

# Proposta de Ferramenta de Priorização de Requisitos de Software a partir de uma Revisão Sistemática da Literatura

Julia Caldeira Mendonça<sup>1</sup>, Camila Zacché de Aguiar<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal do Espírito Santo (UFES)  
Vitória – ES – Brasil

julia.c.mendonca@edu.ufes.br, camila.z.aguiar@ufes.br

**Abstract.** *This article presents a systematic review of software requirements prioritization tools based on publications in the IEEE database from 2010 to 2023. The analysis identifies challenges such as stakeholder interaction and multifactor prioritization, highlighting the use of intuitive interfaces and artificial intelligence to address them. In response to these gaps, we present a tool proposal to integrate dynamic visualization and collaborative voting, in order to increase the efficiency and effectiveness of prioritization decisions.*

**Resumo.** *Este artigo apresenta uma revisão sistemática das ferramentas de priorização de requisitos de software, baseada em publicações na base IEEE entre 2010 e 2023. A análise identifica desafios como a interação entre stakeholders e a priorização multifatorial, destacando o uso de interfaces intuitivas e inteligência artificial para mitigá-los. Em resposta a estas lacunas, apresentamos uma proposta de ferramenta para integrar a visualização dinâmica e a votação colaborativa, a fim de aumentar a eficiência e eficácia das decisões de priorização.*

## 1. Introdução

A priorização de requisitos é crucial na engenharia de software, pois define a ordem e relevância dos requisitos, assegurando que os mais críticos sejam tratados primeiro e maximizando o valor entregue ao cliente. Em mercados voláteis, a priorização dinâmica evita desperdício de recursos em funcionalidades menos urgentes, conforme evidenciado em falhas de projeto [Sommerville 2003, Montealegre 1996].

Diversas técnicas de priorização, como MoSCoW, Planning Poker, Análise de Kano e Priorização Baseada em Valor, têm sido amplamente utilizadas para guiar a definição de prioridades em projetos de software. No entanto, o mercado ainda carece de ferramentas integradas e adaptáveis que facilitem ajustes contínuos ao longo do ciclo de vida do desenvolvimento de software [Cohn 2006].

Este artigo visa estudar ferramentas de priorização de requisitos por meio de uma revisão sistemática da literatura na base de dados da IEEE entre 2010 e 2023, explorando as técnicas aplicadas e discutindo as funcionalidades. A partir desta análise, foram identificadas ferramentas que empregam inteligência artificial, elementos de gamificação ou que dedicam-se a disponibilizar interfaces intuitivas. Também foram observados problemas na priorização de requisitos com interdependências ou conflitantes, além de dificuldade de comunicação na priorização em desenvolvimento global de software.

Por fim, apresentamos uma proposta de ferramenta denominada PrioriTech, que integra visualização dinâmica e um sistema de votação colaborativa a fim de superar os problemas identificados e acompanhar as tendências. A visualização dinâmica facilita a compreensão das interdependências e do impacto dos requisitos em tempo real, enquanto o sistema de votação colaborativa assegura que todos os *stakeholders* tenham uma voz no processo de priorização, promovendo decisões mais democráticas e bem-informadas.

O restante do artigo está estruturado da seguinte forma: A Seção 2 descreve o método de pesquisa, incluindo o planejamento, a condução da busca e a análise dos artigos. A Seção 3 analisa os resultados obtidos, abordando a distribuição dos artigos, as ferramentas e técnicas de priorização, funcionalidades encontradas e formas de aquisição. A Seção 4 oferece as conclusões do estudo e sugere direções para pesquisas futuras. Finalmente, a Seção 5 lista as referências bibliográficas utilizadas.

## 2. Método de Pesquisa

A revisão da literatura seguiu a metodologia estruturada de [Kitchenham et al. 2007], visando identificar, avaliar e interpretar as evidências disponíveis sobre ferramentas de priorização de requisitos de software. O processo foi dividido em três etapas: planejamento, condução da busca e análise dos artigos.

### 2.1. Etapa de Planejamento

Durante a fase de planejamento, o escopo do estudo foi cuidadosamente delineado, e as perguntas de pesquisa foram definidas para direcionar a investigação. As Questões de Pesquisa foram elaboradas de forma a abranger os principais aspectos das ferramentas de priorização de requisitos de software, apresentadas na Tabela 1. A *string*

**Tabela 1. Questões de pesquisa e suas motivações**

Perguntas de Pesquisa	Motivação
RQ1. Quais ferramentas são usadas / propostas nas práticas de pesquisa que foram relatadas?	Entender o panorama atual e histórico das ferramentas de priorização de requisitos na literatura recente.
RQ2. Quais as principais lacunas encontradas ou tratadas nas pesquisas que foram relatadas?	Revelar áreas que necessitam de mais investigação e desenvolvimento, direcionando esforços futuros de pesquisa para desafios não resolvidos e emergentes.
RQ3. Quais técnicas foram implementadas pelas ferramentas de priorização de requisitos?	Catalogar as técnicas de priorização de requisitos implementadas, avaliando aplicabilidade das abordagens utilizadas.
RQ4. Quais funcionalidades as ferramentas de priorização de requisitos abrangem?	Identificar as funcionalidades mais importantes e frequentes nas ferramentas.
RQ5. Qual é a forma de aquisição da ferramenta (gratuita, paga, ou outra)?	Esclarecer a acessibilidade e as barreiras econômicas das ferramentas.

de busca foi desenvolvida iterativamente, ajustada com base nos resultados e em estudos semelhantes para aumentar a precisão e abrangência. A estratégia final seguiu diretrizes de metodologias bem-sucedidas de pesquisas anteriores [Sufian et al. 2021, Somohano-Murrieta et al. 2024].

A *string* final, refinada para maximizar a relevância, foi definida como: ("All Metadata": requirements prioritization tool) AND ("All Metadata": Requirements Engineering).

## 2.2. Condução da Busca e Análise dos Artigos

A busca sistemática foi conduzida na plataforma IEEE Xplore<sup>1</sup>, reconhecida como uma das principais bases de dados para literatura científica nas áreas de engenharia e tecnologia. A *string* de busca definida na Seção 2.1 foi utilizada para recuperar artigos publicados entre 2010 e 2023, limitados a contribuições em *Conferences* e *Magazines*, excluindo artigos de *Journals*.

Inicialmente, a *string* de busca aplicada à base de dados IEEE retornou 97 artigos dos quais foram extraídos seus metadados, incluindo ano de publicação, título e resumo. Dos artigos retornados, 16 foram excluídos com base nos seguintes critérios de exclusão aplicados como primeira filtragem: **EC1**: Não fornece um resumo; **EC2**: É apenas um resumo; **EC3**: É uma revisão ou mapeamento sistemático; **EC4**: Não está escrito em inglês ou português; **EC5**: É uma cópia ou uma versão anterior de outra publicação que já foi considerada; **EC6**: Não é um estudo primário (por exemplo, editoriais, resumos de palestras principais, tutoriais, etc.); **EC7**: Não é possível ter acesso à versão completa da publicação.

Posteriormente, foram aplicados os seguintes critérios de inclusão que culminaram em 15 estudos selecionados que ofereciam contribuições relevantes para o desenvolvimento e compreensão de ferramentas de priorização de requisitos: **IC1**: Propõe ou apresenta uma ferramenta para priorização de requisitos; **IC2**: Especifica técnicas e/ou algoritmos implementados em ferramentas de priorização de requisitos de software; **IC3**: Aborda o escopo de priorização de requisitos de software, e não outros tipos de priorização (por exemplo, priorização de testes, priorização de *stakeholders*, etc.).

A metodologia de filtragem aplicada pode ser visualizada na Figura 1. A aplicação rigorosa desses critérios assegura uma seleção qualificada de conteúdo, essencial para atingir os objetivos do estudo conforme as práticas recomendadas em revisões sistemáticas na literatura de Engenharia de Software [Kitchenham et al. 2007, Petersen et al. 2008].

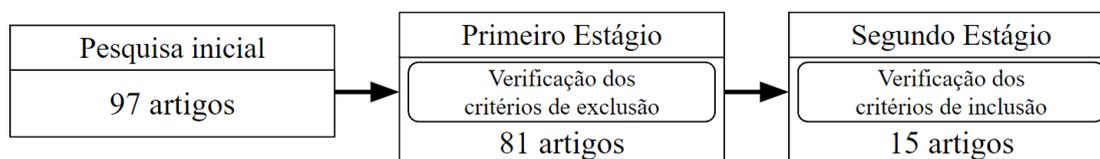


Figura 1. Filtragem dos artigos pelos critérios de exclusão e inclusão

## 2.3. Leitura Detalhada dos Artigos e Registro dos Resultados

A fase de leitura e análise dos artigos permitiu avaliar as técnicas utilizadas, a eficácia das ferramentas discutidas e suas aplicações práticas. O registro detalhado de todos os procedimentos e resultados podem ser encontrados no material suplementar<sup>2</sup>.

<sup>1</sup><https://ieeexplore.ieee.org/Xplore/home.jsp>

<sup>2</sup><https://github.com/julia-caldeira/Mapeamento-Sistematico>

### 3. Análise dos Resultados

#### 3.1. Distribuição dos Artigos pelos Anos

A distribuição temporal dos artigos apresenta uma variação significativa ao longo dos anos estudados conforme a Figura 2. Apesar das lacunas temporais, a continuidade e a dispersão das publicações ao longo dos anos indicam uma persistente relevância do tema dentro da comunidade de engenharia de software.

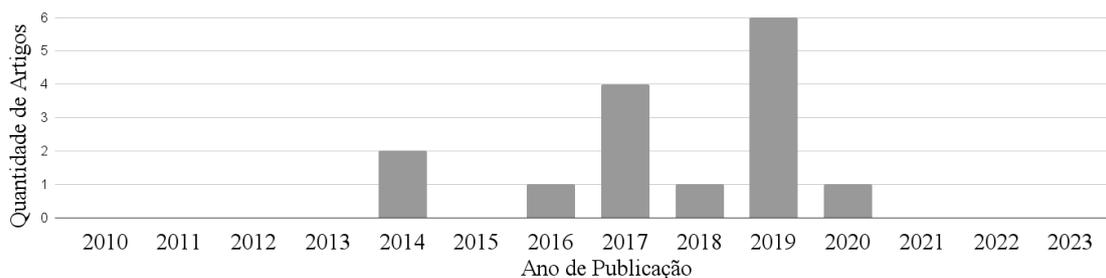


Figura 2. Distribuição dos artigos por ano de publicação

#### 3.2. Ferramentas de Priorização de Requisitos e Lacunas Identificadas

A partir da análise dos artigos, foi possível organizar as ferramentas apresentadas em 6 categorias definidas com base nas lacunas em que cada artigo foca: (i) facilidade de interação, (ii) redução do esforço humano, (iii) priorização baseada em múltiplos fatores ou interdependências entre requisitos, (iv) priorização de requisitos conflitantes, (v) priorização simultânea de requisitos funcionais e não funcionais e (vi) priorização no contexto de desenvolvimento de software global.

**Facilidade de interação** – A análise dos estudos revela um investimento significativo no desenvolvimento e aprimoramento de ferramentas projetadas para tornar mais intuitiva e eficiente a interação dos usuários com as ferramentas. Essas melhorias são importantes para facilitar o engajamento entre engenheiros e *stakeholders* durante o processo de priorização de requisitos e particularmente em cenários onde as reuniões de priorização ocorrem de maneira remota ou com participantes que não possuem formação específica na área. As ferramentas como Decisively [Saxena and Chakraborty 2014] e DMGame [Kifetew et al. 2017a] exemplificam essa categoria ao integrar técnicas de gamificação e análise quantitativa para envolver todos os participantes.

**Redução do esforço humano** – Observa-se uma tendência no desenvolvimento de ferramentas que visam reduzir o envolvimento humano, por meio da implementação de algoritmos de inteligência artificial. Essas soluções contribuem para a eficiência do processo, reduzindo cargas operacionais e melhorando a tomada de decisões em priorização de requisitos de software, conforme discutido em estudos recentes referentes às ferramentas IM-DeCRuD [Jasmis et al. 2019], Plugin para Eclipse [Samer et al. 2019], CLAP [Scalabrino et al. 2019] e Simulador de Gêmeo Digital [Hadar et al. 2020].

**Priorização baseada em múltiplos fatores ou interdependências entre requisitos** – As ferramentas de priorização evoluíram para considerar múltiplos fatores e interdependências, abordando a complexidade dos projetos de software modernos. Essa

abordagem proporciona uma avaliação mais integrada dos requisitos, resultando em decisões alinhadas com os objetivos estratégicos dos projetos, como no AnalyticGraph.com [Gillain et al. 2016], MBRP [Abbas et al. 2019] e na ferramenta com IA não nomeada [Dias et al. 2019].

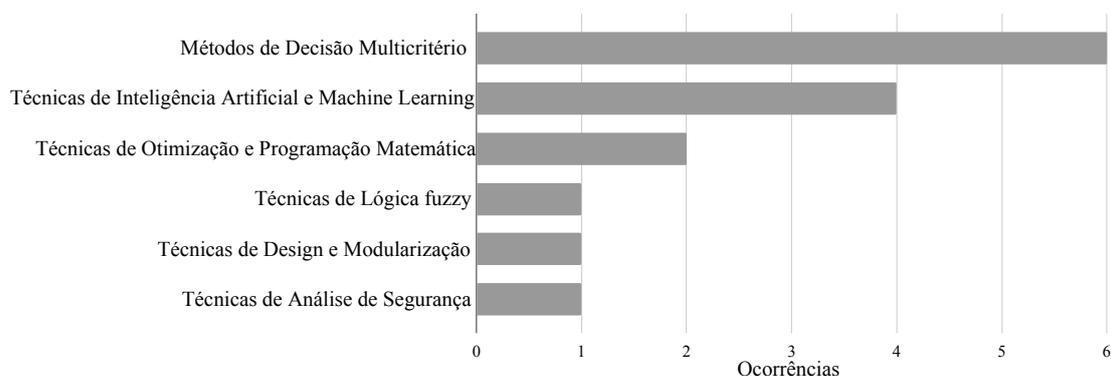
**Priorização de requisitos conflitantes** – As ferramentas têm aplicado técnicas como a lógica *fuzzy* para avaliar e equilibrar diferentes requisitos de usabilidade que podem estar em conflito, facilitando a tomada de decisões em ambientes complexos. Isso foi visto no *framework* com Lógica *fuzzy* [Gulzar et al. 2017] e na ferramenta FLAME [Bang et al. 2017].

**Priorização simultânea de requisitos funcionais e não funcionais** – A ferramenta CHAM [Dabbagh et al. 2014] aborda a priorização simultânea de requisitos funcionais e não funcionais, equilibrando esses dois tipos de requisitos essenciais para o sucesso do projeto.

**Priorização no contexto de desenvolvimento de software global** – Ferramentas que integram IA para priorização e rastreabilidade são fundamentais para projetos globais, garantindo coesão entre equipes distribuídas, como demonstrado pelo estudo com a ferramenta não nomeada em [Haider et al. 2019].

### 3.3. Técnicas Aplicadas pelas Ferramentas de Priorização de Requisitos

As técnicas de priorização utilizadas pelas ferramentas variaram significativamente, refletindo a diversidade de abordagens na área. A Figura 3 apresenta as categorias dos tipos de técnicas e suas ocorrências nos artigos analisados. Em seguida, cada categoria é discutida, explorando suas aplicações específicas e contribuições para a tomada de decisão.



**Figura 3. Categorização dos Artigos por Tipo de Técnica e sua Ocorrência**

Nos métodos de **decisão multicritério**, como o *Analytic Hierarchy Process* (AHP) e o *Hierarchical Analytical Method* (HAM), a tomada de decisão é realizada através da análise de múltiplos critérios em uma estrutura hierárquica, facilitando a priorização e seleção entre diversas alternativas. O AHP, por exemplo, emprega comparações pareadas para determinar pesos relativos dos critérios e avaliar as opções disponíveis, conforme apresentado nos artigos [Saxena and Chakraborty 2014, Kifetew et al. 2017a, Kifetew et al. 2017b, Perini et al. 2018]. O método HAM combina abordagens tradicionais de AHP com matrizes de decisão para melhor capturar as nuances entre diferentes critérios [Dabbagh et al. 2014].

As técnicas de **inteligência artificial e *machine learning***, incluindo métodos como *Case-Based Reasoning* (CBR) e algoritmos de aprendizado supervisionado, são utilizadas para analisar e priorizar requisitos de forma automatizada, adaptando soluções de problemas anteriores e aplicando modelos preditivos para melhorar a tomada de decisão em desenvolvimento de software, conforme observado nos artigos [Haider et al. 2019, Dias et al. 2019, Samer et al. 2019, Scalabrino et al. 2019].

Dentro das técnicas de **otimização e programação matemática**, o uso de *Mixed-Integer Programming* (MIP) e algoritmos como o *PageRank* permitem modelar problemas complexos de liberação e priorização de requisitos, otimizando a alocação de recursos e a sequência de implementações com base em critérios pré-definidos de valor e dependência, conforme observado em [Gillain et al. 2016, Abbas et al. 2019].

A **lógica fuzzy** é uma metodologia matemática aplicada para resolver conflitos e avaliar incertezas em requisitos, especialmente quando as avaliações envolvem julgamentos subjetivos e qualitativos dos *stakeholders*, como demonstrado no método de Mamdani para a priorização de requisitos de usabilidade [Gulzar et al. 2017].

Na categoria de técnicas de **design e modularização**, a utilização de esquemas XML e LOTOS ajuda na estruturação e visualização de requisitos e design, facilitando a modularização e a composição de sistemas complexos, conforme apresentado em [Jasmis et al. 2019].

Por fim, as técnicas de **análise de segurança**, como a análise de gráficos de ataque, fornecem métodos sistemáticos para identificar e priorizar requisitos de segurança, simulando potenciais vetores de ataque e avaliando a eficácia de controles implementados, como apresentado pelo artigo [Hadar et al. 2020].

### 3.4. Funcionalidades Encontradas nas Ferramentas

As ferramentas analisadas a partir dos artigos demonstraram a importância de diversas funcionalidades para a priorização de requisitos em projetos de software. Entre as funcionalidades principais, destacam-se (i) interfaces gráficas e visuais, (ii) a promoção da comunicação e colaboração entre os participantes, (iii) análise e priorização automatizada, (iv) gamificação do processo de priorização e (v) mecanismos para resolver conflitos e facilitar negociações, contribuindo para a gestão eficiente de projetos de software. Em seguida, tais funcionalidades serão discutidas.

**Interfaces gráficas e visuais** são essenciais para facilitar a interação do usuário e a análise dos dados. A criação de interfaces amigáveis e a visualização clara das relações entre os requisitos são cruciais para acelerar o processo de decisão e garantir uma experiência intuitiva em contextos complexos de desenvolvimento [Saxena and Chakraborty 2014, Kifetew et al. 2017a, Perini et al. 2018].

Além disso, funcionalidades que promovem a **comunicação e colaboração** entre os participantes são vitais, especialmente em ambientes distribuídos. Estas funcionalidades, tais como a rastreabilidade dos requisitos ao longo do ciclo de vida do desenvolvimento do software e aplicação de inteligência artificial para reduzir a necessidade de interação humana, suportam a participação de *stakeholders* de diferentes geografias, melhorando a coordenação entre equipes e a sincronização de atividades [Haider et al. 2019].

A **análise e priorização automatizada** de requisitos com uso de técnicas

automatizadas e modelos matemáticos permitem a análise eficiente e precisa das preferências dos *stakeholders*, além de suportar a tomada de decisão em tempo real [Saxena and Chakraborty 2014, Abbas et al. 2019].

Elementos de **gamificação** são implementados para aumentar o engajamento e a motivação dos usuários, tornando o processo de priorização mais atraente e interativo. Estes incluem sistemas de pontuação, competições e recompensas visuais, que são eficazes mesmo em tarefas percebidas como monótonas ou complexas [Kifetew et al. 2017a, Kifetew et al. 2017b, Perini et al. 2018].

Além disso, mecanismos para **resolver conflitos e facilitar negociações** são essenciais em processos colaborativos. Estes mecanismos asseguram que todas as partes sejam ouvidas e que as decisões sejam aceitáveis para todos os envolvidos, contribuindo para a gestão eficiente de projetos de software [Dabbagh et al. 2014].

### 3.5. Formas de Aquisição das Ferramentas

A análise das modalidades de acesso às ferramentas revelou que a maioria dos artigos não fornece informações claras sobre como as ferramentas podem ser adquiridas. Foram explicitamente disponibilizadas ao público e de forma gratuita as seguintes: DMGame [Kifetew et al. 2017a], FLAME [Bang et al. 2017], Plug-in para Eclipse [Samer et al. 2019], CLAP [Scalabrino et al. 2019] e foi observado que AnalyticGraph.com já não se encontra mais acessível online [Gillain et al. 2016]. Frequentemente, sugere-se ou afirma-se que o acesso é restrito ao grupo desenvolvedor e aos seus clientes associados ou que o acesso é pago, isso foi visto nas demais ferramentas analisadas no estudo.

## 4. Proposta de Ferramenta

Com base nas lacunas identificadas, apresentamos uma proposta de ferramenta de priorização de requisitos denominada PrioriTech. A Tabela 2 apresenta como a ferramenta pretende minimizar essas lacunas.

A Figura 4 apresenta a interface da ferramenta PrioriTech com uma simulação de uso. Em 1 está indicado o backlog dos requisitos do projeto e suas interdependências, e em 2 um dos requisitos é selecionado pelo *stakeholder* e está sendo priorizado em tempo real por meio dos botões de votação. Os *stakeholders* expressam-se por meio dos botões de *upvote* em verde ou *downvote* em vermelho para expressar opinião positiva ou negativa a respeito do aumento da prioridade do requisito em questão, respectivamente. A votação ponderada resolve conflitos e a priorização simultânea de requisitos funcionais e não funcionais garante equilíbrio. Em projetos globais, PrioriTech permite votações em tempo real, com histórico rastreável e tradução automática para uma colaboração eficiente.

Portanto, estima-se que a combinação dessas funcionalidades tende a aumentar a aceitação das decisões tomadas pelos *stakeholders*. Além disso, é importante destacar que a proposta da ferramenta foi projetada para atender as lacunas identificadas nas ferramentas analisadas na revisão sistemática, sendo implementada e avaliada em trabalhos futuros.

**Tabela 2. Proposta PrioriTech frente às lacunas identificadas**

<b>Lacunas Identificadas</b>	<b>Proposta PrioriTech</b>
Facilidade de interação com o sistema	Interface dinâmica que facilita a compreensão dos requisitos e suas interdependências em tempo real e feedback visual imediato para cada ação.
Redução do esforço humano na Priorização	Automação do cálculo de prioridades com base em critérios predefinidos e notificações automáticas para alertar sobre mudanças críticas.
Priorização baseada em múltiplos fatores ou interdependências	Visualização completa das interdependências e impacto multifatorial para uma avaliação completa de impacto em tempo real. Suporte a filtros personalizáveis para diferentes critérios (complexidade, valor de negócio, etc.).
Priorização de requisitos conflitantes	Mecanismo de votação ponderada para resolução de impasses de forma democrática, com visualização prática de impacto de conflitos em tempo real.
Priorização de requisitos funcionais e não funcionais	Avaliação simultânea de ambos os tipos de requisitos com indicadores de prioridade.
Priorização no contexto de desenvolvimento global	Votação colaborativa em tempo real para <i>stakeholders</i> distribuídos geograficamente, bem como histórico de decisões com rastreabilidade para todos os <i>stakeholders</i> . Tradução automática de comentários e interações.

## 5. Considerações Finais

Este estudo reforça a necessidade contínua de ferramentas avançadas e eficazes para a priorização de requisitos de software, evidenciando a diversidade de técnicas, que variam de algoritmos matemáticos a métodos baseados em inteligência artificial. A proposta da ferramenta PrioriTech integra visualização dinâmica e votação colaborativa, pretende melhorar a experiência e a eficácia no processo de priorização. Espera-se que esta pesquisa estimule o desenvolvimento de soluções mais intuitivas e colaborativas, contribuindo para o avanço da área.

## Referências

- Abbas, M., Inayat, I., Jan, N., Saadatmand, M., Paul Enoiu, E., and Sundmark, D. (2019). Mbrp: Model-based requirements prioritization using pagerank algorithm. In *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, pages 31–38.
- Bang, J. Y., Brun, Y., and Medvidovic, N. (2017). Continuous analysis of collaborative design. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 97–106.
- Cohn, M. (2006). *Agile Estimating and Planning*. Prentice Hall, Upper Saddle River, NJ, USA.

The image shows a screenshot of the PrioriTech web application interface. On the left, there is a 'Lista Completa' (Complete List) for 'Projeto: TechSolutions Inc.' with a filter icon labeled '1'. Below this is a table of requirements with columns for 'REQUISITO', 'TIPO', 'PRIORIDADE CALCULADA', 'DEPENDÊNCIAS', 'BLOQUEIOS', and 'AÇÕES'. The first requirement is 'Notificar clientes via WhatsApp' with a priority of 5.89, 2 dependencies, and 1 block. It has a dropdown menu showing dependencies like 'Integração com API do WhatsApp Business' and 'Obtenção do consentimento dos clientes'. On the right, there is a 'Debate Atual' (Current Debate) section labeled '2', which shows a detailed view of the 'Notificar clientes via WhatsApp' requirement, including its dependencies and a voting section with 'VOTOS PARA AUMENTAR' and 'VOTOS PARA REDUZIR' buttons. Below the voting section are comments from team members, such as João Silva and André Ribeiro, discussing the implementation costs and dependencies.

**Figura 4. Protótipo de interface da ferramenta PrioriTech**

- Dabbagh, M., Lee, S. P., and Parizi, R. M. (2014). Application of hybrid assessment method for priority assessment of functional and non-functional requirements. In *2014 International Conference on Information Science Applications (ICISA)*, pages 1–4.
- Dias, P. A., João, I. M., and Lourenço, J. C. (2019). Patients’ requirements prioritization on the house of quality: The case of glucose monitoring devices in young adults with type 1 diabetes. In *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*, pages 1–4.
- Gillain, J., Jureta, I., and Faulkner, S. (2016). Planning optimal agile releases via requirements optimization. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pages 10–16.
- Gulzar, K., Sang, J., Ramzan, M., and Kashif, M. (2017). Fuzzy approach to prioritize usability requirements conflicts: An experimental evaluation. *IEEE Access*, 5:13570–13577.
- Hadar, E., Kravchenko, D., and Basovskiy, A. (2020). Cyber digital twin simulator for automatic gathering and prioritization of security controls’ requirements. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 250–259.
- Haider, W., Hafeez, Y., Ali, S., Jawad, M., Ahmad, F. B., and Rafi, M. N. (2019). Improving requirement prioritization and traceability using artificial intelligence technique for global software development. In *2019 22nd International Multitopic Conference (INMIC)*, pages 1–8.
- Jasmis, J., Aziz, A. A., Jamel Elias, S., Hajar Hasrol Jono, M. N., Abd Razak, R., and Mansor, S. (2019). An evolutionary tool for requirements and design crosscutting concerns. In *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–5.

- Kifetew, F., Munante, D., Perini, A., Susi, A., Siena, A., and Busetta, P. (2017a). Dm-game: A gamified collaborative requirements prioritisation tool. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 468–469.
- Kifetew, F. M., Munante, D., Perini, A., Susi, A., Siena, A., Busetta, P., and Valerio, D. (2017b). Gamifying collaborative prioritization: Does pointsification work? In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 322–331.
- Kitchenham, B. A., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2007). Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15.
- Montealegre, R. (1996). An it fiasco: The denver international airport baggage system. *International Journal of Cases on Electronic Commerce*, 1(2):1–17.
- Perini, A., Seyff, N., Stade, M., and Susi, A. (2018). Exploring re knowledge for gamification: Can re achieve a high score? In *2018 1st International Workshop on Affective Computing for Requirements Engineering (AffectRE)*, pages 14–19.
- Petersen, K., Wohlin, C., and Baca, D. (2008). The waterfall model in large-scale development. In *International Conference on Software Engineering and Knowledge Engineering (SEKE)*, pages 285–290. Citeseer.
- Samer, R., Stettinger, M., and Felfernig, A. (2019). Towards issue recommendation for open source communities. In *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 164–171.
- Saxena, S. K. and Chakraborty, R. (2014). Decisively: Application of quantitative analysis and decision science in agile requirements engineering. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 323–324.
- Scalabrino, S., Bavota, G., Russo, B., Penta, M. D., and Oliveto, R. (2019). Listening to the crowd for the release planning of mobile apps. *IEEE Transactions on Software Engineering*, 45(1):68–86.
- Sommerville, I. (2003). *Software Engineering*. Addison-Wesley.
- Somohano-Murrieta, J. C. B., Ocharan-Hernández, J. O., Sanchez-García, A. J., and Arenas-Valdés, M. d. I. Á. (2024). Requirements prioritization techniques in the last decade: A systematic literature review. *School of Statistics and Informatics, University of Veracruz*.
- Sufian, M., Khan, Z., Rehman, S., and Butt, W. H. (2021). A systematic literature review: Software requirements prioritization techniques.