

Impacto do uso de *Architecture Decision Records* (ADRs) em uma equipe de desenvolvedores novatos de software

Enzo D. Barboza Paiva Carvalho¹, Awdren de L. Fontão¹

¹Faculdade de Computação - Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 79070-900 – Campo Grande – MS – Brasil

enzo.paiva, awdren.fontao@ufms.br

Abstract. *This study analyzes the impact of using Architecture Decision Records (ADRs) in architectural decision making during the development of a software project by a team of six novice developers. The objective was to understand the influence that documenting these decisions has on engagement, collaboration, and contribution among team members in a software development context. To this end, an interactive workflow was designed and evolved through five versions, each incorporating gradual improvements, including discussions and documentation of the decisions made. The results highlight that using ADRs, combined with appropriate tools, fosters transparency, engages participants, and aligns the team with the project's goals. Furthermore, it provides a clear record of the decisions made, facilitating system maintenance, evolution, and comprehension, while also creating a more collaborative and efficient working environment.*

Resumo. *Este estudo analisa o impacto do uso de Architecture Decision Records (ADRs) na tomada de decisões arquiteturais durante o desenvolvimento de um projeto de software por uma equipe de seis desenvolvedores novatos. O objetivo foi compreender a influencia que o registro dessas decisões causa no engajamento, colaboração e contribuição dos membros de uma equipe de desenvolvimento de software. Desse modo, foi construído um fluxo de trabalho interativo que passou por cinco versões, cada uma com melhorias graduais, envolvendo debates e documentações das decisões realizadas. Os resultados evidenciam que o uso de ADRs, combinado com ferramentas adequadas, promove transparência, engaja os participantes e alinha a equipe com os objetivos do projeto, além de deixar um histórico das decisões tomadas. Portanto, resulta na facilitação da manutenção, evolução e compreensão do sistema, ao mesmo tempo que cria um ambiente de trabalho mais colaborativo e eficiente.*

1. Introdução

Em um projeto de software, a definição arquitetural envolve diversos fatores, como qualidade, segurança e desempenho, que determinam os componentes técnicos e operacionais responsáveis pelo funcionamento de um sistema. Esses elementos são essenciais para garantir a eficiência e a robustez da arquitetura proposta (Microsoft (2009)). A participação de outras 'visões arquiteturais' contribui para documentar e representar de forma adequada a arquitetura de um sistema, atendendo satisfatoriamente aos diferentes interesses das partes interessadas através de um conjunto de descrições coerentes, lógicas e harmonizadas (Kruchten, Capilla e Dueñas (2009)).

Optou-se por adotar o termo *Architecture Decision Records* (ADRs) ao longo deste trabalho devido à sua predominância nas pesquisas e na literatura científica internacional. A escolha reflete a tentativa de alinhar o estudo com terminologias amplamente aceitas pela comunidade acadêmica e profissional, especialmente em artigos, livros e materiais de referência revisados. Essa decisão visa garantir maior consistência e facilitar o acesso e entendimento do conteúdo por leitores que já estão familiarizados com a literatura global sobre o tema. Além disso, o uso do termo em inglês evita ambiguidades que poderiam surgir em traduções ou adaptações terminológicas para o português.

De acordo com Keeling (2022), a adoção do uso de *Architecture Decision Records* (ADRs) pelas organizações se torna uma alternativa eficaz para registrar decisões que impactam diretamente as características do sistema, como desempenho, flexibilidade, escalabilidade e segurança. Pois, sua estrutura é geralmente constituída de um pequeno arquivo de texto curto que documenta uma única decisão de *design*, incluindo o contexto em que foi tomada, a decisão em si e suas consequências. Desse modo, as ADRs ajudam a superar desafios comuns no desempenho de atividades de arquitetura nos aspectos de colaboração e suporte de gestão, gestão do conhecimento de arquitetura, adoção, desenvolvimento e manutenção de ferramentas e processos para *design* e evolução da arquitetura (Wan et al., (2023)).

Apesar disso, Tyree e Akerman (2005) discute que em organizações com baixo nível de maturidade, a responsabilidade pela fase de definição arquitetural frequentemente recai sobre um único profissional. Nesses contextos, as escolhas arquiteturais tendem a ser influenciadas pelos interesses pessoais desse indivíduo, que, graças à sua posição de destaque, vê suas decisões aceitas pelos colegas com base na confiança e no respeito que sua experiência inspira. Contudo, Greiler, Storey e Noda (2021) observam que essa dinâmica pode resultar na insatisfação da equipe, reduzindo a produtividade e a retenção de profissionais. Pois, a falta de um ambiente colaborativo e inclusivo reduziu a motivação dos desenvolvedores e afeta diretamente a qualidade e o progresso do projeto.

Ao se analisar a literatura especializada revela uma ampla gama de estudos sobre ADRs, como demonstrado pelos autores citados anteriormente. Por exemplo, Tyree e Akerman (2005) destaca como elas podem promover transparência e rastreabilidade, enquanto Wan et al., (2023) descreve as melhorias na gestão do conhecimento e no suporte à evolução arquitetural. No entanto, não foram identificados estudos que avaliem o impacto dessa prática no desempenho e engajamento de equipes compostas por desenvolvedores novatos junto ao projeto. Desse modo, com a crescente expansão do mercado de trabalho na área de tecnologia da informação no Brasil (INSPER (2023)), em que cada vez mais pessoas ingressam nesse setor e novos produtos tecnológicos são desenvolvidos diariamente, torna-se imprescindível avaliar o impacto que o uso dessa ferramenta pode gerar nesse cenário.

Segundo Shinzato et al., (2021), ser novato significa estar nos estágios iniciais de desenvolvimento profissional, com pouca experiência prática e familiaridade com as atividades que irá executar. Esses profissionais enfrentam desafios como a assimilação rápida de novos conhecimentos e a integração em equipes de desenvolvimento (Greiler, Storey e Noda (2021)). Além disso, Wan et al., (2023) aponta que esses profissionais possuem uma compreensão limitada das práticas de desenvolvimento e manutenção de sistemas. O progresso, depende de ambientes que promovam suporte

estruturado e aprendizado progressivo, especialmente ao lidar com ciclos de desenvolvimento e validação arquitetural, conforme complementa Mathur e Malik (2010).

Diante disso, este estudo tem por objetivo identificar e discutir as barreiras, estratégias individuais e coletivas, além dos desafios relacionados ao processo de construção de um documento de referência arquitetural por meio do uso de *ADRs* em equipes formadas por desenvolvedores novatos, ou seja, profissionais que possuem pouca experiência na arquitetura de software, especialmente no que se refere à definição, documentação e implementação de decisões arquiteturais ao longo das fases do ciclo de vida de um projeto. Ademais, busca-se avaliar o engajamento, a percepção e a eficácia dessa ferramenta na fase inicial da construção de um projeto de software.

Portanto, este artigo responde às seguintes questões:

QP1: De que forma a construção de *ADRs* influencia o engajamento e a contribuição dos membros de uma equipe de desenvolvimento de software?

Motivação: A resposta a essa pergunta pode fornecer subsídios para determinar se a prática de utilizar *ADRs* é eficaz para aprimorar a colaboração entre os membros da equipe, bem como sua contribuição e engajamento no projeto de software.

Resultados: Os resultados mostraram que a construção de *ADRs* promoveu a participação ativa de toda a equipe, facilitando discussões colaborativas e prevenindo a repetição de erros, como evidenciado pelo aumento contínuo de comentários nas *ADRs* criadas no GitHub¹. Além de estruturar o desenvolvimento, trouxeram visibilidade para assuntos que antes passavam despercebidas.

O ambiente de diálogo aberto, inclusivo e coletivo através das reuniões de equipe também encorajou a participação de desenvolvedores novatos, tornando-os mais confortáveis para contribuir. A flexibilidade nas discussões e o reconhecimento do envolvimento dos membros fortaleceram ainda mais o engajamento.

QP2: Como os membros de uma equipe de desenvolvimento de software percebem a importância das *ADRs* construídas durante o projeto?

Motivação: Essa pergunta visa compreender as razões pelas quais os *ADRs* são percebidos como uma ferramenta útil para a documentação e a comunicação das decisões arquiteturais que moldam o desenvolvimento do projeto de software.

Resultados: Os resultados revelaram que os membros acreditam que as *ADRs* ajudaram a registrar, documentar e, assim, comunicar de forma mais confiável o estado prático das Decisões Arquiteturais. À medida que foram revisadas periodicamente, e, criadas novas versões devido a mudança nos requisitos ou contexto, foi relatado pela equipe que as melhores práticas foram registradas.

No entanto, houve uma percepção também de que, em alguns momentos, a documentação foi excessivamente detalhada, apresentando assim a necessidade de um equilíbrio entre profundidade e produtividade. O que resultou em falta de transparência em algumas decisões tomadas, por conta da busca da simplificação, e da falta de alinhamento prévio e abordagens mais incisivas com membros mais ociosos, por conta da busca também pela flexibilidade máxima.

¹ <<https://GitHub.com/>>

2. Trabalhos relacionados

Tyree e Akerman (2005) abordam a importância de documentar explicitamente as decisões arquiteturais em projetos de software. É levantado que, em geral, as decisões são subjacentes, feitas a partir da percepção e visão do líder do projeto. Os autores relatam um caso de uma organização financeira na qual, a falta de documentação resultou em dificuldades para transmitir mudanças, falta de transparência sobre as implicações da arquitetura e rastreabilidade das decisões tomadas, tornando o negócio custoso, com altos índices de acoplamento e duplicidade. É proposto utilizar um *template* para documentar essas decisões arquiteturais de forma explícita, transformando o processo de construção em algo mais interativo entre os membros do projeto. Eles concluem que o uso deste conceito resultou na melhora da comunicação e gestão das mudanças no projeto.

Keeling (2022) realizou um estudo para entender porque ADRs vem sendo tão utilizadas em equipes que adotam a metodologia ágil. O autor relata que, inicialmente, as áreas de arquitetura e desenvolvimento ágil não se comunicavam bem por falta de interesse mútuo e percepção de conflito. Contudo, com o advento do *design* centrado em decisões proporcionado pelas ADRs, a acessibilidade das abstrações arquiteturais melhorou, permitindo uma colaboração mais eficaz. Conclui-se que as ADRs envolvem ativamente equipes ágeis, permitindo que os integrantes assumam a posição de arquiteto ao escrevê-las, elevando a sofisticação dos softwares produzidos pela equipe ao proporcionar aos integrantes uma visão mais abrangente sobre o produto.

O grupo de pesquisadores Wan *et al.*, (2023) aborda a compreensão limitada dos profissionais ao realizarem atividades e desafios durante o desenvolvimento e manutenção de sistemas, focando na área de arquitetura de software. Os autores conduziram entrevistas com 32 profissionais de funções diferentes e de 21 organizações diversas, visando obter uma visão abrangente das práticas de arquitetura. A pesquisa identificou temas e padrões nas respostas, envolvendo evolução imprevisível de requisitos, documentações desatualizadas, falta de ferramentas de verificação de conformidade e monitoramento contínuo, dívidas técnicas e cheiros arquiteturais. Eles propõem promover uma cultura empresarial focada na qualidade arquitetural, utilizar melhores ferramentas e aperfeiçoar a documentação e rastreabilidade dos projetos.

Carvalho e Conte (2024) exploram em uma pesquisa parecida, como os profissionais brasileiros da indústria de software tomam decisões críticas relacionadas a arquitetura durante o desenvolvimento de sistemas. Foi realizada uma entrevista semiestruturada com 9 profissionais, de empresas diferentes, onde os autores investigaram fatores contextuais dos projetos que influenciam as decisões tomadas. Constatou-se que a arquitetura é definida focando nos requisitos e limitações do cliente, usando provas de conceito para chegar a uma viabilidade e padrões da empresa como referência. Ademais, foi constatado entre os entrevistados que a documentação das decisões arquiteturais é fundamental para a manutenção e evolução do produto.

Em outro estudo, Greiler, Storey e Noda (2021) investigaram a experiência dos desenvolvedores dentro das organizações de software. Entrevistaram 21 integrantes, a fim de saber sobre suas experiências, saúde do código que produzem, ambiente de desenvolvimento que estão inseridos e a segurança psicológica. A pesquisa resultou no desenvolvimento de um *framework DX*, que ajuda a identificar e melhorar fatores que afetam a experiência do desenvolvedor, abordando barreiras, estratégias e mecanismos de enfrenta-

mento. Notou-se que a melhoria da experiência impacta positivamente sua produtividade, satisfação, engajamento e retenção nos projetos que está participando. Portanto, aplicar essas estratégias no contexto estudo de caso pode deixar o ambiente de desenvolvimento das ADRs mais produtivo e satisfatório.

Com base nos estudos citados, nenhum apresenta os efeitos que as *architecture decision record* (ADRs) têm no engajamento das equipes de software com desenvolvedores novatos e quais estratégias utilizadas ao usá-las. Tyree e Akerman (2005) propõem a utilização de um *template* para documentar decisões arquiteturais. Keeling (2022) concluiu que as ADRs envolvem ativamente equipes e são amplamente utilizadas atualmente. O estudo de Wan et al., (2023) destaca a importância de uma documentação arquitetural robusta e de estratégias eficazes para o engajamento de equipes. Carvalho e Conte (2024) complementam e reforçam através de suas entrevistas que as ADRs são vitais para a manutenção e evolução do software. Além disso, Greiler, Storey e Noda (2021) constataam que melhorar a experiência do desenvolvedor na organização afeta positivamente os projetos em que estão participando, sendo então uma estratégia vital quando correlacionadas com criação de ADRs. Com base nisso, este estudo foca na investigação de como os desenvolvedores novatos se comportam ao trabalhar com ADRs e quais estratégias utilizar aliadas a elas em um projeto de software.

3. Metodologia

Este estudo tem como objetivo analisar como as ADRs afetam uma equipe de desenvolvedores novatos em relação a decidir como será a arquitetura de um novo software. Baseado nisso, é possível obter percepções sobre o processo de tomada de decisão e estratégias usadas para adotar as ADRs no projeto e melhorar a colaboração na equipe. A partir dessa situação, busca-se responder a QP1 identificando como as ADRs afetam os integrantes de um projeto que as usam, e a QP2, buscando compreender a opinião dos integrantes sobre o valor e a utilidade das ADRs no contexto do desenvolvimento de software.

3.1. Escolha do projeto para implementação de ADRs

Escolher um problema que possa ser abordado e transformado em software, com o pré-requisito da ausência de um software que resolvesse o problema específico em questão, é necessário para garantir que a equipe formada para discutir as ADRs não seja influenciada pelas decisões tomadas por outros sistemas e equipes.

Diante disso, foi identificado, em conjunto com colegas, um obstáculo relacionado ao uso do ônibus interno da Universidade Federal de Mato Grosso do Sul (UFMS), popularmente conhecido como businho. O itinerário disponibilizado pela universidade não detalhava todos os pontos e horários que o transporte poderia passar durante seu percurso, dificultando a previsão do horário de chegada do ônibus no ponto. A Figura 1 é o itinerário disponibilizado.



Figura 1. Itinerário businho

Realizar uma pesquisa foi a estratégia adotada para entender as percepções dos estudantes sobre a problemática do transporte da universidade e avaliar a necessidade da implementação de uma solução tecnológica. A pesquisa foi conduzida através de um questionário anônimo na plataforma do Google Formulários², onde 46 estudantes participaram. As questões abordadas foram:

1. Com que frequência você utiliza o ônibus da UFMS?
2. Qual seria sua avaliação sobre a importância de um aplicativo que fornece localização, informações de horários e rotas dos ônibus da UFMS?
3. Como você faz para saber os horários do ônibus da UFMS atualmente ?

Desse modo, foi possível organizar as respostas do questionário referente às perguntas 1 e 2 na Tabela 1, permitindo a análise das respostas, que relevam a distribuição de frequências e a avaliação sobre a importância de um aplicativo para o transporte, segundo os universitários.

Pergunta	Resposta	Quantidade	Porcentagem
Com que frequência você utiliza o ônibus da UFMS?	Muito Frequente	17	37%
	Frequente	5	10,9%
	Ocasionalmente	10	21,7%
	Raramente	5	10,9%
	Nunca	9	19,6%
Qual seria sua avaliação sobre a importância de um aplicativo que fornece localização, informações de horários e rotas dos ônibus da UFMS?	Muito Importante	38	82,6%
	Importante	4	8,7%
	Moderado	2	4,3%
	Às vezes importante	2	4,3%
	Não é importante	0	0%

Tabela 1. Perguntas feita aos alunos e medição dos resultados

Através da pergunta 3, foi possível identificar que os alunos acessam informações sobre os horários de ônibus da UFMS, principalmente, por meio de grupos no WhatsApp.

² <https://docs.google.com/forms/>

Ao observar as interações em um desses grupos, percebeu-se a ausência de informações centralizadas e organizadas, sem supervisão de uma unidade administrativa da instituição que assegure a credibilidade das informações compartilhadas, como podemos ver na Figura 2. Sobretudo, o limite de 1024 por grupo, devido a limitação da ferramenta, dificulta o acesso para novos estudantes (WhatsApp FAQ (2024)).

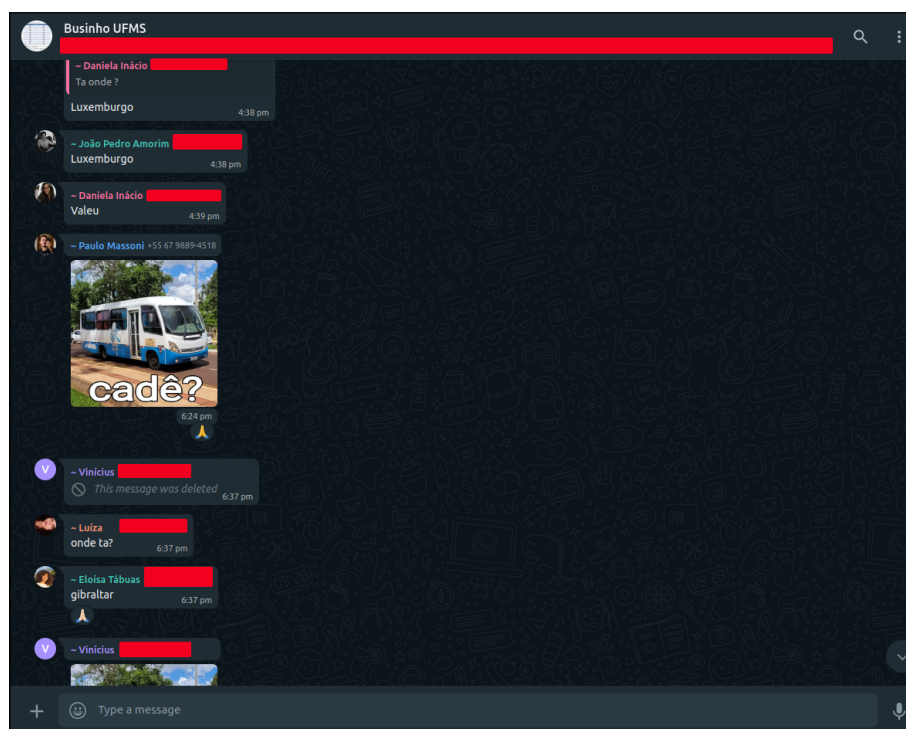


Figura 2. Conversa entre estudantes sobre o businho no whatsapp

A partir desse cenário, criar um aplicativo então surge como a principal proposta de solução, considerando a ampla difusão da tecnologia móvel no Brasil. Onde segundo o (IBGE) (2021), o acesso a dispositivos móveis, como *smartphones*, estão presentes em 98,9% dos lares dos brasileiros.

3.2. Critérios para Definição da Equipe do Projeto

Com a problemática definida, seguindo o ideal de equipe definido por Zimmer (2023) para debater decisões arquiteturais, foi definida uma equipe composta por seis integrantes. A estrutura da equipe, considerou critérios de área de especialidade e experiência em projetos reais como um critério base que cada integrante deve ter para compor a equipe. Além disso, foi considerado competência técnica em áreas específicas que envolvessem a problemática do projeto. Sendo assim, com os seis integrantes definidos, foi definido os papéis que cada um teria conforme apresentado na Tabela 2. A seguir, são detalhadas as responsabilidades e funções de cada integrante do projeto.

ID	Curso	Função no Time	Área de Especialidade	Experiência em Projetos Reais
AL	Sistemas de Informação	Arquiteto-Líder	<i>Backend</i>	Sim
P1	Engenharia de Software	Desenvolvedor	<i>Mobile</i>	Sim
P2	Ciência da Computação	Desenvolvedor	<i>Fullstack</i>	Sim
P3	Engenharia de Software	Desenvolvedor	<i>Backend</i>	Sim
P4	Engenharia de Software	Desenvolvedor	<i>Mobile</i>	Sim
P5	Engenharia da Computação	Desenvolvedor	<i>Frontend</i>	Sim

Tabela 2. Equipe de Desenvolvimento para o Projeto de Aplicativo de Transporte Universitário

3.2.1. Responsabilidades e Funções do Arquiteto-Líder de Software

Liderar tecnicamente, realizar tomada de decisões críticas, organizar o projeto, mediar os debates entre os desenvolvedores novatos, fornecer *feedback* construtivo e objetivo, e orientar a respeito das boas práticas em debater as *ADRs*. Isso permitiu sugerir caminhos de desenvolvimento, os quais foram debatidos pela equipe nas *ADRs*, realizar discussões técnicas e envolver os integrantes na resolução de decisões complexas, assegurando a qualidade do projeto.

Segundo Zimmer (2023), o arquiteto deve garantir a qualidade do documento referencial de arquitetura, evitando anti-padrões. A Tabela 3 apresenta os conceitos para uma documentação precisa e eficaz.

Anti-padrão	Descrição
<i>Pass Through</i>	Revisão superficial com poucos ou nenhum comentário relevante.
<i>Copy Edit</i>	Foco exclusivo em correções de gramática e redação, ignorando o conteúdo técnico.
<i>Siding, Dead End</i>	Comentários que mudam de tópico inesperadamente e não fornecem conselhos tangíveis.
<i>Self Promotion, Conflict of Interest</i>	Recomendações que promovem o trabalho do revisor ou apresentam um conflito de interesses.
<i>Power Game</i>	Comentários que ameaçam os autores com base na posição hierárquica do revisor.
<i>Offended Reaction</i>	Comentários defensivos e subjetivos que mostram ressentimento.
<i>Groundhog Day</i>	Repetição constante da mesma mensagem sem acrescentar valor.

Tabela 3. Anti-padrões a Evitar pelo Arquiteto segundo Zimmer (2023)

Desse modo, foi possível o arquiteto propor possíveis caminhos de desenvolvimento que foram debatidos pela equipe nas *ADRs*. Coordenar a equipe mediando discussões técnicas e engajando os desenvolvedores novatos a participar. E, tomar decisões complexas para resolver problemas críticos rapidamente, garantindo assim a qualidade.

3.2.2. Responsabilidades e Funções do Desenvolvedor

Desempenhar um papel chave para a criação e revisão das *ADRs*, participar ativamente dos debates apresentando opiniões baseadas em critérios técnicos e de regras de negócio, focando sempre na melhoria contínua do sistema e da documentação. Zimmer (2023) aponta questões orientadoras que todo desenvolvedor presente no projeto deve seguir para sugerir argumentos que contribuam para criação de decisões bem fundamentadas. Essas indagações estão apresentadas na Tabela 4.

Dessa maneira, permitindo a atuação como um agente balanceador, responsável por integrar diferentes perspectivas e garantir que as decisões tomadas sejam bem fundamentadas, alinhadas aos objetivos dos *stackholders* e viáveis na prática para implementação pela equipe.

Questão Orientadora
O problema é relevante o suficiente para ser resolvido e registrado em um ADRs?
As soluções propostas têm chance de resolver o problema? Existem outras opções válidas que podem ser aplicadas?
Os critérios usados para se escolher uma solução são bem definidos e abrangem todos os aspectos importantes?
Caso haja conflitos de critérios, existe uma priorização para seguir?
A justificativa para a escolha da solução do problema é bem fundamentada e persuasiva?
As consequências positivas e negativas da solução escolhida são relatadas de maneira imparcial e detalhada?
A solução escolhida está descrita de maneira prática e alinhada com os requisitos iniciais?

Tabela 4. Questões orientadoras para Criação e Revisão de ADRs pelo Desenvolvedor segundo Zimmer (2023)

3.3. Fluxo de desenvolvimento de ADRs

Com a equipe formada, o objetivo passou a ser estruturar um processo eficiente e colaborativo para a tomada de decisões arquiteturais. Desde o modelo inicial até a versão mais atual, foram feitos refinamentos para melhorar o engajamento e a qualidade dos debates, e, conseqüentemente, das ADRs criadas. Foi elaborado um diagrama *BPMN*³ (*Business Process Model and Notation*), pois, segundo (OMG) (2011), essa é uma boa forma de comunicar o fluxo de trabalho.

3.3.1. Fluxo de desenvolvimento de ADRs - Versão 1

Definir um *template* onde todas as ADRs são baseadas, conforme recomenda a ADR Community (2023), é fundamental. Essa comunidade informa que não existe um modelo único e absoluto; o *template* deve refletir as necessidades específicas de cada projeto. Aliado a isso, a recomendação desta comunidade é consultar o repositório do GitHub de Joel Parker Henderson (2023), onde foi possível encontrar modelos de *templates* de decisão de (MADR) (2024) e Nygard (2011), que serviram de inspiração para criação do modelo inicial usado neste projeto. Os elementos usados estão na Tabela 5.

Elemento do <i>template</i> de ADRs - V1	Função
Título	Facilitar a busca e organização das ADRs.
Contexto	Fornece o cenário da decisão, incluindo os requisitos, as restrições e os objetivos.
Decisão	Enuncia a solução inicial escolhida para resolver o contexto.
Justificativa	Enunciar o fundamento lógico para a decisão tomada.
Consequências	Descrever os impactos positivos e negativos da decisão.
Alternativas consideradas	Descrever soluções que também foram pensadas para solução do contexto mas que por algum motivo foi rejeitada inicialmente.

Tabela 5. *Template* de ADRs - V1

³<https://www.bpmn.org/>

O processo inicia-se com o arquiteto-líder escrevendo a *ADRs* utilizando o *template* previamente definido em um documento de texto através da ferramenta do Google Documentos⁴, pois, segundo Google (s.d.), essa ferramenta permite armazenar documentos com segurança e disponibilizá-los para outros integrantes do projeto de maneira facilitada.

Conforme a Figura3, com a *ADRs* escrita e documentada, o próximo passo é o arquiteto marcar uma reunião com os integrantes da equipe para discutir os pontos positivos e negativos dessa decisão arquitetural, propor melhorias e novas soluções através da troca de perspectivas que cada integrante traz para o debate. A reunião ocorre em um dia e horário em que todos os membros possam comparecer, e tem o objetivo de proporcionar um ambiente inclusivo onde todos são incentivados a participar sem julgamento.

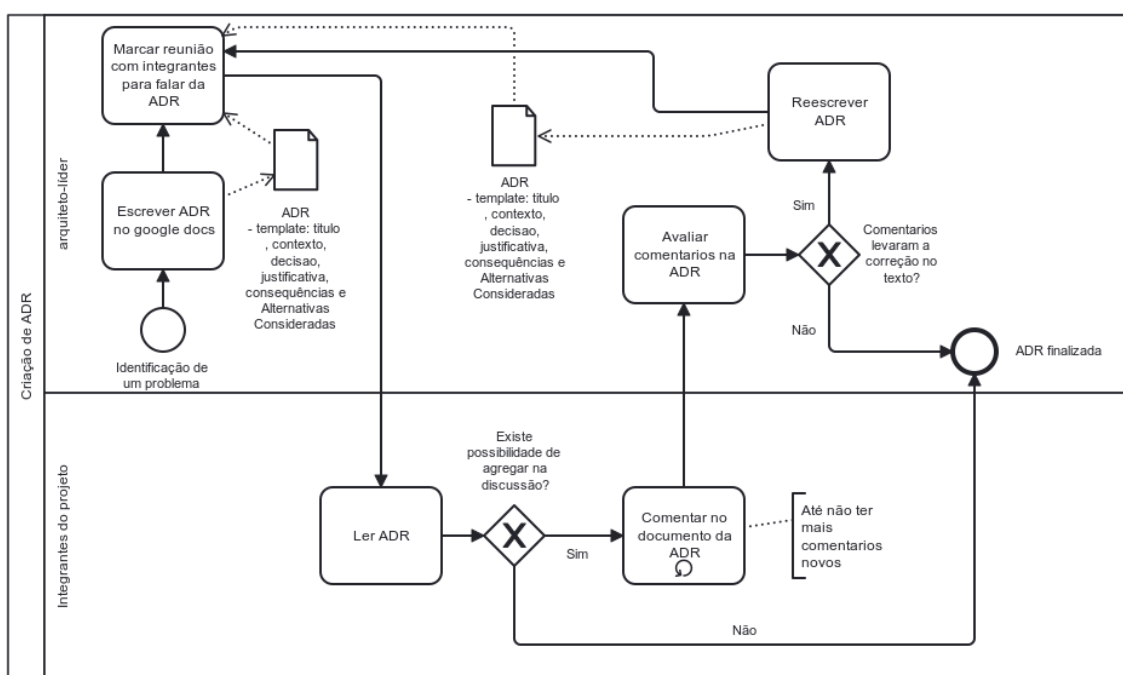


Figura 3. Fluxo de criação de *ADRs* no Google Documentos - Versão 1

A discussão sobre a *ADRs* pode ter dois resultados. Ao finalizar o debate, o arquiteto assume a responsabilidade de ponderar se é necessário refinar a *ADRs* de acordo com os comentários existentes no documento, ou se caso contrário, o mesmo deve convencer a equipe de que a ideia inicial proposta deve ser mantida. Desse modo, proporcionando um ambiente inclusivo e participativo nas decisões tomadas. Entretanto, caso haja necessidade de reescrever a *ADRs*, o arquiteto a reformula e apresenta na próxima reunião. O processo se repete até que a *ADRs* seja finalizada. Além disso, caso o arquiteto tenha alguma nova *ADRs*, ela pode ser debatida logo após a apresentação e finalização da *ADRs* reformulada. Um exemplo de *ADRs* criada usando este modelo está no Anexo A.

⁴ <https://workspace.google.com/intl/pt-BR/products/docs/>

3.3.2. Fluxo de desenvolvimento de ADRs - Versão 2

Ao finalizar a primeira ADRs, a equipe identificou a possibilidade de melhorar o controle sobre as ADRs. Sendo assim, os modelos de *template* de (MADR) (2024) e Nygard (2011) foram revisitados em busca de elementos que pudessem ser adicionados para ajudar a aprimorar o *template* atual, resultando na versão V2, detalhada na Tabela 6.

Elemento do <i>template</i> de ADRs - V2	Função
Número Serial	Identificar a ADRs de forma única para rastreamento.
Título	Facilitar a busca e organização das ADRs.
Data de Criação	Registrar a data em que a ADRs foi escrita, facilitando a organização cronológica.
Status	Indicar o estado atual da ADRs (<i>Pendente, Rejeitada ou Concluída</i>).
Contexto	Fornecer o cenário da decisão, incluindo os requisitos, as restrições e os objetivos.
Decisão	Enunciar a solução inicial escolhida para resolver o contexto.
Justificativa	Enunciar o fundamento lógico para a decisão tomada.
Consequências	Descrever os impactos positivos e negativos da decisão.
Alternativas Consideradas	Descrever soluções que também foram pensadas para solução do contexto, mas que foram rejeitadas inicialmente.
Referências	Documentar as fontes de ideias que influenciaram a ADRs.

Tabela 6. *Template* de ADRs - V2

Implementar essas mudanças no *template* resultou em ajustes e adições de etapas no fluxo de criação e debate de ADRs. O fluxo se inicia com a escrita da ADRs no Google Documentos, sucessivamente é definido o *status* de "*Pendente*" para a ADRs, indicando que ainda não foi debatida entre os integrantes do projeto, em sequência a reunião com os membros é marcada para que a ADRs seja discutida. Com a reunião iniciada, o processo de debate segue a mesma lógica da versão anterior.

Sobretudo, com a adição do elemento "*status*", o arquiteto pode definir se a ADRs é relevante para o projeto com o *status* "*Aceita*", caso a proposta tenha sido aprovada pela equipe para fazer parte do projeto, ou, "*Rejeitada*" caso os membros entrem em consenso que o contexto e a solução apresentado sob nenhuma circunstância devem fazer parte do escopo. Desse modo, a ADRs é finalizada. A Figura 4 ilustra esse processo, e é possível visualizar uma ADRs criada usando este modelo no Anexo B.

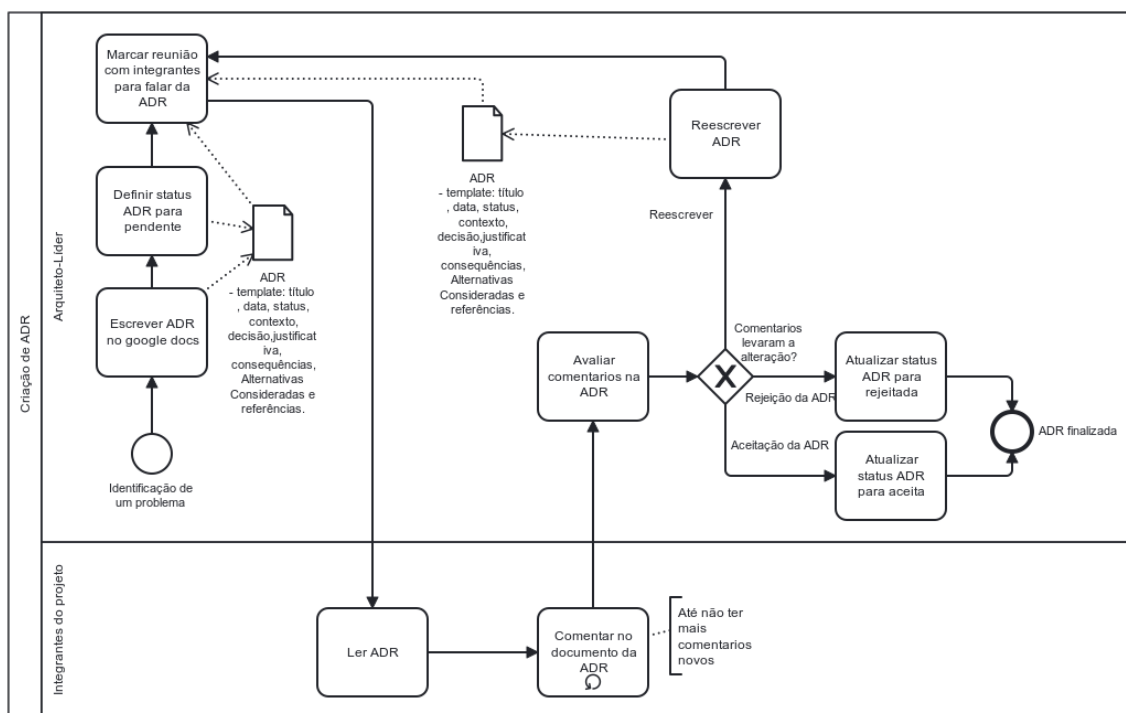


Figura 4. Fluxo de criação de ADRs no Google Documentos - Versão 2

3.3.3. Fluxo de desenvolvimento de ADRs - Versão 3

Armazenar ADRs no Google Drive tornou-se custoso para o debate fluido, devido às ferramentas disponibilizadas pelo ambiente dificultar o acompanhamento de múltiplos comentários e do histórico de versões, tornando difícil a análise para o arquiteto-líder. Além disso, a dificuldade dos integrantes participarem das reuniões presenciais devido a horários conflituosos, fez surgir a necessidade de buscar uma solução onde o debate pudesse ocorrer de forma assíncrona, permitindo a participação de todos.

Segundo recomendações do Google (2024), armazenar as ADRs perto do código fonte da aplicação, de preferência no mesmo sistema de controle de versão do projeto é uma maneira eficaz de manter o projeto e as ADRs atualizados, consequentemente elucidando os problemas enfrentados ao usar a plataforma do Google. Com isso, foi criado um repositório no GitHub⁵ para gerenciar de forma facilitada o controle de versão das ADRs.

Utilizar o Git como sistema de versionamento possibilitou maior eficiência e organização no desenvolvimento do projeto. Ferramentas como Git Log⁶, para o rastreamento das alterações feitas nas ADRs, Git Branch⁷, para separar as ramificações em desenvolvimento e de Pull Requests⁸ para organizar devidamente cada discussão de ADRs, foram essenciais para criar um ambiente mais produtivo, conforme pode ser visto neste debate presente no Anexo E.

⁵ <<https://GitHub.com/>>

⁶ <<https://git-scm.com/book/pt-br/v2/Fundamentos-de-Git-Vendo-o-hist%C3%B3rico-de-Commits>>

⁷ <<https://git-scm.com/docs/git-branch>>

⁸ <<https://docs.GitHub.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>>

Administrar o repositório é a função do arquiteto-líder, no qual foi estruturado em pastas, começando pela raiz do projeto com o diretório 'ADRs', que tem por objetivo armazenar todas as decisões arquiteturais aceitas e negadas pela equipe. Conforme pode ser visto na Figura 5, o líder afim de agilizar o processo de criação de documentos, passou a iniciar a escrita das *ADRs* em uma branch⁹, criada a partir da main, seguindo o *template* já previamente definido.

Ao finalizar o processo de escrita, o status da *ADRs* é marcado como '*aceita*' e é feito *merge* dela no repositório. Em seguida, os integrantes são notificados; caso alguém tenha algo a questionar, abre-se uma *pull request* com os questionamentos, caso contrário a *ADRs* constará como finalizada e o processo se repetirá para as próximas decisões arquiteturais criadas. O Anexo C evidencia esse comportamento sendo realizado.

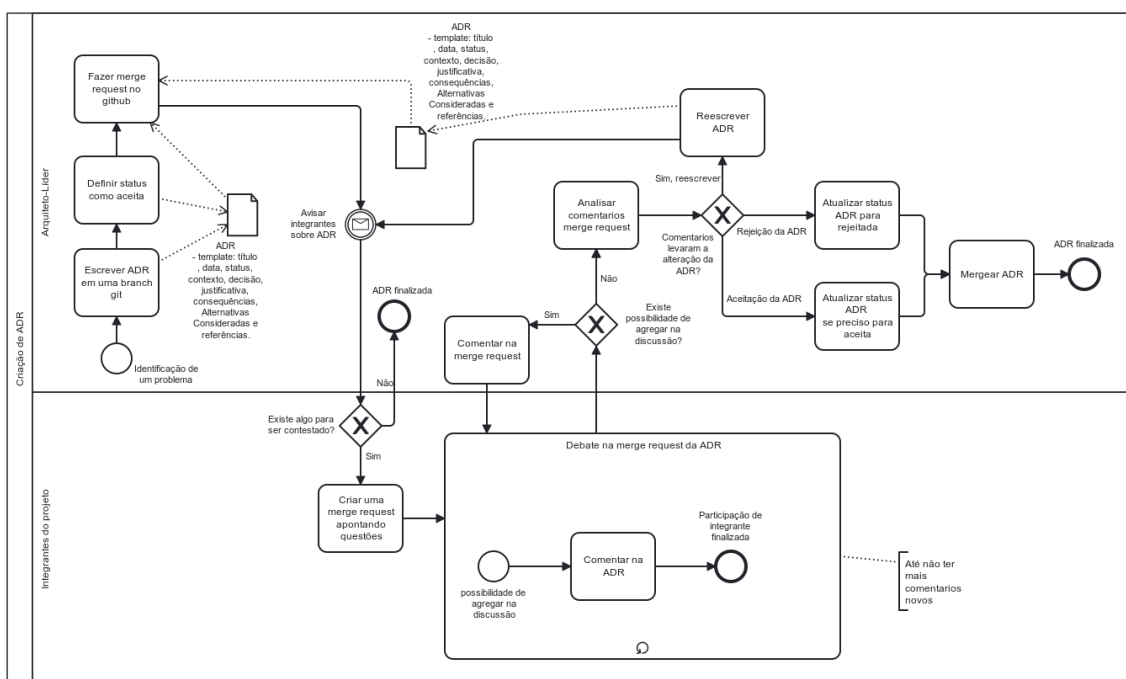


Figura 5. Fluxo de criação de ADRs no GitHub - Versão 3

3.3.4. Fluxo de desenvolvimento de *ADRs* - Versão 4

Usar somente o modelo assíncrono de desenvolvimento e debate de *ADRs*, revelou uma significativa falta de engajamento dos integrantes em acessar o repositório para ler as decisões arquiteturais ou criar *merge request* para questionar as decisões propostas pelo arquiteto-líder. Consequentemente, gerou-se um acúmulo de *ADRs* criadas e não debatidas. Sendo assim, surgiu a necessidade de reavaliar o fluxo para que haja um aumento no comprometimento da equipe com o projeto.

O diagrama *BPMN* apresentado na Figura 6 ilustra o fluxo atualizado de criação e debate de *ADRs* na versão 4, onde, incorpora mudanças feitas para elucidar os problemas mencionados anteriormente. Nesse sentido, uma das principais alterações foi a

⁹<https://docs.GitHub.com/pt/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>

implementação de uma reunião semanal online via Google Meet¹⁰ as segundas-feiras, com o objetivo de aproximar os integrantes ao projeto, e fazer com que o arquiteto-líder pudesse identificar os membros mais distantes e engaja-los com maior foco. Para melhor rastreabilidade e transparência das ações foi elaborado uma ata a cada reunião realizada para registrar as discussões, decisões e encaminhamentos definidos. Além disso, o objetivo da reunião é o de revisar o que foi debatido durante a semana anterior, para iterar todos os integrantes sobre o andamento do projeto, finalizar as *ADRs* que estão com status “*pendente*”, e, planejar quais *ADRs* serão discutidas ao longo da semana seguinte.

Sob a mesma perspectiva, foi implementado uma rotina para acompanhamento contínuo do progresso dos debates. Na qual o líder, nas quartas e sextas-feiras realiza uma verificação, avaliando os novos comentários feitos pelos integrantes até então, e, se necessário, adiciona seu ponto de vista para enriquecer as discussões. Posteriormente, o arquiteto-líder envia notificações ao time, destacando a relevância da participação ativa de cada membro e reforçando a importância das contribuições para o avanço do projeto.

Conforme destacado por Greiler, Storey e Noda (2021), a falta de um ambiente colaborativo pode reduzir o engajamento dos integrantes. Portanto, com o objetivo de promover maior interatividade dos membros com o projeto e estimular a criação de mais *ADRs*, foi implementada a possibilidade de sugerirem propostas para serem debatidas através da ferramenta de Issues¹¹ do GitHub, na qual os integrantes podem registrar sugestões de *ADRs*. Cada sugestão é submetida a uma votação para autenticar se é uma ideia válida para se criar um debate arquitetural ou se deve ser descartada.

¹⁰ <<https://meet.google.com/landing>>

¹¹ <<https://docs.GitHub.com/pt/issues/tracking-your-work-with-issues/about-issues>>

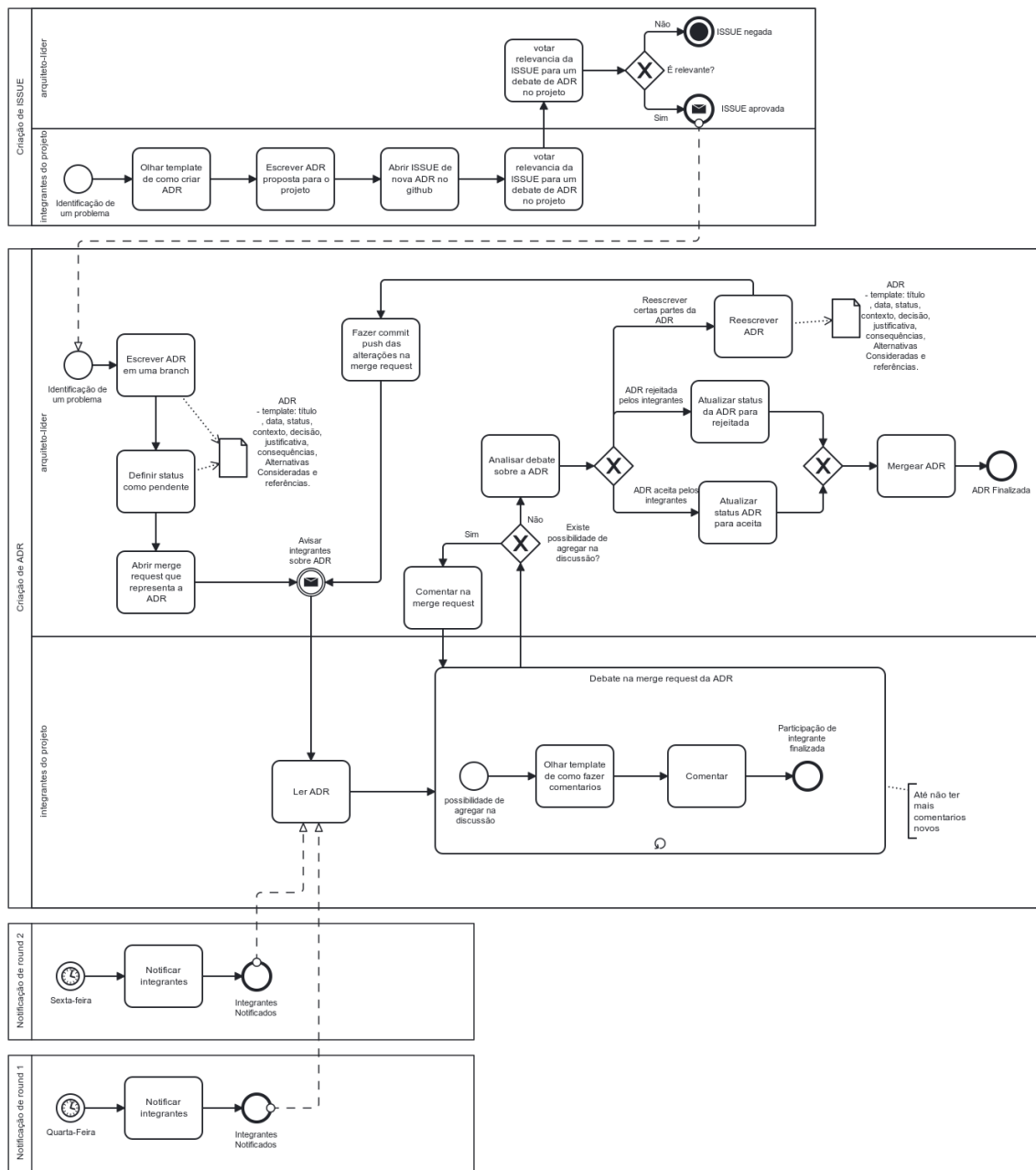


Figura 6. Versão 4 do fluxo de criação de ADRs - Diagrama BPMN

Outro ajuste relevante foi realizado a partir do feedbacks do grupo relatando a dificuldade em realizar comentários nas *PRs* abertas. A incerteza sobre como se posicionar ou interagir com opiniões divergentes, o que, segundo informado, frequentemente resultava na desistência na participação de debates. Portanto, em uma reunião colaborativa com o time, foi aberto um espaço onde os integrantes fizeram sugestões de perguntas orientadoras, que serviram de base para a elaboração de um *template* de como realizar comentários objetivos, concisos e relevantes, conforme a Tabela 7. A utilização do *template* para este tipo de interação pode ser observada em exemplo detalhado no Anexo D.

Questão Orientadora	Descrição
Qual a Avaliação sobre a decisão ou proposta?	Opinião geral sobre a decisão ou proposta apresentada.
Questões Levantadas?	Listar perguntas ou preocupações sobre a decisão ou proposta.
Motivo da Concordância ou Discordância?	Relatar porque é a favor ou contra tal decisão ou proposta.
Alternativas Sugeridas? (se discorda)	Se discorda da decisão ou proposta, relatar quais alternativas podem ser consideradas.
Adições Complementares? (se concorda)	Se concorda com a decisão ou proposta, sugerir ideias para melhorá-la se houver.

Tabela 7. Template de Comentário em *Merge Request* ou Issue de ADRs

3.3.5. Fluxo de desenvolvimento de ADRs - Versão 5

A fim de avaliar o limite de engajamento possível de ser extraído do grupo, aumentou-se o número de *rounds* de notificação estabelecidos anteriormente na Versão 4 do fluxo de desenvolvimento de ADRs. Assim, foram implementadas notificações diárias no grupo de comunicação da equipe, com o objetivo de reforçar o compromisso dos integrantes e estimular maior participação nos debates.

Assegurar o consenso da equipe nas soluções arquiteturais em debate revelou-se de suma importância. Com a participação ativa e engajada dos integrantes, surgiram propostas de soluções alternativas, como ilustrado no Anexo F. Sendo assim, foi decidido implementar um mecanismo de votação para lidar com esses casos, conforme pode ser visto no Anexo G. Esse sistema permite dois possíveis resultados: se a solução atual da ADRs receber a maioria dos votos, ela será marcada como “*aceita*”, encerrando o fluxo de desenvolvimento relacionado. Caso a solução alternativa seja escolhida, a ADRs atual será marcada como “*rejeitada*”, e uma nova ADRs será criada baseada na solução alternativa aprovada.

O aumento considerável na quantidade de ADRs criadas, resultante da implementação das Issues, que embora tenha promovido maior interação, também gerou sobrecarga para a equipe. Essa situação tornou inviável acompanhar todas as discussões de forma eficiente. Para solucionar esse problema, foi acordado com o time um limite semanal de cinco ADRs em debate. Esse número foi definido com base na média de ADRs resolvidas nas semanas anteriores e considerando a capacidade do grupo de interagir nas discussões.

Dessa forma, à medida que uma *ADRs* dentro da lista de cinco for finalizada, uma nova *ADRs* irá imediatamente ser inserida no ciclo de debates. Esse sistema busca manter um fluxo contínuo de discussões, equilibrando a quantidade de trabalho com a capacidade de engajamento da equipe. A Figura 7 apresenta o diagrama *BPMN* atualizado para a Versão 5, ilustrando as mudanças descritas.

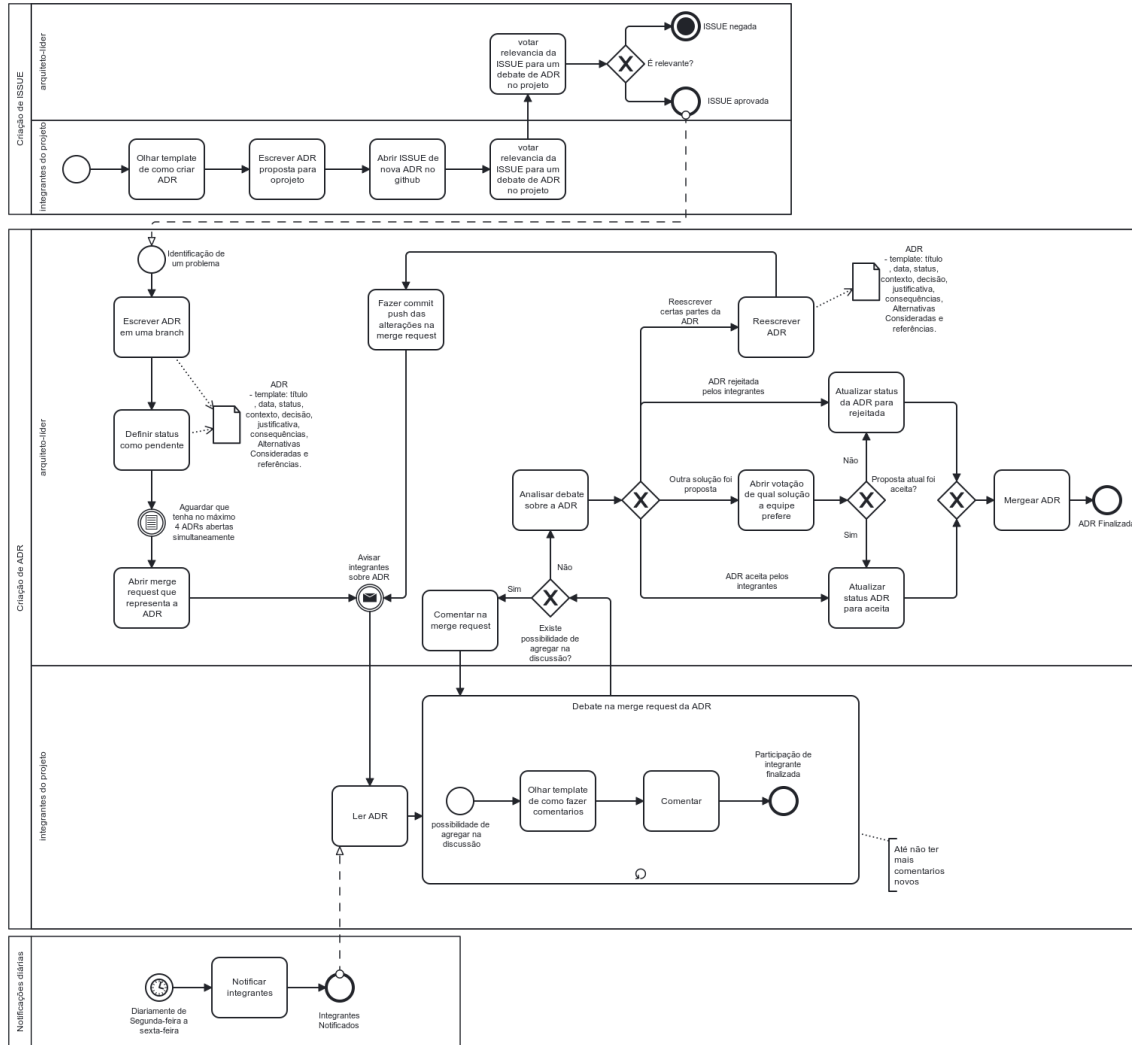


Figura 7. Versão 5 do fluxo de criação de *ADRs* - Diagrama *BPMN*

Em suma, a Versão 5 representa o grau mais desenvolvido alcançado no processo de desenvolvimento de *ADRs* deste projeto, integrando as melhorias realizadas e lições aprendidas nas versões anteriores. A interação entre os integrantes buscada é um esforço para manter um equilíbrio entre a produtividade e a capacidade da equipe de se envolver no processo e, ao mesmo tempo, fornece um ecossistema de tomada de decisões arquiteturais bem organizado e cooperativo. Dessa forma, sintetiza maneiras mais eficazes de práticas vindas do desenvolvimento do projeto e aumenta a taxa de alinhamento e participação da equipe.

4. Resultados e Discussões

Para realizar as análises, foram utilizadas 3 ferramentas para auxiliar na interpretação dos resultados. O **Google Formulários**¹² foi essencial para extrair dos participantes do projeto suas experiências, desafios e benefícios percebidos ao adotar *ADRs* como uma prática de documentação de decisões arquiteturais. Essa ferramenta também permitiu coletar informações sobre o impacto das estratégias de enfrentamento adotadas durante o desenvolvimento das *ADRs*, analisando a influência pessoal e profissional, bem como a eficácia percebida de cada uma delas a partir da perspectiva dos desenvolvedores novatos.

Além disso, **GitHub**¹³ e **Google Documents**¹⁴ desempenharam um papel crucial ao fornecer o histórico quantitativo e qualitativo das interações dos usuários com as *ADRs*. Portanto, permitindo uma análise detalhada dos registros de colaboração de cada membro no projeto, desse modo, ajudando a compreensão mais abrangente dos procedimentos.

4.1. QP1: De que forma a construção de *ADRs* influencia o engajamento e a contribuição dos membros de uma equipe de desenvolvimento de software?

A Figura 8 evidencia a participação de todos os integrantes da equipe conforme as versões de fluxo de desenvolvimento foram se aperfeiçoando com o tempo e o time foi se entrosando e acostumando.

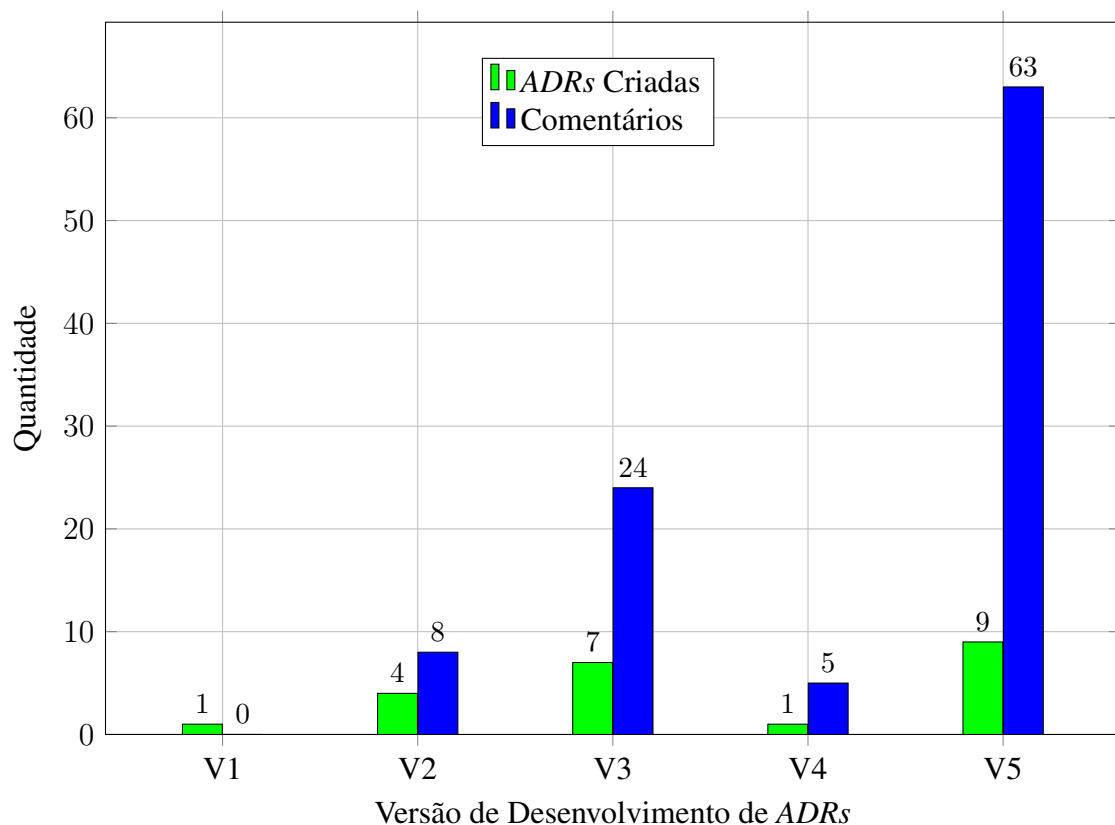


Figura 8. Evolução do Número de ADRs e Comentários por Versão

¹² <https://www.google.com/intl/pt-BR/forms/about/>

¹³ <https://GitHub.com/>

¹⁴ <https://workspace.google.com/intl/pt-BR/products/docs/>

No contexto, a V1 teve um início abaixo do esperado pois somente uma *ADR*s foi criada (presente no Anexo A) e nenhuma interação da equipe comentando sobre a proposta aconteceu, esse comportamento pode ser explicado devido a adaptação inicial da equipe ao processo de registro e documentação das decisões arquiteturas, onde os integrantes se mostraram sem saber o que falar em relação ao que foi proposto como solução pelo líder-arquiteto, que até então era enxergado como um ser que não deve ser questionado, por isso ocorreu o receio de uma primeira interação e a retaliação que isso poderia gerar, bem como o sentimento de julgamento que os outros integrantes, até então desconhecidos, do grupo poderiam ter sobre quem estava interagindo.

A V2 demonstrou uma evolução significativa de *ADR*s criadas, e, um aumento do número de comentários, indicando os primeiros passos rumo a um ambiente engajado da equipe com o projeto, isso aconteceu devido a familiarização dos membros um com os outros e com a problemática do projeto. Ao todo, nesta etapa do desenvolvimento usando o Google Drive resultou no Gráfico da Figura 9.

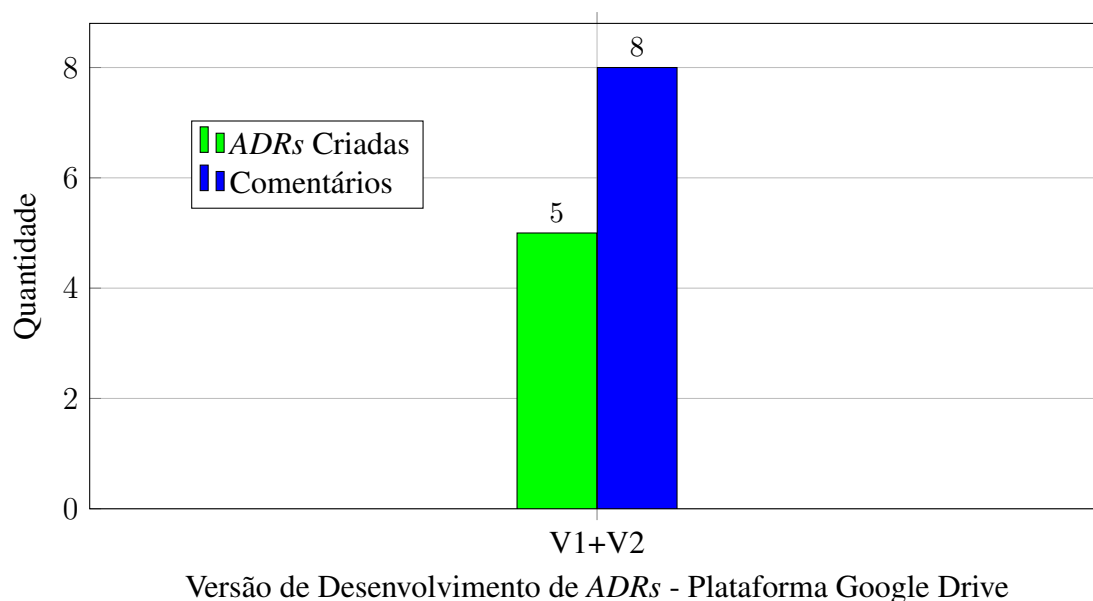


Figura 9. Desenvolvimento de *ADRs* na Plataforma Google Drive

Devido às limitações do Google Drive, nas versões subsequentes V3, V4 e V5 foram feitas mudanças significativas. O uso a partir de então do GitHub, proporcionou um ambiente mais robusto para controle e registro de interações, resultando em um aumento no engajamento dos membros conforme representado no Gráfico da Figura 10.

O novo repositório e metodologia possibilitou identificar as barreiras técnicas e comportamentais que a equipe estava enfrentando. Desse modo, foi possível implementar medidas como a interação assíncrona que possibilita aos integrantes colaborar com as *ADRs* a qualquer momento e lugar, reuniões semanais para proporcionar um espaço para interação de forma mais direta e promover a construção de relacionamento positivo entre os colegas e o uso de *rounds* que possibilitou ao líder monitorar o progresso de maneira estruturada, e incentivar a participação, colaboração e engajamento contínuo da equipe de forma mais eficiente baseado-se em estatísticas proporcionadas pelo repositório.

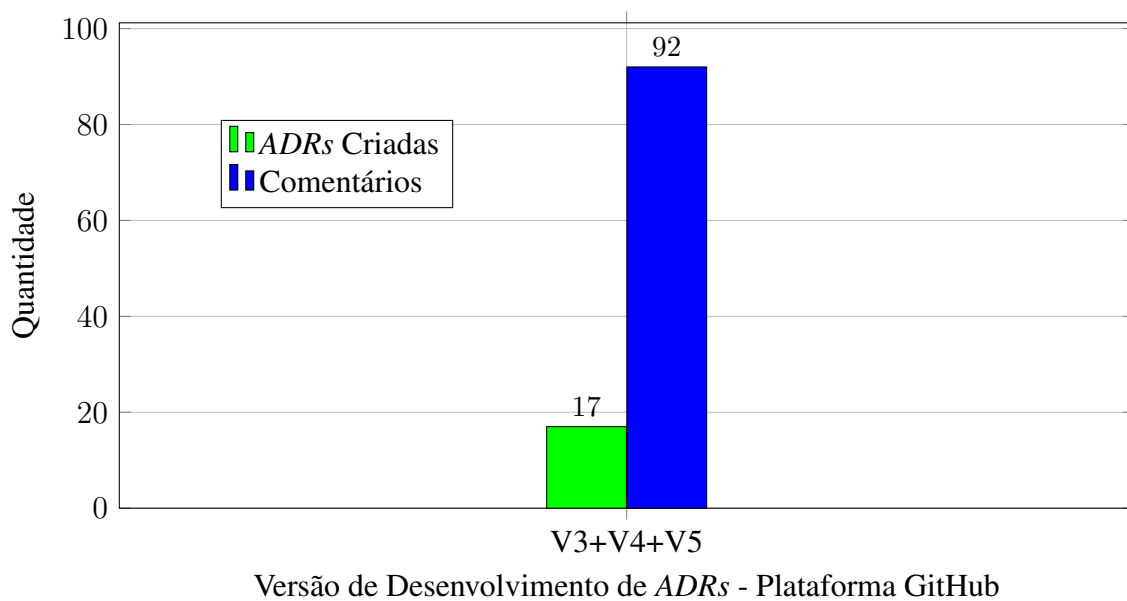


Figura 10. Desenvolvimento de ADRs na Plataforma GitHub

Portanto, durante a etapa inicial do projeto, onde foi usado o Google Documents do Google Drive, o impacto esperado das ADRs sobre o time ocorreu de forma mais lenta. Isso se deveu, à incerteza dos integrantes em relação a como interagir com a metodologia, já que é algo que nunca haviam interagido antes. Associado a isso, a ferramenta do Google demonstrou ser limitada e ineficaz para promover o engajamento do time, devido a sua interface e funcionalidades que não atendem de forma satisfatória às demandas deste projeto.

Com a transição para o GitHub e a maturação do projeto, novas estratégias implementadas, como a adição de fluxos auxiliares para dar apoio aos integrantes, foram adotadas. Essas mudanças trouxeram vantagens significativas, como um ambiente mais propício à discussão, registro de alterações e integração com ferramentas de controle de versão, facilitando a participação ativa dos membros.

A Figura 11, ilustra uma linha do tempo organizada por versões (V1 a V5) dos fluxos de desenvolvimento existentes ao longo do projeto, onde está estipulado os períodos que cada versão teve e a quantidade de *rounds* que os integrantes participaram.

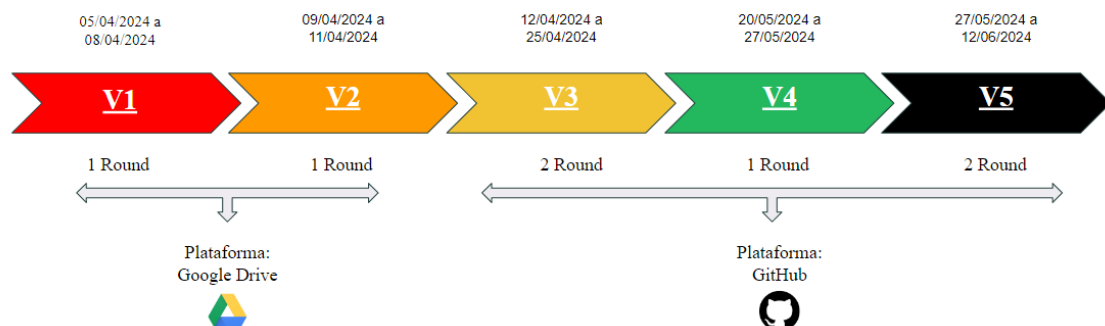


Figura 11. Linha do tempo das versões das ADRs

Por fim, a implementação do uso de Issues no GitHub ajudou a transformar o envolvimento dos integrantes. Como mostrado no Gráfico 12, os membros começaram a propor soluções e participar mais ativamente, para que suas ideias fossem implementadas no projeto. Portanto, é possível observar que com o passar do tempo, resultou-se um ganho de experiência e conhecimento em cima do domínio, onde a participação dos desenvolvedores novatos passou de passiva para ativa, promovendo um ambiente que estimula em seus participantes a troca de ideias, aprendizado contínuo e a cultura de liderança.

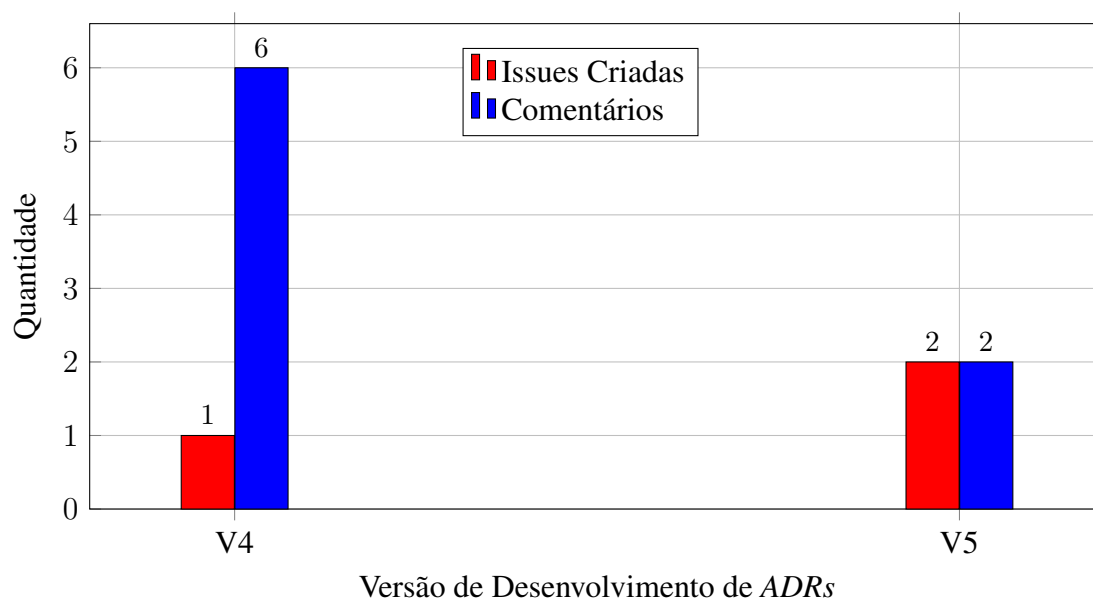


Figura 12. Comparação de Issues e Comentários entre as Versões V4 e V5

O engajamento causado é evidenciado pelo gráfico na Figura 13, que demonstra como parte dos integrantes consegue enxergar e propor soluções alternativas através da ferramenta de Issues e comentários nas PRs no GitHub. Entretanto, o gráfico também indica que alguns integrantes enfrentaram dificuldades nesse aspecto, reforçando a importância do apoio coletivo e do tempo de maturação para todos no projeto.

A barreira de Sem Soluções Viáveis te afetou ? Sem Soluções Viáveis: Não há soluções eficazes disponíveis para resolver os problemas identificados na ADR.

4 respostas

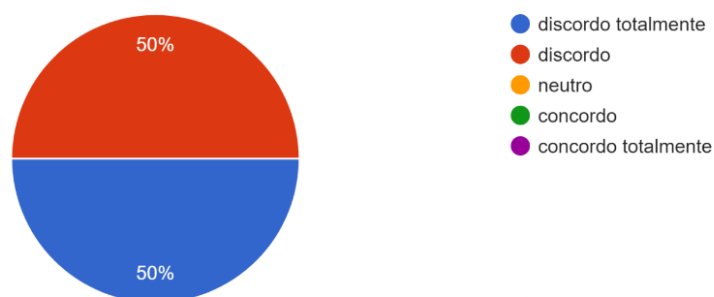


Figura 13

4.2. QP2: Como os membros de uma equipe de desenvolvimento de software percebem a importância das ADRs construídas durante o projeto?

Para responder essa questão, foi aplicado um questionário de Avaliação do Desenvolvimento de ADRs, que em suas respostas trouxeram importantes aspectos individuais e coletivos percebidos através da elaboração de ADRs. Desse modo, o envolvimento dos membros trouxe aspectos que foram relatados na Figura 14.

O que você achou de fazer parte de uma equipe de construção de decisões arquiteturais? *

foi valioso compartilhar experiencias e conhecer um pouco mais sobre outras visões

O que você achou de fazer parte de uma equipe de construção de decisões arquiteturais? *

Uma equipe com foco nessa área possibilitou que discussões pudessem ser muito bem estruturadas e que alcançassem níveis de detalhamento que eu não tinha presenciado antes.

O que você achou de fazer parte de uma equipe de construção de decisões arquiteturais? *

Particularmente, gostei da experiência. Acredito que é essencial para qualquer desenvolvedor desenvolver habilidades em todas as áreas de um projeto, inclusive na parte de documentação das decisões arquiteturais.

O que você achou de fazer parte de uma equipe de construção de decisões arquiteturais? *

Agregou bastante, tanto em conhecimento quanto em desempenho

Figura 14. Percepção dos Integrantes sobre a Participação no Grupo de ADRs

Entretanto, inicialmente houve um período de adaptação para os integrantes se habitarem a essa metodologia de prototipação de ideias, conforme podemos ver no baixo engajamento das primeiras versões do fluxo de desenvolvimento. Em um questionário de retrospectiva aplicado ao final do projeto, os integrantes relataram dificuldades relacionadas ao processo, como pode ser visto neste relato em especial na Figura 15.

Quais as atividades e tarefas que foram executadas durante esse projeto que devemos evitar em projetos futuros? * 1. Falta de Clareza Inicial nas Decisões: Em algumas ocasiões, as decisões documentadas nos ADRs não foram totalmente discutidas ou amadurecidas antes de serem implementadas, levando a revisões posteriores que poderiam ter sido evitadas. 2. Documentação Excessiva: Em alguns momentos, a documentação de ADRs foi muito detalhada ou extensa, o que consumiu tempo desnecessário sem grande valor agregado. Buscar um equilíbrio entre a profundidade da documentação e a produtividade pode ser interessante importante para futuros projetos. 3. Revisão Tardia de ADRs: A revisão de algumas decisões foi deixada para momentos muito próximos do prazo final, o que gerou pressão em decisões críticas. Planejar revisões com mais antecedência pode evitar esse problema. Quais as atividades e tarefas que foram executadas durante esse projeto que devemos evitar em projetos futuros? * em certo modo, as reuniões presenciais vejo que poderiam ser trocadas em certos momentos, porém são importantes em certos momentos	Quais as atividades e tarefas que foram executadas durante esse projeto que devemos evitar em projetos futuros? * A baixa adesão de um membro pode ser tratada de formas diferentes e mais incisivas do que a que foi empregada no projeto. Quais as atividades e tarefas que foram executadas durante esse projeto que devemos evitar em projetos futuros? * Não realizar um alinhamento prévio de forma individual com os participantes de um projeto para que ele possa ser alocado nas ADR's que ele tenha capacidade de colaborar diretamente no projeto e nas decisões arquiteturais.
---	---

Figura 15. Relato de barreiras sofridas pelos integrantes

Aliado a isto, Greiler, Storey e Noda (2021) indica barreiras que podem prejudicar o desempenho do desenvolvedor no dia a dia. Desse modo, ao indagar os membros sobre essas barreiras, observou-se, através dos indicativos da Figura 16, que mais da metade dos integrantes relataram ter enfrentado as tais barreiras que afetam a capacidade de avaliação,

compreensão e disposição. De acordo com o estudo, essas barreiras contribuíram para uma participação menos efetiva nas etapas de criação e debate das *ADRs* como um todo.



Figura 16. Indicadores de barreiras sofridas pelos integrantes

Foi sugerido pelo líder, estratégias individuais e coletivas apresentadas por Greiler, Storey e Noda (2021), que quando aplicadas se mostraram eficazes em diferentes graus para destravar e fazer o grupo dar continuidade no desenvolvimento do trabalho, apesar das barreiras. As Figuras 18 e 17 evidenciam a percepção quantitativa dos integrantes sobre as estratégias individuais e coletivas aplicadas ao longo do projeto.

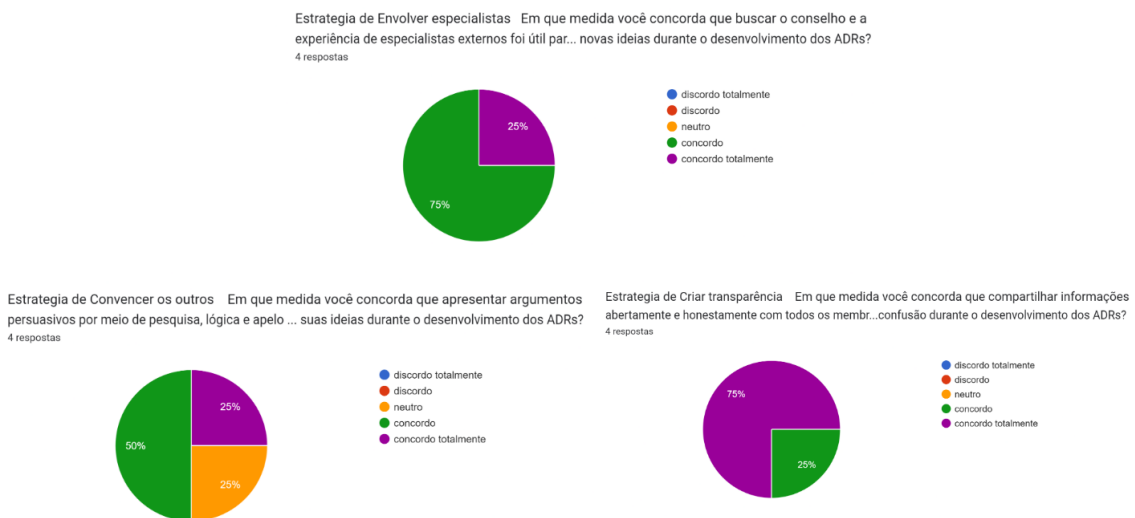


Figura 17. Estratégias coletivas Utilizadas para a Criação e Revisão de *ADRs*

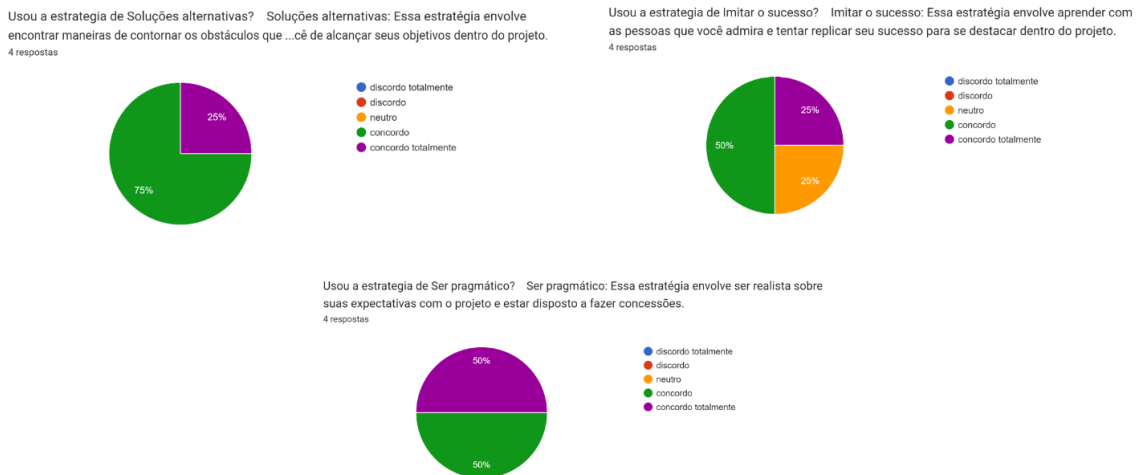


Figura 18. Estratégias Individuais Utilizadas para a Criação e Revisão de ADRs

Além disso, através dos relatos dos integrantes na Figura 19, fica evidente que as estratégias podem ser benéficas se replicadas em outros projetos. Portanto, o projeto acaba refletindo para os membros lições aprendidas e um conjunto de abordagens que fortalecem a colaboração, organização e engajamento de equipes. Desse modo, tais práticas são primordiais para um ambiente de trabalhos mais eficientes, adaptável e alinhado aos objetivos de negócio, capaz de promover sucesso em iniciativas futuras.

Quais as atividades e tarefas foram executadas durante o projeto que devemos repetir em projetos futuros? A estratégia de reconhecimento por participação nas ADRs foi um ponto positivo para o engajamento, pois entregava uma recompensa que reforçava o comportamento. Determinar um intervalo semanal para a discussão das ADRs permitia maior flexibilidade e consequentemente maior participação da equipe. As mensagens para recordar das ADRs em aberto diminuam a vaporização. As reuniões semanais engajavam a equipe e alinhavam as expectativas. A escolha de uma decisão por meio de votação após a discussão de uma ADR permitiu que todos se sentissem participantes da decisão. Se manter flexível e em constante mudança para adequação das especificidades da equipe foi um ponto crucial para o sucesso da proposta. Trazer ADRs que foram resultantes de discussões que ocorriam em outras ADRs ajudou no sentimento de coletividade e sucesso.	Quais as atividades e tarefas foram executadas durante o projeto que devemos repetir em projetos futuros? 1. Documentação de ADRs: A prática de usar Architecture Decision Records (ADRs) para registrar decisões arquiteturais ajudou a organizar o desenvolvimento, permitindo rastreabilidade de escolhas técnicas. Isso facilitou discussões posteriores e evitou repetição de erros. 2. Colaboração: O uso de ADRs em conjunto com o trabalho colaborativo entre os membros da equipes foi eficaz, assegurando que todas as partes estivessem alinhadas nas decisões. 3. Revisões Periódicas: Revisões regulares dos ADRs, especialmente em momentos críticos de mudança de requisitos ou contexto, foram muito úteis para garantir que as decisões estivessem atualizadas e refletissem as melhores práticas.
Quais as atividades e tarefas foram executadas durante o projeto que devemos repetir em projetos futuros? discussões assíncronas sobre as PRs e usar a própria PR para discutir	Quais as atividades e tarefas foram executadas durante o projeto que devemos repetir em projetos futuros? Revisões contínuas sobre o nível de organização e comunicação durante o processo de desenvolvimento.

Figura 19. Estratégias Bem-Sucedidas para a Criação e Revisão de ADRs

Portanto o desenvolvimento de ADRs ajudou a evitar barreiras como baixa priorização, falta de propriedade intelectual, falta de incentivos e questões político-organizacionais, conforme pode visto na Figura 20.

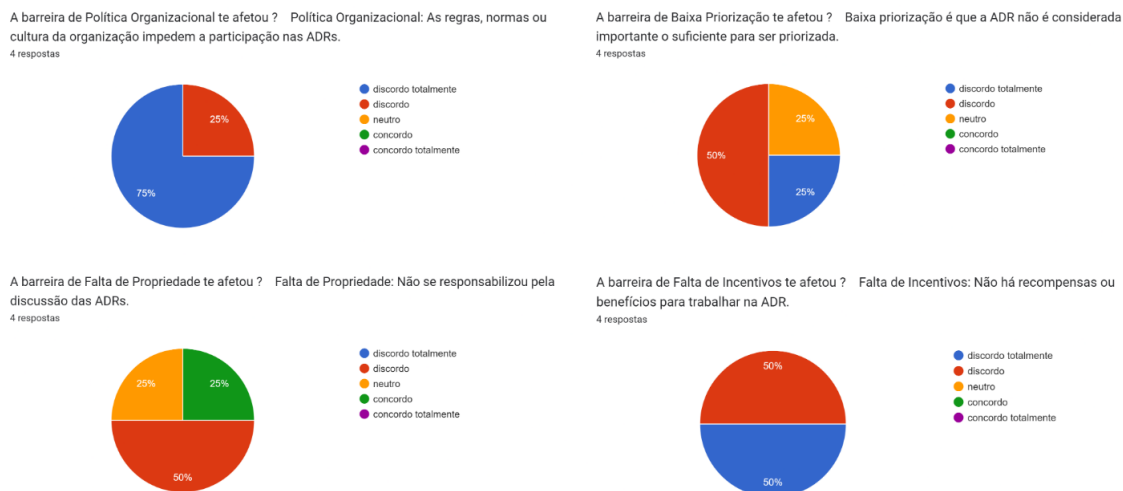


Figura 20. Indicadores de Engajamento com ADRs que superaram barreiras

Sendo assim, um modelo de desenvolvimento reconhecido pela equipe deste projeto como um elemento crucial, principalmente quando combinada com as revisões periódicas. No qual, contribuiu para diminuição da frustração do grupo em relação ao projeto, conforme ilustrado na Figura 21.

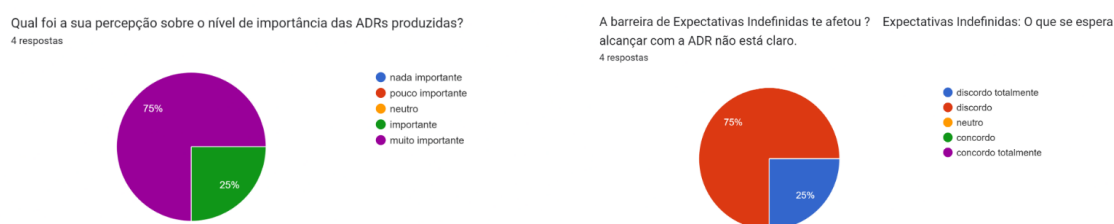


Figura 21. Percepção Geral dos Integrantes sobre as ADRs

5. Conclusão

Neste estudo, foi analisado o impacto do uso de *Architecture Decision Records* (ADRs) na tomada de decisões arquiteturais em um projeto de software composto por uma equipe de 6 membros. Dessa forma, procurou-se entender como esse tipo de documentação influencia no engajamento, colaboração e na contribuição da equipe. Com a problemática do software definida e a equipe estruturada, foi criado fluxo de trabalho que passou por cinco versões, cada uma com melhorias graduais, abrangendo debates e documentações das decisões realizadas entre Abril a Junho de 2024. Os resultados a partir de uma análise teórica e prática, deixaram constatado que o registro estruturado de decisões arquiteturais desempenha um papel essencial na promoção da transparência, colaboração e alinhamento entre os membros da equipe.

Nas versões iniciais (V1 e V2), desenvolvidas no Google Drive, a equipe enfrentou desafios de familiaridade com as práticas de ADRs e limitações da plataforma, que deixaram evidente uma dificuldade na dinâmica colaborativa eficiente. A introdução de novas estratégias e a migração do repositório para o GitHub nas versões V3, V4 e V5,

possibilitou criar um ambiente mais propício à colaboração, auxiliado pela adição de novas práticas como interações assíncronas, reuniões semanais e *rounds* de monitoramento. Dessa forma, aumentando o engajamento e participação dos integrantes.

Conclui-se que o uso de *ADRs*, quando aliado a ferramentas adequadas, facilita o registro rastreável da arquitetura de um software. Além disso, promove coletivamente maior colaboração, engajamento e resultados mais consistentes e alinhados com os objetivos do projeto. Individualmente, as *ADRs* aumentam a visibilidade e o entendimento dos membros sobre o contexto e as motivações do projeto, contribuindo para o seu desenvolvimento profissional.

6. Trabalhos Futuros

Para trabalhos futuros, busca-se aumentar a quantidade de integrantes em um novo projeto e melhorar a percepção e extração de dados. Além disso, realizar uma análise qualitativa mais aprofundada sobre os integrantes a fim de explorar questões emocionais e de saúde mental ligadas a produção de *ADRs*. Com foco em encontrar nuances que afetem a participação dos integrantes em um projeto de software.

Referências

ADR Community. Architectural Decision Records (ADR). 2023. Acesso em: 05 abr. 2024. Disponível em: <<https://adr.github.io/>><https://adr.github.io/>.

CARVALHO, L. de O.; CONTE, T. U. Tomada de decisões arquiteturais em projetos de software: um estudo exploratório com profissionais. In: Congresso Ibero-Americano em Engenharia de Software - CIBSE. [s.n.], 2024. Disponível em: <<https://repositorio.usp.br/directbitstream/25bd8a88-9799-42ee-bec2-3d2630da2135-/3200320.pdf>><https://repositorio.usp.br/directbitstream/25bd8a88-9799-42ee-bec2-3d2630da2135-/3200320.pdf>.

Google. Registros de Decisões Arquiteturais. 2024. <<https://cloud.google.com/architecture/architecture-decision-records?hl=pt-br>>. Acesso em: 18 jun. 2024.

GOOGLE. Primeiros passos no Drive. O que você pode fazer no Google Drive. s.d.

GREILER, M.; STOREY, M.-A.; NODA, A. An actionable framework for understanding and improving developer experience. Journal of Transactions on Software Engineering, 2021.

(IBGE), I. B. de Geografia e E. Pesquisa Nacional por Amostra de Domicílios Contínua: Acesso à Internet e à Televisão e Posse de Telefone Móvel Celular para Uso Pessoal 2021. 2021. Acesso em: 15 jun. 2024. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv102040_informativo.pdf>https://biblioteca.ibge.gov.br/visualizacao/livros/liv102040_informativo.pdf.

INSPER. Mercado de Tecnologia em Constante Evolução: Tendências e Oportunidades de Carreira. 2023. <<https://www.insper.edu.br/noticias/mercado-de-tecnologia/>>. Acesso em 7 de junho 2024.

Joel Parker Henderson. Architecture Decision Record (ADR) templates. 2023. Acesso em: 05 abr. 2024. Disponível em: <https://github.com/joelparkerhenderson/architecture_decision_record>https://github.com/joelparkerhenderson/architecture_decision_record.

KEELING, M. A love unrequited: The story of architecture, agile, and how architecture decision records brought them together. IEEE Software, 2022.

KRUCHTEN, P.; CAPILLA, R.; DUEÑAS, J. C. The decision view's role in software architecture practice. IEEE software, IEEE, v. 26, n. 2, p. 36–42, 2009.

(MADR), M. A. D. R. Decision Record Template of the MADR Project 4.0. 2024. Acesso em: 9 nov. 2024. Disponível em: <<https://adr.github.io/madr/>><https://adr.github.io/madr/>.

MATHUR, S.; MALIK, S. Advancements in the v-model. International Journal of Computer Applications, ACM, v. 1, n. 12, p. 29–34, 2010.

MICROSOFT. Microsoft Application Architecture Guide. 2nd. ed. [S.l.]: Microsoft Press, 2009.

NYGARD, M. Decision Record Template by Michael Nygard. 2011. Acesso em: 9 nov. 2024. Disponível em: <<https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>><https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>.

(OMG), O. M. G. Business Process Model and Notation (BPMN) Version 2.0. [S.l.], 2011. Disponível em: <<http://www.omg.org/spec/BPMN/2.0>><http://www.omg.org/spec/BPMN/2.0>.

SHINZATO, F. et al. So you've graduated college and need to test apps: What barriers might you face? In: Proceedings of the XX Brazilian Symposium on Software Quality (SBQS). Vitória, ES, Brazil: ACM, 2021. p. 1–9.

TYREE, J.; AKERMAN, A. Architecture Decisions: Demystifying Architecture. [S.l.]: Capital One Financial, 2005.

WAN, Z. et al. Software architecture in practice: Challenges and opportunities. In: ESEC/FSE 2023: ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. [S.l.: s.n.], 2023.

WhatsApp FAQ. Qual é o limite de participantes em um grupo do WhatsApp? 2024. Acesso em: 29 out. 2024. Disponível em: <<https://faq.whatsapp.com/3242937609289432>><https://faq.whatsapp.com/3242937609289432>.

ZIMMER, O. How to review Architectural Decision Records (ADRs) — and how not to. 2023. Disponível em: <<https://ozimmer.ch/practices/2023/04/05/ADRReview.html>><https://ozimmer.ch/practices/2023/04/05/ADRReview.html>.

A. Exemplo de ADRs criada utilizando a Versão 1 do Fluxo de criação de ADRs

ADR: Utilização do Framework Flutter para Desenvolvimento Mobile
Contexto
Desenvolvimento de um aplicativo para facilitar o acesso e a comunicação entre usuários e administradores do sistema de transporte público interno da UFMS. O aplicativo precisa ser desenvolvido de forma eficiente, garantindo uma experiência de usuário agradável e ágil.
Decisão
Utilizar o framework Flutter para o desenvolvimento do aplicativo, devido à sua capacidade de criar interfaces de usuário nativas de forma rápida e eficiente para Android e iOS a partir de um único código-base.
Justificativa
• Eficiência no Desenvolvimento: O Flutter permite o desenvolvimento de aplicativos para Android e iOS a partir de um único código-base, reduzindo o tempo e os esforços necessários para manter duas bases de código separadas. • Performance: O Flutter utiliza a linguagem de programação Dart e a renderização direta no hardware. • UI Rica e Responsiva: O Flutter oferece um conjunto abrangente de widgets personalizáveis que permitem criar interfaces de usuário bonitas e responsivas, garantindo uma experiência de usuário agradável. • Suporte da Comunidade: O Flutter é suportado por uma comunidade ativa de desenvolvedores e possui uma ampla gama de plugins e pacotes disponíveis, o que facilita a integração de recursos adicionais ao aplicativo.
Consequências
• Desenvolvimento Mais Rápido: Com o Flutter, podemos desenvolver e lançar o aplicativo mais rapidamente, reduzindo o tempo de desenvolvimento e o custo total do projeto. • Compatibilidade com Diferentes Dispositivos: O Flutter oferece suporte para diferentes tamanhos de tela e dispositivos, garantindo uma experiência consistente para os usuários em dispositivos Android e iOS. • Possível Dependência da Google: Como o Flutter é mantido pela Google, pode haver uma dependência da empresa em relação ao suporte contínuo e atualizações da plataforma.
Alternativas Consideradas
• Desenvolvimento Nativo: Consideramos o desenvolvimento nativo para Android e iOS utilizando Java/Kotlin e Swift/Objective-C. No entanto, essa abordagem exigiria o desenvolvimento e a manutenção de duas bases de código separadas, aumentando o tempo e os custos de desenvolvimento.

Figura 22. ADRs criada a partir da versão 1 do fluxo de criação de ADRs

B. Exemplo de ADRs criada utilizando a Versão 2 do Fluxo de criação de ADRs

Número do ADR: 002

Título: Utilização do git para controle de versão do código fonte da aplicação

Data: 08/04/2023

Status: Pendente

|

Contexto

- Durante o desenvolvimento do sistema, é preciso controlar o histórico de criações, deleções e alterações de arquivos.

Decisão

- Utilizar o git como sistema de controle de versão.

Justificativa

- Sistema open source mais utilizado no mundo.
- Permite mais flexibilidade no fluxo de trabalho, segurança e desempenho.

Consequências

- Suporte para desenvolvimento não-linear.
- Ajuda a lidar com projeto de forma eficiente.

Alternativas Consideradas

- SVN para controle de versão centralizado. Oferecendo abordagem linear para histórico de revisões. No entanto, vem sendo substituído pelo git no mercado, por ter uma natureza centralizada e a flexibilidade limitada em comparação com o Git.

Referências

- Git:
<https://blog.betrybe.com/git/#~:text=Git%20%C3%A9%20o%20sistema%20de%20trabalho%2C%20seguran%C3%A7a%20e%20desempenho>
- SVN:
<https://www.sorocaba.unesp.br/#!/programas-especiais/pef-eca/pesquisa/o-que-e-svn/>

Figura 23. ADRs criada a partir da versão 2 do fluxo de criação de ADRs

C. Processo de Criação e Revisão de ADRs no GitHub (Versão 3): Etapa de Contestação e *Merge Request*

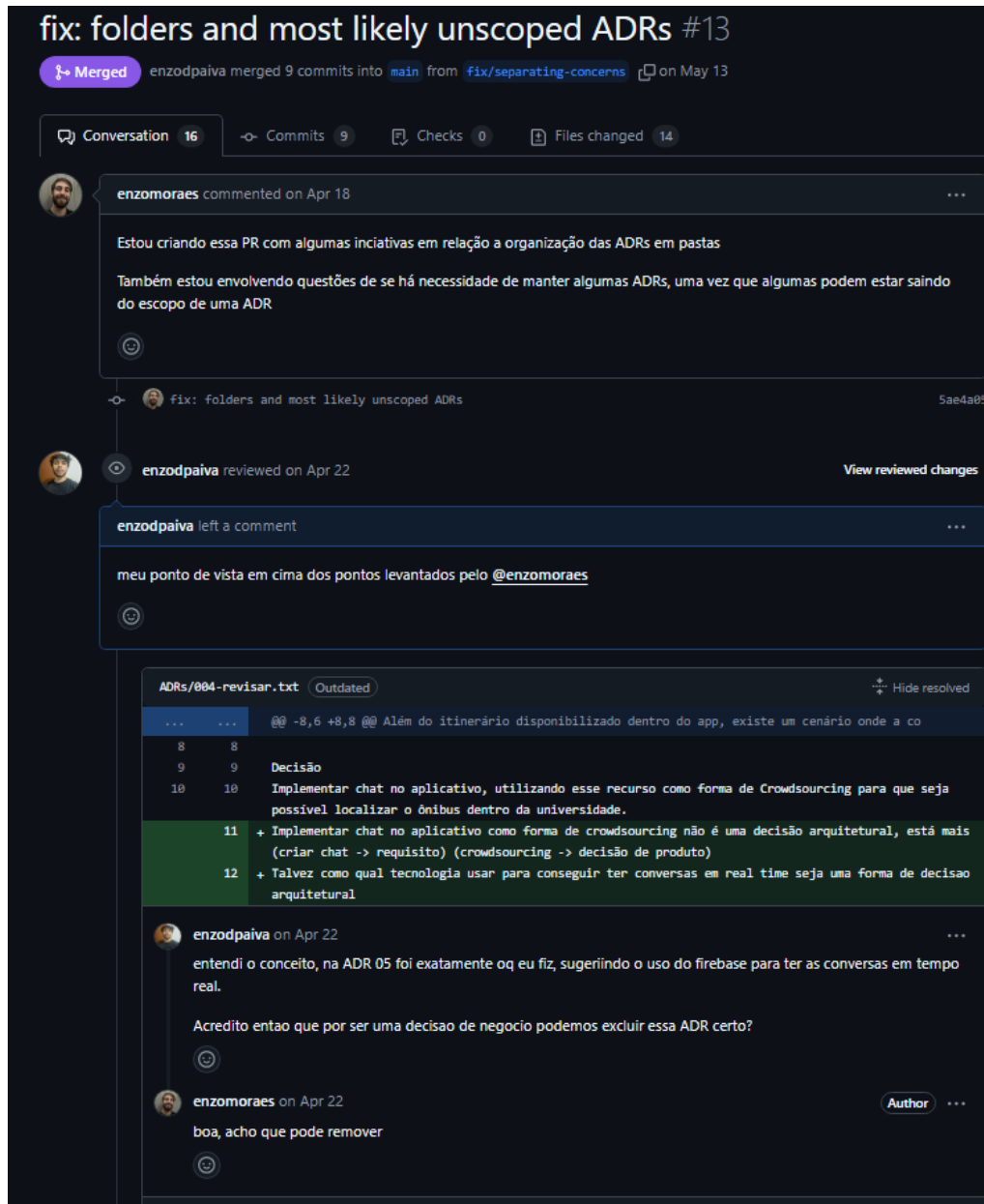


Figura 24. Processo de Criação e Revisão de ADRs no GitHub (Versão 3): Etapa de Contestação e *Merge Request*.

D. Discussão de Comentários Usando Template - Fluxo de Criação de ADRs

The screenshot shows a GitHub pull request interface. At the top, a purple banner indicates the pull request is 'Merged'. Below this, the pull request title is 'feat/ADR-017-utilizacao-arquitetura-microservicos-containerizado #24', with a note that 'enzodpaiva merged 1 commit into main from feat/ADR-017-utilizacao...' and a clock icon indicating it was merged '2 weeks ago'.

The main discussion area shows a comment from 'enzodpaiva' from 'last month'. The comment text is: 'Para o desenvolvimento do nosso sistema, que inclui um aplicativo móvel (ADR-001), uma API (ADR-016), uma aplicação web (ADR-015) e o banco de dados (ADR-005), precisamos de uma arquitetura que permita a integração eficiente de todos esses componentes.' Below the text is a smiley face emoji icon.

Below the comment, a commit link is shown: 'Implantação de uma arquitetura de microservicos com docker' by 'a22283f'.

Below the commit link, a note indicates that 'enzodpaiva changed the title Implantação de uma arquitetura de microservicos com docker feat/ADR-017-utilizacao-arquitetura-microservicos-containerizado last month'.

Below the title change, a note indicates that 'enzodpaiva requested review from paulosreis, enzomoraes, Awdren, joseaugustovital, mateusamado and RobertoFORTs last month'.

Below the review request, a comment from 'mateusamado' from '3 weeks ago' is shown. The comment has a title 'Opinião sobre a Decisão' and the text: 'Eu concordo com a decisão. Manter todos esses componentes em um monolito poderia transformar o desenvolvimento em um caos.' Below the text is a smiley face emoji icon.

Below the first comment, a section titled 'Questões Levantadas' is shown. The text is: 'A equipe tem experiência com Docker? Caso não tenham, haverá um tempo dedicado para o estudo dessa ferramenta?' Below the text is a smiley face emoji icon.

Below the second comment, a comment from 'RobertoFORTs' from '3 weeks ago' is shown. The comment text is: 'Eu entendo a ideia dos microseviços. Mas considero que deveríamos levar em conta também o quão esse sistema é escalável e se não estaríamos utilizando um "canhão para matar uma formiga". Como alternativa, poderíamos, ainda assim, trazer uma arquitetura mais baseada em serviços pontuais que se comunicam entre si via uma API que poderia ser monolítica modular ou monolítica, dependendo de variáveis como tempo de desenvolvimento, manutenibilidade e escalabilidade.' Below the text is a smiley face emoji icon and a speech bubble icon with the number '1'.

Figura 25. Discussão de Comentários Usando *Template*

E. Debates de ADRs no GitHub

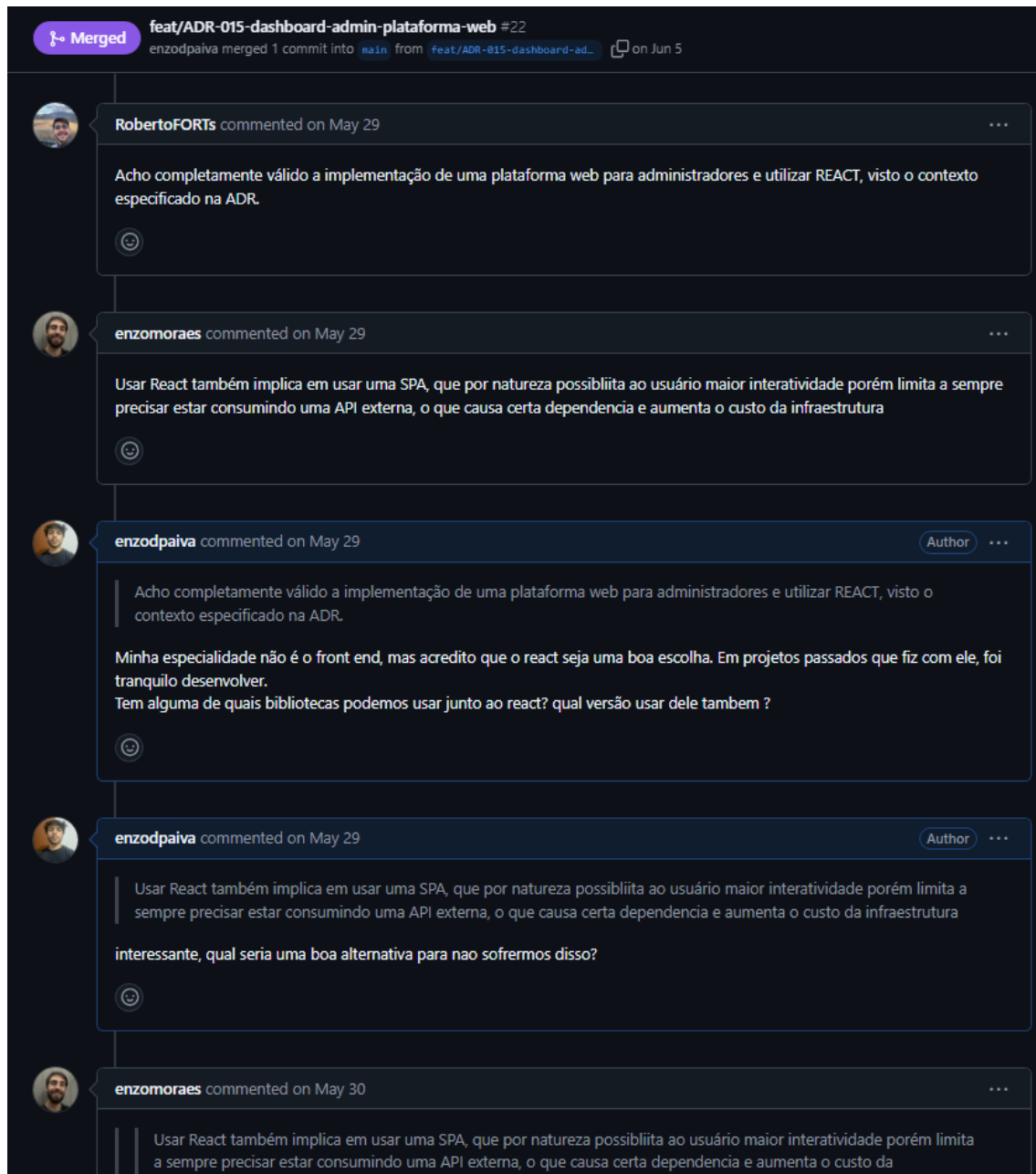


Figura 26. Exemplo de debates de ADRs realizados no GitHub.

F. Contraponto que gerou votação sobre ADRs

The screenshot shows a GitHub pull request titled "feat/ADR-017-utilizacao-arquitetura-microservicos-containerizado #24". The pull request was merged by "enzodpaiva" 2 weeks ago. The discussion includes the following comments:

- enzodpaiva** commented last month: "Para o desenvolvimento do nosso sistema, que inclui um aplicativo móvel (ADR-001), uma API (ADR-016), uma aplicação web (ADR-015) e o banco de dados (ADR-005), precisamos de uma arquitetura que permita a integração eficiente de todos esses componentes."
- A commit titled "Implantação de uma arquitetura de microservicos com docker" was pushed by "a22283f".
- enzodpaiva** changed the title to "Implantação de uma arquitetura de microservicos com docker feat/ADR-017-utilizacao-arquitetura-microservicos-containerizado" last month.
- enzodpaiva** requested review from **paulosreis**, **enzomoraes**, **Awdren**, **joseaugustovital**, **mateusamado** and **RobertoFORTs** last month.
- mateusamado** commented 3 weeks ago: "Opinião sobre a Decisão" and "Questões Levantadas".
- RobertoFORTs** commented 3 weeks ago: "Eu entendo a ideia dos microseviços. Mas considero que deveríamos levar em conta também o quão esse sistema é escalável e se não estaríamos utilizando um 'canhão para matar uma formiga'."

The comment by **RobertoFORTs** has 1 reaction (thumbs up).

Figura 27. Fluxo de decisão sobre ADRs com contraponto gerando necessidade de votação.

G. Votação de ADRs com soluções alternativas

Merged

feat/ADR-017-utilizacao-arquitetura-microservicos-containerizado #24
enzodpaiva merged 1 commit into main from feat/ADR-017-utilizacao-a... on Jun 5

- Possibilidade de crescimento do projeto podendo o integrar a novos serviços de maneira mais rapida

Acredito que a ideia do monolito modular pode ser util na aplicação web, porém aplicar o conceito no projeto todo, tende somente a dificultar sua entrega final.



enzodpaiva commented on Jun 4

Author ...

Após a semana de discussões nesta ADR, chegamos a um impasse sobre qual arquitetura adotar na comunicação entre os serviços, perante isso com o intuito de decidirmos o que vamos usar, cada integrante do projeto deve por favor VOTAR em qual arquitetura gostaria de desenvolver para o projeto levando em consideração os seguintes aspectos:

Escalabilidade: Qual arquitetura você acredita que oferece melhor suporte para escalabilidade do nosso projeto?
Desempenho: Qual arquitetura você acredita que oferece melhor desempenho para garantir uma experiência rápida e responsiva?
Segurança: Qual arquitetura você acredita que oferece melhores práticas de segurança para proteger os dados do nosso projeto?
Manutenibilidade: Qual arquitetura você acredita que é mais fácil de manter e adicionar novas funcionalidades no futuro?
Custo: Qual arquitetura você acredita que oferece um custo mais adequado para o desenvolvimento e manutenção do projeto?
Recursos e Expertise: Qual arquitetura você acredita que tem mais recursos disponíveis e que vc tem mais expertise para desenvolver o projeto?
Integração: Qual arquitetura você acredita que se integra melhor com nossos sistemas existentes?
Confiabilidade: Qual arquitetura você acredita que tem um histórico mais confiável de uptime e tolerância a falhas?
Comunidade: Qual arquitetura você acredita que tem uma comunidade mais ativa e suporte online?
Ciclo de Vida: Qual arquitetura você acredita que tem um ciclo de vida sustentável e evolutivo?
Conformidade: Qual arquitetura você acredita que está em conformidade com os padrões e regulamentações necessários?
Reutilização: Qual arquitetura você acredita que suporta melhor a reutilização de padrões e componentes?
Experiência do Usuário: Qual arquitetura você acredita que proporcionará a melhor experiência de usuário para nosso projeto?

VOTEM COM OS EMOJIS ABAIXO:

❤️ - Micro-serviços
🔧 - Monolito Modular

 5

Figura 28. Processo de votação para definir a solução escolhida em uma ADRs com alternativas apresentadas.