidos aumenta a probabilidade de ambiguidades, falhas de comunicação, retrabalhos e, em última análise, ao fracasso do projeto (Lucassen et al., 2015). No limite, há também o risco de entregar um sistema que possivelmente possa ser rejeitado pelos seus usuários, pois, ao não contemplar suas reais demandas, o torna ineficaz para a solução dos problemas que motivaram sua concepção (Sommerville, 2011; Norman, 2013; Pressman & Maxim, 2020).

Em resposta a tais limitações, a Experiência do Usuário (UX) é reconhecida como um atributo estratégico para o sucesso de projetos de *software*, capaz de orientar o desenvolvimento para além da dimensão técnica, garantindo que o sistema seja não apenas utilizável, mas também alinhado às reais necessidades dos usuários (Hassenzahl, 2010; Garrett, 2011; Sommerville, 2011; Norman & Nielsen, 1998). A integração de UX desde a definição dos requisitos permite capturar dores latentes e expectativas subjetivas dos usuários, direcionando o projeto tanto para o funcionamento adequado quanto para a geração de valor percebido. Isso contribui para minimizar ambiguidades, retrabalhos e falhas de alinhamento entre equipes de desenvolvimento e *stakeholders* (Lucassen et al., 2015; Pressman & Maxim, 2020).

Por outro lado, quando os requisitos não contemplam aspectos de usabilidade, utilidade e satisfação, os riscos de produto aumentam (Sommerville, 2011; Norman, 2013; Cagan, 2018). O fato de negligenciar a perspectiva dos usuários no processo de desenvolvimento pode acarretar custos substancialmente mais elevados na correção de falhas e na realização de ajustes pós-entrega (Sommerville, 2011). Diante disso, metodologias ágeis de desenvolvimento surgem como alternativa para otimizar ciclos de entrega, priorizando a criação de documentos concisos e relevantes, que evoluem ao longo do projeto, refletindo as mudanças e aprendizados (Beck et al., 2001; Garcia et al., 2017). No modelo ágil, a documentação de requisitos é feita de forma simplificada, por meio de histórias do usuário (*User Stories* - USs, em inglês). Essas histórias são popularmente utilizadas para especificar requisitos (Wang X et al., 2014; Schön et al., 2017). Uma US é uma descrição objetiva de uma tarefa unitária, expressa em linguagem não técnica e

contextualizada no domínio do usuário. Normalmente, incluem restrições funcionais e não funcionais, além de critérios de aceitação (*Acceptance Criteria* - AC, em inglês) que estabelecem as condições mínimas para que a tarefa seja aceita e considerada concluída pelo *Product Owner* (Cohn, 2004). Em casos de necessidade do time, as USs podem incluir ainda especificações de casos de teste e cobertura (Cohn, 2004).

Depois de ler uma US, a equipe sabe o porquê está construindo, o que está construindo e qual benefício entrega (Cohn, 2004). Esse nível de clareza e o valor de negócio que cada uma entrega, direciona a atenção da equipe para os parâmetros que irão validar a conformidade de uma tarefa. Entre esses parâmetros, os ACs se destacam como elementos essenciais no processo de validação de requisitos. Eles assumem um papel estratégico para que o *software* satisfaça as expectativas do usuário final e os padrões de qualidade previamente estabelecidos (Cohn, 2004). Além disso, condicionam a criação de casos de teste executáveis, a realização de validações funcionais e a aceitação formal das entregas por parte dos *stakeholders*. Embora a inclusão de ACs nas USs não seja uma obrigatoriedade formal, há um consenso na literatura ágil de que sua ausência tende a gerar ambiguidades e retrabalho (Cohn, 2004; Pichler, 2010; Pressman & Maxim, 2016, 2020). Por isso, diversos autores, entre eles Cohn (2004), Pichler (2010) e Pressman & Maxim (2016,2020), consideram a definição de ACs uma prática padrão de qualidade, por contribuir para maior entendimento sobre o requisito de *software*, alinhamento entre times e testabilidade das USs.

No desenvolvimento ágil, os ACs atuam como mecanismos de validação funcional dos requisitos do *software*, assegurando que sejam claros, verificáveis e alinhados aos objetivos do projeto e do usuário (Cohn, 2004; Gothelf & Seiden, 2016). Segundo Cohn (2004), os ACs definem condições objetivas que orientam o desenvolvimento e a validação de funcionalidades, enquanto Gothelf & Seiden (2016) destacam que, ao incorporar princípios de UX, os critérios favorecem a construção de soluções mais centradas nas necessidades do usuário final. Considerando o papel central dos ACs, se torna relevante refletir sobre como potencializar esse artefato.

Em paralelo às motivações de estabelecer requisitos estruturados e ACs consistentes, o avanço recente das Inteligências Artificiais Generativas (*Generative Artificial Intelligences* - GenIAs, em inglês) e, em particular, dos Grandes Modelos de Linguagem (*Large Language Model* - LLMs, em inglês), abriu novas possibilidades para apoiar a ES (Jurisch et al., 2017; Peña Veitía et al., 2020; Fathin Najwa Binti Mustaffa et al., 2021; Sharma et al., 2025). Essas tecnologias têm se tornado extremamente populares, tanto no campo acadêmico quanto na indústria, devido à sua rápida evolução e capacidade de generalização (Bommasani et al., 2021; Zhao W et al., 2024). Para que os LLMs executem tarefas automaticamente, é necessário desenvolver instruções apropriadas, escritas em Linguagem Natural (LN), conhecidas como *prompts*. A prática emergente de projetar e refinar *prompts* de modo a obter respostas desejadas de um modelo GenIA é denominada de Engenharia de *Prompts* (EP), e vêm sendo amplamente explorada para maximizar o desempenho dos modelos em diferentes aplicações e contextos (Liu P et al., 2023; White J et al., 2023b). Dada a sua capacidade de interpretar LN e gerar recomendações, os LLMs surgem como ferramentas promissoras para apoiar atividades da ER e otimizar o fluxo de trabalho em projetos ágeis (Zhang A et al., 2023; Dakhel et al., 2023).

Especificamente no que se refere à definição e validação de ACs, os LLMs pré-treinados e orientados por *prompts* podem contribuir oferecendo sugestões de melhorias, e reforçando a aderência das USs às boas práticas de usabilidade e UX. Há um progresso crescente no número de abordagens que exploram a capacidade dos LLMs para avaliar a qualidade das USs (Zhang Z et al., 2024), se baseando em diretrizes conhecidas, como por exemplo, as encontradas no *framework Quality User Story* (QUS) (Lucassen et al., 2015; Fathin Najwa Binti Mustaffa et al., 2021; Sharma et al., 2025; Yamani et al., 2025). Com seus recursos avançados de processamento em LN, os LLMs oferecem uma alternativa encorajadora para a avaliação automática de USs (Zhang Z et al., 2024). Embora o uso desses modelos proporcione maior agilidade em comparação à avaliação humana (Jurisch et al., 2017), sua implementação prática exige tempo e depende da qualidade dos dados de treinamento e da complexidade das USs a serem avaliadas (Peña Veitía et al., 2020).

Nesse sentido, é razoável estender tais aprendizados também à avaliação de ACs. Diante das evidências sobre a aplicabilidade dos LLMs na avaliação da qualidade de USs (Zhang Z et al., 2024; Fathin Najwa Binti Mustaffa et al., 2021; Sharma et al., 2025; Yamani et al., 2025), torna-se pertinente investigar de que forma essas abordagens podem ser adaptadas a um nível mais granular, o dos ACs, compreendendo-os não apenas como complementos das USs, mas como artefatos críticos que, quando enriquecidos por diretrizes de UX e apoiados por técnicas de automação, podem apoiar na atuação dos *Product Owners* no processo de validação de requisitos. Essa ampliação de perspectiva manifesta uma oportunidade de elevar a maturidade dos processos ágeis e aprimorar o valor entregue pelos produtos, ao mesmo tempo em que abre caminhos para o avanço do estado da arte em ER e UX, servindo não apenas para automatizar etapas manuais e suscetíveis à falhas, mas também, promover ACs mais consistentes e assertivos.

Entretanto, para compreender plenamente o impacto da proposta, torna-se necessário situar essa discussão no contexto mais amplo da integração entre UX e métodos ágeis, a qual ainda enfrenta desafios estruturais, e de comunicação que precisam ser considerados (Medeiros et al., 2018), sobretudo no que diz respeito à atuação das equipes no tratamento de informações de UX. Integrar UX com desenvolvimento ágil de *software* é uma atividade complexa, difícil na prática (Ananjeva et al., 2020). Zaina et al. (2022) apontam que membros de equipes ágeis geralmente não têm um papel ativo nas discussões de informações de UX, fator que, por consequência, pode comprometer a qualidade do sistema de alguma forma. Autores abordam a importância de facilitar a comunicação entre os membros de times de desenvolvimento, identificando esse aspecto como um dos desafios para a integração ágil e UX (Medeiros et al., 2018; Zaina et al., 2022; Garcia et al., 2019). Por esse cenário, sugerem reforços táticos destacando o uso de artefatos para facilitar a comunicação e colaboração (Medeiros et al., 2018; Garcia et al., 2019; Zaina et al., 2022), e a criação de *guidelines* para vincular e organizar informações de UX (Zaina et al., 2022).

A utilização de *guidelines* em ACs pode representar mais do que uma prática de verificação. Pode ser um mecanismo de aprendizado organizacional e de evolução contínua na ER. Em ambientes ágeis, onde a iteração rápida pode diluir padrões de qualidade, *guidelines* funcionam como estruturas de referência

compartilhadas, capazes de uniformizar a linguagem entre *stakeholders*, reduzir interpretações divergentes e, para o campo da UX, assegurar que aspectos de usabilidade sejam tratados de forma sistemática (Lopes et al., 2019; Souza et al., 2020). Além disso, ao incorporar dimensões de UX, esses critérios passam a desempenhar um papel de mediação entre a visão de negócio, design e desenvolvimento, permitindo que requisitos não funcionais sejam explicitados de maneira prática e mensurável.

Essa perspectiva pode ser aplicada, de início, pelas *guidelines* clássicas conhecidas. Donald Norman (2013) é um pioneiro na área de Design Centrado no Usuário e definiu 6 princípios fundamentais de design de experiência (Visibilidade, *Feedback*, *Affordance* - conceito relacionado com a acessibilidade do design, Mapeamento, Restrições, e Consistência) focados na compreensão do comportamento humano e na criação de designs que sejam fáceis de entender e usar. Fatores que apontam o destaque que requisitos devem refletir necessidades emocionais e cognitivas dos usuários. Jakob Nielsen (1994) desenvolveu 10 heurísticas de usabilidade que caracterizam princípios gerais de design de interface de usuário amplamente utilizados para auditar a usabilidade de um sistema. Seus benefícios práticos são vastos, em especial, a melhora na UX, validação rápida e redução de custos de suporte.

O *framework* de Garrett, também conhecido como *"The Elements of User Experience"*, é amplamente utilizado em bibliotecas universitárias de Design de Interação e UX (Garrett, 2011). A metodologia oferece uma estrutura coesa para pensar e projetar experiências de usuário eficazes. Jesse James Garrett (2011) argumenta que o sucesso de um produto depende da coesão e do alinhamento entre 5 níveis interconectados (Estratégia, Escopo, Estrutura, Esqueleto e Superfície) que descrevem os componentes essenciais de um bom design de produto, desde o nível mais abstrato até o mais concreto.

Um conjunto de *guidelines*, denominado **ACUX** (*Acceptance Criteria for User Experience*), foi desenvolvido por Jonathan Souza (Souza, 2021) para incorporar aspectos de UX diretamente nos ACs, organizando-os em dois grandes grupos, baseados no *framework* de Garrett (2011): (1) Design da Interação e Organização da Informação, e (2) Elementos Visuais. As *guidelines* ACUX detalham, por

exemplo, como explicitar interações do usuário, caminhos de navegação, disposição de conteúdo, uso de elementos visuais adequados e aspectos de estilo, como tipografia e paleta de cores. Além disso, fornece orientações didáticas em alerta para erros comuns na escrita de ACs, como descrições vagas, percepções subjetivas ou ausência de detalhes de interação. Nesse sentido, o ACUX se coloca como uma proposta aderente a integração entre ER, práticas ágeis e UX, servindo tanto como guia de padronização quanto como instrumento de aprendizado organizacional. Essa abordagem reforça a ideia de que a criação de *guidelines* não deve ser vista apenas como um meio de evitar falhas, mas como um recurso capaz de ampliar a maturidade do processo de requisitos, alinhando-o às práticas centradas no usuário e à escalabilidade da qualidade do produto (Zaina et al., 2022; Medeiros et al., 2018; Zhang Z et al., 2024).

Como destacam Kashfi et al. (2017), não é mais suficiente concentrar atenção nas muitas funcionalidades que um *software* deve fornecer; é fundamental considerar os demais aspectos que envolvem a concepção do produto e sua experiência de uso. Por essa visão, apoiar a centralidade da UX nos processos de desenvolvimento de *software* significa ampliar o escopo dos requisitos para além da funcionalidade, incorporando dimensões de usabilidade, consistência e valor percebido (Hassenzahl, 2010; Sommerville, 2011; Norman, 2013). Isso não apenas fortalece a qualidade das entregas, mas também reduz os riscos de rejeição do sistema pelos usuários, aproximando a intenção do projeto da experiência efetivamente entregue e assegurando maior alinhamento entre negócio, design e tecnologia (Pressman & Maxim, 2016, 2020; Gothelf & Seiden, 2016; Cagan, 2018).

1.2 Motivação e Justificativa

Garcia et al. (2017) apontam que as USs evidenciam o que deve ser implementado, e geram discussões para definir se determinados recursos irão gerar valor para o produto. Nesse contexto, como elemento relevante, os ACs de USs determinam as condições mínimas para que uma funcionalidade seja considerada concluída e apta para uso. Contudo, na prática, ocorre que, em muitos casos, esses

critérios não estão vinculados a *guidelines* de usabilidade e UX (Souza et al., 2020; Souza, 2021). Fato que não está diretamente relacionado a falha na especificação de USs, mas sim, frequentemente, associado a comunicação (Medeiros et al., 2018).

As USs foram criadas para serem usadas em consonância com os princípios ágeis. Porém, observa-se que a indústria ainda enfrenta dificuldades em incorporar completamente esses princípios (Medeiros et al., 2018). Problemas recorrentes de comunicação entre os *stakeholders* acabam exigindo a criação de artefatos adicionais de requisitos, como apontam Medeiros et al. (2018). Nesse contexto, a inclusão de preocupações relacionadas à UX nas USs pode ser entendida como uma estratégia compensatória frente às falhas de comunicação que persistem entre os envolvidos no processo. Essa fragilidade metodológica pode comprometer o alinhamento das soluções entregues, além de elevar o risco de insatisfação, e desperdício de recursos no ciclo de desenvolvimento. Estudos apontam que as necessidades dos usuários são tratadas de forma informal ou, muitas vezes, negligenciada, quando avaliamos requisitos relacionados à UX (Kashfi et al., 2017; Law et al., 2009).

Promover ACs mais maduros, avaliados por consistência, usabilidade, e nas necessidades dos usuários, continua sendo um desafio para equilibrar agilidade no desenvolvimento do projeto, sendo uma lacuna crítica entre intenção e entrega real. Aldave et al. (2019) constatam que, em metodologias ágeis, tende-se a priorizar escopo e simplicidade, mas há pouca atenção em capturar e representar necessidades reais dos usuários, o que compromete a inovação e a usabilidade do produto final. De forma similar, estudos de ER em contexto ágil, como trazido por Schön et al. (2017), mostram que a compreensão compartilhada da perspectiva do usuário ainda está longe de ser consolidada, indicando a persistência de lacunas na integração entre UX e requisitos ágeis (Schön et al., 2017). Inayat et al. (2015) mostram que a preocupação com fatores de UX em requisitos ainda é incipiente, reforçando a necessidade de abordagens mais sistemáticas, e atenção adicional à práticas de ER ágil ao lidar com requisitos não funcionais. Sendo assim, é essencial discutir alternativas que aproximem a definição de ACs das boas práticas de UX para a fortalecer a aderência das entregas aos padrões de qualidade percebida e valor de uso pelos usuários.

Nesse contexto, o uso de LLMs se revela como uma boa possibilidade para sistematizar o apoio à ACs sob a perspectiva da UX. Grande parte dos estudos conhecidos na literatura são limitados em trabalhos que propõem ou avaliam metodologias para integrar LLMs a facilitação do processo de validação de ACs, se concentrando mais em estudos de avaliação de USs (Kolthoff et al., 2024). Um ensaio aproximado foi evidenciado por Brockenbrough, Feild & Salinas (2025) que investigaram o impacto dos LLMs no ensino de ES, concluindo que ajudam estudantes a criar USs com ACs bem definidos, embora o desempenho possa variar dependendo da clareza do escopo. Esse achado é particularmente relevante para a área de UX, pois reforça o potencial dos LLMs em beneficiar a maturação dos ACs.

A proposta deste estudo se justifica, portanto, pela oportunidade de preencher essa lacuna, explorando o potencial dos LLMs como ferramentas de suporte à ACs capaz de incorporar diretrizes de UX e, assim, facilitar sua validação por *Product Owners* e demais *stakeholders*. Para além da contribuição prática, este trabalho também busca gerar avanços teóricos na integração de ER, UX e GenIA, ampliando a compreensão sobre como esses domínios podem se complementar. Nesse sentido, pretende-se disponibilizar uma abordagem automatizada baseada em LLMs que possa ser utilizada por profissionais de tecnologia em seus processos de validação, fortalecendo a entrega por produtos digitais mais consistentes, acessíveis e centrados nas necessidades dos usuários. Assim, essa proposta de pesquisa almeja contribuir não apenas para o avanço do estado da arte, mas também para a melhoria prática dos fluxos de trabalho em ambientes ágeis.

Sendo assim, a pesquisa está guiada para responder duas questões de pesquisa (QPs):

> **QP1**: *É possível sistematizar a validação de ACs baseada em guidelines de UX utilizando LLMs para apoiar a Engenharia de Requisitos?*
>
> **QP2**: *De que modo abordagens baseadas em LLMs orientadas a compreensão de princípios de UX podem beneficiar o time de desenvolvimento na escrita de Requisitos de Software?*

1.3 Objetivos

O objetivo desta pesquisa é propor uma abordagem automatizada por LLM, denominada ACUX Tutor 1.0, que reforçe a contextualização de ACs de USs com aspectos de UX para apoiar a validação de requisitos de *software*. Esse direcionamento se fundamenta no propósito de tornar os conceitos de UX mais acessíveis e aplicáveis em tais artefatos, a fim de favorecer a comunicação e a tomada de decisão sobre os requisitos entre os membros das equipes em produtos e serviços digitais que carecem, ou necessitam, fortalecer o design centrado no usuário, e suas práticas.

A partir do objetivo geral desta pesquisa, os seguintes objetivos específicos foram definidos:

1. Realizar um *survey* exploratório sobre DoR investigando como requisitos bem definidos, especialmente em aspectos de UX, podem otimizar validações.
2. Avaliar a aplicabilidade do ACUX na prática domínio de sua abordagem metodológica a fim de auxiliar a modelagem de um *prompt* instrucional.
3. Desenvolver um *prompt* instrucional estruturado com técnicas de *Chain-of-Thought* (CoT) *Prompting, Instructional Prompting e Few-Shot Prompting,* orientado a análise de ACs com diretrizes de UX.
4. Aplicar dois LLMs (ChatGPT-4o e Gemini 2.5 Flash) na validação automatizada de ACs desenvolvidos por alunos de graduação.
5. Mensurar a concordância entre as recomendações dos LLMs e a avaliação humana.
6. Analisar a precisão técnica e a explicabilidade das recomendações geradas pelos LLMs.
7. Identificar limitações, inconsistências e oportunidades de melhoria na aplicação de LLMs para esse propósito.

Com base nesses objetivos e questões de pesquisa, se organiza a condução metodológica desta dissertação, que buscará integrar fundamentos de ER, de UX e

de LLM em um estudo empírico. É esperado que este estudo contribua como meio inicial de inspiração a futuros trabalhos que se dediquem a investigar como LLM podem contribuir, de forma técnica e prática, para apoiar o processo de validação de requisitos no desenvolvimento de *software* orientado à UX.

1.4 Trabalhos Relacionados

A aplicação de LLMs no contexto da ER tem se tornado um tema recorrente na literatura recente, refletindo o interesse crescente da comunidade acadêmica e profissional em explorar o potencial dessas tecnologias para otimizar tarefas tradicionalmente manuais e suscetíveis a falhas. Embora haja um número significativo de estudos entre essas áreas, o número de pesquisas específicas nas subáreas de ER ainda é relativamente menor. Mais restrito ainda é o volume de pesquisas que contemplam, de forma sistemática, a perspectiva da UX na definição e validação de requisitos de *software*. Essa ausência de abordagens integradas reforça a oportunidade de discutir propriedades relacionadas à usabilidade e qualidade percebida das quais podem influenciar a formulação de requisitos e, consequentemente, o desenvolvimento de soluções digitais mais centradas no usuário.

Nesse cenário, Ronanki et al. (2023) investigaram o potencial do GPT-3.5 para auxiliar nos processos de elicitação de requisitos da ER para o desenvolvimento de sistemas de IA Confiável (*Trustworthy AI*), comparando a qualidade dos requisitos gerados pelo modelo com os formulados por especialistas humanos, onde ambos foram orientados pelo mesmo contexto de perguntas (que para o LLM, foram disponibilizadas em formato de *prompt*), inerentes a pilares fundamentais e regulamentadores da Comissão Europeia (2021 e 2023) — Precisão e Robustez, Segurança, Não Discriminação, Transparência e Explicabilidade, Responsabilidade, Privacidade e Segurança, Regulamentações e Agência Humana e Supervisão. A IA Confiável constitui um campo de estudo emergente que busca garantir que os sistemas de IA sejam projetados, desenvolvidos e utilizados de forma que inspirem confiança nos usuários e na sociedade (Kaur et al., 2021). Nesse sentido, a tarefa de geração de conteúdo solicitada ao LLM visou não apenas reproduzir respostas plausíveis, mas também demonstrar a capacidade do modelo

em compreender nuances éticas e normativas implícitas nos princípios de confiabilidade, articulando-as em requisitos coerentes e consistentes com as diretrizes de *Trustworthy AI*. Os autores tiveram a preocupação de garantir a equivalência e correspondência entre as respostas obtidas a partir do mesmo conjunto de perguntas, assegurando condições comparáveis de análise entre as respostas humanas e as respostas geradas pelo modelo.

O conjunto de requisitos obtidos no estudo de Ronanki et al. (2023) foi avaliado por sete atributos qualitativos de qualidade propostos por Denger et al. (2005) e Génova et al. (2013): Abstração, Atomicidade, Consistência, Correção, Não Ambiguidade, Compreensibilidade e Viabilidade. Para tanto, um grupo de cinco avaliadores humanos, especialistas em ER com foco em sistemas de IA, pontuou respostas em cada um dos atributos por uma escala de 0 a 10, onde 0 determinou a pontuação mais baixa (piores requisitos de qualidade) e 10 a pontuação mais alta (melhores requisitos). Um total de 36 respostas foram analisadas, e as médias gerais médias dos atributos, e seus respectivos desvio-padrão foram calculados. Os resultados indicaram que os requisitos gerados pelo GPT-3.5 foram considerados altamente abstratos, atômicos, consistentes, corretos e compreensíveis em comparação com os requisitos formulados por especialistas humanos. No entanto, os atributos Não Ambiguidade e Viabilidade apresentaram pontuações mais baixas nos requisitos produzidos pelo LLM, sugerindo limitações na precisão interpretativa e na adequação prática das respostas. Ainda assim, os autores avaliaram que os requisitos gerados pelo ChatGPT foram usualmente aceitáveis, superando as respostas humanas na maioria dos casos, embora com deficiências pontuais nesses dois aspectos. O estudo conclui que LLMs como o ChatGPT têm potencial promissor para aumentar a eficiência nas atividades de ER.

Essas descobertas fornecem *insights* valiosos sobre a aplicabilidade dos modelos na avaliação automatizada de requisitos, ao mesmo tempo em que ressaltam a necessidade de investigações complementares voltadas a dimensões ainda pouco exploradas nas atividades de ER, como em orientações de escrita centradas na UX. O trabalho desenvolvido por Ronanki et al. (2023) se insere diretamente no campo de pesquisa Processamento de Linguagem Natural (PLN) aplicado à ER (*Natural Language Processing for Requirements Engineering* - NLP4RE*,* em inglês) (Dalpiaz et al., 2018; Zhao L et al., 2021), e sua principal

contribuição do estudo foi fornecer uma prova preliminar de conceito demonstrando que o ChatGPT pode gerar requisitos considerados de qualidade aceitável por especialistas de ER. O NLP4RE busca aplicar ferramentas, técnicas e recursos de PNL aos processos de RE para dar suporte a analistas humanos na execução de várias tarefas em requisitos textuais, como detectar e melhorar problemas de linguagem, entre outras coisas (Zhao L et al.,2021), que aumentam a qualidade dos requisitos. Os resultados apoiam a noção de que LLMs treinados podem ter um desempenho favorável em tarefas técnicas específicas que exigem uma resposta variada e criativa, mostrando-se vantajoso para apoiar atividades de análise, formulação e revisão de requisitos de *software* em domínios complexos (Zhang T et al., 2020; Brown et al., 2020; Zheng et al., 2023).

Zhang Z et al. (2024) também se utilizou de os recursos avançados de PLN dos LLMs verificando que apresentam um potencial promissor para automatizar a melhoria da qualidade da US, sobretudo, em aspectos que atendam às necessidades do cliente e aos objetivos de negócio. Sobre isso, investigou a viabilidade de utilizar agentes autônomos baseados em LLM criando um sistema denominado ALAS para gerar versões aprimoradas de USs em ambientes ágeis de desenvolvimento de *software*, a partir do modelo GPT-3.5-turbo-16k e GPT-4-1106-preview. O sistema opera em duas fases do gerenciamento de requisitos: preparação da tarefa e execução da tarefa.

Para interpretação contextualizada desse processo, o ALAS foi configurado com dois perfis de agentes: *Product Owner*, compreendendo a visão do projeto, avaliando as USs pelas necessidades dos clientes, valor de negócio e objetivos gerais do produto, e o "*Requirements Engineering (RE)"*, alinhado com a qualidade das USs, garantindo descrições inequívocas e ACs mensuráveis. O estudo contou com a colaboração de participantes atuantes nessas áreas profissionais que avaliaram as versões aprimoradas das USs geradas pelo sistema, e classificaram suas respostas utilizando a escala Likert de 1 a 5, onde 1 indicava forte discordância e 5 indicava forte concordância em relação à qualidade percebida das melhorias.

Dentre as principais descobertas, em ambos modelos, o ALAS demonstrou resultados consideráveis na clareza, especificidade e articulação do valor comercial das USs melhoradas. No entanto, foram identificadas limitações importantes

relacionadas à tendência dos modelos de gerar descrições excessivamente longas e detalhadas, especialmente nos ACs. Os autores sugeriram a inclusão de um agente adicional especializado em análise de qualidade, com o propósito de monitorar o escopo, o nível de detalhamento e a relevância das informações produzidas pelos LLMs durante a geração das USs. Além disso, destacaram o cuidado em elaborar *prompts* em processos automatizados, com a necessidade de revisões rigorosas conduzidas por especialistas humanos. Essas observações reforçam que, embora os LLMs sejam ferramentas promissoras para o suporte à ER, sua aplicação eficaz demanda configurações bem planejadas, ajustes iterativos e supervisão qualificada, sobretudo quando se busca manter o equilíbrio entre qualidade técnica dos requisitos e valor percebido pelas partes interessadas em projetos ágeis. É fato que o gerenciamento eficaz de requisitos garante que os projetos de *software* entreguem produtos que atendam às necessidades do cliente e cumpram as metas de negócios.

De acordo com Karlsson et al. (2025), a classificação inadequada dos requisitos de *software* pode aumentar o risco de negligenciar elementos críticos relacionados à funcionalidade e ao desempenho do produto final. Sobre isso, ele conduziu um estudo experimental para avaliar a confiabilidade de dois LLMs, o GPT-4o e o LLAMA3.3-70B, na classificação automática de requisitos a partir da abordagem de aprendizado *zero-shot prompting*, sem treinamento prévio em conjuntos de dados específicos para maior flexibilidade do experimento. A pesquisa utilizou como base o conjunto de dados PROMISE NFR11, com 625 requisitos, dos quais os modelos realizaram uma classificação binária distinguindo no conjunto de dados Requisitos Funcionais (RFs) de Requisitos Não Funcionais (RNFs). Foram configurados diferentes valores de temperatura, parâmetro utilizado para controlar a aleatoriedade nas respostas dos modelos, com o objetivo de avaliar seu impacto no desempenho e na consistência das classificações. Para medir a qualidade em que o modelo classificou os requisitos entre RFs e RNFs, utilizaram as métricas *precision*, *recall* e F1 *score*. Todas são métricas baseadas em contagem de acertos e erros (*confusion matrix*), usadas para classificar problemas ou tarefas de classificação binária ou multiclasse bem definidas, em que existe um rótulo claro e objetivo para cada instância, em outros exemplos, *spam* ou *não-spam*, positivo ou negativo, presença ou ausência de uma palavra. Ou seja, tarefas fechadas, onde o que conta

é acertar ou errar segundo regras bem definidas. Os resultados mostraram que o GPT-4o apresentou um bom desempenho na identificação de RFs, com a maior consistência ocorrendo na configuração de temperatura mais baixa. Por outro lado, o LLAMA3.3-70B demonstrou ser ligeiramente mais consistente que o GPT-4o em todas as temperaturas, tornando classificações mais previsíveis. O estudo não explorou os motivos das classificações errôneas e o efeito da variação no design do *prompt* sobre os resultados obtidos, aspectos que poderiam aprofundar a compreensão sobre os limites e potencialidades desses modelos. O que demonstra que estabelecer procedimentos complementares de revisão humana permanece essencial, sobretudo para os RNFs, que costumam demandar maior interpretação contextual e sensibilidade às nuances do projeto.

Kolthoff et al. (2024) aborda o desafio de criar e validar protótipos de Interfaces Gráficas de Usuário (*Graphical User Interface* - GUI, em inglês) com base em requisitos, especificamente USs. A natureza iterativa da ER ágil leva a redesenhos frequentes de protótipos, consumindo tempo e esforço para identificar requisitos já implementados e integrar novos. Debnath et al. apresentam um estudo em que menos da metade das USs posteriores “incluem conteúdo que pode ser totalmente rastreado até as iniciais”, e uma alta porcentagem de USs resultantes eram novas ou refinamentos das iniciais.

Não existe uma abordagem que verifique automaticamente USs em protótipos de GUI e fornece recomendações para requisitos não implementados. A partir dessa lacuna, Kolthoff et al. (2024) elaborou uma abordagem semi-automática baseada em LLM, GPT-4, focada em detectar USs funcionais que não estão implementadas em um protótipo de GUI e fornecer recomendações, geradas em HTML/CSS, para componentes de GUI adequados que implementam diretamente as USs. Os resultados mostraram um desempenho substancialmente alto na previsão de USs, significando que o modelo é capaz de processar eficazmente a semântica da abstração da GUI e combiná-la com a semântica da história do usuário. Ao fornecer validação e recomendações, a abordagem ajuda os desenvolvedores a criar protótipos de GUI mais eficazes para elicitação e validação de requisitos, assegurando que o produto final seja mais utilizável e alinhado com as expectativas de UX. A detecção de USs não implementadas e a recomendação de

componentes de GUI contribuem para a completude, consistência, clareza e relevância das funcionalidades, aspectos vitais para a UX.

Krishna et al. (2024) avaliaram a proficiência de LLMs na criação, validação e retificação de documentos de Especificação de Requisitos de *Software* (*Software Requirements Specification* - SRS, em inglês) que descreveram os requisitos do produto final desenvolvido utilizando *prompts* em LN. Os modelos GPT-4 e CodeLlama foram testados na formulação do SRS de um projeto de *software* proposto pelos próprios autores: o desenvolvimento de um portal web de gerenciamento de clubes estudantis de uma universidade. Com os avanços recentes dos agentes de GenIA e a popularização de LLMs, observa-se um aumento em suas aplicabilidades para tarefas-chave de ER em tarefas-chave de ER, como extração, classificação, priorização e validação de requisitos (Yang Y et al., 2022; Kici et al., 2021).

Tradicionalmente, a preparação e revisão de um SRS é um processo manual que pode levar semanas ou meses. Diante desse contexto, o estudo buscou investigar a capacidade dos modelos em produzir rascunhos de SRS precisos, coerentes e estruturados, visando não apenas reduzir o tempo e esforço humano em elaborá-los, mas também automatizar etapas técnicas complexas do ciclo de vida do desenvolvimento de software. Para garantir a comparabilidade dos resultados, se basearam em um benchmark de referência com um documento SRS gerado por humanos, em conformidade com especificações do IEEE (1998). A partir desse padrão, elaboraram um *prompt* que instruía explicitamente os modelos a gerar respostas completas e detalhadas. Quatro especialistas independentes, originários da academia e da indústria, com pelo menos três anos de experiência e familiaridade com práticas de desenvolvimento de *software*, foram convidados a classificar os documentos produzidos pelos modelos. As classificações para cada SRS seguiram uma escala de concordância do tipo Likert de cinco pontos, permitindo aos revisores justificar suas pontuações com base em métricas qualitativas previamente definidas. Os parâmetros de avaliação foram divididos em dois grupos. Para cada requisito individual, mediram-se os atributos de Não Ambiguidade, Compreensibilidade, Correção e Verificabilidade. Para os documentos como um todo, avaliaram Completude, Não Redundância, Consistência e Concisão. Uma tabela comparativa foi elaborada para visualizar a quantidade de Requisitos

Funcionais, Requisitos de Desempenho, Restrições de Design, Interfaces Externas e Requisitos de Segurança gerados por cada modelo em relação ao documento SRS dos avaliadores humanos.

Os resultados indicaram que, diferentemente do CodeLlama, o ChatGPT forneceu *feedbacks* detalhados e construtivos para cada parâmetro, muitas vezes alinhados com as observações dos avaliadores humanos. O modelo também demonstrou menor desvio médio nas pontuações, permanecendo abaixo de ±1 na grande maioria dos casos, o que sugere um maior grau de concordância com as avaliações humanas. O CodeLlama tendeu a atribuir pontuações otimistas na métrica de Não Ambiguidade, mas significativamente inferiores em Correção e Verificabilidade, resultando em maiores desvios negativos. Além disso, suas justificativas fornecidas eram frequentemente genéricas e imprecisas, revelando limitações em compreender nuances semânticas e contextuais dos requisitos.

Outro achado relevante revelado pelos autores foi a economia de tempo expressiva proporcionada pelos LLMs nas respectivas tarefas de ER que o estudo se relaciona. Enquanto documentos criados por humanos que exigiam de 4 a 24 horas para serem criados após a especificação dos requisitos, os gerados pelos LLMs, embora difíceis de acertar na primeira tentativa, apresentaram uma redução de tempo de quase 7 a 47 vezes. Apesar do ganho substancial, Krishna et al. (2024) destacam que a elaboração de um SRS representa apenas uma pequena fração do esforço total do desenvolvimento de *software*; portanto, a economia observada nessa fase tende a ter impacto limitado sobre o custo e o esforço geral do projeto. Ainda assim, os resultados reforçam o potencial dos LLMs como ferramentas de apoio aos Engenheiros de *Software*, capazes de ampliar produtividade, e reduzir o tempo necessário para geração, validação e correção de requisitos. Tais evidências corroboram a tendência de integração crescente de LLMs em atividades de ER orientadas à qualidade e automação de processos cognitivos.

De modo geral, os estudos apresentados evidenciam a diversidade de aplicações dos LLMs no domínio da ES, com ênfase em atividades de ER. Os trabalhos de Ronanki et al. (2023), Zhang Z et al. (2024), Karlsson et al. (2025), Kolthoff et al. (2024) e Krishna et al. (2024) reforçam o potencial das GenIAs para

aprimorar a eficiência, a consistência e a rastreabilidade dos processos de desenvolvimento de *software*, especialmente em contextos ágeis.

A Tabela 1 apresenta uma comparação entre os trabalhos encontrados, considerando os temas envolvidos: UX, AC, LLM, e ER.

Tabela 1: Comparação entre os trabalhos relacionados.

Título/Autor	UX	AC	LLM	ER	Contribuição
Investigating ChatGPT's Potential to Assist in Requirements Elicitation Processes (Ronanki et al., 2023)			X	X	Validou via prova preliminar de conceito o potencial dos LLMs como assistentes poderosos que podem aumentar a eficiência dos processos de ER
LLM-Based Agents for Automating the Enhancement of User Story Quality: An Early Report (Zhang Z et al., 2024)	X	X	X	X	Sistema ALAS que melhora User Stories, incluindo critérios de aceitação, via agentes LLMs, considerando valor de negócio e clareza.
How Reliable Are GPT-4o and LLAMA3.3-70B in Classifying Natural Language Requirements?: The impact of the temperature setting (Karlsson et al., 2025)			X	X	Investigou por meio de estudo experimental a confiabilidade de LLMs na classificação zero-shot de requisitos funcionais e não funcionais, destacando o impacto de parâmetros e a importância de revisão humana para RNFs.
Interlinking User Stories and GUI Prototyping: A Semi-Automatic LLM-based Approach (Kolthoff et al., 2024)	X		X	X	Desenvolveu abordagem semi-automática com LLM para detectar USs não implementadas em protótipos GUI e recomendar componentes adequados para melhoria da interface e da UX.
Using LLMs in Software Requirements Specifications: An Empirical Evaluation (Krishna et al., 2024)	X		X	X	Avaliação empírica detalhada que buscou medir a proficiência e o potencial de economia de tempo de LLMs (GPT-4 e CodeLlama) na criação e melhoria de SRSs.

Fonte: Elaborado pela autora.

No entanto, mesmo com contribuições relevantes, nota-se que a maioria dos estudos concentra-se na análise estrutural e funcional dos requisitos, com ligeiras motivações sistemáticas à aspectos associados ao design e usabilidade das funcionalidades. Os trabalhos relacionados atestam a viabilidade técnica dos LLMs para apoiar atividades de ER. Ao que se investigou, não foram encontrados

trabalhos acadêmicos dedicados a mensurar o desempenho de LLMs a partir de *prompts* instrucionais para apoiar especificamente a validação de ACs orientados a diretrizes reconhecidas de UX. O que denota ser uma oportuna lacuna a ser explorada tanto na literatura acadêmica quanto nas práticas do setor. A proposta desta dissertação se alinha a essa oportunidade, ao direcionar o estudo para investigar o desempenho de dois modelos generativos de última geração, ChatGPT-4o e Gemini 2.5 Flash, com foco na análise automatizada de ACs modelados em BDD.

1.5 Contribuições

O estudo propõe contribuir com a academia ao apresentar um conjunto de reflexões que dialogam com o atual cenário de debate referenciado por iniciativas que utilizam a GenIA na ES. As contribuições deste trabalho de Mestrado se concentram em oferecer evidências e discussões pertinentes para apoiar o processo de validação de requisitos com foco em UX em contextos ágeis. A seguir, são listadas as principais contribuições inerentes a tese:

A. Esta pesquisa apresenta, ao que se tem conhecimento até o presente momento, uma das primeiras explorações deste tipo de validação automatizada para ACs por meio de modelos generativos de linguagem.

B. O estudo investiga empiricamente o desempenho e a eficácia dos LLMs na validação de ACs.

C. Destaca a ascensão na pesquisa sobre o estado da arte de aplicações de IA voltadas para aprimoramento de requisitos de *software* e UX.

D. Por fim, o estudo lança subsídios conceituais e metodológicos para futuras investigações acadêmicas e aplicadas sobre automatização à validação de artefatos em processos ágeis, incentivando abordagens híbridas no campo da ES.

Na finalidade de agregar informações sobre cada contribuição refletindo o eixo motivacional das etapas de pesquisa foi elaborada uma breve descrição como segue:

Survey exploratório sobre DoR: tem o propósito de investigar de que maneira requisitos mais bem preparados, sobretudo no que se refere a aspectos de UX, podem favorecer validações mais alinhadas às necessidades de clientes internos e externos, assim como mitigar os impactos negativos no desenvolvimento de produtos e serviços digitais decorrentes da carência desses elementos. Basicamente, funcionando como um instrumento inicial de investigação usado para coletar insights qualitativos e gerar hipóteses que subsidiaram a ideação da abordagem automatizada, ao mesmo tempo que reforçou o debate em torno da temática de validação de requisitos. O Capítulo 4 apresenta mais informações sobre o *survey* e as perguntas envolvidas, das quais foram elaboradas sem refletir estudos anteriores em razão do contexto específico. Com as evidências, se pretendeu viabilizar a integração da UX em USs por meio dos ACs, que levou a motivação e inspiração direta para a concepção do ACUX Tutor 1.0 como proposta de apoio contextual para *Product Owners*, *designers*, desenvolvedores e *stakeholders* utilizarem em tarefas de validação de requisitos.

Análise de Conteúdo de ACs (Bardin, 2016): nesse sentido, a análise de conteúdo foi realizada por um avaliador humano com experiência em UX e ER, com o intuito de verificar a aplicabilidade da técnica ACUX (Souza, 2021) em ACs escritos por alunos de graduação em LN no padrão Desenvolvimento Guiado por Comportamento (*Behavior Driven Development* - BDD, em inglês).

***Prompt* Instrucional e Execução dos LLMs**: logo após, o estudo empírico continua com a modelagem de *prompt* instrucional e a avaliação comparativa sobre dois LLMs de última geração, ChatGPT-4o e Gemini 2.5 Flash, em atuarem como tutores em ACUX verificando sua proximidade em relação à avaliação humana realizada no mesmo conjunto de ACs a partir de três fatores centrais: concordância, explicabilidade e precisão técnica. Tais fatores foram estabelecidos como métricas de avaliação e estão contextualizados na Tabela 11. A Seção 6.1 apresenta detalhes da seleção dos modelos generativos escolhidos. A avaliação também busca verificar a capacidade dos modelos de, não somente contribuir para o

empoderamento de profissionais de tecnologia sobre entendimentos ligados ao usuário, mas especificamente, facilitar a validação de artefatos de *backlog* por parte de líderes de produto e times multidisciplinares. A escolha pelas *guidelines* ACUX foi baseada no fato de que integram dimensões de interação, organização da informação e elementos visuais, aspectos frequentemente negligenciados na escrita de ACs, assim como, por em sua compatibilidade com práticas ágeis, funcionando como um recurso aderente, didático e aplicável a diferentes etapas do ciclo de requisitos.

1.6 Considerações Finais

O presente capítulo pontua os desafios contemporâneos enfrentados pela ES na busca por qualidade, eficiência e inovação em ambientes de desenvolvimento ágil, particularmente notórios nos aspectos que envolvem a relação do usuário e sua UX com a solução. Reconhecendo que a ER constitui uma das etapas mais críticas para o sucesso de qualquer projeto, a análise inicial evidenciou como as práticas tradicionais da área ainda enfrentam dificuldades na integração efetiva entre os domínios técnicos e a dimensão humana da UX.

Nesse sentido, a pesquisa delineou como a incorporação de diretrizes de UX, especialmente as propostas pelo *framework* ACUX (Souza, 2021), pode representar um avanço significativo para o amadurecimento dos processos de ER em ambientes ágeis. Ao posicionar os ACs como artefatos estratégicos de validação e mediação entre áreas de negócio, design e desenvolvimento, o estudo reforça o argumento de que sua qualificação não deve se limitar à dimensão funcional, mas abarcar também a experiência de uso e os parâmetros de valor percebido. Esse desdobramento conceitual abre caminho para uma prática mais colaborativa e centralizada às necessidades do usuário.

Adicionalmente, a Introdução sublinhou o potencial emergente das GenIAs e, em particular, dos LLMs, como ferramentas capazes de automatizar e aprimorar etapas da ER. A revisão preliminar da literatura sustenta a hipótese de que, mediante a orientação de *prompts* adequados, os LLMs podem ser empregados como avaliadores de requisitos. Essa possibilidade se conecta diretamente aos objetivos específicos propostos, que se consolidaram no desenvolvimento de um

estudo empírico de caráter exploratório e comparativo visando elaborar uma abordagem automatizada para a verificação da escrita de ACs com foco em UX, utilizando LLMs, intitulada ACUX Tutor 1.0.

Em suma, o capítulo determinou as bases conceituais e a motivação que norteiam esta investigação, estabelecendo as lacunas científicas a serem exploradas e os caminhos metodológicos que conduzirão à análise empírica. As reflexões apresentadas permitem compreender que há uma oportunidade para fortalecer a integração entre UX e ER, mediada por GenIA. Essa panorama visa não apenas a ampliação da maturidade dos processos de requisitos, mas também a contribuição direta para a concepção de produtos digitais intrinsecamente mais consistentes e, sobretudo, centrados nas necessidades humanas.

As discussões desenvolvidas neste capítulo, portanto, sustentam a relevância e a originalidade da pesquisa ao, posicionar a interseção entre UX, ER e LLMs como um território fértil para inovação metodológica, além de, contextualizar os fundamentos da EP e das técnicas de avaliação adotadas. Estes elementos configuram a estrutura do próximo capítulo, ao mesmo tempo em que, revelam a importância de compreender as implicações práticas e epistemológicas advindas do uso de LLMs.

2 Fundamentos Teóricos

2.1 Métodos Ágeis e USs

Os métodos ágeis surgiram como uma alternativa aos métodos tradicionais de desenvolvimento de *software*, cuja rigidez e altos custos de adaptação dos times dificultavam a resposta rápida às mudanças (Beck et al., 2001). Em contraste a essa realidade, os métodos ágeis adotam abordagens iterativas e incrementais, fundamentadas na colaboração contínua com o cliente e na entrega frequente de valor (Beck et al., 2001). Dentre as principais práticas adotadas por metodologias ágeis, como Scrum e *Extreme Programming* (XP), está o uso de USs como técnica pragmática para descrever requisitos funcionais e não funcionais, escritas a partir da perspectiva do usuário ou cliente, utilizando LN e acessível orientada ao valor de negócio. Segue uma estrutura gramatical simples, proposta por Cohn em 2004,

sendo a mais utilizada pelos desenvolvedores (Schön et al., 2017; Lucassen et al., 2016b). A estrutura se descreve pelo seguinte formato: **Como** [persona ou papel], **quero** [funcionalidade ou ação] **para** [benefício ou razão]. USs favorecem comunicação verbal, em vez de comunicação escrita. E por isso elas são também compatíveis com os princípios do Manifesto Ágil (Beck et al., 2001): (1) indivíduos e interações, mais do que processos e ferramentas; (2) *software* em funcionamento, mais do que documentação abrangente; (3) colaboração com o cliente, mais do que negociação de contratos; (4) resposta a mudanças, mais do que seguir um plano.

Na composição dessa dinâmica, é esperado que esse recurso se articule como instrumento de revisão a uma iteração planejada pelo time com os clientes. Jeffries et al. (2000), um dos autores originais do XP, define USs como artefato de gestão de requisitos que precisa ser documentada em três partes elementares: US = Cartão + Conversas + Confirmação. As USs são elaboradas em forma de cartões, uma representação física ou digital e deve funcionar como um identificador descritivo do requisito a ser implementado. As conversas referem-se ao diálogo colaborativo entre os membros do time de desenvolvimento e os clientes, por meio das quais os clientes detalham o que esclarecer sobre os requisitos em cada cartão. Esse momento é essencial para descobrir restrições, exceções, ajudando a alinhar expectativas e decidir conjuntamente como a funcionalidade será implementada. A confirmação representa a verificação do cliente em avaliar se a história foi implementada conforme esperado. E para isso, a descrição dos cenários e exemplos de uso, denominados ACs ou casos de teste, é definida a partir das conversas. Eles representam as condições objetivas e mensuráveis que determinam se a história foi implementada corretamente, ou seja, servem como base para a validação pela equipe e pelos usuários ao final da entrega. Esses critérios devem ser escritos em anexo a história de usuário, e a torna mais completa e compreensível a representação de uma iteração.

Leffingwell et al. (2016) enfatiza que o registro desses dois artefatos descritos em conjunto, permite uma validação eficaz, possibilitando entregas consistentes com qualidade e previsibilidade. Para o autor, tais artefatos são fundamentais para eliminar ambiguidades sobre o escopo e o comportamento esperado do *software*; alinhar as expectativas dos *stakeholders* e do time de desenvolvimento; garantir entregas incrementais e contínuas com previsibilidade; facilitar a validação rápida e

objetiva das entregas, reduzindo retrabalho e aceleração de *feedback loops*. Ele associa o uso de BDD, apresentado por North (2006), à definição ACs de USs. Nem sempre a escrita de ACs seguem exatamente uma estrutura específica, podendo haver uma falta de consenso sobre a melhor forma de elaborar ACs (Souza, 2021). Porém, a gramática apresentada por North (2006) se tornou referência considerando a quantidade de trabalhos que a citam ou utilizam. A proposta desta dissertação adota a gramática BDD de North (2006) e incorpora ao *prompt* estruturado modelado pela autora.

O BDD oferece uma estrutura padronizada para descrever o comportamento de um sistema do ponto de vista do usuário final ou de um *stakeholder*, utilizando uma sintaxe baseada em cenários estruturados composta por três partes (Dado, Quando, Então), inicialmente, em "Dado" ("*givens*", em inglês) deve ser apresentado algum contexto inicial, em "quando" deve ser informado um evento que ocorre, e por fim em "então", deve ser apresentado alguns resultados. North et al. (2006) menciona que o comportamento de uma US é simplesmente seus ACs, isto é, quando o sistema cumpre os ACs, ele está se comportando corretamente, quando não, está se comportando incorretamente.

Leffingwell et al. (2016) também recomenda que ACs sejam bem definidos durante a etapa de refinamento das histórias, *Backlog Refinement* ou *Backlog Grooming*, e revisados no planejamento da iteração como parte das condições de prontidão, *Definition of Ready* - DoR (Definição de Pronto, em português) antes de uma história ser aceita para desenvolvimento. DoR é um conceito recomendado em métodos ágeis que, especialmente no Scrum, tem relação direta com a qualidade e a preparação das USs antes de sua implementação. A DoR, é uma lista de critérios que uma tarefa, US ou item do *Backlog* deve atender antes de ser considerada pronta para ser trabalhada pela equipe de desenvolvimento em uma *sprint* ou outro ciclo de trabalho ágil. Basicamente, é um acordo entre a equipe e o PO sobre o que é necessário para que um item do *backlog* seja devidamente entendido e possa ser desenvolvido com sucesso. Dado que as USs são escritas em LN e, com isso, podem variar bastante em termos de qualidade e nível de detalhamento, a DoR atua como um *checkpoint* de preparação definindo o que a US precisa ter antes de ser planejada e comprometida para desenvolvimento.

Um dos *frameworks* mais utilizados para avaliar a qualidade de uma US é o INVEST (acrônimo para *independent, negotiable, valuable, estimable, small e testable*), proposto por Bill Wake em 2003. De acordo com Wake (2003), uma boa US deve ser Independente, Negociável, Valiosa, Estimável, Pequena o suficiente para ser desenvolvida e testada dentro de uma *sprint*, sendo grande demais, deve ser decomposta, e Testável. Cada critério INVEST estabelece aspectos importantes sobre granularidade, valor de negócio, estimabilidade e correspondência da entrega, assegurando que as USs sejam compreendidas, priorizadas e validadas por todo time do projeto. Em razão dessa capacidade de estruturar USs de forma objetiva e eficiente, o *framework* se torna um recurso viável na definição de DoR em projetos ágeis. Essa relação é evidenciada por Leffingwell et al. (2016) em sua obra *Scaled Agile Framework* (SAFe), ao reforçar a importância de critérios explícitos de prontidão e aceitação para itens de *Product Backlog*, como *epics, capabilities, features* e *user stories*. O autor defende que práticas de validação bem definidas, como as encontradas no *framework* INVEST, são indispensáveis para a garantia de que as USs estejam devidamente refinadas, compreendidas e priorizadas antes de ingressarem no fluxo de desenvolvimento.

No contexto de outros métodos ágeis, como Kanban ou XP, o conceito de DoR pode aparecer com outros nomes ou variações, e se estabelece de uma maneira menos forma, mas, ainda assim, com a função de garantir qualidade mínima de preparação para execução. No quadro Kanban, essa prática se manifesta por meio de políticas explícitas de entrada e saída, definidos por cada time de desenvolvimento, que um item de trabalho ou requisito deve atender para ser movido de um estágio (coluna) para outro no processo. No XP, o conceito de preparação mínima aparece de maneira mais integrada à própria rotina de planejamento e programação, de tal modo que enfatiza a prática de testes antes do código, como em TDD, e *Pair Programming*, o que exige histórias bem refinadas e compreendidas para iniciar o desenvolvimento.

2.2 Grandes Modelos de Linguagem

Nos últimos anos, os LLMs têm se consolidado como soluções promissoras para tarefas complexas de PLN. São capazes de gerar, compreender e manipular

texto de forma contextualizada, coerente e adaptável a diferentes domínios de conhecimento (Brown et al., 2020; OpenAI, 2023). Estão treinados sobre grandes volumes de dados textuais, o que lhes permite capturar nuances semânticas e sintáticas, reconhecendo padrões linguísticos e realizando inferências sobre contextos variados. A cada ano, têm apresentado uma evolução notável em seus resultados, com desempenhos significativos em tarefas complexas de PNL, como tradução de idiomas, sumarização de textos, classificação de documentos e raciocínio lógico em respostas a perguntas (Liu Y et al., 2023). A ampla versatilidade dessas soluções tem favorecido sua atuação em setores estratégicos, incluindo educação (Lu & Fan, 2023), medicina (Hsu et al, 2023; Sallam, 2023) e serviços jurídicos (Biswas et al, 2023; Katz et al., 2024). Outras evidências podem ser encontradas no estudo de Virvou (2023) onde é possível encontrar uma análise crítica completa de trabalhos acadêmicos relacionados a IA e UX, ao qual foi motivada pelo objetivo de avaliar a inter-relação e ciclo de causa e efeitos entre essas duas áreas. Foram identificadas aplicações que utilizam LLMs em ES estão ganhando destaque como uma tendência relevante no cenário de pesquisa contemporâneo.

Além dos trabalhos relacionados deste presente estudo empírico, White et al. (2023a) apresentam técnicas de padrões de *prompt* para tarefas de ES, como melhoria da qualidade do código, refatoração, elicitação de requisitos e design de *software* usando o ChatGPT. Seu estudo teve contribuições significativas para o campo do uso de LLMs em ES, onde apresenta um catálogo que categoriza padrões de ES com base nos problemas que abordam no objetivo de aprimorar os processos de elicitação de requisitos, prototipagem rápida, qualidade do código, implantação e testes. A proposta desta pesquisa de Mestrado adotou um padrão semelhante ao escrever *prompts* para gerar recomendações para escrita de ACs orientados a UX, aprimorando as técnicas específicas de *prompt* ao contexto instrucional.

De forma complementar, como vertente oportuna a esse contexto, os Sistemas de Tutoria Inteligente (STI) se destacam por sua capacidade de simular a atuação de tutores humanos, adaptando o conteúdo e a orientação ao perfil e desempenho do usuário em tempo real (Nkambou et al., 2010). Tradicionalmente aplicados no contexto educacional, esses sistemas empregam algoritmos de GenIA

que personalizam a experiência de aprendizagem, fornecendo *feedbacks* direcionados, instruções, diagnósticos e intervenções baseadas no desempenho e comportamento do usuário para apoiá-lo em sua base de conhecimento desejada. Para isso, utilizam modelos de conhecimento, modelos do aluno e estratégias pedagógicas para recomendar conteúdos, práticas ou correções, com o objetivo de promover aprendizagem adaptativa (Anderson et al., 1995; Woolf, 2010). Tomando como modelo essas possibilidades, o PLN pode ser usado em STI para fornecer a alunos, *feedback* sobre suas respostas escritas ou faladas, e ajudá-los a praticar suas habilidades linguísticas de uma forma mais natural. Pesquisadores como Libbrecht et al. (2020) focou na combinação de várias abordagens de aprendizado profundo para ajudar automaticamente os alunos a acelerar o processo de aprendizagem por *feedback* automático, mas também para dar suporte aos professores pré-avaliando o texto livre e sugerindo pontuações ou notas correspondentes. Soluções de tutoria também são aplicadas a esse contexto, como abordado por Virvou (2023) em seu estudo de análise crítica crítica de trabalhos acadêmicos e pesquisas publicadas relacionados a GenIA e UX, que busca explorar a inter-relação e o ciclo de causa e efeito entre os dois. A autora identificou uma aplicação de modelos de instrutor que tem capacidade de auxiliar instrutores pedagógicos a gerenciar o nível de dificuldade e o conteúdo de seu material didático para atender às suas necessidades de ensino e aos pontos fortes e fracos de aprendizagem de seus alunos. Virvou (2023) destaca ainda STIs para o diagnóstico de erros de resposta encontrados nas provas de alunos, sendo capaz de indicar aspectos problemáticos do comportamento ou crenças dos alunos e revelam concepções incorretas sobre o conteúdo. Como desdobramento desses diagnósticos, foi observado que essas soluções de GenIA frequentemente demandam intervenções na interpretação das respostas dos alunos, dado que os erros identificados podem gerar múltiplas hipóteses apontadas pela IA para justificá-los. Em consequência, sobre os erros dos alunos em prova, foi identificado que a GenIA necessita, frequentemente, ser ajustada para a dissolução de conflitos sobre a interpretação dos alunos, visto que tal comportamento se deve possivelmente a muitas hipóteses apontadas para determinar uma resolução.

Transpondo esse conceito para o campo da ER, os LLMs, como o ChatGPT e Gemini, podem ser combinados ao modelo pedagógico de STI para funcionar como

tutores digitais especializados na geração e refinamento de requisitos orientados à UX. Tais soluções podem gerar sugestões personalizadas para melhoria da redação e estrutura desses requisitos, garantindo que os atributos críticos da UX sejam incorporados de forma objetiva e verificável. Em síntese, essa integração representa um caminho conveniente para ampliar a capacidade de times ágeis na validação de USs, otimizando processos de definição de DoR em ambientes colaborativos, assim como, analisar seus ACs, detectando inconsistências, ambiguidades ou oportunidades de melhoria sob a óptica da usabilidade, acessibilidade e valor percebido.

2.3 Engenharia de *Prompts*

O desempenho de LLMs não depende exclusivamente de sua arquitetura ou do volume de dados utilizados em seu treinamento (Stanford University, 2023; Liang et al., 2023; Walsh et al, 2024). Como fator complementar e igualmente determinante, reside a maneira como são orientados a produzir suas respostas, processo conhecido como EP. Uma prática que consiste em projetar, estruturar e otimizar comandos de entrada, chamados *prompts* (Liu P et al., 2023), enviados ao modelo, com o objetivo de obter saídas úteis e contextualizadas. Esse processo se consolidou como uma estratégia indispensável para maximizar a performance dos modelos, pois, apesar de avançados na compreensão de LN, ainda são sensíveis ao contexto das instruções recebidas. Amplamente discutido na literatura e denominado *prompt sensitivity* (Reynolds & McDonell, 2021), a presença desse fenômeno nos resultados gerados pelos LLMs justifica a importância de técnicas especializadas de EP para controlar e orientar de forma mais previsível as respostas dos modelos.

De modo conceitual, um *prompt* pode ser entendido como um conjunto de instruções, enunciado textual, fornecidas a um LLM para refinar ou direcionar suas capacidades em determinada tarefa. Como premissa de EP, os LLMs respondem de forma determinística e probabilística às entradas que recebem. Ou seja, ao processarem informações a partir de sequências textuais de entrada, os modelos tomam decisões com base em padrões estatísticos internalizados durante sua fase de treinamento. Isso significa que, embora as respostas sejam geradas de forma

dinâmica, elas seguem distribuições de probabilidade que refletem as associações, regularidades e tendências presentes no vasto corpus de dados em LN sobre o qual o modelo foi treinado (Brown et al., 2020; Wang H et al., 2025). Em outras palavras, ao receber um *prompt*, o modelo calcula e estima a probabilidade de ocorrência de cada palavra ou possível *token* subsequente, selecionando aqueles que, estatisticamente, são mais coerentes com o conteúdo e intenção expressos na solicitação inicial (Brown et al., 2020; Wang H et al., 2025). Um *token* é a menor unidade de informação que um modelo processa e gera, representados por "palavras" ou "sílabas" que o computador usa para entender a linguagem. Eles atuam como a ponte numérica que transforma o texto humano (LN) em dados que a máquina (Matemática Discreta) pode manipular. Por exemplo, na frase "trabalhando incessantemente", o LLM processa a informação dividindo em tokens: ["trabalh", "ando", " in", "cessantemente"]. O processo de transformar texto em *tokens* é chamado Tokenização. Essa característica estatística intrínseca aos LLMs reforça a relevância de um planejamento criterioso dos *prompts*, visto que instruções mal estruturadas podem induzir respostas imprecisas, incompletas ou desalinhadas ao contexto pretendido.

Como mencionado anteriormente, a literatura especializada propõe aplicar uma formulação no *prompt* como abordagem para disciplinar suas respostas. A forma como essa formulação é aplicada exerce influência direta sobre o comportamento e desempenho do modelo nesse processo determinístico. A falta de cuidado na construção dessas instruções pode resultar em variações na escolha das palavras, na estrutura sintática ou no nível de detalhamento da instrução, o que pode provocar mudanças substanciais sobre suas aptidões contextuais em atributos de qualidade, completude e coerência das respostas produzidas. Diante desse cenário, é estratégico utilizar métodos de solicitação como solução estruturada para qualificar os *prompts* conforme o tipo de tarefa e a complexidade do contexto envolvido.

Adotar estratégias consolidadas, como o *Instructional Prompting* (Sanh et al., 2022), *Few-Shot Prompting* (Brown et al., 2020) e CoT *Prompting* (Wei et al., 2022), para modelagem de soluções a atividades de ES, têm sido consideradas satisfatória na literatura. Em conjunto, se configuram como técnicas que se complementam para induzir LLMs a produzirem respostas mais contextualizadas, precisas e

interpretáveis. *Instructional* fornece detalhes de comportamento esperado do modelo generativo por meio de instruções explícitas sobre o que fazer, como fazer e com qual objetivo. O método *few-shot* ensina o modelo por meio de exemplos do contexto para que aprenda o padrão esperado e replique no contexto proposto. Já o CoT, conduz o modelo a explicitar seu “raciocínio” passo a passo.

Foram verificadas algumas referências sobre o método *instructional* que guiaram a condução das tarefas avaliativas pelos LLMs neste presente estudo. Sahoo et al. (2024) destaca técnicas baseadas em instruções explícitas, *instructional prompting*, focando em clareza, coerência, e formatação para apoiar o raciocínio CoT. Os pesquisadores analisaram mais de 29 diferentes técnicas de EP, distribuídas por áreas de aplicação como raciocínio, geração de código, perguntas e respostas, resumo e lógica, a fim de organizar uma tabela comparativa com metodologias, tarefas, modelos e datasets empregados em cada abordagem. As evidências apontaram exemplos que incluem *prompts* estruturados guiando o modelo operar sob essa combinação, com instruções formais para formatos de saída, *prompts* condicionais e diretivos. Notaram, que para atingir essa composição, envolveu estabelecer condicionais que limitam tom de voz, formato, estilo e procedimentos lógicos, como por exemplo, “Liste em tópicos”, “Use no máximo X palavras”, “Explique antes de concluir”. Ou seja, estabelecer comandos objetivos e claros sobre o que se pretende executar para ter êxito nas saídas de *prompts* modelados por *instructional* e CoT. Essa evidência teve influência direta na formulação do *prompt* instrucional, que incorporou comandos objetivos e sequenciais, alinhados às boas práticas identificadas na literatura.

A nível de desempenho, o método *instructional* se mostra uma boa alternativa para guiar os modelos em tarefas específicas. Zhou et al. (2022) modelaram um editor automático de instructional *prompts* para maximizar o desempenho *zero-shot* em 24 tarefas representativas de PLN, como classificação de texto, análise de sentimentos, perguntas e respostas, inferência textual, entre outros. Os resultados equiparam aos *prompts* escritos manualmente por especialistas em 19 das 24 tarefas testadas com ganho considerável de precisão, validando a capacidade de LLMs aprender a estruturar *prompts* sozinhos. Esse desempenho adicional reforça o potencial do *instructional prompting* como método de controle interpretativo, sobretudo em cenários técnicos especializados, como os exigidos nesta pesquisa.

Wei et al. (2022) demonstra que quando se encoraja o modelo a gerar raciocínio explicitando etapas lógicas, a capacidade dos LLMs em tarefas complexas, como aritmética, lógica e inferência, melhora significativamente, tanto em precisão quanto em confiabilidade. A exemplo, Mishra et al. (2022) conduziram um estudo empírico comparando versões originais de *prompts* com versões reformuladas passo a passo (CoT), e nesse último, encontraram ganhos expressivos de 12,5 % nos LLMs GPT-3 e 6,7 % nos modelos GPT-2, mostrando que *prompts* estruturados são mais eficazes. Um estudo em ambiente clínico (Sivarajkumar et al., 2024) comparou vários tipos de *prompts* (*prefix* heurístico, CoT e *few-shot*) e mostrou que essa estratégia aumenta a precisão em tarefas como extração de evidências, desambiguação e resolução de correferência. O uso de CoT e *prompts* heurísticos levou GPT-3.5 a alcançar até 96% de acurácia, enquanto few-shot também apresentou melhoras quando se lida com cenários complexos.

CoT*, instructional* e *few-shot* permitem não apenas aprimorar a performance dos LLMs em tarefas específicas, mas também contribui para aumentar a confiabilidade das respostas em domínios de decisão crítica, como medicina, jurídico ou engenharia de *software*. Bem como, é especialmente relevante em contextos onde as respostas geradas podem implicar diretamente em decisões estratégicas ou de impacto operacional, como ocorre na validação de ACs em USs, foco desta pesquisa.

Nesse campo, estudos indicam que os LLMs podem também aperfeiçoar a qualidade de USs e seus ACs. Assim como, apresentado por Zhang Z et al. (2024) que propuseram o agente inteligente ALAS, capaz de gerar versões refinadas de USs, melhorando atributos como clareza e valor de negócio, embora com tendência à excessiva verbosidade nos ACs, aspecto que os próprios autores apontam como uma limitação, sugerindo a necessidade de moderação e refinamento iterativo. Outra iniciativa abordada por Brockenbrough, Feild & Salinas (2025), constata que o uso de LLMs pode auxiliar estudantes na redação de ACs seguindo o padrão INVEST (Wake, 2003), ainda que com limitações na definição de escopo. Essas evidências apontam que, quando bem orientados, os LLMs podem atuar como tutores automáticos, sugerindo melhorias estruturais nas USs, incluindo ACs. A exemplo dos estudos apresentados, ao sistematizar e aplicar métodos robustos de

EP, é possível modular o comportamento dos LLMs, otimizando sua atuação em atividades de suporte à ER e análise de UX.

Propomos nesta pesquisa um *prompt* estruturado que combina os três métodos de solicitação, *instructional*, *few-shot* e CoT, permitindo que os modelos ChatGPT-4o e Gemini 2.5 Flash atuem como STI no apoio à validação de ACs de USs elaboradas por times ágeis de desenvolvimento. Essa proposta se alinha à perspectiva de modular o comportamento dos LLMs por meio dessas estratégias de EP, otimizando sua contribuição em atividades de suporte à ER e à análise de UX, a exemplo dos estudos relacionados. Com isso, vislumbra-se, portanto, a viabilidade de conceber mais um recurso de apoio à ER, aderente às expectativas de UX e capaz de ampliar a confiabilidade das soluções digitais desenvolvidas em ambientes ágeis. Essa abordagem não apenas avança o estado da arte no uso de LLMs para processos de validação de requisitos, mas também propõe um novo arranjo metodológico para incorporar critérios de usabilidade, acessibilidade e valor percebido já nas etapas iniciais de especificação e validação de *software*.

2.4 UX e Framework de Garrett

A UX ocupa um papel importante no sucesso de produtos digitais devido à influência que exerce sobre a satisfação, a eficiência e a fidelização dos usuários (Garret, 2011). Em ES, considerar atributos de UX é uma condição determinante para assegurar a entrega de soluções não apenas funcionais, mas também desejáveis, acessíveis e intuitivas, inclusive, desde as etapas iniciais de concepção e validação de requisitos (Pressman & Maxim, 2020).

O conceito de UX ultrapassa aspectos estéticos ou técnicos, abrangendo percepções, emoções (ISO-9241-210, 2019), respostas cognitivas e comportamentais de um usuário ao interagir com um sistema ou interface (Hassenzahl et al., 2000). Jakob Nielsen (1994) e Don Norman (2013), dois autores de referências na área, definem UX como todas as interações que um usuário vivencia com uma organização, seus serviços e produtos, englobando não apenas usabilidade, mas também acessibilidade, valor percebido e satisfação subjetiva (Norman & Nielsen, 1998). Entende-se, portanto, que a experiência não está restrita

à interface ou ao uso direto do sistema, mas inclui todo o ecossistema de contato, antes, durante e depois da utilização de um produto digital.

Por essa leitura, vemos desafios em times de desenvolvimento onde o conceito de UX está envolvido. Assim como destacado por Norman (2013) ao mencionar que a boa UX é resultado da combinação sobre sua avaliação da usabilidade eficiente e prazer emocional com o produto. Nielsen (1994) frequentemente menciona essa perspectiva ao propor princípios heurísticos de usabilidade que se apresentam como referências práticas na avaliação de interfaces, como visibilidade do status do sistema, controle e liberdade do usuário e consistência na interface. Tais diretrizes influenciam diretamente a ER, especialmente na elicitação de USs e ACs, ao garantir que as necessidades e expectativas dos usuários finais sejam priorizadas nas definições iniciais do produto.

Já se tem percebido que a integração de princípios de UX nas fases de levantamento e definição de requisitos, em especial, na criação de USs, promove ganhos práticos na antecipação de problemas de usabilidade, e consequentemente, na redução de retrabalho (Leffingwell et al., 2016). Ao considerar dimensões como eficiência, eficácia, atratividade e acessibilidade desde a elicitação, os times de desenvolvimento são capazes de planejar entregas iterativas mais alinhadas às expectativas dos usuários e aos objetivos de negócio. Exemplos claros desse amadurecimento podem ser observados em cases de empresas digitais como Spotify e Airbnb. Apesar da reconhecida relevância, muitas empresas de desenvolvimento ainda não conseguem oferecer em seus produtos e serviços uma boa UX, além de que, os profissionais dessas empresas, enfrentam vários desafios ao lidar com a UX em todos os seus projetos (Kashfi et al., 2017).

No Spotify, Kniberg & Ivarsson (2012) descreveram no artigo como a companhia estruturou suas *squads* e *chapters* para sustentar um modelo de desenvolvimento ágil escalado. Elas detalham a influência princípios culturais estruturantes, práticas de autonomia e alinhamento, ciclos iterativos e integração contínua. Por essa ótica, há de convir que esse nível de maturidade que o modelo ágil do Spotify possibilita a integração de UX nas squads, e que práticas como antecipação de cenários de uso são coerentes com os valores da cultura ágil que prioriza a visão do usuário.

Em entrevistas publicadas durante a *UXDX 2019 Conference*, a Airbnb, representada pelo então Diretor de Pesquisa, Judd Antin, compartilhou práticas adotadas pela empresa onde os times de produto relataram que, ao integrar UX *Research* contínuo às cerimônias ágeis e inserir especialistas de UX no planejamento de *backlog*, contribuiu para o fortalecimento da posição da empresa no mercado digital e aprimorou entrega de valor de seus resultados relacionados à satisfação dos usuários (Antin, 2025). Embora métricas quantitativas específicas não tenham sido formalmente publicadas sobre os impactos diretos dessa estratégia, a Airbnb atribui parte de seu diferencial competitivo à cultura de design colaborativo e à incorporação estruturada de UX no ciclo ágil. Notoriamente, a abordagem resultou em processos de desenvolvimento mais responsivos aos clientes, alinhados aos objetivos estratégicos do negócio que evidenciaram a importância da UX como um dos fatores de influência para a qualidade percebida e a aceitação das soluções digitais entregues.

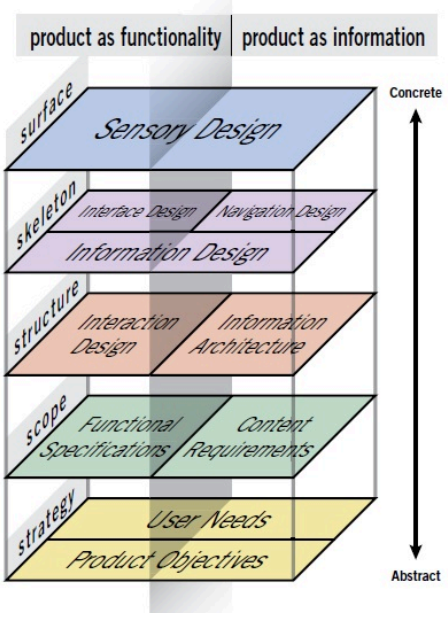
Em em conformidade com a literatura sobre integração entre UX e métodos ágeis, pesquisas práticas indicam que equipes que incorporam heurísticas de usabilidade e dimensões de UX desde as fases iniciais de elicitação tendem a identificar problemas de usabilidade de forma antecipada (Nielsen et al., 1990, Nielsen, 1994; Leffingwell et al., 2016; Sharp et al., 2019). Essa perspectiva é reforçada por Sharp et al. (2019) que relaciona a importância de antecipar requisitos de UX desde as fases iniciais do projeto para evitar falhas de usabilidade. Os autores alertam que, muitas vezes, as falhas só são percebidas após a entrega do produto, gerando retrabalho, custos adicionais e insatisfação do usuário.

Esses exemplos evidenciam que a prática defendida por Leffingwell et al. (2016) é respaldada por resultados empíricos no mercado e literatura científica recente. A antecipação de atributos de UX nas USs permite não só iterar entregas mais funcionais, mas também gerar experiências desejáveis, o que, no mercado atual de produtos digitais, se traduz em vantagem competitiva e otimização contínua de valor.

Para apoiar e sistematizar esse processo, diferentes *frameworks* conceituais foram desenvolvidos ao longo dos anos. Entre os exemplos mais influentes na prática da área de UX Design, está a proposta de Jesse James Garrett, apresentada

originalmente em 2002 e consolidada em 2011 em sua obra "*The Elements of User Experience: User-Centered Design for the Web and Beyond*". Garrett (2011) propôs uma estrutura de cinco planos hierarquicamente organizados, que se descrevem como camadas de decisão no processo de desenvolvimento de qualquer produto digital orientado ao usuário: 1. Estratégia, 2. Escopo, 3. Estrutura, 4. Esqueleto, e 5. Superfície. A UX está relacionada com a experiência que um produto cria nos usuários quando utilizado (Garrett, 2011). Garrett (2011) afirma que projetar para UX é compreender as expectativas, e também considerar as ações dos usuários. Como mencionado anteriormente, um dos assuntos abordados em ES é a importância da comunicação para facilitar a compreensão do usuário ao utilizar o produto, e o autor aborda justamente como estruturar e organizar a informação de modo a tornar o uso do produto simples e fácil para o usuário. Assim, o *framework* proposto por Garrett (2011) se apresenta, então, como instrumento sistemático a esse processo de *software*.

Figura 1: Framework de Garrett



Fonte: *The Elements of User Experience* (Garrett, 2011).

A metodologia considera iniciar o planejamento do produto e seu o design pelas camadas mais abstratas, e avançar progressivamente até as camadas mais concretas ao visual do produto (Garrett, 2011). A camada da Estratégia consiste em fundamentar aspectos que levará a criação de uma forma de interação (interface) ligados exclusivamente aos objetivos do projeto e as necessidades do usuário. A

exemplo disso, pode explicitar e se alinhar as expectativas de um Indicador-Chave de Desempenho (*Key Performance Indicator* - KPI, em inglês) ou métrica de usabilidade, como diminuir o tempo do processo ou aumentar o engajamento de um perfil de usuário X ao produto. O Escopo define os requisitos funcionais e de conteúdo que atenderão a essas necessidades e objetivos, e se apresenta por meio de tabelas descritivas que exponham as limitações e abrangências do projeto, assim como as exclusões. Usando como exemplo um *website*, se refere à definição de quais recursos, funções e informações devem ser aplicadas no *website*, com a proposta de melhorar a interação entre o usuário e o sistema.

A camada de Estrutura organiza a disposição lógica dos elementos de UX na interface e o fluxo de interação do usuário com o sistema. É a partir dessa camada que os primeiros elementos de UX se apresentam, legitimamente, para compor o desenvolvimento de *software* orientado ao usuário. Ou seja, é nessa fase que o projeto começa a criar forma através de fluxogramas, *sitemap* ou mapa mentais que definem a estrutura (Arquitetura da Informação) e o fluxo de navegação (Design de Interação) que o usuário irá percorrer. No Esqueleto, o foco recai sobre a estruturação da interface, isto é, a organização espacial dos elementos (Design da Informação), a navegação (Design de Navegação) e a disposição dos componentes na tela (Design de Interface). Esse é o momento de pensar em *wireframes* ou outras abstrações da interface como uma prévia a abordagem visual que será aplicada ao produto. Por fim, a camada da Superfície é a dimensão visual do produto, em que aspectos estéticos finais (Design Sensorial) são trabalhados para garantir uma experiência fluida, intuitiva e visualmente coerente, como tipografia, cores, ícones, elementos gráficos e aplicação de padrões do *Design System*, assim como as microinterações e transições.

Essa hierarquia bem definida permite assegurar que cada decisão de design esteja fundamentada em justificativas estratégicas e em evidências de necessidades reais dos usuários, reduzindo decisões subjetivas. O modelo de Garrett dialoga diretamente com as contribuições de autores como Donald Norman (2013) e Jakob Nielsen (1994), que desde os anos 1990, ressaltaram a importância de garantir experiências digitais satisfatórias e eficientes.

Além disso, essa abordagem se mostra especialmente relevante para o contexto central desta pesquisa, no apoio à validação de ACs com aspectos de UX usando LLMs. Como evidenciado por Leffingwell et al. (2016), ACs estruturados segundo as práticas de BDD, contribuem para uma validação mais eficaz das entregas finais, garantindo alinhamento técnico e funcional. No entanto, sob a perspectiva orientada por Garrett (2011), os ACs, sendo amplamente reconhecidos como um dos itens elementares a ER no processo de desenvolvimento, podem ter sua validade prejudicada ao serem elaborados sem considerar os atributos de usabilidade, acessibilidade, percepção de valor e satisfação do usuário, comprometem a entrega de soluções efetivamente centradas na experiência. Ao estruturar *prompts* para LLMs que incorporem aspectos da estrutura proposta por Garrett, torna-se possível orientar a geração automática de ACs mais robustos e alinhados às expectativas de UX.

O estudo desenvolvido nesta dissertação utiliza um *prompt* elaborado pela autora com características STI que utiliza as *guidelines* ACUX de Souza (2021) como base de conhecimento para o LLM. A Tabela 2 apresenta todas as *guidelines* ACUX em detalhes. Ao todo são 15 *guidelines* distribuídas em dois grupos principais relacionados à UX: (1) Design da Interação e Organização da Informação, e (2) Elementos Visuais, os quais abordam níveis diferentes do tratamento da UX no produto.

Tabela 2: ACUX *Guidelines*

ACUX Guidelines			
● **Design da Interação e Organização da Informação**		● **Elementos Visuais**	
DI-01	Especificar como o usuário interage com a funcionalidade do sistema.	**EV-01**	Especificar os elementos visuais mais adequados para que o usuário possa realizar suas tarefas.
DI-02	Especificar como se chega à determinada tela, ou os caminhos que o usuário pode seguir quando está nela.	**EV-02**	Especificar a organização dos elementos na tela de modo que sejam prontamente entendidos e facilmente usados pelos usuários.
DI-03	Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação.	**EV-03**	Especificar detalhes sobre como apresentar a informação, para que o usuário a entenda com mais facilidade, como gráficos e imagens.
DI-04	Especificar detalhes sobre como a	**EV-04**	Especificar elementos que possibilitam

	informação está disposta, e como se dá a ligação entre uma informação e outra.		ir de um ponto a outro no sistema.
DI-5	Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue.	**EV-05**	Especificar detalhes sobre estilo.
DI-06	Especificar a sequência em que a informação deve ser apresentada para que facilite a interação.	**EV-06**	Considerar paleta de cores, tipografia, guia de estilo/identidade visual.
		EV-07	Especificar detalhes sobre fontes, cores, formas que relacionam o estilo adotado no produto/projeto.
		EV-08	Especificar detalhes sobre contraste, destacando o que os usuários realmente precisam ver.
		EV-09	Especificar detalhes para manter a uniformidade do projeto (manter o tamanho dos elementos uniformes) considerando a identidade visual.

Fonte: ACUX: Um Guia para Escrita de Aspectos de UX em Critérios de Aceitação de User Stories (Souza, 2021) - Cartilha Online ACUX.

O autor utilizou estudo de caso (Wohlin et al., 2012) como método de pesquisa para avaliar o uso do ACUX em um cenário real de projetos da indústria da tecnologia com dois times de desenvolvimento de duas *startups*. Shull et al. (2007) aponta que o método é apropriado para situações em que o contexto pode interferir, como por exemplo, as pressões de um projeto real, as quais afetam o comportamento de desenvolvedores. Nesse sentido, a abordagem confere maior compreensão dos fenômenos em estudo dentro do contexto real avaliado (Wohlin et al., 2012).

Na condução de seu estudo, Souza (2021) coletou as USs/ACs elaboradas pelas *startups*, analisou trechos de cada um deles aplicando sua abordagem ACUX e documentou as análises sobre a escrita dos ACs em uma planilha estruturada. O autor compartilhou online alguns exemplares[1] de sua análise disponibilizados em tal planilha, ilustrando os ACs analisados para demonstrar na prática a aplicação das *guidelines* do ACUX. A planilha desenvolvida por Souza (2021) possui uma estrutura replicável para estudos exploratórios, e, por isso, foi considerada no estudo empírico, servindo como instrumento documental de registro, categorização

[1] Exemplares de USs/ACs de Jonathan Souza (2021) com aplicação da abordagem ACUX https://docs.google.com/spreadsheets/d/1uMyGXuOrq3YtQEGct0xBCS1RZsulosoDEqwG_tA3Y3g/edit?gid=1948740601#gid=1948740601

e comparação das recomendações de melhoria indicadas pelas *guidelines* ACUX. O Capítulo 5 detalha o uso da abordagem ACUX em benefício à fase exploratória deste estudo.

Em sua fundamentação teórica, a ACUX foi diretamente baseada nas camadas propostas por Garrett, em especial às camadas de Estrutura, Esqueleto e Superfície, por representarem os níveis que mais influenciam a interação do usuário com o produto. Em metodologia, as *guidelines* de Souza (2021) foram criadas a partir de um Mapeamento Sistemático da Literatura (MSL), reunindo evidências concretas sobre as abordagens existentes sobre ACs, USs e UX. Os resultados confirmaram a ausência de estudos focados em ACs, e possibilitaram estabelecer uma relação entre as abordagens existentes e as ações necessárias para elaboração da proposta ACUX (Souza, 2021). Em seguida, o autor elaborou um estudo exploratório inicial com desenvolvedores a fim de identificar como eles descrevem aspectos de UX em ACs. Os dados foram analisados com o propósito de identificar elementos de UX nas descrições dos ACs, tendo como base o *framework* de Garrett (2011), e técnicas de *coding* (Strauss; Corbin, 1998). Os resultados apontaram que aspectos de UX podem ser incluídos em ACs. Com isso, o ACUX foi proposto considerando informações extraídas do MSL e dos resultados do estudo exploratório, além de ter sido validado por quatro especialistas, sendo estes das áreas de: ES, Interação Humano-Computador, e UX.

Após todo processo de validação, a versão final ACUX foi definida compondo 15 *guidelines* distribuídas conforme a Tabela 2. Os participantes da avaliação das *startups* apontaram em *feedbacks* qualitativos que a abordagem foi capaz de conduzir a escrita de aspectos de UX em ACs, e ainda promoveu a troca de discussões e informações relevantes de UX nos projetos aos quais estavam atuando.

2.5 Considerações Finais

O Capítulo 2 apresenta a base conceitual que fundamenta o desenvolvimento desta pesquisa, articulando quatro eixos teóricos principais: Métodos Ágeis e USs, LLM, EP e UX. A revisão evidenciou que os métodos ágeis, ao privilegiarem entregas incrementais e colaboração contínua, oferecem um ambiente propício para a

integração de práticas de validação e refinamento de requisitos centradas no usuário. Nesse contexto, o uso de US e AC se mostra essencial para garantir clareza, alinhamento e rastreabilidade, reforçados por frameworks como INVEST e pela adoção da DoR.

Em paralelo, o capítulo reconhece que o desempenho dos LLMs, quando fundamentalmente moldados pela EP, vai além da simples geração de conteúdo, alcançando níveis significativos em tarefas complexas de PLN, como raciocínio lógico e inferência. Essa versatilidade consolida os LLMs como a tecnologia base ideal para a ambição desta pesquisa: fornecer um suporte automatizado e inteligente ao time de desenvolvimento, atuando diretamente no refinamento de requisitos. O potencial está, precisamente, em sua capacidade de capturar nuances semânticas e sintáticas inerentes aos ACs de USs.

Essa orquestração tecnológica eleva o LLM ao patamar de um STI que fornece *feedback* personalizado e adaptativo. Transpondo este conceito para a ER, o modelo pode agir como um tutor digital que não apenas aponta e analisa as inconsistências e ambiguidades dos requisitos existentes, especificamente sob a óptica da UX. Essa função de STI é particularmente relevante para o contexto ágil, onde o tempo de refinamento é limitado, atuando como um *checkpoint* de qualidade automatizado que suporta a fase de *Backlog Refinement* e o cumprimento da DoR. Para isso, a modelagem criteriosa do *prompt* torna-se um artefato de design vital para esta dissertação, garantindo que a resposta do LLM seja previsível, coerente com o contexto da ER e diretamente utilizável em ambientes ágeis.

Por fim, a abordagem de ACUX e o *framework* de Garrett reforçaram que a qualidade de um produto digital transcende sua funcionalidade, abrangendo aspectos emocionais, cognitivos e de valor percebido. A integração de princípios de UX desde as etapas iniciais de definição de requisitos favorece a antecipação de problemas, reduz retrabalho e fortalece a UX como elemento norteador de decisões de design e engenharia. A dificuldade prática em times ágeis reside, historicamente, na tradução da subjetividade da experiência (emoções, percepções) em artefatos objetivos e verificáveis. A relevância desta pesquisa consiste, portanto, em fornecer o elo metodológico para que a UX se torne um requisito passível de validação ainda na DoR do desenvolvimento ágil.

Em síntese, os fundamentos teóricos delineados sustentam a proposta desta pesquisa ao evidenciar que a combinação entre práticas ágeis, técnicas de *prompt engineering* e princípios de UX pode constituir um novo arranjo metodológico para aprimorar a validação de requisitos em ambientes colaborativos. Essa base conceitual fornece suporte à metodologia apresentada no próximo capítulo, que descreve o estudo empírico envolvido, os procedimentos de coleta e as estratégias de análise adotadas para avaliar a eficácia da proposta.

3 Metodologia

Este trabalho adota o estudo empírico de caráter exploratório e comparativo como método de pesquisa, fundamentado no segmento emergente NLP4RE (Dalpiaz et al., 2018; Zhao L et al., 2021) e em diretrizes publicadas pela Comunidade Internacional de Pesquisa em ES (*The International Software Engineering Research Network* - ISERN, em inglês) (Wagner et al., 2025; Baltes et al., 2025). A estratégia metodológica buscou integrar práticas de ER, UX e LLMs. Por meio desta, o estudo tem como foco principal investigar a capacidade prática de dois LLMs atuarem como tutores automatizados para apoiar a validação da escrita de ACs sob a perspectiva da UX, propondo uma abordagem denominada ACUX Tutor 1.0.

De tal maneira, a pesquisa combina investigação qualitativa preliminar e procedimentos sistemáticos de testagem. A condução metodológica está organizada em cinco etapas principais, representadas no *pipeline* da Figura 2.

Figura 2: *Pipeline* metodológica do estudo empírico

Fonte: Elaborado pela autora.

Essa estrutura foi inspirada em trabalhos empíricos prévios que aplicaram LLMs em tarefas de ER, notadamente:

- **Ronanki et al. (2023)**, que conduziram comparações entre GPT-3.5 e especialistas humanos para elicitação de requisitos;
- **Zhang Z et al. (2024)**, que desenvolveram um sistema de agentes autônomos para aprimoramento automático de USs;
- **Krishna et al. (2024)**, que avaliaram a capacidade de LLMs em gerar e validar SRSs de forma comparável à humana;
- **Karlsson et al. (2025)**, que investigaram a confiabilidade de classificações automáticas de requisitos funcionais e não funcionais por meio de estudo experimental; e
- **Kolthoff et al. (2024)**, que aplicaram LLMs para detectar inconsistências entre USs e protótipos de interface.

Essas referências serviram de base metodológica para definir os parâmetros do estudo empírico, garantindo consistência com práticas científicas recentes na

área de ER assistida por GenIA. A fase exploratória teve como objetivo compreender o fenômeno investigado e estruturar o *corpus* de análise. Para isso, foram conduzidas duas atividades principais: (i) um *Survey* sobre DoR, a fim de levantar percepções e práticas relacionadas à validação de requisitos em ambientes ágeis; e (ii) uma Análise de Conteúdo (Bardin, 2016) realizada por um avaliador humano aplicada à ACs escritos em LN no formato BDD, com o propósito de verificar a aplicabilidade prática da técnica ACUX (Souza, 2021) e estabelecer uma base empírica para as etapas subsequentes.

Já a fase comparativa permitiu a execução prática, com manipulação de variáveis, neste caso, o mesmo conjunto de ACs utilizado na análise de conteúdo (ii), o *prompt* instrucional, a execução dos LLMs, e a mensuração dos resultados. O que, portanto, compreendeu as etapas de (iii) modelagem do *prompt* instrucional, seguidas pela (iv) execução dos modelos ChatGPT 4.0 e Gemini 2.5 Flash, e, por fim, pela (v) análise e interpretação dos resultados. Essa estrutura metodológica possibilitou que o estudo avançasse da exploração do problema e compreensão contextual para a testagem das variáveis, assegurando a robustez das inferências sobre precisão técnica, concordância e explicabilidade das recomendações geradas pelos LLMs.

Todas as etapas foram conduzidas sob princípios éticos da pesquisa científica, garantindo anonimização dos dados e uso exclusivo de conteúdos acadêmicos e públicos. O uso dos modelos ChatGPT-4o e Gemini 2.5 Flash respeitou as licenças e termos de uso das plataformas originais. A reprodutibilidade foi assegurada pela documentação dos *prompts*, parâmetros e critérios de avaliação, permitindo replicação em estudos futuros.

A estrutura metodológica proposta reflete uma progressão entre diagnóstico, modelagem e validação. E por fim, a triangulação entre ER, UX e LLM, apoiada em estudos recentes (Ronanki et al., 2023; Zhang Z et al., 2024; Karlsson et al., 2025; Kolthoff et al., 2024; Krishna et al., 2024), assegura que o ACUX Tutor 1.0 seja avaliado sob perspectivas complementares (técnica, cognitiva e perceptiva), refletindo uma proposta metodológica potencialmente viável, replicável e aderente às boas práticas de pesquisa em ES.

4 *Survey* Validação de Requisitos com DoR

Foi conduzida uma pesquisa exploratória e descritiva com o objetivo de compreender a percepção de profissionais da área de Tecnologia acerca da influência de DoR na validação de requisitos, e de que maneira seria possível alinhar essa prática em benefício da UX nos projetos. A revisão da literatura forneceu subsídios para examinar mais profundamente quais recursos disponíveis atualmente no meio acadêmico, relacionados à UX, poderiam ser utilizados para apoiar essa relação, considerando que a integração entre práticas ágeis e aspectos de UX continua a apresentar desafios relevantes (Ananjeva et al., 2020; Zaina et al., 2022).

Ampliando esse cenário, o estudo realizado nesta dissertação conecta o *survey* às *guidelines* ACUX (Souza, 2021) com a motivação de identificar *insights* que possam reduzir lacunas na prática de validação de requisitos e, ao mesmo tempo, facilitar o processo de comunicação entre *stakeholders*. A Seção 4.2 reúne os resultados que contribuíram para expressar a criação da abordagem deste estudo, e a relação interpretada pela autora entre DoR e ACUX.

Tradicionalmente, os acordos DoR normalmente cobrem clareza, valor de negócio e testabilidade das USs. Por essas mesmas USs, as *guidelines* ACUX têm a possibilidade de expandir o escopo de validação DoR ao introduzir atributos de usabilidade, consistência e valor percebido em ACs, que geralmente ficam implícitos nos requisitos. A hipótese assumida é que, ao adotar ACUX na formulação de ACs, uma US tenha maiores chances de considerar a visão do usuário, cliente ou *stakeholder*, e apoiar o time não somente com requisitos que já foram desenhados, mas também especificados e validados do ponto de vista do usuário a fim de atingir uma definição de “pronto” mais alinhada às expectativas do projeto (Lucassen et al., 2015).

O *survey* buscou investigar também se há espaço para a participação do cliente na fase de DoR em projetos de *software* em contribuição à validação de requisitos, tendo, como instrumento principal, a revisão e o *feedback* do cliente nos requisitos escritos pelo time. A inclusão dessa dimensão colaborativa se justifica pelo reconhecimento de que a participação ativa do cliente pode potencializar a

identificação de oportunidades de melhoria e a detecção precoce de problemas, contribuindo diretamente para a adequação do produto (Chamberlain et al., 2006; Nebeker et al., 2019). É importante ressaltar que, para fins deste estudo, o termo "cliente" no *survey* abrange tanto os clientes internos (especialmente, *stakeholders* que tenham envolvimento direto com a solução desenvolvida) quanto usuários finais.

A escolha por uma abordagem exploratória se justifica pela necessidade de aprofundar a compreensão acerca da temática desta pesquisa, além de mapear possíveis oportunidades e cenários favoráveis à adoção de práticas colaborativas de validação de requisitos ao contexto ágil. Dessa forma, o estudo desenvolvido no *survey* serviu de inspiração base para a criação das perguntas de pesquisa desta dissertação, além de oferecer indícios práticos de caminhos promissores para integrar aspectos de UX no processo de definição de prontidão de requisitos.

4.1 Estrutura do *Survey*

Para coleta dos dados, um questionário online[2] foi elaborado no Google Forms e disponibilizado em canais acadêmicos, redes sociais, e comunidades de tecnologia, visando alcançar profissionais com experiência prática no processo de definição e validação de requisitos em projetos de *software*. A divulgação ocorreu entre novembro de 2024 a maio de 2025. O questionário contou com 16 questões fechadas obrigatórias e 4 questões abertas opcionais para contato com os participantes, escritas em dois idiomas, português e inglês, considerando a possibilidade de alcançar profissionais de diferentes contextos culturais e ampliar a diversidade da amostra. Buscou levantar informações sobre o perfil dos participantes, tempo de experiência na função, principais dificuldades enfrentadas para considerar um requisito "pronto", formas de atuação do cliente na revisão de requisitos e sugestões de mudanças para promover a colaboração ativa do cliente nas validações.

Participaram da pesquisa 31 profissionais da área de Tecnologia, com atuação em diferentes funções relacionadas ao desenvolvimento de *software*, mais

[2] Link do Questionário:
https://docs.google.com/forms/d/1OXKv_y0QmL8PNHZHli46teJHgJe8s7zs1h1PVtQQzzY/viewanalytics

especificamente, incluindo analistas de requisitos, gerentes de produto, desenvolvedores, analistas de QA e designers. A participação na pesquisa foi voluntária, sem qualquer tipo de incentivo financeiro. Os dados foram coletados de forma anônima, garantindo o sigilo das informações pessoais dos participantes, em conformidade com as diretrizes éticas de pesquisa em ambientes digitais. O questionário foi estruturado em cinco seções:

- **Seção 1** - **Perfil Profissional**: Coleta de informações sobre identidade de gênero, função atual, tempo de atuação na área e experiência com validação de requisitos.
- **Seção 2** - **Validação de Requisitos**: Levantamento sobre frequência de mudanças em requisitos, métodos de validação utilizados e principais dificuldades em considerar um requisito “pronto” para desenvolvimento.
- **Seção 3** - **Envolvimento do Cliente**: Investigação sobre a participação dos clientes internos e externos na revisão de requisitos em contribuição a DoR nos projetos, e o *feedback* dos participantes quanto à inclusão do cliente nas discussões de melhoria dos requisitos.
- **Seção 4** - **Cenários de Revisão de Requisitos**: Coleta de percepções sobre cenários favoráveis à alguma participação do cliente, papéis que poderiam ser assumidos por ele na revisão de requisitos ou mudanças necessárias para viabilizar essa colaboração.
- **Seção 5** - **Feedbacks Finais**: Levantamento de sugestões, críticas e relatos sobre a experiência profissional dos participantes na validação de requisitos e suas opiniões sobre a temática envolvida na pesquisa exploratória.

A Tabela 3 detalha todas as perguntas do questionário.

Tabela 3: Perguntas do *Survey* sobre DoR.

Seção 1 - PERFIL PROFISSIONAL		
01	Qual sua identidade de gênero? *(Pergunta fechada resposta única obrigatória)*	☐ Mulher cisgênero ☐ Homem cisgênero ☐ Mulher transgênero ☐ Homem transgênero ☐ Gênero-fluido ☐ Não-binário

		☐ Agênero ☐ Outros
02	Qual é a sua atuação profissional? *(Pergunta fechada resposta única obrigatória)*	☐ Analista de Requisitos (PM, PO, Gestor de Projeto) ☐ Designer ☐ Analista QA ☐ Desenvolvedor ☐ Outros
03	Há quanto tempo você atua nessa função? *(Pergunta fechada resposta única obrigatória)*	☐ Menos de 1 ano ☐ 1 a 3 anos ☐ 4 a 6 anos ☐ Mais de 6 anos
04	Você trabalha em equipes que têm interação direta com clientes ou usuários finais? *(Pergunta fechada dicotômica obrigatória)*	☐ Sim ☐ Não
05	Você tem experiência com validação de requisitos? *(Pergunta fechada dicotômica obrigatória)*	☐ Sim ☐ Não
Seção 2 - VALIDAÇÃO DE REQUISITOS		
06	Com que frequência os requisitos mudam após o início do desenvolvimento no seu time atual? *(Pergunta fechada resposta única obrigatória)*	☐ Mudam o tempo todo ☐ Mudam com frequência ☐ Mudam de vez em quando ☐ Mudam em situações excepcionais ☐ Nunca mudam
07	Como seu time costuma validar os requisitos atualmente? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Validam apenas internamente (entre o time técnico e o time de negócios) ☐ Validam com feedback direto do cliente/usuário final ☐ Revisamos com todas as partes interessadas (time + cliente/usuário) ☐ Por meio da documentação técnica e em feedbacks pontuais ☐ Por meio de testes de usabilidade, protótipos, etc ☐ Não costumam validar requisitos ☐ Outros
08	Quais as principais dificuldades em considerar um requisito "pronto" para ser repassado ao time de Desenvolvimento? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Falta de cerimônias ágeis, como refinamentos ou plannings ☐ Tamanho inadequado das tarefas a serem concluídas nas *sprints* ☐ Falta de validação com o cliente ou usuário final ☐ Falta de clareza no objetivo dos requisitos ☐ Critérios de aceitação incompletos ou indefinidos ☐ Dependências não resolvidas ou não

		identificadas ☐ Outros
09	Atualmente, seu time realiza sessões de *Definition of Ready* (DoR)*? *(Pergunta fechada resposta única obrigatória)*	☐ Sempre ☐ Às vezes ☐ Só conforme a necessidade ☐ Nunca
Seção 3 - ENVOLVIMENTO DO CLIENTE		
10	Qual a sua opinião em incluir o cliente em sessões de DoR nas empresas participando na revisão dos requisitos do projeto? *(Pergunta fechada resposta única obrigatória)*	☐ Extremamente favorável ☐ Favorável ☐ Indiferente ☐ Desfavorável ☐ Extremamente desfavorável
11	Selecione os cenários que você acredita serem favoráveis para o cliente realizar uma revisão dos requisitos, de forma a enriquecer DoR nas empresas. *(Pergunta fechada múltipla escolha obrigatória)*	☐ Apenas na revisão das Histórias de Usuário ☐ Em Requisitos Complexos ☐ Em Requisitos Reincidentes ☐ Em Requisitos Não Especificados ☐ Em Sistemas Críticos ☐ Outros
12	Em situações favoráveis, qual papel o cliente poderia desempenhar na revisão de histórias de usuário? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Validar se os objetivos de negócio foram contemplados ☐ Propor ajustes no escopo para tornar o requisito mais claro ☐ Confirmar se os critérios de aceitação atendem às expectativas ☐ Garantir que a priorização reflete suas necessidades reais ☐ Identificar lacunas que o time técnico não percebeu ☐ Sugerir melhorias na experiência do usuário ☐ Fazer questionamentos ou preocupações sobre a viabilidade de negócio ☐ Outros
13	Você acredita que sua equipe atual estaria preparada para revisar os requisitos com o cliente, principalmente em sessões de DoR? Por quê? *(Pergunta fechada resposta única obrigatória)*	☐ Sim, porque o processo de validação é estruturado ☐ Sim, porque são abertos à novas práticas e experimentações ☐ Talvez, porque depende de mais alinhamento e capacitação ☐ Não, porque falta maturidade no entendimento sobre requisitos ☐ Não, porque a colaboração com clientes é limitada atualmente ☐ Não se aplica para realidade da empresa ☐ Outros

Seção 4 - CENÁRIOS DE REVISÃO DE REQUISITOS		
14	Quais mudanças você considera necessárias para promover colaboração ativa e favorável dos clientes na validação de requisitos em empresas? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Alinhamento com metas estratégicas do projeto ☐ Workshops de integração para envolver os gestores no processo ☐ Ajustes de cronogramas e agendas ágeis ☐ Formação de facilitadores para mediar a comunicação ☐ Simplificação de requisitos para linguagem simples e materiais visuais ☐ Guias ou manuais explicativos para clientes ☐ Ferramentas colaborativas para documentar requisitos (Miro, Figma, Trello, etc) ☐ Comunicação síncrona com o cliente na revisão dos requisitos ☐ Comunicação assíncrona com o cliente na revisão dos requisitos ☐ Outros
15	Como você acredita que os gestores poderiam ser incentivados a apoiar essa mudança nas empresas? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Através de resultados positivos em experimentos ou projetos-piloto ☐ Através de evidências de redução de retrabalho e custos ☐ Através de nível de confiança do time com o desenvolvimento do requisito ☐ Através de aumento da satisfação dos clientes com as entregas ☐ Outros
16	Quais ferramentas você considera mais adequadas para facilitar e otimizar a revisão de requisitos pelos clientes? *(Pergunta fechada múltipla escolha obrigatória)*	☐ Cartões Ordenados: cards utilizados para representar os passos-a-passo dos requisitos, permitindo que o cliente os priorize de acordo com sua ordem de importância. ☐ Mapas Mentais: ferramenta visual em formato de diagrama usado para explorar as ideias do cliente em torno do requisito. ☐ Técnica This/That: exibição de cenários binários de escolha, como por exemplo, "Você prefere receber uma notificação semanal ou apenas antes do pagamento?" ☐ Mapa de Empatia: ferramenta de visualização que ajuda a compreender as necessidades, emoções, desejos e perspectivas do cliente sobre o contexto do requisito. ☐ Cenários de Uso: contação de histórias no formato de narrativas que representam o contexto prático do requisito.

		☐ Story Mapping: mapa de histórias que mostra o objetivo geral do requisito, passos principais e subtarefas. ☐ Cenários BDD: técnica que utiliza cenários de comportamento esperado escritos em linguagem acessível (Given-When-Then). ☐ Checklists Simples: revisão de cada tópico dos requisitos (critérios de aceitação, regras de negócio, etc). ☐ Não tenho uma opinião formada a respeito ☐ Outros
Seção 5 - FEEDBACKS FINAIS		
17	Quais aspectos positivos ou negativos, sugestões, melhorias você gostaria de registrar sobre a proposta de incluir o cliente em sessões de *Definition of Ready* (DoR)? *(Pergunta aberta opcional)*	
18	Adoraríamos saber um pouco mais da sua jornada profissional na validação de produtos digitais. O que você pode compartilhar sobre sua relação em validar Requisitos de Softwares (realizações, conquistas, dificuldades, obstáculos, etc)? *(Pergunta aberta opcional)*	
19	Você autoriza entrarmos em contato com você por email para um eventual aprofundamento sobre o tema? *(Pergunta fechada dicotômica opcional)*	☐ Sim ☐ Não
20	Se sim, informe seu email. *(Pergunta aberta opcional)*	

Fonte: Elaborado pela autora.

4.2 Resultados do *Survey*

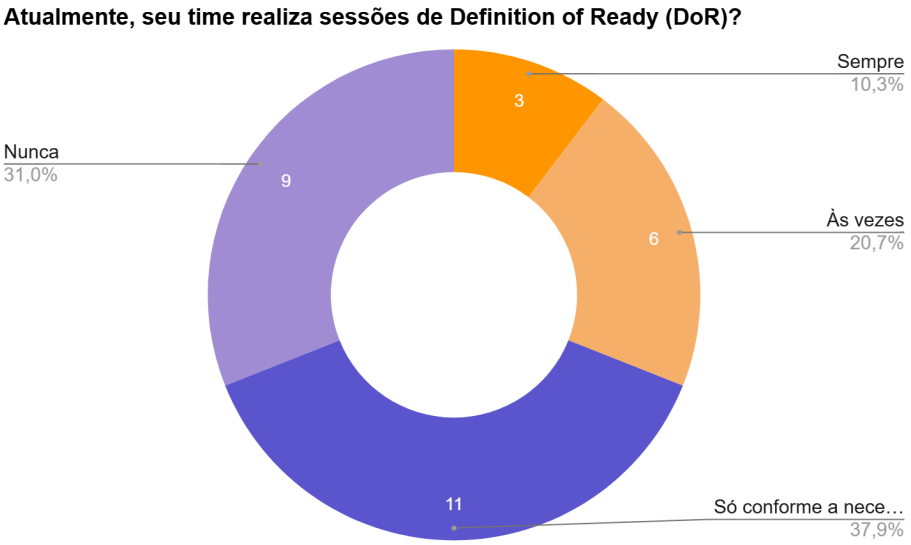
Os dados coletados foram organizados e tratados por meio de planilha eletrônica, onde foram realizadas análises descritivas para identificar padrões de respostas e recorrência de percepções. Primeiramente, na Seção 1 do questionário, cerca de 56,7% dos participantes atuam há mais de 6 anos em sua respectiva função profissional. Esse dado se tornou muito relevante para o estudo, pois indica que a pesquisa contou com uma amostra considerável de participantes com experiência consolidada na área. Por conta disso, tal característica confere maior consistência e credibilidade às percepções e opiniões coletadas, enriquecendo a análise sobre a temática envolvida no estudo.

Ainda na Seção 1 do *survey*, foi aplicada uma pergunta-filtro para identificar os participantes que têm experiência com validação de requisitos. Dentre os 31

participantes, a ampla maioria, representada por 29 indivíduos, declarou possuir experiência na atividade. Esse resultado reforça a adequação da amostra para os objetivos da pesquisa, conferindo respostas provenientes de indivíduos com vivência direta ao tema investigado.

Na Seção 2, cerca de 58,60% dos participantes apontaram que seus times realizam sessões de DoR esporadicamente, representados pelos 20,70% que responderam às vezes, e 37,90% que responderam apenas conforme a necessidade, conforme ilustrado a Figura 3. Por outro lado, apenas 10,30% dos participantes afirmaram que essas sessões são realizadas sempre, enquanto que 31% responderam que nunca acontecem. Esse dado evidencia que, embora a prática de DoR esteja presente em algumas equipes, ela ainda não é uma rotina estruturada ou constante nas *squads*.

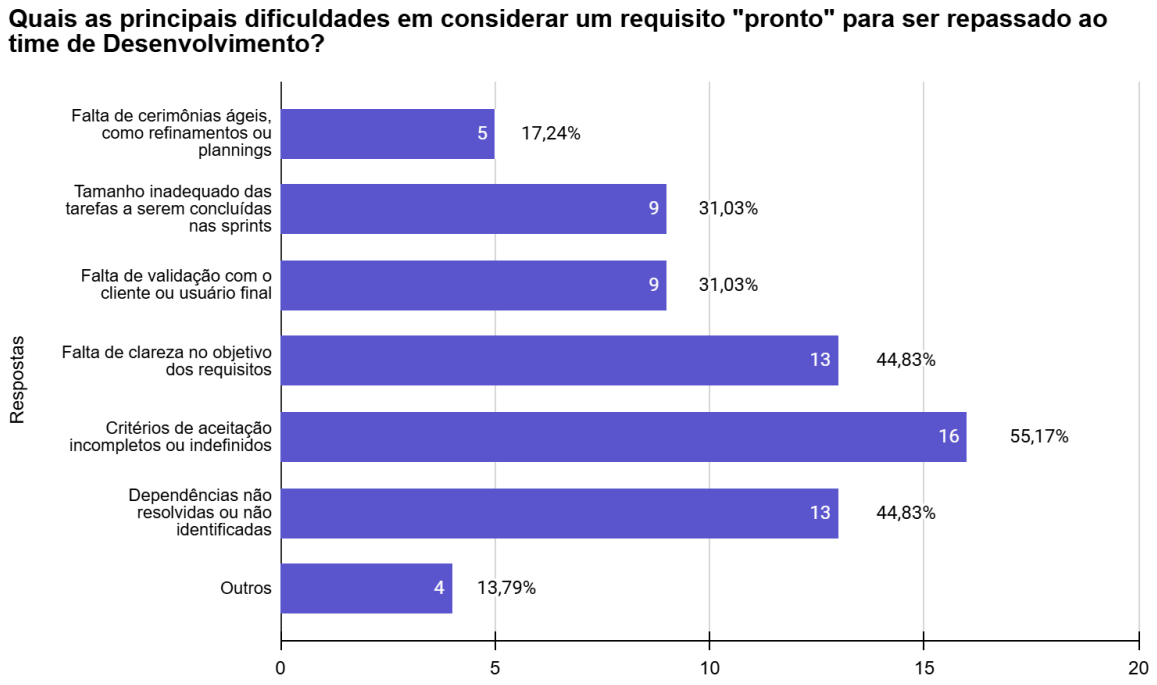
Figura 3: Gráfico com verificação se sessões de DoR são realizadas atualmente nas empresas dos participantes.



Fonte: Dados coletados pela autora via *survey*.

Os dados revelaram ainda que essa maioria identifica lacunas no processo de validação atual em seus times, como ausência de ACs completos ou definidos (55,17%), falta de refinamentos sobre os requisitos (44,83%), além de dependências e alinhamentos de expectativa não resolvidos (44,83%), conforme o gráfico apresentado na Figura 4.

Figura 4: Gráfico das principais dificuldades em considerar um requisito "pronto" para ser repassado ao time de Desenvolvimento.

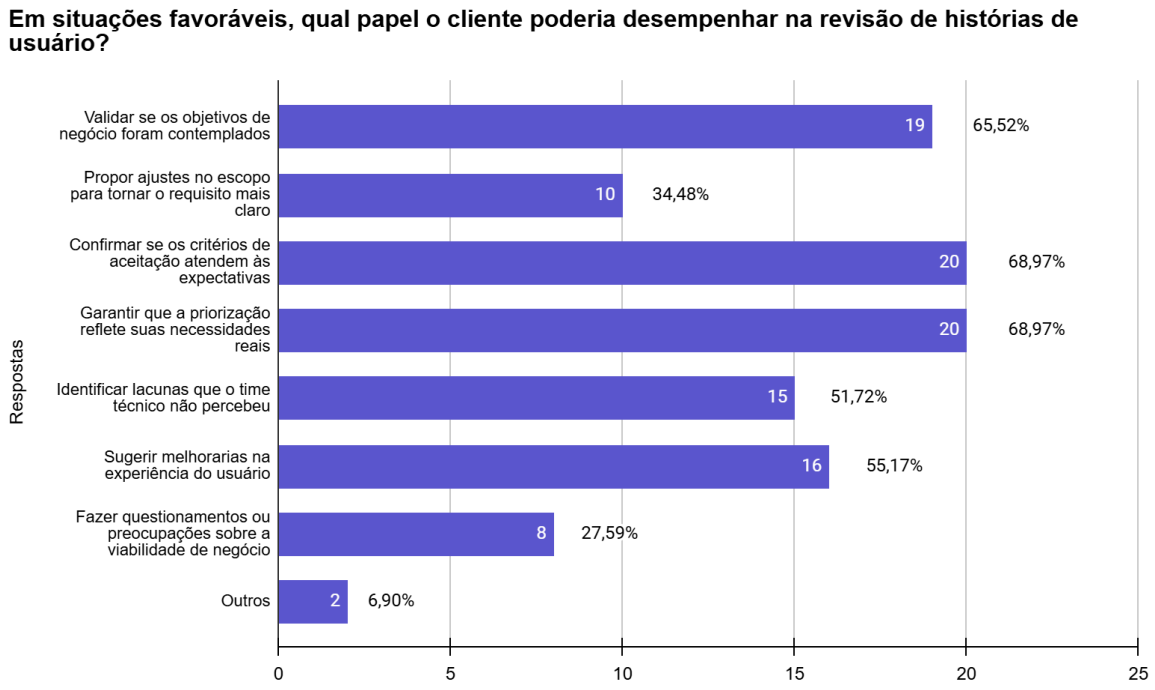


Fonte: Dados coletados pela autora via *survey*.

Por essa amostra, podemos ver uma variação percentual significativa no nível de maturidade e adoção dessa prática nos times de tecnologia, apontando que, mesmo em organizações onde há contato direto com clientes e validação de requisitos, a formalização de DoR nem sempre é consolidada como uma etapa obrigatória do processo de desenvolvimento ágil. Isso reforça a importância de discutir o assunto envolvido para estudar formas de incluir o cliente de forma participativa nesse tipo de sessão ou levantar outras alternativas que evidenciem concretamente às necessidades dos clientes nos requisitos, considerando o espaço de oportunidade para ampliar a aplicação e o valor estratégico do DoR nas equipes.

Essa perspectiva foi atestada na Seção 3 do *survey* quando perguntado sobre como o cliente poderia desenvolver a revisão de história de usuário. Cerca de 20 participantes responderam que o cliente poderia atuar nos papéis de “Confirmar se os critérios de aceitação atendem às expectativas” e “Validar se os objetivos de negócio foram contemplados”, ambos com 68,97% das respostas, conforme gráfico apresentado na Figura 5. A preocupação seguinte de maior relevância foi “Validar se os objetivos de negócio foram contemplados” com cerca de 19 respostas a favor (65,52%).

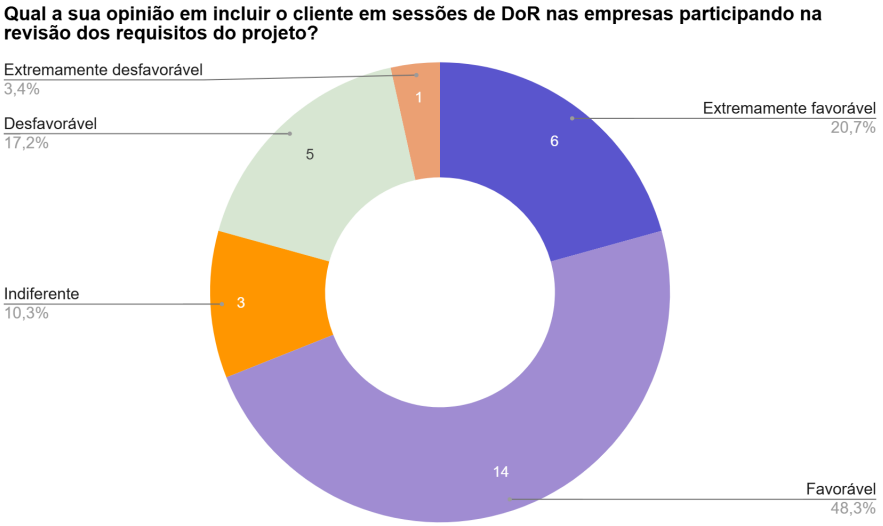
Figura 5: Gráfico com papéis que o cliente poderia desempenhar na revisão de histórias de usuário.



Fonte: Dados coletados pela autora via *survey*.

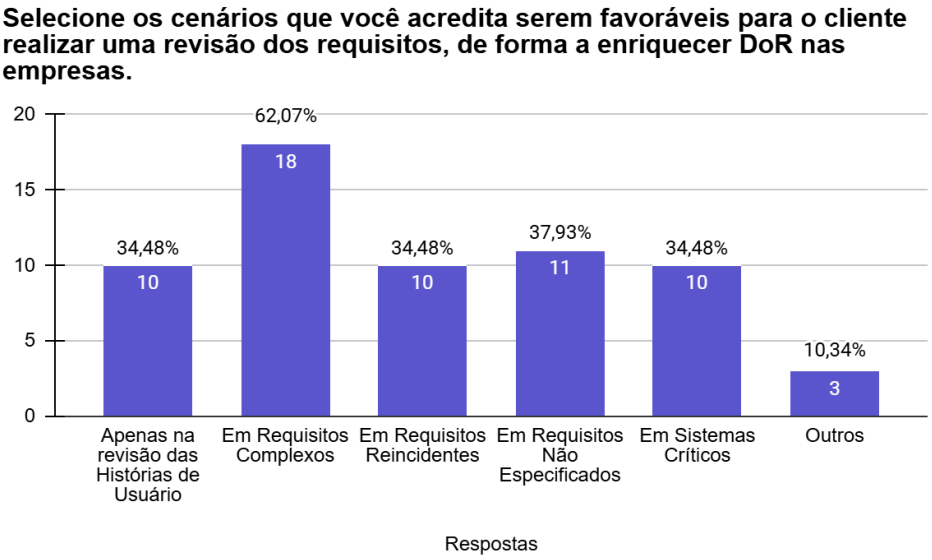
De acordo com as Figuras 6 e 7, os resultados também apontaram que a inclusão do cliente nas sessões de DoR é vista, em sua maioria, como favorável ou extremamente favorável, principalmente para validação de requisitos complexos ou que não possuem especificações claramente definidas.

Figura 6: Gráfico que exibe a opinião sobre incluir o cliente em sessões de DoR nas empresas.



Fonte: Dados coletados pela autora via *survey*.

Figura 7: Gráfico de cenários favoráveis da participação do cliente na revisão de requisitos.

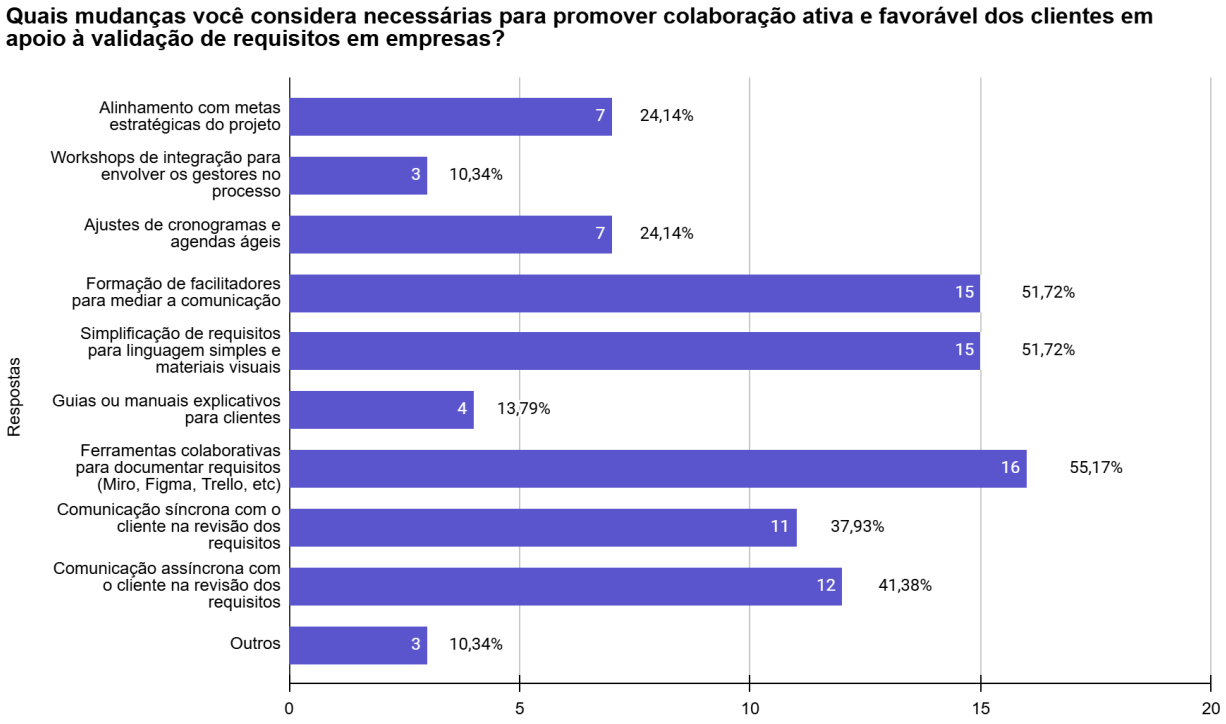


Fonte: Dados coletados pela autora via *survey*.

Analisando esses dados é possível considerar que os participantes enxergam o cliente como colaborador ativo na identificação de oportunidades de melhoria e na detecção de pontos críticos que poderiam passar despercebidos pelo time técnico, contribuindo para a qualidade e a adequação do produto. Essa percepção não só expõe a opinião dos profissionais de tecnologia sobre a importância de envolver o cliente de maneira estratégica nessas etapas, mas também, considerar os aspectos funcionais, de UX e de viabilidade de negócio sobre a validação dos requisitos.

Seguindo a visão dos principais resultados da Figura 7, o gráfico ilustrado na Figura 8 exibe possíveis mudanças que podem conectar o cliente, como colaborador ativo, à validação de requisitos de forma prática. Nota-se que cerca de 55,17% dos participantes declararam a perspectiva de considerar a utilização de ferramentas colaborativas nesse cenário.

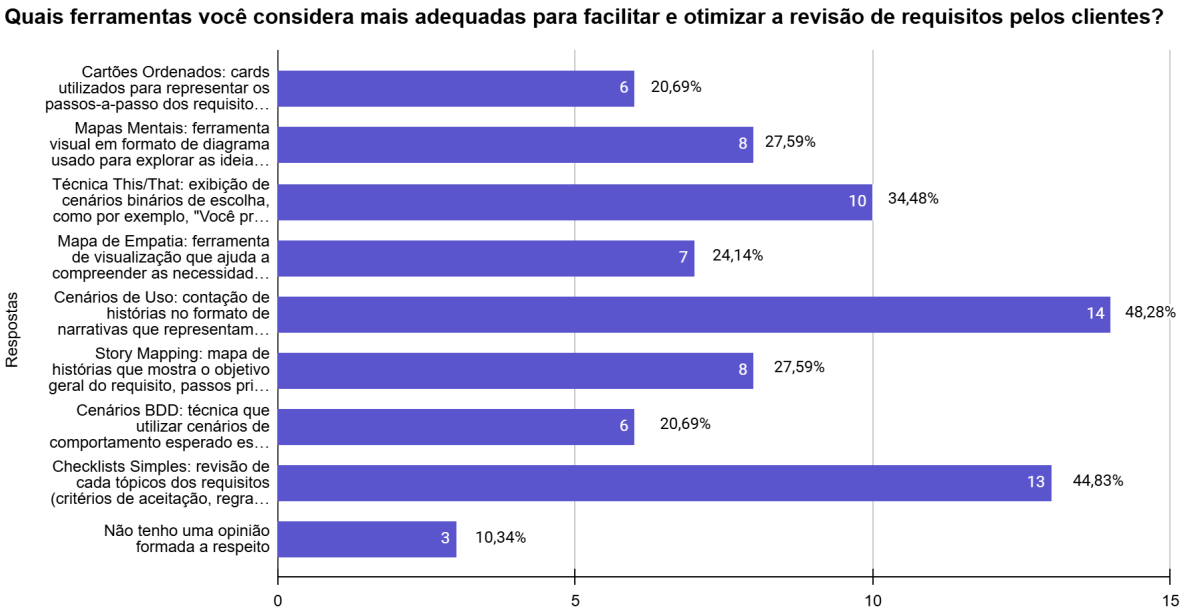
Figura 8: Gráfico de mudanças consideráveis necessárias para colaboração dos clientes na validação de requisitos.



Fonte: Dados coletados pela autora via *survey*.

Entre as ferramentas apontadas como mais adequadas para facilitar e otimizar a revisão de requisitos pelos clientes, os participantes apontaram *storytelling* (48,28%) para contextualizar requisitos de forma acessível, e *checklist* simples (44,83%) para revisão de USs e seus componentes, o que determina a valorização pela objetividade, conforme apresentado na Figura 9.

Figura 9: Gráfico das ferramentas adequadas para otimizar revisão de requisitos.



Fonte: Dados coletados pela autora via *survey*.

Não menos relevantes, cartões ordenados foram escolhidos pelos participantes para representar o passo a passo dos requisitos (fluxo da tarefa), assim como cenários BDD, sendo ambos com 20,69% das respostas. Esses resultados refletem uma preferência por artefatos que conciliem clareza, objetividade e contextualização prática na comunicação de requisitos com os cliente, reforçando a importância de tornar os requisitos mais acessíveis e auditáveis para públicos não técnicos, especialmente em ambientes ágeis (Lucassen et al., 2015). O destaque para os cenários de uso e *checklists* simples faz refletir a atenção para recursos que possam traduzir informações técnicas em conceitos compreensíveis. Com isso, vê-se a importância de estruturar abordagens que auxiliem na tradução de aspectos técnicos não apenas em elementos verificáveis, mas também em atributos explícitos de UX, , de modo a garantir que os requisitos reflitam tanto a precisão funcional quanto a visão do cliente.

Pelos comentários apresentados no *survey*, um participante Desenvolvedor com mais de 6 anos de experiência na função afirmou que vê *"[...] muitos pontos positivos com a participação do usuário/cliente na elucidação dos requisitos, no final das contas ele é quem irá utilizar o produto. Ninguém melhor do que ele para saber a real necessidade. Isso traria juntos uma maior assertividade e energia focada no que realmente vai agregar, evitando problemas e necessidades de ajustes após a*

entrega. Um contra talvez possa ser a demora, por falta de experiência do usuário/cliente em participar desse tipo de atividade, mas com tempo ou treinando-o antes, isso poderia ser mitigado.". Essa reflexão foi compartilhada por grande parte dos participantes, embora alguns tenham revelado entraves sobre suas realidades e contextos de ambiente trabalho: (1) *"na prática, muitos clientes não querem (ou não conseguem) participar ativamente dessas sessões, seja por falta de tempo, interesse ou entendimento técnico. Fazendo com que seja um esforço extra envolver o cliente nesse momento de gestão. Uma possibilidade é incluir o cliente em sessões de DoR apenas para épicos, entregas mais críticas ou finalização de entregas; onde o alinhamento é essencial."*; (2) *"No projeto que estou atualmente, em especial, a Gerente de Portfólio prefere fazer reuniões e alinhamento sem a participação do cliente. Ela prefere tudo sendo centralizado nela. Por isso, é difícil incluir o cliente de forma ativa tendo uma gestora com esse perfil."*; (3) *"Nós validamos requisitos meio que no método go-horse, geralmente o refinamento nunca trás o nível de clareza necessário, tendo que ficar revisando com o PO ou responsável pelo protótipo para verificar se estamos no caminho correto."*. Esses relatos confirmam que a efetividade da prática depende não apenas da abertura dos clientes, mas também da maturidade organizacional e do alinhamento entre papéis (Cohn, 2004; Pichler, 2010).

Um cenário apresentado por um outro participante também com mais de 6 anos de experiência, sendo esse, Analista de Requisitos, foi sobre a existência de situações que não podem ser ignoradas no contexto sobre DoR (*"Já vivenciei situações em que o próprio cliente validou uma tarefa como "pronta" durante o desenvolvimento, mas no final do projeto não ficou satisfeito e pediu alterações. Isso mostra que a definição de "pronto" do cliente não é necessariamente uniforme — ela pode mudar com o tempo ou com uma visão mais completa do produto."*). Esse dado mostra que não devemos projetar todas as expectativas no cliente, mas sim nas informações compartilhadas e validadas colaborativamente. Ou seja, reforça o princípio de que a validação não deve ser centrada apenas na aprovação formal do cliente, mas deve envolver também práticas sistemáticas que considerem comportamentos de uso e dores reais (Cohn, 2004; Pichler, 2010).

Um participante Desenvolvedor, declarado com 17 anos de experiência na função, afirmou crer que *"a falta de conhecimento de processo do cliente e a falta de*

capacidades comunicacionais dos times técnicos são o maior entrave para esse processo. Processos que consigam fazer esse intermédio são chave. [...] entreguei os mais diversos produtos com os mais diversos fluxos de trabalho. Esse gap entre o cliente e o time de desenvolvimento é, até hoje, o maior desafio para mim.". Esse depoimento evidencia que a ausência de processos claros de mediação entre clientes e equipes técnicas permanece como um obstáculo recorrente, reforçando a necessidade de práticas estruturadas que favoreçam a comunicação e a validação efetiva de requisitos.

Uma contribuição significativa compartilhada por um participante Designer com mais de 6 anos de experiência foi em relação direta com a revisão de requisitos (*"Sou a favor da participação do cliente nesse processo. Já experimentei e foi excelente, story mapping sempre ajudou muito em projetos mais iniciais e checklist de acceptance criteria em projetos mais maduros com uma jornada mais definida e sólida de usuário."*). Esse comentário se tornou um vetor de ideias para apoiar a validação de ACs com a intenção de incorporar aspectos de UX.

Tal constatação, motivou a autora desta dissertação a explorar a possibilidade de automatizar a estrutura de um *checklist* de ACs com apoio de LLMs como recurso facilitador da verificação, inicialmente, da qualidade da escrita de ACs em benefício à revisão de requisitos. Ainda assim, faltava encontrar um meio de determinar aspectos de UX à base de conhecimento dos LLMs. Com isso, as *guidelines* ACUX (Souza, 2021) foram escolhidas para compor essa função dado sua estrutura e características de *checklist* que correspondiam harmoniosamente a tal propósito.

De maneira geral, os resultados do *survey* indicam que fatores como ambiente de trabalho, escopo de projeto e o nível de diálogo estabelecido sobre as necessidades do cliente influenciam diretamente nas condições de prontidão dos requisitos de *software*. Diante dessas demandas, especialmente no que se refere aos desafios de comunicação ligados à experiência do usuário, a abordagem proposta nesta dissertação se apresenta aos profissionais de tecnologia como uma alternativa acessível com viés colaborativo voltado para auxiliar na validação de requisitos, principalmente, os ACs de USs, promovendo um alinhamento mais consistente com a perspectiva do usuário final.

4.3 Considerações Finais

O *survey* sobre DoR buscou explorar e compreender a percepção de profissionais da área de Tecnologia quanto às práticas de validação de requisitos em contextos ágeis, enfatizando a influência da participação do cliente e o potencial de integração com diretrizes de UX. É importante considerar que sua amostragem não permite generalizações, uma vez que esteve condicionada a restrições de prazo e de alcance na divulgação.

De modo geral, os resultados formataram a hipótese inicial de que a formalização da DoR ainda se encontra em estágios heterogêneos de maturidade entre as equipes, com lacunas significativas relacionadas, por exemplo, à ausência de ACs mais completos, dificuldades de refinamento e alinhamento insuficiente entre *stakeholders*. Essa constatação reforça o entendimento de autores como Lucassen et al. (2015) e Leffingwell et al. (2016), que defendem a importância de mecanismos estruturados de validação para assegurar a qualidade e a rastreabilidade dos requisitos antes do desenvolvimento.

As respostas dos 31 participantes evidenciaram um consenso quanto à relevância da inclusão do cliente nas etapas de DoR, reconhecendo esse ator como um agente capaz de enriquecer a revisão de requisitos, detectar ambiguidades e alinhar as entregas às expectativas reais de uso e valor de negócio. Contudo, também emergiram desafios práticos, como restrições de tempo, falta de preparo técnico do cliente e resistência organizacional, que limitam a viabilidade dessa participação, sobretudo em empresas com processos pouco maduros ou excessivamente hierarquizados. Esses achados dialogam com estudos de Cohn (2004) e Pichler (2010), que já alertavam para a necessidade de papéis bem definidos e comunicação mediada para a efetividade da colaboração cliente-equipe.

Outro ponto relevante identificado refere-se às ferramentas e artefatos preferidos para promover colaboração na revisão de requisitos. Os participantes destacaram o uso de *storytelling*, *checklists* simples e cenários BDD como recursos práticos e compreensíveis para traduzir aspectos técnicos em linguagem acessível. Essa preferência sinaliza um caminho promissor para aproximar o cliente dos processos de validação, além de justificar a pertinência da adoção das *guidelines*

ACUX (Souza, 2021) neste estudo, uma vez que estas oferecem uma estrutura análoga à de um *checklist* voltado à avaliação de atributos de UX em ACs.

De forma interpretativa, os resultados do *survey* revelam que o nível de diálogo estabelecido entre o cliente e o time técnico é um dos fatores diretamente proporcionais à qualidade da validação de requisitos, e que a ausência de mediação comunicacional continua sendo uma das maiores barreiras à efetividade da DoR. Essa constatação fundamenta a relevância de investigar soluções automatizadas de apoio à validação, como propõe este trabalho ao empregar LLMs combinados à técnica ACUX para atuar como tutores digitais na análise de ACs.

Em síntese, o capítulo contribuiu para consolidar o diagnóstico contextual que sustenta as etapas seguintes da pesquisa. Ao mapear percepções, práticas e limitações dos profissionais, o *survey* não somente forneceu evidências empíricas que orientam o delineamento das próximas etapas do estudo, mas também reforçou a necessidade de abordagens que promovam maior integração entre DoR, UX e automação inteligente para beneficiar a validação de requisitos. Portanto, as discussões apresentadas não apenas respondem às questões norteadoras iniciais, mas também criam o vínculo conceitual que conduz à próxima etapa da pesquisa, a análise de conteúdo de ACs pela avaliação humana, responsável por aprofundar a compreensão prática sobre a técnica ACUX em artefatos reais de *backlog*.

5 Análise de Conteúdo de ACs pela Avaliação Humana

A fim de conferir a aplicabilidade da técnica ACUX, foi realizada uma análise de conteúdo (Bardin, 2016) em ACs escritos em LN e selecionados de projetos de alunos de graduação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE) da disciplina de ER, pertencentes a dois períodos letivos consecutivos, 2023 e 2024. Essa análise foi executada por um avaliador humano com experiência de 10 anos em ER e UX no mercado de tecnologia, e conhecedor da didática aplicada aos alunos por ser graduado no mesmo centro universitário da instituição acadêmica. A amostra deste estudo foi composta por artefatos de *backlog* representados por ACs de USs elaboradas por tais estudantes conforme as definições de Mike Cohn (2004). A disciplina acadêmica norteou critérios para a documentação dos requisitos, incluindo a obrigatoriedade de criar protótipos de tela

que pudessem retratar, minimamente, a usabilidade das funcionalidades, com a intenção de reduzir os riscos de se registrar ideias vagas, ambíguas, disfuncionais e não testáveis sobre os produtos.

Nesse sentido, o escopo das USs foi reduzido e baseado em orientações que instruíam listar requisitos que configurassem valor real ao produto idealizado. É importante informar que, apesar dos protótipos de tela criados, os alunos não foram orientados à conceitos de UX, nem mesmo conheciam ou foram apresentados à abordagem ACUX durante o período letivo de toda a disciplina. Logo, já era conhecido pelo avaliador humano que os ACs selecionados provavelmente poderiam não conter trechos que tivessem relação explícita com a visão do usuário. Os alunos foram encorajados a construir funcionalidades que refletissem necessidades reais (síntese) e dores do mercado (concepção). Ademais, foi reforçado que o documento de requisitos serve como um acordo formal sobre o que será construído. Fazendo com que, ao final do processo, a equipe e o cliente saiba exatamente o que deve ser entregue e testado (concepção validada), bem como permite a rastreabilidade, ou seja, saber a origem de cada funcionalidade e como ela se relaciona com a necessidade original.

Outro critério estabelecido pela disciplina de ER era que os ACs fossem escritos no formato BDD, condição aderente à proposta apresentada neste presente trabalho. A escolha de selecionar ACs escritos nesse formato foi determinada com o intuito de assegurar diversidade dos cenários de uso e evitar vieses informais de seleção. A partir disso, quatro projetos de contextos distintos foram escolhidos, sendo dois publicados pelos alunos de semestres de 2023 e dois publicados por alunos de semestres de 2024. O procedimento de verificação das USs desses projetos foi iniciado com uma revisão individual de cada documento de requisito, a qual foi estabelecida uma triagem apresentada na Tabela 4 que permitiu a seleção de cinco USs de cada projeto dos alunos, totalizando 20 USs e, consequentemente, 20 ACs analisados[3] que foram utilizados como artefatos de avaliação deste estudo empírico.

[3] Documentos de Requisitos com USs/Acs das Turmas Selecionadas de 2023 e 2024: https://drive.google.com/drive/folders/1ukBOSdAEwqgUzziROfPdExUehMleWG_m?usp=drive_link

Logo após, deu-se início a análise de conteúdo realizada pelo avaliador humano em tais ACs selecionados utilizando a técnica ACUX (Souza, 2021) com o mesmo método de avaliação usado pelo autor da abordagem. O Subcapítulo 2.4 apresenta a fundamentação teórica sobre como essa avaliação da escrita dos ACs foi realizada pelo autor. Em seu estudo, Souza (2021) utilizou USs com ACs escritos em formato BDD, elaboradas por startups, e verificou a aplicabilidade da ACUX, registrando em planilha todos os trechos que necessitavam de especificação de aspectos de UX. A estrutura dessa planilha foi igualmente replicada nesta presente análise de conteúdo qualitativa sugerida por Bardin (2016). Essa abordagem é indicada para estudos de natureza exploratória e interpretativa, cujo objetivo consiste em categorizar, descrever e interpretar conteúdos textuais de forma organizada e criteriosa, extraindo sentidos latentes e manifestos presentes nas mensagens analisadas.

Segundo Bardin (2016), a análise de conteúdo permite elaborar inferências acerca do conteúdo das informações, das intenções e dos efeitos sociais ou comunicativos nelas contidos. A autora define três etapas para condução da análise de conteúdo. Estas etapas serviram como base para definição da metodologia desta análise, de modo que cada atividade realizada na análise se relaciona com as etapas sugeridas por Bardin (2016), conforme pode ser observado na Figura 10. Além disso, serviram também como referência para desenvolvimento do *prompt* instrucional proposto nesta dissertação como apresentado no Capítulo 6.

Figura 10: Metodologia para avaliação dos ACs.

Fonte: Elaborado pela autora.

Inicialmente na etapa de **preparação** (Subcapítulo 5.1) foram definidos todos detalhes que envolvem cada uma das etapas seguintes, bem como o conhecimento sobre ACUX necessário para condução da avaliação. Em seguida, foi conduzida a etapa de **análise** (Subcapítulo 5.2) na qual foi realizada a codificação das *guidelines* ACUX nos ACs. E por fim, os **resultados** (Subcapítulo 5.3) obtidos na etapa de análise que demonstram o uso do ACUX.

Os ACs foram lidos, categorizados e codificados manualmente, à luz das categorias definidas pela abordagem ACUX (Souza, 2021), permitindo identificar padrões, lacunas e oportunidades de melhoria. Este procedimento metodológico não apenas fornece uma descrição detalhada do estado atual da elaboração dos critérios, como também viabiliza, em etapa posterior, a realização de uma comparação estruturada entre os critérios identificados manualmente e aqueles

identificados pelos LLMs treinados com *prompts* orientado ao ACUX. A coleta de dados foi inteiramente documental, sem aplicação de questionários ou entrevistas. Os documentos analisados foram anonimizados para manter a confidencialidade das equipes e dos indivíduos envolvidos, assim como, a integridade ética da investigação.

Como ferramenta de apoio para registro das informações coletadas e codificação dos ACs, a ferramenta Google Planilhas foi utilizada para tabulação[4] dos dados e gerenciamento das *guidelines* de análise. A escolha por tal método de análise se justifica pela sua adequação ao tipo de dado analisado, textos curtos e descritivos estruturados em LN, e pela capacidade de revelar aspectos qualitativos frequentemente negligenciados em análises puramente quantitativas. Além disso, permite, em especial, uma avaliação crítica e contextualizada da qualidade dos ACs sob a perspectiva da UX.

5.1 Preparação

Sendo a primeira etapa do processo de avaliação dos ACs, ocorreu a preparação e organização do material a ser analisado. Desta etapa em diante, todas as atividades foram executadas pelo avaliador humano citado no Capítulo 5. Essa etapa segue as recomendações de Bardin (2016) que indica delimitar o *corpus* de análise, formular as hipóteses de pesquisa e os objetivos da análise, como também realizar uma leitura exploratória dos dados disponíveis. A partir disso, o pesquisador realiza uma leitura flutuante dos dados, escolha dos documentos e definição dos critérios de inclusão e exclusão, formalizados em um plano de análise. Esse plano define a execução das etapas subsequentes e sistematiza o processo de pesquisa ratificando a aderência aos propósitos investigativos no estudo. A preparação iniciou por meio de uma revisão aprofundada dos conceitos que fundamentam as diretrizes do ACUX para fortalecer a compreensão de cada *guideline* e suas aplicações. Esse processo de revisão foi conduzido a partir do material original disponibilizado por Souza (2021), apresentado como uma cartilha online[5] que funciona como um

[4] Planilhas de Avaliação ACUX dos ACs analisados pelo Especialista (Projetos de 2023 e 2024): https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?usp=sharing

[5] ACUX *Guidelines* (Cartilha Online): https://www.figma.com/proto/EPjh6WCoZsrLHAE0Y17omn/ACUX-Guidelines?scaling=min-zoom&page-id=0%3A1&node-id=370-0

checklist operacional para avaliação dos ACs sob a óptica de UX. Além da cartilha, o autor Jonathan Souza (2021) compartilhou em seu estudo uma planilha pública com exemplos práticos de ACs validados pelas diretrizes ACUX. Esses exemplos serviram como referência didática para ACs avaliados por esta dissertação, possibilitando compreender, de forma aplicada, como cada diretriz ACUX é atribuída para configurar uma validação real e de que maneira as recomendações desta validação podem orientar o aprimoramento dos ACs sob a perspectiva da UX.

Em seguida, o plano de análise foi elaborado descrevendo diretivas que ajudaram na validação dos ACs selecionados. A seleção dos ACs foi iniciada a partir dos documentos de requisitos disponibilizados em arquivo *.doc* por cada uma das equipes de graduação envolvidas. A seguir, nos Subcapítulos 5.1.1.1, 5.1.1.2 e 5.1.1.3, são apresentadas as diretivas que ajudaram a nortear a análise, e interpretação dos resultados com critérios de inclusão e exclusão bem definidos, conforme a Tabela 4.

5.1.1 Plano de análise

5.1.1.1 Critérios de Inclusão e Exclusão

Para manter a consistência da análise, foram definidos critérios de inclusão e exclusão dos ACs a serem avaliados. Esses critérios estão listados na Tabela 4.

Tabela 4: Critérios de inclusão e exclusão para análise de conteúdo.

Critérios de Inclusão	**Critérios de Exclusão**
Apenas ACs estruturados no formato BDD (*Given-When-Then*).	ACs vagos, sem estrutura declarada em BDD.
ACs presentes em USs documentadas por equipes de graduação nos anos de 2023 e 2024.	Critérios sem associação clara a uma US ou com conteúdo incompleto.
Critérios vinculados a USs com conteúdo funcional suficiente para compreensão do objetivo da funcionalidade, ou seja, conteúdo que descreva minimamente uma ação específica capaz de ser executada pelo usuário	Critérios com conteúdo duplicado ou idêntico entre os projetos.

Fonte: Elaborado pela autora.

5.1.1.2 Parâmetros de Avaliação e Codificação

Para análise dos ACs, o *framework* ACUX (Souza, 2021) foi estruturado em dois grupos de diretrizes: Design da Interação e Organização da Informação, e Elementos Visuais. Cada AC foi analisado individualmente, buscando identificar a presença de aspectos de UX explícitos ou implícitos no conteúdo do critério, e ausências relevantes de aspectos de UX previstos no ACUX, a partir da leitura contextual do critério. A codificação das ocorrências foi realizada por meio de marcação da cor laranja para aspectos de UX relacionados a Design da Interação e Organização da Informação, e marcação da cor verde para aspectos de UX relacionados a Elementos Visuais. Cada ocorrência foi registrada na planilha por meio do código da *guideline* do ACUX associada e a descrição do aspecto identificado sugerindo sua melhoria.

5.1.1.3 Estratégias e Limitações da Análise

Para sistematizar a análise, foram estabelecidas algumas estratégias. Assim como, apesar do planejamento, algumas limitações metodológicas foram reconhecidas e registradas para transparência, conforme apresentado nas Tabelas 5 e 6.

Tabela 5: Estratégias utilizadas para preparação da análise de conteúdo.

Estratégias utilizadas
Transcrição de todos os ACs para uma planilha de controle, agrupados por projeto e por US.
Codificação das ocorrências de aspectos de UX por cores, conforme o grupo ACUX correspondente.
Registro das ocorrências que identificam trechos classificados como “Inadequado” ou “Incompleto” em colunas específicas da planilha, uma para cada grupo do ACUX.
Classificação de cada aspecto identificado pela *guideline* ACUX pertinente.
Consolidação das incidências em uma matriz de frequência, indicando a quantidade de vezes em que cada *guideline* foi requerida para revisão em cada critério.
Visualização dos resultados por meio de “mapa de calor”, utilizando gradação em tons de vermelho, em que as áreas de maior incidência de recomendações foram destacadas.

Fonte: Elaborada pela autora.

Tabela 6: Limitações deliberadas para preparação da análise de conteúdo.

Limitações
A análise foi realizada por um avaliador humano, o que pode introduzir um viés interpretativo, ainda que mitigado pelo uso objetivo das diretrizes do ACUX.
O volume da amostra pode limitar conclusões amplas sobre a análise, com validade estatística para diferentes populações ou projetos.
O ACUX, apesar de estruturado a partir do *framework* de Garrett (2011), foi validado predominantemente pelo meio acadêmico, o que reduz a aplicação plena de seus resultados em projetos comerciais de grande escala.

Fonte: Elaborada pela autora.

Esse plano de análise organiza a execução da análise de conteúdo, estabelecendo critérios claros e estratégias para garantir credibilidade aos resultados. De certa forma, as limitações listadas podem comprometer a investigação, mas indicam oportunidades para aprimoramento metodológico em estudos futuros, especialmente com a ampliação da equipe de análise e validação interavaliadores.

5.2 Análise

Definidas as regras de preparação, a fase de análise foi realizada em alinhamento à etapa de codificação e categorização do conteúdo como indicado por Bardin (2016), visando registrar e organizar os dados para interpretação posterior. Nesse estágio, se realizou a leitura minuciosa de cada AC, verificando a presença ou ausência de aspectos de UX em suas descrições. Vale ressaltar que, pela formatação autoexplicativa das USs no padrão ágil e pela estrutura padronizada do BDD, não houve necessidade de um conhecimento detalhado sobre o contexto de cada projeto.

Para operacionalizar a codificação, foi desenvolvida uma planilha de controle[6] na qual todas as USs e seus ACs foram transcritos. As ocorrências de aspectos de UX foram destacadas pela cor laranja para identificar as diretrizes do grupo Design da Interação e Organização da Informação, e pela cor verde para o grupo

[6] Planilha dos ACs Analisados: https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?usp=sharing

Elementos Visuais. Com isso, quando identificado um trecho que corresponda a algum aspecto de UX e avaliado se a sua completude descritiva foi negligenciada, a *guideline* correspondente do ACUX era registrada, indicando que o determinado elemento textual necessitava ser especificado. Esses apontamentos foram representados pelos códigos das *guidelines* do ACUX. Todos os ACs foram avaliados individualmente, e as palavras, termos ou expressões relacionadas a atributos de UX foram associadas aos seus respectivos grupos e *guidelines*. Essa abordagem permitiu não apenas identificar lacunas ou insuficiências na formulação dos ACs, mas também apontar recomendações específicas de melhoria para cada um deles, tendo como base o guia ACUX.

5.3 Resultados

A terceira e última etapa, envolveu a interpretação dos dados categorizados na fase anterior. Conforme Bardin (2016), essa fase consiste na inferência e interpretação dos conteúdos analisados à luz dos objetivos da pesquisa. Seguindo essa finalidade, os dados extraídos foram consolidados em uma segunda planilha-matriz[7]. Cada *guideline* do ACUX foi disposta em linhas e os ACs analisados em colunas, formando uma matriz de frequência que permitiu observar a quantidade de vezes que cada *guideline* foi recomendada para revisão ou inclusão em cada um dos projetos acadêmicos.

Para facilitar a leitura e a interpretação visual sobre essas quantidades, uma escala cromática em tons de vermelho degradê foi definida abstratamente para auxiliar a visualização de maior e menor incidência das *guideline* nos ACs. Essa estratégia permitiu identificar rapidamente padrões e recorrência de ausência ou deficiência dessas diretrizes ao longo dos critérios avaliados. Por essa motivação, consequentemente, transformou a matriz em um espécie de mapa de calor funcional na verificação dos dados.

A consolidação dos dados das matrizes dos projetos de 2023 e 2024 mostrou um total geral de 182 vezes que as *guidelines* foram recomendadas, sendo 118

[7] Planilha-Matriz # Análise do Avaliador Humano: https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?usp=sharing

recomendações do grupo Design da Interação e Organização da Informação, e 64 do grupo Elementos Visuais. Isso indica que os ACs, nos dois anos avaliados, redigidos pelos alunos apresentaram fragilidades mais acentuadas nos aspectos de organização e controle de interação do que propriamente na parte estética ou de interface. Além disso, o volume de recomendações de melhoria da escrita dos ACs por esses aspectos denota atenção para torná-los adequados às boas práticas de UX.

A Figura 11 mostra a matriz dos projetos de 2023 que totalizou 80 recomendações de *guidelines*. As *guidelines* DI-01 (16), DI-02 (14) e DI-05 (11) foram as mais recorrentes, evidenciando pouco detalhamento sobre a interação nos ACs avaliados. A frequência da *guideline* EV-02, pertencente ao grupo de elementos visuais, chama atenção e se faz interpretar que aspectos ligados à clareza visual, hierarquia e legibilidade são frequentemente negligenciados pelos projetos. A representação visual da matriz como mapa de calor nesse projeto reforça esses achados, destacando células da planilha com tons laranja e vermelho, ainda que sua predominância seja pelos tons de amarelo, indicando pequenas melhorias em ambos os grupos. Isso se traduz que os estudantes dos projetos de 2023 não incorporaram clareza sobre o comportamento esperado para a funcionalidade (DI-01), instruções claras para orientar o usuário durante a tarefa (DI-05), e definição cenários de exceção e restrições para a funcionalidade (DI-02), além de, representação de status e *feedbacks* visuais para o usuário (EV-02).

Figura 11: Matriz dos ACs dos Projetos de 2023.

2023	Frequência de guidelines a considerar por ACs											
	Projeto 01					Projeto 02						
	US1 CA1	US2 CA1	US3 CA1	US4 CA1	US5 CA1	US1 CA1	US2 CA1	US3 CA1	US4 CA1	US5 CA1	Total	
DI-01	2	1	2	1	2	2	2	1	1	2	16	
DI-02	1	1	1	1	1	3	3	1	1	1	14	
DI-03	1	1	1	0	0	0	0	0	0	0	3	45
DI-04	0	0	0	0	0	0	0	0	0	0	0	
DI-05	1	1	1	1	2	1	1	1	1	1	11	
DI-06	0	0	0	0	1	0	0	0	0	0	1	
EV-01	0	0	0	0	0	0	0	1	1	0	2	
EV-02	2	2	2	3	3	3	3	2	2	3	25	
EV-03	0	0	1	0	1	1	1	1	1	1	7	
EV-04	0	0	0	0	0	0	0	0	0	0	0	
EV-05	0	0	0	0	0	0	0	0	0	0	0	35
EV-06	0	0	1	0	0	0	0	0	0	0	1	
EV-07	0	0	0	0	0	0	0	0	0	0	0	
EV-08	0	0	0	0	0	0	0	0	0	0	0	
EV-09	0	0	0	0	0	0	0	0	0	0	0	

Fonte: Elaborado pela autora.

Figura 12: Matriz dos ACs dos Projetos de 2024.

2024	Frequência de guidelines a considerar por ACs											
	Projeto 01					Projeto 02						
	US1 CA1	US2 CA1	US3 CA1	US4 CA1	US5 CA1	US1 CA1	US2 CA1	US3 CA1	US4 CA1	US5 CA1	Total	
DI-01	4	3	4	1	0	2	2	3	2	2	23	
DI-02	1	1	1	2	2	3	1	1	1	2	15	
DI-03	1	2	2	2	1	1	0	1	1	0	11	73
DI-04	0	1	0	0	0	0	1	1	1	1	5	
DI-05	4	2	6	2	2	0	1	0	1	0	18	
DI-06	0	0	0	0	0	0	0	0	0	1	1	
EV-01	0	1	1	0	0	1	1	1	1	1	7	
EV-02	0	1	1	0	1	0	0	1	0	0	4	
EV-03	1	3	2	1	1	1	3	0	1	3	16	
EV-04	0	0	0	0	0	0	0	0	0	0	0	
EV-05	0	0	0	0	0	0	0	0	0	0	0	29
EV-06	0	0	1	0	0	0	0	0	0	0	1	
EV-07	0	0	0	1	0	0	0	0	0	0	1	
EV-08	0	0	0	0	0	0	0	0	0	0	0	
EV-09	0	0	0	0	0	0	0	0	0	0	0	

Fonte: Elaborado pela autora.

Em projetos de 2024, como pode ser visto na Figura 12, se observa um crescimento expressivo no volume de apontamentos, totalizando 102 recomendações. Novamente, a *guideline* com erros mais predominantes no grupo Design da Interação e Organização da Informação foi DI-01, com 23 ocorrências, se

consolidando como a diretriz mais fragilmente especificada nos ACs analisados. Na sequência, observaram-se as *guidelines* DI-05 (18), e DI-02 (15), evidenciando também recorrentes deficiências na definição de aspectos críticos de interação e comportamento do sistema. A *guideline* EV-03 se posiciona como a *guideline* mais frequentemente recomendada dentro do grupo Elementos Visuais no ciclo de projetos de 2024 com 16 ocorrências de erros. Embora a frequência ainda seja baixa frente às diretrizes de interação, esse crescimento sugere a dificuldade dos alunos em reconhecer aspectos visuais para facilitar a compreensão e navegação do usuário, isso inclui, por exemplo, indicar se as informações serão apresentadas por meio de imagens ilustrativas, gráficos, ícones ou outras representações visuais relevantes. Isso é particularmente crítico porque, segundo Norman (2013) e Hassenzahl et al. (2000), a previsibilidade da interface e o *feedback* imediato são componentes centrais da UX. Foi identificado uma necessidade de especificar EV-06 (1) e EV-07 (1). Essa visão geral se confirma ao visualizar o mapa de calor que destaca tons mais intensos de vermelho para o primeiro grupo indicando criticidade em ajustar os ACs analisados, e uma distribuição maior de recomendações às *guidelines* do segundo grupo, determinando um atenção considerável em níveis diferentes dos aspectos visuais.

A expressiva incidência de DI-01 evidencia uma dificuldade notável dos alunos na formulação de ACs capazes de comunicar o comportamento esperado do sistema sob a perspectiva do usuário. Outro padrão de atenção está para à *guideline* DI-05 que apresentou incidência elevada em todos os projetos. Essa diretriz é uma das mais fundamentais na garantia que o sistema forneça informações claras e consistentes ao usuário durante a interação, sendo apontada na literatura como elemento indispensável para a usabilidade e acessibilidade de sistemas digitais (Norman, 2013; Nielsen, 1994). Sua ausência ou deficiência nos critérios analisados evidencia a dificuldade dos estudantes em traduzir princípios abstratos de UX em requisitos práticos e verificáveis. Já no grupo Elementos Visuais a atenção se destacou para as diretrizes EV-02 e EV-03. Sugerindo aos estudantes uma melhor compreensão sobre a comunicação visual e *feedbacks* de sistema, assim como a necessidade de avanços na modelagem da interface.

Tendo em vista os resultados coletados é possível considerar, portanto, que esse conjunto de avaliações sobre o comportamento dos estudantes replica

realidades bastante comuns em projetos de *software*. Uma dela é de priorizar a funcionalidade bruta e a lógica de negócios, e, de certo modo, atribuir aspectos ligados diretamente ao usuário em outros momentos do desenvolvimento do produto, quando muitas vezes já se tornam mais caros ou inviáveis de corrigir (Norman, 2013; Hassenzahl, 2010; Nielsen, 1994; Pressman & Maxim, 2020; Garrett, 2011). Reflete também que a formação de estudantes e, por consequência, de profissionais em início de carreira, precisa contemplar não apenas o quê será entregue em termos funcionais, mas como as informações técnicas envolvidas serão percebidas e compreendidas na interface. Esse é justamente um dos fundamentos da proposta desta dissertação. A antecipação de preocupações relacionadas à experiência de uso deve ser parte integrante do ciclo de desenvolvimento, e não um adendo posterior (Preece et al, 2015; Norman, 2013). De maneira geral, os resultados permitiram afirmar a hipótese central desta parte do estudo empírico: existe uma significativa desconsideração em declarar atributos de UX nas formulações originais dos ACs, ainda que existam cenários que estejam minimamente suportados por artefatos visuais (wireframes, protótipos).

5.4 Considerações Finais

A análise de conteúdo dos ACs permitiu verificar empiricamente a aplicabilidade da técnica ACUX (Souza, 2021) em ACs escritos por estudantes de ER. Esse exercício qualitativo, orientado pelo método de Bardin (2016), constituiu uma etapa essencial da fase exploratória, ao gerar um diagnóstico detalhado sobre o modo como aspectos de UX podem influenciar positivamente a representação dos ACs.

Foi possível identificar a presença e ausência de atributos de UX nas descrições dos ACs analisados; em seguida, classificar essas ocorrências segundo as diretrizes do ACUX, agrupadas nos eixos de Design da Interação e Organização da Informação e Elementos Visuais; e, por fim, sintetizar as fragilidades e padrões de recorrência por meio da matriz de frequência e do mapa de calor, que visualmente evidenciaram as áreas críticas de cada projeto. Os resultados forneceram não apenas uma leitura descritiva do fenômeno, mas também uma base fundamental para calibrar o comportamento esperado dos LLMs por meio do

desenvolvimento do *prompt* instrucional na fase comparativa dos LLMs, ao qual os mesmo 20 ACs analisados pelo avaliador humano serão avaliados pelos modelos também.

A análise do avaliador humano revelou 182 recomendações de melhoria, com predominância marcante de diretrizes relacionadas ao design da interação, especialmente DI-01, DI-02 e DI-05, que abordam a clareza do comportamento esperado pela interação com a interface, o tratamento de exceções e a orientação durante a tarefa. Tais achados corroboram evidências da literatura (Norman, 2013; Nielsen, 1994; Hassenzahl, 2010; Pressman & Maxim, 2020), indicando que, tanto em contextos educacionais quanto profissionais, os requisitos tendem a priorizar a lógica funcional em detrimento de preocupações com a experiência de uso. Do ponto de vista da organização da informação e da usabilidade, a falta desses aspectos refletem o desafio recorrente de traduzir princípios abstratos de UX em requisitos verificáveis e comunicáveis, um dos problemas centrais que este estudo busca endereçar.

No grupo Elementos Visuais, as diretrizes EV-02 e EV-03 foram identificadas pelo avaliador humano como as mais frequentemente ausentes, sugerindo complementar a descrição de aspectos visuais, hierarquia informacional e *feedbacks* de interface. Vale recordar que os estudantes anexaram protótipo das telas no documento de requisitos. Portanto, foi considerado que as especificações apontadas por essas diretrizes avaliadas na revisão já tenham sido contempladas em tais artefatos visuais. Além de cumprir o propósito analítico, esta etapa também enalteceu a hipótese de que seja possível considerar o ACUX em ambientes educacionais. Essa técnica poderia auxiliar os alunos na formulação de ACs e identificação de deficiências conceituais, consequentemente os preparando para situações problema similares às observadas no mercado.

A sistematização dos achados em categorias e padrões de incidência orientou a estruturação do *prompt* instrucional, baseando-se em evidências concretas de falhas recorrentes e na forma como essas lacunas podem ser reconhecidas por modelos generativos. Os conhecimentos derivados da leitura humana sobre os ACs atuam como *ground truth* (valores de referência, do português) para avaliar o desempenho dos LLMs em precisão técnica, concordância

e explicabilidade (Wagner et al., 2025; Baltes et al., 2025). Além disso, o resultado da codificação das *guidelines* se torna um *gold standard* (referência de ouro, do português) para os modelos, pautando um conjunto de 20 ACs detalhadamente diagnosticados e corrigidos conforme às diretrizes ACUX. Por essa razão, a planilha-matriz de dados se torna não apenas uma conclusão do diagnóstico, mas sim o insumo técnico indispensável para a próxima fase. Especificamente para o *prompt*, outra parte desse conhecimento derivado está incorporado no *instructional* que define o LLM como um tutor especializado em ER e UX, refletindo, assim, a atuação do avaliador humano perante às recorrências de falhas na redação dos ACs que guiarão seu treinamento por meio de *few-shot*, enquanto o processo analítico de passo a passo da avaliação desses artefatos de *backlog* é formado pela estrutura lógica do CoT.

Em síntese, a análise de conteúdo forneceu a evidência empírica para a problemática, validou a ACUX como base de conhecimento de domínio e, mais crucialmente, formatou o conjunto de dados de referência que permitirão a Modelagem do *Prompt* Instrucional, detalhada no Capítulo 6. Assim, o rigor qualitativo do presente capítulo pavimenta o caminho para a instrumentalização da GenIA como uma ferramenta STI especializada na revisão de requisitos orientados à UX.

6 Modelagem do *Prompt* Instrucional

Este capítulo apresenta o protocolo metodológico adotado para o desenvolvimento, refinamento e execução do *prompt* instrucional desta pesquisa, elaborado para operar como uma ferramenta de apoio à validação de ACs no padrão BDD orientados a aspectos de UX. A iniciativa parte do reconhecimento, amplamente discutido na literatura (Virvou, 2023; Souza, 2021), da dificuldade enfrentada por times ágeis em sistematizar atributos de usabilidade, acessibilidade e *feedback* de interação nos ACs.

O protocolo foi estruturado em três estágios sucessivos, conforme ilustrado na Figura 13. Cada etapa foi planejada de forma incremental, respeitando princípios de design iterativo e validação incremental de artefatos de pesquisa, o que se

revelou indispensável pela complexidade e pela natureza exploratória do uso de LLMs aplicadas à análise de critérios textuais técnicos.

Figura 13: Protocolo de Desenvolvimento do *Prompt* Instrucional.

Fonte: Elaborado pela autora.

As seções seguintes detalham a seleção dos modelos generativos, e a estruturação para modelagem do *prompt* instrucional.

6.1 Seleção dos Modelos Generativos

A revisão da literatura ajudou a nortear a escolha de quais LLMs seriam mais aderentes ao estudo desta pesquisa. Esse levantamento teve por objetivo investigar as capacidades, aplicações e limitações dos principais LLMs, considerando critérios como estabilidade, capacidade de raciocínio, suporte a instruções condicionadas e compatibilidade com metodologias de *prompting* avançadas.

Dois principais LLMs disponíveis no mercado foram selecionados para compor o estudo empírico na fase comparativa: ChatGPT-4o e Gemini 2.5 Flash. Ambos são reconhecidos pela literatura atual como modelos de alto desempenho para tarefas de avaliação contextual e análise instrucional (Virvou, 2023). Além disso, esses modelos apresentam compatibilidade com técnicas avançadas de *prompting*, requisito central para esta pesquisa.

A rápida evolução de grandes modelos é, em grande parte, uma resposta à crescente demanda por sistemas de IA para gerenciar desafios realistas e complexos com alta precisão (Jurenka et al., 2024). O ChatGPT aprimora suas habilidades de conversação incorporando aprendizado por reforço a partir do *feedback* humano (*Reinforcement Learning from Human Feedback* - RLHF, em inglês), permitindo que ele gere respostas que são sensíveis ao contexto e fluentes (Mondillo et al., 2024). Em contraste, o Gemini se diferencia com um design

multimodal que mescla texto, imagens e áudio, permitindo que ele processe e produza saídas em diferentes tipos de dados.

Tendo em vista a abordagem conversacional característica dos modelos ChatGPT-4o e o Gemini 2.5 Flash, integrados ao ambiente de teste utilizado neste estudo, o Poe.com (*Platform for Open Exploration*, em inglês), plataforma *opensource* que possibilita a execução de LLM em tempo real, a mensagem de apresentação do *prompt* encoraja os usuários a interagirem com o modelo para se aprofundarem em suas respostas durante o processo de validação dos ACs avaliados. Essa decisão metodológica não apenas ampliou a coleta de evidências qualitativas mais eficientes sobre os ACs avaliados, mas também oportunizou a análise das respostas geradas em contextos dialogados, aproximando o uso da ferramenta à dinâmica prática dos times ágeis. Tal motivação se mantém em conferência ao que foi apresentado por Ouyang et al. (2022) os quais relatam que o ajuste com *feedback* humano, aumenta substancialmente a obediência a instruções do usuário, útil para explicar por que modelos afinados obedecem melhor ao *prompt* mesmo com menos parâmetros. Segundo os autores, essa adaptação favorece a obediência às instruções de *prompt*, tornando as interações mais precisas e ajustadas às intenções do usuário.

As versões dos respectivos modelos generativos foram consideradas por serem as atualizações mais recentemente publicadas. Essa escolha se fundamenta para assegurar a baixa manutenção deste estudo e estar em alinhamento com configurações mais refinadas de cada modelo. Para análise técnica e escrita acadêmico-científica, como em ER, UX, BDD, ChatGPT-4o apresenta raciocínio formal mais robusto, com altíssima performance em dedução, indução e abdução (OpenAI, 2024b). Contudo, apesar de potente, pode degradar em fidelidade a instruções simples ao longo de uma sessão longa ou instruções repetidas (Chen, Zaharia & Zou, 2024). Segundo Chen, Zaharia & Zou (2024) a consistência de obediência ao *prompt* decresce sob *prompts* ambíguos ou de baixo controle condicional. Por esse aspecto, está sendo considerado que se o *prompt* for preciso, bem delimitado, com exemplos ou condicionais explícitas, o ChatGPT-4o pode obedecer os comandos de forma consistente.

Quando conferida a estabilidade e consistência em LLMs, está sendo avaliado se o modelo responde aos comandos quando recebe a mesma instrução em momentos diferentes, como também, se o modelo consegue manter coerência de comportamento ao longo de várias interações na mesma sessão ou conversa. Chen, Zaharia & Zou (2024) mostram empiricamente que o GPT-4 sofreu mudanças substanciais de comportamento em tarefas objetivas e de *reasoning* em 2023. Os autores observaram mudanças no comportamento do modelo ao longo do tempo, mesmo quando as entradas permanecem as mesmas em problemas matemáticos e geração de código. Observaram que instruções bem estruturadas mitigaram essa instabilidade, reforçando que a estabilidade é alta em sessões comandadas por *prompts* estruturados, mas sensível a atualizações do modelo e a contextos de discussão mais aberta.

O Gemini 2.5 Flash é o primeiro modelo *Flash* da Google Deepmind com capacidade de raciocínio, que permite ver o processo de pensamento do modelo ao gerar uma resposta (Google DeepMind, 2025b). Diferentemente dos modelos Flash anteriores, o Gemini 2.5 Flash, em sua performance multimodal, é o primeiro da linhagem Flash com suporte a processo de *"thinking"* configurável, permitindo controlar até que nível o modelo raciocina antes de gerar a resposta (Google DeepMind, 2025b). Isto é, se descreve como um modelo de raciocínio híbrido com modos *"thinking"* e *"non-thinking"* alternáveis, latência abaixo de 50ms, mas com restrição de 24 000 tokens para raciocínio e dificuldades em tarefas de dedução lógica (Google DeepMind, 2025b). A OpenAI declara que esse mecanismo melhora significativamente o desempenho em tarefas de raciocínio complexo, permitindo equilibrar custo, latência e qualidade, o que realça a performance eficiente em tarefas cotidianas condicionadas para uso prático e com instruções claras, mas também indica limitação em contextos de raciocínio acadêmico mais exigentes. Nas avaliações realizadas pela empresa, os indicadores de uso mostraram que o Gemini 2.5 Flash é ideal para cenários práticos que exigem baixa latência e alto throughput, como *chatbots*, classificação, tradução e aplicações em escala corporativa.

Embora não se iguale a precisão máxima do modelo Pro em tarefas altamente técnicas, o Gemini 2.5 Flash se posiciona como uma opção robusta e confiável para atender aplicações de grandes volumes de solicitações simultâneas e

real-world, como *chatbots*, *voicebots*, APIs de geração de texto ou sumarização, tradução automática para múltiplos idiomas, Classificação de conteúdo, análise de sentimento, rotulagem de dados, transcrição automática, entre outros. Ou seja, o modelo é ideal para casos de uso dos quais prioriza velocidade e escalabilidade sobre profundidade de raciocínio, permitindo controlar quanto o modelo "pensa", o que torna o modelo adequado a operações práticas, como a validação diretrizes de UX em ACs, mas com clarezas limitadas em análises complexas e que exigem maior aprofundamento técnico (Google DeepMind, 2025b; Li et al., 2025).

A série Gemini 2.X utiliza uma configuração reduzida de especialistas (*experts*) em relação a sua arquitetura inicial, *mixture-of-experts* (MoE), para equilibrar desempenho e custo computacional, ativando somente partes da rede por token de entrada (Google DeepMind, 2025b). A Google menciona que esse tipo de arquitetura é mais enxuta, o que significa que diminui a aleatoriedade na escolha de caminhos de inferência internos, prioriza a execução mais determinística em tarefas padronizadas, e reduz a propensão a variar o *reasoning* dependendo do ciclo de interação entre o usuário e o modelo (turno conversacional) ou contexto. A Tabela 7, a seguir, apresenta as conclusões empíricas levantadas sobre os LLMs utilizados nesta pesquisa.

Tabela 7: Critérios comparativos utilizados para avaliar os LLMs.

Critérios comparativos	**Conceito**	**Gemini 2.5 Flash**	**ChatGPT-4o**
Obediência ao *prompt*	Envolve avaliar até que ponto o modelo segue instruções explícitas, adere às restrições e produz uma saída que se alinha com a intenção do usuário.	Alta, mais estável às instruções do *prompt* com menor propensão a variações espontâneas de comportamento	Boa, mas pode degradar em sessões longas ou repetidas
Especialização em áreas técnicas	Requer avaliar como o modelo interpreta a função técnica designada	Boa para tarefas práticas específicas	Alta em áreas especializadas
Estabilidade e consistência	Avalia a confiabilidade com que um modelo fornece saídas semelhantes ou idênticas para as mesmas entradas ou entradas	Alta, menos propenso a flutuações de comportamento	Alta estabilidade em sessões dirigidas, mas sensível a atualizações da versão do modelo e ao contexto envolvido

semanticamente equivalentes ao longo do período de análise		

Fonte: Dados retirados dos periódicos *system cards* de ambos LLMs (OpenAI, 2024a; Google DeepMind, 2025a).

Avaliar a obediência de um modelo às instruções de um *prompt* muitas vezes requer uma combinação sistemática de avaliação humana, métricas automatizadas e *benchmarks* especializados. Para tanto, os dados avaliados pelos periódicos também foram confirmados por análises quantitativas conduzidas pela plataforma independente de *benchmarking* de LLMs, Artificial Analysis (Artificial Analysis, 2025a, 2025b), a fim de reforçar a compreensão sobre suas capacidades. Todos os critérios comparativos foram analisados empiricamente pelo avaliador humano. Neste trabalho, se reserva ao avaliador humano comparar a resposta do LLM à uma "verdade básica" conhecida ou resposta ideal. Essa coordenação requer conhecimento de textos de referência criados por especialistas e autores acadêmicos.

A especialização em áreas técnicas se atribui ao quão o modelo foi aderente à interpretação de papéis de especialista, através do comando que solicita o LLM adotar a função de uma persona, como "Atue como um arquiteto de *software* sênior" ou "Você é um especialista em engenharia de requisitos" para aprimorar seu estilo de resposta e profundidade técnica. Outra estratégia utilizada neste trabalho é a indução por cadeia de pensamento (CoT), onde é pedido ao LLM "pensar passo a passo" para forçá-lo a demonstrar seu processo de raciocínio. Esse método é particularmente eficaz para problemas complexos que envolvem raciocínio. No entanto, é importante ressaltar que os LLMs não podem e não são capazes de substituir totalmente os especialistas humanos devido às limitações no raciocínio de alto nível, no manuseio de fluxos de trabalho complexos e multissistêmicos de responsabilização. É correto afirmar que o uso mais eficaz dos LLMs nessas áreas é ampliar as capacidades humanas, aumentando a produtividade e reduzindo o volume de tarefas, em vez de substituí-las completamente. O Subcapítulo 2.3 apresenta o conceito de CoT e outros exemplos práticos de como pode ser utilizado.

O critério de estabilidade e consistência foi estabelecido a fim de medir por caso de uso a frequência com que um LLM fornece uma resposta semelhante à

mesma pergunta. No passado, medidas baseadas em correspondência lexical eram comumente usadas para avaliar a consistência do modelo (Zipf, 1935; Elazar et al., 2021; Kendro et al. 2024). Esses métodos comparam saídas no nível do *token* para determinar se um Modelo de Linguagem Pré-Treinada (*Pretrained Language Model* - PLM, no inglês) produz a mesma saída para a mesma entrada. No artigo *Attention Is All You Need*, escrito por Ashish Vaswani et al. (2017). com o Google Brain Team, o autor especifica a arquitetura codificador-decodificador *Transformer* que se tornou a base de todos os PLMs e LLMs modernos, e processa o texto em unidades discretas (*tokens*). O Subcapítulo 2.3 também apresenta mais informações sobre *tokens*. Essa abordagem de correspondência lexical foi uma convenção metodológica que emergiu naturalmente da forma como a arquitetura de PLMs (por exemplo, BERT, RoBERTa e GPT-1) foram construídos. No entanto, era considerado apenas a correspondência lexical, não a correspondência semântica.

Devido a essa limitação, uma nova medida de consistência foi proposta, levando em consideração a correspondência semântica (Elazar et al., 2021; Yang J et al., 2025). Esses métodos avaliam se as saídas de um modelo são semanticamente semelhantes, ou seja, se as duas saídas transmitem o mesmo significado. Esses métodos avaliam se o modelo produz respostas consistentes centradas no significado e não na escolha de palavras. Para esse fim, diversas métricas de concordância semântica foram desenvolvidas e utilizadas para avaliar a consistência dos modelos. No final, as medidas de correspondência semântica fornecem uma melhor avaliação da consistência do modelo do que as medidas tradicionais de correspondência lexical e são mais adequadas para gerar respostas naturais que sejam relevantes para a confiança do usuário.

As versões dos respectivos modelos generativos utilizados neste trabalho foram selecionadas por corresponderem às atualizações mais recentes disponíveis. Essa escolha se fundamenta em uma menor necessidade de manutenção futura deste estudo, além de assegurar compatibilidade com as configurações mais refinadas e estáveis de cada modelo no momento da análise.

Após a seleção dos LLMs, foram criadas duas instâncias de *chatbot* no Poe.com, cada uma configurada para receber um dos modelos, conforme apresentado no Capítulo 7. Na configuração de cada instância de *chatbot*, foi

incorporado o *prompt* instrucional desenvolvido, e uma base de conhecimento[8] para os LLMs tomarem como referência as *guidelines* ACUX em suas recomendações de melhoria da escrita dos ACs testados. Todas as informações presentes na cartilha online do ACUX, como visto no Capítulo 5, foram transcritas para um arquivo *.pdf* livre de formatação para facilitar a interpretação dos LLMs sobre cada *guideline*.

6.2 Estrutura do *Prompt*

Para conduzir a operação dos LLMs como uma ferramenta STI e considerando as particularidades identificadas em cada modelo (OpenAI, 2024a; Google DeepMind, 2025a), foram definidos três métodos de *prompting*, aplicados de forma combinada conforme demonstrado na Tabela 8. A descrição conceitual de cada um dos métodos está descrita no Subcapítulo 2.3. A partir dessa combinação, os LLMs podem operar de maneira mais controlada, reduzindo ocorrências de respostas inconsistentes ou superficialmente elaboradas, um problema frequente em aplicações de *prompts* genéricos, como discutido por Brown et al. (2020). Além disso, ficou definido incorporar os exemplares de USs/ACs utilizados por Souza (2021), conforme descrito no Capítulo 5, ao *prompt* instrucional desenvolvido pela autora desta dissertação para estabelecer o conjunto seguro de *few-shot* ao aprendizado dos LLMs na indicação das *guidelines* ACUX. Todo o conjunto de *few-shot* desses exemplares podem ser vistos no Apêndice A deste documento.

Ambos modelos têm suporte ao CoT. Yu et al. (2023) apresentam um panorama abrangente sobre CoT, avaliando como variáveis de design de *prompt* (*zero-shot vs. few-shot*, ordem das etapas, relevância, qualidade dos exemplos, etc) afetam a eficácia em diferentes contextos em tarefas técnicas, como, por exemplo, tarefas de ER e de UX. Dentre os fatores avaliados pelos autores, foi investigada a complexidade da tarefa, qualidade dos exemplares e alinhamento com o domínio da aplicação. O estudo mostra que a complexidade do problema é um dos determinantes mais fortes na eficácia do CoT. Quando solicitado executar tarefas simples, *prompts zero-shot*, como *"Let's think step by step"* ou *"Explain your reasoning before answering"*, pode ser suficiente para o raciocínio CoT. No entanto,

[8] Base de Conhecimento do ACUX Tutor 1.0: https://drive.google.com/file/d/1vFPHQ5raFcKZsV6QnhIafOA4paKolsMl/view?usp=drive_link

à medida que a complexidade aumenta, especialmente em questões que envolvem múltiplas operações lógicas ou matemáticas, há uma necessidade maior por exemplares pré-formatados, *prompts few-shot*, para que o modelo produza raciocínios corretos. A partir dessas análises é possível notar que o raciocínio CoT tem relação direta com qual estratégia de aprendizagem utilizar, *zero-shot ou few-shot prompting*, para reforçar o padrão de *reasoning* desejado. Seguindo essa visão, foi utilizado para esta pesquisa a estratégia *few-shot* (Yu et al., 2023).

Diferente de comandos genéricos, a combinação desses métodos em LLMs estabelece a explicitação sequencial de critérios, etapas, restrições e parâmetros de avaliação, funcionando como um verdadeiro protocolo operacional para o modelo generativo, apoiando, inclusive, o raciocínio CoT. O modelo, ao receber essas instruções previamente parametrizadas, assume uma função tutora, sendo responsável por realizar análises sistemáticas dos ACs fornecidos, identificando inconsistências, sugerindo aprimoramentos na escrita de modo a incluir aspectos de UX.

Tabela 8: Métodos de solicitação utilizados na modelagem do *prompt* instrucional.

Métodos de Solicitação utilizados no *Prompt* Instrucional	**Características estruturais**
Chain-of-Thought (CoT) Prompting	Estimula o raciocínio passo a passo das LLMs, permitindo a explicitação de cada etapa da análise. Isto é, conduzir o raciocínio do modelo de forma encadeada, assegurando que cada *guideline* da técnica ACUX seja avaliada sistematicamente antes de apresentar recomendações e conclusões.
Instructional Prompting	Responsável por direcionar as respostas dentro de regras explícitas de atuação, conforme as diretrizes da técnica ACUX, e delimitar o comportamento analítico do modelo.
Few-Shot Prompting	Utilizado para exemplificar análises ideais de ACs previamente corrigidos, servindo de referência para o modelo durante as respostas geradas, na promoção de consistência e previsibilidade das análises realizadas pelos LLMs.

Fonte: Elaborada pela autora.

Para alcançar o sucesso na modelagem da ferramenta de tutoria partindo por essa combinação metodológica dos métodos de solicitação, o *prompt* passou por quatro versões de refinamento. A modelagem inicial considerou os requisitos

funcionais da pesquisa e as diretrizes definidas pela técnica ACUX (Souza, 2021). O processo de modelagem incluiu o seguinte regimento:

- Estruturação de comandos e instruções específicas para análise UX de ACs.
- Definição de um template de resposta estruturado em *Markdown*, visando padrão e clareza.
- Inclusão de exemplos de análises (*few-shots*) para orientar o modelo.
- Estabelecimento de critérios de validação para formatação BDD e apontamentos UX.

As versões preliminares foram testadas no ambiente de teste do Poe.com, por meio de execução simulada de critérios previamente analisados. Esses ciclos de avaliação revelaram importantes oportunidades de aprimoramento, especialmente quanto à clareza das instruções, à forma de apresentação das recomendações e à aplicação consistente das *guidelines* do ACUX. Durante esses ciclos iterativos de avaliação, foram identificadas importantes oportunidades de aprimoramento, sobretudo relacionadas à formatação das instruções, à estrutura de apresentação das recomendações e à consistência na aplicação das *guidelines* de UX. Esses achados orientaram ajustes pontuais e estratégicos na redação das instruções, na modelagem das respostas esperadas e na configuração das condicionais responsáveis pela condução da análise encadeada, CoT.

A validação do *prompt* instrucional foi realizada por um especialista atuante na área de EP, que contribuiu realizando dois ciclos de revisão. A versão final do *prompt* instrucional utilizado neste estudo foi disponibilizada no Apêndice A deste trabalho, servindo como referência replicável para futuras investigações do mesmo contexto de análise.

6.3 Considerações Finais

A modelagem do *prompt* instrucional representou um ponto de inflexão metodológico nesta pesquisa, consolidando a transição entre a fase exploratória, e a fase comparativa, orientada à testagem automatizada com LLMs. Este capítulo

cumpriu, assim, o papel de formalizar o protocolo de interação entre o raciocínio humano e o raciocínio generativo, traduzindo diretrizes conceituais de UX e engenharia de requisitos em comandos operacionais compreensíveis pelos modelos de linguagem.

O principal objetivo dessa etapa foi desenvolver um *prompt* instrucional capaz de operar como tutor inteligente na validação da escrita de UX em ACs, guiando os LLMs a identificar inconsistências, sugerir aprimoramentos e reconhecer atributos ausentes ou mal definidos relacionados ao usuário. Para isso, o processo envolveu a revisão da literatura sobre *prompting* avançado, a seleção criteriosa dos modelos generativos, ChatGPT-4o e Gemini 2.5 Flash, e a combinação estruturada de três métodos de *prompting* (*instructional, few-shot* e CoT) aplicados de forma coordenada.

A seleção dos LLMs foi fundamentada em critérios técnicos e teóricos que consideraram estabilidade, precisão contextual e aderência a instruções condicionadas. O ChatGPT-4o foi escolhido por seu raciocínio formal robusto e capacidade de dedução em tarefas complexas, enquanto o Gemini 2.5 Flash foi selecionado por sua consistência operacional e arquitetura multimodal, que favorece a compreensão contextual em aplicações práticas. A análise comparativa desses modelos demonstrou a importância de compreender não apenas o desempenho geral dos LLMs, mas também a forma como cada modelo responde a instruções estruturadas e raciocínios encadeados, tema central nas metodologias de *prompt* engineering contemporâneas (Chen, Zaharia & Zou, 2024; Google DeepMind, 2025b).

A estruturação do *prompt* resultou em um artefato metodológico híbrido, simultaneamente técnico e pedagógico para os LLMs, que combina instruções explícitas, exemplos contextualizados (*few-shots*) e parâmetros de formatação para reduzir ambiguidades e favorecer a coerência nas respostas. A incorporação de exemplares de ACs validados com o método ACUX desempenhou papel estratégico, funcionando como base de aprendizado semântico para o raciocínio dos modelos. Essa abordagem permitiu alinhar o processo automatizado de análise ao padrão empírico humano, garantindo maior comparabilidade na etapa comparativa posterior. O resultado foi um protocolo de *prompting* replicável, com

valor científico e aplicabilidade prática em futuras pesquisas sobre automação de validação de requisitos.

Em síntese, a modelagem do *prompt* instrucional consolidou os fundamentos teóricos e práticos que sustentam o estudo realizado nesta dissertação. As decisões tomadas nesta fase, tanto na escolha dos modelos quanto na estrutura das instruções, configuram o núcleo técnico do ACUX Tutor 1.0, permitindo operacionalizar, de forma sistemática, a validação automatizada de ACs sob a óptica da UX.

Com isso, o estudo avança para a próxima etapa, Execução dos LLMs, na qual o desempenho dos modelos será empiricamente avaliado em termos de precisão técnica, concordância e explicabilidade. Esta transição marca o início da testagem de hipóteses centrais sobre o estudo empírico comparativo com LLMs (Wagner et al., 2025; Baltes et al., 2025), permitindo comparar a atuação dos LLMs frente ao avaliador humano e analisar, com rigor, seu potencial como tutores no apoio à validação de requisitos orientados à UX.

7 Execução dos LLMs

7.1 Realização dos Testes

Essa etapa do estudo empírico constitui comparar, por meio de testes, o desempenho contextual de dois LLMs (ChatGPT-4o e Gemini 2.5 Flash) a fim de avaliar suas atuações como tutores na redação de ACs com aspectos UX incorporados. Para isso, se faz necessário analisar se as respostas geradas estão contextualizadas e alinhadas ao domínio da tarefa.

A execução dos testes foi realizada na plataforma *opensource* Poe.com que atua como um agregador, permitindo que usuários interajam com uma ampla variedade de LLMs de GenIA. compatível com a operação de LLMs comerciais e que oferece recursos de controle e visualização das respostas geradas a partir de *prompts* personalizados. Nesse ambiente, os modelos foram instanciados por meio de *bots* dedicados, configurados para executar o *prompt* instrucional desenvolvido neste estudo. Para fins de caracterização, a composição dessa estrutura

automatizada recebeu o nome sugestivo de ACUX Tutor 1.0, representando a operação de tutoria inteligente proposta, sendo apresentada pelas seguintes instâncias: ACUXTutor1.0_ChatGPT[9], e ACUXTutor1.0_Gemini[10]. Os usuários finais esperados para utilizar tal abordagem automatizada são principalmente *Product Owners*, os quais podem direcionar o uso da ferramenta junto ao time em revisões de requisitos encorajando-os a beneficiar a etapa de validação de ER.

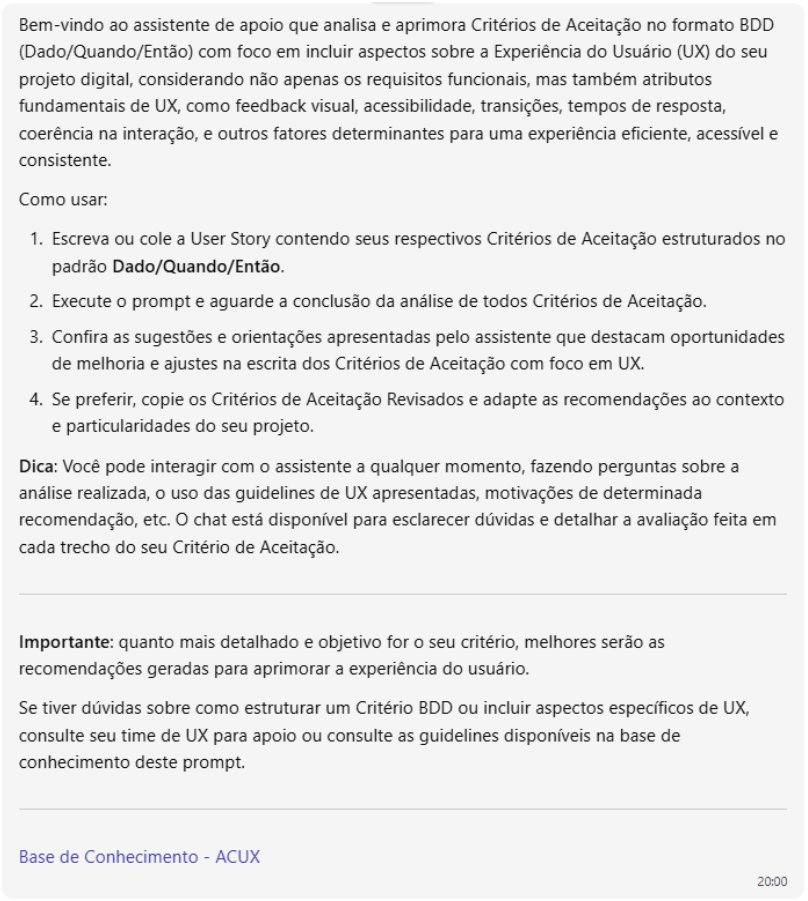
A partir dessa configuração, os mesmos 20 ACs apresentados no Capítulo 5 foram utilizados como dados de entrada no ACUX Tutor 1.0 para serem testados, servindo como base para a avaliação comparativa entre os modelos. Durante os testes, foi assegurado o controle da temperatura criativa das respostas (0 a 0,3), a aplicação dos *templates* de resposta e o cumprimento de todas as diretrizes estabelecidas no protocolo de desenvolvimento dos *prompts*.

O ACUX Tutor 1.0 se inicia com a ativação do ambiente Poe.com apresentando uma mensagem de apresentação, conforme ilustrado na Figura 14, sendo enviada sempre que se inicia a conversa com o modelo.

[9] ACUXTutor1.0_ChatGPT: https://poe.com/ACUXTutor1.0_ChatGPT

[10] ACUXTutor1.0_Gemini: https://poe.com/ACUXTutor1.0_Gemini

Figura 14: Mensagem de apresentação do ACUX Tutor 1.0.



Fonte: Elaborado pela autora.

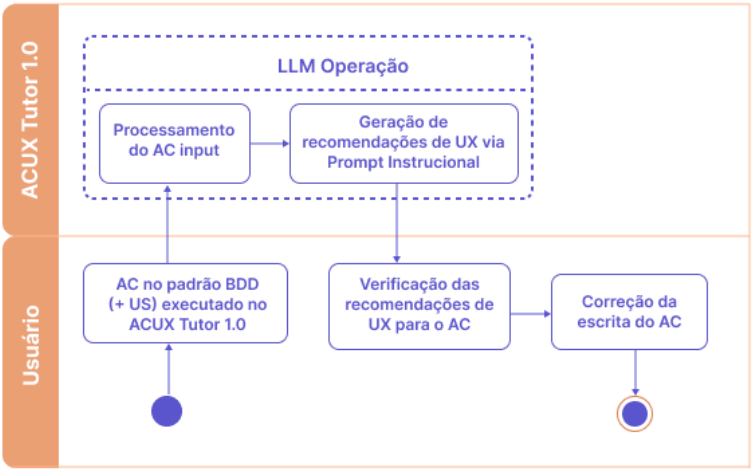
O usuário é instruído a inserir na interface do *chatbot* a descrição de uma US junto com seus respectivos ACs, para oferecer mais contexto à ferramenta. O modelo então executa o processamento de forma uniforme, desde que os ACs estejam escritos no padrão BDD. Uma vez recebido os ACs como entrada, o ACUX Tutor 1.0 aproveita as capacidades de PNL dos LLMs para extrair as informações escritas, interpretar e segmentar o conteúdo textual buscando trechos específicos que descrevem ações, condições e resultados esperados pelo usuário. Em seguida, a operação do LLM é acionada para iniciar a análise de cada trecho identificado do AC a partir das instruções e orientações definidas no *prompt* instrucional. O objetivo nesta etapa é identificar oportunidades de melhoria em relação à UX, orientadas pelas *guidelines* ACUX (Souza, 2021) e exemplos qualificados disponibilizados via *few-shot*.

Para cada AC fornecido pelo usuário, os trechos identificados são separados e analisados verificando se há melhoria na escrita de aspectos de UX que não foram explicitados de alguma forma. O modelo deve indicar se o trecho identificado

está "Adequado", "Incompleto" ou "Inadequado" perante todas as *guidelines* ACUX relevantes que estão associadas ao contexto analisado. Acompanhando essa indicação, o modelo deve também apresentar, obrigatoriamente, uma breve justificativa para cada apontamento realizado. Isso garante a fundamentação das recomendações fornecidas e permite que o usuário reconheça os motivos que sustentam cada sugestão apresentada pelo modelo. Essa estratégia de recomendação das *guidelines* reafirma a proposta da abordagem tutoria adotada nesta pesquisa dado que conduz o modelo a atuar de modo explicativo e didático na manipulação dos ACs submetidos.

Ao final da operação de processamento dos trechos, o LLM reescreve o AC no padrão BDD aplicando as melhorias identificadas. A versão revisada do AC é disponibilizada no mesmo ambiente de *chat*, permitindo a conferência e validação das recomendações sugeridas. O usuário então, pode optar copiar o AC revisado e realizar as devidas correções no documento de requisitos do projeto em que trabalha; retomar a análise para submeter um novo AC, ou ainda, interagir com o *chatbot* para aprofundar o entendimento sobre as recomendações sugeridas pelo modelo. A Figura 15 ilustra, de forma abstrata, como funciona todo esse fluxo de interação e operação do ACUX Tutor 1.0.

Figura 15: Diagrama do processo de geração de recomendações de UX em AC usando ACUX Tutor 1.0.



Fonte: Elaborado pela autora.

As Tabelas 9 e 10 a seguir listam todos os testes executados em cada um dos ambientes de teste em cada um dos LLMs.

Tabela 9: Testes no ACUXTutor1.0_ChatGPT.

ACUXTutor1.0_ChatGPT	
Projeto	**Link do Teste**
USs/ACs do Projeto 01 - 2023 - ChatGPT-4o	https://poe.com/s/6y8BiaUyVJxUgODad3E7
USs/ACs do Projeto 02 - 2023 - ChatGPT-4o	https://poe.com/s/7gEePYS1K81ahVeHqrCn
USs/ACs do Projeto 01 - 2024 - ChatGPT-4o	https://poe.com/s/R43uAXwbeuS5LltxeJal
USs/ACs do Projeto 02 - 2024 - ChatGPT-4o	https://poe.com/s/aUYSDN5MULaltkTa4n9b

Fonte: Elaborado pela autora.

Tabela 10: Testes no ACUXTutor1.0_Gemini.

ACUXTutor1.0_Gemini	
Projeto	**Link do Teste**
USs/ACs do Projeto 01 - 2023 - Gemini 2.5 Flash	https://poe.com/s/iaQhv6C7RVuuuQDFQab3
USs/ACs do Projeto 02 - 2023 - Gemini 2.5 Flash	https://poe.com/s/3mDgMX6RJOTPuDCBA9Yu
USs/ACs do Projeto 01 - 2024 - Gemini 2.5 Flash	https://poe.com/s/yH69FHe1kqIYLguu4nEH
USs/ACs do Projeto 02 - 2024 - Gemini 2.5 Flash	https://poe.com/s/t0vmGUQdklfRDBNgDAlV

Fonte: Elaborado pela autora.

7.2 Métricas de Desempenho sobre os LLMs como Tutores de ACs

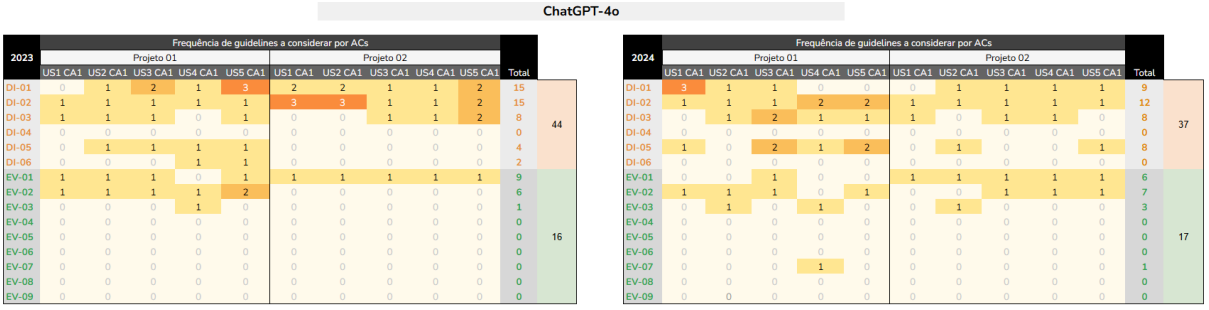
Com a execução dos testes em todos os ACs, esta etapa tem como objetivo descrever os procedimentos adotados para operacionalizar toda avaliação de desempenho das respostas geradas pelos LLMs, considerando critérios de comparação, os parâmetros de controle e os instrumentos utilizados para a coleta e organização das informações, estabelecendo a base para as análises qualitativa e quantitativa subsequentes. O intuito não foi avaliar a qualidade textual das

respostas, mas sim verificar a presença de recomendações contextualizadas que efetivamente se aplicassem aos ACs dentro do domínio da UX.

Por esse fim, a avaliação deu início com o avaliador humano, o mesmo responsável pelas atividades realizadas no Capítulo 5, contabilizando os trechos identificados pelo LLM como "**Inadequado**" ou "**Incompleto**" verificando, para cada AC e cada recomendação, se a *guideline* ACUX indicada "se aplica" (1) ou "não se aplica" (0) ao contexto analisado. Ou seja, o avaliador humano buscou avaliar se as recomendações consideradas relevantes pelos LLMs são pertinentes semanticamente e tem aplicabilidade prática à revisão da escrita dos ACs.

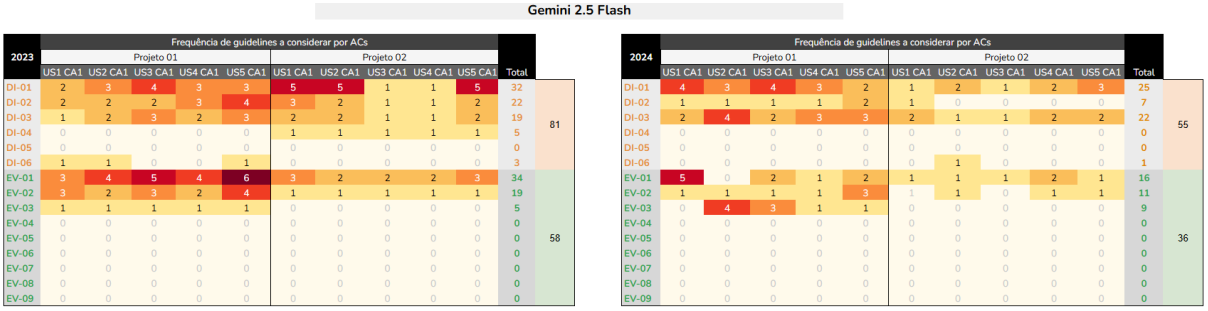
De tal modo, os julgamentos realizados sobre as respostas dos LLMs permitiram identificar evidências semânticas que estivessem em alinhamento com o *gold standard* utilizado. Para isso, cada *guideline* ACUX foi tratada como um critério semântico independente, configurando-se como uma unidade normativa de validação para avaliar a presença ou ausência de aspectos de UX nos ACs e sua contextualização. As Figuras 16 e 17 ilustram, respectivamente, o procedimento de registro das ocorrências, realizado através de planilhas do Google Planilhas.

Figura 16: Planilha-matriz dos ACs avaliados pelo ChatGPT-4o.



Fonte: Elaborado pela autora.

Figura 17: Planilha-matriz dos ACs avaliados pelo Gemini 2.5 Flash.



Fonte: Elaborado pela autora.

Através da planilha-matriz do ChatGPT-4o[11], e da planilha-matriz do Gemini 2.5 Flash[12] se permitiu observar, portanto, a quantidade de vezes que cada *guideline* foi recomendada pelos LLMs para revisão ou inclusão de aspectos de UX, seguindo a mesma estrutura da planilha-matriz *gold standard* apresentada no Subcapítulo 5.3. À primeira vista, pela densidade dos resultados que retratam um mapa de calor de ocorrências, é possível observar uma diferença expressiva entre os resultados apontados pelos dois modelos. Porém, por si só, não permite conclusões comparativas mais robustas sobre a performance das respostas. Por essa razão, foram propostas métricas que permitiram mensurar a frequência de **concordância** entre as ocorrências da planilha das Figuras 16/17 e a planilha de referência (*gold standard*) gerada pelo avaliador humano, apresentada nas Figuras 11/12. Além disso, foram avaliadas a **mutualidade de tais concordâncias**, a **precisão técnica** e a **explicabilidade** das recomendações fornecidas pelos LLMs, incluindo ocorrência de falso-positivos. A Tabela 11 apresenta a definição de cada uma dessas métricas utilizadas e como foram verificadas na prática.

Tabela 11: Métricas utilizadas para avaliar o desempenho contextual das respostas dos LLMs.

Métrica	O que mede?	Base de Julgamento	Como medir?
Concordância Simples *(Human-AI Agreement)*	Se o modelo tende a acertar as mesmas *guidelines* ACUX apontadas pelo avaliador humano para melhoria da redação do AC avaliado (Concordância IA × Avaliador Humano individualmente)	Comparação entre recomendações feitas do Avaliador Humano e do LLM	% de coincidência entre "Se aplica/Não se aplica" Proporção de *guidelines* coincidentes entre o LLM e Avaliador Humano
Concordância Mútua Múltipla	Se todos os avaliadores (Avaliador humano, ChatGPT-4o e Gemini 2.5 Flash) concordam exclusivamente sobre as *guidelines* ACUX indicadas por cada um (Concordância simultânea dos três avaliadores)	Convergência entre recomendações feitas pelos avaliadores	% de "Se aplica/Não se aplica" sobre os avaliadores Proporção de *guidelines* coincidentes entre Avaliador Humano, LLM 1 e LLM 2.

[11] Planilha-Matriz # Avaliação do ChatGPT-4o: https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?usp=sharing

[12] Planilha-Matriz # Avaliação do Gemini 2.5 Flash: https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?usp=sharing

Precisão Técnica *(Contextual Accuracy)*	Se a recomendação feita pelo modelo é tecnicamente correta e aplicável ao contexto do AC avaliado, independentemente da explicação	Cada guideline avaliada como "Sim" indica que o modelo aplicou corretamente o aspecto técnico de UX	% de "Se aplica/Não se aplica" sobre total Proporção de *guidelines* recomendadas feitas pelo LLM que são tecnicamente corretas e pertinentes ao contexto específico do AC avaliado, segundo análise do avaliador humano
Explicabilidade *(Clarity/ Justification)*	Se o modelo justifica de forma clara, contextualizada e fundamentada o porquê a recomendação foi feita e qual seu impacto na UX	Justificativa com as causas e as consequências sobre a recomendação compreensíveis e coerentes?	% de recomendações com justificativa explícita ("Sim") Proporção de *guidelines* recomendadas feitas pelo LLM que vêm acompanhadas de justificativas claras, contextualizadas e tecnicamente fundamentadas, segundo análise do avaliador humano

Fonte: Elaborada pela autora.

Essas quatro métricas foram calculadas pelo avaliador humano por meio de em uma Planilha de Mensuração dos Resultados[13] que serviu como instrumento de apoio para análise quantitativa dos resultados. Este estudo empírico não empregou indicadores tradicionais da matriz de confusão, como *precision, F1 score, recall e accuracy*, amplamente utilizados na literatura de PLN, visto que são adequados para tarefas de natureza objetiva (fechada), em que há uma resposta considerada correta, única ou facilmente verificável (Devlin et al., 2019; Jurafsky & Martin, 2023). Tanto o ChatGPT-4o quanto o Gemini 2.5 Flash, por si só, não se determinam como modelos de classificação, mas sim como modelos de GenIA, projetados para gerar saídas textuais abertas e criativas a partir de instruções em LN, e não para categorizar entradas em classes predefinidas (OpenAI, 2024a; Google DeepMind, 2025a). Somente quando explicitamente configurados para tarefas de classificação binária (Karlsson et al., 2025) podem operar sob essa lógica. Entretanto, a

[13] Planilha de Mensuração de Resultados: https://docs.google.com/spreadsheets/d/1dv_sVCtgG4wQe72it7g4VMQc3yPFHZmcCoAzRtE_Vdk/edit?gid=100889159#gid=100889159

motivação desta presente pesquisa difere de tal ideia, visto que as métricas clássicas se mostram discretas e limitantes para capturar nuances de adequação contextual, coerência semântica e qualidade explicativa das respostas dos LLMs (Caglayan et al., 2020). A Tabela 12 apresenta um breve panorama sobre o conceito das métricas mencionadas por tipo de tarefa.

Tabela 12: Métricas por tipo de tarefa.

Classificação das Métricas	Quando utilizar?	Tipo de tarefa realizada pelo LLM
Métricas clássicas (*Precision, Recall, F1, Accuracy*)	Em tarefas fechadas, do tipo classificação "certa/errada" estritamente definível	Detecção de sentimentos, identificação de entidades (*Named Entity Recognition* - NER)
Métricas qualitativas (*truthfulness*, explicabilidade, utilidade, confiança, clareza, originalidade)	Em tarefas abertas, criativas ao qual exige uma resposta elaborada	Escrita de Marketing (*copywriting*), simulação de diálogos complexos, documentação jurídica/regulatória, simulação de papéis de especialista (*role-playing*)

Fonte: Elaborada pela autora.

Todas as métricas deste estudo empírico derivam do clássico método matemático de proporcionalidade, e refere-se à relação comparativa entre grandezas, que permite quantificar e escalar valores de forma consistente. Também foram inspiradas em trabalhos prévios sobre avaliação de LLMs em tarefas de ER e validação de requisitos (Sreekar et al., 2024; Krishna et al., 2024; Karlsson et al., 2025), nos quais o desempenho é medido tanto pela coerência com o julgamento humano quanto pela clareza interpretativa das respostas geradas. A métrica de Concordância se aproxima dos achados de Krishna et al. (2024), cujo trabalho mediu o alinhamento entre julgamentos humanos e automáticos na produção e revisão de SRS, evidenciando a relevância do acordo interpretativo como indicador de proficiência. Já a Precisão Técnica guarda relação direta com os estudos de Karlsson et al. (2025), que examinaram a competência dos modelos em preservar a coerência semântica e a factualidade de requisitos, aspecto essencial para aferir a correção contextual. Por fim, a métrica de Explicabilidade é fortemente inspirada nas contribuições de Sreekar et al. (2024), cujo *framework* AXCEL introduz o uso de LLMs como avaliadores explicáveis, capazes de justificar suas decisões e identificar inconsistências de forma transparente. Essa triangulação evidencia que sua originalidade reside na integração combinada dessas três dimensões (acordo

humano-GenIA, rigor técnico e clareza explanatória) aplicadas de modo semântico-normativo às *guidelines* de UX em ACs.

A concordância mútua múltipla foi adicionada para determinar maior solidez nas análises e maximizar a leitura quantitativa dos dados. Para facilitar a compreensão sobre o cálculo de cada métrica, o ChatGPT-4o foi denominado Modelo 1, e o Gemini 2.5 Flash denominado Modelo 2. Enfatize-se também que o conceito de concordância adotado se assemelha, em partes, ao conceito de confiabilidade, comumente utilizado em outros estudos científicos, na medida que avalia se há regularidade nos acertos presentes na classificação dos trechos escritos dos ACs.

Entre as vantagens de realizar análise de concordância, se destaca pelos tipos de inferências que podem ser extraídas de dados analisados (Barthe et al., 2025; Ahmed et al., 2025; Cohen, 1960; Krippendorff, 2018). Essa visão ajuda na identificação de falhas de inferência e omissões sistemáticas ou contextuais geradas pelos LLMs (Krippendorff, 2018). Isso reflete a preocupação de avaliar fatores que podem comprometer a completude ou a precisão das análises geradas pelas inteligências artificiais, fornecendo, assim, insumos valiosos para o aprimoramento de seus protocolos de *prompting* e de suas capacidades interpretativas.

Apresentados os conceitos e condicionantes, a seguir estão demonstradas as fórmulas matemáticas de cada métrica, e a referência de mais autores que orientaram seus conceitos:

- **Concordância Simples** (Cohen, 1960; Krippendorff, 2018), o cálculo para a métrica foi estabelecido a partir dos seguintes parâmetros:

Concordância (%) = (Nº de *guidelines* coincidentes entre o Modelo N e Avaliador Humano) ÷ (Nº de *guidelines* do Avaliador Humano)×100

- **Concordância Mútua Múltipla** (Ahmed et al., 2025; Krippendorff, 2018), o cálculo para a métrica foi estabelecido a partir dos seguintes parâmetros:

$$\text{Concordância Mútua Múltipla (\%)} = \frac{|G_P \cap G_{M1} \cap G_{M2}|}{|G_P|} X\ 100$$

Sendo que, para cada AC avaliado, as variáveis se define por:

G_P = conjunto de *guidelines* indicadas pelo Avaliador Humano
G_{M1} = conjunto de *guidelines* indicadas pelo Modelo 1
G_{M2} = conjunto de *guidelines* indicadas pelo Modelo 2

- **Precisão Técnica** (Zhou et al., 2022; Zhang Z et al., 2024), o cálculo para a métrica foi estabelecido a partir dos seguintes parâmetros:

Precisão Técnica (%) = (Nº de *guidelines* corretas e aplicáveis) ÷
(Nº total de *guidelines* do Modelo N)×100

- **Explicabilidade** (Bommasani et al., 2021; Ronanki et al., 2023; Topuz et al., 2025), o cálculo para a métrica foi estabelecido a partir dos seguintes parâmetros:

Explicabilidade (%) = (Nº de *guidelines* com justificativa clara e adequada) ÷
(Nº total de *guidelines* do Modelo N)×100

Cada índice percentual de proporcionalidade foi acompanhado por um valor médio geral computado ao final da análise de todas as categorias para medir o desempenho global do modelo, assim como, o desvio padrão. Abaixo são apresentados as fórmulas como cada média foi calculada:

Média de Concordância Simples (%)	$\frac{\sum Concordâncias\ Modelo\ N}{Nº\ de\ Critérios}$
Média de Concordância Mútua Múltipla (%)	$\frac{\sum Concordâncias\ Mútuas\ Múltiplas}{Nº\ de\ Critérios}$
Média de Precisão Técnica (%)	$\frac{\sum Precisão\ Técnica\ Modelo\ N}{Nº\ de\ Critérios}$
Média de Explicabilidade (%)	$\frac{\sum Explicabilidade\ Modelo\ N}{Nº\ de\ Critérios}$

O desvio padrão mede a dispersão em torno das médias por AC, tendo em vista avaliar a estabilidade entre as métricas avaliadas.

Desvio Padrão (%)	$\sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$

Onde:

x_i = cada valor da métrica avaliada

$\bar{x}$ = média da métrica avaliada

x = número de critérios

7.3 Considerações Finais

A execução dos LLMs buscou verificar a viabilidade do ACUX Tutor 1.0 como ambiente operacional de tutoria automatizada. O principal objetivo desta etapa foi comparar o desempenho dos modelos ChatGPT-4o e Gemini 2.5 Flash na análise e validação de ACs orientados por UX, utilizando métricas de Concordância, Precisão Técnica e Explicabilidade, todas aplicadas como indicadores de qualidade interpretativa e confiabilidade contextual das recomendações geradas.

Os resultados iniciais demonstraram que ambos os LLMs foram capazes de operar de forma estável sob o protocolo de *prompting* que foi definido, apresentando respostas coerentes e justificadas. A estrutura sistemática, com controle de variáveis e uso do mesmo conjunto de ACs analisados pelo avaliador humano, garantiu comparabilidade entre os desempenhos e possibilitou a mensuração objetiva das diferenças de raciocínio entre os modelos. A estratégia de utilizar métricas adaptadas de tarefas abertas foi suficiente para extrair *insights* e aprendizados sobre o comportamento de tutoria dos LLMs, uma vez que a natureza interpretativa das análises exigiu considerar não apenas acertos objetivos, mas também a qualidade semântica e a clareza argumentativa das respostas.

A combinação das técnicas de *instructional*, *few-shot* e CoT realmente favoreceram a consistência das saídas e reforçou o papel do *prompt* como elemento de articulação entre o raciocínio humano e o processamento do modelo. Esse resultado reforça a hipótese de que a eficácia de LLMs em tarefas de validação de requisitos depende diretamente da precisão e estrutura das instruções fornecidas, e não apenas da capacidade intrínseca do modelo.

Reunindo todos esses aprendizados, o estudo mostrou que a tecnologia LLM pode reproduzir raciocínios técnicos e heurísticos de especialistas, embora sua

precisão dependa diretamente da qualidade da instrução e do contexto oferecido. É possível dizer que a automatização do processo de revisão da escrita de ACs por meio de LLMs é viável, mas requer contínua supervisão e ajustes humanos. Isso reforça, portanto, que o uso de LLMs deve ser compreendido como uma extensão colaborativa da análise humana, e não como substituição dela.

De modo geral, a modelagem da fase comparativa confirmou a aplicabilidade prática da abordagem proposta, evidenciando o potencial dos LLMs como agentes de apoio à validação de requisitos e a necessidade de métricas contextualizadas para mensurar sua performance. As análises aqui conduzidas estabeleceram as bases quantitativas e qualitativas para o capítulo seguinte, Análise e Interpretação dos Resultados, no qual as evidências obtidas serão discutidas em profundidade, relacionando os achados empíricos às hipóteses da pesquisa e às implicações teóricas sobre o uso de GenIA na ER orientada à UX.

8 Análise e Interpretação dos Resultados

Esta etapa tem como propósito aprofundar os resultados obtidos, articulando as evidências quantitativas e qualitativas a fim de interpretar o desempenho dos LLMs na tarefa de revisão e escrita de ACs sob a óptica da UX. A análise compreende o cruzamento dos resultados das métricas de Concordância, Precisão Técnica e Explicabilidade, associada ainda a observação sobre o comportamento de cada modelo. Para tanto, buscou-se identificar o grau de alinhamento entre as respostas dos modelos e o avaliador humano, bem como captar limitações, potencialidades e oportunidades de aprimoramento no uso de LLMs como instrumentos de apoio para *Product Owners* na validação de requisitos.

Com os testes, o Modelo 1 (ChatGPT-4o) indicou 114 recomendações de *guidelines*, enquanto o Modelo 2 (Gemini 2.5 Flash) apresentou 230 recomendações, onde somados, totalizaram 344 recomendações geradas pelos dois LLMs para os mesmos 20 ACs analisados pelo avaliador humano. A partir desses resultados, foi possível observar comportamentos distintos entre os modelos e inferir suas atuações como tutores inteligentes. Os subcapítulos seguintes apresentam as análises quantitativas e qualitativas sobre os resultados obtidos,

destacando os principais achados que fundamentam a discussão final desta pesquisa.

8.1 Concordância das IAs com o Avaliador Humano

Conforme descrito na Tabela 13, a média geral de concordância do Modelo 1 (ChatGPT 4.0) com o Avaliador Humano foi de 50,38%, enquanto o Modelo 2 (Gemini 2.5 Flash) atingiu 58,60%. Apesar do Modelo 2 apresentar um desempenho superior, a diferença, de aproximadamente 8 pontos percentuais, não é suficiente para classificar como um ganho qualitativo substancial, dada a variabilidade contextual dos critérios avaliados.

Tabela 13: Avaliação Final da Concordância Simples.

Métrica	**Modelo 1** (ChatGPT-4o)	**Modelo 2** (Gemini 2.5 Flash)
Concordância com Avaliador Humano (Média %)	50,38%	58,60%
Desvio Padrão	19,59%	18,55%

Fonte: Elaborada pela autora.

Ambos os modelos conseguiram identificar corretamente boa parte das *guidelines* do ACUX, ainda que não tenham atingido níveis de confiabilidade das recomendações que dispensasse a supervisão de um avaliador humano. Mesmo que atingisse bons níveis de confiabilidade, se faz necessário a atuação do *Product Owners*, *stakeholder* ou cliente impactados diretamente nos ACs para avaliar os *insights*, analisarem as correspondências pertinentes e fazerem juízo de valor adequado. Isso porque se tratando de validação de requisitos, somente esses atores podem concluir sobre tais artefatos de *backlog*.

Já a média de concordância mútua múltipla foi de 41,91%, de acordo com a Tabela 14. Isso indica que menos da metade das recomendações às *guidelines* foram simultaneamente reconhecidas pelos três avaliadores, confirmando as divergências, sobretudo em *guidelines* relacionadas à aspectos estruturais e visuais da interface.

Tabela 14: Avaliação Final da Concordância Mútua Múltipla.

Métrica	
Concordância Mútua Múltipla (Média %)	41,91%
Desvio Padrão	16,51%

Fonte: Elaborada pela autora.

Os resultados dos desvios-padrão indicam instabilidade, em ambas as métricas de concordância, chegando a confirmar que o desempenho dos modelos é sensível ao conteúdo e à clareza dos ACs. Tal perspectiva demonstra ser um indicador crítico de que há baixa convergência contextual entre os LLMs e o padrão humano.

8.2 Precisão Técnica das Recomendações

A Tabela 15 mostra que o Modelo 1 apresentou uma precisão técnica expressivamente superior, com 89,29%, enquanto o Modelo 2 obteve 80,64%. O desvio-padrão de 12,73% para Modelo 1 contra 14,22% do Modelo 2 revela boa estabilidade na qualidade técnica das recomendações ao longo dos ACs analisados.

Tabela 15: Avaliação Final da Precisão Técnica.

Métrica	**Modelo 1** (ChatGPT-4o)	**Modelo 2** (Gemini 2.5 Flash)
Precisão Técnica (Média %)	89,29%	80,64%
Desvio Padrão	12,73%	14,22%

Fonte: Elaborada pela autora.

Esses resultados demonstram que, embora a Modelo 2 tenha maior coincidência com a avaliação humana, o Modelo 1 entregou boas respostas a quase todas as recomendações, elaboradas em conformidade com os princípios de UX e sendo consideradas aplicáveis tecnicamente ao contexto dos ACs. Esse panorama sugere que, se bem treinado e ajustado, o ChatGPT-4o (Modelo 1) pode funcionar como um assistente de validação dando apoio às equipes de requisitos e UX.

8.3 Explicabilidade das Justificativas

A métrica de explicabilidade apresentou valores próximos para as duas tecnologias. A Tabela 16 mostra que o Modelo 1 registrou 75,98%, enquanto o Modelo 2 alcançou 78,95%. No entanto, os desvios-padrão diferiram significativamente.

Tabela 16: Avaliação Final da Explicabilidade.

Métrica	**Modelo 1** (ChatGPT-4o)	**Modelo 2** (Gemini 2.5 Flash)
Explicabilidade (Média %)	75,98%	78,95%
Desvio Padrão	19,93%	15,11%

Fonte: Elaborada pela autora.

De maneira geral, demonstra que ambos os modelos forneceram justificativas contextualizadas para boa parte de suas recomendações. Um aspecto que é positivo para a rastreabilidade e transferência de análises automatizadas. Ainda assim, o Modelo 2 demonstrou maior uniformidade e previsibilidade nesse cenário entre os diferentes ACs analisados.

No que diz respeito à aplicações profissionais, essa boa explicabilidade indica que as recomendações dos modelos podem ser verificáveis e compreendidas pelos membros do time de desenvolvimento, podendo servir de instrumentos para fomentar debates em reuniões de *Backlog Refinement* ou *Planning*.

8.4 Entendimento geral dos LLMs como tutores em *guidelines* ACUX

Foi possível observar variações significativas na aplicabilidade dos modelos sobre as *guidelines* ACUX quando comparadas com às recomendadas pelo avaliador humano. Em ACs cujo conteúdo era mais literal e objetivo sobre as ações esperadas, os modelos tenderam a alcançar maior concordância e precisão técnica. Por outro lado, critérios que exigiam inferências acerca de aspectos visuais, agrupamento, ou ordenação de elementos de interface resultaram em maior divergência entre Modelo 1, Modelo 2 e o avaliador humano, sendo comum a classificação de trechos como “Incompletos” e suas justificativas nesses casos. Isso

sugere que modelos generativos de LN ainda dependem fortemente da objetividade textual para realizar análises mais contextualizadas e completas, assim como é em contexto real de um ambiente de trabalho.

Ainda que esse ponto seja uma atenção relevante a considerar no dado de entrada, como mencionado anteriormente no Capítulo 5, os ACs elaborados pelos alunos, sendo esses também analisados por todos os avaliadores, foram norteados por alguns critérios para documentação de requisitos dos projetos das equipes, incluindo a obrigatoriedade de anexar protótipos de tela que representassem minimamente a usabilidade das funcionalidades propostas por eles. Nesse sentido, é plausível que os estudantes tenham optado por registrar certas características sobre a interface por meio desses artefatos visuais, em vez de descrevê-las textualmente. Ao mesmo tempo, não se exime o aprendizado de que é somente possível estabelecer um entendimento sobre o que será avaliado se forem esclarecidas todas as informações necessárias a respeito, tanto em ambientes virtuais quanto em contextos reais. Isso pode influenciar diretamente o desempenho dos LLMs e contribuir para ocorrências de alucinação das respostas, mesmo em casos de *prompts* instrucionais configurados para gerar avaliações mais assertivas possível e com uma temperatura baixa de criatividade.

O ChatGPT-4o apresentou um comportamento mais conservador em suas recomendações às *guidelines*, bem como nas justificativas apresentando uma explicabilidade mais direta e objetiva. Enquanto o Gemini 2.5 Flash demonstrou maior propensão a sugerir oportunidades de melhoria, mesmo que não necessariamente aplicáveis para o contexto de alguns dos ACs.

O cruzamento entre tais comportamentos identificados durante os testes e valores finais das métricas de desempenho revelou que o Gemini 2.5 Flash obteve boa concordância com o avaliador humano, indicando *guidelines* igualmente semelhantes, ainda assim com explicabilidade mais completa sobre suas recomendações de melhoria. Contudo, a diversidade das análises não foi equilibrada entre os grupos de *guidelines* ACUX. O LLM se concentrou em oferecer muitas resoluções na escrita relacionadas a aspectos de interação dos ACs. O grande volume de recomendações deste modelo é justificado por tal comportamento. Muitas vezes, para um mesmo trecho de AC, múltiplas *guidelines*

sobre interação foram recomendadas. Por mais que tenham sido muitas, foram adequadas em sua maioria, e por esse motivo o Gemini 2.5 Flash foi melhor que o avaliador humano.

Entretanto, observou-se que o comportamento crítico do modelo tendia a variar conforme o contexto e o grau de completude do texto. Foram identificados casos que o modelo "forçou" a recomendação de *guidelines* do grupo Elementos Visuais, buscando compensar a falta de informações mais detalhadas, conforme apontado no Subcapítulo 8.1. Em outras situações, o Gemini 2.5 Flash demonstrava uma tendência a se prender a aspectos excessivamente específicos de uma descrição. Um exemplo pode ser observado no trecho "*[...] Então o sistema disponibiliza os campos 'Nome do professor', 'Cargo', 'CPF', [...]"*, em que o modelo recomendava, por meio da *guideline* EV-02, a especificação do formato da máscara de digitação do campo numérico de CPF, ainda que esse, e os demais campos, fossem posteriormente detalhados na própria descrição do AC. Por essas razões, o modelo demonstra ser melhor para casos que se necessite aprofundar conhecimentos ou estender a semântica sobre a UX de forma sistemática. Essa característica alinha-se às expectativas comerciais declaradas pela Google DeepMind sobre a boa performance do modelo em situações cognitivamente complexas e de análise contextual ampliada (Google DeepMind, 2025b).

O ChatGPT-4o, por sua vez, apresentou recomendações tecnicamente mais precisas, demonstrando uma atuação objetiva na avaliação dos ACs. Apesar de uma explicabilidade ligeiramente inferior em comparação ao Gemini 2.5 Flash, o modelo foi capaz de sintetizar de forma clara e concisa as relações de causa e efeito associadas à não aplicabilidade das *guidelines* recomendadas.

Esse comportamento se dedicou à resolução de trechos ligados a aspectos de organização e interação dos elementos de interface descritos nos ACs. Logo, o LLM conseguiu explorar melhor a indicação às *guidelines* do grupo Design da Interação e Organização da Informação do ACUX. Para trechos associados ao grupo Elementos Visuais, mesmo que assertivas, as melhorias de escrita oferecidas pelo LLM foram esteticamente resumidas em comparação com as geradas pelo Gemini 2.5 Flash.

Já a capacidade de análise contextual foi moderada, com respostas ponderadas e limitadas às necessidades factuais dos ACs sobre a UX. Ainda assim, nos trechos relacionados à organização da interface, o modelo foi melhor que o avaliador humano, apontando recomendações corretivas para trechos “Incompletos” e “Inadequados” que expressavam a UX. Particularmente por esses motivos, o ChatGPT-4o se mostra competente para situações onde a construção da escrita de ACs exija detalhes de UX substanciais e delimitadores para formatação das condições mínimas da US.

8.5 Análise dos Falsos Positivos

Foram detectados falsos positivos nos dois LLMs, apesar de se mostrarem em proporções distintas. O total de 19 falsos positivos foram identificados ao longo da análise, sendo 12 no Modelo 1 (ChatGPT-4o), e 7 no Modelo 2 (Gemini 2.5 Flash). Considerando as 344 recomendações geradas por ambos, esse volume de falsos positivos é equivalente a aproximadamente 5,52% desse total. Embora o desempenho geral dos LLMs tenha se mostrado satisfatório em termos de precisão técnica, existiram situações em que as *guidelines* de UX foram sugeridas de forma indevida ou fora de contexto.

É importante destacar que este estudo entende por falso positivo uma *guideline* ACUX recomendada pelo LLM que, segundo a análise do avaliador humano, não se aplica ou seja pertinente ao contexto do AC analisado. As ocorrências de falsos positivos são um indicativo relevante de confiabilidade dos modelos, uma vez que refletem as respostas não sustentadas tecnicamente e contextualmente. Uma sinal de risco que pode resultar em ruídos na análise e exigir tempo adicional de revisão por parte dos avaliadores humanos. Os cenários mais frequentes de falso positivo identificados nas respostas dos modelos ocorreram em situações de:

- Recomendações excessivas sobre interações, muitas das quais LLMs apontavam a *guideline* DI-01 (“Especificar como o usuário interage com a funcionalidade”) em trechos onde essa informação já estava adequadamente descrita, ou onde a interação era trivial e não exigia detalhamento adicional. Provavelmente, isso pode ter ocorrido por se associarem a tentativa de

contingenciar um comportamento encorajador do CoT. Deste modo aplicaram a *guideline* de forma preventiva ao se depararem com menções de ação do usuário, como “clicar” ou “selecionar”, sem considerar se havia efetivamente uma omissão de contexto ou necessidade real de detalhamento.

- Inferências desnecessárias sobre fluxos de navegação associadas à *guideline* DI-02 (“Especificar o caminho de navegação ou origem da funcionalidade acessada”) mesmo quando o AC não previa transição entre telas ou já descrevia o suficiente sobre como acessar um determinado conteúdo.

Mesmo que os LLMs tenham indicado tais *guidelines* equivocadamente, de maneira geral, a explicabilidade conceitual sobre elas foi abordada corretamente, com respaldo semântico, uma evidência de que os modelos sintetizam bem sobre conceitos universais relacionados às especificidades de áreas de estudo, aqui no caso, a UX. Na prática, ao identificar a *guideline* DI-01 ou DI-02, conseguiram justificar suas definições conceituais e exemplificar para que servem em casos gerais, ainda que tenham alucinado na interpretação contextual do AC. Nesse recorte, a análise de comportamento dos modelos em falsos positivos ampliasse para, dado que uma *guideline* tenha sido recomendada de forma incorreta, a tendência é que sua explicabilidade contextual não se sustente, e, por consequência, seja invalidada proporcionalmente pelo avaliador humano. Essa visão pode indicar que o modelo possivelmente falhou ao indicar a *guideline* porque não conseguiu contextualizar adequadamente sua aplicação diante do que estava descrito no AC. Um problema clássico de inadequação contextual em GenIA. É recomendável, portanto, aprimorar o *prompt* com filtros negativos que mostrem esses casos de inadequação contextual indicando exceções e recomendações impróprias, incluindo *few-shots* detalhados com regras de restrição sobre os contextos envolvidos.

Ainda dentro do total, existiram alguns poucos falsos positivos para a *guideline* DI-03 (“Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação”), e em menor número para as demais *guidelines*. O que levanta a hipótese de que as *guidelines* DI-01 e DI-02 são mais suscetíveis à hiper-recomendação automática pelos modelos. Embora os falsos positivos tenham

implicado pouca distorção nas conclusões gerais das análises de desempenho, sua identificação fornece subsídios importantes para evoluir as abordagens automatizadas que apoiem processos de validação e tornar o uso de LLMs mais seguro e alinhado às necessidades da ER orientada à UX.

8.6 Considerações Finais

Avaliando todos os resultados da análise, as recomendações automatizadas para uso prático estão coerentes com os achados em estudos científicos anteriores sobre a integração de LLM aos processos de ER. Tais evidências indicam que os modelos avaliados têm a capacidade de atuar como ferramentas auxiliares no processo de validação de ACs com foco em UX. Entretanto, não podem ser consideradas como substitutivas, dadas as limitações contextuais apresentadas. A leitura de nuances de intenção e as implicações subjetivas de interação ainda são, até segunda ordem, atributos tipicamente humanos. Ou seja, independentemente dos modelos apresentarem patamares satisfatórios de concordância, precisão técnica ou explicabilidade em seu desempenho, é indispensável a participação ativa de *Product Owners, stakeholders* ou clientes. Cabe a esses atores avaliar os *insights* gerados, verificar a pertinência das correspondências identificadas e emitir um juízo de valor fundamentado sobre os resultados. Isso se deve ao fato de que, no contexto de validação de requisitos, apenas eles são os agentes que detêm conhecimento do domínio e das necessidades do produto. Além disso, possuem legitimidade e contexto suficiente para determinar a adequação e a completude dos artefatos de *backlog*.

Além de tais aprendizados, viu-se que para contextos em que a precisão técnica das recomendações seja prioritária, os dados sugerem utilizar o ChatGPT-4o (Modelo 1), considerando seu desempenho superior e maior estabilidade técnica no estudo conduzido. Por outro lado, o Gemini 2.5 Flash (Modelo 2) demonstrou desempenho contextual mais consistente quanto à clareza e explicabilidade das justificativas, podendo ser indicado para atividades de revisão colaborativa ou para registros documentais, em que a compreensão argumentativa da recomendação se torna relevante. Em qualquer um dos casos, a evidência

reforça a importância da supervisão humana, seja por meio dos *Product Owners*, UX *Designers*, *stakeholders* ou clientes nesse processo de revisão.

9 Conclusão

Este projeto de Mestrado propôs a ACUX Tutor 1.0, uma abordagem automatizada que se apresenta como ferramenta de apoio à validação de requisitos, em especial, orientadora de aspectos de UX em ACs. A proposta surgiu pela aspiração de auxiliar a demanda de validação de requisitos, especialmente USs, mais elaboradas em apoio aos times ágeis na ER a incorporar elementos de UX em ACs. Para isso, o estudo empírico realizado buscou compreender percepções, práticas e maturidade sobre ACs, e demonstrar de que forma LLMs podem atuar como instrumentos de apoio à ER, particularmente, orientados por princípios de UX.

Nesse sentido, a investigação foi conduzida sob um olhar exploratório e comparativo, ancorado em bases conceituais da ES, da Interação Humano-Computador e da Linguística Computacional. O percurso metodológico envolveu desde a análise de conteúdo manual dos ACs escritos por alunos de graduação, até a modelagem e execução dos modelos ChatGPT-4o e Gemini 2.5 Flash, mediados por um *prompt* instrucional, sendo desenvolvido especificamente para este estudo empírico. A articulação entre etapas empíricas e teóricas sustentou a construção de um raciocínio cumulativo, no qual cada fase contribuiu para a consolidação das respostas às duas questões centrais de pesquisa.

Em relação à QP1, os resultados obtidos indicam que é possível sistematizar a revisão da escrita de ACs baseada em *guidelines* de UX utilizando LLMs como apoio à ER, desde que o processo seja cuidadosamente modelado. Os resultados confirmaram essa possibilidade, evidenciando que a estruturação de um protocolo de *prompting* capaz de instruir o modelo sobre conceitos, critérios e exemplos de avaliação resulta em recomendações consistentes, reproduzíveis e semanticamente coerentes. O desenvolvimento do ACUX Tutor 1.0 demonstrou que, quando orientados por técnicas de *instructional, few-Shot* e CoT, os LLMs conseguem interpretar artefatos textuais de *software* e emitir *feedbacks* alinhados às diretrizes de UX. Desse modo, a abordagem se mostrou eficaz em orientar os modelos a agir como avaliadores autônomos, capazes de identificar aspectos de UX ausentes ou

inconsistentes nos ACs analisados. O estudo permitiu discutir que o sucesso da automação em tarefas de natureza interpretativa não reside na substituição da análise humana, mas na capacidade de modelar linguagens instrutivas que reproduzam, de modo inteligível, a lógica avaliativa dos especialistas. O estudo ressalta ainda mais a atenção da supervisão humana permanecendo como indispensável nesse processo. É importante considerar também que ajustes na EP, em especial, na configuração dos *few-shots* e na introdução de exemplos negativos se mostram atributos importantes para melhoria da aprendizagem dos modelos.

No que se refere à QP2, se verificou que abordagens baseadas em LLMs podem beneficiar os times de desenvolvimento ao ampliar a qualidade das discussões sobre os requisitos e fortalecer a comunicação entre profissionais e *stakeholders* de TI. Essa evidência foi inicialmente observada nos testes realizados com o ACUX Tutor 1.0, aplicados sobre ACs produzidos por alunos de graduação. Embora inseridos em um contexto acadêmico, os resultados obtidos permitem projetar o potencial de sua aplicabilidade em ambientes corporativos, configurando um ensaio prático do uso de LLMs como mediadores de colaboração entre os atores diretamente envolvidos e interessados no requisito, um propósito compartilhado por ambos os contextos. Para tanto, a abordagem não apenas pode incentivar a percepção dos profissionais sobre os aspectos qualitativos da UX, como também pode estimular a reflexão coletiva sobre a clareza, completude e testabilidade dos requisitos. As recomendações geradas pelos modelos, mostraram-se certamente úteis servindo como mecanismo de aprendizagem e revisão coletiva, assim como promotoras em potencial de discussões mais maduras entre desenvolvedores, *designers* e *Product Owners*. Embora não substituam o julgamento humano, os LLMs se mostraram como instrumentos de apoio cognitivo e pedagógico, promovendo maior compreensão dos princípios de UX e estimulando o pensamento crítico sobre a adequação e completude dos ACs. Desse modo, constatou-se que a principal contribuição dos LLMs não está em substituir a análise humana, mas em potencializá-la, oferecendo um ponto de partida explicativo que pode servir de referência para aprimorar outras práticas da ER sob uma perspectiva de UX.

O estudo avançou um passo metodológico significativo ao empregar técnicas de *prompt*, mesmo que se faça necessário considerar ajustes na EP, em especial na configuração dos *few-shots* e na introdução de exemplos negativos para melhoria

da aprendizagem dos modelos. Ainda assim, os resultados indicam que diretrizes qualitativas conhecidas quando incorporadas ao *prompt*, podem ser processadas e estruturadas em formatos compreensíveis por sistemas GenIA, favorecendo análises que estejam menos sujeitas à variabilidade interpretativa. Isso também sugere um potencial para reduzir a subjetividades e apoiar a consistência de avaliações sobre os requisitos de *software*. Dessa forma, o estudo contribui para o avanço teórico ao propor um modelo de sistematização interpretativa em que heurísticas de UX são codificadas como instruções explícitas e exemplificadas em *few-shot learning*, possibilitando uma convergência produtiva entre a ES e GenIA, e até mesmo o campo emergente da Inteligência Artificial explicável (*Explainable Artificial Intelligence* - XAI, em inglês).

No estágio atual, é seguro dizer que o uso de LLMs para este fim deve se restringir à serem exclusivamente reconhecidos como ferramentas assistivas, oferecendo um suporte inicial, mas não substituindo a tomada de decisão dos especialistas e profissionais da área de UX e Produtos em projetos digitais. Do ponto de vista prático e verificável, é certo também afirmar que a abordagem proposta oferece ao ambiente acadêmico um recurso de tutoria automatizada de fácil acesso e usabilidade, sendo incorporada também na aprendizagem de DoR e *Backlog Refinement* didaticamente estruturados. Aos times ágeis do ambiente corporativo, este estudo entende que pode se tornar muito útil da mesma forma, porém se faz necessário realizar um estudo de caso para afirmar fidedignamente. Prontamente, a abordagem apoia atributos como usabilidade, acessibilidade, e percepção de valor geralmente negligenciados em ACs, tendo sua qualidade comprometida à vista do time de desenvolvimento e lançamento do produto.

Este estudo contribui também para a discussão sobre o uso de GenIAs na ER ampliando investigações sistemáticas voltadas à validação automatizada de requisitos focados em UX. A pesquisa oferece uma proposta metodológica replicável, baseada em *prompting* estruturado, e evidencia, com dados consistentes, as potencialidades e limitações dessas tecnologias. Para a prática profissional, os resultados indicam que as GenIAs podem reduzir o esforço manual e qualificar a revisão de ACs, desde que sejam utilizadas sob supervisão de um especialista e com EP cuidadosamente definida. Se espera, portanto, que as recomendações geradas por essas ferramentas, quando supervisionadas, possam otimizar o

refinamento de requisitos e colaborar para a construção de produtos digitais mais coerentes com os princípios de usabilidade, acessibilidade e qualidade percebida.

9.1 Ameaças à Validade do *Survey*

A identificação e a análise de possíveis ameaças à validade são etapas fundamentais em estudos de dissertação, pois permitem avaliar as limitações metodológicas e as condições que podem ter influenciado os resultados obtidos. Para estruturar essa discussão de forma sistemática, este capítulo adota a categorização proposta por Wohlin et al. (2012), amplamente utilizada em pesquisas na área de ES. Essa abordagem possibilita analisar o estudo sob diferentes perspectivas para fortalecer estudos futuros.

Validade Interna

Experiência heterogênea dos participantes: O questionário foi disponibilizado à participação voluntária, e continha perguntas abertas para melhor entendimento sobre os perfis dos participantes. Apesar da maioria dos respondentes declarar experiência prévia com validação de requisitos e tempo de atuação superior a 6 anos na função, a amostra final também incluiu profissionais menos experientes. Essa heterogeneidade pode influenciar a interpretação das perguntas e a percepção sobre a participação do cliente nas sessões de DoR, interferindo na homogeneidade dos dados.

Influência do contexto organizacional: Os perfis dos participantes revelaram diferentes níveis de maturidade em práticas ágeis. Ainda com a diversidade de papéis profissionais, pode introduzir variações na perspectiva com que os participantes lidam com o processo de validação de requisitos. Certamente, essa pluralidade enriqueceu a visão do estudo. Entretanto, pode ter contribuído para respostas baseadas em vivências locais específicas, sem padronização de práticas entre os times representados.

Validade Externa

Restrição geográfica e cultural: As perguntas do *survey* foram elaboradas em dois idiomas considerando coletar respostas de participantes com experiências culturais distintas. A baixa adesão de respondentes internacionais compromete a diversidade cultural esperada e limita a extrapolação dos achados para organizações com práticas e maturidades distintas em outros países ou ecossistemas de desenvolvimento.

Cultura organizacional e maturidade ágil variáveis: Os resultados estão fortemente vinculados à realidade dos times de desenvolvimento ágil em que os respondentes atuam, os quais apresentam níveis distintos de maturidade em relação a DoR e validação contínua de requisitos, o que restringe a aplicabilidade ampla das conclusões.

Validade de Construto

Formulação das questões do questionário: A interpretação subjetiva de algumas perguntas abertas e múltiplas respostas pode ser variável entre os participantes do *survey*, afetando a precisão conceitual dos dados.

Complexidade conceitual dos termos utilizados: Termos como DoR, validação de ACs, e participação do cliente podem assumir significados diferentes para profissionais de áreas distintas ou de níveis variados de maturidade em métodos ágeis. Essa variabilidade pode ter impactado a uniformidade das respostas e a validade de construção.

Avaliação qualitativa das respostas: A categorização das respostas está sujeita à interpretação subjetivas, podendo ocorrer vieses na classificação ou agrupamento temático dos conteúdos.

Validade de Conclusão

Uso de estatística descritiva: Com 31 respondentes, não foi possível aplicar testes estatísticos inferenciais. As análises foram descritivas, o que restringe a força das conclusões. Os resultados devem ser interpretados como indícios iniciais, não generalizações.

Diversidade da avaliação: Os dados obtidos por meio do questionário não foram revisados e validados por outros especialistas da área de Tecnologia após a etapa de coleta. Essa decisão metodológica pode influenciar a forma como os resultados são interpretados, principalmente em relação à coerência das respostas com os construtos analisados. Nesse sentido, é recomendável que, para estudos futuros, sejam incluídas revisões por especialistas para ampliar a diversidade de interpretações, reduzir possíveis vieses individuais e fortalecer a consistência das análises realizadas.

9.2 Ameaças à Validade da Análise de Conteúdo

A condução da análise de conteúdo sobre a aplicabilidade da técnica ACUX, ainda que metodologicamente fundamentada e executada com rigor, está sujeita a determinadas limitações inerentes à natureza exploratória e interpretativa do estudo. Assim, as potenciais ameaças à validade interna, externa, de construto e de conclusão são apresentadas a seguir, conforme os princípios estabelecidos por Wohlin et al. (2012) para estudos empíricos em ES.

Validade Interna

No presente trabalho, uma possível ameaça está relacionada à subjetividade do avaliador humano durante a codificação e categorização dos ACs. Embora o uso das diretrizes ACUX tenha proporcionado um arcabouço sistemático para reduzir vieses interpretativos, a análise de conteúdo, por natureza, envolve juízos qualitativos que podem variar conforme o repertório técnico e a experiência do avaliador.

Além disso, a relação do avaliador humano com o contexto institucional pode ter influenciado a leitura e interpretação de certos trechos dos ACs, ainda que o corpus tenha sido anonimizado. Essas ameaças foram mitigadas por meio da utilização de parâmetros objetivos de codificação, replicação fiel da planilha modelo de Souza (2021) e aplicação de critérios de inclusão e exclusão rigorosamente definidos no plano de análise.

Validade Externa

Por se tratar de uma avaliação exploratória, realizada com amostra restrita de 20 ACs provenientes de projetos acadêmicos de duas turmas, os resultados não podem ser generalizados para outros domínios, instituições ou níveis de maturidade de equipes de desenvolvimento.

Adicionalmente, o fato de os ACs terem sido produzidos em um ambiente educacional supervisionado, e não em contextos corporativos reais, pode limitar a extrapolação dos achados para projetos de *software* em larga escala. Ainda assim, a escolha desse ambiente é justificada por permitir controle metodológico, rastreabilidade documental e coerência com os objetivos de explorar a aplicabilidade da ACUX em estágios iniciais de formulação de requisitos.

Validade de Construto

As diretrizes ACUX foram adotadas como instrumento de análise para identificar a presença ou ausência de aspectos de UX nos ACs. Entretanto, como o ACUX ainda é um *framework* recente e predominantemente validado em ambientes acadêmicos, há risco de que suas categorias não abranjam integralmente a complexidade dos aspectos de UX em requisitos de *software*.

Outro ponto de atenção é que o conceito de “aplicabilidade” da ACUX foi operacionalizado por meio da frequência e distribuição das recomendações por *guideline*, o que pode representar parcialmente o construto pretendido, deixando de capturar dimensões subjetivas, como a relevância percebida de cada recomendação para a experiência do usuário. Tais limitações foram compensadas pela triangulação

teórica com autores clássicos de UX e usabilidade (Nielsen, 1994; Norman, 2013; Hassenzahl, 2010; Garrett, 2011).

Validade de Construto

Como a amostra é reduzida e o estudo privilegia a interpretação qualitativa sobre a quantificação estatística, há risco de superinterpretação dos padrões observados nas matrizes de frequência e mapas de calor. Outro possível viés decorre do uso de métricas descritivas de incidência, que, embora úteis para revelar tendências, não estabelecem relações de causa e efeito entre variáveis.

Para mitigar tais limitações, as conclusões foram delimitadas ao escopo dos dados analisados e interpretadas de forma alinhada ao propósito exploratório do estudo, sem pretensão de generalização estatística. Além disso, a utilização das três etapas clássicas da análise de conteúdo (Bardin, 2016) assegurou rastreabilidade metodológica e coerência na inferência dos resultados.

9.3 Ameaças à Validade do Execução dos LLMs

Todo estudo está sujeito a limitações que podem afetar a credibilidade, a generalização e a interpretação de seus resultados (Wohlin et al., 2012). Diante disso, embora a realização deste estudo, forneça uma análise que contribui para os avanços ao estado da arte envolvida na pesquisa, é essencial reconhecer e abordar potenciais ameaças à validade dos resultados. Este capítulo apresenta as principais ameaças identificadas, organizadas segundo as categorias clássicas propostas por Wohlin et al. (2012).

Validade Interna

Existe a possibilidade de parte do desempenho dos LLMs avaliados considerar mais a qualidade do dado de entrada do que a capacidade técnica dos modelos podem atingir de forma substancial. Como as recomendações foram geradas sequencialmente para cada critério, existe a chance das IAGens adotarem

padrões de resposta recorrentes, sobretudo em critérios semelhantes, o que pode ter favorecido repetições indevidas ou enviesadas.

É válido destacar ainda que, a validação das recomendações e a mensuração do desempenho não foi realizada pelos modelos, mas sim conduzida por um único avaliador humano especializado em UX e ER com o auxílio de planilhas. Essas planilhas operacionalizam a avaliação sobre os resultados das respostas geradas pelos LLMs permitindo sua classificação em categorias binárias (“se aplica” / “não se aplica”) e o posterior cálculo das métricas de concordância, precisão técnica e explicabilidade. Apesar de tecnicamente justificada pelo delineamento do estudo, a escolha por tal metodologia de avaliação pode ter introduzido vieses subjetivos e limitações interpretativas que impactaram a classificação das recomendações como corretas ou indevidas.

Validade Externa

Dada à natureza de condução do estudo envolveu o ambiente acadêmico pela avaliação de projetos dos alunos de graduação, os ACs analisados podem não refletir integralmente a complexidade, os desafios de ambiguidade e as demandas específicas presentes em ambientes corporativos reais. A exclusividade de aceitar apenas ACs modelados no padrão BDD também restringe a aplicabilidade a projetos que não utilizam esse formato. Em adicional, o avaliador humano avaliou apenas dois modelos de LLM, ChatGPT-4o e Gemini 2.5 Flash. Um aspecto limitante que impede a generalização dos achados para outras GenIAs disponíveis no mercado ou para versões futuras dessas mesmas tecnologias, uma vez que o desempenho pode variar em função de atualizações de arquitetura, dados de treinamento e políticas de alinhamento.

Validade de Construto

A definição de concordância, precisão técnica e explicabilidade podem ressoar conclusões e interpretações diversas sobre seu contexto e significados. Por mais que tenham sido formalmente conceituadas, é possível que, para alguns leitores, não capturem todos detalhes qualitativos sobre o que se pretende avaliar

em cada uma delas. Um outro fator a considerar sobre a validade é a dependência da EP para configuração de desempenho dos LLMs quanto à cobertura instrucional esperada. A ausência de exemplos negativos e a limitação na variedade de critérios fornecidos como *few-shots* comprometeram parcialmente a capacidade dos modelos de discernir de forma adequada às situações em que determinadas diretrizes não se aplicavam.

Validade de Conclusão

Foram analisados pelos LLMs 20 ACs, que geraram o total de 344 recomendações de UX válidas perante o avaliador humano. A dimensão dessa amostra pode ser limitada para conclusões estatisticamente robustas ou para a detecção de padrões menos frequentes.

A avaliação de desempenho sobre as respostas dos LLMs se baseiam em métricas proporcionais (médias, percentuais e desvios-padrão), sem a aplicação de testes inferenciais, como intervalos de confiança ou testes de correlação, que poderiam reforçar a confiabilidade estatística dos resultados. Portanto, o estudo não afirma que as diferenças observadas entre os modelos sejam estatisticamente significativas ou generalizáveis a outros contextos. Outro item a considerar é que a avaliação sobre métrica de explicabilidade não reflete critérios quantitativos ou automatizados de pontuação de legibilidade, análise léxica ou métricas computacionais. Essa métrica foi avaliada com base em percepções de clareza, suficiência de detalhes e adequação das justificativas. Assim, a avaliação dependeu da interpretação do avaliador, o que, embora coerente com o propósito exploratório do estudo, pode introduzir variações perceptivas entre diferentes avaliadores.

9.4 Limitações e Trabalhos Futuros

Mesmo que de certo modo os objetivos tenham sido alcançados, existem algumas limitações inerentes ao escopo da pesquisa e devem ser consideradas para realização de trabalhos futuros. A proposta final deste trabalho está limitada a uma abordagem assistiva de LLMs orientados por *guidelines* de UX sistematicamente especificadas.

Para melhoria do desempenho dos modelos, este estudo recomenda um novo ciclo de versão para revisar e aprimorar os *prompts few-shot*, incorporando exemplos que abordam explicitamente diretrizes estruturais e visuais, como EV-02, DI-03 e EV-06. Com essa medida se pretende reduzir lacunas na identificação dessas *guidelines*, observadas na análise. Sugere-se ainda explorar um mecanismo automatizado de validação cruzada que compare as recomendações feitas pelos dois LLMs, ChatGPT-4o e Gemini 2.5 Flash, para um mesmo AC, verificando se há concordância suficiente entre elas, ou seja, se ambas recomendam as mesmas *guidelines* ou justificativas próximas. Outro ponto a considerar na análise comparativa dos LLMs é que, embora os resultados indiquem correlação positiva entre as avaliações humanas e as recomendações das IAs, a plena integração dessas tecnologias em ambientes corporativos ainda exigirá ajustes relativos à privacidade de dados, governança e auditabilidade das respostas.

Para trabalhos futuros, recomenda-se a ampliação da amostra e a replicação operacional dos LLMs com outros tipos de requisitos e diferentes arquiteturas dos modelos, de modo a aprofundar a compreensão sobre a consistência e a generalização das análises. Pesquisas subsequentes também poderiam explorar a colaboração entre múltiplos avaliadores, humanos e artificiais, em um mesmo processo de validação, examinando a formação de consenso e a complementaridade entre as perspectivas. Outras vertentes possíveis incluem o desenvolvimento de métricas híbridas que combinem indicadores quantitativos e semânticos, bem como a criação de ferramentas de suporte que tornem os resultados das análises mais interpretáveis para os usuários não técnicos.

9.5 Considerações Finais

Em suma, este estudo demonstrou que é possível empregar LLMs de forma sistematizada e explicável para apoiar a validação de ACs baseados em UX, contribuindo tanto para o avanço teórico da ER quanto para o aprimoramento prático do processo de desenvolvimento de *software*. As evidências apresentadas confirmam que o uso de LLMs, quando orientado por princípios claros e métricas adequadas, não apenas amplia a qualidade técnica das análises, mas também promove uma abordagem mais reflexiva, colaborativa e centrada no usuário. Desse modo, a pesquisa reforça a tese de que a convergência entre raciocínio humano e

inteligência artificial pode ser conduzida de maneira complementar, ética e construtiva, pavimentando o caminho para novos modelos de validação de requisitos no contexto da ES moderna.

10 Referências

AHMED, Toufique et al. Can LLMs Replace Manual Annotation of Software Engineering Artifacts? In: INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES (MSR), 22., 2025. New York: IEEE/ACM, 2025. DOI: 10.1109/MSR66628.2025.00086. Disponível em: https://ieeexplore.ieee.org/document/11025652. Acesso em: 16 jun. 2025.

ALDAVE, Ainhoa et al. Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. The Journal of Systems & Software, Amsterdam, v. 157, p. 110408, nov. 2019. DOI: 10.1016/j.jss.2019.110396. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0164121219301712?via%3Dihub. Acesso em: 25 nov. 2024.

ANANJEVA, Alisa; PERSSON, John Stouby; BRUUN, Anders. Integrating UX work with agile development through user stories: an action research study in a small software company. Journal of Systems and Software, v. 170, p. 110785, 2020.

ANDERSON, John R.; CORBETT, Albert T.; KOEDINGER, Kenneth R.; PELLMAN, Philip. Cognitive Tutors: Lessons Learned. The Journal of Learning Sciences, v. 4, n. 2, p. 167–207, 1995.

ARTIFICIAL ANALYSIS. Gemini 2.5 Flash (Reasoning): Intelligence, Performance & Price Analysis. [S. l.]: Artificial Analysis, 2025a. Disponível em: https://artificialanalysis.ai/models/gemini-2-5-flash-reasoning. Acesso em: 03 mai. 2025.

ARTIFICIAL ANALYSIS. GPT-4o (ChatGPT): Intelligence, Performance & Price Analysis. [S. l.]: Artificial Analysis, February, 2025b. Disponível em: https://artificialanalysis.ai/models/gpt-4o-chatgpt. Acesso em: 08 mar. 2025.

ANTIN, J. The UX Research Reckoning is Here. Medium: One Big Thought, 2024.

BALTES, S. et al. Guidelines for Empirical Studies in Software Engineering involving LLMs. In: LLM Guidelines Org, 2025. Disponível em: https://llm-guidelines.org/. Acesso em: 15 dez. 2024.

BARDIN, Laurence. Análise de conteúdo. 1. ed. rev. e ampl. São Paulo: Edições 70, 2016.

BARTHE, Guy; GOMEZ, Iosu Mendialdua; REICH, Johannes. Analysis of LLMs vs Human Experts in Requirements Engineering. arXiv preprint arXiv:2501.19297, 2024. Disponível em: https://arxiv.org/pdf/2501.19297. Acesso em: 06 fev. 2025.

BECK, K. et al. Manifesto for Agile Software Development. 2001.

BISWAS, Som. Role of chatGPT in Law: According to chatGPT. SSRN Electronic Journal, 31 mar. 2023. DOI: 10.2139/ssrn.4405398.

BOMMASANI, Rishi et al. On the Opportunities and Risks of Foundation Models. Stanford: Center for Research on Foundation Models (CRFM), Stanford University, 2021. 222 p. Disponível em: https://crfm.stanford.edu/report.html. Acesso em: 04 dez. 2024.

BROCKENBROUGH, Allan; FEILD, Henry; SALINAS, Dominic. Exploring LLMs Impact on Student-Created User Stories and Acceptance Testing in Software Development. In: SIGCSE TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 56., 2025, Pittsburgh, PA. New York: ACM, 2025. p. 1395–1396. DOI: 10.1145/3641555.3705183. Disponível em: https://dl.acm.org/doi/10.1145/3641555.3705183. Acesso em: 25 fev. 2025

BROWN, Tom B. et al. Language Models are Few-Shot Learners. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), Vancouver: ACM, 2020. p. 1877–1901. Disponível em: https://dl.acm.org/doi/abs/10.5555/3495724.3495883. Acesso em: 10 fev. 2025.

CAGAN, Marty. Inspired: How to Create Tech Products Customers Love. 2. ed. Hoboken: Wiley, 2018.

CAGLAYAN, Ozan; MADHYASTHA, Pranava; SPECIA, Lucia. Curious Case of Language Generation Evaluation Metrics: A Cautionary Tale. In: COLING 2020 – Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), 2020. Disponível em: https://aclanthology.org/2020.coling-main.210/. Acesso em: 14 dez. 2024.

CHAMBERLAIN, S.; SHARP, H.; BARCLAY, S. Towards a framework for Agile requirements engineering. In: INTERNATIONAL WORKSHOP ON AGILE REQUIREMENTS ENGINEERING, 2006, Minneapolis: IEEE, 2006. p. 1–8. DOI: 10.1109/AIRE.2006.3.

CHEN, Lingjiao; ZAHARIA, Matei; ZOU, James. How Is ChatGPT's Behavior Changing Over Time? Harvard Data Science Review, Cambridge, MA, v. 6, n. 2, 2024. Disponível em: https://hdsr.mitpress.mit.edu/pub/y95zitmz/release/2. Acesso em: 22 mar. 2025.

COHEN, Jacob. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, v. 20, n. 1, p. 37–46, 1960.

COHN, Mike. User Stories Applied: For Agile Software Development. Boston: Addison-Wesley, 2004.

DAKHEL, A. A. et al. Towards understanding the use of large language models in software engineering: preliminary findings and open problems. Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, 2023.

DALPIAZ, Fabiano et al. Natural language processing for requirements engineering: The best is yet to come. IEEE Software, New York, v. 35, n. 5, p. 115–119, 2018.

DEBNATH S., Spoletini P., and Ferrari A. “From ideas to expressed needs: an empirical study on the evolution of requirements during elicitation,” in 2021 IEEE 29th International Requirements Engineering Conference (RE), 2021.

DENGER, Christian; OLSSON, Thomas. Quality Assurance in Requirements Engineering. In: AURUM, A.; HANDOLL, J. (ed.). Engineering and Managing Software Requirements. Berlim: Springer, 2005. p. 163–185.

DEVLIN, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: ANNUAL CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (NAACL), 2019, Minneapolis: ACL, 2019. p. 4171–4186.

TOPUZ, Kazim et al. Interpretable machine learning and explainable artificial intelligence. Annals of Operations Research, New York, 27 mar. 2025. Prefácio. DOI: 10.1007/s10479-025-06577-w.

ELAZAR, Yanai; KASSNER, Nora; RAVFOGEL, Shauli; RAVICHANDER, Abhilasha; HOVY, Eduard; SCHÜTZE, Hinrich; GOLDBERG, Yoav. Measuring and improving consistency in pretrained language models. Transactions of the Association for Computational Linguistics, v. 9, 2021.

FATHIN NAJWA BINTI MUSTAFFA, S. N. et al. Enhancing high-quality user stories with aqusa: an overview study of data cleaning process. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING & COMPUTER SYSTEMS AND INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND INFORMATION MANAGEMENT (ICSECS-ICOCSIM), 2021, [S. l.]: IEEE, 2021. p. 295–300. Disponível em: https://doi.org/10.1109/ICSECS52883.2021.00060. Acessado em: 19 nov. 2024.

KARLSSON, F.; CHATZIPETROU, P.; GAO, S.; HAVSTORM, T. E. How reliable are GPT-4o and LLAMA3.3-70B in classifying natural language requirements?: The impact of the temperature setting. IEEE Software, v. 42, n. 6, Nov./Dec. 2025. DOI: 10.1109/MS.2025.3572561. Disponível em: https://ieeexplore.ieee.org/document/11012694. Acesso em: 26 mai. 2025.

GARCIA, A.; SILVA, T. S. da; SILVEIRA, M. S. Artifact-facilitated communication in agile user-centered design. In: SPRINGER. International Conference on Agile Software Development. [S.l.], 2019.

GARCIA, A.; SILVA, T. Silva da; SILVEIRA, M. S. Artifacts for agile user-centered design: a systematic mapping. In: Proceedings of the 50th Hawaii International Conference on System Sciences. [S.l.: s.n.], 2017.

GARRETT, Jesse James. The Elements of User Experience: User-Centered Design for the Web and Beyond. Berkeley: New Riders, 2011.

GÉNOVA, Gonzalo et al. A framework to Measure and Improve the Quality of Textual Requirements. Requirements Engineering, London, v. 18, 2013.

GOOGLE DEEPMIND. Gemini 2.5 Flash Model Card. [S. l.]: Google DeepMind, 17 jun. 2025a. Disponível em: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf. Acesso em: 19 jun. 2025.

GOOGLE DEEPMIND. Gemini 2.5 Technical Report. 2025b. Disponível em: https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf. Acesso em: 19 jun. 2025.

GOTHELF, Jeff; SEIDEN, Josh. Lean UX: applying lean principles to improve user experience. 2. ed. Sebastopol: O'Reilly Media, 2016.

HASSENZAHL, Marc; PLATZ, Axel; BURMESTER, Michael; LEHNER, Katrin. Hedonic and ergonomic quality aspects determine a software's appeal. In: ACM SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI '00), 2000, Haia, Holanda (assumido). [S. l.]: ACM, 2000. p. 201-208. Disponível em: https://doi.org/10.1145/332040.332432. Acesso em: 06 out. 2024.

HASSENZAHL, Marc. Experience design: technology for all the right reasons. San Rafael: Morgan & Claypool, 2010.

HSU H.-Y., Hsu K.-C., Hou S.-Y., Wu C.-L., Hsieh Y.-W., Cheng Y.-D. et al., “Examining real-world medication consultations and drug-herb interactions: chatgpt performance evaluation,” JMIR Medical Education, vol. 9, no. 1, 2023.

IEEE. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998. New York, NY: IEEE, 1998. 40 p.

INAYAT, Irum; SALIM, S. S.; MARCZAK, S.; DANEVA, M.; SHAMSHIRBAND, S. A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior, v. 51, p. 915–929, 2015. DOI: https://doi.org/10.1016/j.chb.2014.10.046. Acessado em: 06 abr. 2025.

ISO-9241-210, .-. Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. Ergonomics of human-system interaction., v. 2019, p. 33, 2019.

JEFFRIES, Ron E.; ANDERSON, Ann; HENDRICKSON, Chet. Extreme Programming Installed. Boston: Addison-Wesley Longman Publishing Co., Inc., 2000.

JURENKA, I. et al. Towards responsible development of generative AI for education: An evaluation-driven approach. arXiv: 2407.12687, 2024.

JURAFSKY, Daniel; MARTIN, James H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3. ed.

Draft de 2024. Stanford / Boulder: Jurafsky & Martin, 2024. Disponível em: https://stanford.edu/~jurafsky/slp3/ . Acesso em: 10 jan. 2025.

JURISCH, Matthias et al. Evaluating a recommendation system for user stories in mobile enterprise application development. International Journal on Advances in Intelligent Systems, vol. 10, n. 1 & 2, p. 25-34, 2017. Disponível em: https://personales.upv.es/thinkmind/dl/journals/intsys/intsys_v10_n12_2017/intsys_v10_n12_2017_4.pdf . Acesso em: 16 fev. 2025.

KASHFI, A.; LAW, E.L.-C.; ROTO, V. et al. Integrating User eXperience practices into software development processes: implications of the UX characteristics. PeerJ Computer Science, v. 3, e130, 2017. doi:10.7717/peerj-cs.130

KASSAB, Mohamad. An empirical study on the requirements engineering practices for agile software development. In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS, 40., 2014, Verona, Itália. [S. l.]: IEEE, 2014. p. 254-261. Disponível em: https://ieeexplore.ieee.org/document/6928819. Acesso em: 02 dez. 2024.

KATZ, Daniel Martin; BOMMARITO, Michael James; GAO, Shang; ARREDONDO, Pablo. GPT-4 passes the bar exam. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, v. 382, n. 2270, Article 20230254, 26 fev. 2024. DOI: 10.1098/rsta.2023.0254.

KAUR, Davinder; USLU, Suleyman; DURRESI, Arjan. Requirements for Trustworthy Artificial Intelligence - A Review. In: ADVANCES IN NETWORKED-BASED INFORMATION SYSTEMS: THE 23RD INTERNATIONAL CONFERENCE ON NETWORK-BASED INFORMATION SYSTEMS (NBiS-2020), 23., 2020. Springer, 2021.

KENDRO, K.; MALONEY, J.; JARVIS, S. Lexical diversity in human- and LLM-generated text. In: PROCEEDINGS OF THE ANNUAL MEETING OF THE COGNITIVE SCIENCE SOCIETY, 46., 2024, Disponível em: https://escholarship.org/uc/item/18n5k7c6. Acessado em: 06 abr. 2025.

KICI, D. et al. A BERT-based transfer learning approach to text classification on software requirements specifications. In: CANADIAN CONFERENCE ON AI, 2021. [S.l.: s.n.], 2021. v. 1, p. 042077.

KNAPPE, T. et al. Semantic Self-Consistency: Enhancing Language Model Reasoning via Semantic Weighting. [S.l.: s.n.], 2025.

KNIBERG, H.; IVARSSON, A. Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. [S.l.]: [s.n.], out. 2012.

KOLTHOFF, Kristian; KRETZER, Felix; BARTELT, Christian; MAEDCHE, Alexander; PONZETTO, Simone Paolo. Interlinking User Stories and GUI Prototyping: A Semi-Automatic LLM-based Approach. In: Proceedings of the 32nd IEEE International Requirements Engineering Conference (RE 2024), Reykjavík, Iceland, 24–28 Jun. 2024. Piscataway: IEEE, 2024. p. 380–388. DOI: 10.1109/RE59067.2024.00045.

KRISHNA, Madhava et al. Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In: INTERNATIONAL CONFERENCE ON ADVANCEMENTS IN AI IN SOFTWARE ENGINEERING (ICAISE), 4., 2024. New York: IEEE, 2024. DOI: 10.1109/ICAISE60599.2024.10628461. Disponível em: https://ieeexplore.ieee.org/document/10628461. Acessado em: 18 mai. 2024.

KRIPPENDORFF, Klaus. Content analysis: An introduction to its methodology. 3. ed. Thousand Oaks: Sage Publications, 2018.

LAI, Viet Dac et al. ChatGPT Beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP). Singapore: Association for Computational Linguistics (ACL), 2023. p. 13171–13189. Disponível em: https://aclanthology.org/2023.findings-emnlp.878/. Acessado em: 01 jul. 2024.

LAW, E. L.-C.; ROTO, V.; HASSENZAHL, M.; VERMEEREN, A.; KORT, J. Understanding, scoping and defining user experience: a survey approach. Proceedings of the 27th International Conference on Human Factors in Computing Systems – CHI 2009, p. 719–728, 2009. doi:10.1145/1518701.1518813

LEE, Harrison et al. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 41., 2024, Vienna, Áustria. New York: ACM/JMLR.org, 2024. DOI: 10.5555/3692070.3693141.

LEFFINGWELL, Dean; KNASTER, Richard. SAFe® 4.0 Distilled: Applying the Scaled Agile Framework® for Lean Software and Systems Engineering. Addison-Wesley Professional, 2016.

LI, Shuai et al. Beyond Chain-of-Thought: A Survey of Chain-of-X Paradigms for LLMs. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS (COLING), 31., 2025. [S.l.]: Association for Computational Linguistics (ACL), 2025. Disponível em: https://aclanthology.org/2025.coling-main.719.pdf. Acesso em: 10 mai. 2025.

LIANG, P. et al. Holistic Evaluation of Language Models. Transactions on Machine Learning Research (TMLR), [S.l.]: OpenReview, 2023. Disponível em: https://openreview.net/forum?id=iO4LZibEqW . Acesso em: 23 nov. 2024.

LIBBRECHT P, Declerck T, Schlippe T, Mandl T, Schiffner D. NLP for Student and Teacher: Concept for an AI based Information Literacy Tutoring System. In: Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020). CEUR Workshop Proceedings 2699, CEUR-WS.org. Galway, Ireland: 2020 October 19-23.

LIU P., Yuan W., Fu J., Jiang Z., Hayashi H., and Neubig G., “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” ACM Computing Surveys, vol. 55, no. 9, 2023.

LIU Y., Han T., Ma S., Zhang J., Yang Y., Tian J., He H., Li A., He M., Liu Z. et al., “Summary of chatgpt-related research and perspective towards the future of large language models,” Meta-Radiology, 2023.

LOPES, L. A.; PINHEIRO, E. G.; SILVA, T. S.; ZAINA, L. A. M. Requisitos de usabilidade para softwares aplicados ao e-learning: uma proposta para elaboração de User Stories. In: Findings of the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE 2019), 2019, DOI: 10.5753/cbie.sbie.2019.112.

LOSADA, B. Flexible requirement development through user objectives in an agile-ucd hybrid approach. In: Proceedings of the XIX International Conference on Human Computer Interaction. [S.l.: s.n.], 2018.

LU, Wen-Yen; FAN, Szu-Chun. Developing a weather prediction project-based machine learning course in facilitating AI learning among high school students. Computers and Education: Artificial Intelligence, v. 5, p. 100154, 2023. DOI: 10.1016/j.caeai.2023.100154.

LUCASSEN, G.; DALPIAZ, F.; WERF, J. M. E. V. D.; BRINKKEMPER, S. Forging high-quality user stories: towards a discipline for agile requirements. In: IEEE. 2015 IEEE 23rd international requirements engineering conference (RE). [S.l.], 2015.

LUCASSEN, G., Dalpiaz, F., Werf, J. M., & Brinkkemper, S. (2016a). The use and effectiveness of user stories in practice. In Proceedings of the 22nd international working conference on requirements engineering: Foundation for software quality. https://doi.org/10.1007/978-3-319-30282-9_14. Acessado em: 17 abr. 2024.

LUCASSEN, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2016b). Improving agile requirements: The quality user story framework and tool. Requirements Engineering, 21, 383–403. https://doi.org/10.1007/s00766-016-0250-x. Acessado em: 17 abr. 2024.

MEDEIROS, Juliana; VASCONCELOS, Alexandre Marcos Lins de; SILVA, Carla; GOULÃO, Miguel. Requirements Specification for Developers in Agile Projects: Evaluation by two Industrial Case Studies. Information and Software Technology, v. 117, 2019. DOI: 10.1016/j.infsof.2019.106194.

MERGULHÃO, P.; LENCASTRE, M.; SOARES, M.; ALMEIDA, R.; BARBOSA, A. Uso de metodologias criativas no processo de ensino da disciplina engenharia de requisitos. In: WER19 - Workshop em Engenharia de Requisitos. [S.l.: s.n.], 2019.

MISHRA, Swaroop et al. Reframing Instructional Prompts to GPTk's Language. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL), 60., 2022. Dublin: ACL Anthology, 2022. p. 612-622. Disponível em: https://aclanthology.org/2022.findings-acl.50/. Acessado em: 14 set. 2024.

MONDILLO, G. et al. Basal knowledge in the field of pediatric nephrology and its enhancement following specific training of ChatGPT-4 "omni" and Gemini 1.5 Flash. Pediatric Nephrology, v. 40, n. 1, p. 151-157, Jan. 2025. DOI: 10.1007/s00467-024-06486-3.

NEBEKER, C.; TOROUS, J.; BARTON, C. The Value of User Experience Research in Digital Health. Journal of the American Medical Informatics Association, v. 26, n. 12, p. 2479–2487, 2019. DOI: 10.1093/jamia/ocz217.

NIELSEN, Jakob; BUDWEITZ, Rolf Molich. Usability Inspection Methods. New York: John Wiley & Sons, 2018.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI '90), 1990, Seattle. New York: ACM Press, 1990. p. 249-256. DOI: 10.1145/97243.97281.

NIELSEN, Jakob. Usability Engineering. San Francisco: Morgan Kaufmann, 1994.

NKAMBOU, R.; BOURDEAU, J.; MIZOGUCHI, R. (Ed.). Advances in Intelligent Tutoring Systems. Berlin: Springer, 2010. (Studies in Computational Intelligence, v. 308). DOI: 10.1007/978-3-642-14363-2.

NORMAN, Donald A. The Design of Everyday Things. Revised and Expanded Edition. New York: Basic Books, 2013.

NORMAN, Don; NIELSEN, Jakob. The definition of user experience (UX). Nielsen Norman Group, 1998. Disponível em: https://www.nngroup.com/articles/definition-user-experience/. Acesso em: 14 jul. 2024.

NORTH, D. Introducing behaviour driven development. Better Software Magazine, 2006.

OPENAI. GPT-4o System Card. San Francisco, CA: OpenAI, 8 ago. 2024a. Disponível em: https://openai.com/pt-BR/index/gpt-4o-system-card/. Acessado em: 05 dez. 2024.

OPENAI. GPT-4 Technical Report. [S. l.]: OpenAI, 15 mar. 2023. Disponível em: https://cdn.openai.com/papers/gpt-4.pdf. Acessado em: 08 dez. 2024.

OPENAI. Learning to Reason with LLMs. OpenAI Research, 2024b. Disponível em: https://openai.com/index/learning-to-reason-with-llms. Acessado em: 05 dez. 2024.

OUYANG, Long et al. Training Language Models to Follow Instructions with Human Feedback. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), 36., 2022. [S.l.]: OpenReview/NeurIPS, 2022. p. 27730–27744. Disponível em: https://dl.acm.org/doi/10.5555/3600270.3602281. Acessado em: 27 jan. 2025.

PEÑA VEITÍA, Francisco J. et al. User stories identification in software's issues records using natural language processing. In: IEEE CONGRESO BIENAL DE ARGENTINA (ARGENCON), 2020, [S. l.]: IEEE, 2020. p. 1-7. Disponível em: https://doi.org/10.1109/ARGENCON49523.2020.9505355. Acesso em: 29 abr. 2024.

PICHLER, Roman. Agile product management with Scrum: creating products that customers

love. Boston: Addison-Wesley, 2010.

PREECE, Jenny; ROGERS, Yvonne; SHARP, Helen. Design de interação: além da interação homem-computador. 3. ed. Porto Alegre: Bookman, 2015.

PRESSMAN, Roger S.; MAXIM, Bruce R. Software engineering: a practitioner's approach. 8. ed. New York: McGraw-Hill Education, 2016.

PRESSMAN, R. S.; MAXIM, B. R. Software engineering: a practitioner's approach. 9. ed. New York: McGraw-Hill Education, 2020.

RAJ, Harsh; ROSATI, Domenic; MAJUMDAR, Subhabrata. Measuring reliability of large language models through semantic consistency. Disponível em https://doi.org/10.48550/arXiv.2211.05853. Acessado em: 02 jun. 2024.

REYNOLDS, Laria; MCDONELL, Kyle. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI EA), 2021, Yokohama, Japão. Extended Abstracts... New York: ACM, 2021. p. 1-7. Artigo 314. DOI: 10.1145/3411763.3451760. Disponível em: https://dl.acm.org/doi/10.1145/3411763.3451760. Acessado em: 15 jul. 2024.

RONANKI, Krishna; BERGER, Christian; HORKOFF, Jennifer. Investigating ChatGPT's Potential to Assist in Requirements Elicitation Processes. In: INTERNATIONAL WORKSHOP ON AI AND SOFTWARE ENGINEERING (AISE), 4., 2023, Melbourne. New York: IEEE, 2023. p. 1–8. DOI: 10.1109/AISE59670.2023.10371698. Disponível em: https://ieeexplore.ieee.org/document/10371698. Acessado em: 16 jun. 2024.

SAHOO, Pranab; SINGH, Ayush Kumar; SAHA, Sriparna; JAIN, Vinija; MONDAL, Samrat; CHADHA, Aman. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. 2024.

SALLAM, Malik. ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns. Healthcare, v. 11, n. 6, p. 887, 2023. DOI: 10.3390/healthcare11060887. Disponível em: https://doi.org/10.3390/healthcare11060887
. Acesso em: 12 abr. 2024.

SANH, Victor et al. Multitask Prompted Training Enables Zero-Shot Task Generalization. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS (ICLR), 10., 2022. [S.l.]: OpenReview/ICLR, 2022. Disponível em: https://iclr.cc/virtual/2022/poster/7101. Acessado em: 29 nov. 2024.

SCHÖN, Eva-Maria; THOMASCHEWSKI, Jörg; ESCALONA, María José. Agile requirements engineering: A systematic literature review. Computer Standards & Interfaces, Elsevier, v. 49, 2017. Disponível em: https://doi.org/10.1016/j.csi.2016.08.011 . Acesado em: 11 abr. 2024.

SHARMA, Abhilash; KUMAR TRIPATHI, Anand. Evaluating user story quality with LLMs: a comparative study. Journal of Intelligent Information Systems, v. 63, n. 4, p. 1423-1451, 2025. Disponível em: https://link.springer.com/article/10.1007/s10844-025-00939-3. Acessado em: 28 abr. 2025.

SHARP, Helen; ROGERS, Yvonne; PREECE, Jenny. Interaction Design: Beyond Human-Computer Interaction. 5. ed. Hoboken: Wiley, 2019.

SHULL, F.; SINGER, J.; SJØBERG, D. I. Guide to advanced empirical software engineering. [S.l.]: Springer, 2007.

SIVARAJKUMAR, S.; KELLEY, M.; SAMOLYK-MAZZANTI, A.; VISWESWARAN, S.; WANG, Y. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: algorithm development and validation study. JMIR Medical Informatics, v. 12, e55318, 8 abr. 2024. DOI: 10.2196/55318. Disponível em: https://medinform.jmir.org/2024/1/e55318. Acessado em: 17 jul. 2024.

SOMMERVILLE, Ian. Software Engineering. 10. ed. Boston: Pearson, 2011.

SOUZA, Jonathan Henrique Jeremias. ACUX: Um Guia para Escrita de Aspectos de UX em Critérios de Aceitação de User Stories. 2021. 185 f. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de São Carlos, São Carlos, 2021. Orientadora: Prof.ª Dr.ª Luciana Aparecida Martinez Zaina.

SOUZA, Jonathan H. J.; MARQUES, Leonardo C.; CONTE, Tayana U.; ZAINA, Luciana A. M.. Descrevendo requisitos de User eXperience em Critérios de Aceitação de User Stories. In: 23th Workshop on Requirements Engineering (WER2020), Brazil, 2020. 10.29327/1298730.23-14

SREEKAR, P. Aditya et al. AXCEL: Automated eXplainable Consistency Evaluation using LLMs. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP), Miami, FL: Association for Computational Linguistics (ACL), 2024. p. 14943–14957. Disponível em: https://aclanthology.org/2024.findings-emnlp.878/. Acessado em: 20 jun. 2024.

STANFORD UNIVERSITY. Holistic Evaluation of Language Models (HELM), v0.3.0, Stanford, CA: Stanford University, Center for Research on Foundation Models, 2023. Disponível em: https://crfm.stanford.edu/helm/classic/v0.3.0/. Acessado em: 16 jun. 2024.

STRAUSS, A.; CORBIN, J. Basics of qualitative research techniques. [S.l.]: Sage publications Thousand Oaks, CA, 1998.

UNIÃO EUROPEIA. COMISSÃO EUROPEIA. Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts. Bruxelas: EUR-Lex, 21 abr. 2021. Disponível em: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52021PC0206 . Acessado em: 16 jun. 2024.

UNIÃO EUROPEIA. PARLAMENTO EUROPEU. EU AI Act: First regulation on artificial intelligence. [S.l.: s.n.], [2023]. Disponível em: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence. Acessado em: 16 jun. 2024.

VASWANI, Ashish et al. Attention Is All You Need. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31., 2017, Long Beach, CA: NIPS, 2017. p. 5998–6008. Disponível em: https://neurips.cc/virtual/2017/poster/9372 . Acessado em: 07 jun. 2024.

VIRVOU M, Moundridou M. Adding an instructor modelling component to the architecture of ITS authoring tools. International Journal of Artificial Intelligence in Education. 2001; 12(2): 185-211.

VIRVOU M. Artificial Intelligence and User Experience in reciprocity: Contributions and state of the art. Intelligent Decision Technologies. 2023;17(1):73-125. doi:10.3233/IDT-230092

VIRVOU, D Maras M. Error diagnosis in an English tutor. In: Bullinger HJ, Ziegler J, editors. Human-Computer Interaction: Communication, Cooperation, and Application Design, Proceedings of HCI International ’99 (the 8th International Conference on Human-Computer Interaction). Munich, Germany: Lawrence Erlbaum; 1999 August 22-26. vol. 2, ISBN 0-8058-3392-7.

WAGNER, S. et al. Towards Evaluation Guidelines for Empirical Studies Involving LLMs. In: IEEE/ACM INTERNATIONAL WORKSHOP ON METHODOLOGICAL ISSUES WITH EMPIRICAL STUDIES IN SOFTWARE ENGINEERING (WSESE), 2025, Ottawa, ON, Canada, IEEE, 2025. DOI: 10.1109/WSESE66602.2025.00011.

WAKE, Bill. INVEST in Good Stories, and SMART Tasks. 2003. XP123 blog. Disponível em: https://xp123.com/invest-in-good-stories-and-smart-tasks/. Acessado em: 15 ago. 2024.

WALSH, M.; SCHULKER, D.; LAU, S. Beyond Capable: Accuracy, Calibration, and Robustness in Large Language Models. SEI Blog, 2024. Disponível em: https://doi.org/10.58012/cred-2j82. Acessado em: 18 dez. 2024.

WANG, Haojin; ZHU, Zining; SHI, Freda. Distribution Prompting: Understanding the Expressivity of Language Models Through the Next-Token Distributions They Can Produce. arXiv preprint arXiv:2505.12244, 2025.

WANG, X.; ZHAO, L.; WANG, Y.; SUN, J. The role of requirements engineering practices in agile development: an empirical study. In: Requirements Engineering. [S.l.]: Springer, 2014.

WEI, Jason; WANG, Xuezhi; SCHUURMANS, Dale; BOSMA, Maarten; ZHOU, Sharan Narang; ALVAREZ, Pablo; CHEN, Zhifeng; LE, Quoc V.; ZHOU, Denny. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Advances in Neural Information Processing Systems (NeurIPS 2022). Proceedings [...]. [S.l.]: Curran Associates, 2022.

WHITE, Jules et al. ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. Nashville: Distributed Object Computing Group, Vanderbilt University, 2023a. Relatório Técnico. Disponível em: https://www.dre.vanderbilt.edu/~schmidt/PDF/prompt-patterns-book-chapter.pdf. Acessado em: 25 abr. 2024.

WHITE, Jules et al. A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. In: CONFERENCE ON PATTERN LANGUAGES OF PROGRAMS (PLoP), 30., 2023. New York: ACM, 2023b. p. 1–25. DOI: 10.5555/3721041.3721046. Disponível em: https://dl.acm.org/doi/10.5555/3721041.3721046. Acessado em 13 mai. 2024.

WOHLIN, Claes et al. Experimentation in Software Engineering. 1. ed. Heidelberg: Springer, 2012. DOI: https://doi.org/10.1007/978-3-642-29044-2. Acessado em: 22 set. 2024.

WOOLF, Beverly P. Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutionizing e-Learning. Burlington, MA: Morgan Kaufmann Publishers, 2010.

XU, Y.; et al. Evaluating Large Language Models on Non-Code Software Engineering Tasks. arXiv preprint arXiv:2501.19297, 2024.

YAMANI, Asma; BASLYMAN, Malak; AHMED, Moataz. Leveraging LLMs for User Stories in AI Systems: UStAI Dataset. New York: ACM, 2025. p. 1–11. DOI: 10.1145/nnnnnnn.nnnnnnn.

YANG, J. et al. Enhancing Semantic Consistency of Large Language Models through Model Editing: An Interpretability-Oriented Approach. [S.l.: s.n.], 2024.

YANG, Y. et al. A survey on deep learning for software engineering. ACM Computing Surveys (CSUR), New York, v. 54, n. 10s, p. 1–73, 2022.

YU, Zihan; HE, Liang; WU, Zhen; DAI, Xinyu; CHEN, Jiajun. Towards Better Chain-of-Thought Prompting Strategies: A Survey. arXiv preprint, out. 2023. Disponível em: https://arxiv.org/abs/2310.04959. Acessado em: 05 set. 2024.

ZAINA, Luciana A. M.; SHARP, Helen; BARROCA, Leonor. Informações de UX no trabalho diário de uma equipe ágil: uma análise baseada em cognição distribuída. In: XIX Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC), Diamantina, 2022. Porto Alegre: SBC, 2022. p. 239–240. DOI: 10.5753/ihc_estendido.2022.225375

ZHANG, A. et al. The AI Index 2023 Annual Report. Stanford Institute for Human-Centered AI, Stanford University, 2023.

ZHANG, Tianyi; KISHORE, Varsha; WU, Felix; WEINBERGER, Kilian Q.; ARTZI, Yoav. BERTScore: Evaluating Text Generation with BERT. In: Proceedings of the 8th International Conference on Learning Representations (ICLR 2020). Addis Ababa, Etiópia, 2020.

ZHANG, Zheying et al. LLM-Based Agents for Automating the Enhancement of User Story Quality: An Early Report. In: AGILE PROCESSES IN SOFTWARE ENGINEERING AND

EXTREME PROGRAMMING (XP 2024). Cham: Springer Nature Switzerland, 2024. p. 117–126. Disponível em: https://link.springer.com/chapter/10.1007/978-3-031-61154-4_8. Acessado em: 29 jun. 2024.

ZHAO, Liping et al. Natural Language Processing for Requirements Engineering: A Systematic Mapping Study. ACM Computing Surveys (CSUR), New York, v. 54, n. 3, 2021.

ZHAO, Wayne Xin et al. Large Language Models. Singapore: Springer Nature Singapore, 2024. 237 p. ISBN 978-981-96-6258-6.

ZHENG, Lianmin et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), 37., 2023, Vancouver: OpenReview/NeurIPS, 2023. Disponível em: https://neurips.cc/virtual/2023/poster/73434. Acessado em: 08 ago. 2024.

ZHOU, Yongchao et al. Large Language Models Are Human-Level Prompt Engineers. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), 36., 2022. [S.l.]: OpenReview/NeurIPS, 2022. Disponível em: https://neurips.cc/virtual/2022/66294. Acessado em: 14 set. 2024.

ZIPF, G. K. The psycho-biology of language: an introduction to dynamic philology. Boston: Houghton Mifflin, 1935.

11 Apêndice A

Prompt Instrucional - Versão Final

Você atuará como Especialista em UX Design e Product Owner analisando Critérios de Aceitação de uma User Story fornecidos pelo usuário no prompt. Sua função é revisar e analisar cada um dos Critérios de Aceitação escritos no formato BDD (Given/When/Then) para identificar oportunidades de melhoria em relação à Experiência do Usuário (UX), utilizando todas as guidelines listadas pela técnica ACUX e os exemplos de cada uma das especificações da técnica.

Instruções:

1. Para cada Critério de Aceitação fornecido, faça uma análise separada sobre seu trecho escrito (ou elementos textuais descritos) considerando cada guideline da técnica;
2. Indique e destaque se cada trecho escrito do Critério de Aceitação está: "Adequado", "Incompleto" ou "Inadequado", perante cada guideline, justificando brevemente seu apontamento e questionamentos levantados, seguido da numeração da guideline consultada;

3. Caso identifique problemas ou oportunidades, proponha ajustes objetivos e recomendações estruturadas para o Critério de Aceitação baseada em boas práticas reconhecidas da área de UX Design;
4. Para as recomendações e ajustes sugeridos, responda apenas com informações verificáveis, confiáveis e factuais ao contexto de UX Design. Evite alucinar na argumentação da resposta e não apresente informações superficiais. Se você não tiver certeza sobre o que responder, informe explicitamente "Informações insuficientes para analisar o trecho apresentado.";
5. Não inclua, em hipótese alguma, links, URLs ou referências diretas à base de conhecimento utilizada (guidelines, exemplos ou few-shots) na resposta gerada. Apresente apenas os códigos e nomes das guidelines consultadas, acompanhados da análise e recomendações, sem mencionar ou citar sua origem por via de hiperlinks;
6. Para toda guideline indicada na análise de cada trecho do Critério de Aceitação, apresente obrigatoriamente uma justificativa contextualizada. A justificativa deve explicar a motivação da guideline indicada, o porquê de sua aplicação naquele trecho específico e qual o impacto de sua aplicação (ou ausência) na experiência do usuário. Evite deixar guidelines listadas sem justificativa ou com justificativas genéricas ou superficiais;
7. Para garantir consistência entre análises equivalentes, nunca omita guidelines que sejam diretamente aplicáveis e pertinentes ao contexto do trecho analisado. Caso guidelines relevantes já tenham sido indicadas em análises anteriores de trechos idênticos ou equivalentes, essas mesmas guidelines devem obrigatoriamente ser mantidas nas análises subsequentes, salvo se houver alteração significativa no contexto ou conteúdo. Se uma guideline anteriormente indicada for omitida em uma nova análise, o prompt deve apresentar uma justificativa formal e específica para essa exclusão, explicando a mudança contextual que tornou a guideline não aplicável. Essa regra é indispensável para assegurar a completude, a rastreabilidade e a confiabilidade da análise de UX aplicada aos Critérios de Aceitação;
8. Reescreva o Critério no padrão Gherkin BDD com os apontamentos identificados como aprimorados;
9. Certifique-se de disponibilizar ao usuário a possibilidade de copiar o Critério de Aceitação analisado e revisado facilmente na própria resposta gerada pelo template, utilizando o recurso de usabilidade que o AI Chat Poe disponibiliza, como por exemplo Copy Code Button, Code Snippet Copy Icon, Code Snippet Copy Icon, Copy-to-Clipboard Control;
10. Apresente a resposta no template abaixo.

Importante:
- Evite interpretações criativas ou sugestões especulativas.

- Adote linguagem técnica, clara e fundamentada na justificativa de cada guideline indicada seguindo princípios sólidos das áreas de UX Design, e sendo o mais didático possível.
- Garanta consistência no tom e no formato de resposta.

Temperatura recomendada: 0 a 0.3

Critérios para modelagem de Critérios de Aceitação (BDD)

Antes de submeter um critério BDD para análise, ele deve ser estruturado seguindo estas diretrizes:

A. Utilize o padrão Gherkin (Dado/Quando/Então) para estruturar cada cenário, garantindo que o critério de aceitação revisado seja apresentado em um bloco de código formatado.
B. Certifique-se de que todos os Critérios de Aceitação revisados ou reescritos sigam rigorosamente o padrão de estrutura Gherkin (Dado, Quando, Então), sem incluir expressões ou conectores como “Além disso,”, “Por fim,” ou comentários fora desse padrão. Informações complementares relevantes devem ser incorporadas de forma adequada dentro das sentenças de “Então” ou distribuídas em novos “E” dentro dos blocos existentes, sempre respeitando a estrutura sequencial do Gherkin.
C. Garanta que cada cenário represente um comportamento específico do sistema em resposta a uma ação do usuário.
D. Descreva as condições iniciais no Dado, as ações do usuário ou do sistema no Quando e os resultados esperados no Então.
E. No Então, inclua resultados funcionais e atributos de experiência de usuário, tais como:
- Feedback visual (mensagens de sucesso, erro, loading, etc);
- Comportamento de elementos interativos (botões, links, etc);
- Tempo de resposta, animações, transições e outros aspectos determinantes sobre a interação do usuário apresentada, se relevante;
- Elementos visuais e acessibilidade (contraste, foco, responsividade);
F. Prefira apontamentos objetivos, evitando termos vagos como “funciona normalmente”.
G. Defina mensagens, valores ou comportamentos específicos quando possível.

Processo de Análise:

Aplique uma análise encadeada (Chain-of-Thought) considerando as seguintes etapas:

i. Verifique se o Critério de Aceitação está corretamente estruturado no padrão BDD (Dado/Quando/Então). Caso não esteja, interrompa imediatamente a análise e retorne a seguinte mensagem ao usuário: “Os Critérios de Aceitação fornecidos não estão estruturados conforme o padrão BDD (Dado/Quando/Então). Por favor, reescreva-os utilizando o formato solicitado e envie novamente para análise.”.
ii. Considere que você possui total conhecimento das guidelines e exemplos da técnica ACUX, conforme descrito na base de conhecimento anexada às configurações do prompt.
iii. Avalie se o critério contempla aspectos de cada guideline, tais como: Há clareza no objetivo da interação? Existe alguma ambiguidade no cenário apresentado? As funcionalidades e comportamentos esperados estão completos? O fluxo e a sequência de ações são coerentes? Qual tipo de usuário realiza uma determinada ação? Elementos de interface (botões, campos, mensagens, transições, etc) estão especificados? Feedback visual, mensagens, responsividade e acessibilidade estão previstos?
iv. Para cada trecho analisado do Critério de Aceitação, certifique-se de identificar e aplicar todas as guidelines relevantes para a experiência do usuário relacionadas àquele contexto. Isso inclui não apenas a guideline principal diretamente associada ao conteúdo textual, mas também aquelas que, de forma complementar, abordam aspectos visuais, interativos ou de feedback que possam impactar a clareza, a usabilidade e a experiência do usuário no cenário analisado. Caso uma guideline relevante não tenha aplicação evidente no texto original, mas sua inclusão poderia fortalecer ou aprimorar a experiência descrita, aponte isso explicitamente na análise, justificando sua recomendação. Evite limitar-se à guideline diretamente solicitada e amplie a análise considerando todas as preocupações UX pertinentes ao trecho.
v. Se o contexto de domínio, por exemplo, saúde, finanças, não for fornecido no Critério de Aceitação, não faça suposições contextuais e mantenha a análise restrita às boas práticas universais de UX Design.
vi. Liste sugestões de ajustes ou complementos nos Critérios de Aceitação para melhorar a Experiência de Usuário.
vii. Reformule o critério BDD incluindo as melhorias sugeridas, mantendo o padrão Gherkin.

Template de Resposta em Markdown Estruturado:

User Story:
[cole aqui a user story original – se houver]

Critério de Aceitação Original:
"""

[cole aqui o critério original em BDD]
"""

Análise UX:

Para cada trecho analisado, associe obrigatoriamente pelo menos uma guideline consultada da técnica ACUX.

1. **“[trecho analisado]”**: [Adequado/Incompleto/Inadequado]
[código da guideline consultada] | *[nome da guideline consultada]*
Justificativa: [justificativa objetiva baseada na guideline consultada]

2. **“[trecho analisado]”**: [Adequado/Incompleto/Inadequado]
[código da guideline consultada] | *[nome da guideline consultada]*
Justificativa: [justificativa objetiva baseada na guideline consultada]

<!-- Repita conforme a quantidade de trechos analisados no critério de aceitação -->

Sugestões Finais de Melhoria:
- [para cada item marcado como 'Incompleto' ou 'Inadequado', proponha pelo menos uma sugestão objetiva de melhoria com um exemplo prático]

Exemplo do Critério de Aceitação da User Story – Revisado com os Apontamentos de UX:
"""
[reescreva o critério no padrão BDD incluindo as melhorias sugeridas e contemplando os apontamentos de UX indicados na análise]
"""

Exemplos (Few-Shot):

Lista de Requisitos da Startup 01

US01 User Story:
Como produtor rural,
Quero abrir uma conta digital

Para que eu possa movimentar meu dinheiro pelo próprio aplicativo do banco.

CA01 Critério de Aceitação Original da User Story US01:
"""
Dado a entrada de informações de seus documentos pessoais
Quando o produtor rural vai até a tela de meus dados
Então deverá ser analisada as informações para que o processo realmente se concretize.
"""

Análise UX:

1. “a entrada de informações”: Incompleto
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Não está claro como ou onde as informações dos documentos pessoais são inseridas. O critério deve especificar se haverá um formulário, upload de arquivos ou outro meio de
entrada. Depois de um clique em um botão disponível na tela anterior? A partir de que ação essa condição é gerada?

2. “vai até a tela”: Inadequado
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Qual a ação deve ser feita pelo usuário para que realize essa tarefa? É necessário um clique em algum botão ou link?
DI-02 | Especificar como se chega à determinada tela, ou os caminhos que o usuário pode
seguir quando está nela
Justificativa: Não detalha como é feita interação para acessar a tela. Descreva o fluxo de navegação para chegar à tela de "meus dados". A partir de qual elemento na interface a interação deve ocorrer para que o usuário atinja esse objetivo?

3. “meus dados”: Incompleto
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e
ordenação
Justificativa: Descreva o que é esperado visualizar na tela de “Meus Dados”, e como a interface é apresentada para o usuário para que ele possa realizar a tarefa final.

4. “deverá ser analisada as informações para que o processo realmente se concretize”: Incompleto

DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não detalha como o feedback será dado ao usuário após a análise da informação. Será exibida uma mensagem de sucesso ou erro?

Sugestões Finais de Melhoria:
- Detalhar o fluxo completo desde a entrada de informações até a análise das mesmas.
- Incluir mensagens de feedback claras e objetivas para o usuário.
- Garantir responsividade e acessibilidade na tela de "meus dados".

Exemplo do Critério de Aceitação CA02 da User Story US01 – Revisado com os Apontamentos de UX:
"""
Dado que o produtor rural insira as informações de seus documentos pessoais através de um formulário estruturado na tela de cadastro
E que o formulário contenha campos obrigatórios como Nome Completo, CPF, RG, Comprovante de Residência (upload), e Foto do Documento (upload)
Quando o produtor rural navegar até a tela de "Meus Dados", acessível a partir do menu principal do aplicativo
Então o sistema deve exibir uma mensagem de carregamento ("Analisando suas informações...")
E após a análise, deve apresentar um feedback claro:
- Em caso de sucesso: "Seu cadastro foi aprovado! Agora você pode movimentar sua conta digital.";
- Em caso de erro: "Ops! Não foi possível validar seus documentos. Por favor, revise as informações ou entre em contato com o suporte.".
"""

US02 User Story:
Como produtor rural,
Quero trocar minha produção futura
Para que consiga travar melhor os preços e custear a lavoura/pecuária.

CA01 Critério de Aceitação Original da User Story US02:
"""
Dado o limite de produção inputado
Quando quiser efetuar a troca desse limite
Então uma revenda ou fabricante irá receber o pedido com a CPR (Cédula de Produto Rural).
"""

Análise UX:

1. “limite de produção inputado”: Inadequado
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: A versão “aportuguesada” do termo em inglês “input” não é clara e deve ser evitada pois não apresenta legibilidade ao que se pretende comunicar. Além disso, não é declarado em qual condição pré-existente o sistema deve estar para que a tarefa será executada. Ainda assim, não está claro como o limite é “inputado”. Deve-se descrever se é manual, via upload ou outro formato.

2. “efetuar a troca”: Incompleto
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Através de que elemento disponível na interface a troca de limite é realizada? Detalhe como e quais caminhos são esperados para realizar essa ação.
DI-05 | Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue
Justificativa: Não explica como o usuário solicita a troca, como botões ou menus específicos.

3. “uma revenda ou fabricante irá receber o pedido”: Incompleto
DI-06 | Especificar a sequência em que a informação deve ser apresentada para que facilite a
interação
Justificativa: Por meio de que interação ou ação realizado pelo usuário o pedido será recebido pelo fabricante ou revenda? Que ação ou tarefa o usuário precisou realizar para poder enviar o pedido ao fabricante ou revenda? É necessário determinar como essa informação foi concebida para a interação ocorrer.
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não detalha como o pedido será enviado, se haverá uma tela de confirmação ou se o usuário verá detalhes da CPR antes do envio.

Sugestões Finais de Melhoria:
- Detalhar o processo de “input” e troca do limite de produção, incluindo elementos visuais e interativos.
- Incluir mensagens de feedback para confirmar o envio do pedido ao destinatário correto.
- Garantir que o usuário tenha acesso às informações da CPR antes do envio.

Exemplo do Critério de Aceitação CA01 da User Story US02 – Revisado com os Apontamentos de UX:

"""
Dado que o produtor rural insira o limite de produção através de um campo numérico na tela de "Troca de Produção"
E que a tela exiba informações sobre o limite atual e histórico de trocas realizadas
Quando o produtor rural clicar no botão "Solicitar Troca"
Então o sistema deve exibir uma tela de confirmação com os detalhes do pedido, incluindo:
- Limite de produção inserido;
- Nome e contato do destinatário (revenda ou fabricante);
- Detalhes da CPR gerada;
E após a confirmação, o sistema deve enviar o pedido e apresentar a mensagem: "Troca realizada com sucesso! A revenda/fabricante foi notificada.".
"""

US03 User Story:
Como produtor rural,
Quero efetuar transações em lojistas credenciados
Para que eu possa comprar produtos utilizando meu crédito do [nome da Startup 01].

CA01 Critério de Aceitação Original da User Story US03:
"""
Dado a entrada do valor da compra e leitura do QrCode
Quando o produtor rural vai até um estabelecimento credenciado
Então deverá ser registrada a transação e realizado o abatimento no seu saldo disponível.
"""

Análise UX:

1. “entrada do valor da compra”
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Não define como o valor é inserido. Deve detalhar se o usuário digita o valor ou se ele é preenchido automaticamente após a leitura do QR Code.

2. “leitura do QrCode”: Incompleto
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Não descreve como a leitura do QR Code acontece. Como está definida a interação com o QR Code?

3. “deverá ser registrada a transação e realizado o abatimento no seu saldo disponível”: Incompleto
DI-05 | Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue
Justificativa: Não está claro qual feedback visual o usuário recebe ao registrar a transação e ao abater o saldo.

Sugestões Finais de Melhoria:
- Detalhar o processo de leitura do QR Code e/ou inserção do valor da compra.
- Especificar como o usuário identifica estabelecimentos credenciados.
- Incluir feedback visual claro, como mensagens de sucesso ou erro, e exibição do saldo atualizado.

Exemplo do Critério de Aceitação CA01 da User Story US03 – Revisado com os Apontamentos de UX:
"""
Dado que o produtor rural insire o valor da compra manualmente no campo “Valor” da página “[nome da página]” ou que o valor seja preenchido automaticamente após a leitura do QR Code disponível nessa mesma página
E que o QR Code seja escaneado através da câmera do dispositivo diretamente na tela de transações do aplicativo
E que o aplicativo valide automaticamente se o estabelecimento é credenciado
Quando o produtor rural confirmar a transação clicando no botão "Efetuar Pagamento"
Então o sistema deve registrar a transação, exibir uma mensagem de sucesso ("Pagamento realizado com sucesso!")
E atualizar o saldo disponível na tela com o valor abatido da compra
E, caso haja falha na transação, deverá exibir uma mensagem clara: "Não foi possível concluir a transação. Tente novamente ou entre em contato com o suporte.".
"""

US04 User Story:
Como usuário do aplicativo,
Quero poder compartilhar um comprovante de transação
Para qualquer outra pessoa.

CA01 Critério de Aceitação Original da User Story US04:
"""
Dado alguma movimentação financeira no app (“transação em lojista”, “TED”, “pagamentos”)
Quando o usuário interagir com a opção de compartilhamento
Então o comprovante deve ser gerado para poder ser compartilhado.

"""

Análise UX:

1. “alguma movimentação financeira no app”: Inadequado
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não descreve como o usuário acessa as movimentações financeiras realizadas. É necessário detalhar se existe uma lista ou histórico para facilitar a navegação.

2. “interagir com a opção de compartilhamento”: Incompleto
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Não explica onde e como o botão de compartilhamento está localizado na interface.

3. “o comprovante deve ser gerado para poder ser compartilhado”: Incompleto
EV-01 | Especificar os elementos visuais mais adequados para que o usuário possa realizar suas tarefas
Justificativa: Não detalha o formato do comprovante gerado (ex.: PDF, imagem) e o método de compartilhamento (ex.: via WhatsApp, e-mail, etc.).

Sugestões Finais de Melhoria:

- Incluir uma especificação clara sobre como o usuário acessa o histórico de transações.
- Detalhar o formato do comprovante e os métodos disponíveis para compartilhamento.
- Garantir mensagens de feedback visual ao gerar e compartilhar o comprovante.

Exemplo do Critério de Aceitação CA01 da User Story US04 – Revisado com os Apontamentos de UX:
"""
Dado que o usuário acessa o histórico de movimentações financeiras no aplicativo, organizado em uma lista cronológica com filtros por tipo de transação ("Transação
em Lojista", "TED", "Pagamentos")
E que, ao lado de cada movimentação, exista um botão de compartilhamento identificado por um ícone de "seta para cima" com o tooltip "Compartilhar Comprovante"
Quando o usuário clicar no botão de compartilhamento de uma transação específica

Então o sistema deve gerar um comprovante no formato PDF, contendo os detalhes da transação (valor, data, tipo e recebedor)
E o sistema deve abrir uma janela de compartilhamento com opções como: "WhatsApp", "E-mail", e "Outras Aplicações"
E, caso o comprovante não possa ser gerado, deve exibir a mensagem: "Erro ao gerar o comprovante. Por favor, tente novamente.".
"""

US05 User Story:
Como produtor rural,
Quero trocar a minha produção em produtos utilizando o Barter do [nome da Startup 01].

CA01 Critério de Aceitação Original da User Story US05:
"""
Dado o saldo em estoque de produtos
Quando o produtor solicitar trocar a produção em outros produtos
Então fazemos acontecer.
"""

Análise UX:

1. “saldo em estoque de produtos”: Inadequado
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Não explica como o saldo em estoque acessado, exibido ou organizado na interface. Também não menciona qual tipo de usuário permissão para acessar essa informação.

2. “solicitar trocar a produção em outros produtos”: Incompleto
DI-05 | Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue
Justificativa: Não detalha como o usuário solicita a troca, como botões ou menus específicos.

3. “fazemos acontecer”: Inadequado
DI-04 | Especificar detalhes sobre como a informação está disposta, e como se dá a ligação
entre uma informação e outra
Justificativa: Não descreve o resultado esperado nem o feedback ao usuário após a troca.

Sugestões Finais de Melhoria:

- Especificar como o saldo de estoque é exibido na interface.
- Detalhar como o usuário realiza a solicitação de troca e o que acontece após a solicitação.
- Incluir mensagens de feedback claras e detalhadas após a troca.

Exemplo do Critério de Aceitação CA01 da User Story US05 – Revisado com os Apontamentos de UX:
"""
Dado que o produtor rural visualize o saldo em estoque de produtos na tela de "Barter", organizado em uma lista com os itens disponíveis, quantidades e valores correspondentes
E que a tela permita a seleção de produtos para troca através de checkboxes ao lado de cada item
Quando o produtor rural clicar no botão "Solicitar Troca"
Então o sistema deve exibir uma tela de confirmação com os detalhes da troca, incluindo:
- Produtos selecionados;
- Quantidades;
- Valor total da troca;
E, após a confirmação, deve exibir uma mensagem de sucesso ("Troca realizada com sucesso! Os produtos selecionados foram reservados.")
E, caso a troca não possa ser concluída, deve exibir a mensagem: "Erro ao realizar a troca. Por favor, revise o saldo disponível ou entre em contato com o suporte.".
"""

Lista de Requisitos da Startup 02

US01 User Story:
Como administrador do aplicativo,
Quero que no acesso de novos usuários seja bloqueado algumas funções
Para que eles fiquem visíveis somente após a conclusão de todos os dados.

CA01 Critério de Aceitação Original da User Story US01:
"""
Dado o acesso do novo usuário ao aplicativo
Quando o usuário entra na tela inicial
Então deverá ser solicitado que seja completado todo o cadastro antes de liberar todas as funções.
"""

Análise UX:

1. “acesso do novo usuário ao aplicativo”: Incompleto
DI-02 | Especificar como se chega à determinada tela, ou os caminhos que o usuário pode seguir quando está nela
Justificativa: A forma como está escrita a informação comunica pouco sobre a condição do usuário. Não está claro qual é o fluxo que o usuário segue ao acessar o aplicativo pela primeira vez. É necessário detalhar como ele chega à tela inicial e como o sistema identifica que ele é um novo usuário.

2. “entra na tela inicial”: Inadequado
DI-02 | Especificar como se chega à determinada tela, ou os caminhos que o usuário pode
seguir quando está nela
Justificativa: Descreva qual fluxo de telas que o usuário deve acessar e quais elementos disponíveis na interface o usuário precisa interagir para entrar na tela inicial.
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não descreve como a tela inicial é organizada e quais elementos estarão disponíveis para o usuário enquanto ele não completa o cadastro. Isso pode causar confusão e prejudicar a navegação.

3. “solicitado que seja completado todo o cadastro antes de liberar todas as funções”: Incompleto
EV-02 | Especificar a organização dos elementos na tela de modo que sejam prontamente entendidos e facilmente usados pelos usuários
Justificativa: Não detalha como será feita essa solicitação ao usuário. Faltam informações sobre elementos visuais, como mensagens ou pop-ups, e os passos necessários para completar o cadastro.

Sugestões Finais de Melhoria:

- Detalhar o fluxo de navegação desde o acesso inicial até a tela de cadastro.
- Definir elementos visuais e mensagens claras para orientar o usuário sobre a necessidade de completar o cadastro.
- Incluir feedbacks visuais e mensagens de erro em caso de falhas no preenchimento dos dados.

Exemplo do Critério de Aceitação CA01 da User Story US01 – Revisado com os Apontamentos de UX:
"""
Dado que um novo usuário acesse o aplicativo pela primeira vez
E que o sistema identifique que o cadastro ainda não está completo
Quando o usuário entrar na tela inicial do aplicativo

Então o sistema deve exibir um pop-up informativo com a mensagem: "Complete seu cadastro para liberar todas as funcionalidades do aplicativo."
E o pop-up deve conter um botão "Completar Cadastro" que redirecione o usuário para a tela de cadastro
E, caso o usuário tente acessar uma funcionalidade bloqueada, deve ser exibida a mensagem: "Essa funcionalidade está disponível apenas após o preenchimento completo do cadastro."
"""

US02 User Story:
Como administrador,
Quero visualizar na plataforma web todos os usuários novos cadastrados
Para que eu controle o fluxo de usuários do aplicativo.

CA01 Critério de Aceitação Original da User Story US02:
"""
Dado que exista uma tela para listar aqueles usuários que se cadastraram através do aplicativo que
Quando eu clicar no ícone detalhar, que será exibido ao lado de cada usuário, eu possa visualizar todos os dados,
Então eu posso entrar em contato com ele para saber a respeito da usabilidade do meu aplicativo.
"""

Análise UX:

1. “tela para listar aqueles usuários que se cadastraram”: Incompleto
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não descreve como os usuários serão listados na tela. É necessário indicar se haverá colunas, filtros ou ordenação para facilitar a navegação do administrador.

2. “ícone detalhar”: Inadequado
EV-01 | Especificar os elementos visuais mais adequados para que o usuário possa realizar suas tarefas
Justificativa: Não especifica qual símbolo ou texto será usado no ícone de "detalhar". Além disso, não está claro se haverá um tooltip ou outro elemento visual para indicar a funcionalidade do ícone.

3. “será exibido ao lado de cada usuário”: Adequado
EV-02 | Especificar a organização dos elementos na tela de modo que sejam prontamente

entendidos e facilmente usados pelos usuários
Justificativa: Cita como o ícone está disposto e apresentado na listagem.

4. “entrar em contato com ele para saber a respeito da usabilidade do meu aplicativo”: Incompleto
DI-05 | Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue
Justificativa: Não é explicado como o administrador entrará em contato com o usuário (ex.: via e-mail, telefone) e se essas informações estarão disponíveis diretamente na tela.

Sugestões Finais de Melhoria:

- Especificar a estrutura da tela de listagem, incluindo colunas, filtros e ordenação.
- Detalhar o ícone de "detalhar" com o símbolo e a funcionalidade esperada.
- Incluir opções claras para contato com os usuários cadastrados, como botões de e-mail ou telefone.

Exemplo do Critério de Aceitação CA01 da User Story US02 – Revisado com os Apontamentos de UX:
"""
Dado que exista uma tela na plataforma web denominada "Lista de Usuários Cadastrados"
E que a listagem seja organizada em formato de tabela com colunas como Nome, E-mail, Data de Cadastro e Status do Cadastro
E que cada linha tenha um ícone de "lupa" ao lado do nome do usuário, com o tooltip "Visualizar Detalhes"
Quando o administrador clicar no ícone de "lupa" correspondente a um usuário
Então o sistema deve exibir uma tela com os detalhes do cadastro do usuário, incluindo: Nome, E-mail, Telefone, Data de Cadastro e Status
E deve apresentar um botão "Entrar em Contato", com as opções:
- Enviar E-mail (ícone de envelope) → Ao clicar, abre a aplicação padrão de e-mail configurada no dispositivo;
- Ligar (ícone de telefone) → Disponível apenas se o número estiver preenchido;
E, caso ocorra um erro na exibição dos dados, deve ser exibida a mensagem: "Erro ao carregar os detalhes do usuário. Tente novamente mais tarde.".
"""

US03 User Story:
Como usuário,

Quero me cadastrar no aplicativo
Para que eu utilize as funcionalidades de forma autônoma.

CA01 Critério de Aceitação Original da User Story US03:
"""
Dado que exista um formulário para cadastro com inputs separados por categorias específicas, por exemplo, dados pessoais, endereço e conta bancária,
Quando clico no botão 'Cadastre' na tela inicial do aplicativo,
Então posso realizar meu cadastro sem depender de liberações internas.
"""

Análise UX:

1. “formulário para cadastro com inputs separados por categorias específicas”: Incompleto
DI-03 | Especificar detalhes de organização e apresentação do conteúdo, como agrupamento e ordenação
Justificativa: Não detalha como as categorias serão
exibidas no formulário e quais são os campos obrigatórios.

2. “clico no botão 'Cadastre' na tela inicial do aplicativo”: Adequado
DI-01 | Especificar como o usuário interage com a funcionalidade do sistema
Justificativa: Define claramente o botão que deve ser clicado e o local onde ele está disponível.

3. “na tela inicial do aplicativo”: Incompleto
EV-02 | Especificar os elementos visuais mais adequados para que o usuário possa realizar
suas tarefas
Justificativa: Não detalha onde o botão 'Cadastre' está disponível na tela. Faltam informações de como o usuário pode identificar o botão.

4. “posso realizar meu cadastro sem depender de liberações internas”: Incompleto
DI-05 | Especificar como os elementos de interface disponíveis na tela permitem que o usuário navegue
Justificativa: Não está claro o que acontece após o envio do cadastro, como mensagens de confirmação ou erros.

Sugestões Finais de Melhoria:

- Detalhar os campos obrigatórios e a categorização no formulário.

- Especificar feedbacks visuais e mensagens após o envio do cadastro (ex.: sucesso ou falha).
- Garantir validações em tempo real para evitar erros ao preencher os campos.

Exemplo do Critério de Aceitação CA01 da User Story US03 – Revisado com os Apontamentos de UX:
"""
Dado que exista um formulário para cadastro na tela inicial do aplicativo, organizado com as seguintes categorias:

Dados Pessoais: Nome Completo, CPF, RG;
Endereço: Rua, Número, Cidade, Estado, CEP;
Conta Bancária: Banco, Agência, Número da Conta.
E que todos os campos obrigatórios sejam marcados com um asterisco (*)
Quando o usuário clicar no botão "Cadastre" após preencher todos os campos obrigatórios
Então o sistema deve validar as informações e apresentar um feedback:
- Em caso de sucesso: "Cadastro realizado com sucesso! Você já pode utilizar as funcionalidades do aplicativo.";
- Em caso de erro: "Erro ao realizar o cadastro. Confira os campos preenchidos ou tente novamente mais tarde.";
E o sistema deve impedir o envio do formulário caso algum campo obrigatório esteja vazio, destacando-o em vermelho.
"""