



Migrações de Legados Governamentais: Análise do Processo e Criação de um Modelo de Avaliação para a Utilização de *Microservices* no Ciclo de Vida do *Software**

Government Legacy Migrations: Process Analysis and Creation of an Evaluation Model for the Use of Microservices in the Software Lifecycle

Isabela Carolina Ribeiro Santos¹
Fabio Leandro Rodrigues Cordeiro²

Resumo

Em virtude do contexto organizacional, a maioria das migrações de legados no setor público ocorrem quando a tecnologia perde suporte. Isso dificulta as evoluções nos aspectos do *software* e leva à criação de sistemas pouco escaláveis, que não atendem as necessidades do usuário. Isto posto, o presente trabalho propõe a arquitetura de *microservices* para as atualizações de sistemas governamentais. Seu objetivo é criar um modelo de avaliação que forneça parâmetros para a adesão da arquitetura nos processos de migrações dessas aplicações. Para tal, foi realizado um estudo de caso que investigava a relação do setor com os cenários gerados pelas conjunturas institucionais, criado o modelo e realizado uma pesquisa de opinião para validação dele. Os resultados do estudo mostraram que o modelo, mesmo não contemplando alguns aspectos relevantes para a execução de um projeto de *software*, pode contribuir para a adesão da arquitetura e para a concessão de sistemas melhores.

Palavras-chave: Microservices. Sistemas legados. Migração de sistemas. Sistemas governamentais.

*Submetido em 18/07/2022 - Aceito em 13/09/2023

¹Graduanda em Sistemas de Informação pela PUC Minas - Barreiro, Brasil– isabelacarolina58@gmail.com

²Professor do Instituto de Ciências Exatas e de Informática da PUC Minas, Brasil– fabio@pucminas.br

Abstract

Due to the organizational context, most legacy migrations in the public sector occur when technology loses support. This makes it difficult to evolve aspects of the software and leads to the creation of systems that are not very scalable, which do not meet the user's needs. The present work proposes the microservices architecture for government system updates. The objective is to create an evaluation model that provides parameters for the adherence of the architecture in the migration processes of these applications. To this end, a case study was carried out that investigated the sector's relationship with the scenarios generated by institutional conjunctures, created the model, and carried out an opinion survey to validate it. The results of the study showed that the model, even not contemplating some relevant aspects for the execution of a software project, can contribute to the adhesion of the architecture and the concession of better systems.

Keywords: Microservices. Legacy systems. Systems migration. Government systems.

1 INTRODUÇÃO

Todo sistema sofre alteração independente do contexto que se insere. Pois, conforme Cuadrado *et al.* (2008) descrevem, durante seu ciclo de vida, erros, novos requisitos ou tecnologias mais eficientes aparecem em diferentes estágios de sua construção. Essas mudanças impactam em muitos aspectos do projeto e são importantes para garantir que ele atenda de maneira satisfatória as necessidades do usuário. Mas, mesmo sendo inevitáveis, muitas organizações resistem a essas transições e utilizam os chamados sistemas legados como a espinha dorsal do negócio.

Segundo Strobl, Bernhart e Grechenig (2020), legados são ferramentas funcionais ainda ativas na empresa, desenvolvidas com tecnologias desatualizadas, e utilizados por possuírem lógicas e dados importantes para os negócios. Eles são caros de manter, difíceis de modificar ou evoluir e em muitos casos possuem mão de obra escassa. Além disso, esses sistemas carregam inúmeras linhas de código e normalmente, mesmo que de maneira insatisfatória, atendem as demandas da companhia. Por isso, apesar dos problemas trazidos por eles e as evidências dos benefícios da atualização, cerca de 180-200 bilhões de linhas de código legado ainda estão em uso (KHADKA *et al.*, 2014).

Muitos são os motivos que atrasam essa modernização, mas dentre os principais está a segurança trazida pelos legados. Essas ferramentas rodam há anos e possuem um papel fundamental para o negócio. Sendo assim, optar por migrá-las não é uma tarefa fácil, pois diferentemente da construção de um novo sistema, a atualização de um legado já se inicia com o sistema sendo crucial para a entidade.

Nas organizações públicas essa resistência é ainda maior. Pois além dos receios que permeiam a empresa, há fatores ligados a administração e oposição da equipe de TI para a realização desse processo. Sendo assim, será apresentado como essa apreensão afeta as entidades públicas e como e por que este estudo pretende contribuir para a resolução do problema.

Mesmo com as evoluções da tecnologia, há uma forte utilização de legados no setor público. Segundo Alexandrova, Rapanotti e Horrocks (2015), o gerenciamento desses sistemas já é uma parte integrante da prestação de serviço público. Isso ocorre principalmente porque fatores como pouca rotatividade, nenhuma concorrência, políticas, burocracias, entre outras, formam nessas organizações uma tendência à estagnação.

Essa predisposição, dificulta a modernização dos sistemas da empresa. Para os usuários, a comodidade do conjunto de *software* existente, juntamente com experiências de migrações ruins, cria a ideia de que a atualização pode resultar em sistemas modernos e deficientes, que não atendem suas necessidades ou são difíceis de manuseio. Já para a equipe de TI, evoluir o legado pode forçar muitos profissionais a saírem de suas zonas de conforto, ou gerar medos ligados à substituição da mão de obra.

Como consequência, a migração dos legados é adiada até que a tecnologia utilizada seja descontinuada ou deixe de dar suporte às demandas grandes do setor. Essa atualização forçada, normalmente é realizada às pressas e não considera a possibilidade de evoluções nos fluxos ou

características do *software*. Conforme Alexandrova, Rapanotti e Horrocks (2015) mostram em seus estudos, muitas vezes é realizada apenas a atualização da tecnologia e desconsiderado a oportunidade de realizar mudanças que impactem positivamente no futuro do sistema.

Nesse contexto, muitas dessas organizações estão criando sistemas críticos com potencial de se tornarem legados problemáticos. Pois, as novas aplicações além de reproduzirem os erros da antiga, em alguns casos não são desenvolvidas com a tecnologia adequada, são pouco escaláveis e de difícil manutenção e evolução.

Com o intuito de contribuir para evoluções no processo de migração de legados do setor público, este artigo propõe a arquitetura de *microservices* como alternativa para a atualização de sistemas governamentais. A arquitetura, pode ser benéfica em dois cenários comuns dessas organizações.

A primeira situação é a migração por obsolescência. Nela, os profissionais precisam migrar sistemas grandes em prazos curtos, pois os fluxos da empresa podem parar devido a limitações da tecnologia. Para esse contexto, *microservices* permitem migrações mais lentas, uma vez que é possível atualizar as funcionalidades de maneira independente e colocá-las para conversar com o legado até que todo o sistema seja substituído.

A segunda circunstância é a migração por transformação digital. Ela ocorre quando o legado atende às necessidades da empresa, mas é preciso migrá-lo para que alguns fluxos suportem demandas que a tecnologia legada não apoia. Em tais casos, a arquitetura pode ser usada para evitar a migração de todo o sistema, permitindo a criação de unidades que atendem o solicitado.

Por conseguinte, tanto a utilização integral quanto parcial de *microservices* pode minimizar os impactos negativos das conjunturas apresentadas, gerar benefícios ligados às características do *software* e facilitar sua atualização.

O objetivo geral deste estudo é apresentar um modelo de avaliação para a utilização de *microservices* em sistemas governamentais. A proposta pretende ajudar os profissionais de TI do setor público a decidirem quando optar pela arquitetura, dado o contexto da organização.

- Investigar a relação das empresas responsáveis pelos sistemas públicos com a arquitetura de *microservices* e os cenários de migração por obsolescência e por transformação digital.
- Identificar e analisar os principais obstáculos para utilização de *microservices* nos processos de migração dos legados governamentais.
- Validar a eficiência e aplicabilidade do modelo criado com profissionais que lidam com os cenários apresentados.

Com o advento do governo digital faz-se necessária a reflexão acerca da eficiência das migrações de legados no setor público.

As aplicações governamentais caminham para dar suporte a procedimentos complexos da administração pública e possibilitar integração entre setores. Porém, muitos órgãos ainda precisam lidar com sistemas ineficientes criados a partir de fluxos inapropriados de atualização

que, em sua maioria, modificam apenas a tecnologia e não consideram as possibilidades de melhoria nos aspectos do *software* ou do negócio.

Isso posto, como tentativa de melhorar a metodologia de modernização e o *software* resultante, este trabalho apresenta um modelo de avaliação para apoiar profissionais de TI nas migrações. O modelo propõe a utilização de *microservices* e estrutura as variáveis importantes ao processo. Com ele pretende-se mostrar que em determinados contextos do *upgrade*, a arquitetura pode contribuir para aumentar a qualidade dos sistemas.

2 REFERENCIAL TEÓRICO

Esta seção tem o objetivo de propor uma revisão literária dos principais conceitos que permeiam o tema deste trabalho. São descritos os conceitos de migração de sistemas (seção 2.1), arquiteturas de *software* com foco em monolíticos e *microservices* (seção 2.2) e sistemas governamentais (2.3), além de expor os trabalhos relacionados ao estudo (2.4).

2.1 Migração de Sistemas

Ao longo de seu ciclo de vida, um sistema se depara com três fatores que conduzem sua modificação. O primeiro e mais comum é a correção de erros. Ela mantém a lógica do negócio e altera o código para sanar problemas. O segundo é a evolução dos requisitos. Nesses casos, a lógica do negócio pode ser alterada ou acrescida. O terceiro é a substituição. Ela ocorre quando a ferramenta atual já não atende às necessidades do negócio e conduz a criação de um novo *software*.

No caso dos legados, tanto a evolução quanto a substituição podem levar a migração do sistema. Esses procedimentos, conhecidos respectivamente como migração por modernização e substituição, ocorrem para melhorar a aplicação por meio de atualização da tecnologia ou redesenho do sistema atual ou parte dele (ALTHANI; KHADDAJ, 2017b).

Na substituição, um novo sistema é criado, mantendo os fluxos da ferramenta antiga e modificando sua tecnologia para outra mais recente.

Na modernização há grupos diretos e indiretos. Os diretos, são abordagens caixa preta e geralmente estão associadas ao encapsulamento de funcionalidade legada (SALVATIERRA; MATEOS; ET.AL, 2013). Um exemplo seria o *screen scraping*, que propõe a criação de uma nova interface (UI) que conversa com o sistema legado. Já os indiretos são metodologias caixa branca e incluem a reengenharia das rotinas legadas (SALVATIERRA; MATEOS; ET.AL, 2013). Uma amostra dessa técnica é o redesenho de funcionalidades de sistemas *desktop* para a WEB.

Logo, a migração de sistema é muito importante no contexto dos legados. Ela permitirá que essas ferramentas sejam descontinuadas, dando lugar a novas mais eficientes e capazes de

atender de maneira satisfatória às necessidades de seus usuários.

2.2 Arquitetura de *Software*

A arquitetura de *software* é o conjunto de normas, princípios e técnicas para a construção e evolução de um sistema. Ela é responsável por decisões sobre as estruturas da aplicação, protocolos de comunicação, sincronização e acesso a dados, atribuição de funcionalidade a elementos do sistema etc. (SHAW; GARLAN, 1996).

Suas soluções são chamadas de padrões arquiteturais de *software* e ditam as melhores decisões para um determinado contexto. Dentre os estilos arquiteturais frequentemente usados estão *Model-view-controller* (MVC), *Client-server*, *Layers*, *Service-Oriented Architecture* (SOA) e *microservices*.

Isso posto, os tópicos, a seguir, tratam sobre os padrões arquiteturais que permeiam o tema deste trabalho. Primeiro, foram contextualizadas as arquiteturas monolíticas (tópico 2.2.1). Em seguida, apresentado o conceito de *microservices* com suas vantagens e desvantagens (tópico 2.2.2). E por fim, foi realizado um comparativo dos dois tipos de estrutura com o intuito de expor suas principais diferenças (tópico 2.2.3).

2.2.1 Monolíticos

Arquiteturas monolíticas são soluções para a construção de um sistema, que mantém amarrados todos os componentes e instâncias de um *software*. Nelas, todas as funcionalidades são implantadas juntas e podem compartilhar recursos de servidor, banco de dados, *logs* etc.

Newman (2019) apresenta três tipos de monolíticos. O primeiro é o de processo único. Neles todo código é compactado e implantado em um único processo. O segundo é o distribuído. Ele possui suas funcionalidades separadas em serviços, mas implanta todas elas juntas. E o terceiro é o de caixa preta, que é um *software* comprado de terceiros, implantado e gerido pela organização.

Essas estruturas são vantajosas porque são simples de desenvolver, testar e implantar. Uma vez que todos os componentes estão juntos, não é necessário criar vários processos para a gestão do ciclo de vida do sistema. Além disso, muitos autores afirmam que a reutilização de código, em alguns casos, é mais simples que em outras estruturas. Basta chamar diretamente a funcionalidade, sem a necessidade de requisições à outras aplicações.

Em contrapartida, os monolíticos tendem a se tornar sistemas muito grandes, difíceis de escalar, gerir e evoluir. Seu tamanho exige recursos de infraestrutura mais robustos, equipes maiores de desenvolvimento e mais esforços para a realização de alterações. Além disso, esse tipo de paradigma possui alto acoplamento, uma vez que erros em uma funcionalidade pode afetar outras, e dependência de tecnologias, dado que todos os fluxos do sistema precisam ser

desenvolvidos com os mesmos recursos.

Sendo assim, optar por uma arquitetura monolítica nem sempre é a melhor escolha. Isso porque, embora simplifique os procedimentos de criação e implantação de um *software*, a longo prazo, em alguns contextos, essa estrutura tende a ser de difícil gestão, manutenção, evolução e de baixa escalabilidade.

2.2.2 *Microservices*

A arquitetura de *microservices* é uma abordagem de desenvolvimento que permite construir um *software* com um conjunto de unidades separadas. Esses serviços são independentemente implantados (LUZ *et al.*, 2018), reutilizáveis e responsáveis por apenas uma função da aplicação.

Diferentemente dos demais modelos arquitetônicos baseados em serviços, *microservices* possuem unidades menores que conversam por mensagens. Essa comunicação pode ocorrer utilizando o protocolo HTTP com APIs REST, por centros de distribuição de mensagens como o Apache Kafka, modelos de mensagens como GRPC e Json, entre outros.

Embora seja uma metodologia relativamente nova, os *microservices* já possuem muitos benefícios comprovados, quais sejam, escalabilidade, tolerância a falhas, independência de tecnologia no projeto, equipes de desenvolvimento menores, simplificação da criação de funcionalidades, facilidade de integração do DevOps, diminuição dos esforços para manutenção e evolução etc.

Em contrapartida, essa arquitetura dispõe desvantagens como: problemas de rede podem atrasar ou impedir a comunicação dos serviços e sua construção pode ser até 20% mais trabalhosa que a dos monolíticos (TAIBI; LENARDUZZI; PAHL, 2017). Além disso, o processo de migração para essa arquitetura exige maturidade da equipe de desenvolvimento, uma vez que a quebra de um sistema monolítico não é uma tarefa fácil.

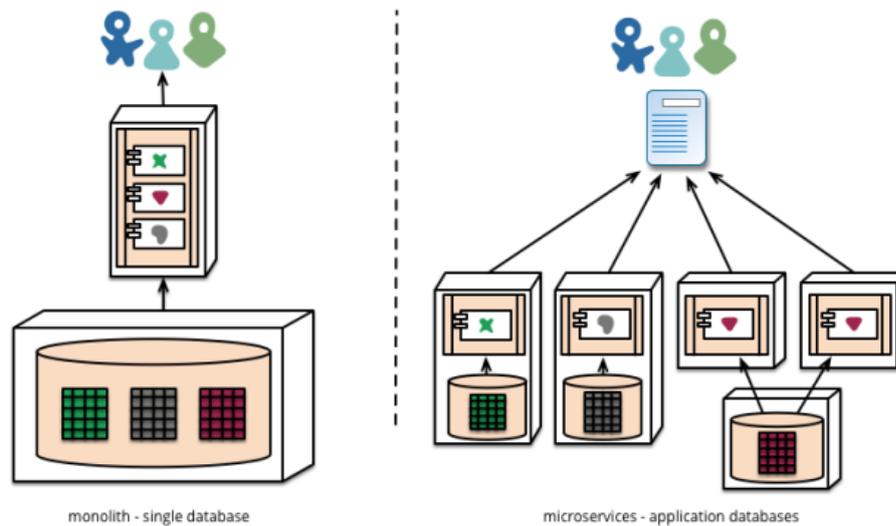
Portanto, *microservices* é uma abordagem que se mostra vantajosa para sistemas grandes e complexos, que possuem alto custo de evolução ou manutenção. Optar por essa arquitetura, embora seja uma tarefa trabalhosa, pode a longo prazo ser benéfico para as organizações que lidam com esse tipo de *software*.

2.2.3 *Microservices vs Monolíticos*

O principal diferencial entre as duas arquiteturas é a estruturação das funcionalidades. Nos monolíticos todos os fluxos estão no mesmo processo compartilhando recursos. Já na abordagem de *microservices*, eles podem ser divididos em serviços independentes, que contêm seus próprios mecanismos de execução. A Figura 1 exemplifica essa distinção.

Para que uma estrutura seja de *microservices*, além independência das rotinas é neces-

Figura 1 – Arquitetura *microservices* e monolítica



Fonte: (LEWIS; FOWLER, 2014).

sário que a infraestrutura dos fluxos seja autônoma. Cada serviço precisa possuir seus próprios instrumentos de execução. Quando as unidades compartilham qualquer elemento, a aplicação passa a ser um monolítico distribuído.

2.3 Sistemas de Informação Governamentais

Sistemas de informação governamentais são aplicações usadas para apoiar os processos da administração pública. Eles sustentam os fluxos internos dessas organizações e, com o advento do governo digital, tornaram-se fundamentais para integrar vários setores.

Essas ferramentas possuem grandes fluxos de dados pessoais e sensíveis, e muitos de seus requisitos são definidos por projetos de leis (PL) ou proposta de emenda constitucional (PEC). Além disso, Abreu e Fernandes (2014) afirmam que muitos deles são criados para orientar programas de longa duração e, por isso, são utilizados durante anos, sem grandes modificações em suas rotinas.

Em suma, essas aplicações são desenvolvidas especialmente para demandas governamentais e por isso não possuem concorrência no mercado. Isso, alinhando com os receios de modernização que permeiam essas entidades, reduz a possibilidade de melhoria desses sistemas e gera uma gama de legados atendendo às necessidades do governo.

Leon e Horita (2020), em seus estudos, mostram que essas aplicações ainda possuem arquiteturas centralizadas e que nem sempre permitem integrações para atender a transformação digital. Além disso, muitos deles são mantidos em servidores da própria empresa, e ainda não utilizam *cloud computing*, o que dificulta ainda mais sua modernização.

Assim, pode-se dizer que sistemas de informação governamentais são ferramentas importantes ao setor público e possuem tendência à estagnação. Lidam com diferentes fluxos da

companhia e nem sempre são modificados pensando na satisfação do usuário ou na expansão digital. Torna-se evidente a relevância de encontrar meios que aperfeiçoam tanto os processos do ciclo de vida, quanto às características técnicas desse grupo de *software*.

2.4 Trabalhos Relacionados

Khadka *et al.* (2014) mostram a importância dos legados para as organizações. Seu estudo expõe a visão dos profissionais de TI, sobre esses sistemas e apresenta a migração de dados, a arquitetura do legado, a extração e priorização de requisitos, a resistência organizacional, etc como fatores que dificultam a modernização desses sistemas. Esse trabalho foi escolhido porque exhibe os principais desafios para a migração.

Análises de algumas das abordagens de migração citadas nos estudos anteriores, podem ser vistas na pesquisa de Althani e Khaddaj (2017a). Os autores trazem os aspectos de qualidade como variáveis para a seleção da topologia de modernização. O intuito disso é tentar garantir que o novo sistema atenda bem as demandas para as quais está sendo construído. Eles também fornecem uma divisão das técnicas existentes em quatro grupos de generalizações, sendo eles: migração total, parcial, incremental e de empacotamento. O estudo foi escolhido porque possui um foco na etapa de planejamento da migração e considera a qualidade como um fator importante a ser considerado no processo.

Os estudos de Fritsch *et al.* (2019) e Taibi, Lenarduzzi e Pahl (2017) fornecem a visão da arquitetura de *microservices* na indústria. Eles demonstram conceitos de *microservices* e a partir da opinião de profissionais experientes, identificam os benefícios e dificuldades encontrados durante a migração para essa arquitetura. Embora as pesquisas possuam metodologias diferentes, ambas mostram as motivações, estratégias e desafios enfrentados pelas empresas que optam pela estrutura.

Alexandrova, Rapanotti e Horrocks (2015) mostram como as migrações por substituição estão sendo realizadas em organizações governamentais e se as oportunidades de evoluir o negócio estão sendo aproveitadas nesse processo. Ele expõe que, em governos do mundo, os problemas com sistemas legados ainda são salientes e que o setor público nem sempre enxerga a ideia de mudança do sistema atual como algo benéfico. A pesquisa foi selecionada porque mostra como o setor público vê e executa mudanças em seus programas.

Já o trabalho de Luz *et al.* (2018) foca na migração de sistemas governamentais brasileiros para a arquitetura de *microservices*. Os autores mostram os desafios e ganhos da mudança. Também é exposto a burocracia e tendência à estagnação do setor, além dos impactos causados pela pouca rotação, nenhuma concorrência e muita resistência à inovação desses órgãos. O estudo foi escolhido porque fornece uma visão real da implantação de *microservices* no setor público. Ele leva em conta os fatores internos e o tipo de profissional dessas organizações.

3 METODOLOGIA

Como o debate acerca da migração de sistemas legados é algo que há muito tempo permeia a indústria e o meio acadêmico, foi realizada uma pesquisa aplicada, que propõe uma solução para os problemas enfrentados pelos processos de migrações.

O estudo foi dividido em três etapas cruciais. A primeira possui cunho exploratório e tem o intuito de realizar um estudo de caso numa empresa governamental (seção 3.1). Já a segunda, tem um caráter analítico, pois a partir das informações coletadas foi criado um modelo de avaliação que considera *microservice* como uma saída para o problema (seção 3.2). Por fim, o último estágio foi de natureza descritiva, e avaliou o padrão criado com os profissionais que lidam com a demanda (seção 3.3). As subseções a seguir descrevem detalhadamente os procedimentos de cada fase.

3.1 Realização do Estudo de Caso

Um estudo de caso “fornece uma maneira sistemática de observar eventos, coletar dados, analisar informações e relatar os resultados” (VERNER *et al.*, 2009). Sua realização permite conhecer as variáveis acerca de um determinado problema e, por isso, é fundamental para essa primeira etapa da pesquisa.

Dessa forma, foram realizadas entrevistas com profissionais do desenvolvimento de *software* da prefeitura de Belo Horizonte. O intuito dessas conversas era entender a relação desses colaboradores com os legados, a arquitetura de *microservices* e os processos de migrações.

Para permitir uma identificação clara a respeito de como os tipos de profissionais lidam com as questões apresentadas, foram elaborados roteiros específicos para gerentes e desenvolvedores. Com os gerentes buscou-se identificar a quantidade de sistemas que se encaixam nos cenários descritos, quais as estratégias adotadas e se *microservices* foram ou não uma opção durante as definições do projeto de migração. Já com os desenvolvedores, concentrou-se nos impactos das decisões tomadas pela gerência acerca das migrações e em suas opiniões sobre a adoção de *microservices* como arquitetura do projeto migrado. As respostas obtidas foram salvas em uma planilha e estão descritas nesta exploração.

3.2 Criação do Modelo de Avaliação

A partir das informações de como os profissionais lidam com os cenário de migração por obsolescência e por transformação digital coletadas no estudo de caso, criou-se um modelo de avaliação da utilização de *microservices* em empresas governamentais. Esse modelo tem o objetivo de mostrar as vantagens dos *microservices* nos contextos apresentados e as variáveis importantes para a estruturação das decisões acerca da utilização da arquitetura nos sistemas do

setor público.

A seleção de tais aspectos de avaliação foi realizada em três etapas. Primeiramente, identificou-se, em outros estudos, os principais desafios da implantação da arquitetura. Essas informações foram escolhidas sem considerar as especificidades dos sistemas públicos. Em seguida, esses dados foram analisados sob a ótica dos contextos de migração por obsolescência e por transformação digital e filtrados de acordo com sua relevância para os cenários. Nessa fase, foram determinados os fatores exclusivos para o setor público reconhecidos no estudo de caso e no referencial teórico. Por fim, todos os elementos foram descritos considerando o porquê da escolha e como devem ser.

3.3 Validação do Modelo Criado

A validação do modelo ocorreu por meio de uma pesquisa de opinião realizada com responsáveis pela criação e manutenção dos sistemas governamentais. Foi disseminado um *site*¹ que continha a contextualização do problema, descrições sobre o modelo, um *pitch* em vídeo e perguntas acerca da proposta apresentada. Tais questões foram divididas em três quesitos, sendo eles eficiência, aplicabilidade e possibilidade de aderência.

Os selecionados para a pesquisa eram gestores táticos, estratégicos, especialistas em segurança, gerentes de projeto e desenvolvedores. O intuito da distinção foi buscar opiniões dos diferentes tipos de profissionais que lidam com as conjunturas da migração por obsolescência e por transformação digital.

Para análise dos *feedbacks*, classificou-se os dados a partir do aspecto e tipo da pergunta. Em seguida, validou-se quantitativamente as questões de múltipla escolha e qualitativamente as indagações dissertativas. Por fim, foi estabelecida a relação entre as respostas e as informações deste estudo, de modo a certificar-se que os pontos negativos apresentados eram consistentes e estavam ligados com a experiência profissional do indivíduo e não com a tendência de estagnação do setor público.

Os resultados do exame foram descritos nesta exploração e permitiram, além de medir fatores sobre o protótipo, identificar trabalhos futuros para o aperfeiçoamento do mesmo.

4 ESTUDO DE CASO

Foi escolhida uma organização que desenvolve sistemas para a prefeitura de Belo Horizonte. Por questões de sigilo e com o intuito de preservar a imagem da companhia, neste trabalho ela será chamada de C1. A empresa localizada na capital mineira, já atua há anos no mercado e conta com colaboradores de diferentes idades e experiência de mercado. Ela foi escolhida porque sua gama de funcionários permite identificar como os diferentes perfis de pro-

¹Link <https://sites.google.com/view/modelo-avaliacao-microservices/home>

fissionais do setor lidam com os cenários apresentados nesta pesquisa. Os resultados obtidos estão descritos nas próximas subseções abaixo.

4.1 Entrevistas com os desenvolvedores

Para as conversas com os desenvolvedores, selecionou-se um grupo de cinco indivíduos responsáveis por diferentes domínios da organização, de modo que cada funcionário participante lidasse com um sistema específico. A experiência com criação de sistemas governamentais do grupo selecionado variava entre 2 e 12 anos, algumas pessoas já haviam participado de projetos de migração de legados e outras ainda lidavam com os problemas trazidos por esses sistemas. Os colaboradores com menos tempo de experiência, eram recém chegados ao mercado de TI e lidavam com servidores com anos de trabalho no serviço público.

As entrevistas foram realizadas via videochamada e levantavam os seguintes questionamentos:

- Q01: Você identifica algum sistema de sua empresa que, nos últimos três anos, passou ou passa pelos cenários de migração por obsolescência ou por transformação digital?
- Q02: Você julga as decisões da alta gerência adequadas para o contexto? (Considere aspectos de arquitetura, padrões de projeto, tecnologias, capacitação etc.)
- Q03: Você vê *microservices* como uma alternativa para esses cenários?

As respostas obtidas para a Q01 expuseram que, nos últimos anos, a companhia vem sofrendo, em diversos setores, os impactos das conjunturas da migração por obsolescência e por transformação digital. Todos os participantes afirmaram que, devido ao contexto organizacional, as iniciativas de atualização só ocorrem quando as tecnologias param de dar suporte aos fluxos do *software*. Nos exemplos apresentados, há sistemas em processos de migração por obsolescência devido a descontinuidade de seus elementos de execução, a evoluções não suportadas pelo legado e a problemas de baixa usabilidade da aplicação, além de ferramentas modernizadas para atender a transformação digital forçada pela pandemia em 2020.

Os retornos de Q02 mostraram que, em suma, as decisões das gerências sobre os aspectos do *software* são satisfatórias. Os colaboradores afirmam que, mesmo os recursos escolhidos não sendo as melhores opções, atendem às demandas as quais são destinados. As poucas queixas realizadas estavam ligadas a capacitação e aos testes dos novos sistemas. Para eles, falta tratar de maneira relevante a necessidade de treinamentos e os fluxos de criação e execução de testes.

Por fim, os *feedbacks* de Q03 explicitaram que os profissionais viam *microservices* como uma alternativa válida para os cenários de suas aplicações. Segundo eles, nas situações apresentadas, os sistemas possuíam fluxos independentes, criados com alto acoplamento devido a

estrutura monolítica utilizada. Porém, fatores como resistência de outros colaboradores, complexidade de implantação, escassez de mão de obra e receios quanto às consequências da inovação dificultam a adesão da arquitetura.

4.2 Entrevistas com os gestores

Para as entrevistas com os gestores, buscou-se dois indivíduos experientes que possuíssem ciência da maioria dos processos da empresa. Um dos dirigentes lidava com uma das maiores equipes da organização e o outro dando suporte a todas as equipes e processos de evolução dos projetos. Ambos tinham mais de 10 anos de experiência com tais processos e começaram na empresa como desenvolvedores. As perguntas realizadas foram:

- Q01: Você identifica algum sistema de sua empresa que, nos últimos três anos, passou ou passa pelos cenários de migração por obsolescência ou por transformação digital?
- Q02: Que estratégias são utilizadas para lidar com esses cenários?
- Q03: Você vê *microservices* como uma alternativa para esses cenários?
- Q04: Quais os parâmetros analisados para a escolha da arquitetura em suas migração?

Assim como na conversa com os desenvolvedores, Q01 mostrou a grande presença dos cenários de migração por obsolescência e por transformação digital na organização. Os gerentes afirmam que o problema não ocorre apenas nas migrações de sistemas, mas também nas evoluções de ferramentas de bancos de dados e mecanismos de segurança da infraestrutura. Para as modernizações do conjunto de *software*, eles alegaram que os processos apresentados estão ocorrendo com frequência nos últimos anos. Isso ocorre porque algumas aplicações foram desenvolvidas especificamente para as tecnologias de sua época, e diante da expansão digital nem todas estão suportando a crescente adesão dos usuários.

Com relação às estratégias utilizadas para lidar com as circunstâncias descritas, Q02 identificou que a instituição frequentemente opta por migrar sistemas criados com tecnologia descontinuada e sem suporte. Para as aplicações legadas que ainda possuem suporte no mercado, devido à baixa mão de obra, escolhe-se postergar a modernização até que a equipe arranje tempo hábil para fazer. Outro ponto apresentado foi que todas as decisões ligadas a escolha das ferramentas para a construção do novo sistema precisam estar de acordo com o catálogo de tecnologias aceitas pela companhia. Qualquer escolha diferente deve ser submetida à aprovação das diretorias necessárias e do setor de arquitetura.

Os retornos de Q03 explanaram que os gerentes conhecem parcialmente os benefícios da arquitetura de *microservices*. A maioria reconhece que ela pode contribuir para melhorar características dos sistemas e facilitar suas manutenções. Porém, há muitos receios quanto à complexidade de utilização, a adaptação dos recursos e ao tempo necessário para os esforços iniciais do projeto.

Por fim, nas respostas de Q04, foi exposto que os principais parâmetros avaliados para a decisão de qual arquitetura utilizar no novo sistema são os requisitos funcionais e não funcionais, o tempo para a migração e o conhecimento da equipe.

4.3 Resultados

Os levantamentos deste estudo de caso confirmaram que os cenários de migração por obsolescência e por transformação digital estão presentes em empresas responsáveis por sistemas governamentais. Algumas equipes, devido às circunstâncias organizacionais, só realizam a atualização da aplicação em última instância, quando os fluxos estão prestes a parar e não há outras opções para sanar o problema.

Conjuntamente, foi apontado que, embora o tipo de sistema seja avaliado durante as decisões das propriedades do *software*, os resultados de atualizações passadas pesam mais nas escolhas do que os requisitos da aplicação. Além disso, devido a burocracia da organização e o perfil de muitos colaboradores, as gerências, mesmo reconhecendo as oportunidades, optam por não evoluir particularidades do negócio ou da ferramenta.

Assim sendo, esta exploração evidenciou que muitas organizações que lidam com sistemas públicos, mesmo com os avanços tecnológicos impostos pela pandemia em 2020, ainda passam pelos desafios descritos nas pesquisas de Alexandrova, Rapanotti e Horrocks (2015), Luz *et al.* (2018) e Leon e Horita (2020).

Portanto, não basta apenas propor a arquitetura de *microservices* como benéfica para os cenários de migração apresentados, é preciso mostrar aos profissionais parâmetros para aderir-la. Assim, foram selecionados e avaliados os principais desafios na visão dos colaboradores de C1, para o emprego de *microservices*. Os resultados obtidos deram origem ao modelo de validação descrito na seção seguinte.

5 MODELO DE AVALIAÇÃO

5.1 Considerações iniciais

Foi criado um modelo que fornece parâmetros para a estruturação das decisões acerca da utilização de *microservices* em organizações governamentais. Seu objetivo é direcionar a análise dos fatores ligados aos principais obstáculos da aderência da arquitetura, possibilitando que a equipe responsável trace caminhos para lidar com eles.

Seu fluxo principal não contempla a fase de desenvolvimento e propõe que todas as definições necessárias sejam realizadas em um estágio anterior, considerando as funcionalidades do sistema. Isso porque, em projetos de *software* governamentais, dado o contexto de sobrecarga dos times, dificilmente um atributo inicialmente determinado é modificado. Pois, quando sua

deliberação é feita e aprovada, todas as rotinas são criadas e adaptadas para se encaixar a ele.

Nas subseções seguintes, foram expostos quais, como e quando determinados aspectos podem ser analisados e propostas as técnicas que melhor se adaptam às circunstâncias das migrações.

5.2 Definição dos fatores e fluxo principal

Observando o processo de migração de legados no setor público, identificou-se três variáveis determinantes para a viabilidade da atualização para *microservices*: o tempo, o custo e a infraestrutura.

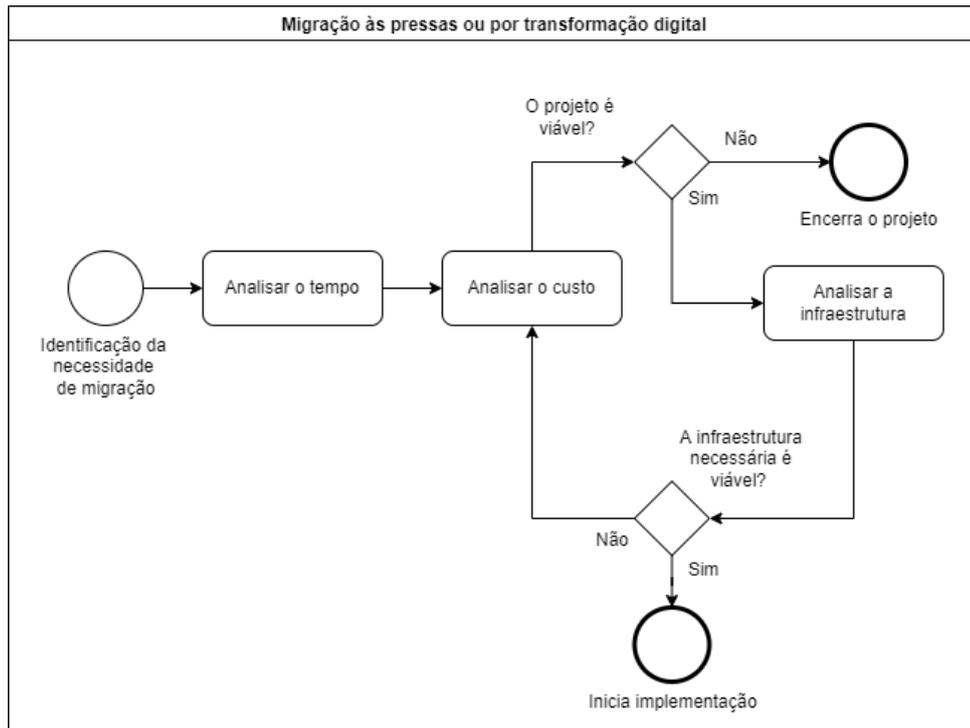
O tempo é o primeiro fator a surgir e rege todas as análises do projeto. Diante do tipo de requisito dos sistemas governamentais, ele é estático e pode começar a contar antes mesmo da identificação da necessidade do processo de migração. Sua definição normalmente é feita pelo cliente e o mesmo é utilizado para estudos de viabilidade, implementação e realização da mudança.

Os custos são todos os esforços, não ligados à infraestrutura, para a migração. Ele será avaliado considerando os investimentos necessários para realizar a alteração e irá determinar se a nova aplicação é viável. Em organizações públicas, as análises dessas variáveis podem englobar diferentes diretorias e gerências além da equipe de desenvolvimento.

A infraestrutura está ligada com o arranjo necessário para manter os serviços funcionando corretamente. Como visto anteriormente, *microservices* demandam uma sustentação robusta, pois cada serviço precisa ter seus próprios mecanismos de execução. Sendo assim, é importante avaliar se a organização suporta essa mudança sem gerar impactos negativos, uma vez que, nesse contexto, o novo sistema será concebido possuindo informações importantes ao cliente e à equipe de criação.

Diante da definição dos fatores, o fluxo principal de avaliação da arquitetura pode ser realizado conforme mostra a Figura 2.

Figura 2 – Fluxograma do modelo de avaliação



Fonte: Elaborado pelos autores.

5.3 Análise de tempo

Em propostas exclusivas de modernização é válido pular para a fase de análise de custo. Já em migrações providas de demandas de evoluções ou manutenções, embora não seja possível mudar a data de entrega do projeto, podem existir metodologias que permitirão dar ao usuário o que ele deseja, sem sobrecarregar a equipe de desenvolvimento.

Para isso, primeiro define-se o que o cliente realmente precisa. Vale lembrar que a identificação da necessidade de migração, não quer dizer que o indivíduo carece de todo o sistema em uma nova plataforma, no dia determinado. Assim sendo, é relevante discernir qual de fato é o foco do problema. Nessa etapa, é significativo manter o usuário por perto, pois quanto mais informações sobre o que originou a demanda e qual o objetivo final dela, mais fácil será a tomada de decisão seguinte.

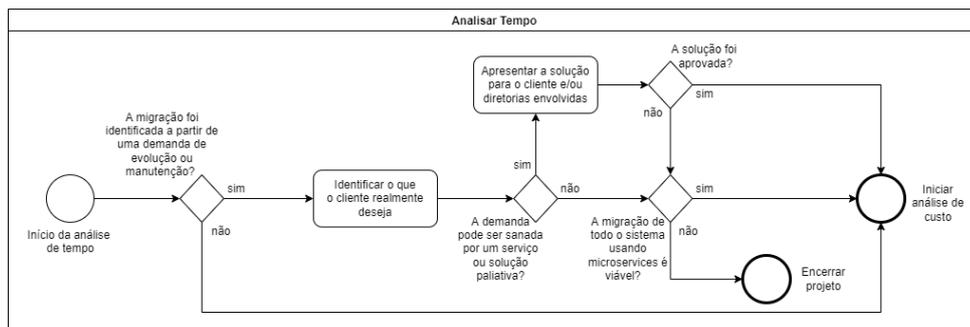
O próximo passo será ponderar se a demanda apresentada pode ser sanada por um serviço que converse com o legado ou por uma correção paliativa, que impeça a interrupção dos fluxos, enquanto a migração ocorre. Caso alguma dessas opções seja possível, pode-se prosseguir para a etapa seguinte. Do contrário, é preciso medir se o tamanho do sistema e o conhecimento e a adaptação dos recursos favorecem a utilização da arquitetura. Se constatar um parecer positivo, é plausível avançar para a análise de custo, do oposto o balanço se encerra e avalia-se outras opções de arquitetura.

Por fim, se determinada a criação de uma unidade ou uma saída mitigadora, é crucial

apresentar o projeto para aprovação do cliente e/ou diretorias envolvidas. Apesar das soluções aderidas atenderem às exigências dos mesmos, elas fogem do escopo inicial e, por isso, é fundamental o consentimento de todos os afetados pelo planejamento. Uma vez que o parecer for positivo, pode-se seguir com a análise de custo, de outro modo faz-se o exame da dimensão do sistema e a adaptação dos recursos.

A Figura 3 explicita em forma de fluxograma a tarefa de analisar tempo.

Figura 3 – Fluxograma da análise de tempo



Fonte: Elaborado pelos autores.

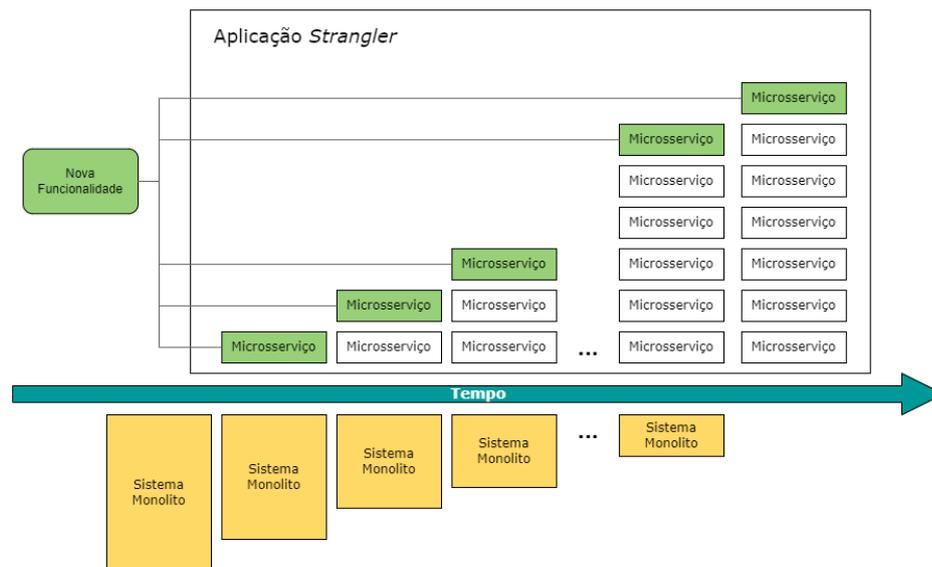
5.4 Análise de custo

A análise de custo se inicia com o reconhecimento dos processos de negócio. A gerência do projeto junto ao cliente apontará quais fluxos estarão presentes no novo *software*, direcionando a equipe ao que precisa ser feito. Nessa etapa é indispensável avaliar as possibilidades de melhorias nas rotinas e como corrigir ou remover as soluções paliativas criadas no sistema antigo. As funcionalidades legadas serão examinadas e um valor de peso atribuído para cada uma. Tal numeração será distribuída considerando a prioridade da função e representará a ordem de criação dos serviços.

Feito isso, torna-se viável examinar os subfatores que permeiam os custos. As validações são realizadas pela equipe de desenvolvimento e a gerência do projeto e podem ser feitas em paralelo, desde que haja boa comunicação entre os envolvidos.

Na definição das técnicas, avalia-se padrões de projeto, técnicas de quebra e comunicação de serviço e estratégias de teste para a nova aplicação. Durante as análises, é importante considerar a fragmentação da aplicação e do banco de dados e o número de recursos da equipe. Recomenda-se a utilização de padrões de fragmentação das rotinas similares ao *Strangler*. Essa abordagem propõe a criação de serviços que conversem com o legado, permitindo que, a partir da lista de prioridades anteriormente definida, seja possível criar unidades que atendam à demanda do cliente, até que todo sistema seja atualizado. Vale ressaltar que, nos cenários de migração por transformação digital, apenas as funcionalidades necessárias serão modificadas. A Figura 4 é uma tradução do diagrama de Richardson (2018) e exemplifica como a metodologia funciona.

Figura 4 – Abordagem Strangler



Fonte: Adaptado de (RICHARDSON, 2018).

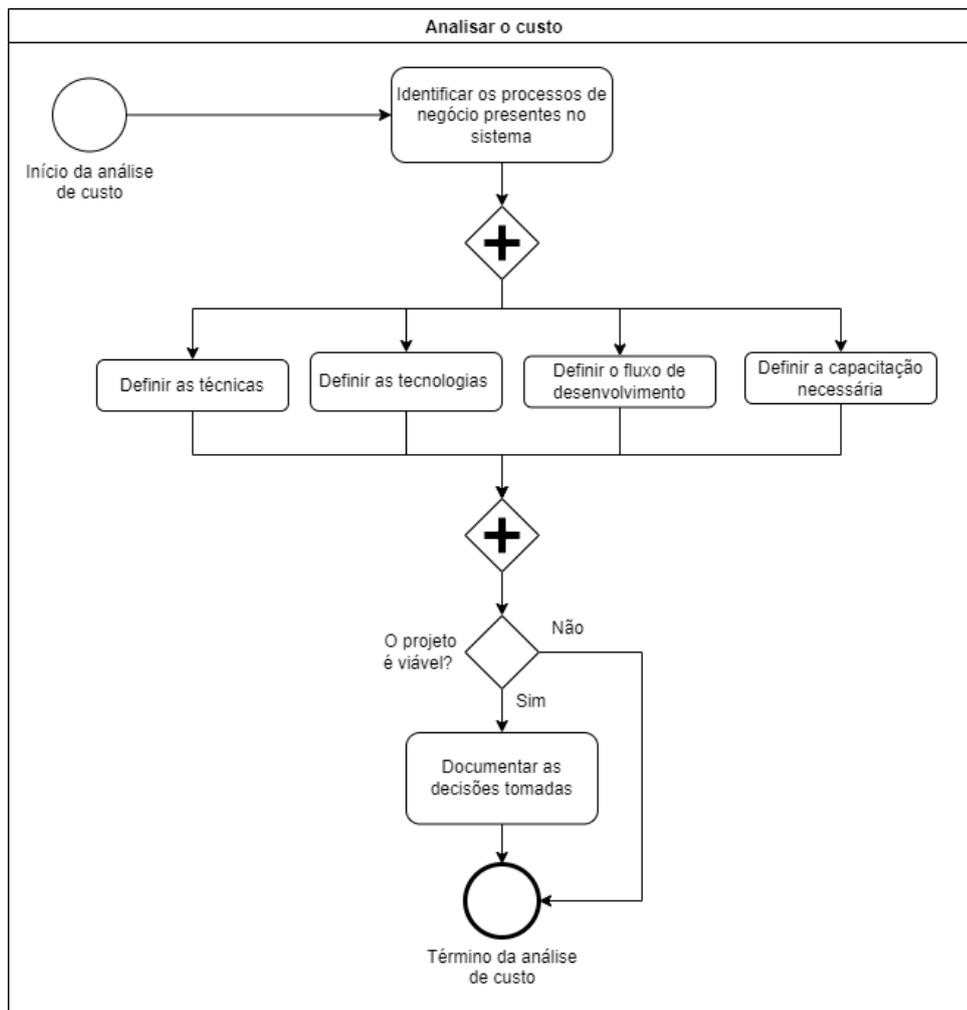
Para a deliberação das tecnologias, escolhem-se linguagens de programação, servidores de aplicação e, quando necessário, propriedades da base de dados. Essa tomada de decisão considera as características e a possibilidade de expansão do novo sistema. Por isso, é aconselhado buscar opções que tenham suporte dos fornecedores ou da comunidade, que não dificultam a manutenção e que atendam a todos os tipos de processamentos e rotinas presentes no *software*. É imprescindível que os envolvidos não se prendam apenas a experiências de migrações passadas, avaliem as alternativas do portfólio de tecnologias da organização e, quando necessário, façam pesquisas, entrevistas ou *brainstorm* para alcançar o objetivo.

A descrição do fluxo de desenvolvimento é realizada para definir como os processos de especificação, criação, testes e implantação ocorrerão. Ela precisa considerar a relevância da documentação e as mudanças necessárias na política e no modelo de trabalho. Sua função será garantir que os recursos não repliquem os paradigmas do trabalho legado, pois muitos colaboradores, após anos de experiência, tendem a reproduzir no novo *software* remendos e processos do anterior. Nessa fase, também é interessante a definição de uma metodologia para a gestão do conhecimento, de modo a permitir que todas as lições aprendidas fiquem disponíveis para migrações futuras da organização.

Para a definição da capacitação necessária é levantado o que os funcionários precisarão aprender para a realização do projeto. Ela considerará os receios, o conhecimento e adaptação da equipe e resultará no conjunto de cursos que cada colaborador deverá fazer. Caso a organização não possua, recomenda-se a criação de um sistema de gestão dos cursos, de modo que eles fiquem disponíveis aos colaboradores. Após a execução das quatro tarefas, será julgado se o projeto é viável. Para isso, leva-se em conta a possibilidade de fazer, em tempo hábil, tudo o que foi definido previamente. Se porventura não for, encerra-se a análise de custo. Do contrário, documenta-se as decisões tomadas e termina a tarefa. O fluxograma da atividade de analisar o

custo pode ser visto na Figura 5.

Figura 5 – Fluxo de análise de custo



Fonte: Elaborado pelos autores.

5.5 Análise de infraestrutura

Uma vez deliberados os custos, a documentação resultante da definição de tecnologias precisa ser repassada para a equipe de infraestrutura. Assim sendo, serão identificadas todas as necessidades levantadas pelo time de desenvolvimento e avaliado se os recursos solicitados já existem ou possuem um projeto de construção criado. Caso haja, a tarefa se encerra, do oposto analisa-se os subfatores da infraestrutura. Como nos custos, esses sub-aspectos podem ser examinados paralelamente desde que haja uma boa comunicação entre a equipe.

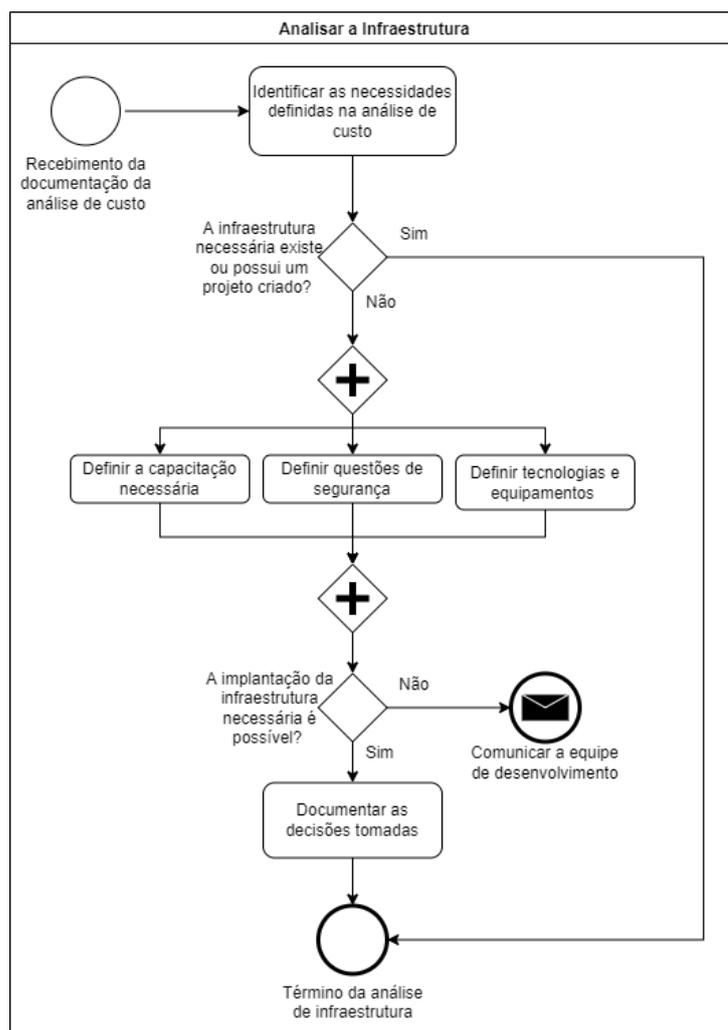
A definição de tecnologias e equipamentos aponta tudo o que é essencial para manter os serviços funcionando. Ela considera a possibilidade de expansão do *software*, as tecnologias, as técnicas escolhidas para a quebra de banco de dados e as comunicações entre os serviços. Busca-se a partir do portfólio disponível ou experiências passadas, os recursos que mais se adequam às unidades. Nessa fase, recomenda-se considerar a utilização de *cloud computing*

ou pensar em maneiras de automatizar processos. Além disso, investir em formas para gerir o conhecimento e documentar a criação dos ambientes pode facilitar a replicação da primeira estrutura criada para as demais semelhantes.

Para a caracterização das questões de segurança, determina-se regulamentos, mecanismos de permissionamento e protocolos relevantes para a proteção do *software*. Nela é importante considerar os níveis de acesso de todos os profissionais do projeto e como resguardar em nível físico e lógico a comunicação entre os serviços. Aconselha-se estabelecer os aspectos de defesa a partir das normas da família ISO/IEC 27000.

A definição da capacitação necessária ocorrerá similar à atividade presente na análise de custo. Serão levantados, a partir dos receios, conhecimento e adaptação dos recursos, os cursos necessários para a criação e manutenção do projeto. Após a realização das três atividades, será classificado se a estrutura necessária é viável. Na condição de não ser, comunica-se à equipe de desenvolvimento os resultados do exame, para que seja realizada uma nova análise de custo. Do contrário, documentam-se todas as decisões e encerra-se a tarefa. O fluxograma da Figura 6 expõe a tarefa de análise de infraestrutura.

Figura 6 – Fluxo de análise de infraestrutura



Fonte: Elaborado pelos autores.

6 VALIDAÇÃO DO MODELO

Durante uma semana, o *site* do modelo foi disseminado para funcionários de três empresas que lidam com a construção e manutenção de sistemas governamentais. Com o auxílio do *pitch* e de descrições sobre a proposta, treze participantes opinaram a respeito da eficiência, aplicabilidade e possibilidade de aderência do modelo. Os resultados da avaliação estão descritos a seguir.

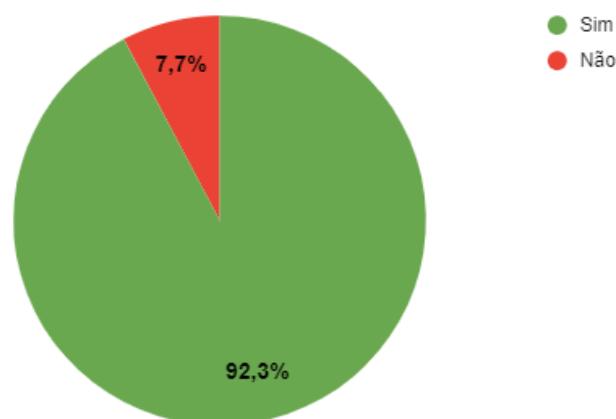
6.1 Eficiência

As indagações sobre eficiência buscavam identificar se os indivíduos, a partir de suas experiências, consideravam que o modelo seria benéfico nos cenários de atualização por obsolescência ou por transformação digital. Aplicou-se duas questões, a primeira, QE01, era de múltipla escolha e a segunda, QE02, dissertativa. Seus conteúdos estão descritos abaixo.

- QE01: Diante da sua experiência, você julga que o modelo proposto pode ser eficiente para os cenários de migrações por obsolescência ou por transformação digital?
- QE02: Quais pontos fortes você observa no modelo? Caso não ache eficiente, o que julga que pode ser melhorado?

As respostas obtidas em QE01, conforme mostra o gráfico da Figura 7, constataram que 92,3% dos entrevistados julgam que o modelo pode ser competente para as duas situações. Já os demais 7,7% não veem o fluxo como uma alternativa potente para minimizar os impactos dessas conjunturas.

Figura 7 – Gráfico de opiniões sobre eficiência do modelo



Fonte: Elaborado pelos autores.

Segundo os retornos de QE02, os participantes acreditam que a simplificação da migração de grandes sistemas, a abrangência dos fluxos e a adaptabilidade do modelo, são seus

pontos fortes. Para alguns, a proposta pode possibilitar o aumento da produtividade, o reaproveitamento de módulos e fornecer uma visão geral do processo, facilitando sua execução. Em contrapartida, parte dos entrevistados julgaram que a abordagem em cascata, a impossibilidade de mudanças nas decisões iniciais, a falta de etapas para análises de riscos e a ausência de avaliações de retrabalhos, são aspectos que precisam de atenção.

6.2 Aplicabilidade

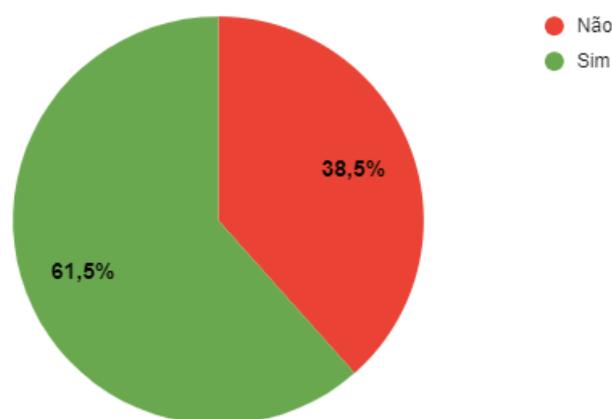
Nas diligências acerca de aplicabilidade, procurou-se validar se a proposta é de fácil entendimento e reprodução. Foram realizadas três perguntas, nas quais duas, QAP01 e QAP02, eram de múltipla escolha e uma, QAP03, dissertativa. Suas temáticas estão expostas a seguir.

- QAP01: Você considera as descrições do modelo de fácil entendimento?
- QAP02: Você julga que o modelo seria de fácil reprodução nas empresas governamentais?
- QAP03: Justifique por que você considera ou não o modelo de fácil reprodução.

Nos *feedbacks* de QAP01, os participantes unanimemente consideraram o modelo de fácil entendimento. Segundo relatos, os fluxos são simples, objetivos e estão bem descritos, o que facilita a compreensão.

As respostas de QAP02 e QAP03, em conformidade com a Figura 8, expõem que 61,5% dos indivíduos consideraram que o modelo é de fácil reprodução nas companhias. Os envolvidos citaram a adaptabilidade do processo, o escopo de descrição e a familiaridade dos profissionais com determinadas recomendações e tarefas, como aspectos favoráveis à adesão. Os demais 38,5% não julgaram que a proposta é de simples aplicação. Para eles, fatores como burocracias, cultura organizacional, paradigmas de desenvolvimento, limitação de recursos e a abordagem utilizada, impedem a aceitação dos fluxos.

Figura 8 – Gráfico de opiniões sobre aplicabilidade do modelo



Fonte: Elaborado pelos autores.

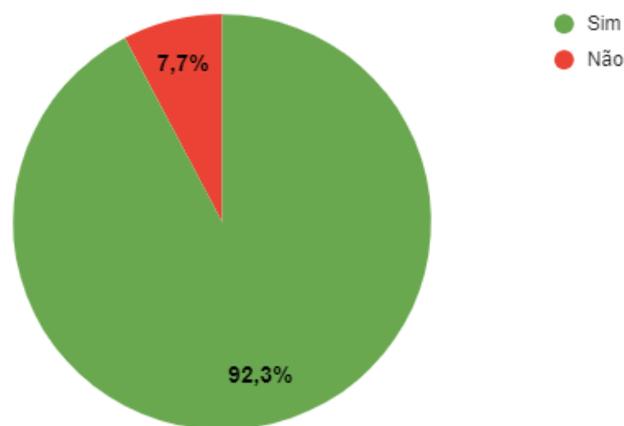
6.3 Possibilidade de aderência

Nos quesitos de aderência, investigou-se a possibilidade de aplicação do modelo nas companhias responsáveis por grupos de *software* governamentais. Foram realizadas as duas perguntas abaixo, QAD01 era de múltipla escolha e QAD02 dissertativa.

- QAD01: Você incorporaria o modelo em sua organização?
- QAD02: Conte sua opinião sobre a proposta de avaliação. Sinta-se à vontade para apontar pontos positivos e negativos e dar sugestões.

Consoante ao gráfico da Figura 9 e os resultados das investigações sobre eficiência, QAD01 mostrou que 92,3% dos colaboradores utilizariam o modelo em suas empresas e 7,7% não. Todos os partícipes que julgaram a proposta eficiente, afirmaram que a introduziriam em suas empresas.

Figura 9 – Gráfico de opiniões sobre aderência do modelo



Fonte: Elaborado pelos autores.

Nas perspectivas apuradas em QAD02, os profissionais citaram como pontos positivos a clareza, abordagem preventiva e adaptabilidade do processo. De acordo com as exposições, o modelo pode ser aplicado em projetos de baixo orçamento, contribui para impedir a formação de legados problemáticos e facilita a aderência da arquitetura. Como pontos negativos, alguns afirmaram que a proposta não leva em conta a validação da tomada de decisão, possui recomendações limitadas, pode gerar retrabalho e cancelamento do projeto e não contempla fluxos de avaliação contínua, análise de riscos ou políticas de governança.

7 CONSIDERAÇÕES FINAIS

O presente trabalho propôs um modelo de avaliação para a utilização de *microservices* em empresas governamentais. Esse modelo pretende contribuir para o aumento da qualidade dos sistemas, a adesão de *microservices* e a diminuição dos impactos das migrações.

Para sua concepção, definiu-se três objetivos específicos. Os dois primeiros visavam identificar a relação das companhias com os cenários de migração e a arquitetura de *microservices*, e foram alcançados por meio de um estudo de caso. A exploração mostrou que, mesmo após avanços tecnológicos no setor, as conjunturas de atualizações por obsolescência e por transformação digital ainda estão fortemente presentes nessas companhias, e que o número de recursos, a priorização de funcionalidade e o perfil de alguns colaboradores são os principais obstáculos para a utilização da arquitetura.

O terceiro propósito estabelecido foi a validação do modelo para o uso de *microservices* e ocorreu via pesquisa de opinião. Com ele, verificou-se que muitos profissionais consideram a arquitetura como uma maneira fácil e segura de migrar um legado. Além disso, segundo eles, os fluxos propostos podem realmente contribuir para a diminuição da sobrecarga dos colaboradores, a utilização da arquitetura e o aumento da qualidade do processo de migração. Para a maioria dos participantes, a proposta é capaz de ser eficiente e cumprir de forma satisfatória o objetivo para o qual foi criado, mesmo possuindo pontos negativos e não considerando alguns aspectos importantes.

Sendo assim, analisando sob os critérios dos aspectos de eficiência, aplicabilidade e aderência, conclui-se que o modelo é aceitável. Pois é de fácil compreensão e reprodução, propõe uma visão geral do processo e pode ser usado em projetos de baixo orçamento. Ademais, a proposta atende bem a demanda a qual descreve, mas ainda é uma solução primitiva e possui pontos a serem evoluídos.

Portanto, para pesquisas futuras, pretende-se adicionar aos fluxos uma abordagem iterativa e incremental, englobar a etapa de desenvolvimento e implantação no processo, inserir mais recomendações para as tarefas e propor comparativos entre essas dicas e introduzir o modelo em uma companhia.

REFERÊNCIAS

ABREU, V. F. d.; FERNANDES, A. A. **Implantando a governança de TI**. 4. ed. [S.l.]: BRASPORT Livros e Multimídia Ltda, 2014.

ALEXANDROVA, A.; RAPANOTTI, L.; HORROCKS, I. The legacy problem in government agencies: an exploratory study. In: INTERNATIONAL CONFERENCE ON DIGITAL GOVERNMENT RESEARCH. **Proceedings...** New York: Association for Computing Machinery, 2015. p. 150–159. Disponível em: <<https://dl.acm.org/doi/abs/10.1145/2757401.2757406>>. Acesso em: 1 set. 2021.

ALTHANI, B.; KHADDAJ, S. The applicability of system migration life cycle (smlc) framework. In: INTERNATIONAL SYMPOSIUM ON DISTRIBUTED COMPUTING AND APPLICATIONS TO BUSINESS, ENGINEERING AND SCIENCE (DCABES), 16, 2017. **Proceedings...** [S.l.], 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8253054>>. Acesso em: 28 ago. 2021.

ALTHANI, B.; KHADDAJ, S. Systematic review of legacy system migration. In: INTERNATIONAL SYMPOSIUM ON DISTRIBUTED COMPUTING AND APPLICATIONS TO BUSINESS, ENGINEERING AND SCIENCE (DCABES), 16, 2017. **Proceedings...** [S.l.], 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8253057>>. Acesso em: 20 ago. 2021.

CUADRADO, F. *et al.* A case study on software evolution towards service-oriented architecture. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS - WORKSHOPS, 22. **Proceedings...** [S.l.], 2008. p. 1399–1404. DOI: 10.1109/WAINA.2008.296. Disponível em: <<https://ieeexplore.ieee.org/document/4483115>>. Acesso em: 3 out. 2021.

FRITZSCH, J. *et al.* Microservices migration in industry: Intentions, strategies, and challenges. In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE AND EVOLUTION (ICSME), 2019. **Proceedings...** [S.l.], 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8919170/>>. Acesso em: 20 ago. 2021.

KHADKA, R. *et al.* How do professionals perceive legacy systems and software modernization? In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE), 2014. **Proceedings...** New York: Association for Computing Machinery, 2014. Disponível em: <<https://dl.acm.org/doi/abs/10.1145/2568225.2568318>>. Acesso em: 1 set. 2021.

LEON, P.; HORITA, P. Modernização de arquiteturas de sistemas para suporte à transformação digital. In: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 16, 2020, Porto Alegre. **Anais [...]**. Porto Alegre: SBC, 2020. p. 61–66. Disponível em: <https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/13128>.

LEWIS, J.; FOWLER, M. Microservices. 2014. Disponível em: <<https://martinfowler.com/articles/microservices.html>> Acesso em: 7 out. 2021.

LUZ, W. *et al.* An experience report on the adoption of microservices in three brazilian government institutions. In: BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING (SBES '18), 32, 2018. **Proceedings...** New York: Association for Computing Machinery, 2018. DOI: 10.1145/3266237.3266262. Disponível em: <<https://doi-org.ez93.periodicos.capes.gov.br/10.1145/3266237.3266262>>. Acesso em: 12 set. 2021.

NEWMAN, S. **Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith**. 1. ed. [S.l.]: O'Reilly Media, 2019.

RICHARDSON, C. Pattern: Strangler application. 2018. Disponível em: <<https://microservices.io/patterns/refactoring/strangler-application.html>> Acesso em: 01 abr. 2022.

SALVATIERRA, G.; MATEOS, C.; ET.AL. Legacy system migration approaches. In: IEEE LATIN AMERICA TRANSACTIONS, 11, 2013. **Proceedings...** [S.l.], 2013. p. 840–851. DOI: 10.1109/TLA.2013.6533975. Disponível em: <<https://ieeexplore.ieee.org/document/6533975>>. Acesso em: 22 ago. 2021.

SHAW, M.; GARLAN, D. **Software Architecture. Perspectives on an Emerging Discipline.** 1. ed. [S.l.]: Prentice Hall, 1996.

STROBL, S.; BERNHART, M.; GRECHENIG, T. Towards a topology for legacy system migration. In: PROCEEDINGS OF THE IEEE/ACM, 42, 2020, New York. **Proceedings...** New York: Association for Computing Machinery, 2020. Disponível em: <<https://doi-org.ez93.periodicos.capes.gov.br/10.1145/3387940.3391476>>. Acesso em: 22 ago. 2021.

TAIBI, D.; LENARDUZZI, V.; PAHL, C. Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation. In: IEEE CLOUD COMPUTING, 5, 2017. **Proceedings...** [S.l.], 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8125558>>. Acesso em: 29 ago. 2021.

VERNER, J. *et al.* Guidelines for industrially-based multiple case studies in software engineering. In: INTERNATIONAL CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE, 3., [S.l.], 2009. **Anais [...]**. [S.l.], 2009. p. 313–324.