

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
PERNAMBUCO

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**

**CAMPUS JABOATÃO DOS GUARARAPES**

Pós-Graduação em Gestão e Qualidade em Tecnologia da Informação  
e Comunicação

DANDHARA PATRICIA RIBEIRO DO MONTE

**ARQUITETURA DE MICROSERVIÇOS: QUANDO VALE A PENA  
MIGRAR?**

**Jaboatão dos Guararapes**

**2020**



DANDHARA PATRICIA RIBEIRO DO MONTE

## ARQUITETURA DE MICROSERVIÇOS: QUANDO VALE A PENA MIGRAR?

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do diploma de Pós-Graduação *Lato Sensu* em Gestão e Qualidade em Tecnologia da Informação e Comunicação do IFPE Campus Jaboatão dos Guararapes, sob orientação o professor Roberto Luiz Sena Alencar.

**Jaboatão dos Guararapes**

**2020**



## FICHA CATALOGRÁFICA

M772a Monte, Dandhara Patrícia Ribeiro do.

Arquitetura de microsserviços: quando vale a pena migrar? / Dandhara Patrícia Ribeiro do Monte; orientador Prof. Ms. Roberto Luiz Sena Alencar. Jaboatão dos Guararapes: IFPE, 2020.

58f.; il.

Trabalho de Conclusão de Curso (Especialização em Gestão e Qualidade em Tecnologia da Informação e Comunicação) – IFPE - Campus Jaboatão dos Guararapes.

Inclui Referências.

1. Tecnologia da Informação e Comunicação 2. Engenharia de Software.  
3. Arquitetura de Software. 4. Migração arquitetural. I. Monte, Dandhara Patrícia Ribeiro do. II. IFPE. III. Título.

CDD 005.1068

## Resumo

O crescimento contínuo da complexidade nos atuais sistemas ocasionado, dentre outros, pelo exponencial alcance da internet, levou a indústria de software e a comunidade acadêmica à investigar novas formas de resolver problemas que atendessem a necessidade dos sistemas operarem em um contexto mais amplo, dado a explosão de informações e a maior necessidade de integração entre sistemas e serviços. Tradicionalmente, as empresas constroem seus sistemas de forma monolítica e posteriormente se entenderem que há a necessidade de "dividir" a aplicação utiliza-se de maneira incremental a separação desta criando assim os microsserviços (J. THÖNES, 2015). Microsserviços é um estilo e padrão de arquitetura de software em que sistemas complexos são decompostos em serviços menores que trabalham em conjunto para formar serviços maiores. Os Microsserviços são serviços autônomos, independentes e independentemente implantáveis. Este trabalho se propõe a auxiliar o arquiteto e/ou desenvolvedor de software através de uma proposta de aplicação de um questionário que visa facilitar a tomada de decisão em relação a migração arquitetural de um projeto de software. Foi realizado um estudo bibliográfico, onde foram analisados e identificados vários fatores em um projeto, com isto, sugerem que uma migração arquitetural para microsserviços que pode trazer benefícios substanciais ao projeto. Nesse contexto, os arquitetos e/ou desenvolvedores, podem utilizar o questionário proposto por este trabalho como uma ferramenta auxiliar no processo de apoio à decisão para organização, através de embasamento teórico sobre as principais características que motivam a migração arquitetural, como manutenibilidade, escalonamento e utilização tecnologias distintas, junto a exemplos citados na literatura sobre quando se deve migrar ou não.

**Palavras-chaves:** Engenharia de Software; Arquitetura de Software; Microsserviços  
Migração Arquitetural.

## **Abstract**

*The continuous growth of complexity in current systems caused, among others, by the exponential reach of the internet, has led the software industry and the academic community to investigate new ways of solving problems that meet the need for systems to operate in a larger context, given the explosion of information and the greater need for integration between systems and services. Traditionally, companies build their systems in a monolithic way and if they later understand that there is a need to "divide" the application, it is used incrementally to separate it, so creating microservices (J. THÖNES, 2015). Microservices is a style and standard of software architecture in which complex systems are broken down into smaller services that work together to form larger services. Microservices are autonomous, independent and independently implantable services. This work aims to assist the architect and / or software developer through a proposal to apply a questionnaire that wants to facilitate decision making regarding the architectural migration of a software project. A bibliographic study was made, where several factors were analyzed and identified in a project, with this, they suggest that an architectural migration to microservices that can bring substantial benefits to the project. In this context, architects and / or developers can use the questionnaire proposed by this work as an auxiliary tool in the decision support process for the organization, through a theoretical basis on the main characteristics that motivate architectural migration, such as maintainability, scalability and use different technologies, along with examples mentioned in the literature that say when to migrate or not.*

**Keywords:** *Software Engineering; Software Architecture; Microservices Architectural Migration.*



## Índice de Tabelas

<b>Tabela 3.1</b> -Comparação das arquiteturas monolítica e microsserviços (Autoria Própria, 2019).....	28
---	----



## Índice de Figuras

<b>Figura 1.1-</b> Custo de produtividade de uma arquitetura de microsserviços em relação a um sistema monolítico (FOWLER, 2015).....	12
<b>Figura 2.1-</b> Evolução do acoplamento intra-arquitetural (MORRISON, 2015).....	16
<b>Figura 2.2-</b> SOA tradicional versus microsserviços (MORRISON,2015).....	17
<b>Figura 2.3-</b> Dependências da perspectiva de um desenvolvedor (MORRISON, 2015).....	18
<b>Figura 2.4-</b> Pipeline de construção básica (FOWLER & JAMES, 2014).....	22
<b>Figura 4.1-</b> Análise das respostas dos cargos (Fonte: relatório extraído da ferramenta SurverMonkey, 2019).....	37
<b>Figura 4.2-</b> Análise das respostas sobre tamanho da empresa (Fonte: relatório extraído da ferramenta SurverMonkey, 2019).....	38
<b>Figura 4.3-</b> Análise das respostas sobre viabilidade de recursos (Fonte: relatório extraído da ferramenta SurverMonkey, 2019).....	39
<b>Figura 4.4-</b> Análise de respostas sobre o nível de acoplamento do sistema (Fonte: relatório extraído da ferramenta SurverMonkey, 2019).....	40
<b>Figura 4.5-</b> Análise das respostas sobre escalabilidade do sistema (Fonte: relatório extraído da ferramenta SurverMonkey, 2019).....	41
<b>Figura 4.6-</b> Análise das respostas sobre o uso de tecnologias distintas no mesmo projeto (Fonte relatório do SurverMonkey, 2019).....	42
<b>Figura 4.7-</b> Análise das respostas sobre identificar erros na manutenção do sistema (Fonte relatório do SurverMonkey, 2019).....	43
<b>Figura 4.8-</b> Análise das respostas sobre complexidade manutenção do sistema (Fonte relatório do SurverMonkey, 2019).....	44
<b>Figura 4.9-</b> Análise das respostas sobre reutilização de partes do sistema (Fonte relatório do SurverMonkey, 2019).....	45
<b>Figura 4.10-</b> Análise das respostas sobre integração do sistema (Fonte relatório do SurverMonkey, 2019).....	46



<b>Figura 4.11-</b> Análise das respostas sobre implementação de módulos no sistema (Fonte relatório do SurverMonkey, 2019).....	47
<b>Figura 4.12-</b> Análise das respostas sobre disponibilização de novas versões (Fonte relatório do SurverMonkey, 2019).....	48
<b>Figura 4.13-</b> Análise das respostas sobre monitoramento de partes específicas do sistema (Fonte relatório do SurverMonkey, 2019).....	49
<b>Figura 4.14-</b> Análise das respostas sobre disponibilidade do sistema (Fonte relatório do SurverMonkey, 2019).....	50
<b>Figura 4.15-</b> Resumo da apuração das respostas do questionário. (Fonte: autoria própria, 2019).....	51





## Sumário

<b>1. Introdução</b>	<b>10</b>
<b>1.1. Justificativa e Motivação</b>	<b>11</b>
<b>1.2. Objetivo Geral</b>	<b>13</b>
<b>1.3. Objetivos Específicos</b>	<b>13</b>
<b>2. Fundamentação teórica</b>	<b>14</b>
<b>2.1. Arquitetura de Software</b>	<b>14</b>
<b>2.2. Arquitetura de Monolítico</b>	<b>15</b>
<b>2.3. Arquitetura de Microsserviços</b>	<b>17</b>
<b>2.4. Análise Comparativa Entre as Principais Características das Arquiteturas Monolítica e Microsserviço</b>	<b>19</b>
<b>3. Questionário de Migração de uma Aplicação Monolítica para Microsserviços</b>	<b>24</b>
<b>3.1. Metodologia do Experimento</b>	<b>24</b>
<b>3.2. Principais Características da Migração de Sistemas com Arquitetura Monolítica para Microsserviços</b>	<b>25</b>
<b>3.3. Elaboração das Perguntas do Questionário</b>	<b>29</b>
<b>4. Aplicação do Questionário em Aplicações Monolíticas</b>	<b>36</b>
<b>5. Conclusão</b>	<b>53</b>
<b>5.1. Trabalhos futuros</b>	<b>54</b>
<b>Referências Bibliográficas</b>	<b>54</b>
<b>Apêndices</b>	<b>59</b>



## **Agradecimentos**

Gostaria de começar agradecendo a Deus pela oportunidade de ingressar nesta especialização e todo cuidado até aqui. Agradeço ao meu professor e orientador Roberto por toda paciência e todo apoio no desenvolvimento deste trabalho. Por fim, agradeço a minha família e amigos, principalmente ao meu marido Lennon, por sempre me apoiar e acreditar em mim.

## 1. Introdução

Com o passar dos anos o cenário de desenvolvimento vem crescendo bastante e com isso vem surgindo novas tecnologias, como linguagens de programação, paradigmas e processos. Então, para acompanhar tais mudanças as empresas estão atualizando-se principalmente na área de arquitetura de aplicações web e migrando seus serviços (Di Francesco, 2017).

Segundo Machado (2017) existem muitas arquiteturas de software e uma bastante utilizada é a monolítica. Conforme Martins (2007, p. 18), “na arquitetura monolítica o software é construído em um único módulo [...]”. Um exemplo simples seria uma aplicação implementada na linguagem de programação C# utilizando um banco de dados relacional, como SqlServer.

Rouse (2016) salienta que os sistemas de arquitetura monolíticas são planejadas para serem autocontidas e os componentes da aplicação são interconectados e interdependentes. Em caso de realização de alterações de funcionalidades da mesma, existe a possibilidade de interferir em outras funcionalidades, por conta da alta dependência da aplicação.

Devido a tais fatos, outra arquitetura mais recente está sendo bastante disseminada, a mesma é chamada de microsserviços, que conforme Di Francesco (2017) sua proposta possui o objetivo de reutilização do código e de facilitar a manutenção de aplicações por meio de serviços.

Segundo Folwer e James (2014) um componente é uma unidade de software que é independentemente substituível e atualizável. Então, uma das principais características da arquitetura de microsserviços é a forma de componentizar seu próprio software o dividindo em serviços.

Outro benefício do microsserviços é que pode ser escrita por diferentes equipes e diferentes linguagens de programação, podendo também ser testada

individualmente (STOIBER, 2017). Então facilita a agilidade de implementação e na aplicabilidade da melhor linguagem de programação focada para cada microsserviços.

Um desafio ao usar microsserviços é justamente decidir quando faz sentido usá-la. Muitas vezes ao desenvolver a primeira versão de um aplicativo não ocorrem os problemas que essa abordagem resolve (RICHARDSON, 2019).

Este trabalho possui o intuito de ajudar o arquiteto e a organização a decidir quando migrar um sistema de arquitetura monolítica para arquitetura de microsserviços. A proposta é a aplicação de um questionário que visa facilitar esta tomada de decisão se deve migrar ou não, com embasamento teórico sobre as principais características e exemplos citados na literatura.

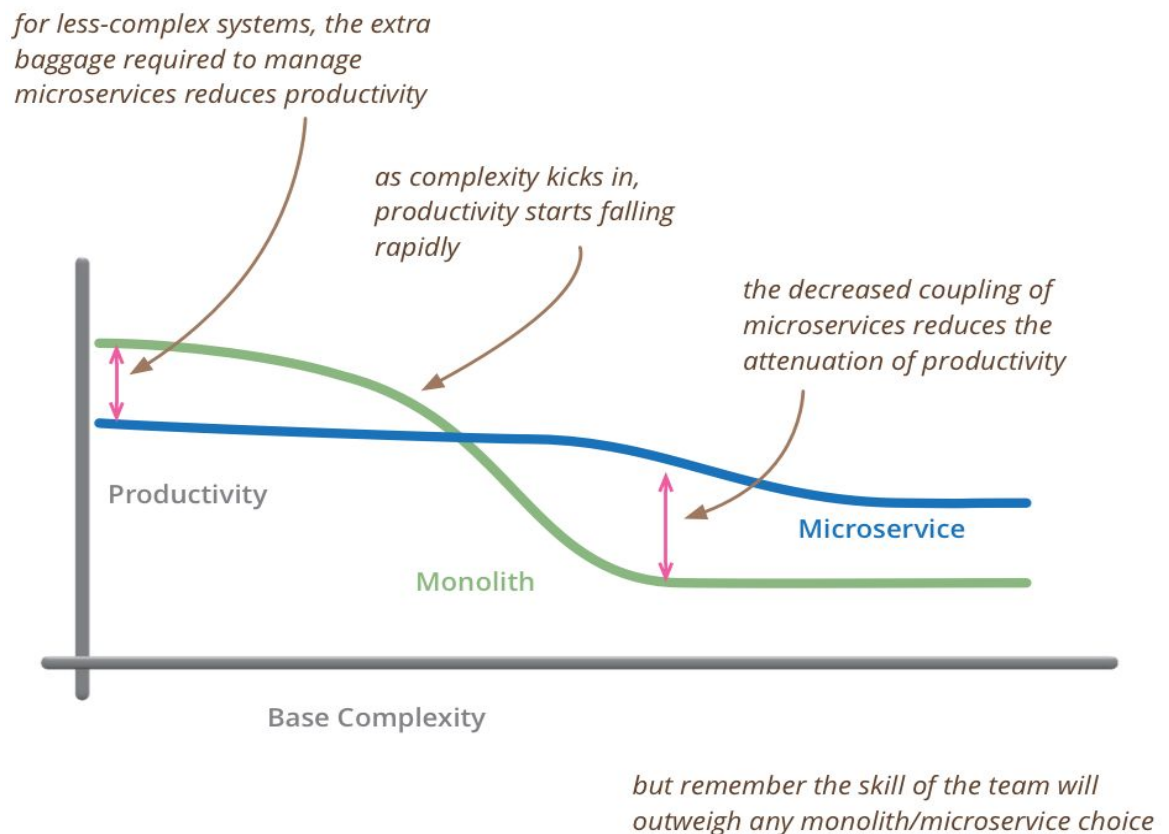
## **1.1. Justificativa e Motivação**

Com o crescimento contínuo da complexidade nos atuais sistemas ocasionado, dentre outros, pelo exponencial alcance da internet, levou a indústria de software e a comunidade acadêmica à investigar novas formas de resolver problemas que atendessem a necessidade dos sistemas operarem em um contexto mais amplo, dado a explosão de informações e a maior necessidade de integração entre sistemas e serviços. Muitas organizações de software implementaram ao longo do tempo grandes aplicações estruturadas em um único "bloco de software", seguindo uma estrutura conhecida como arquitetura monolítica. Empiricamente a maioria das empresas constroem seus sistemas de forma monolítica e posteriormente entenderem que há a necessidade de "dividir" a aplicação utiliza-se de maneira incremental a separação desta criando assim os microsserviços (J. THÖNES, 2015).

Muitas organizações descobriram que a utilização de arquiteturas de microsserviços proporciona maior liberdade para a adoção de novas tecnologias resultando muitas vezes num processo de entregas mais rápidas. Os microsserviços proporcionam muito mais liberdade para reagir e tomar decisões mais rapidamente, permitindo atender às constantemente mudanças de requisitos (NEWMAN, 2015).

De acordo com Fowler (2015), a complexidade do sistema a ser desenvolvido deve ser considerada para decidir pela utilização da arquitetura de microsserviços ou não, pois a “perda” de produtividade proporcionada pelo custo de projetar uma arquitetura desse tipo é superada à medida que a complexidade do sistema aumenta. Com isso, Fowler sugere que a produtividade de desenvolvimento de um sistema complexo é maior utilizando arquitetura de microsserviços em relação a criação de um sistema monolítico, conforme pode ser observado na Figura 1.1.

**Figura 1.1-** Custo de produtividade de uma arquitetura de microsserviços em relação a um sistema monolítico



Fonte: (FOWLER, 2015)

Microsserviços demonstraram recentemente ser um paradigma arquitetural eficaz para lidar com a complexidade do software e, em particular, com a crescente necessidade de proporcionar escalabilidade aos sistemas. Decidir se e quando migrar um sistema de arquitetura monolítica para arquitetura de

microsserviços pode ser bastante difícil para o arquiteto e a organização. Portanto, existe a necessidade de ferramentas de apoio a decisão de para o processo de migração de um sistema monolítico para um sistema de microsserviços. Nesse contexto, este trabalho pode auxiliar o arquiteto e o desenvolvedor de software através da proposta de aplicação de um questionário que visa facilitar a tomada de decisão da migração de arquitetura ou não. O questionário pode ser utilizado como apoio a decisão estratégia para organização, através de embasamento teórico sobre as principais características e exemplos citados na literatura sobre quando se deve migrar ou não.

## **1.2. Objetivo Geral**

Este trabalho tem como objetivo auxiliar o arquiteto ou desenvolvedor de software através do desenvolvimento de um questionário que visa auxiliar a tomada de decisão em relação migração de arquitetura ou não.

## **1.3. Objetivos Específicos**

- Identificar as características citadas na literatura sobre principais das arquiteturas de softwares monolítica e microsserviços;
- Identificar quais fatores levam a migração ou não de um sistema monolítico para microsserviços;
- Enriquecer a fonte de pesquisa sobre o assunto, analisando os resultados da aplicação do questionário através de embasamento teórico sobre as principais características e exemplos citados na literatura sobre quando se deve migrar ou não;

Para alcançar os objetivos o trabalho encontra-se organizado da maneira descrita a seguir. No capítulo 2 é abordada a fundamentação teórica. No capítulo 3 são levantadas as principais características da migração de sistemas com arquitetura monolítica para microsserviços e apresentada a proposta do

questionário. No capítulo 4 será mostrado o resultado da aplicação do questionário e à análise quantitativa de suas respectivas respostas. No capítulo 5 é apresentado a conclusão deste trabalho e trabalhos futuros.

## **2. Fundamentação teórica**

### **2.1. Arquitetura de Software**

Com o passar dos anos pode-se perceber que junto ao aumento da complexidade e abrangência de sistemas de informação, também ocorreu o aumento da complexidade em organizar e estruturar os componentes dos softwares (SORDI, MARINHO & NAGY, 2006). A arquitetura é considerada o esqueleto do software, então, torna-se o plano de mais alto-nível da construção dos novos sistemas. O planejamento da arquitetura de software faz com que a estrutura suporte melhor o sistema idealizado (KRAFZIG, BANKE & SLAMA, 2004).

Pode-se dizer que a arquitetura comporta-se agindo como um facilitador na construção dos sistemas que provêm da mesma linha de produção, então significa que essa arquitetura facilita a elaboração de softwares que possuem a mesma estrutura ou escopo (QUEIROZ & BRAGA, 2008). A estrutura do sistema, que são os elementos de software, junto às propriedades visíveis e externas desses elementos, incluindo a relação entre eles. Essa estrutura também é compreendida como arquitetura de software de um programa ou um sistema computacional (BASS, CLEMENTS & KAZMAN, 2003).

### **2.2. Arquitetura de Monolítico**

A arquitetura de monolítico é a mais usada em softwares de corporativos, desde os primeiros dias de implementação de aplicativos para a Web, é aquela que agrupa todos os componentes do lado do servidor em uma única unidade. Muitos aplicativos Java, Ruby e até C++ consistem em um único arquivo (RICHARDSON, 2014).

A arquitetura monolítica possui uma lógica para lidar com a solicitação sendo executada em um único processo. Então permite usar os recursos mais básicos de suas linguagens, inclusive testar e executar o aplicativo no laptop do próprio desenvolvedor (FOWLER, 2014).

De acordo com Richardson (2014) aplicações monolíticas são simples de desenvolver, cuja as IDEs e outras ferramentas de desenvolvimento são orientados para o desenvolvimento de um único aplicativo. Também salienta que funciona bem para aplicativos relativamente pequenos, são fáceis de testar, pois apenas precisa iniciar o aplicativo e são simples de implantar, pois basta copiar a unidade de implementação.

Toda a lógica de um aplicativo monolítico é executada apenas em um único processo, assim possibilita que utilize recursos básicos da sua linguagem para dividir a aplicação em classes, funções e namespaces. Esta arquitetura proporciona a execução de testes do aplicativo no computador do desenvolvedor e usar um pipeline de implantação para garantir que as alterações sejam testadas e implantadas adequadamente na produção. Podendo também ser executado em muitas instâncias atrás de um balanceador de carga (FOWLER & JAMES, 2014).

Infelizmente, as aplicações monolíticas acabam se tornando difíceis de entender e modificar, principalmente quando começam a crescer ou tenham de substituir desenvolvedores nas equipes (NAMBIOT, 2014).

### **2.3. Arquitetura de Microsserviços**

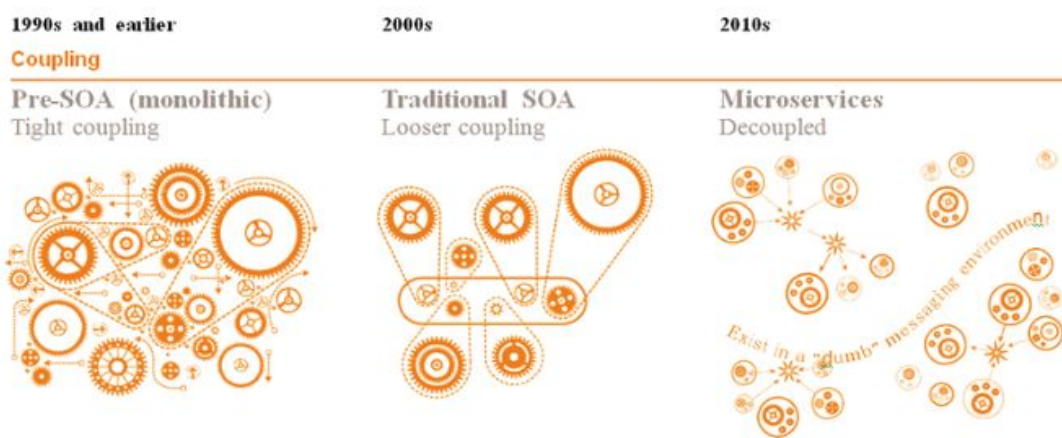
O termo microsserviços surgiu durante a conferência de arquitetos de software (ICSAE) em maio 3 de 2011 e vem sendo bastante difundido desde então na comunidade de arquitetos, engenheiros de softwares e desenvolvedores (MACHADO, 2017).

Segundo Fowler e James (2014), arquitetura de microsserviços é uma maneira de projetar aplicativos utilizando suítes de serviços sendo implementadas independentemente. (FOWLER, 2014). Esta arquitetura constrói aplicativos como



conjuntos de serviços implantados independentemente de maneira escaláveis. Então, esta arquitetura proporciona que diferentes serviços sejam escritos em diferentes linguagens de programação. Essa arquitetura é uma abordagem para desenvolver um único aplicativo como um conjunto de pequenos serviços, onde são executados individualmente. Esses serviços são implementados de maneira independente e com implantação automatizada (FOWLER & JAMES, 2014).

**Figura 2.1:** Evolução do Acoplamento Intra-arquitetura



Fonte: (MORRISON, 2015).

Como pode-se analisar na figura 2.1, a arquitetura de microsserviços pode ser considerada uma variação da Arquitetura Orientada a Serviços (SOA). Segundo Zimmermann (2016) o SOA e microsserviços possuem um nível de desenvolvimento e implantação de aplicações distribuídas. Porém, microsserviços difere-se pela utilização de um mecanismo leve e individual para a comunicação dos serviços conforme está na figura 2.2.

**Figura 2.2-** SOA tradicional versus microsserviços

	<b>Traditional SOA</b>	<b>Microservices</b>
<b>Messaging type</b>	Smart, but dependency-laden ESB	Dumb, fast messaging (as with Apache Kafka)
<b>Programming style</b>	Imperative model	Reactive actor programming model that echoes agent-based systems
<b>Lines of code per service</b>	Hundreds or thousands of lines of code	100 or fewer lines of code
<b>State</b>	Stateful	Stateless
<b>Messaging type</b>	Synchronous: wait to connect	Asynchronous: publish and subscribe
<b>Databases</b>	Large relational databases	NoSQL or micro-SQL databases blended with conventional databases
<b>Code type</b>	Procedural	Functional
<b>Means of evolution</b>	Each big service evolves	Each small service is immutable and can be abandoned or ignored
<b>Means of systemic change</b>	Modify the monolith	Create a new service
<b>Means of scaling</b>	Optimize the monolith	Add more powerful services and cluster by activity
<b>System-level awareness</b>	Less aware and event driven	More aware and event driven

Fonte: (MORRISON, 2015)

Segundo Machado (2017), com a chegada da cultura DevOps, no contexto de desenvolvimento web, surgiu a proposta da arquitetura de microsserviços e ela baseia-se em micro aplicações independentes, diferentemente da que existe na arquitetura monolítica. Microsserviços se comporta de forma isolada, específica e independente de um sistema de porte maior, onde cada serviço possui responsabilidades únicas dentro da arquitetura.

De acordo com Machado (2017) microsserviços são importantes para os negócios, pois as empresas desejam acompanhar a evolução do mercado para reduzir os custos e também proporcionar agilidade na criação de novos produtos e

serviços. Para que ocorra tais evoluções existem três pontos importantes para as aplicações:

- Uma boa experiência de usuário para o cliente, bastante interativa e dinâmica e disponíveis em diferentes plataformas, inclusive dispositivos móveis.
- As aplicações devem ser escaláveis, com alto nível de disponíveis e devem estar em ambientes de nuvem.
- As atualizações devem ocorrer de maneira suave, sem instabilidade e paradas, para não prejudicar os usuários.

Fowler e James (2014) citam que este estilo arquitetural é uma ideia importante e merece ser considerada pelas aplicações corporativas. Empresas como Amazon, Netflix, The Guardian, Serviço Digital do Governo Do Reino Unido, realestate.com.au, Forward e comparethemarket.com são pioneiros na utilização desta arquitetura.

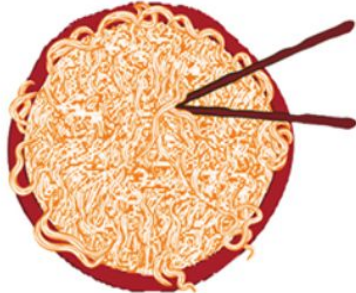
Na visão de Morrison (2015) esta arquitetura objetiva diminuição das dependências do projeto, proporciona melhores testes e uma rápida implantação de novas versões dos módulos do sistema, desta forma, possibilitando mais modularidade no sistema. Na figura 2.3, pode-se observar a diferença de modularidade dos tipos de arquitetura.

**Figura 2.3** - Dependências da perspectiva de um desenvolvedor



1990s and earlier

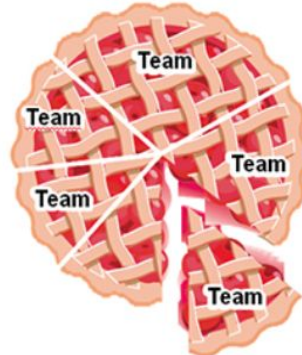
Pre-SOA (monolithic)  
Tight coupling



For a monolith to change, all must agree on each change. Each change has unanticipated effects requiring careful testing beforehand.

2000s

Traditional SOA  
Looser coupling



Elements in SOA are developed more autonomously but must be coordinated with others to fit into the overall design.

2010s

Microservices  
Decoupled



Developers can create and activate new microservices without prior coordination with others. Their adherence to MSA principles makes continuous delivery of new or modified services possible.

Fonte: (Morrison, 2015)

## 2.4. Análise Comparativa Entre as Principais Características das Arquiteturas Monolítica e Microsserviço

Uma facilidade da arquitetura monolítica é na sua implantação, pois a aplicação consiste em um único programa executável monolítico (MACHADO, 2017). Entretanto ele afirma que aplicações que usam arquitetura monolítica tendem a aumentar sua complexidade com o tempo, então acaba tornando a leitura e manutenção mais árduos e lentos.

Um software monolítico possui algumas vantagens e uma delas é a maior facilidade no desenvolvimento da aplicação, pois possui acervo bem maior de ferramentas no mercado que facilitam no suporte ao desenvolvimento. As principais forças que os softwares monolíticos é a possibilidade de possuir apenas uma equipe trabalhando no aplicativo e também novas pessoas na equipe de desenvolvimento tornam-se rapidamente produtivas (RICHARDSON, 2019).

Machado (2017) salienta mais duas vantagens de usar arquitetura monolítica é a facilidade de implantação e escalonamento de alguma aplicação, pois em microsserviços tem que fazer implantação de vários serviços. Mas para manter

escalonamento em aplicações monolíticas existe um custo muito alto, pois é necessário replicar todo o sistema, mesmo que apenas algumas funcionalidades sejam necessárias. Uma das vantagens de microsserviços é em relação a necessidade de consistência e integridade dos dados, pois para aplicações monolíticas essa é uma tarefa mais complexa.

Segundo Richardson (2019) o aplicativo monolítico deve ser fácil de entender e ser modificado, também deve existir a possibilidade de escalar e manter disponível. Realça que é simples de desenvolver, implantar e dimensionar. Mas quando o aplicativo se torna grande possui algumas desvantagens:

- Quando o código da aplicação estiver muito grande, pode ser mais difícil de ser compreendida e modificada. Com isso, o desenvolvimento normalmente acaba ficando mais lento, pois os intimida os desenvolvedores, principalmente se são novos no time.
- Os desenvolvedores acabam sendo menos produtivos quando o código está muito grande, pois o IDEs são mais lentos e podem ser sobrecarregados.
- O processo de implantação contínua pode ser mais difícil, pois para atualizar uma funcionalidade tem que atualizar o sistema inteiro.
- Para escalar a aplicação também pode ser mais complexo, já que a arquitetura não pode escalar cada componente independentemente.
- Para dividir organizar que os integrantes da equipe consigam trabalhar divididos por funcionalidade, também fica mais complexo, pois todo o sistema é desenvolvido em um único componente.
- Tem que manter todas as tecnologias escolhidas no início do projeto, caso deseje mudar em alguns casos é arriscado ou impraticável.

Então, observa-se que ao tratar de aplicações mais complexas e de base de código grande, a arquitetura monolítica pode ser um processo mais complexo, arriscado, demorado, principalmente se exigir a coordenação de muitos desenvolvedores. Também é necessário realizar longos ciclos de teste para casos de implantações mais frequentes, tornando-se mais difícil de manter e também na

adoção de novas tecnologias. Concluindo que a arquitetura monolítica não é dimensionável para suportar aplicativos grandes e de longa duração (RICHARDSON, 2014).

Segundo Fowler e James (2014) os ciclos de mudança passam a ser mais interligados quando trabalha-se com implantação na nuvem, quando a aplicação é monolítica uma alteração feita em uma pequena parte do aplicativo é necessário atualizar todo o sistema. Desta forma realizar a manutenção com estrutura modular torna-se mais difícil (FOWLER & JAMES, 2014).

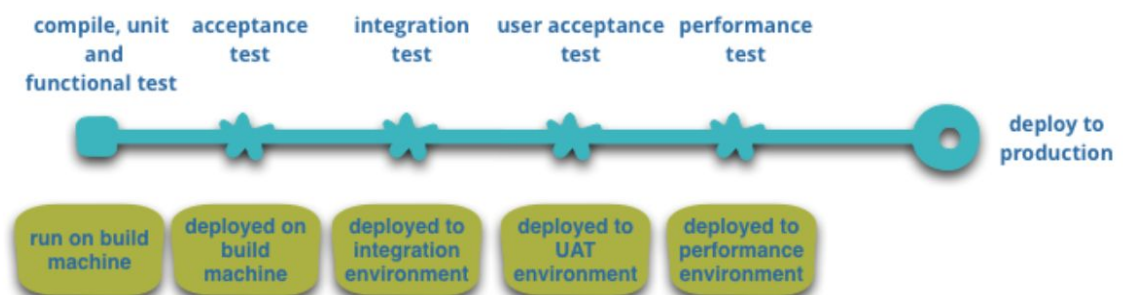
No estudo feito por Odravison Amaral e Marcus Carvalho (2017), realizou uma comparação teórica entre as arquiteturas monolítica e microsserviços. Os mesmos realizaram experimentos em duas aplicações idênticas, mas utilizando as diferentes arquiteturas. A conclusão baseia-se que para aplicações que exigem baixo nível de complexidade, que precisam de um bom desempenho e não precisam ser escaláveis, o ideal é a arquitetura monolítica, por ter mais poder de resposta neste contexto. A arquitetura de microsserviços se aplica a soluções que necessitam escalonamento, facilidade de integrar os dados. Também é ideal para casos de alta produtividade do time de desenvolvimento, pois segundo foi analisado em sua pesquisa, o isolamento das funcionalidades de uma aplicação é um dos principais fatores que facilitam a produtividade.

Segundo Toby Clemson (2014) testes unitários desempenham um papel importante dentro de um microsserviços, pois na medida do possível, objetos de domínio reais devem ser usados para todos os colaboradores da unidade do mesmo. Neste nível observa-se que é fornecido um feedback mais rápido do que os testes de integração e podem forçar as condições de erro fazendo com que respondam como uma dependência externa faria em circunstâncias excepcionais. Porém salienta que os testes de unidade não garantem em relação ao comportamento do sistema, que já é característica do teste de integração. Informa que para manter e escrever testes “end-to-end” pode ser muito difícil, e possui um custo adicional de manutenção do conjunto de testes.

Segundo Richardson (2014), um ponto a negativo em em relação a microsserviços é que ele possui um custo maior de planejamento dos arquitetos e desenvolvedores de software. Mas, um ponto positivo é a total independência, possibilitando que cada microsserviços possua seu próprio banco de dados.

Com um monólito qualquer alteração requer compilação e implementação completas de todo o aplicativo, diferente do microsserviços, pois como coloca componentes em serviços possibilita uma oportunidade de implementação mais modularizadas. As equipes que buscam construir software com esta arquitetura, buscam utilizar técnicas de automação de infraestrutura para integração contínua. Tal procedimento é ilustrado na figura abaixo (FOWLER & JAMES, 2014).

**Figura 2.4-** Pipeline de construção básica



Fonte: (FOWLER & JAMES, 2014)

Na figura 2.4 representa um pipeline de construção, onde Folwer e James (2014) salientam que são utilizados devido ao desejo de proporcionar maior confiabilidade no software, assim surgiu a necessidade de executar muitos testes automatizados.

Segundo Fowler (2015) Microsserviços é uma abordagem superior a arquitetura monolítica. O mesmo informou que existem três benefícios, o primeiro seria a estrutura modular, que é importante para equipes maiores.

Outro seria a implantação independente, pois, os serviços simples são mais fáceis de implantar e já que são autônomos possuem melhor isolamento de falhas. O terceiro seria a facilidade de utilizar tecnologias distintas, pois pode misturar vários idiomas, estruturas de desenvolvimento e diferentes tecnologias de armazenamento de dados.

Uma desvantagem levantada por Odravison Amaral e Marcus Carvalho (2017) é que devido à recente aparição da arquitetura de microsserviços existe falta de ferramentas de desenvolvimento para aplicações.

Microsserviços possuem três desvantagens, a primeira seria em relação à distribuição, pois que acabam sendo mais difíceis de programar, visto que as chamadas remotas são lentas e possuem mais risco de falha. A segunda seria a consistência eventual, pois para mantê-la forte é extremamente difícil para um sistema distribuído e sendo assim, para usar esta arquitetura deve-se realizar também este gerenciamento. A terceira seria levantou sobre a complexidade operacional, onde é necessário uma equipe de operações madura para gerenciar muitos serviços, que estão sendo atualizados continuamente.



### **3. Aplicação de um Questionário para Avaliar a Necessidade de Migração de uma Aplicação Monolítica para Microsserviços**

A contribuição deste trabalho trata-se da aplicação de um questionário que visa facilitar a tomada de decisão da migração de arquitetura ou não. O questionário pode ser utilizado como apoio a decisão estratégica para organização, através de embasamento teórico sobre as principais características e exemplos citados na literatura sobre quando se deve migrar ou não. A decisão de migração é identificada conforme as necessidades específicas de cada sistema e deve ser analisado minuciosamente.

Esta proposta de questionário será realizada especificamente para sistemas monolíticos. Os pontos que foram levados em consideração foram termos como escalabilidade, manutenção e tecnologias distintas.

#### **3.1 Metodologia do Experimento**

Neste trabalho foi aplicada uma pesquisa qualitativa e quantitativa do tipo survey. De acordo com Babie (1999), a adoção desse tipo de pesquisa permite a elaboração clara e rigorosa de um modelo lógico, possibilitando ao pesquisador

documentar processos causais mais complexos. O mesmo destaca a possibilidade da generalização, permitindo entender a população da qual a amostra foi retirada. O autor destaca que nesta pesquisa os estudiosos possuem, à sua disposição, um vasto número de variáveis, estando em situação bastante favorável para realizar análise criteriosa da importância de cada uma delas, podendo quantificá-las e processá-las em computador. O mesmo cita que é bastante utilizado no mundo acadêmico.

Desta forma, este trabalho está dividido em três partes. Primeiro, será levantado um embasamento teórico sobre as principais características e exemplos citados na literatura sobre quando se deve migrar ou não. Segundo, com as principais características encontradas será realizado o embasamento teórico a elaboração das perguntas. Terceiro, será realizado o estudo quantitativo e qualitativo do resultado da aplicação do questionário.

As elaborações das perguntas estão divididas em duas seções. A primeira apresenta um embasamento teórico sobre as principais características e exemplos citados na literatura sobre quando se deve migrar ou não e mostra uma tabela comparativa entre a arquitetura monolítica e a de microsserviços. Na segunda sessão, são apresentadas as perguntas que, utilizando o embasado nas características levantadas anteriormente, são utilizadas no questionário proposto por este trabalho.

### **3.2 Principais Características da Migração de Sistemas com Arquitetura Monolítica para Microsserviços**

De acordo Fowler (2015), por mais que exista a certeza que o sistema será robusto o suficiente para usar microsserviços, não é interessante iniciar um novo projeto com tal arquitetura. Ele cita que uma das maneiras mais comuns de migração para arquitetura em microsserviços é a partir de um monolítico, onde, em quase todos os casos de microsserviços bem-sucedidos, foram quando um software monolítico ficou muito complexo e grande.

Um dos desafios arquitetônicos é executar a decomposição de um monólito para um ecossistema de microsserviços (ZHAMAK DEGHANI, 2018). Morrison (2015) realça que Netflix e também Gilt, PayPal e Condé Nast são consideradas organizações com capacidade de escalar sites de alto volume. Elas estão usando abordagem mais modulares e flexíveis, baseadas em arquiteturas de microsserviços, com o objetivo de acrescentar novas funcionalidades e alterar antigas com mais facilidade. Folwer (2015) destacou que nos últimos anos a Netflix passou por esse grande processo de reformulação de seus sistemas ao longo dos últimos anos.

Na entrevista feita por Werner Vogels, para revista *acmqueue* em junho de 2006, Obidos era uma aplicação monolítica que foi migrada para microsserviço. Ele afirmou que passou por um período de introspecção para permitir a construção isolamento dos serviços, já que eles foram um dos primeiros a adotar essa abordagem. O gateway da amazon.com chamará mais de 100 serviços para coletar os dados e construir a página para o usuário.

De acordo com Fowler e James (2014) existem algumas consequências para a área de operações e seria necessário garantir três competências básicas para o processo de decisão de migração:

- Provisionamento rápido, pois, existe a facilidade de abrir um novo servidor em questão de horas.
- Monitoramento Básico, pois é necessário para detectar problemas sérios rapidamente.
- Implantação rápida de aplicativos, tem que levar em consideração que haverão muitos serviços para que serem implantados tanto no ambiente de teste quanto em produção.
- Cultura Devops que passa a ser um recurso muito importante na mudança organizacional, pois é a colaboração entre desenvolvedor e operações. Também é importante garantir que possa reagir rapidamente quando o monitoramento indicar um problema.

Fowler (2015) salientou que microsserviços possuem três custos que devem ser levados em consideração, um seria em relação à distribuição, ele alega que acabam sendo mais difíceis de programar, visto que as chamadas remotas são lentas e possuem mais risco de falha. Outro seria a consistência eventual, pois para mantê-la forte é extremamente difícil para um sistema distribuído e sendo assim, para usar esta arquitetura deve-se realizar também este gerenciamento. E por último levantou sobre a complexidade operacional, onde é necessário uma equipe de operações madura para gerenciar muitos serviços, que estão sendo atualizados continuamente.

No artigo de Dragoni, et al (2017) apresenta um estudo de caso do mundo real onde demonstra como a escalabilidade é afetada positivamente pela re-implementação de uma arquitetura monolítica em microsserviços. A comparação não foi realizada com métricas quantitativas, pois o principal argumento seria a escalabilidade. Foi realizada uma investigação completa de como implementar uma arquitetura de microsserviços escalável e salienta o que o processo de transição é bastante complexo. Ao comparar os dois projetos, visualiza-se que microsserviços possui melhor escalabilidade e proporciona melhores soluções para os principais problemas que são causados pelo monolítico.

Alessandra Levcovitz, Ricardo Terra e Marco Tulio Valente (2016) descrevem uma técnica para identificar e definir microsserviços em um ambiente monolítico, para um sistema empresarial. Ao definir os serviços que mais se adequam ao microsserviços reduziu o tamanho do sistema original. Em relação a implementação do sistema, os serviços foram desenvolvidos e implantados de forma independente e sem dependência de tecnologia. No estudo identificaram subsistemas que eram bons candidatos para migrar para microsserviços. Segue exemplos de alguns cenários que exigiram uma considerável contribuição de esforço adicional:

- Subsistemas que compartilham a mesma tabela do banco de dados;

- Microserviços representa uma operação que está sempre em meio de outra operação;
- Operações comerciais de escopo de transação, como exemplo, transferência de dinheiro de uma conta corrente para uma conta poupança;
- Informa que é possível migrar alguns módulos do sistema para arquitetura de microserviços que podem coexistir em uma solução de sistema monolítica, portanto, é possível lucrar com arquitetura de microserviços desenvolvendo serviços e implantando-os independentemente de tecnologia, sem a necessidade de migrar o sistema todo.

Segundo Machado (2017) um dos obstáculos para adoção da arquitetura baseada em microserviços pode possuir um maior custo na entrada, comparado com o custo de um sistema monolítico. Ele realça que para ter mais atenção no aumento da complexidade da coordenação, comunicação entre os microserviços e na governança, focando bastante no planejamento. Conseqüentemente existe a necessidade de um esforço maior para a gestão dos microserviços que deve ser levado em consideração.

Segundo Valentina Lenarduzzi, Francesco Lomio, Nyyti Saarimaki e Davide Taibi (2019) . Realizou uma análise durou mais de quatro anos, dois anos antes e dois anos depois a migração, em um projeto que tinha 12 anos. Eles salientam que a migração de sistemas monolíticos para microserviços requer uma refatoração profunda do sistema. Levantaram que essa migração tem grande impacto econômico e para acelerar a migração, as empresas tendem a adiar várias atividades durante esse processo. Foi realizado o monitoramento do sistema monolítico legado para microserviços, e puderam analisar alterações no código antes e depois da migração e foram extraídos cinco processos. Em relação aos serviços migrados, os desenvolvedores analisaram que na arquitetura de microserviços era menor a complexidade de manutenção e maior velocidade de desenvolvimento e também relataram que houve um aumento da liberdade da equipe. Os desenvolvedores também informam que no processo inicial de migração ocorreu um esforço extra devido ao re-desenvolvimento do sistema.

**Tabela 3.1-** Tabela comparativa das principais características da arquitetura monolítica e microsserviços

<b>Característica</b>	<b>Monolítica</b>	<b>Microsserviços</b>
Implementação contínua	Mais complexa	Mais Simples
Reusabilidade	Mais complexa	Mais simples
Manutenção	Mais complexa	Mais simples
Escalabilidade	Mais complexa	Mais simples
Implantação contínua	Mais complexa	Mais simples
Planejamento	Menos tempo	Mais tempo
Acoplamento	Mais Alto	Mais Baixo
Integração contínua	Mais complexo	Mais simples
Teste	Mais simples	Mais eficiente
Disponibilidade	Mais complexo	Mais eficiente
Tecnologias distintas	Mais complexo	Mais simples

Fonte: Autoria própria

### 3.3 Elaboração das Perguntas do Questionário

Inicialmente, serão realizadas perguntas para análise quantitativa e qualitativa dos tipos de usuários. Para que seja possível identificar os interessados na pesquisa. Os cargos que ficaram na lista serão desenvolvedores, arquitetos, analistas de sistemas e estagiário que são os públicos alvos da pesquisa. A pergunta ficará no formato abaixo:

1. Qual é o seu cargo em relação ao sistema em análise?
  - a. Estagiário
  - b. Desenvolvedor

- c. Analista
- d. Arquiteto
- e. Devops
- f. Outros

Serão realizadas perguntas para análise quantitativa e qualitativa para identificar quais tipos de organizações necessitam ou não utilizar a arquitetura microsserviços. A pergunta ficará no formato abaixo:

- 2. Qual o tamanho da sua empresa?
  - a. 1 a 20 colaboradores
  - b. 20 a 50 colaboradores
  - c. 50 a 100 colaboradores
  - d. Mais de 100 colaboradores

As elaborações das perguntas serão divididas em tópicos baseados nas características levantadas na tabela 3.1, demonstrada anteriormente.

Segundo Machado (2017) informa que ao contrário da arquitetura monolítica que possui um único banco de dados e centraliza todas as informações necessárias para aplicação nele. A arquitetura de microsserviços precisa de mais planejamento, porém, o mesmo salienta que a desvantagem da monolítica é no momento do escalonamento e na replicação dos dados, devido ao alto acoplamento. Morrison (2015) também cita que os microsserviços proporcionam maior modularidade, baixo acoplamento e diminuição de dependência a fim de simplificar a integração. Com objetivo de eliminar dependências e permitir testes e implantação rápidos de alterações de código.

- 3. Numa escala de 0 a 5, considerando que 0 não existe a possibilidade e 5 existe a possibilidade; No contexto do seu sistema, há a possibilidade de promover uma mudança arquitetural para uma arquitetura distribuída e

escalável, ou seja, há recursos humanos com autonomia, tempo e conhecimento para planejar essa mudança?

4. Acoplamento em um sistema de software é entendido como a relação/ligação de um componente (função, método e/ou classe) com outros componentes do sistema, de forma a atender uma determinada funcionalidade. Considerando o conceito de acoplamento em um sistema: numa escala de 0 a 5, onde 0 representa um sistema não acoplado e 5 um sistema altamente acoplado, você considera o sistema que vocês está utilizando para responder esse questionário acoplado?

O segundo tópico levantado é sobre a necessidade de escalabilidade. Pois, como microsserviços funciona com mais independência e autonomia, um dos benefícios de utilizar o microsserviços é relacionado a dividir o aplicativo em unidades pequenas, pois proporciona que cada parte seja independentemente implantada e facilita a escalabilidade da aplicação (STOIBER, 2017).

Conforme Zhamak Dehghani (2018) os que escolhem realizar esse processo de migração para esta arquitetura é justamente devido a aspirações como aumentar a escala da operação do sistema. A mesma informa que até empresas grandes como a Netflix passaram por esse processo de migração para permitir que possam acrescentar novas funcionalidades e alterar antigas com mais agilidade. Pois segundo Machado (2017) e Richardson (2019) as abordagens de escalabilidade dos sistemas monolíticos são limitadas e não podem escalar os componentes interdependentes.

5. Numa escala de 0 a 5, onde 0 significa não atende e 5 significa atende perfeitamente; O seu sistema atende de forma satisfatória (satisfatória: performance adequada) a demanda atual de processamento? Considere a demanda atual e a demanda futura (caso haja a expectativa de um aumento de carga no sistema).



Conforme Martin Fowler e James Lewis (2014) o sistema em microsserviços pode ser escrito em diferentes linguagens de programação e usar diferentes tecnologias de armazenamento de dados. Richardson (2014) levanta que o monolítico dificulta a adoção de novas tecnologias.

Machado (2017) realça que o monolítico possui uma falta de flexibilidade quando se tratar de mudança de tecnologia, pois os desenvolvedores ficam amarrados a tecnologia que escolheram no início do projeto.

6. Numa escala de 0 a 5, onde 0 representa que não teria benefício e 5 representa que teria muito benefício; Haveria algum benefício concreto se seu sistema fosse dividido em partes distintas e estas partes utilizassem tecnologias diferentes (outras linguagens, frameworks e etc)?

O terceiro tópico considerado foi em relação ao alto nível de manutenibilidade do sistema em estudo. Segundo Toby Clemson (2014) a capacidade de implantar e manter de forma independente cada componente possibilita paralelizar a equipe de desenvolvimento, assim, facilitando a manutenção do sistema. Fowler (2015) alega que para sistemas mais complexos usando microsserviços o custo de manutenção é bem menor comparado a sistema um monolítico. Existe também dificuldade de colocar as alterações em produção em sistemas monolíticos, segundo Machado (2017), pois as funções são muito interdependentes e acopladas, conforme citado na seção 2.4.

7. Numa escala de 0 a 5, onde 0 significa que para identificar um erro é simples e 5 significa que é bastante complexo. O sistema em análise tem qual nível de manutenibilidade?

8. Numa escala de 0 a 5, considere que 0 representa ser muito fácil desenvolver e 5 representa muito complexo desenvolver; Atualmente é muito trabalhoso desenvolver novas funcionalidades no seu sistema?

A proposta de um estímulo a reutilização de componentes e padronizá-los é definido como um padrão arquitetural. A maneira de padronização é feita através do uso de rotineiro de soluções existentes que podem ser realizados por meio de alguma referência comum das atividades que posteriormente deverão ser executadas (BASS, CLEMENTS & KAZMAN, 2003). Os componentes são interdependentes e planejados para não serem acoplados segundo Machado (2017). Então, a mesma possibilita projetar um sistema que facilita na manutenção e reutilização do código.

9. Numa escala de 0 a 5, onde 0 representa que nenhuma parte e 5 representa pelo menos 60% do código do sistema poderia ser reutilizado. No seu sistema existem muitas partes que poderiam ser reutilizadas?

No segundo estudo feito por Odravison Amaral e Marcus Carvalho (2017), arquitetura de microsserviços se aplica a soluções que necessitam escalonamento e facilidade de **integrar** os dados. Também é ideal para casos de alta produtividade do time de desenvolvimento, pois segundo o estudo analisado na pesquisa, o isolamento das funcionalidades de uma aplicação é um dos principais fatores que facilitam a produtividade. Pois um projeto com crescimento contínuo necessita de uma melhor integração entre sistemas e serviços (J. THÖNES, 2015).

No estudo feito por Odravison Amaral e Marcus Carvalho (2017) informa que a arquitetura de microsserviços facilita em relação a integração dos dados.

10. Numa escala de 0 a 5, considerando que 0 representa nenhuma necessidade e 5 muita necessidade; Existe a necessidade que seu sistema se integre com outros sistemas?

Segundo Fowler (2015) é comum de migração de um monolítico para arquitetura em microsserviços. O mesmo argumenta que por mais que exista a certeza que o sistema será grande o suficiente para usar microsserviços, não é interessante iniciar um novo projeto com tal arquitetura. Stoiber (2017) afirma que um sistema em microsserviços pode ser implementado por diferentes equipes e podendo também ser testada individualmente, facilitando a implementação dos módulos.

Conforme Toby Clemson (2014) às arquiteturas de microsserviços fornecem mais opções de como e onde testar. Embora esse processo de tomada de decisão ainda seja possível em uma arquitetura monolítica, a adição de limites mais claros e bem definidos facilita a visualização dos componentes e do tratamento isolado seu sistema. Um sistema em microsserviços pode ser testado individualmente (STOIBER, 2017). Os serviços são menores e mais fáceis de testar, porém, é necessário investir tempo no teste de integração dos sistemas (RICHARDSON, 2019).

11. Numa escala de 0 a 5, considere que 0 representa que não faz sentido e 5 muito sentido; Faz sentido no contexto do seu sistema ter equipes trabalhando em módulos separados?

Segundo Toby Clemson (2014) a capacidade de implantar e manter de forma independente cada componente possibilita paralelizar a equipe de desenvolvimento. Fowler (2014) enfatiza que existem características comuns em torno da organização, implantação contínua automática, da capacidade comercial, camadas de interface do sistema com mais inteligência. Já quando se trata de aplicações monolíticas a implantação contínua são mais difícil segundo Richardson (2019).

Serviço pode ser implantado independentemente de outros serviços. Em caso dos desenvolvedores responsáveis por um serviço precisarem implantar uma alteração que seja local para esse serviço, eles não precisarão coordenar com os outros desenvolvedores. Eles podem simplesmente implantar suas alterações. Uma arquitetura de microsserviços torna a implantação contínua viável. Cada serviço pode ser implantado no hardware mais adequado aos seus requisitos de recursos, diferente de uma arquitetura monolítica de acordo com Richardson (2014).

12. Numa escala de 0 a 5, considere que 0 representa que o processo não é trabalhoso e 5 é bastante trabalhoso; O processo de disponibilização de novas versões do seu sistema é um processo trabalhoso?

Com microsserviços é que o sistema passa a possuir um melhor isolamento de falhas. Se um serviço for comprometido, exemplo um vazamento de memória, apenas esse serviço será afetado e os outros continuarão disponíveis, uma característica de disponibilidade. Em comparação, um componente comprometido de uma arquitetura monolítica pode afetar todo sistema. Por possuir um melhor isolamento de falhas. Se um serviço for comprometido, exemplo um vazamento de memória, apenas esse serviço será afetado e os outros continuarão disponíveis. Em comparação, um componente comprometido de uma arquitetura monolítica pode afetar todo sistema (Richardson, 2019).

13. Considere que 0 representa que não seria interessante e 5 que seria muito interessante para seu sistema; Há necessidade de monitorar partes específicas do seu sistema?

14. Considere que 0 representa que não seria interessante e 5 que seria muito interessante para seu sistema; Seria interessante se fosse possível substituir uma dessas partes, caso haja a necessidade de prover uma

funcionalidade mais robusta para atender uma eventual sobrecarga do sistema?

#### **4. Aplicação do Questionário em Aplicações Monolíticas**

O questionário manteve-se disponível do dia 7 de outubro de 2019 ao dia 18 de novembro de 2019 na plataforma SurveyMonkey. As respostas possuem pontuação na escala 0, 1, 2, 3, 4, 5 que representam:

- Escala 0: A característica em questão não é interessante para o sistema, portanto não seria necessário migrar o sistema para microsserviços.;
- Escala 1: A característica em questão é pouco interessante para o sistema, portanto o sistema possui pequena tendência migrar para microsserviços;
- Escala 2: A característica em questão possui uma leve relevância para o sistema, portanto o sistema possui uma leve tendência de migrar para microsserviços;

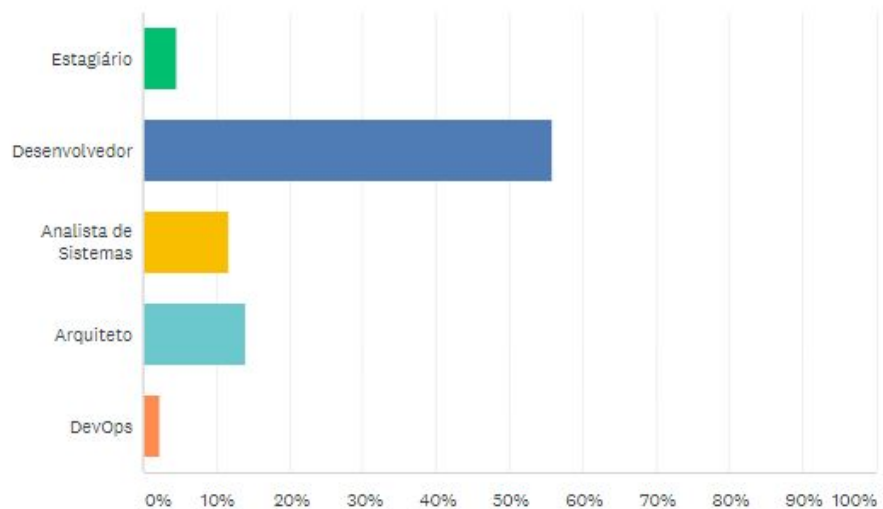
- Escala 3: A característica em questão possui uma média relevância para o sistema, portanto o sistema tem uma tendência mediana de migrar para microsserviços;
- Escala 4: A característica em questão possui uma grande relevância para o sistema, portanto o sistema possui uma grande tendência de migrar para microsserviços;
- Escala 5: A característica em questão possui uma forte relevância para o sistema, portanto o sistema possui uma forte tendência de migrar para microsserviços;

Em relação aos cargos dos participantes, foram obtidos os seguintes resultados:

**Figura 4.1-** Análise das respostas dos cargos

## Qual é o seu cargo em relação ao sistema em análise?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

Observa-se no gráfico que 4,65% dos pesquisados estão no cargo de estagiário, o cargo de desenvolvedor possui 55,81% das respostas, correspondendo a maioria das respostas, o cargo de Análise de Sistemas corresponde a 11,63% das respostas. O cargo de Arquiteto corresponde a 13,95% das respostas. O cargo de DevOps corresponde a 2,33% das respostas. O cargo de “Outros” corresponde a 11,63% das respostas.

Em relação às respostas que foram obtidas como “Outros” se tratavam 2,33% de Estudantes, 2,33% eram Analistas de Suporte, 2,33% Gerentes, 2,33% da área de Infraestrutura e 2,33% Analistas de BI.

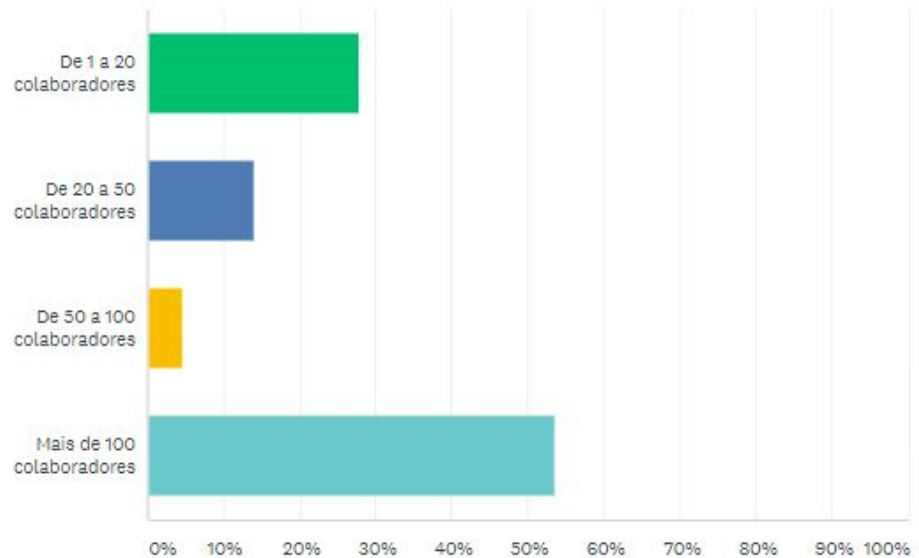
Portanto, observa-se que a maioria dos pesquisados são desenvolvedores, seguidos de analistas e arquitetos de softwares, onde correspondem os papéis que mais se envolvem no processo de tomada de decisão de migração e podem utilizar o questionários como sugestão de melhoria para a organização.

Em relação ao tamanho da empresa, foram obtidos os seguintes dados:

**Figura 4.2-** Análise das respostas sobre tamanho da empresa

### Qual o tamanho da sua empresa?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

Em relação ao gráfico 4.2, foi levantado que entre 1 a 20 colaboradores foram correspondentes a 27,91% das respostas. De 20 a 50 colaboradores foram correspondentes a 13,95%. De 50 a 100 colaboradores foram correspondentes a 4,65%. Com mais de 100 colaboradores foram correspondentes a 53,49%, sendo a maioria das respostas. Portanto, observa-se que a maioria dos pesquisados são de empresas de mais de 100 colaboradores, representando empresas de maior porte e que possuem uma maior estrutura organizacional.

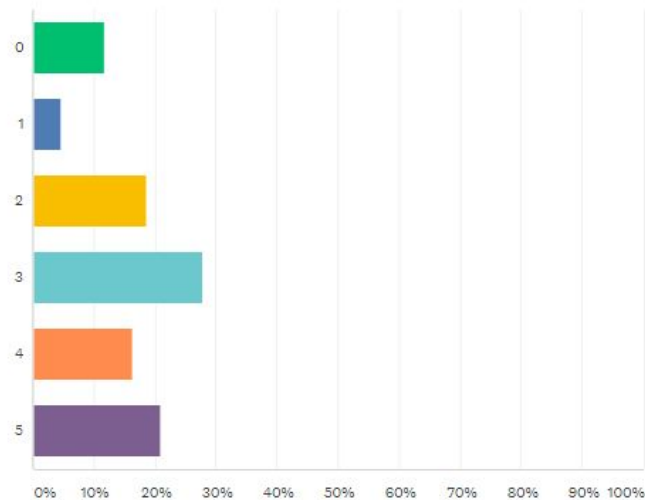
Em relação a viabilidade de recursos, foram levantados os seguintes dados:

**Figura 4.3-** Análise das respostas sobre viabilidade de recursos



Em relação a viabilidade de recursos ...Numa escala de 0 a 5, considerando que 0 não existe a possibilidade e 5 existe a possibilidade; No contexto do seu sistema, há a possibilidade de promover uma mudança arquitetural para uma arquitetura distribuída e escalável, ou seja, há recursos humanos com autonomia, tempo e conhecimento para planejar essa mudança?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

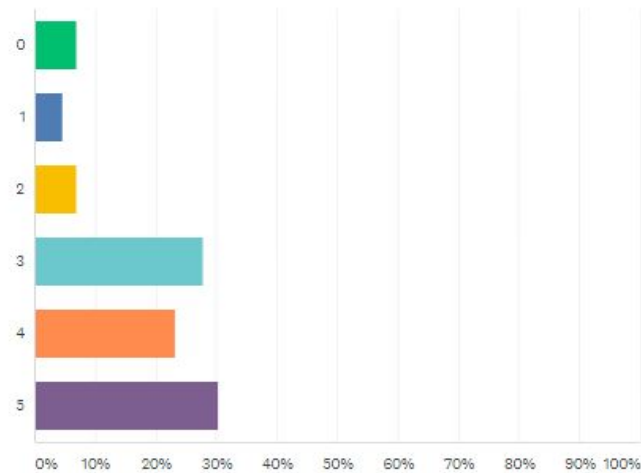
No gráfico 4.3, observa-se que obtiveram 11,63% respostas com escala 0. Com a escalada 1 corresponde a 4,65% das respostas. Com a escala 2 corresponde a 18,60% das respostas. Com a escala 3 corresponde a 27,91% das respostas. Com a escala 4 corresponde a 16,28% das respostas. Com a escala 5 corresponde a 20,93% das respostas. Portanto, observa-se que a maior frequência dos pesquisados correspondem às escalas 3, 4 e 5, onde representa a existência da possibilidade de viabilidade de recursos, caracterizando uma tendência a migração do sistema para microsserviços nesta questão.

Em relação ao nível de acoplamento do sistema, foram levantados os seguintes dados:

**Figura 4.4-** Análise de respostas sobre o nível de acoplamento do sistema

Em relação ao nível de acoplamento do sistema...Acoplamento em um sistema de software é entendido como a relação/ligação de um componente (função, método e/ou classe) com outros componentes do sistema, de forma a atender uma determinada funcionalidade. Considerando o conceito de acoplamento em um sistema: numa escala de 0 a 5, onde 0 representa um sistema não acoplado e 5 um sistema altamente acoplado, você considera o sistema que vocês está utilizando para responder esse questionário acoplado?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

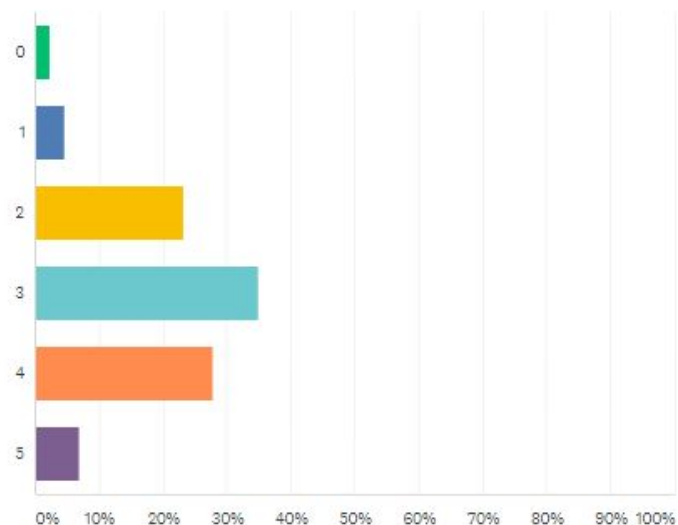
No gráfico 4.4, observa-se que obtiveram 6,98% respostas com escala 0. Com a escalada 1 corresponde a 4,65% das respostas. Com a escala 2 corresponde a 6,98% das respostas. Com a escala 3 corresponde a 27,91% das respostas. Com a escala 4 corresponde a 23,26% das respostas. Com a escala 5 corresponde a 30,23% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3, 4 e 5, onde representa a existência de um alto acoplamento no sistema, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Em relação a escalabilidade do sistema, foram levantados os seguintes dados:

**Figura 4.5-** Análise das respostas sobre escalabilidade do sistema

Em relação à escalabilidade do sistema ...Numa escala de 0 a 5, onde 0 significa não atende e 5 significa atende perfeitamente; O seu sistema atende de forma satisfatória (satisfatória: performance adequada) a demanda atual de processamento? Considere a demanda atual e a demanda futura (caso haja a expectativa de um aumento de carga no sistema).

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

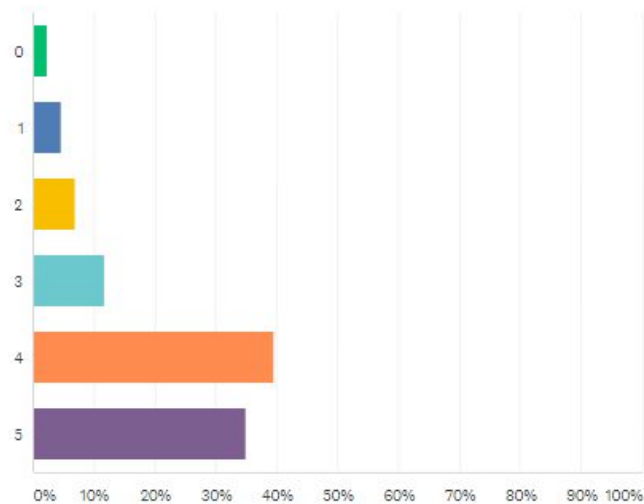
No gráfico 4.5, observa-se que obtiveram 2,33% respostas com escala 0. Com a escalada 1 corresponde a 4,65% das respostas. Com a escala 2 corresponde a 23,26% das respostas. Com a escala 3 corresponde a 34,88% das respostas. Com a escala 4 corresponde a 27,91% das respostas. Com a escala 5 corresponde a 6,98% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 2, 3 e 4, onde representa que o sistema atual atende as demandas atuais e futuras de escalabilidades, caracterizando uma leve tendência a migração do sistema para microsserviços nesta questão

Em relação ao uso de tecnologias distintas em um mesmo projeto, foram levantados os seguintes dados:

**Figura 4.6-** Análise das respostas sobre o uso de tecnologias distintas no mesmo projeto

Em relação ao uso de tecnologias distintas em um mesmo projeto ...Numa escala de 0 a 5, onde 0 representa que não teria benefício e 5 representa que teria muito benefício; Haveria algum benefício concreto se seu sistema fosse dividido em partes distintas e estas partes utilizassem tecnologias diferentes (outras linguagens, frameworks e etc)?

Answered: 43 Skipped: 0



Fonte:(Relatório extraído na plataforma SurveyMonkey, 2019).

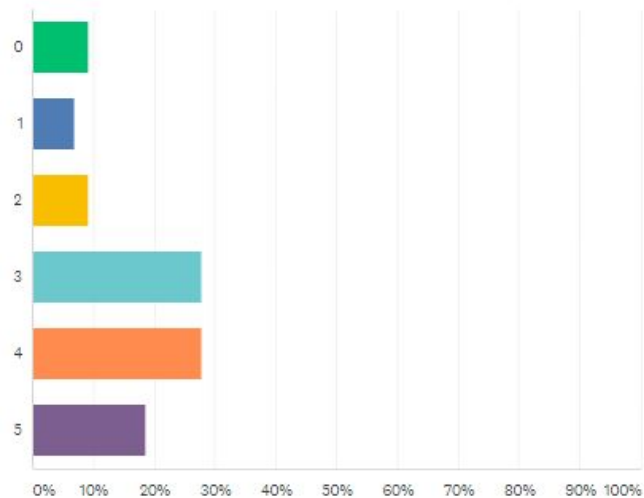
No gráfico 4.6, observa-se que obtiveram 2,33% respostas com escala 0. Com a escalada 1 corresponde a 4,65% das respostas. Com a escala 2 corresponde a 6,98% das respostas. Com a escala 3 corresponde a 11,63% das respostas. Com a escala 4 corresponde a 39,53% das respostas. Com a escala 5 corresponde a 34,88% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3 e 4, onde representa que seria interessante para o sistema dividir o sistema em partes distintas e com diferentes tecnologias, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Em relação a manutenção do sistema, foram levantados os seguintes dados:

**Figura 4.7-** Análise das respostas sobre identificar erros na manutenção do sistema

Em relação a manutenção do sistema...Numa escala de 0 a 5, onde 0 significa que para identificar um error é simples e 5 significa que é bastante complexo. O sistema em análise tem qual nível de manutenibilidade?

Answered: 43 Skipped: 0



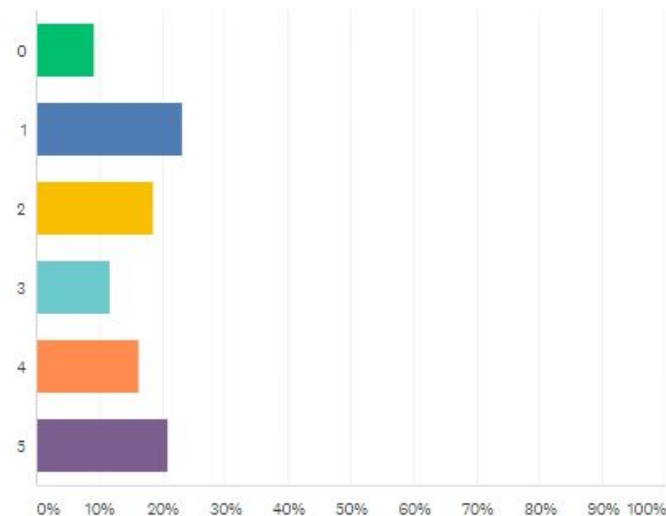
Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

No gráfico 4.7, observa-se que obtiveram 9,30% respostas com escala 0. Com a escalada 1 corresponde a 6,98% das respostas. Com a escala 2 corresponde a 9,30% das respostas, sendo ao todo 4 respostas. Com a escala 3 corresponde a 27,91% das respostas. Com a escala 4 corresponde a 27,91% das respostas. Com a escala 5 corresponde a 18,60% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3 e 4, onde representa que no sistema atual é fácil encontrar erros, caracterizando uma pequena tendência a migração do sistema para microsserviços nesta questão.

**Figura 4.8-** Análise das respostas sobre complexidade manutenção do sistema

Em relação a manutenção do seu sistema...Numa escala de 0 a 5, considere que 0 representa ser muito facil desenvolver e 5 representa muito complexo desenvolver; Atualmente é muito trabalhoso desenvolver novas funcionalidades no seu sistema?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

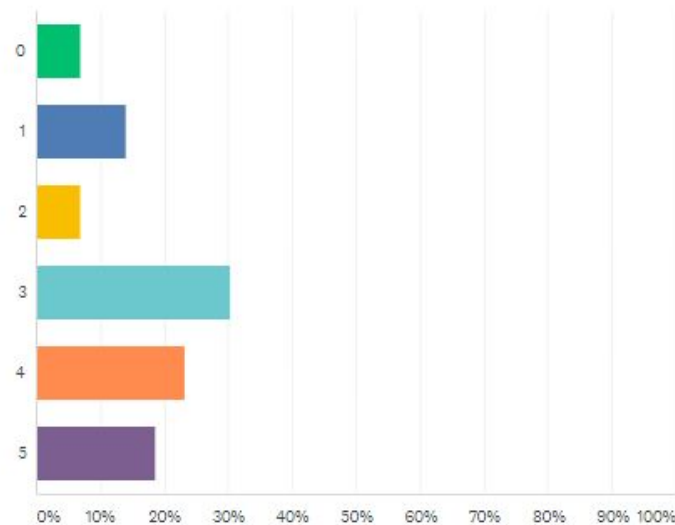
No gráfico 4.8, observa-se que obtiveram 9,30% respostas com escala 0. Com a escalada 1 corresponde a 23,26% das respostas. Com a escala 2 corresponde a 18,60% das respostas. Com a escala 3 corresponde a 11,63% das respostas. Com a escala 4 corresponde a 16,28% das respostas. Com a escala 5 corresponde a 20,93% das respostas. Portanto, observa-se que os pesquisados não tiveram uma tendência de respostas, desta forma, fica inconclusivo a tendência de migração do sistema nesta questão.

Em relação a reutilização de partes do sistema, foram levantados os seguintes dados:

**Figura 4.9-** Análise das respostas sobre reutilização de partes do sistema

Em relação a reutilização de partes do sistema...Numa escala de 0 a 5, onde 0 representa que nenhuma parte e 5 representa pelo menos 60% do código do sistema poderia ser reutilizado. No seu sistema existem muitas partes que poderiam ser reutilizadas?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

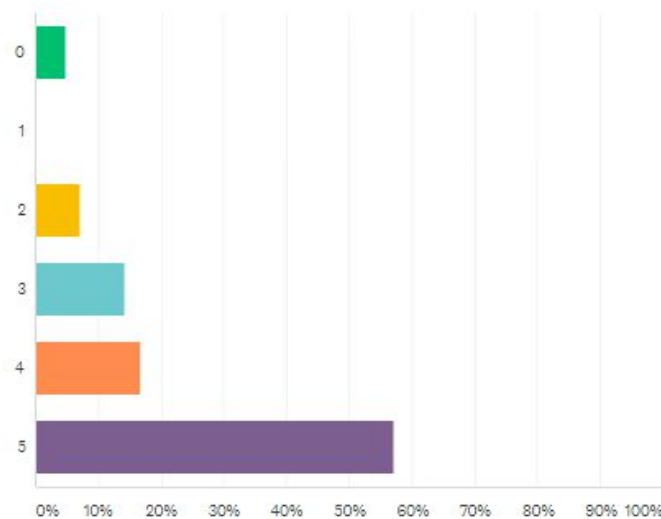
No gráfico 4.9, observa-se que obtiveram 6,68% respostas com escala 0. Com a escalada 1 corresponde a 13,95% das respostas. Com a escala 2 corresponde a 6,98% das respostas. Com a escala 3 corresponde a 30,23% das respostas. Com a escala 4 corresponde a 23,26% das respostas. Com a escala 5 corresponde a 18,60% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3, 4 e 5, onde representa que no sistema atual possui cerca de 60% que podem ser reutilizados, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Em relação a integração do sistema, foram levantados os seguintes dados:

**Figura 4.10-** Análise das respostas sobre integração do sistema

Em relação a integração de sistemas...Numa escala de 0 a 5, considerando que 0 representa nenhuma necessidade e 5 muita necessidade; Existe a necessidade que seu sistema se integre com outros sistemas?

Answered: 42 Skipped: 1



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019).

No gráfico 4.10, observa-se que obtiveram 4,76% respostas com escala 0. Com a escalada 1 corresponde a 0,00% das respostas. Com a escala 2 corresponde a 7,14% das respostas. Com a escala 3 corresponde a 14,29% das respostas. Respostas com a escala 4 corresponde a 16,67% das respostas. Com a escala 5 corresponde a 57,14% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 5, onde representa que no sistema existe uma grande necessidade de integração com outros sistemas, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Em relação a implementação de módulos no sistema, foram levantados os seguintes dados:

**Figura 4.11-** Análise das respostas sobre implementação de módulos no sistema

Em relação a implementação de módulos no sistema...Numa escala de 0 a 5, considere que 0 representa que não faz sentido e 5 faz muito sentido; Faz sentido no contexto do seu sistema ter equipes trabalhando em módulos separados?

Answered: 41 Skipped: 2



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

No gráfico 4.11, observa-se que obtiveram 7,32% respostas com escala 0. Com a escalada 1 corresponde a 4,88% das respostas. Com a escala 2 corresponde a 12,20% das respostas. Com a escala 3 corresponde a 17,07% das respostas. Respostas com a escala 4 corresponde a 17,07% das respostas. Com a escala 5 corresponde a 41,46% das respostas. Nestas respostas um participante não respondeu esta pergunta. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3, 4 e 5, onde representa que no sistema faz muito sentido ter equipes trabalhando em módulos separados, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Em relação a disponibilidade de novas versões de sistema, foram levantados os seguintes dados:

**Figura 4.12-** Análise das respostas sobre disponibilização de novas versões

Em relação a disponibilização de novas versões...Numa escala de 0 a 5, considere que 0 representa que o processo não é trabalhoso e 5 é bastante trabalhoso; O processo de disponibilização de novas versões do seu sistema é um processo trabalhoso?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

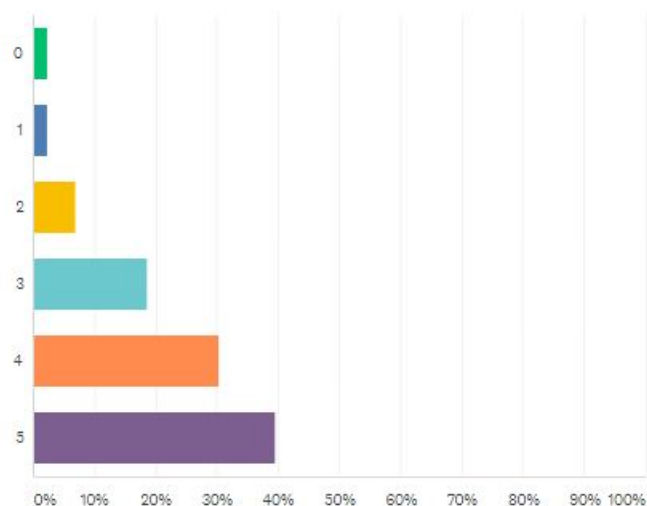
No gráfico 4.12, observa-se que obtiveram 6,98% respostas com escala 0. Com a escalada 1 corresponde a 11,63% das respostas. Com a escala 2 corresponde a 11,63% das respostas. Com a escala 3 corresponde a 23,26% das respostas. Com a escala 4 corresponde a 30,23% das respostas. Com a escala 5 corresponde a 16,28% das respostas. Nestas respostas dois participantes não responderam esta pergunta. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 3 e 4, onde representa que no sistema atual é trabalhoso disponibilizar novas versões, caracterizando uma grande tendência a migração do sistema para microsserviços nesta questão.

Em relação a monitoramento e disponibilidade, foram levantados os seguintes dados:

**Figura 4.13-** Análise das respostas sobre monitoramento de partes específicas do sistema

Em relação ao monitoramento e disponibilidade...Considere que 0 representa que não seria interessante e 5 que seria muito interessante para seu sistema; Há necessidade de monitorar partes específicas do seu sistema?

Answered: 43 Skipped: 0



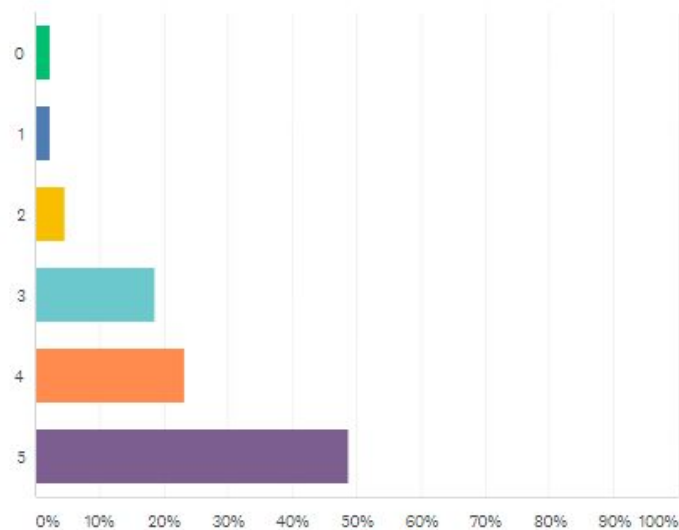
Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

No gráfico 4.13, observa-se que obtiveram 2,33% respostas com escala 0. Com a escalada 1 corresponde a 2,33% das respostas. Com a escala 2 corresponde a 6,98% das respostas. Com a escala 3 corresponde a 18,60% das respostas. Com a escala 4 corresponde a 30,23% das respostas. Com a escala 5 corresponde a 39,53% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 4 e 5, onde representa que seria bastante interessante monitorar partes específicas do sistema, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

**Figura 4.14-** Análise das respostas sobre disponibilidade do sistema

Considere que 0 representa que não seria interessante e 5 que seria muito interessante para seu sistema; Seria interessante se fosse possível substituir uma dessas partes, caso haja a necessidade de prover uma funcionalidade mais robusta para atender uma eventual sobrecarga do sistema?

Answered: 43 Skipped: 0



Fonte: (Relatório extraído na plataforma SurveyMonkey, 2019)

No gráfico 4.14, observa-se que obtiveram 2,33% respostas com escala 0. Com a escalada 1 corresponde a 2,33% das respostas. Com a escala 2 corresponde a 4,65% das respostas. Com a escala 3 corresponde a 18,60% das respostas. Com a escala 4 corresponde a 23,26% das respostas. Com a escala 5 corresponde a 48,84% das respostas. Portanto, observa-se que a maioria dos pesquisados correspondem às escalas 4 e 5, onde representa que seria bastante interessante substituir partes do sistema, em caso de uma sobrecarga, caracterizando uma forte tendência a migração do sistema para microsserviços nesta questão.

Segue abaixo um resumo geral dos pesquisados:

**Figura 4.15-** Resumo da apuração das respostas do questionário



Fonte: (Autoria própria, 2019).

Para realizar avaliação se o sistema do pesquisado deve migrar ou não para microserviços, as respostas desta pesquisa possuem uma pontuações diferentes, que correspondem aos seguintes pesos: “0” para a escala 0, “1” para escala 1, “2” para escala 2, “3” para escala 3, “4” para escala 4 e “5” para escala 5.

Baseado no somatório geral dos pesos, foram considerados 3 ranges para classificar a sugestão de migrar ou não migrar para microserviços, onde os resultados que, no conjunto de critérios avaliados para cada participante, corresponde até 45% da pontuação geral, foram classificados como “indicado a continuarem em uma estrutura monolítica”. No segundo range foram os resultados entre 46% a 70% da pontuação geral, onde foram classificados como “provavelmente a estrutura mais indicada seja uma estrutura baseada em microserviços”, mas neste caso foi indicado um estudo mais aprofundado. O terceiro range foram os resultados entre 71% a 100% da pontuação geral, foram classificados “indicado fortemente que os sistemas sejam migrados”.

No gráfico 4.5, observa-se que obtiveram 2,33% respostas com escala até 45%, indicando-os a continuarem em uma estrutura monolítica.

Com base nas características analisadas no questionário que correspondem entre 46% a 70%. Identificam-se 55,81% dos pesquisados, onde provavelmente a estrutura mais indicada seja uma estrutura baseada em microsserviços. Foi recomendado uma análise mais aprofundada de forma a avaliar os benefícios obtidos com a migração dessa estrutura.

Observa-se que entre 71% a 100% de compatibilidade com as características de microsserviços, foram identificados 41,86% dos pesquisados, onde foi indicado fortemente que os sistemas sejam migrados de uma estrutura monolítica para uma estrutura baseada em microsserviços.

## **5. Conclusão**

Observa-se que o processo de migração de um sistema de arquitetura monolítica para microsserviços pode-se ter grande impacto para organização, pois

existem fatos como grande investimento em planejamento devido ao maior monitoramento dos serviços, análise para refatoração profunda e para acelerar a migração, é necessário adiar algumas atividades de manutenção e melhorias. A proposta deste trabalho pode auxiliar o apoio à decisão nesta difícil escolha se deve migrar seu sistema de arquitetura monolítica para microsserviços.

Na primeira fase do embasamento teórico, foram identificadas as características fortes dos sistemas de microsserviços, como, escalonamento, manutenibilidade, disponibilidade, reuso de código e uso de tecnologias distintas. Mas vale salientar que esta arquitetura possui uma grande fase de planejamento e autores não indicam para projetos com pouca complexidade, tendo em vista que algumas destas características podem ser tratadas com outras abordagens, não necessariamente com a migração do sistema.

Baseando-se no resultado geral da pesquisa, percebe-se que a maioria dos pesquisados possuem características que recomenda-se a migração para microsserviços. No resultado da pesquisa, foram dadas sugestões de como o pesquisado faria a migração fornecendo dicas e instruções com indicações de artigos sobre sistemas que sofreram tal procedimento.

Com os dados deste questionário, os arquitetos e desenvolvedores podem utilizar este trabalho como apoio a decisão estratégia para organização, através de embasamento teórico levantado sobre as principais características e exemplos citados na literatura sobre projetos que sofreram migração de monolítico para microsserviço.

## **5.1. Trabalhos Futuros**

Para trabalhos futuros foram considerados alguns pontos, um deles seria aumentar mais a amostragem de pesquisados, liberando a pesquisa por mais tempo e com mais pessoas de projetos distintos, assim, seria possível realizar uma análise quantitativa mais diversificada. Outro ponto interessante seria realizar



estudos de casos sobre os projetos que decidiram migrar, desta forma, pode-se analisar quais os benefícios que os mesmos possuíam com a migração, então, será possível aprimorar as margens de indicação de qual tipo de arquitetura seria mais adequada para o pesquisado.



## Referências Bibliográficas

ALESSANDRA LEVCOVITZ, RICARDO TERRA E MARCO TULLIO. **Towards a Technique for Extracting Microservices from Monolithic Enterprise Systems**, 2016.

BABBIE, E. R. **Métodos de pesquisa de survey**. Belo Horizonte: UFMG, 1999. 519 p.

BASS, LEN; CLEMENTS, PAUL; KAZMAN, RICK. **2. ed. Software Architecture in Practice**. Addison-Wesley, 2003.

DI FRANCESCO, P. Architecting microservices. **In 1st International Conference on Software Architecture Workshops (ICSAW)**, pages 224–229, 2017.

FOWLER, M. AND JAMES LEWIS. **Microservices, a definition of this new architectural term**, 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acessado em 22 de janeiro de 2019.

FOWLER, M. **MicroservicePremium**, 2015. Disponível em: <https://martinfowler.com/bliki/MicroservicePremium.html>. Acessado em 03 de agosto de 2019.

FOWLER, M. **MicroservicePrerequisites**, 2015. Disponível em: <https://www.martinfowler.com/bliki/MicroservicePrerequisites.html>. Acessado em 07 de agosto de 2019.

FOWLER, M. **Microservice Trade-Offs**, 2015. Disponível em: <https://www.martinfowler.com/microservices/>. Acessado em 07 de agosto de 2019

FOWLER, M. **MonolithFirst**, 2015. Disponível em: <https://www.martinfowler.com/microservices/> . Acessado em 22 de janeiro de 2019

FOWLER, Martin. **Patterns of Enterprise Architecture**. Addison-Wesley, 2002.

J. THÖNES. "Microservices" in IEEE Software, vol. 32, no. 1, pp. 116-116, Jan.-Feb. 2015.

KRAFZIG, D., BANKE, K. & SLAMA, D. **Enterprise SOA: Service-Oriented Architecture Best Practices**. Indianapolis: Prentice Hall, 2004.

MACHADO, M. G. **Micro Serviços: Qual a diferença para arquitetura monolítica?** , 2017. Disponível em: <https://www.opus-software.com.br/micro-servicos-arquitectura-monolitica/>. Acessado em: 23 de janeiro de 2019

MARTINS, J. C. C. **Técnicas para Gerenciamento de Projetos de Software, Brasport**, cap. A, p. 18, 2007.

MORRISON, Alan. **Agile coding in enterprise IT: Code small and local**, 2015. Disponível em: <http://usblogs.pwc.com/emerging-technology/agile-coding-in-enterprise-it-code-small-and-local/> . Acessado dia 24 de julho de 2019.

NAMIOT, D. **On micro-services architecture** , 2014. Disponível em: [https://www.researchgate.net/publication/268919364\\_Micro-service\\_Architecture\\_for\\_Emerging\\_Telecom\\_Applications](https://www.researchgate.net/publication/268919364_Micro-service_Architecture_for_Emerging_Telecom_Applications). Acessado dia 25 de janeiro de 2019.

NAMIOT, D. E MAFRED SNEPS-SNEPPE. **Micro-service Architecture for Emerging Telecom Applications**, 2014. Disponível em: [https://www.researchgate.net/publication/268919364\\_Micro-service\\_Architecture\\_for\\_Emerging\\_Telecom\\_Applications](https://www.researchgate.net/publication/268919364_Micro-service_Architecture_for_Emerging_Telecom_Applications). Acessado dia 25 de janeiro de 2019.

NICOLA DRAGONI, SCHAHRAM DUSTDAR, STEPHAN T. LARSEN E MANUEL MAZZARA. **Microservices: Migration of a Mission Critical System**, 2017.

ODRAVISON AMARAL E MARCUS CARVALHO. **Arquitetura de Micro Serviços: uma Comparação com Sistemas Monolíticos**. Universidade Federal da Paraíba (UFPB), 2017.



QUEIROZ, P.G. & BRAGA, R.T.V. **Desenvolvimento de Linhas de Produtos de Software com uma Arquitetura Orientada a Serviços**. 13º Workshop de Teses e Dissertações em Engenharia de Software, 2008.

RICHARDSON, C. **Pattern: Microservices Architecture**, 2019. Disponível em: <https://microservices.io/patterns/microservices.html> Acessado em: 24 de Julho de 2019.

RICHARDSON, C. **Microservices: Decomposing Applications for Deployability and Scalability**, 2014. Disponível em: <https://www.infoq.com/articles/microservices-intro/> .Acessado 2 de agosto de 2019.

RICHARDSON, C. **Pattern: Monolithic Architecture**, 2019. Disponível em: <https://microservices.io/patterns/monolithic.html> . Acessado 7 de agosto de 2019.

ROUSE, M. **Monolithic Architecture**, 2016. Disponível em: <https://whatis.techtarget.com/definition/monolithic-architecture> . Acessado em: 23 de janeiro de 2019.

SORDI, J.O., MARINHO, B.L. & NAGY, M. **Benefícios da Arquitetura de Software Orientada a Serviços para as Empresas: Análise da Experiência do ABN AMRO Brasil**. Revista de Gestão da Tecnologia e Sistemas de Informação, 2006.

S. NEWMAN. **“Building Microservices”**. O’Reilly Media, Inc., 2015.

STOIBER, M. **Build your first Node.js microservice**, 2017. Disponível em: <https://mxstbr.blog/2017/01/your-first-node-microservice/> . Acessado em 23 de janeiro de 2019.

TOBY CLEMSON. **Testing Strategies in a Microservice Architecture**, 2014. Disponível em: <https://martinfowler.com/articles/microservice-testing/> . Acessado em 13 de junho de 2019

VALENTINA LENARDUZZI, FRANCESCO LOMIO, NYTYI SAARIMAKI E DAVIDE TAIBI. **Does Migrating a Monolithic System to Microservices Decrease the Technical Debt?** ,2019.



ZHAMAK DEGHANI, M. **How to break a Monolith into Microservices** , 2018.

Disponível em:

<https://martinfowler.com/articles/break-monolith-into-microservices.html>. Acessado em

31 de janeiro de 2019.

ZIMMERMANN, O. **Microservices tenets: agile approach to service development and deployment. Computer Science-Research and Development**, 32(3):301–310, 2016.

WERNER VOGELS. CTO da Amazon, falou sobre a experiência da Amazon com serviços em 2011. **A Conversation with Werner Vogels. Learning from the Amazon technology platform**. Revista acmqueue. ACM Queue, 2006.



## **Apêndices**

Link onde ficou disponível o questionário da pesquisa:

<https://pt.surveymonkey.com/r/VN5WTHQ>