





ARTIGO DE PESQUISA/RESEARCH PAPER


# Investigando a utilização de protótipos de interface do usuário no desenvolvimento do software

## *Investigating the Use of User Interface Prototypes in Software Development*

Ana Almeida   [Universidade Federal Rural da Amazônia | carolinafernades2208@gmail.com ]

Paulo Malcher  [Universidade Federal Rural da Amazônia | paulo.malcher@ufra.edu.br ]

Fabrizio Garcia  [Universidade Federal Rural da Amazônia | fabricio.garcia@ufra.edu.br ]

 Campus Capitão Poço, Universidade Federal Rural da Amazônia, Tv. Pau Amarelo, S/N, Vila Nova, Capitão Poço, PA, 68650-000, Brasil.

**Resumo.** A prototipação de interfaces de usuário é amplamente utilizada no desenvolvimento de software para apoiar a validação precoce e a comunicação de decisões de design. Entretanto, ainda é limitada uma compreensão estruturada dos fatores que influenciam sua utilização. Este estudo apresenta uma Revisão Rápida da Literatura de 30 estudos primários com o objetivo de investigar os fatores que influenciam o uso de protótipos de UI no desenvolvimento de software. Foram identificados 21 fatores, organizados de acordo com o *framework* Tecnologia–Organização–Ambiente (TOE). Os resultados indicam a predominância de fatores tecnológicos, especialmente aqueles relacionados a ferramentas de prototipação, automação e abordagens baseadas em modelos, seguidos por fatores ambientais associados à comunicação e ao alinhamento entre stakeholders. Os fatores organizacionais destacam a influência da redução de custos, tempo e retrabalho nas decisões de prototipação. Além disso, a revisão identificou uma variedade de soluções de prototipação, com destaque para protótipos de baixa e alta fidelidade apoiados por ferramentas digitais e avaliados principalmente por meio de testes de usabilidade e inspeções por especialistas. Esses achados fornecem uma visão concisa dos fatores sociotécnicos que moldam o uso da prototipação de UI no desenvolvimento de software.

**Abstract.** User interface prototyping is widely used in software development to support early validation and communication of *design* decisions. However, a structured understanding of the factors that influence its use is still limited. This study presents a Rapid Review of 30 primary studies to investigate factors influencing the use of UI prototypes in software development. A total of 21 factors were identified and organized according to the Technology–Organization–Environment (TOE) framework. The results indicate a predominance of technological factors, especially those related to prototyping tools, automation, and model-based approaches, followed by environmental factors associated with communication and stakeholder alignment. Organizational factors highlight the influence of cost, time, and rework reduction on prototyping decisions. Additionally, the review identified a variety of prototyping solutions, with emphasis on low- and high-fidelity prototypes supported by digital tools and evaluated mainly through usability testing and expert inspections. These findings provide a concise overview of the sociotechnical factors that shape the use of UI prototyping in software development.

**Palavras-chave:** Prototipação de Interface do Usuário, Revisão Rápida, Framework TOE, Fatores Sociotécnicos

**Keywords:** User Interface Prototyping, Rapid Review, TOE Framework, Sociotechnical Factors

**Recebido/Received:** 01 April 2026 • **Aceito/Accepted:** 03 May 2026 • **Publicado/Published:** 12 June 2026

## 1 Introdução

A prototipagem da interface do usuário é uma parte essencial da engenharia de software, pois facilita a elicitação de requisitos [Chen *et al.*, 2021a]. No processo de desenvolvimento de software, a prototipagem é compreendida como uma técnica aplicada às atividades do fluxo de requisitos, sendo utilizada em diferentes etapas para apoiar a visualização, a comunicação e a validação de ideias [Garcia *et al.*, 2017]. Os protótipos constituem ferramentas importantes para a validação do *design* e da usabilidade do sistema, permitindo que a interface seja apresentada aos usuários e que a interação seja testada em estágios iniciais do desenvolvimento. Dessa forma, os usuários podem fornecer *feedback* aos projetistas, o que facilita a identificação de soluções que efetivamente atendam aos requisitos do projeto [OLIVEIRA NETTO, 2004].

Um protótipo é uma versão inicial de um sistema de software, utilizada para demonstrar conceitos, experimentar opções de projeto e compreender melhor o problema e suas

possíveis soluções [Koscianski e dos Santos Soares, 2007]. No contexto da interface do usuário, os protótipos são representações visuais que ilustram como a interface de um software deve ser utilizada durante a interação com o usuário final [Ricca, 2014]. Essas representações podem variar desde desenhos simples, empregados para apresentar as interações entre o usuário e o sistema, até versões mais detalhadas, que incluem elementos gráficos mais elaborados, como logotipos específicos ou cores associadas à identidade visual da marca [Ricca, 2014].

O uso de protótipos pode trazer inúmeras vantagens, que vão desde a redução dos riscos associados a mudanças de requisitos até o aprimoramento das definições de projetos de interface [Soares, 2008]. No entanto, seu uso também pode acarretar alguns inconvenientes, como a omissão da complexidade de determinadas regras de negócio e a frustração dos usuários devido a eventuais mudanças na interface do software final [Soares, 2008]. Apesar do avanço das pesquisas em desenvolvimento de software nos últimos anos,

observa-se que ainda há poucos estudos secundários voltados especificamente à utilização de protótipos de interface do usuário. Diante desse cenário, este estudo busca identificar as principais características, soluções e desafios relacionados à utilização de protótipos de interface do usuário nesse contexto. Para atingir esse objetivo, foi realizada uma Revisão Rápida (RR) [Cartaxo *et al.*, 2018] para analisar o estado da arte sobre o uso de protótipos de interface do usuário no desenvolvimento de software. A RR configura-se como uma abordagem estruturada para identificar, categorizar e analisar evidências científicas, proporcionando uma visão abrangente do domínio Cartaxo *et al.* [2020].

Como resultado, a partir de 30 estudos primários selecionados, foram identificados 21 fatores que influenciam a utilização de protótipos de interface do usuário no desenvolvimento de software. Esses fatores foram organizados segundo o *framework Technology–Organization–Environment (TOE)* [Tornatzky e Fleischer, 1990], evidenciando a predominância de fatores associados à dimensão tecnológica (13 fatores), relacionados a ferramentas, métodos, técnicas e abordagens de automação para a prototipagem de interfaces. Além disso, foram identificados fatores de natureza ambiental (6 fatores), ligados a aspectos humanos e sociais, como comunicação, alinhamento entre *stakeholders* e compartilhamento de conhecimento, bem como fatores organizacionais (2 fatores) relacionados à gestão de projetos, incluindo custos, prazos e redução de retrabalho. Adicionalmente, as soluções mapeadas abrangem métodos, técnicas, ferramentas e práticas, com destaque para a prototipagem de baixa e alta fidelidade apoiada por ferramentas digitais como Figma e Sketch.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica; a Seção 3 apresenta os trabalhos relacionados; a Seção 4 descreve a metodologia de pesquisa; a Seção 5 apresenta os resultados; a Seção 6 discute as contribuições do estudo e apresenta as ameaças à validade; e, por fim, a Seção 7 conclui o artigo com considerações finais e trabalhos futuros.

## 2 Fundamentação teórica

A prototipagem de interface do usuário é uma técnica importante no processo de desenvolvimento de software, atuando como ponte entre a concepção da ideia e a implementação final [Bjarnason *et al.*, 2021]. Por meio de protótipos, *designers* de interface podem demonstrar a funcionalidade do software por meio de representações visuais [Deininger *et al.*, 2017]. De acordo com Manzo *et al.* [2022], os protótipos funcionam como modelos que permitem testar e ajustar um produto antes de sua versão final. Assim, um protótipo pode ser compreendido como um modelo de trabalho construído para desenvolver e testar ideias de *design*, permitindo examinar conteúdo, estética e interação técnica sob a perspectiva de *designers*, clientes e usuários [Manzo *et al.*, 2022].

No processo de desenvolvimento de software, a prototipagem de interface do usuário é amplamente empregada para apoiar a exploração e a validação de soluções de *design*. Seu principal objetivo é permitir a avaliação de alternativas em estágios iniciais do desenvolvimento, por meio da coleta de *feedback* de usuários reais de forma econômica e eficiente [Manzo *et al.*, 2022]. Dessa forma, a prototipagem contri-

bui para reduzir incertezas relacionadas aos requisitos e para aprimorar a comunicação entre os diferentes envolvidos no desenvolvimento do sistema. Mesmo após décadas de consolidação, essa técnica continua evoluindo e se mantém como um recurso relevante para a engenharia de requisitos, especialmente em contextos de desenvolvimento ágil [Larrea *et al.*, 2024].

A diversidade de aplicações da prototipagem pode ser categorizada principalmente pelo nível de fidelidade, distinguindo-se protótipos de baixa e de alta fidelidade, cada qual associado a benefícios específicos [Pernice, 2016]. Protótipos de baixa fidelidade, como esboços e modelos em papel, exigem menor esforço de preparação e facilitam alterações no *design* durante os testes. Em contrapartida, protótipos de alta fidelidade, geralmente desenvolvidos com ferramentas digitais, proporcionam interações mais realistas e possibilitam a avaliação de fluxos de trabalho e de elementos visuais específicos, como a hierarquia de páginas [Pernice, 2016].

Sob uma perspectiva sociotécnica, o uso de protótipos de interface do usuário pode ser compreendido como um fenômeno influenciado por múltiplos fatores inter-relacionados. O *framework* TOE [Tornatzky e Fleischer, 1990] propõe que a adoção e o uso de tecnologias em contextos organizacionais são condicionados por fatores tecnológicos, organizacionais e ambientais. Aplicado ao contexto da prototipagem de interfaces, esse arcabouço permite compreender como características das ferramentas e técnicas empregadas, práticas e decisões organizacionais, bem como aspectos humanos e sociais, influenciam a forma como os protótipos são utilizados ao longo do desenvolvimento de software.

## 3 Trabalhos relacionados

Garcia *et al.* [2017] realizaram um mapeamento sistemático da literatura (MSL) para identificar quais artefatos são utilizados na comunicação entre práticas de *design* centrado no usuário e métodos ágeis. A partir da análise de 56 estudos publicados entre 2002 e 2016, os autores identificaram 20 grupos de artefatos, incluindo protótipos, histórias de usuário, personas, cenários, *wireframes* e *mockups*. O estudo mostrou que cerca de 48% dos artefatos eram físicos, como protótipos de papel, e 52% eletrônicos, refletindo a coexistência entre práticas tradicionais e digitais no contexto ágil. Os resultados destacam protótipos e histórias de usuário como os principais instrumentos de alinhamento e validação iterativa, além de reforçar o papel da comunicação mediada por artefatos para reduzir ambiguidades e promover entendimento entre equipes.

Freitas *et al.* [2020] conduziram uma revisão sistemática da literatura (RSL) sobre ferramentas de prototipação rápida para realidade aumentada, com o objetivo de identificar ferramentas, tendências e oportunidades na área. O estudo identificou 30 artefatos voltados à criação de protótipos de baixa e alta fidelidade e evidenciou que as técnicas de prototipação em papel foram as mais citadas. Como contribuição, o trabalho oferece uma visão consolidada sobre o uso de prototipagem rápida em um domínio tecnológico específico, destacando os benefícios e limitações das abordagens existentes.

Larrea *et al.* [2024] investigaram o uso de protótipos

como ferramenta de apoio ao desenvolvimento de software em contextos ágeis, por meio de uma abordagem que combinou um MSL e entrevistas com profissionais da indústria. Os autores analisaram como os protótipos são utilizados ao longo do ciclo de desenvolvimento, identificando benefícios, desafios e um conjunto de boas práticas relacionadas, como o alinhamento entre protótipos e documentação, a escolha adequada do nível de fidelidade e o envolvimento da equipe na sua construção.

Os trabalhos analisados evidenciam diferentes abordagens à prototipação no desenvolvimento de software. Garcia et al. [2017] destacam o uso de protótipos como artefatos de comunicação entre práticas de *design* centrado no usuário e métodos ágeis. Freitas et al. [2020] concentram-se em ferramentas de prototipação rápida aplicadas ao domínio da realidade aumentada. Larrea et al. [2024] discutem o uso de protótipos como apoio às práticas de desenvolvimento ágil. Diferentemente desses trabalhos, o presente estudo analisa a utilização de protótipos de interface do usuário ao longo do ciclo de vida do desenvolvimento de software, adotando uma perspectiva sociotécnica que considera múltiplos atores, contextos e objetivos envolvidos nessa atividade.

## 4 Método de pesquisa

Para este estudo, foi conduzida uma RR entre maio e julho de 2025, com foco na utilização de protótipos de interface do usuário no desenvolvimento de software. As RR são estudos secundários orientados à prática, cujo principal objetivo é fornecer evidências que apoiem a tomada de decisões visando solucionar, ou ao menos atenuar, os problemas enfrentados pelos profissionais na prática [Cartaxo et al., 2020].

Uma RR se caracteriza como um método de pesquisa realizado em prazos mais curtos quando comparado às RSL, com algumas etapas da RSL sendo omitidas ou simplificadas [Cartaxo et al., 2018, 2020]. Para conduzir esta RR, foi seguido o modelo de protocolo proposto por Cartaxo et al. [2020]. Além disso, foram consideradas as diretrizes para a condução de RSL apresentadas por Kitchenham e Charters [2007].

A escolha pela RR, em detrimento de abordagens tradicionais como a RSL e MSL, justifica-se pelo equilíbrio entre o rigor científico e a necessidade de respostas céleres para problemas da indústria. Enquanto a RSL e o MSL priorizam a exaustividade e a abrangência total da literatura, processos que frequentemente demandam prazos extensos e um volume de dados que pode distanciar a pesquisa da realidade imediata do mercado, a RR foca na agilidade e na aplicabilidade prática. Os *trade-offs* envolvem uma simplificação deliberada de etapas, como a busca em bases mais restritas para garantir que as evidências sejam entregues em tempo hábil para apoiar decisões de projeto. Assim, a adoção da RR configura-se como um diferencial metodológico deste estudo: ao contrário de trabalhos correlatos que utilizam métodos exaustivos, esta pesquisa propõe uma síntese orientada à prática, conectando diretamente os achados acadêmicos às dores reais relatadas pelo profissional entrevistado.

Para identificar o problema prático deste estudo, um profissional da indústria de software foi entrevistado. O profissional possui mais de 4 anos de experiência em atividades

de prototipação de interfaces de usuário. Suas atividades incluem a criação de sistemas de *design* para padronização de produtos, pesquisa de mercado e testes A/B. Durante a entrevista, o profissional relatou suas experiências, limitações e desafios práticos, “*Com o protótipo, conseguimos validar e identificar alguns problemas que o usuário possa ter, conseguindo propor soluções melhores. Então, na minha opinião esses são os principais fatores do porquê utilizar um protótipo para desenvolver um sistema, produto, um software, o que for*”. No entanto, o profissional destacou que, apesar do uso recorrente da prototipagem em seu cotidiano, não há uma compreensão sistematizada sobre quais fatores efetivamente influenciam a utilização de protótipos ao longo do desenvolvimento de software, tampouco sobre como esses fatores se relacionam entre si ou impactam as decisões de projeto. Essa lacuna dificulta a adoção consciente e estratégica da prototipagem, que frequentemente ocorre de forma empírica, baseada na experiência individual, e não apoiada por um entendimento estruturado proveniente da literatura científica.

Também foi identificada a necessidade de investigar quais soluções (e.g., ferramentas, métodos, técnicas e práticas) voltadas à criação de protótipos de interface do usuário disponíveis e como essas soluções têm sido avaliadas. Nesse contexto, o profissional destacou a importância de ferramentas que facilitem a padronização e a visualização integrada do *design* para a equipe de desenvolvimento. Ressaltou-se, ainda, que a adoção de práticas de *User Experience (UX)* e o uso de protótipos, para além da ferramenta Figma, foram fundamentais para estabelecer essa padronização no processo de desenvolvimento, visto que “*Os desenvolvedores iam fazer uma tela e cada tela era diferente... Não havia padronização, cada desenvolvedor fazia o que ele achava que tinha que ser feito. Então, era um sistema bem feio... E atrapalhava muito a questão da usabilidade*”. Anteriormente, a ausência de um modelo centralizado levava os desenvolvedores a criar interfaces com *design* inconsistente, resultando em percepções distintas dos sistemas por parte do usuário final e comprometendo a usabilidade e a experiência do produto como um todo “*O desenvolvedor tinha uma tela que lista usuários e outra tela que lista produtos. O design em cada tela era totalmente diferente, a sensação que dava ao usuário era de navegar por dois sistemas diferentes, dois produtos diferentes, sendo que era o mesmo produto. Só que eram páginas diferentes, rotas diferentes ... Que na cabeça do desenvolvedor fazia sentido, mas para o usuário não*”. Por fim, evidenciou-se a necessidade de compreender os principais desafios associados à utilização de protótipos de interface do usuário.

Diante desse cenário, a demanda por uma RR emergiu da ausência de uma compreensão sistematizada dos fatores que influenciam a utilização de protótipos de interface do usuário ao longo do desenvolvimento de software, bem como das soluções e dos desafios associados à sua aplicação. Assim, a RR mostrou-se adequada para sintetizar evidências relevantes da literatura de forma alinhada às necessidades práticas dos profissionais da área.

### 4.1 Questões de pesquisa

Com a finalidade de alcançar o objetivo da pesquisa, este estudo estabeleceu três questões de pesquisa (QP), as quais foram formuladas em colaboração próxima com o especialista

na área envolvido na RR. A Tabela 1 apresenta as QP definidas para esta pesquisa.

**Tabela 1. Questões de pesquisa do estudo**

ID	Questão de Pesquisa
QP1	Quais são os fatores que influenciam a utilização de protótipos de interface do usuário no desenvolvimento de software?
QP2	Quais são as soluções (e.g., métodos, técnicas, ferramentas e práticas) utilizadas para a criação de protótipos de interface do usuário no desenvolvimento de software?
QP3	Quais são os desafios encontrados na utilização de protótipos de interface do usuário no desenvolvimento de software?

## 4.2 Estratégia de busca

Para identificar os estudos primários que abordam às QP descritas na Tabela 1, foi utilizada a base de dados da *Scopus*<sup>1</sup>, um indexador de várias bibliotecas digitais [Cartaxo et al., 2018]. A condução das buscas sistemáticas foi estabelecida por meio de uma *string* de busca formulada com base nos principais termos-chave derivados das QP deste estudo. A fase de elaboração incluiu a criação e o teste de múltiplas variações da *string* de busca nas bases de dados selecionadas, visando maximizar a recuperação de estudos relevantes. Para concluir o processo, a *string* de busca foi revisada por dois pesquisadores com experiência em Engenharia de Software e na condução de estudos secundários. A *string* de busca utilizada foi: (“*user interface prototype*” OR “*user interface prototyping*”) OR (“*UI prototype*” OR “*UI prototyping*”) AND (“*software development*” OR “*software engineering*”) OR (“*software process*” OR “*software lifecycle*” OR “*system development*”).

## 4.3 Seleção de estudos

Segundo Kitchenham e Charters [2007], em estudos secundários, o processo de seleção de estudos consiste em duas etapas, sendo a primeira a definição dos critérios de seleção e a segunda a definição do processo de seleção propriamente dito. As definições adotadas para cada etapa são descritas nas subseções a seguir.

### 4.3.1 Critérios de seleção dos estudos

A seleção dos estudos primários foi conduzida com base em critérios de inclusão (CI) e de exclusão (CE), definidos previamente para garantir a aderência aos objetivos da pesquisa. Conforme os critérios listados na Tabela 2, os estudos foram incluídos se atendiam ao CI e excluídos caso se enquadrassem em pelo menos um dos CE.

**Tabela 2. Critérios de seleção**

Tipo	ID	Descrição
Inclusão	CI1	O estudo responde pelo menos uma QP.
Exclusão	CE1	O estudo não atende ao CI.
	CE2	O estudo é duplicado.
	CE3	O estudo não está disponível para <i>download</i> gratuito ou por meio de acesso institucional.
	CE4	O estudo é uma dissertação de mestrado ou tese de doutorado.
	CE5	O estudo não está escrito em inglês.

### 4.3.2 Processo de seleção dos estudos

O processo de seleção dos estudos foi estruturado em seis etapas distintas, conforme descrito na Tabela 3. O método

iniciou-se com a busca no indexador Scopus, resultando em um conjunto inicial de estudos. A partir desse conjunto, aplicou-se uma série de filtros sucessivos para eliminar duplicatas e refinar a amostra. A análise da pertinência dos estudos foi realizada progressivamente, por meio da aplicação dos CI e CE, garantindo a seleção dos estudos relevantes.

**Tabela 3. Processo de seleção dos estudos**

Etapa	Descrição
Etapa 1	Execução da busca automatizada dos estudos primários, utilizando a <i>string</i> de busca desenvolvida.
Etapa 2	Aplicação do 1º filtro (remoção de duplicatas) ao conjunto de estudos recuperados na Etapa 1. Estudos duplicados são enquadrados no CE2.
Etapa 3	Aplicação do 2º filtro (aplicação dos CI e CE após a leitura do título, resumo e palavras-chave) no conjunto de estudos resultantes da Etapa 2.
Etapa 4	Aplicação do 3º filtro (aplicação dos CI e CE após a leitura da introdução e conclusão) no conjunto de estudos resultantes da Etapa 3.
Etapa 5	Aplicação do 4º filtro (aplicação do CI e CE após a leitura completa) no conjunto de estudos resultantes da Etapa 4.
Etapa 6	Extração e síntese dos dados dos estudos resultantes da Etapa 5.

## 4.4 Extração e síntese dos dados

O processo de extração de dados foi conduzido por um pesquisador, e os dados extraídos foram posteriormente verificados por dois pesquisadores mais experientes. Para garantir a sistematicidade do processo, foi utilizada a ferramenta Microsoft Excel com o objetivo de registrar, consolidar e organizar as informações necessárias para a análise.

Para responder às QP, utilizou-se um formulário estruturado com os seguintes campos: (1) identificador do estudo (ID); (2) título; (3) autores; (4) ano de publicação; (5) país de afiliação do primeiro autor; e (6) QP abordadas. Em conformidade com as diretrizes de RR estabelecidas por autores como Cartaxo et al. [2020] e King et al. [2022], a análise dos dados neste estudo foi conduzida por meio de um método de síntese descritiva (qualitativa). A Tabela 4 apresenta exemplos do processo de codificação aplicado aos estudos selecionados, incluindo os códigos resultantes e suas respectivas categorias. Os detalhes da extração e da síntese dos dados podem ser encontrados na Seção 5.

## 5 Resultados

A RR foi conduzida com o objetivo de identificar e sintetizar a literatura mais relevante e atual sobre a utilização de protótipos de interface do usuário no desenvolvimento de software, com foco nos fatores que influenciam seu uso, nas soluções propostas e nos desafios associados. Ao priorizar a agilidade na coleta de evidências, esse método permitiu organizar e analisar o conhecimento existente sobre esses aspectos. Não foram aplicadas restrições de tempo, o que permitiu a inclusão de estudos de diferentes períodos até maio de 2025 e garantiu um panorama temporal amplo da literatura.

A Figura 1 ilustra o processo de seleção dos estudos, detalhando o número de artigos remanescentes em cada etapa. Após a seleção, 30 estudos foram considerados relevantes e incluídos na extração e síntese dos dados.

A Figura 2 mostra a distribuição dos estudos selecionados ao longo dos anos. Nota-se uma variação na quantidade de estudos ao longo dos anos, com destaque para 2021 e 2025, que registraram o maior número de estudos (5 em cada ano), seguidos por 2024 (4 estudos). Após um período inicial com baixa incidência de estudos (2000 e 2020) geralmente com

<sup>1</sup><https://www.scopus.com/>

Tabela 4. Ilustração do processo de codificação

<b>Unidade:</b> During software development, the team needs to communicate effectively and share requirements information in order to achieve understanding, consensus, and commitment to the project's objectives. Requirements communication problems can cause productivity losses or even design failures.		
<b>Código</b>	<b>Código focado</b>	<b>Categoria</b>
Requirements communication problems can cause productivity losses or even design failures.	Comunicação de requisitos	Fator
<b>Unidade:</b> A drawback in enriching the use cases with the screen mockups is the burdensome task of guaranteeing the consistency between the graphical representation of the screen mockups and the textual descriptions of the use cases. As a matter of fact, the consistency between screen mockups and use cases cannot be guaranteed if the former are not disciplined by a structure or some rules.		
<b>Código</b>	<b>Código focado</b>	<b>Categoria</b>
Burdensome task of guaranteeing the consistency between the graphical representation of the screen mockups and the textual descriptions of the use cases.	Ambiguidade, incompletude e inconsistência dos requisitos	Desafio
<b>Unidade:</b> Historically, problems in GUI development include a fragmented, poorly integrated tool landscape and high synchronization efforts between stakeholders. Recent approaches suggest using large language models (LLMs) to recognize requirements fulfillment in GUIs and automatically propose new GUI components.		
<b>Código</b>	<b>Código focado</b>	<b>Categoria</b>
LLM-based assistant as a Figma plug-in that bridges the gap between user stories and GUI prototyping.	Geração Automatizada de Componentes e Layouts de GUI (via LLMs)	Solução

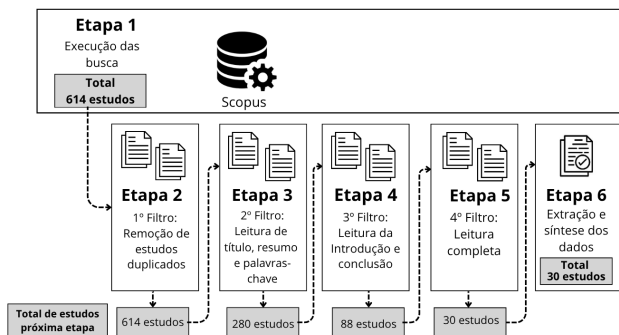


Figura 1. Processo de Seleção dos Estudos

um ou dois artigos por ano. No entanto, a partir de 2021, a produção acadêmica sobre o tema intensificou-se, evidenciando um crescimento no interesse da comunidade pela área investigada.

A Tabela 5 lista os estudos incluídos, apresentados em ordem decrescente por ano de publicação e referenciados por identificadores numéricos (E1-E30) para facilitar a citação nas seções subsequentes. Todos os dados brutos e os passos para a reprodução da pesquisa estão disponíveis em um material suplementar acessível publicamente<sup>2</sup>.

A Figura 3 apresenta a distribuição geográfica dos estudos selecionados, considerando o país de afiliação do primeiro autor. Observa-se que a Alemanha é o país com a maior contribuição, com quatro estudos (E03, E06, E08, E18). Em seguida, destacam-se quatro países com dois estudos cada: China (E07, E15), Eslováquia (E01, E02), Estados Unidos (E04, E05) e Portugal (E17, E23). A análise evidencia uma ampla dispersão geográfica, com outros países, como o Brasil (E14), o Canadá (E16) e a Turquia (E11), contribuindo com

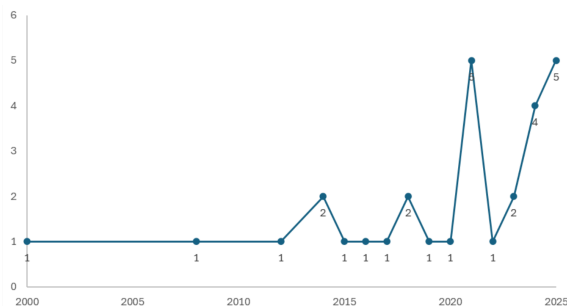


Figura 2. Número de estudos por ano

um estudo cada.



Figura 3. Número de estudos por países e períodos dos estudos mapeados

### 5.1 QP1 – Quais são os fatores que influenciam a utilização de protótipos de interface do usuário no desenvolvimento de software?

Esta QP teve como objetivo identificar, na literatura analisada, os principais fatores que influenciam a utilização de protótipos de interface do usuário no desenvolvimento de software. Ao todo, foram identificados 21 fatores, organizados de modo a permitir uma análise mais estruturada. Para a classificação desses fatores, adotou-se o *framework* TOE [Tornatzky e Fleischer, 1990], amplamente utilizado para analisar a adoção e o uso de tecnologias [Nahedh Alsehani *et al.*, 2024; Brahmastra *et al.*, 2024; Jais *et al.*, 2024]. O TOE estrutura os fatores de influência em três dimensões complementares: **Ambiental**, **Tecnológica** e **Organizacional**. A dimensão **Ambiental** abrange fatores de natureza humana e social, como comunicação, alinhamento entre *stakeholders*, compartilhamento de conhecimento e redução de ambiguidades. A dimensão **Tecnológica** reúne fatores relacionados ao produto e ao processo de desenvolvimento, incluindo ferramentas, técnicas, métodos e mecanismos de automação empregados na prototipagem de interfaces. A dimensão **Organizacional** contempla aspectos associados à gestão do projeto, como custos, prazos e redução de retrabalho, bem como práticas organizacionais que influenciam a adoção e o uso de protótipos.

A análise dos resultados indica a predominância de fatores associados à dimensão **Tecnológica**, que concentra 13 fatores (F07–F19), relacionados à disponibilidade e adequação de soluções tecnológicas para a prototipagem de interface do usuário. Em seguida, destacam-se os fatores da dimensão **Ambiental**, representada por seis fatores (F01–F06), que

<sup>2</sup><https://doi.org/10.5281/zenodo.19075422>

Tabela 5. Lista de estudos selecionados

ID	Título	Autores	País	Ano
E01	Is usability testing valid with prototypes where clickable hotspots are highlighted upon misclick?	[Krajcovic et al., 2025]	Eslováquia	2025
E02	Validation of information architecture: Cross-methodological comparison of tree testing variants and prototype user testing	[Kuric et al., 2025]	Eslováquia	2025
E03	TESY: A Usability Test-Driven Prototyping Assistant Connecting Designers with Crowd-Testers	[Kretzer e Maedche, 2025]	Alemanha	2025
E04	Misty: UI Prototyping Through Interactive Conceptual Blending	[Lu et al., 2025]	Estados Unidos	2025
E05	GenieWizard: Multimodal App Feature Discovery with Large Language Models	[Yang et al., 2025]	Estados Unidos	2025
E06	Closing the Loop between User Stories and GUI Prototypes: An LLM-Based Assistant for Cross-Functional Integration in Software Development	[Kretzer et al., 2025]	Alemanha	2025
E07	NLDesign: A UI Design Tool for Natural Language Interfaces	[Zhang et al., 2024]	China	2024
E08	Interlinking User Stories and GUI Prototyping: A Semi-Automatic LLM-Based Approach	[Kolthoff et al., 2024]	Alemanha	2024
E09	From Models to Interfaces: Leveraging the Two-Hemisphere Model for Automated UI Generation	[Babris e Nikiforova, 2024]	Letônia	2024
E10	User Centered Design Methods in Software Development: A Case Study in a Peruvian Office of Financial Aid and Scholarships	[Trujillo et al., 2024]	Peru	2024
E11	UIBee: An improved deep instance segmentation and classification of UI elements in wireframes	[Kazangirler et al., 2023]	Turquia	2023
E12	Towards Designing Intuitive Mobile UIs Considering Tapping Behaviour of Elderly Users	[Reza et al., 2022]	Bangladesh	2023
E13	What do Researchers Need when Implementing Novel Interaction Techniques?	[Raffaillac e Huot, 2022]	França	2022
E14	A framework for evaluating and improving requirements specifications based on the developers and testers perspective	[Oran et al., 2021]	Brasil	2021
E15	image2emmet: Automatic code generation from web user interface image	[Xu et al., 2021]	China	2021
E16	Cocapture: Effectively communicating ui behaviors on existing websites by demonstrating and remixing	[Chen et al., 2021b]	Canadá	2021
E17	Towards the integration of user interface prototyping and model-based development	[Machado e Campos, 2021]	Portugal	2021
E18	Design Principles for a Crowd-Based Prototype Validation Platform	[Gottschalk et al., 2021]	Alemanha	2021
E19	Prototypes as starting point in MDE: Proof of concept	[Sánchez-Villarín et al., 2020]	Espanha	2020
E20	MoCaDiX: Designing cross-device user interfaces of an information system based on its class diagram	[Vanderdonck e Nguyen, 2019]	Bélgica	2019
E21	DUSM: A Method for Requirements Specification and Refinement Based on Disciplined Use Cases and Screen Mockups	[Reggio et al., 2018]	Itália	2018
E22	A roadmap for user interface design of interactive systems: An approach based on a triad of patterns	[Ruíz et al., 2018]	Colômbia	2018
E23	MODUS: Model-based user interfaces prototyping	[Machado et al., 2017]	Portugal	2017
E24	Generating a language-independent graphical user interfaces from UML models	[Shatnawi e Shatnawi, 2016]	Jordânia	2016
E25	Software prototypes: Enhancing the quality of requirements engineering process	[Suranto, 2015]	Indonésia	2015
E26	Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering	[Rivero et al., 2014]	Argentina	2014
E27	Automated generation of user-interface prototypes based on controlled natural language description	[Juárez-Ramírez et al., 2014]	México	2014
E28	Generating interface prototype for EnergyPlus IDD file using unified modeling language and coloured petri-nets	[Abdeldjebbar e Azeddine, 2012]	Argélia	2012
E29	Rapid and rich prototyping: Proof of concepts for experience	[Hennipman et al., 2008]	Holanda	2008
E30	User interface prototyping based on UML scenarios and high-level petri nets	[Elkoutbi e Keller, 2000]	Marrocos	2000

evidenciam o papel dos protótipos na mediação da comunicação, da colaboração e do alinhamento entre os envolvidos no desenvolvimento de software. A dimensão **Organizacional** reúne dois fatores (F20–F21), associados a decisões de gestão, como custos, prazos e retrabalho. A Tabela 6 apresenta a classificação dos fatores segundo as dimensões do *framework* TOE e os estudos que os reportaram.

### 5.1.1 Dimensão ambiental

A dimensão ambiental do *framework* TOE abrange fatores humanos e sociais relacionados à adoção e ao uso de tecnologias. Neste estudo, essa dimensão é analisada no contexto da utilização de protótipos de interface do usuário, com ênfase na comunicação, no alinhamento entre *stakeholders* e na construção de entendimento compartilhado ao longo do desenvolvimento de software.

F01 aborda a comunicação de requisitos, um desafio recorrente na literatura (E04, E06, E10, E14, E16, E22, E24, E29). Esse fator é ilustrado por E04, que apresenta o sistema *Misty*, voltado à redução das barreiras entre *designers* e desenvolvedores por meio da integração direta de exemplos visuais ao código da interface, o que favorece uma compreensão mais imediata das decisões de *design*. De forma complementar, F02 trata do alinhamento e do consenso entre os *stakeholders* (E02, E03, E04, E10, E14, E28), exemplificado por E14, que propõe um método iterativo de validação contínua com usuários, a partir de protótipos derivados de cenários baseados na linguagem de modelagem unificada (do inglês, *Unified Modeling Language* - UML).

F03 evidencia a importância da representação visual da interface (E04, E09, E11, E16, E24, E25), contribuindo para a clarificação de ideias e para a redução de falhas de comunicação. Esse fator pode ser observado em E09, que propõe um modelo para a geração automatizada de protótipos, permitindo a rápida visualização e avaliação de diferentes estilos, cores e *layouts*. Em complemento, F04 destaca o papel dos protótipos na promoção da consistência e na redução de ambiguidades (E04, E08, E14, E16, E22, E24, E30), exemplificado pelo *framework ReComp*, apresentado em E14, que demonstra reduções significativas de defeitos de omissão e de ambiguidade nos requisitos.

F05 ressalta o uso de protótipos como apoio à documentação e ao compartilhamento de conhecimento (E04, E06, E14, E16, E25, E29). Esse fator é evidenciado em E25, que destaca o potencial dos protótipos como insumos para a documentação e o treinamento de usuários finais. F06 aborda a flexibilidade na expressão da interface do usuário (E04, E13, E14, E24, E28), indicando que limitações técnicas das ferramentas de prototipagem podem restringir a representação de interações complexas. Esse fator é discutido em E13, que aponta que a ausência de extensibilidade em *frameworks* existentes frequentemente exige adaptações não planejadas, comprometendo a fidelidade à visão original do projeto.

### 5.1.2 Dimensão tecnológica

A dimensão tecnológica do *framework* TOE abrange fatores relacionados a ferramentas, métodos, técnicas e abordagens que sustentam a adoção e o uso de tecnologias. Neste estudo,

**Tabela 6. Fatores que influenciam o uso de protótipos de interface no desenvolvimento de software segundo o *framework* TOE**

Dimensão Ambiental		
ID	Fator de Influência	Estudos
F01	Comunicação de requisitos	E04, E06, E10, E14, E16, E22, E24, E29
F02	Alinhamento e consenso	E02, E03, E04, E10, E14, E28
F03	Representação visual da interface	E04, E09, E11, E16, E24, E25
F04	Consistência e redução de ambiguidade	E04, E08, E14, E16, E22, E24, E30
F05	Apoio à documentação e conhecimento	E04, E06, E14, E16, E25, E29
F06	Flexibilidade na expressão da interface do usuário	E04, E13, E14, E24, E28
Dimensão Tecnológica		
ID	Fator de Influência	Estudos
F07	Verificação das expectativas do cliente	E04, E10, E24, E28, E29
F08	Validação de requisitos (funcionais e não funcionais)	E04, E08, E10, E14, E20, E24, E28
F09	Identificação e correção de problemas de usabilidade	E01, E03, E04, E10, E12, E14, E20
F10	Avaliação da arquitetura da informação	E02, E23
F11	Geração e avaliação de histórias de usuário	E06, E08, E14, E21, E29
F12	Iteração e prototipagem ágil	E06, E08, E10, E18, E24, E28, E29
F13	Ferramentas de prototipagem e <i>design</i>	E01, E02, E03, E05, E06, E07, E08, E09, E13, E14, E19, E23, E24, E25, E26, E28, E29
F14	Automação de tarefas manuais	E04, E05, E06, E09, E11, E15, E19, E22, E26, E27
F15	Desenvolvimento de interface do usuário baseado em modelos	E08, E09, E12, E17, E19, E21, E23, E25, E26, E27, E29
F16	Uso de diagramas da UML	E19, E23, E24, E25, E26, E27, E30
F17	Abordagens de geração automática de interface do usuário	E06, E07, E08, E09, E12, E17, E23, E25, E26, E27, E29, E30
F18	Reutilização e aplicação de padrões de <i>design</i>	E03, E04, E09, E12, E19, E21, E23, E25, E26, E29
F19	Integração com LLM	E04, E05, E06, E07, E08, E19
Dimensão Organizacional		
ID	Fator de Influência	Estudos
F20	Redução de custos e tempo de desenvolvimento	E05, E06
F21	Redução de erros e retrabalho	E01, E03, E05, E06, E14, E20, E24

essa dimensão é analisada no contexto da utilização de protótipos de interface do usuário, refletindo a dependência desse uso em relação à disponibilidade e à maturidade das soluções tecnológicas no desenvolvimento de software.

F07 trata da verificação das expectativas do cliente (E04, E10, E24, E28, E29), destacando o uso de protótipos como instrumentos para validar precocemente se a solução visual está alinhada aos objetivos de negócio. Esse fator é discutido em E29, que exemplifica o uso de uma “tela única” (*whiteboard metaphor*) para manter todos os artefatos visíveis, facilitando a construção de uma visão de *design* compartilhada e acelerando a tomada de decisão junto ao cliente. De forma complementar, F08 aborda a validação de requisitos funcionais e não funcionais (E04, E08, E10, E14, E20, E24, E28), assegurando que as funcionalidades e as restrições estejam adequadamente representadas no *design* da interface. Esse fator é evidenciado no E24, que propõe a geração de interfaces a partir de modelos UML para revelar erros e omissões nos requisitos funcionais, ainda no estágio de elicitação.

F09 refere-se à identificação e correção de problemas de usabilidade (E01, E03, E04, E10, E12, E14, E20), enfatizando o papel dos testes com usuários na detecção de falhas de interação. Esse fator pode ser observado em E01, que investiga a validade de testes de usabilidade realizados com protótipos nos quais “hotspots” clicáveis são destacados após um clique incorreto, analisando como essa técnica influencia o comportamento e a satisfação dos usuários. Em complemento, F10 trata da avaliação da arquitetura da informação

(E02, E23), concentrando-se na validação da estrutura de navegação e da organização do conteúdo da interface. Esse fator é exemplificado em E02, que realiza uma comparação metodológica entre variantes de *tree testing* e testes de usuário com protótipos para validar a arquitetura da informação de sistemas de software.

No contexto das práticas de desenvolvimento, F11 destaca a geração e avaliação de histórias de usuário (E06, E08, E14, E21, E29), conectando diretamente a prototipagem às práticas ágeis. Esse fator é ilustrado em E06, que apresenta um assistente, baseado em grandes modelos de linguagem (do inglês, *Large Language Models* - LLM), capaz de avaliar se um requisito, expresso como história de usuário, foi implementado no protótipo e de identificar os componentes de interface responsáveis por sua realização. De forma relacionada, F12 enfatiza a iteração e a prototipagem ágil (E06, E08, E10, E18, E24, E28, E29), evidenciando a importância de ciclos rápidos de criação, teste e refinamento. Esse fator é discutido em E18, que apresenta princípios de *design* para plataformas de validação de protótipos baseadas em multidão (*crowdsourcing*), permitindo ajustes iterativos com base no *feedback* contínuo de usuários.

No âmbito das ferramentas de suporte ao desenvolvimento, F13 aborda o uso de ferramentas de prototipagem e *design* (E01, E02, E03, E05, E06, E07, E08, E09, E13, E14, E19, E23, E24, E25, E26, E28, E29), ressaltando o papel central de plataformas como Figma, Sketch e Adobe XD no suporte ao trabalho colaborativo entre equipes. Esse fator é evidenciado em E03, que menciona o uso de ferramentas, como o Figma, para integrar testes de usabilidade diretamente no ambiente de prototipagem, tornando os resultados dos testes acionáveis para os *designers*. Em complemento, F14 trata da automação de tarefas manuais (E04, E05, E06, E09, E11, E15, E19, E22, E26, E27), destacando os ganhos de produtividade obtidos por meio da automação da geração de código e de artefatos de *design*. Esse fator é ilustrado em E05, no qual o sistema *GenieWizard* utiliza LLMs para apoiar a descoberta de funcionalidades e a realização de testes de *layout* sem exigir implementação manual de código.

Outro conjunto relevante de fatores relaciona-se às abordagens baseadas em modelos para o desenvolvimento de interfaces. F15 trata do desenvolvimento de interfaces do usuário baseado em modelos (E08, E09, E12, E17, E19, E21, E23, E25, E26, E27, E29), no qual modelos conceituais são utilizados para estruturar e orientar o processo de *design* da interface. Esse fator é exemplificado em E23, que apresenta a ferramenta *MODUS*, voltada à prototipagem evolutiva a partir de modelos arquiteturais da lógica de negócio. De forma complementar, F16 aborda o uso de diagramas da UML (E19, E23, E24, E25, E26, E27, E30), evidenciando a utilização de artefatos tradicionais da engenharia de software como base para apoiar a especificação e a validação de interfaces. Esse fator pode ser observado em E30, onde diagramas de sequência da UML são transformados em especificações de Redes de Petri coloridas para apoiar a geração de protótipos executáveis.

A automação avançada é abordada em F17, que trata de abordagens de geração automática de interface do usuário (E06, E07, E08, E09, E12, E17, E23, E25, E26, E27, E29, E30), nas quais modelos são utilizados como insumo para gerar automaticamente artefatos de interface ou código

executável. Esse fator é ilustrado em E09, que propõe uma solução baseada em Engenharia Direcionada por Modelos (MDE) para gerar automaticamente o código-fonte do *front-end* de protótipos de aplicações web a partir do modelo de “dois hemisférios”. É importante ressaltar que, embora o F17 concentre um volume expressivo de estudos voltados à eficiência técnica da geração de interfaces, não foram identificadas evidências ou interesses específicos nesses trabalhos voltados à investigação da prototipação para a validação da acessibilidade da interface de usuário. De forma relacionada, F18 trata da reutilização e aplicação de padrões de *design* (E03, E04, E09, E12, E19, E21, E23, E25, E26, E29), enfatizando o aproveitamento de soluções visuais e lógicas já consolidadas. Esse fator é exemplificado em E21, que apresenta o método DUSM, baseado em modelos de casos de uso disciplinados e *templates* de telas para garantir consistência e reduzir o esforço de *design*.

F19 aborda a integração com LLMs (E04, E05, E06, E07, E08, E19), apontando para uma tendência emergente na prototipagem de interfaces. Esse fator é evidenciado em E08, que demonstra o uso de técnicas de *prompting* para verificar se uma história de usuário está representada em um protótipo e recomendar componentes de interface adequados em HTML/CSS a partir de descrições em linguagem natural.

### 5.1.3 Dimensão organizacional

A dimensão organizacional do *framework* TOE contempla fatores relacionados à gestão e ao contexto organizacional de projetos de software. Neste estudo, essa dimensão é analisada no contexto da utilização de protótipos de interface do usuário, evidenciando como decisões estratégicas influenciam sua adoção e impactam a viabilidade econômica e o sucesso dos projetos.

F20 refere-se à redução de custos e tempo de desenvolvimento (E05, E06), destacando a prototipagem como um investimento estratégico que possibilita a identificação precoce de problemas e a aceleração do ciclo de desenvolvimento. Esse fator é ilustrado em E05, que ressalta que o uso de protótipos permite que *designers* resolvam questões de usabilidade ainda na fase de projeto, economizando recursos que seriam gastos em implementações baseadas em código que precisariam ser refeitas. De forma complementar, F21 aborda a redução de erros e retrabalho (E01, E03, E05, E06, E14, E20, E24), evidenciando como a identificação antecipada de falhas de usabilidade, requisitos e *design* evita correções dispendiosas em fases posteriores do projeto. Esse fator é exemplificado no *framework ReComP*, apresentado em E14, que auxilia na identificação de problemas de comunicação em artefatos de requisitos, como histórias de usuário e casos de uso, reduzindo significativamente a frequência de erros identificados por desenvolvedores e testadores.

Em conjunto, esses fatores demonstram que a utilização de protótipos contribui para decisões organizacionais mais eficientes, alinhando a qualidade técnica, a redução de riscos e o melhor controle de recursos nos projetos de software.

## 5.2 QP2 - Quais são as soluções (e.g., métodos, técnicas, ferramentas e práticas) utilizadas para a criação de protótipos de interface do usuário no desenvolvimento de software?

A Tabela 7 apresenta as principais soluções para a criação de protótipos de interface do usuário, conforme identificadas nos estudos selecionados. Tais soluções representam os diferentes recursos e estratégias empregados no processo, sendo categorizadas de acordo com sua natureza em **métodos, técnicas, ferramentas e práticas**. Os **métodos** dizem respeito aos *frameworks* estratégicos e abordagens estruturadas que orientam o processo; as **técnicas** compreendem os procedimentos específicos executados para criar ou validar um artefato; as **ferramentas** referem-se aos instrumentos de software e tecnologias que viabilizam a construção dos protótipos; e as **práticas** representam as convenções e formas de trabalho recomendadas para garantir a qualidade do processo [Vasconcelos e Oliveira, 2016].

**Tabela 7. Soluções para a criação de protótipos de interface do usuário**

ID	Solução	Tipo	Estudos
S01	Abordagem baseada em cenários e UML	Método	E24, E28, E30
S02	Abordagem baseada em modelo	Método	E09, E17, E20, E22, E23, E27
S03	<i>Design</i> centrado no usuário (DCU)	Método	E08, E17, E22
S04	Prototipagem iterativa/evolutiva	Método	E08, E09, E14, E21, E23
S05	Desenvolvimento ágil e engenharia de requisitos ágeis	Método	E13, E14, E26
S06	<i>Mockup-Driven Development</i> (MockupDD)	Método	E21, E26
S07	<i>Disciplined Use Cases with Screen Mockups</i> (DUSM)	Método	E21
S08	Automação da geração de interface de usuário	Método	E09, E11, E15, E17, E20, E27
S09	Protótipos de baixa fidelidade	Técnica	E01, E21
S10	Testes de usabilidade assíncronos e baseados em multidão	Técnica	E01, E03, E18
S11	Geração automatizada de componentes e <i>layouts</i> de interface de usuários	Técnica	E06, E08
S12	Geração de interfaces a partir de linguagem natural	Técnica	E07, E27
S13	Geração de código a partir de imagens de interface de usuário	Técnica	E11, E15
S14	Deteção e correspondência de histórias de usuário em protótipos	Técnica	E06, E08
S15	Figma	Ferramenta	E01, E03, E06, E09, E17, E18
S16	Adobe XD	Ferramenta	E01, E09, E12, E18
S17	Sketch	Ferramenta	E01, E09, E12
S18	Mockingbird	Ferramenta	E21, E26
S19	NLDesign	Ferramenta	E06, E27
S20	Uso de cenários e histórias de usuário	Prática	E08, E14, E21, E26, E28, E30
S21	Aplicação de padrões de <i>design</i> de interface de usuário	Prática	E22
S22	Criação e utilização de protótipos de diferentes fidelidades	Prática	E01, E21, E25

## 5.3 QP3 – Quais são os desafios encontrados na utilização de protótipos de interface do usuário no desenvolvimento de software?

Essa QP busca destacar os principais desafios e obstáculos na criação e utilização de protótipos de interface, bem como as lacunas existentes na literatura. Dentre os diversos desafios mapeados, alguns se destacaram pela alta frequência nos estu-

**Tabela 8. Principais Desafios na Criação e Utilização de Protótipos de UI**

ID	Desafios	Estudos
D01	Comunicação inadequada ou insuficiente	E14
D02	Ambiguidade, incompletude e inconsistência dos requisitos	E14, E24, E28
D03	Dificuldade em identificar informações essenciais	E14
D04	Volatilidade dos requisitos e esforço de sincronização	E06, E14
D05	Perda de rastreabilidade e informações	E19
D06	Requisitos incorretos e incompletos nos diagramas UML	E27
D07	Uso de termos próprios pelas partes interessadas	E28
D08	Fragmentação e baixa integração das ferramentas	E03, E06
D09	Esforço manual excessivo e custo/tempo elevado	E08, E11, E14, E30
D10	Automação limitada e interfaces de usuário rígidas	E17, E27
D11	Dificuldade em replicar detalhes e comportamentos interativos	E16
D12	Custos dos testes de usabilidade	E01, E06
D13	Distorção do comportamento do usuário	E01
D14	Lacuna de habilidades dos <i>designers</i>	E06, E28
D15	Código gerado não estruturado e difícil de manter	E09, E17

dos, como as dificuldades relacionadas à comunicação efetiva de requisitos entre as partes interessadas e o desafio contínuo da identificação e correção de problemas de usabilidade em estágios iniciais do projeto. Esses aspectos evidenciam a complexidade do processo de prototipação e a necessidade de abordagens que apoiem a colaboração entre os envolvidos no desenvolvimento. Além disso, demonstram que a prototipação também exerce um papel importante na validação e no refinamento dos requisitos do sistema. A Tabela 8 apresenta a consolidação de todos os desafios identificados e os estudos que os suportam.

## 6 Discussão

Esta RR teve como objetivo consolidar o estado da arte sobre a utilização de protótipos de interface do usuário no desenvolvimento de software. A análise dos 30 estudos primários confirma que a prototipagem ocupa um papel central no apoio à comunicação, validação e evolução de requisitos ao longo do desenvolvimento. Esse resultado converge com os achados de Garcia *et al.* [2017], que destacam o papel dos artefatos de *design* centrado no usuário como instrumentos de mediação entre equipes em ambientes ágeis.

Os resultados da QP1 permitiram identificar 21 fatores que influenciam a utilização de protótipos de interface do usuário, organizados segundo as dimensões Tecnológica, Ambiental e Organizacional do *framework* TOE. Observou-se a predominância de fatores associados à dimensão tecnológica (F07–F19), indicando que a utilização de protótipos está fortemente relacionada à disponibilidade e à maturidade das soluções empregadas no desenvolvimento de software. Em particular, destacam-se o papel das ferramentas de prototipagem e *design*, mecanismos de automação e abordagens voltadas à geração automática de interfaces, incluindo soluções recentes baseadas em grandes modelos de linguagem. Além disso, práticas tradicionais da engenharia de software, como o uso de modelos, diagramas e padrões de *design*, também aparecem como elementos estruturantes no uso da prototipagem.

Embora os fatores tecnológicos predominem, os fato-

res associados à dimensão ambiental (F01–F06) indicam que muitos dos benefícios da prototipagem estão relacionados a aspectos humanos e sociais do desenvolvimento de software. Esses resultados reforçam o papel dos protótipos como artefatos de mediação capazes de apoiar a comunicação entre equipes, o alinhamento entre *stakeholders* e a construção de entendimento compartilhado sobre os requisitos do sistema, aspecto também destacado por Garcia *et al.* [2017] no contexto de práticas de *design* centrado no usuário.

Em menor número, os fatores associados à dimensão organizacional (F20–F21) evidenciam que decisões relacionadas à adoção da prototipagem também estão vinculadas a aspectos de gestão de projetos. Em particular, os estudos indicam que o uso de protótipos pode contribuir para reduzir retrabalho, antecipar a identificação de problemas de *design* e melhorar a eficiência do processo de desenvolvimento, reforçando seu papel como instrumento estratégico no planejamento e na condução de projetos de software.

As soluções identificadas na QP2 mostram a coexistência entre métodos tradicionais e abordagens tecnologicamente mais avançadas. A literatura apresenta desde técnicas mais consolidadas, como prototipação baseada em modelos e cenários, até soluções voltadas à automação e ao uso de inteligência artificial. Essa diversidade corrobora os resultados de Garcia *et al.* [2017], que identificaram a presença simultânea de artefatos físicos e eletrônicos no processo de *design* centrado no usuário.

Entretanto, os resultados desta revisão indicam que a natureza das soluções eletrônicas evoluiu significativamente. Enquanto Freitas *et al.* [2020] observaram uma predominância de técnicas de prototipação em papel em contextos como aplicações de realidade aumentada, os estudos analisados nesta RR apontam para uma maior adoção de ferramentas digitais de alta fidelidade e de abordagens de geração automática de interfaces. A integração com LLMs sugere uma tendência recente de apoio automatizado à criação e evolução de protótipos, reduzindo o esforço manual associado ao *design* e à implementação de interfaces.

A análise da QP3 indica que, apesar da diversidade de soluções identificadas, a utilização de protótipos ainda enfrenta desafios relevantes. Entre os principais desafios estão problemas de comunicação entre as partes interessadas, ambiguidade e incompletude de requisitos e dificuldades na identificação de informações essenciais durante o processo de *design*. Esses resultados reforçam a importância dos protótipos como mecanismos de apoio à comunicação e à validação iterativa, aspecto também discutido por Garcia *et al.* [2017] no contexto de métodos ágeis.

Além disso, foram identificadas limitações técnicas relacionadas às ferramentas e práticas de prototipagem, incluindo fragmentação entre plataformas, esforço manual elevado e dificuldades na representação de interações complexas. Esses aspectos indicam que, embora as ferramentas tenham evoluído significativamente, sua integração ao fluxo de desenvolvimento ainda depende de adaptações metodológicas e organizacionais. Esse cenário converge com as observações de Larrea *et al.* [2024], que ressaltam que a eficácia da prototipagem depende do contexto do projeto, do nível de fidelidade adotado e do envolvimento das equipes no processo de desenvolvimento.

## 6.1 Ameaças à validade

Ao longo da realização deste estudo, foram adotadas medidas para mitigar possíveis ameaças à validade e reduzir seu impacto nos resultados. O protocolo desta RR seguiu as diretrizes propostas por Cartaxo *et al.* [2020] e as recomendações metodológicas de Kitchenham e Charters [2007] e Petersen *et al.* [2015].

Para garantir a validade descritiva, utilizou-se um formulário de extração padronizado e a ferramenta Microsoft Excel para o registro e consolidação dos dados. A extração foi conduzida por um pesquisador e, posteriormente, verificada por dois pesquisadores experientes em engenharia de software, o que assegurou a qualidade e minimizou vieses individuais.

A validade teórica foi considerada por meio da aplicação rigorosa de CI e CE em um processo de seleção de 6 etapas. A *string* de busca foi elaborada com base nas questões de pesquisa, sem restrições de tempo, permitindo a recuperação de estudos até maio de 2025.

A validade interpretativa foi considerada na análise dos dados, realizada por meio de síntese qualitativa e descritiva. A classificação dos 21 fatores de influência (QP1), por exemplo, inicialmente não foi pré-definida, mas surgiu de um processo indutivo de análise temática, em que fatores com propósitos semelhantes foram agrupados.

Quanto à generalização, os resultados obtidos refletem os estudos disponíveis na base de dados Scopus. Além das restrições de busca, este estudo apresenta limitações metodológicas intrínsecas ao formato de Revisão Rápida. Diferente de trabalhos secundários mais extensos, como a RSL, o protocolo de RR permite a simplificação de etapas para acelerar a entrega de evidências. O uso de uma única base de dados é uma característica desta RR, que visa prazos mais curtos. Adicionalmente, a exclusão de estudos não escritos em inglês (CE5) e de teses ou dissertações (CE4) restringe a abrangência. Portanto, essa limitação não compromete a análise realizada dos 30 estudos finais, apenas restringe sua generalização.

## 7 Conclusão

Este estudo apresentou uma revisão rápida da literatura (RR) sobre a utilização de protótipos de interface do usuário no desenvolvimento de software, sintetizando evidências de 30 estudos primários. A análise permitiu identificar 21 fatores que influenciam essa prática, organizados nas dimensões Tecnológica, Ambiental e Organizacional do *framework* TOE. Os resultados evidenciam a importância crescente da prototipagem como instrumento de apoio à engenharia de requisitos e ao alinhamento entre *stakeholders* ao longo do ciclo de vida do software. Observou-se ainda um aumento significativo de publicações a partir de 2021, indicando o interesse recente da comunidade científica pelo tema.

Os resultados também mostram que a evolução das soluções de prototipagem acompanha o avanço das tecnologias de desenvolvimento de software. Métodos consolidados, como abordagens baseadas em modelos, convivem com ferramentas de *design* digital amplamente utilizadas e com soluções emergentes voltadas à automação e ao uso de inteligência artificial, como geração automática de interfaces e integração com grandes modelos de linguagem. Esse cenário reforça o papel da prototipagem não apenas como mecanismo de re-

presentação visual da interface, mas como um componente ativo no processo de validação de requisitos, avaliação de usabilidade e apoio à comunicação entre equipes.

Do ponto de vista prático, esta RR oferece uma visão consolidada dos fatores que influenciam a adoção de protótipos e das soluções utilizadas na literatura. Os resultados sugerem que a escolha de ferramentas e métodos de prototipagem deve considerar não apenas critérios tecnológicos, mas também sua capacidade de apoiar a comunicação, a validação de requisitos e a construção de entendimento compartilhado entre os envolvidos no desenvolvimento de software.

Como trabalhos futuros, recomenda-se investigar de forma mais aprofundada o impacto das soluções emergentes de prototipagem, especialmente aquelas baseadas em automação e em LLMs, sobre aspectos como comunicação entre *stakeholders*, qualidade dos requisitos e redução de ambiguidades. Além disso, futuras pesquisas podem explorar abordagens de avaliação mais sistemáticas que permitam mensurar os benefícios organizacionais da prototipagem, incluindo sua influência na redução de retrabalho, no tempo de desenvolvimento e na qualidade dos artefatos produzidos.

## Declarações complementares

### Financiamento

Este estudo recebeu apoio financeiro através de uma bolsa concedida pela Fundação Amazônia de Amparo a Estudos e Pesquisas (FAPESPA), o que possibilitou a execução integral da pesquisa e a realização do estudo apresentado.

### Contribuições dos autores

Ana Carolina contribuiu para investigação, a aplicação da metodologia, a sistematização dos dados e a redação do rascunho original. Juntamente com Fabrício Garcia e Paulo Malcher que também foram responsáveis pela validação do estudo, revisão crítica do manuscrito e aprovação final do material. Todos os autores leram e aprovaram a versão final do manuscrito.

### Conflitos de interesse

Os autores declaram não ter conflitos de interesse.

### Disponibilidade de dados e materiais

Os dados completos e os materiais utilizados para as análises e a revisão rápida estão disponíveis no Material Suplementar em <https://doi.org/10.5281/zenodo.19075422>.

### Agradecimentos

Durante a preparação deste trabalho, os autores usaram o ChatGPT-5.2 para aprimorar a redação e estrutura do texto, além de corrigir erros ortográficos e gramaticais. Após utilizar esta ferramenta/serviço, os autores revisaram e editaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo do artigo publicado.

## Referências

- Abdeldjebbar, B. e Azeddine, B. (2012). Generating interface prototype for energyplus idd file using unified modeling language and coloured petri-nets. *Energy Procedia*, 18:1458–1484. DOI: 10.1016/j.egypro.2012.05.164.
- Babris, K. e Nikiforova, O. (2024). From models to interfaces: Leveraging the two-hemisphere model for automated ui generation. In *2024 IEEE 65th International Scientific Conference on Information Technology and Management*

- Science of Riga Technical University (ITMS)*, pages 1–6. IEEE. DOI: 10.1109/ITMS64072.2024.10741944.
- Bjarnason, E., Lang, F., e Mjöberg, A. (2021). A model of software prototyping based on a systematic map. In *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ESEM '21, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3475716.3475772.
- Brahmastra, N. W., Saputra, M., e Witjaksono, W. (2024). Factors influencing erp adoption in the hospitality industry: A toe framework analysis. In *2024 2nd International Symposium on Information Technology and Digital Innovation (ISITDI)*, pages 90–95. DOI: 10.1109/ISITDI62380.2024.10796659.
- Cartaxo, B., Pinto, G., e Soares, S. (2018). The role of rapid reviews in supporting decision-making in software engineering practice. In *International Conference on Evaluation and Assessment in Software Engineering 2018*, pages 24–34. DOI: 10.1145/3210459.3210462.
- Cartaxo, B., Pinto, G., e Soares, S. (2020). *Rapid reviews in software engineering*, pages 357–384. Springer. DOI: 10.1007/978-3-030-32489-613.
- Chen, Y., Cao, S., Ma, G., e Li, T. (2021a). Saip: A prototype tool for semi-automatic user interface prototyping. In *Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium*, volume 2858 of *CEUR-WS*, pages 1–9, Essen, Germany. CEUR Workshop Proceedings. Disponível em: <https://ceur-ws.org/Vol-2857/pt4.pdf>.
- Chen, Y., Lee, S. W., e Oney, S. (2021b). Cocapture: Effectively communicating ui behaviors on existing websites by demonstrating and remixing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14. DOI: 10.1145/3411764.3445573.
- Deininger, M., Daly, S. R., Sienko, K. H., e Lee, J. C. (2017). Novice designers' use of prototypes in engineering design. *Design Studies*, 51:25–65. DOI: 10.1016/j.destud.2017.04.002.
- Elkoutbi, M. e Keller, R. K. (2000). User interface prototyping based on uml scenarios and high-level petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 166–186. Springer. DOI: 10.1007/3-540-44988-4\_11.
- Freitas, G., Pinho, M. S., Silveira, M. S., e Maurer, F. (2020). A systematic review of rapid prototyping tools for augmented reality. In *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pages 199–209. IEEE. DOI: 10.1109/SVR51698.2020.00041.
- Garcia, A., da Silva, T. S., e Silveira, M. S. (2017). Artifacts for agile user-centered design: A systematic mapping. In *Proceedings of the 50th Hawaii International Conference on System Sciences (HICSS)*, pages 5858–5867, Waikoloa Village, Hawaii, USA. IEEE Computer Society. DOI: 10.24251/HICSS.2017.706.
- Gottschalk, S., Aziz, M. S., Yigitbas, E., e Engels, G. (2021). Design principles for a crowd-based prototype validation platform. In *International Conference on Software Business*, pages 205–220. Springer. DOI: 10.1007/978-3-030-91983-2\_16.
- Hennipman, E.-J., Oppelaar, E.-J. R., van der Veer, G. C., e Bongers, B. (2008). Rapid and rich prototyping: proof of concepts for experience. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction*, pages 1–6. DOI: 10.1145/1473018.1473053.
- Jais, R., Ngah, A. H., Rahi, S., Rashid, A., Ahmad, S. Z., e Mokhlis, S. (2024). Chatbots adoption intention in public sector in malaysia from the perspective of toe framework. the moderated and mediation model. *Journal of Science and Technology Policy Management*. DOI: 10.1108/JSTPM-04-2024-0151.
- Juárez-Ramírez, R., Huertas, C., e Inzunza, S. (2014). Automated generation of user-interface prototypes based on controlled natural language description. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, pages 246–251. IEEE. DOI: 10.1109/COMPSACW.2014.44.
- Kazangirler, C. B., Özcan, C., e Tekin, B. Y. (2023). Uibee: An improved deep instance segmentation and classification of ui elements in wireframes. *Turkish Journal of Electrical Engineering and Computer Sciences*, 31(3):516–532. DOI: 10.55730/1300-0632.3999.
- King, V. J., Stevens, A., Nussbaumer-Streit, B., Kamel, C., e Garritty, C. (2022). Paper 2: Performing rapid reviews. *Systematic Reviews*, 11(1):151. DOI: 10.1186/s13643-022-02011-5.
- Kitchenham, B. e Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Evidence-Based Software Engineering (EBSE) Project. Disponível em: <https://www.researchgate.net/publication/258968007>.
- Kolthoff, K., Kretzer, F., Bartelt, C., Maedche, A., e Ponzetto, S. P. (2024). Interlinking user stories and gui prototyping: A semi-automatic llm-based approach. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 380–388. IEEE. DOI: 10.1109/RE59067.2024.00045.
- Koscianski, A. e dos Santos Soares, M. (2007). *Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. Novatec Editora. Disponível em : <https://www.researchgate.net/publication/353342399>.
- Krajcovic, M., Demcak, P., e Kuric, E. (2025). Is usability testing valid with prototypes where clickable hotspots are highlighted upon misclick? *Journal of Systems and Software*, 226:112446. DOI: 10.1016/j.jss.2025.112446.
- Kretzer, F., Kolthoff, K., Bartelt, C., Ponzetto, S. P., e Maedche, A. (2025). Closing the loop between user stories and gui prototypes: an llm-based assistant for cross-functional integration in software development. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19. DOI: 10.1145/3706598.3713932.
- Kretzer, F. e Maedche, A. (2025). Tesy: A usability test-driven prototyping assistant connecting designers with crowd-testers. *Proceedings of the ACM on Human-Computer Interaction*, 9(2):1–24. DOI: 10.1145/3711082.
- Kuric, E., Demcak, P., e Krajcovic, M. (2025). Validation of information architecture: Cross-methodological com-

- parison of tree testing variants and prototype user testing. *Information and Software Technology*, 183:107740. DOI: 10.1016/j.infsof.2025.107740.
- Larrea, V., Silveira, M. S., e Da Silva, T. (2024). The use of prototypes as a tool in agile software development. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24*, page 842–849, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3605098.3636064.
- Lu, Y., Leung, A., Swearngin, A., Nichols, J., e Barik, T. (2025). Misty: Ui prototyping through interactive conceptual blending. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17. DOI: 10.1145/3706598.3713924.
- Machado, C. e Campos, J. C. (2021). Towards the integration of user interface prototyping and model-based development. In *2021 International Conference on Graphics and Interaction (ICGI)*, pages 1–8. IEEE. DOI: 10.1109/ICGI54032.2021.9655284.
- Machado, M., Couto, R., e Campos, J. C. (2017). Modus: Model-based user interfaces prototyping. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pages 111–116. DOI: 10.1145/3102113.3102146.
- Manzo, B. F., Caldas, M. M., Alves, T. F., Tourinho, F. S. V., e Preis, L. C. (2022). Prototipação e validação: não é só ciência, é experiência, facilidade e dinamismo. In *Desenvolvimento de Tecnologias em Pesquisa e Saúde: Da Teoria à Prática*, volume 1, pages 122–137. Editora Científica Digital. DOI: 10.37885/220408593.
- Nahedh Alsehani, F., Abdul Wahab, A. W. B., e Shuib, L. (2024). Factors influencing social media adoption in government organizations: A toe framework and pls-sem model analysis. *IEEE Access*, 12:158150–158173. DOI: 10.1109/ACCESS.2024.3478337.
- OLIVEIRA NETTO, A. A. d. (2004). Ihc interação humano computador-modelagem e gerência de interfaces com o usuário. *Florianópolis: VisualBooks*. Disponível em: <https://www.travessa.com.br/ihc-modelagem-e-gerencia-de-interfaces-com-o-usuario/artigo/a3fe9b47-b77e-4ab8-9ded-d12e647f6e8d?srsId=AfmBOoq1o34arxP-cjmk6j9Qm1cgKT9BCuWKl67uFrnz0WjreXq1ayE6>.
- Oran, A. C., Santos, G., Gadelha, B., e Conte, T. (2021). A framework for evaluating and improving requirements specifications based on the developers and testers perspective. *Requirements Engineering*, 26(4):481–508. DOI: 10.1007/s00766-021-00352-6.
- Pernice, K. (2016). Ux prototypes: Low fidelity vs. high fidelity. *Nielsen Norman Group*, 18:12. Disponível em : <https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/>.
- Petersen, K., Vakkalanka, S., e Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18. DOI: 10.1016/j.infsof.2015.03.007.
- Raffaillac, T. e Huot, S. (2022). What do researchers need when implementing novel interaction techniques? *Proceedings of the ACM on Human-Computer Interaction*, 6(EICS):1–30. DOI: 10.1145/3532209.
- Reggio, G., Leotta, M., Ricca, F., e Clerissi, D. (2018). Dism: A method for requirements specification and refinement based on disciplined use cases and screen mockups. *Journal of Computer Science and Technology*, 33(5):918–939. DOI: 10.1007/s11390-018-1866-8.
- Reza, M. R., Aadeeb, M. S., Saleh, S. B., e Islam, M. N. (2022). Towards designing intuitive mobile uis considering tapping behaviour of elderly users. In *International Conference on Design and Digital Communication*, pages 148–160. Springer. DOI: 10.1007/978-3-031-20364-0\_14.
- Ricca, A. (2014). Mockups as a tool for software requirements elicitation: A systematic mapping study. In *2014 IEEE 12th International Conference on Industrial Informatics (INDIN)*, pages 1046–1051. DOI: 10.1145/2245276.2231945.
- Rivero, J. M., Grigera, J., Rossi, G., Luna, E. R., Montero, F., e Gaedke, M. (2014). Mockup-driven development: providing agile support for model-driven web engineering. *Information and Software Technology*, 56(6):670–687. DOI: 10.1016/j.infsof.2014.01.011.
- Ruiz, A., Giraldo, W. J., Geerts, D., e Arciniegas, J. L. (2018). A roadmap for user interface design of interactive systems: an approach based on a triad of patterns. In *International Conference of Design, User Experience, and Usability*, pages 223–240. Springer. DOI: 10.1007/978-3-319-91797-9\_16.
- Sánchez-Villarrín, A., Santos-Montaña, A., Koch, N., e Casas, D. L. (2020). Prototypes as starting point in mde: Proof of concept. In *WEBIST*, pages 365–372. DOI: 10.5220/0010213403650372.
- Shatnawi, A. e Shatnawi, R. (2016). Generating a language-independent graphical user interfaces from uml models. *Int. Arab J. Inf. Technol.*, 13(6B):1039–1044. Disponível em: <https://www.researchgate.net/publication/299784330>.
- Soares, B. C. (2008). Requisitos para utilização de prototipagem evolutiva nos processos de desenvolvimento de software baseado na web. *Belo Horizonte: UFMG*. Disponível em: <https://homepages.dcc.ufmg.br/rodolfo/dcc823-2-07/Entrega4/Bruno4.pdf>.
- Suranto, B. (2015). Software prototypes: Enhancing the quality of requirements engineering process. In *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, pages 148–153. IEEE. DOI: 10.1109/ISTMET.2015.7359019.
- Tornatzky, L. G. e Fleischer, M. (1990). *The Processes of Technological Innovation*. Lexington Books, Lexington, MA, USA. Disponível em: <https://archive.org/details/processesoftechn0000torn>.
- Trujillo, A., Martínez, H., Flores, J., Sabogal, F., Gonzales, F., e Paz, F. (2024). User centered design methods in software development: a case study in a peruvian office of financial aid and scholarships. In *2024 IEEE ANDESCON*, pages 1–6. IEEE. DOI: 10.1109/ANDESCON61840.2024.10755895.
- Vanderdonckt, J. e Nguyen, T.-D. (2019). Mocadix: Designing cross-device user interfaces of an information system based on its class diagram. *Proceedings of the ACM on human-computer interaction*, 3(EICS):1–40. DOI: 10.1145/3331159.
- Vasconcelos, A. M. L. d. e Oliveira, S. R. B. (2016). *Qualidade, Gestão e Processos de Software*. Editora UFPE,

Recife.

- Xu, Y., Bo, L., Sun, X., Li, B., Jiang, J., e Zhou, W. (2021). image2emmet: Automatic code generation from web user interface image. *Journal of Software: Evolution and Process*, 33(8):e2369. DOI: 10.1002/smr.2369.
- Yang, J., Shi, Y., Gu, C., Zheng, Z., Jain, A., Li, T., Lam, M. S., e Landay, J. A. (2025). Geniewizard: Multimodal app feature discovery with large language models. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17. DOI: 10.1145/3706598.3714327.
- Zhang, T., Peiguo, F., Liu, J., Zhang, Y., e Chen, X. (2024). Nldesign: A ui design tool for natural language interfaces. In *Proceedings of the ACM Turing Award Celebration Conference-China 2024*, pages 153–158. DOI: 10.1145/3674399.3674455.