

# rTiWIM: Uma Ferramenta para Elicitação e Validação de Requisitos de Software Utilizando Linguagem de Interface

João Victor R. Araújo, Carla Bezerra, Andréia Libório Sampaio

Universidade Federal do Ceará, Quixadá, CE, Brasil  
Programa de Pós-Graduação em Computação - PCOMP  
victoraraujo2504@gmail.com, carlailane@ufc.br, andreia.ufc@gmail.com

**Abstract.** O processo de levantamento de requisitos é uma das etapas fundamentais no ciclo de vida de um produto de software. No contexto de desenvolvimento de software, há cada vez mais a necessidade de ferramentas que apoiem os *stakeholders* no processo de elicitação e validação dos requisitos do sistema. A participação do usuário final nesse processo consolida a comunicação entre as partes interessadas para novas descobertas e problemáticas que o sistema possa apresentar. Dessa forma, este artigo propõe uma ferramenta de apoio aos *stakeholders* do sistema, denominada rTiWIM, que utilizará o modelo de comunicação TiWIM para elicitar e validar requisitos de software em sistemas web através da linguagem de interface. A ferramenta irá dispor das seguintes funcionalidades: (i) cadastro de usuários, (ii) validação de páginas, (iii) cadastro de páginas para validação e (iv) verificação das validações realizadas por usuários finais. Para validar a ferramenta, foi realizado um grupo focal com desenvolvedores e profissionais da área. Com base nos argumentos dos grupos, foi possível identificar melhorias e fortalecer a importância desta ferramenta dentro deste cenário.

**Keywords:** Elicitação de Requisitos · Validação de Requisitos · Linguagem de Interface.

## 1 Introdução

A participação do usuário final para levantar requisitos de software tem uma grande relevância dentro do contexto das informações coletadas para construção do software [6, 8, 17]. Apesar de encontrarmos várias técnicas tradicionais que auxiliam nesse processo de coleta de informação, ainda é possível encontrar problemas e lacunas de comunicação entre os usuários e engenheiros de software. Neste sentido, várias pesquisas identificam que requisitos mal elicitados e validados podem levar ao fracasso de projetos de software que não dão importância ao que os usuários finais ou clientes desejam transparecer [3, 2, 8].

Outro fator que confirma a importância de uma boa validação de software, é a utilização do usuário final dentro do processo de implementação do sistema, que vem sendo considerado com bastante frequência por pesquisas científicas

[19, 12]. Segundo [7, 21] a linguagem de interface possibilita e ajuda no aumento das expressões durante o fluxo de análise visual. Esse tipo de abordagem pode flexibilizar e aumentar as formas de coleta de informações de forma natural das necessidades dos usuários finais. Isso demonstra a relevância desse estudo e a participação do usuário no contexto de desenvolvimento de software [5].

O presente artigo visa contemplar a apresentação da ferramenta rTiWIM, que pretende auxiliar na elicitação de requisitos e validação de sistemas web em desenvolvimento, no qual permite que o usuário se expresse utilizando a linguagem de interface (UIL)<sup>1</sup> que para [9] é uma forma mais fácil do usuário final processar as informações que o computador está representando. Dessa forma, a aplicação irá fornecer um ambiente colaborativo para usuários e desenvolvedores do sistema, que possibilita a comunicação dos *stakeholders*. A ferramenta é baseada no modelo de comunicação TiWIM [1], que dá subsídio a principal forma de interação entre a equipe de desenvolvimento. A ferramenta foi validada por meio de um grupo focal [14], no qual desenvolvedores e profissionais da área realizaram validações das funcionalidades da ferramenta, e foram divididos em três grupos: **(i) Cliente**, **(ii) Desenvolvedores** e **(iii) Observadores**. A partir da validação da ferramenta, foram obtidas diversas melhorias para ferramenta, bem como, fortalecer a importância dessa temática.

## 2 Referencial Teórico

### 2.1 Modelo TiWIM

O modelo TiWIM [1] servirá como base para construção da ferramenta. O modelo propõe um ambiente que visa intensificar a comunicação entre o usuário final e os desenvolvedores do sistema. Essa proposta foi contemplada pelo desenvolvimento de um *plugin* para o navegador Firefox (primoTiWIM)<sup>2</sup>. A ferramenta possibilita modificar sistemas web por meio da sua interface, servindo como base para nosso instrumento de pesquisa. No entanto, o TiWIM não possibilita que os *stakeholders* do sistema possam utilizar o *plugin* para elicitar e validar requisitos de *software* através das modificações. O modelo tem como principal objetivo proporcionar um ambiente de discussão que ofereça aos usuários uma nova possibilidade de se expressar através da linguagem de interface (UIL). Assim, o usuário pode dizer o que deseja alterar e evoluir no sistema de uma forma mais detalhada, podendo inserir novos elementos na interface ou remover elementos já existentes, além de anotações contextualizadas na interface, como é exemplificado na Figura 1.

<sup>1</sup> Linguagem de interface vem do acrônimo User Interface Language (UIL), o qual é bastante evidenciado em pesquisas de IHC - Interação Humano Computador. Dessa forma asilaremos neste instrumento de pesquisa como referência a linguagem de interface.

<sup>2</sup> <https://github.com/FagnerSI/tiwim-plugin>

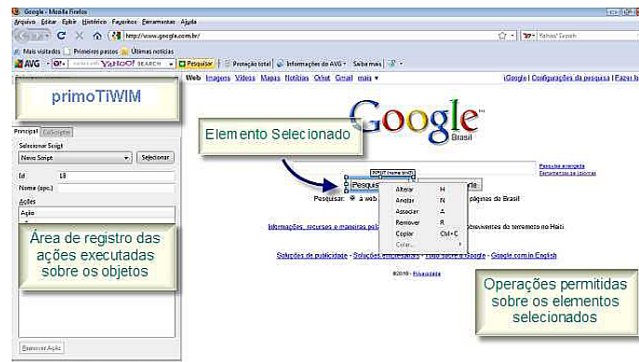


Figura 1: Ferramenta primoTiWIM [1]

## 2.2 Ferramentas para Levantamento de Requisitos

A elicitação de requisitos é um processo complexo que envolve inúmeras técnicas e ferramentas para realização do mesmo [10]. Esta seção trará algumas ferramentas baseadas no modelo em espiral de Engenharia de Requisitos (ER) do [22, 20]. Cada ferramenta pode oferecer um suporte específico determinado pela técnica de implementação que está sendo utilizada pela equipe de desenvolvimento. Nesse sentido, a escolha da ferramenta pode ser de extrema importância dentro do ambiente de elicitação de requisitos.

De acordo com [22, 20], as principais atividades da ER são: **(1) Elicitação e análise:** é o processo de captar as principais informações e premissas da parte interessada do projeto de forma que essas informações possam dar subsídio aos requisitos do sistema; **(2) Estudo de viabilidade:** visa verificar se o projeto de software é realmente viável dentro das perspectivas do usuário final, nesse sentido esta etapa deve ser feita para evitar possíveis perdas de tempo ou saturação de recursos; **(3) Especificação:** é forma de registrar as informações coletados na fase anterior de uma forma mais específica; **(4) Validação:** verifica se as funcionalidades do sistema condizem com as especificações do usuário final, através de uma nova visão ou *feedback* dos mesmos.

Tabela 1: Ferramentas e técnicas de ER

Ferramentas	Elicitação	Viabilidade	Especificação	Validação	Usuário Final	Interface
iRequeri [18]	Sim	Não	Não	Sim	Sim	Sim
AbsFinder [16]	Sim	Não	Sim	Não	Não	Não
Apprentice [13]	Sim	Não	Não	Não	Não	Não
Fame [17]	Sim	Não	Não	Sim	Sim	Não

Na Tabela 1 é possível perceber que cada ferramenta não engloba totalmente todos as atividades elencadas por [22, 20], e também que existe algumas ferra-

mentas que não contemplam a proposta de utilizar o usuário final e interface do sistema como técnicas alternativas para validar e elicitar requisitos. Nesse sentido, podemos verificar que cada ferramenta escolhida dentro do processo de desenvolvimento de software deve ser aplicada de acordo com as necessidades dos *stakeholders*.

Outro fator bastante relevante, que também deve ser considerado através da Tabela 1, foi a análise através dos seguintes tópicos: **(5) Usuário final:** onde é verificado se o usuário final participa de forma direta no processo de levantamento de requisitos e **(6) Interface:** indica a utilização da linguagem de interface (UIL) da ferramenta para levantamento de requisitos. As ferramentas iRequeri [18] e Fame [17] são exemplos de ferramentas que possibilitam e oportunizam a participação do usuário dentro do processo de implementação. No entanto, somente a ferramenta iRequeri [18], auxilia o usuário a fazer alterações através da interface da aplicação.

### 3 rTiWIM

Este artigo apresenta a ferramenta rTiWIM<sup>3</sup>, que tem como objetivo validar e elicitar requisitos de software por meio da linguagem de interface (UIL), fornecendo um ambiente colaborativo para os desenvolvedores do sistema e usuário final. A ferramenta utiliza os princípios do modelo de comunicação TiWIM [1] e foi desenvolvida nas linguagens JavaScript, HTML, CSS e PHP<sup>4</sup>. A ferramenta possui licença da *General Public License* (GPL)<sup>5</sup>. Uma parte do módulo rTiWIM.js foi reutilizada do projeto para discutir alterações em sistemas WEB<sup>6</sup>, trabalho que também utiliza o modelo de comunicação TiWIM através de um *plugin* para navegadores Web. Nas próximas seções serão apresentadas a arquitetura, funcionalidades e a validação do rTiWIM.

#### 3.1 Arquitetura

A arquitetura do sistema é baseada no modelo em camadas, onde a primeira camada é caracterizada pela interface do usuário que pode ser o usuário final ou um desenvolvedor do sistema, em que poderá ser utilizado em qualquer navegador *desktop* para acessar a aplicação Web. Na próxima camada é abordada a lógica do negócio, onde a ferramenta rTiWIM recebe o *feedback* do usuário, seleciona o tipo de implementação (validação ou elicitação) pelos controladores da aplicação. O armazenamento ocorre na próxima camada de dados, onde ficam hospedados no servidor<sup>7</sup>, juntamente com a aplicação e o banco de dados relacional Mysql<sup>8</sup>, como mostra a Figura 2.

<sup>3</sup> <http://rtiwim.com.br/>

<sup>4</sup> <https://github.com/victor2504/rTiWIM>

<sup>5</sup> <https://www.gnu.org/licenses/gpl-3.0.html>

<sup>6</sup> <https://github.com/FagnerSI/tiwim-js>

<sup>7</sup> <https://www.hostinger.com.br/>

<sup>8</sup> <https://www.mysql.com/>

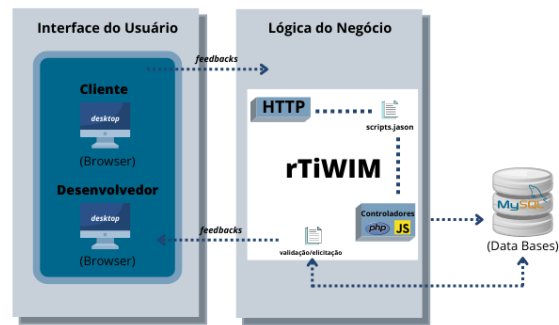


Figura 2: Arquitetura do sistema - rTiWIM

### 3.2 Funcionalidades da rTiWIM

A Figura 3 apresenta as principais funcionalidades da ferramenta rTiWIM para visão Cliente. **(1) Perfil:** referente ao cadastro e alteração de usuário. **(2) Página teste:** essa implementação é uma forma alternativa de testar as funcionalidades da ferramenta sem a necessidade de um cadastro prévio de uma página para validar. **(3) Validar páginas:** para utilizar a ferramenta de uma forma mais profissional, essa funcionalidade exige que o usuário desenvolvedor cadastre uma ou várias páginas web para que o usuário cliente possa validar.



Figura 3: Menu principal visão do cliente - rTiWIM

A Figura 4 apresenta as principais funcionalidades para visão do desenvolvedor. **(1) Perfil:** funcionalidade de alteração do cadastro do usuário que é disponibilizada nas duas visões distintas. **(2) Cadastrar site:** uma das principais funcionalidades da visão do desenvolvedor, pois a mesma possui a opção de cadastrar novas interfaces web especificando quais usuários poderão validar aquelas páginas através do email cadastrado no seu perfil de usuário. **(3) Feed-**

**backs:** nesse menu é possível verificar todas as validações que foram executadas pelos usuários clientes separadas por projeto, usuário e *logs* do sistema. Os *logs* sinalizam quais as modificações que o usuário realizou através do menu validar páginas da visão do cliente. **(4) Instruções:** para que os desenvolvedores possam utilizar as funcionalidades da ferramenta é necessário inserir os *scripts* da aplicação no cabeçalho da página que será avaliada também como o cadastro da mesma na ferramenta. Nós disponibilizamos um video no youtube demonstrando o funcionamento da ferramenta<sup>9</sup>.

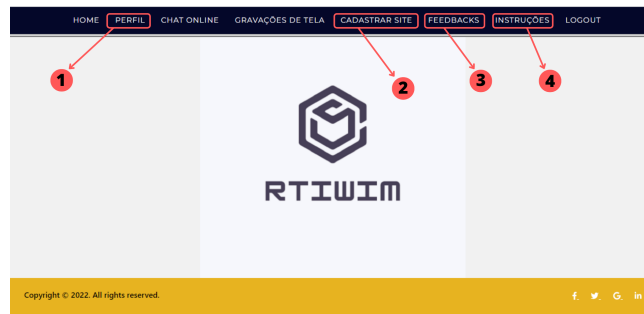


Figura 4: Menu principal visão do desenvolvedor - rTiWIM

## 4 Validação da Ferramenta

Para validação da ferramenta, foi utilizado a técnica de grupo focal que permite coletar dados qualitativos de um determinado grupo de participantes por meio de entrevistas e questionamentos [14]. A implementação dessa etapa foi caracterizada na criação de grupos focais que devem possuir um moderador, dois grupos em comum para criar um ambiente de discussão e, um grupo de observadores que refletem as opiniões e execuções das tarefas.

**Questões de pesquisa** Como motivação para criar as questões de pesquisa, utilizamos o conceito da engenharia semiótica que segundo [4] possibilita a comunicação entre usuários através de sistemas. Dessa forma, com proposta de criar e disponibilizar uma ferramenta que possa validar e elicitare requisitos de software, definimos as seguintes questões: (1) *É possível validar e elicitare requisitos com a ferramenta?*; (2) *Existe algum tipo de melhoria que poderá ser adicionada na ferramenta?*. Respondendo essas interrogativas, conseguimos obter reflexões através de experiências e argumentos dos profissionais da área, a fim de obter

<sup>9</sup> <https://youtu.be/55EiwNoQpE0>

um posicionamento em relação às funcionalidades do sistema. Dessa forma, foi criada uma sessão de grupo focal para obter indícios de validações e futuras melhorias do sistema.

**Perfil dos participantes** Segundo [11], a quantidade de convidados pode ser definida entre seis e doze participantes para realização do grupo focal. Dessa forma, recrutamos oito potenciais usuários do sistema que são profissionais da área e desenvolvedores de software. Depois do processo de recrutamento, iniciamos a divisão do grupo focal que ficou separado da seguinte forma: (3/8) Grupo dos clientes; (3/8) Grupo dos desenvolvedores e (2/8) Observadores. Após a divisão dos grupos, foi investigado sobre a experiência dos participantes com desenvolvimento de software: **(i) Menor que 1 ano, (ii) Maior que 1 ano e menor que 2 anos, (iii) Maior que 2 anos e menor que 10 anos e (iv) Maior que 10 anos.** Também foi perguntado o grau de formação: **(i) Técnico, (ii) Graduado, (iii) Especialista, (iv) Mestre, (v) Doutor e (vi) Pós - Doutorado.** Como resultado dos níveis de experiência, obtivemos os seguintes resultados: 42,9% dos participantes possuem tempo de experiência maior que 2 anos e menor que 10 anos, 28,6% dos participantes que possuem tempo de experiência maior que 10 anos e 28,6% dos participantes que possuem tempo de experiência maior que 1 ano e menor que 2 anos. Em relação aos níveis de formação, alcançamos os seguintes resultados: 57,1% têm formação técnica, 14,3% possuem graduação, 14,3% são especialistas e 14,3% possuem mestrado.

**Planejamento** A sessão do grupo focal foi planejada para ser realizada *on-line* utilizando a ferramenta *google meet*. Os passos do planejamento foram divididos da seguinte forma: (1) O moderador disponibilizou um formulário digital para coletar as informações técnicas de cada convidado com intuito de dividir o grupo de acordo com as seguintes responsabilidades: **(i) Cliente, (ii) Desenvolvedor e (iii) Observador.** (2) O moderador realizou uma breve apresentação da ferramenta, e em seguida disponibilizou acesso ao sistema com tarefas simples para os dois grupos de usuários: clientes e desenvolvedores. Para cada grupo, foi alocado um observador para realizar as principais observações de cada tarefa implementada pelo grupo. (3) O moderador solicitou ao grupo de participantes da visão do cliente a apresentação dos seus principais argumentos. (4) O moderador solicitou ao grupo de participantes da visão do desenvolvedor a apresentação dos seus principais argumentos. (5) O moderador convidou os observadores a argumentar sobre as funcionalidades desempenhadas pelas equipes.

**Execução** Os primeiros 15 minutos da sessão foram destinados para o preenchimento do formulário<sup>10</sup>. Após o término desta primeira sessão, o moderador designou as tarefas para ambos os grupos, que teve um tempo de duração de 20 minutos. Posteriormente foram destinados 25 minutos para as discussões, totalizando 60 minutos na soma de todas as etapas. Em relação à participação,

<sup>10</sup> <https://url.gratis/11z7tV>

todos os colaboradores executaram todas as tarefas exigidas pelo moderador, sendo possível identificar alguns destaques que direcionava os argumentos de cada grupo. Já o grupo de observadores realizou todas as anotações com clareza gerando uma terceira opinião após o *feedback* de cada integrante.

#### 4.1 Resultados

Todos os argumentos foram positivos em relação à finalidade da ferramenta, 100% dos convidados relataram que a ferramenta tem uma ótima proposta e resolve o que se propõe a resolver. Nesse sentido, é possível identificar a importância que a ferramenta possui dentro do contexto de desenvolvimento de software. No entanto, também foi possível identificar várias sugestões de melhoria.

**Grupo Cliente (GC)** (1) *É possível validar e elicitar requisitos com a ferramenta?* Para responder essa questão de pesquisa, tivemos os seguintes argumentos: **(GC1)** “O sistema oferece uma interface pouco agradável para o cliente, não muito intuitiva, porém resolve o que se propõe a resolver”, **(GC2)** “No geral, achei a ferramenta muito legal, a ideia muito boa e o funcionamento já está bem responsivo. Só faltam alguns ajustes”; **(GC3)** “A proposta da aplicação é boa, permite ao cliente interagir diretamente com um protótipo do produto que está em desenvolvimento. Essa interação poderá facilitar a comunicação entre *product owner* e time de desenvolvimento”. (2) *Existe algum tipo de melhoria que poderá ser adicionada na ferramenta?* Alguns argumentos influenciaram em futuras melhorias da ferramenta divididas, como destacado: **(GC1)** “Implementar um histórico de alterações na visão do cliente”, **(GC2)** “Percebi que o campo relacionado ao botão adicionar, por vezes ficou instável com relação a clicar nos campos contidos nele, ficava sumindo”; **(GC3)** “O histórico de modificações poderia ter uma opção de editar a modificação feita ou remover uma específica, ao invés de desfazer uma a uma”. Através dos seguintes relatos, foi possível identificar que a ferramenta é capaz de exercer seu propósito, bem como observar futuras melhorias no desenvolvimento da mesma.

**Grupo Desenvolvedor (GD)** Depois da realização da primeira tarefa que foi designada pelo moderador, os desenvolvedores puderam verificar através do menu “feedback” as alterações que o grupo dos clientes realizaram. Dessa forma, foi possível analisar alguns argumentos de forma técnica e específica que foram sugeridas na seguinte ordem: (1) implementação de responsividade em todas as páginas para utilização da ferramenta em dispositivos móveis; (2) criação de categorias e versionamento para cada alteração que foi realizada pelo cliente; e (3) mostrar o estado do elemento que foi alterado. Nesse sentido, tais argumentos foram de suma importância para validar a interface e implementação das funcionalidades da ferramenta.

**Observadores (O)** Formalizando as duas visões anteriores, temos alguns argumentos que foram identificados pelos observadores dentro das discussões: **(O1)**



“A ferramenta tem grande potencial, e a partir das observações feitas do momento das discussões, pude notar algumas melhorias a serem feitas para que a ferramenta alcance seu objetivo com maestria. Tais melhorias seriam, na parte do usuário, adicionar o versionamento das modificações para o usuário conseguir ver e aplicar diferentes momentos da aplicação, a reorganização dos menus e alguns itens no layout para que facilite o uso”. **(O2)** “Já na parte do desenvolvedor, seria interessante reorganizar como as informações são mostradas para facilitar a compreensão das mesmas e/ou modificações realizadas”.

## 5 Conclusão e Trabalhos Futuros

Este artigo apresentou a ferramenta rTiWIM, que tem por objetivo a elicitação e validação de requisitos pelo usuário e desenvolvedores utilizando a linguagem de interface. Para validar a ferramenta, realizou-se sessões com um grupo focal com profissionais da área e desenvolvedores de sistemas. Concluímos que a utilização da linguagem de interface (UIL) para elicitar e validar requisitos de software, pode ser relevante para o desenvolvimento de um software de qualidade. Durante a validação os participantes indicaram satisfação na utilização da ferramenta, na qual o usuário final pode expressar suas opiniões de uma forma mais detalhada, distinguindo-se das técnicas que eram utilizadas tradicionalmente.

Como trabalhos futuros, vislumbra-se a implementação das funcionalidades: *(i) chat online*: para conversação em tempo real com os desenvolvedores, possibilitando deixar mensagens ou lembretes para que os desenvolvedores possam averiguar posteriormente; *(ii) gravações de tela*: onde as ações do usuário seriam gravadas durante o processo de utilização da ferramenta por vídeo, no qual o desenvolvedor poderá analisar o comportamento do usuário final em tempo de execução; *(iii) sugestões dos participantes da validação/avaliação*: em que foram identificados no tópico resultados desse instrumento de pesquisa.

## Referências

1. Andréia Libório Sampaio. Um Modelo para Descrever e Negociar Modificações em Sistemas Web. Ph. D. Dissertation. PUC-Rio (2010).
2. Arun Kumar Sangaiah, Oluwarotimi Williams Samuel, Xiong Li, Mohamed Abdel Basset and Haoxiang Wang. Towards an efficient risk assessment in software projects Fuzzy reinforcement paradigm. *Computers Electrical Engineering*, 71 , 833–846, (2018).
3. Azham Hussain, Emmanuel OC Mkpojiogu, and Fazillah Mohmad Kamal: The role of requirements in the success or failure of software projects. *International Review of Management and Marketing* 6, 7S , 306–311, (2016).
4. Barbosa, Simone; SILVA, Bruno. *Interação humano-computador*. Elsevier Brasil, (2010).
5. Barricelli, Barbara Rita et al. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software*, v. 149, p. 101-137, (2019).

6. Carla Pacheco, Ivan García, and Miryam Reyes. Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *IET Software* 12, 4 , 365–378, (2018).
7. Coronado, Enrique et al. Visual programming environments for end-user development of intelligent and social robots, a systematic review. *Journal of Computer Languages*, v. 58, p. 100970, (2020).
8. Cristina T. Cerdeiral and Gleison Santos. Software project management in high maturity: A systematic literature mapping. *Journal of Systems and Software* 148, 56–87, (2019).
9. Da Cunha, Cecília Kremer Vieira. Um Modelo Semiótico dos Processos de Comunicação Relacionados à Atividade de Extensão à Aplicação por Usuários Finais. Tese de Doutorado. Pontifical Catholic University of Rio de Janeiro, Brazil. (2001).
10. Didar Zowghi and Chad Coulin. Requirements elicitation: A survey of techniques, approaches, and tools. In *Engineering and managing software requirements*, 19–46, Springer (2005).
11. Galego, Carla; GOMES, Alberto. Emancipação, ruptura e inovação: o “focus group” como instrumento de investigação. *Revista Lusófona de Educação*, v. 5, n. 5, (2005).
12. Goumopoulos, Christos; Mavrommati, Irene. A framework for pervasive computing applications based on smart objects and end user development. *Journal of Systems and Software*, v. 162, p. 110496, (2020).
13. Howard B. Reubenstein and Richard C. Waters. 1991. The requirements apprentice: Automated assistance for requirements acquisition. *IEEE Transactions on Software Engineering* 17, 3 , 226, (1991).
14. Jyrki Kontio, Johanna Bragge, and Laura Lehtola. The focus group method as an empirical tool in software engineering. In *Guide to advanced empirical software engineering*, 93–116, Springer (2008).
15. Kontio, Jyrki e Bragge, Johanna e Lehtola, Laura. The Focus Group Method as an Empirical Tool in Software Engineering. 93-112, (2008).
16. Leah Goldin and Daniel M Berry. 1997. AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. *Automated Software Engineering* 4, 4 , 375–412, (1997).
17. Marc Oriol, Melanie Stade, Farnaz Fotrousi, Sergi Nadal, Jovan Varga, Norbert Seyff, Alberto Abello, Xavier Franch, Jordi Marco, and Oleg Schmidt. FAME: supporting continuous requirements elicitation by combining user feedback and monitoring. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE, 217–227, (2018).
18. Norbert Seyff, Gregor Ollmann, and Manfred Bortenschlager. iRequire: Gathering end-user requirements for new apps. *IEEE 19th International Requirements Engineering Conference*, 347–348, IEEE (2011).
19. Paterno, Fabio; Santoro, Carmen. End-user development for personalizing applications, things, and robots. *International Journal of Human-Computer Studies*, v. 131, p. 120-130, (2019).
20. Pressman, Roger S.; MAXIM, Bruce R. *Engenharia de software-9*. McGraw Hill Brasil, 57-70, (2021).
21. Setlur, Vidya et al. Eviza: A natural language interface for visual analysis. In: *Proceedings of the 29th annual symposium on user interface software and technology*. 365-377, (2016).
22. Sommerville, Ian. *Engenharia de Software*. Edição 10. 85-113, (2019).