

ADESÃO DA ARQUITETURA DE MICROSERVIÇOS NAS GRANDES CORPORAÇÕES PARA DESENVOLVIMENTO OU MIGRAÇÃO DE APLICAÇÕES

GABRIEL PINHEIRO DE SANTANA LIMA¹
JOSÉ ROBERTO DE ARAÚJO FONTOURA²

UNIVERSIDADE DO ESTADO DA BAHIA (UNEB)
2023

RESUMO

A adesão da arquitetura de microsserviços tem se tornado uma tendência cada vez mais frequente pelas grandes corporações interessadas em melhorar a qualidade dos seus softwares, mas diante desse cenário surge o questionamento: Qual o impacto que a arquitetura de microsserviços pode trazer as grandes corporações? Este estudo investiga o impacto da adoção da arquitetura de microsserviços pelas grandes corporações para o desenvolvimento ou migração de aplicações. A metodologia empregada no estudo consiste em realizar uma revisão bibliográfica e documental a fim de buscar informações relevantes sobre o tema, além disso foi realizada uma coleta de dados com característica de observação simples. Ao avaliar os principais resultados obtidos, fica evidente que a adoção de microsserviços proporcionou melhorias significativas às empresas investigadas. Desta forma, este estudo contribui para o avanço do conhecimento em arquitetura de software e microsserviços, fornecendo uma compreensão mais profunda dos conceitos, princípios e benefícios associados à adoção de microsserviços pelas grandes corporações. Essas percepções e evidências são valiosas para as empresas interessadas em explorar essa arquitetura como uma estratégia para impulsionar a inovação, a eficiência e a competitividade no mercado.

Palavras-chave: Arquitetura de Microsserviços. Grandes Corporações. Competitividade.

ABSTRACT

Adherence to microservices architecture has become an increasingly frequent trend for large corporations interested in improving the quality of their software, but from this scenario the question arises: What impact can microservices architecture bring to large corporations? This study investigates the impact of the adoption of microservices architecture by large corporations for the development or migration of applications. The methodology used in the study consists of carrying out a bibliographical and documental review in order to seek relevant information on the subject, in addition, a data collection was carried out with the characteristic of simple observation. When evaluating the main results obtained, it is evident that the adoption of microservices provided significant improvements to the investigated companies. In this way, this study contributes to the advancement of knowledge in software architecture and microservices, providing a deeper understanding of the concepts, principles and benefits associated with the adoption of microservices by large corporations. These insights and evidence are valuable for companies interested in exploring this architecture as a strategy to drive innovation, efficiency and market competitiveness.

Keywords: Microservices Architecture. Large Corporations. Competitiveness.

¹ Graduando em Sistema de Informação – gp.psl55@gmail.com

² Graduado em Ciências Contábeis, Mestre em Contabilidade, Doutor em Difusão do Conhecimento e Professor da UNEB - jfontoura@uneb.br

1 INTRODUÇÃO

A arquitetura de microsserviços tem se destacado como uma arquitetura eficiente para o desenvolvimento ou migração de aplicações em grandes corporações. De acordo com Fowler (2014), essa arquitetura permite que as empresas modularizem seus sistemas em pequenos serviços independentes. No entanto, a adoção de microsserviços em ambientes corporativos complexos enfrenta desafios significativos, o que requer uma compreensão aprofundada de suas vantagens e dificuldades.

Diante desse contexto, surge a seguinte questão: Qual o impacto que a arquitetura de microsserviços pode trazer as grandes corporações ?

O objetivo geral deste trabalho é investigar como a arquitetura de microsserviços trouxe um impacto positivo para as grandes corporações. Para alcançar esse objetivo, foram estabelecidos os seguintes objetivos específicos:

1. Comparar as diferenças entre arquitetura de microsserviços e arquitetura monolítica.
2. Verificar como os microsserviços funcionam.
3. Demonstrar os benefícios que a arquitetura de microsserviços traz para as empresas.

A justificativa para a escolha desse tema está fundamentada na relevância dos microsserviços como uma arquitetura de software nas empresas. A tecnologia tem sido amplamente discutida, e percebe-se a necessidade de compreender melhor os benefícios e desafios associados à sua implementação.

A pesquisa busca resolver a questão de como os microsserviços podem beneficiar as empresas e como sua adoção pode impactar positivamente o desempenho, a eficiência e a agilidade dos softwares.

A pesquisa é relevante pois contribui para o avanço do conhecimento em arquitetura de software e microsserviços. Ela proporciona uma compreensão mais profunda dos conceitos, princípios e melhores práticas relacionadas à adoção de microsserviços. Fornecendo percepções valiosas para empresas interessadas em adotar a arquitetura.

A hipótese que sustenta este trabalho é que a adoção da arquitetura de microsserviços pelas grandes corporações para desenvolvimento ou migração de aplicações traz impactos positivos significativos. Essa arquitetura permite que as empresas desenvolvam e entreguem software de forma mais rápida, com maior qualidade e menor custo operacional.

A metodologia adotada para esta pesquisa tem como classificação a natureza básica, abordagem quali-quantitativa, método hipotético-dedutivo, e procedimentos bibliográficos e documentais. Além disso, a metodologia também aborda coleta de dados com característica de observação simples que foi realizada.

O presente trabalho está estruturado em dez seções, que abordam características importantes voltadas para o tema. Inicialmente, a introdução contextualiza o tema e apresenta os objetivos do estudo. Em seguida, são discutidas as características da arquitetura monolítica e da arquitetura de microsserviços, seguido de um comparativo entre ambas. Será abordado também o funcionamento dos microsserviços e como eles beneficiam as empresas, destacando sua presença cada vez mais frequente no ambiente corporativo. A metodologia empregada no estudo será detalhada, seguida pela análise dos resultados obtidos. Por fim, serão apresentadas as considerações finais, sintetizando os principais pontos discutidos.

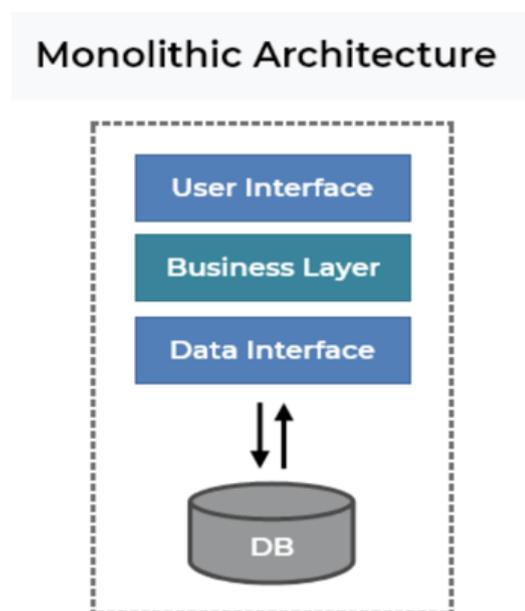
Por fim, espera-se que este trabalho contribua para o enriquecimento do conhecimento teórico e prático sobre a adoção de microsserviços, destacando sua importância como uma abordagem arquitetural moderna e promissora.

2 ARQUITETURA MONOLÍTICA

Neste capítulo, será explorado a arquitetura monolítica, suas características, vantagens e desvantagens. Também será examinado as situações em que a arquitetura monolítica pode ser adequada e quando é necessário considerar abordagens mais modernas destacando a visão de alguns autores sobre o assunto.

Elliott (2022) define a arquitetura monolítica como “um aplicativo composto de uma grande base de código que inclui todos os componentes do aplicativo”. Já Harris (2020) define que a arquitetura monolítica é o modelo tradicional do programa de software, que é construído como unidade unificada e é autossuficiente e independente de outros aplicativos.

Figura 1 - Ilustração da Arquitetura Monolítica



Fonte: Medium (2021)

De acordo com Newman (2015), a arquitetura monolítica é caracterizada por ter um único processo e por ser composta por diversas camadas, como a camada de apresentação, a camada de lógica de negócio e a camada de armazenamento de dados. Todas essas camadas são executadas dentro do mesmo processo, o que significa que as solicitações são processadas em sequência, de uma camada para a próxima.

Outro autor que aborda o tema é Fowler (2014), que define a arquitetura monolítica como uma arquitetura em que "todos os componentes da aplicação são empacotados e implantados juntos". Segundo o autor, essa abordagem torna a aplicação mais fácil de ser implantada e gerenciada, já que não há necessidade de gerenciar a comunicação entre diferentes componentes.

No entanto, a arquitetura monolítica também possui algumas desvantagens. De acordo com Newman (2015), essa abordagem pode dificultar a escalabilidade da aplicação, já que todas as solicitações são processadas dentro do mesmo processo. Além disso, a manutenção e o desenvolvimento de grandes sistemas monolíticos podem ser muito complexos, já que todas as camadas estão interligadas.

Fowler (2014) destaca que a arquitetura monolítica pode ser uma boa escolha para aplicações pequenas e simples, mas pode se tornar inviável em aplicações maiores e mais complexas. Nesse caso, é recomendado que a aplicação seja dividida em componentes menores, que possam ser implantados e gerenciados de forma independente.

3 ARQUITETURA DE MICROSERVIÇOS

Neste capítulo, será explorado a arquitetura de microsserviços, será discutido as características fundamentais dessa arquitetura e a visão de alguns autores em relação as vantagens e desvantagens. Compreender os princípios e as considerações da arquitetura de microsserviços é essencial para explorar seu potencial em projetos de desenvolvimento de software e avaliar se é uma opção adequada para determinadas aplicações.

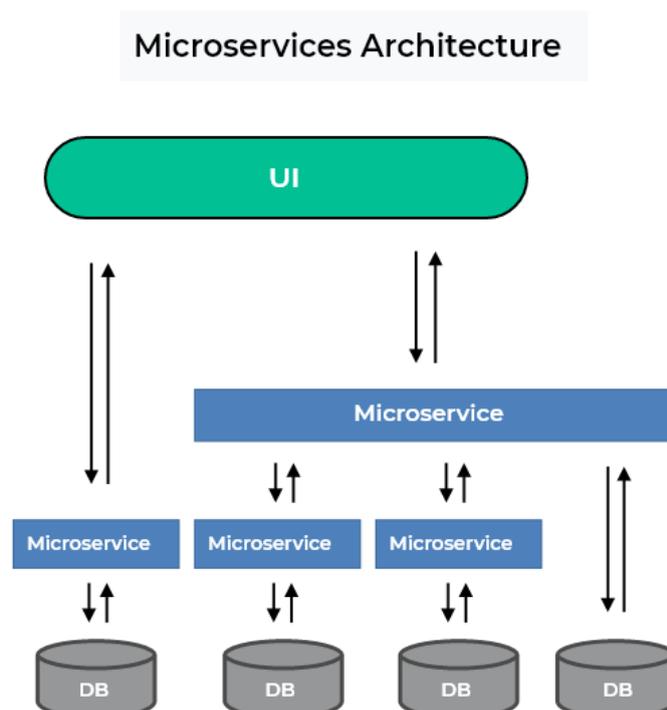
Richardson (2018) define que a arquitetura de microsserviços é um estilo arquitetural de software que estrutura uma aplicação como uma coleção de serviços que são:

- a) Independentes para implantação
- b) Fracamente acoplados
- c) Organizados em torno de capacidades de negócio
- d) Propriedade de uma pequena equipe

De acordo com Newman (2015), a arquitetura de microsserviços é caracterizada por ter serviços independentes que se comunicam por meio de interfaces de programação de aplicação (APIs). Cada serviço é responsável por uma funcionalidade específica da aplicação e pode ser desenvolvido, implantado e escalado de forma independente. Essa arquitetura permite que as equipes de desenvolvimento trabalhem em paralelo em diferentes serviços, o que pode acelerar o desenvolvimento e a entrega da aplicação como um todo.

Outro autor que aborda o tema é Fowler (2014), que define a arquitetura de microsserviços como uma arquitetura em que "um aplicativo é composto por muitos serviços autônomos que se comunicam uns com os outros por meio de APIs simples e bem definidas". Segundo o autor, essa abordagem oferece benefícios como escalabilidade, resiliência e evolução incremental.

Figura 2 - Ilustração da Arquitetura de Microsserviços



No entanto, a arquitetura de microsserviços também possui algumas desvantagens. De acordo com Newman (2015), a complexidade do gerenciamento de múltiplos serviços pode ser maior do que em uma arquitetura monolítica, já que há mais componentes para gerenciar. Além disso, a comunicação entre os serviços pode introduzir latência, o que pode afetar o desempenho da aplicação.

Fowler (2014) destaca que a arquitetura de microsserviços pode ser uma boa escolha para aplicações maiores e mais complexas, mas pode ser excessiva para aplicações menores e mais simples. Nesse caso, a arquitetura monolítica pode ser uma opção mais viável.

4 ARQUITETURA MONOLÍTICA X ARQUITETURA DE MICROSERVIÇOS

Neste capítulo, será abordado a discussão entre arquitetura monolítica e arquitetura de microsserviços no desenvolvimento de software, será explorado essas duas arquiteturas, destacando seus benefícios e pontos críticos, a fim de auxiliar na compreensão sobre a escolha adequada entre elas para diferentes projetos.

A arquitetura monolítica e a arquitetura de microsserviços são duas abordagens amplamente discutidas no desenvolvimento de software. A arquitetura monolítica, caracterizada por uma única aplicação em que todos os componentes estão agrupados e executados em conjunto, tem sido amplamente utilizada ao longo dos anos. Fowler (2014), destaca que essa abordagem é simples de ser desenvolvida e implantada, pois todos os componentes estão acoplados em uma única unidade.

No entanto, a arquitetura monolítica também apresenta desvantagens significativas. Newman (2015), ressalta que a escalabilidade e a manutenção podem ser desafiadoras em aplicações monolíticas. Além disso, um único erro ou falha em um componente pode afetar toda a aplicação.

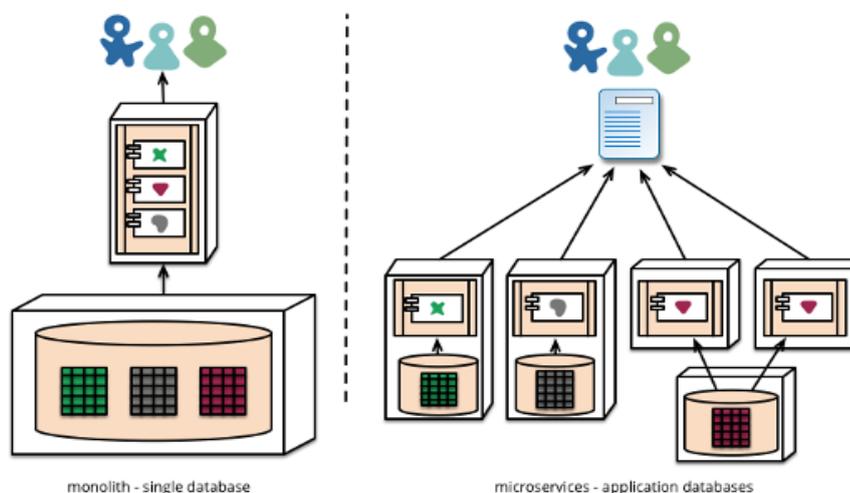
Por outro lado, a arquitetura de microsserviços tem ganhado destaque nos últimos anos. Nessa arquitetura, a aplicação é dividida em serviços independentes, cada um executando em seu próprio processo e se comunicando por meio de APIs. (Harris, 2020). Fowler (2014) destaca que a arquitetura de microsserviços traz benefícios como a escalabilidade horizontal, a agilidade no desenvolvimento e a possibilidade de utilizar diferentes tecnologias em cada serviço.

No entanto, a adoção da arquitetura de microsserviços também apresenta desafios. Newman (2015) salienta que a complexidade de gerenciar múltiplos serviços e a necessidade de lidar com a comunicação entre eles são pontos críticos a serem considerados.

No que diz respeito à manutenção, Beznos (2023) relata que a arquitetura de microsserviços sobressai, pois, cada serviço pode ser atualizado, corrigido ou substituído independentemente, sem afetar os demais. Na arquitetura monolítica, qualquer modificação requer a implantação e teste de toda a aplicação, o que pode ser um processo mais complexo e demorado.

A complexidade é outro aspecto importante a ser considerado. A arquitetura de microsserviços introduz uma maior complexidade devido à necessidade de gerenciar múltiplos serviços, garantir a comunicação entre eles e lidar com a consistência dos dados distribuídos (RICHARDSON, 2018). Isso pode exigir uma estratégia adequada de monitoramento e governança. Por outro lado, Elliott (2022) detalha que a arquitetura monolítica oferece uma visão mais simplificada e integrada da aplicação, facilitando a comunicação e o compartilhamento de dados, porém, ao custo de uma maior dependência e acoplamento entre os componentes.

Figura 3 - Comparação da Arquitetura Monolítica x Arquitetura de Microserviços



Fonte: Martin Fowler (2014)

Em resumo, a escolha entre arquitetura monolítica e arquitetura de microserviços para grandes corporações é um processo complexo. É necessário considerar cuidadosamente as características e desafios específicos de cada projeto.

5 FUNCIONAMENTO DOS MICROSERVIÇOS

Neste capítulo, será explorado o funcionamento dos microserviços, abordando sua divisão em componentes independentes, a forma como são desenvolvidos, implantados e executados. Também será analisado a importância da comunicação entre os microserviços, as opções disponíveis e as considerações a serem feitas nesse aspecto.

De acordo com Fowler (2016, p.24), “O funcionamento de um microserviço geralmente é dividido em três etapas principais: desenvolvimento, implantação e execução”.

Na etapa de desenvolvimento Fowler (2016, p.30) detalha que os microserviços são construídos com uma arquitetura modular, onde cada serviço é responsável por uma única funcionalidade. O desenvolvimento é realizado de forma independente para cada microserviço, geralmente utilizando linguagens de programação diferentes e com sua própria base de dados. A comunicação de microserviços é um aspecto fundamental da arquitetura, de acordo com Klug e Gardler (2022), a comunicação permite que diferentes microserviços se comuniquem e cooperem para fornecer uma funcionalidade mais complexa e completa. Existem várias maneiras pelas quais os microserviços podem se comunicar, e cada uma delas tem vantagens e desvantagens específicas.

Klug e Gardler (2022) também detalha que uma das formas mais comuns de comunicação de microserviços é por meio de interface que dois sistemas de computador usam para trocar informações de forma segura pela internet (APIs RESTful). As APIs RESTful usam o protocolo de transferência de hipertexto (HTTP) para transmitir dados entre os microserviços, usando métodos HTTP como método genérico para qualquer requisição que busca dados do servidor (GET), método genérico para qualquer requisição que envia dados ao servidor (POST), método específico para atualização de dados no servidor (PUT), e o método específico para remoção de dados no servidor (DELETE), para realizar operações em recursos específicos. As APIs RESTful são amplamente utilizadas em sistemas de microserviços devido à sua simplicidade e flexibilidade, bem como ao amplo suporte de ferramentas e bibliotecas. No entanto, os

autores comentam que o uso excessivo de APIs RESTful pode levar a um acoplamento excessivo entre os microsserviços e pode ser menos eficiente do que outras opções de comunicação.

Outra opção de comunicação de microsserviços é o uso de mensagens assíncronas. Ozkaya (2021) relata que nesse modelo, um microsserviço envia uma mensagem para uma fila ou tópico, que é então consumida por outros microsserviços interessados em receber a mensagem. O uso de mensagens assíncronas pode ajudar a reduzir o acoplamento entre os microsserviços e a aumentar a escalabilidade do sistema, uma vez que os microsserviços podem processar mensagens em paralelo. No entanto, Lypchenko (2023) alerta que o uso de mensagens assíncronas pode ser mais complexo do que o uso de APIs, especialmente quando se trata de garantir a consistência dos dados.

Klug e Gardler (2023) também destaca que outra opção de comunicação de microsserviços é o uso de eventos, em que um microsserviço emite um evento para notificar outros microsserviços de uma mudança em um estado ou ação. Os eventos são semelhantes às mensagens assíncronas, mas tendem a ser mais leves e menos acoplados, já que os microsserviços que consomem eventos não precisam se inscrever em filas ou tópicos específicos. Os eventos também podem ser usados para permitir a reação a eventos em tempo real, como a atualização de um painel de informações em tempo real. No entanto, os autores sinalizam que o uso excessivo de eventos pode ser difícil de gerenciar e pode levar a um sistema de microsserviços confuso e difícil de entender.

Em suma, a comunicação de microsserviços é um aspecto crítico da arquitetura, que pode ser abordado de várias maneiras, incluindo o uso de APIs RESTful, mensagens assíncronas e eventos. Cada abordagem tem seus prós e contras específicos, e a escolha da abordagem certa dependerá do contexto e dos requisitos do sistema em questão.

Na etapa de implantação, os microsserviços são implantados em um ambiente de nuvem ou em servidores dedicados. Conforme Ashtari (2022), cada microsserviço é implantado de forma independente, permitindo que as atualizações e correções sejam feitas de forma rápida e sem afetar outros serviços. A implantação é geralmente realizada com o uso de ferramentas de automação.

Aiswarya (2022) salienta que no processo de implantação, é importante levar em consideração o monitoramento dos microsserviços, para garantir que estejam funcionando corretamente e que possíveis problemas sejam identificados e resolvidos rapidamente.

Em resumo, a etapa de implantação de microsserviços é fundamental para garantir que os serviços estejam disponíveis e funcionando corretamente para os usuários. Ferramentas de automação são essenciais para tornar esse processo mais eficiente. O monitoramento também é um aspecto importante a ser considerado para garantir a disponibilidade e o desempenho dos serviços implantados.

Na etapa de execução, de acordo com Harris (2020) os microsserviços são executados de forma independente, cada um em sua própria instância. Cada serviço responde a solicitações específicas e pode ser dimensionado de forma independente, permitindo que os recursos sejam alocados de acordo com a demanda.

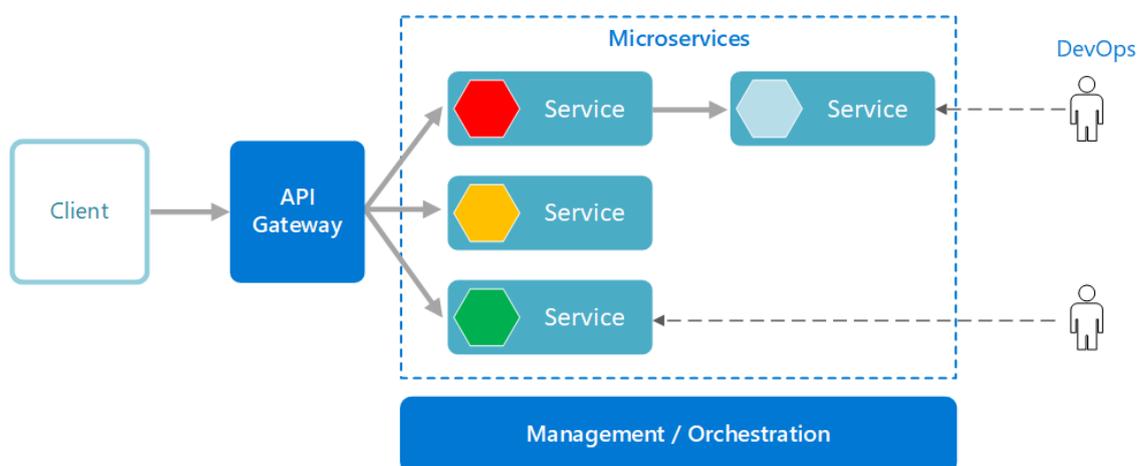
Eldridge (2018) aborda que, quando um serviço é implantado, ele normalmente é executado em um cluster de servidores. O cluster pode ser gerenciado por uma plataforma de orquestração de contêineres.

Dessa forma, o processo de execução de microsserviços é altamente escalável e flexível, permitindo que as empresas se adaptem rapidamente às mudanças na demanda. Além disso, a execução de microsserviços é altamente resiliente, pois se um serviço falhar, apenas aquele serviço específico será afetado e não afetará a operação de outros serviços.

Segundo a Team (2023) o fluxo de funcionamento de um microsserviço pode variar dependendo da arquitetura específica adotada, mas em geral, ele segue os seguintes passos:

1. Um cliente faz uma requisição a um microsserviço através de uma interface de programação de aplicação (API).
2. A requisição é recebida por um gateway, que pode realizar autenticação e autorização para garantir que o cliente tenha permissão para acessar o microsserviço.
3. O gateway encaminha a requisição para o microsserviço correspondente.
4. O microsserviço processa a requisição e retorna uma resposta para o gateway.
5. O gateway envia a resposta de volta ao cliente.
6. Se o microsserviço precisar acessar outros microsserviços para cumprir a requisição, ele fará isso através de chamadas de API, seguindo o mesmo fluxo.
7. Se o microsserviço precisar persistir dados, ele fará isso em seu próprio banco de dados, que pode ser independente dos bancos de dados de outros microsserviços.
8. Cada microsserviço é responsável por sua própria escalabilidade e disponibilidade. Isso significa que ele pode ser replicado horizontalmente para lidar com cargas de trabalho pesadas e pode ser colocado em sua própria máquina virtual ou contêiner para garantir sua disponibilidade.

Figura 4 - Fluxo de Funcionamento dos Microsserviços



Fonte: Microsoft (2022)

A Team (2023) também destaca que, para que a arquitetura de microsserviços funcione adequadamente, é necessário ter uma boa estratégia de comunicação entre os serviços. Isso envolve a escolha de protocolos de comunicação adequados, a definição de APIs claras e padronizadas e o uso de ferramentas de monitoramento para garantir a saúde dos serviços.

6 OS BENEFÍCIOS DOS MICROSERVIÇOS PARA AS EMPRESAS

Ao longo deste capítulo, será explorado como os microsserviços têm se mostrado vantajosos para empresas especialmente para as grandes corporações, proporcionando melhorias no desempenho, implantação e gerenciamento de aplicações.

Ashtari (2022) aborda que, a adoção de microsserviços pode mudar a forma como as equipes são formadas em uma organização. Isso ocorre porque os microsserviços permitem que diferentes partes de um sistema sejam desenvolvidas e

mantidas de forma independente, o que pode levar a uma maior divisão de responsabilidades e à criação de equipes mais especializadas.

Por exemplo, uma equipe pode ser responsável por desenvolver e manter um único microsserviço, em vez de trabalhar em um sistema monolítico completo. Essa equipe pode ser formada por desenvolvedores, testadores, operadores e outros especialistas, que trabalham juntos para garantir que o microsserviço esteja funcionando de forma eficiente e confiável.

Segundo a Digiteum (2022), as grandes corporações têm adotado a arquitetura de microsserviços devido a uma série de benefícios que ela pode oferecer em termos de desempenho, manutenção, implantação e gerenciamento. Algumas das principais razões para isso são:

- a) Escalabilidade: A arquitetura de microsserviços permite que os serviços individuais sejam dimensionados separadamente, o que permite que a aplicação como um todo seja escalável de maneira mais eficiente. Isso significa que, se uma parte da aplicação precisar de mais recursos, é possível adicionar mais instâncias do serviço correspondente, em vez de dimensionar toda a aplicação. Isso é especialmente importante para grandes corporações, que geralmente têm requisitos de escala muito altos.
- b) Resiliência: A arquitetura de microsserviços é projetada para ser tolerante a falhas. Como os serviços individuais são independentes uns dos outros, um problema em um serviço não afetará necessariamente os outros. Além disso, a arquitetura de microsserviços geralmente inclui recursos para lidar com falhas de maneira automática, como a reinicialização de serviços ou a mudança para instâncias de backup.
- c) Agilidade: A arquitetura de microsserviços permite que as equipes de desenvolvimento trabalhem de maneira mais independente e ágil. Cada serviço pode ser desenvolvido e implantado de maneira independente, sem afetar necessariamente os outros. Isso permite que as equipes de desenvolvimento trabalhem de maneira mais rápida e eficiente, sem depender de outras equipes.
- d) Manutenção: A arquitetura de microsserviços também pode facilitar a manutenção de aplicativos. Como os serviços são independentes, é possível atualizar ou substituir um serviço sem afetar os outros. Isso significa que as atualizações podem ser feitas de maneira mais rápida e com menos interrupções.

7 PRESENÇA DOS MICROSERVIÇOS NAS EMPRESAS

Neste capítulo, será abordado a presença dos microsserviços nas empresas e como sua adoção tem ocorrido dentro dos ambientes de trabalho de acordo com pesquisas.

A adoção de microsserviços aconteceu em ondas, desde que o termo foi criado em 2011. De acordo com uma pesquisa da Nginix (2015), 36% das empresas pesquisadas estão usando microsserviços, com outros 26% na fase de pesquisa.

Segundo o IT Fórum (2018), o Walmart Canadá mudou toda a sua arquitetura de software para microsserviços em 2012. A empresa, que não conseguiu lidar com os 6 milhões de páginas vistas por minuto que estava recebendo no momento, obteve resultados instantâneos com um aumento significativo em sua taxa de conversão noturna. O tempo de inatividade foi minimizado, e a empresa conseguiu substituir o hardware caro com servidores x86 virtuais mais baratos, resultando em uma economia global de custos entre 20% a 50%.

A pesquisa da TechRepublic Premium realizada entre março e abril de 2020, com 447 profissionais, mostrou que os microsserviços estão no radar da maioria das empresas:

- a) 96% afirmaram ter familiaridade com o conceito de microsserviços.
- b) 73% já integraram microsserviços em suas aplicações.
- c) Entre os que ainda não incorporaram a tecnologia, 63% pretendem fazer isso.

Em outra pesquisa, esta da Kong, 84% dos líderes de tecnologia consideram que os microsserviços estarão entre os motores da inovação e já o adotaram para acelerar os esforços de inovação e competitividade.

Para os entrevistados de ambas as pesquisas, no entanto, a falta do conjunto de habilidades apropriadas, bem como muitos sistemas legados, e falta de apoio corporativo são as principais razões por que as organizações ainda não integraram microsserviços em suas aplicações ou têm dificuldade de escalar.

8 METODOLOGIA

Esse capítulo tem como objetivo descrever como a pesquisa foi conduzida, abordando as características quanto a classificação da pesquisa em relação a natureza, objetivo, abordagem, método, e procedimentos além da coleta dos dados. Também foi realizado um levantamento de fontes bibliográficas e documentais utilizadas para embasar a pesquisa.

A natureza da pesquisa foi classificada como básica, pois buscou gerar conhecimentos teóricos sobre a adoção de microsserviços em grandes corporações. O foco principal foi no entendimento dos desafios e vantagens associados a essa adoção.

O objetivo da pesquisa foi descritivo, buscando descrever e analisar os principais impactos da adoção de microsserviços em grandes corporações.

Foi adotada uma abordagem quali-quantitativa, buscando compreender as experiências e percepções das empresas em relação à adoção de microsserviços. Além de identificar através dos números uma crescente em relação a adoção da arquitetura. Foi dada ênfase à compreensão dos fatores-chave que influenciam a adoção e ao contexto em que ocorrem.

O método utilizado foi o hipotético-dedutivo. Com base em teorias existentes e em uma revisão científica, foi formulada uma hipótese que sustenta este trabalho: A adoção da arquitetura de microsserviços pelas grandes corporações para desenvolvimento ou migração de aplicações traz impactos positivos significativos. Essa arquitetura permite que as empresas desenvolvam e entreguem software de forma mais rápida, com maior qualidade e menor custo operacional.

Os procedimentos adotados foram bibliográficos e documentais. Foi realizada uma revisão científica, buscando artigos, livros e estudos relevantes sobre a adoção da arquitetura microsserviços. Além disso, foram analisadas algumas situações reais de empresas que passaram por migração de arquitetura, para obter informações sobre os desafios e vantagens encontrados.

A coleta de dados foi realizada por meio da observação simples. Foi realizada uma investigação documental de conteúdos relevantes relacionados a arquitetura de microsserviços. Os dados coletados a partir dessas fontes documentais foram analisados em busca de informações que respondam aos objetivos da pesquisa.

9 ANÁLISE DOS RESULTADOS

Nesta seção, será apresentado os resultados obtidos por meio da análise dos dados coletados no contexto da adoção da arquitetura de microsserviços pelas grandes corporações para o desenvolvimento ou migração de aplicações. Esses resultados fornecem recursos valiosos sobre o impacto da arquitetura de microsserviços nas empresas.

A Amazon inicialmente foi construída como um software monolítico de várias camadas. Com o tempo, à medida que começaram a crescer, a empresa em 2001 começou a sofrer com atrasos no desenvolvimento, desafios de codificação e interdependências de serviço inibiram a sua capacidade de atender aos requisitos de escalabilidade de sua base de clientes em rápido crescimento.

O gerente da Amazon Web Services (AWS), Brigham (2015) comenta que o serviço amazon.com foi arquitetado em várias camadas, e essas camadas continham muitos componentes. Mas todos eles estão muito interligados, comportando-se como um grande monólito. Agora, muitas startups e até mesmo projetos dentro de grandes empresas começam dessa forma. Eles adotam uma abordagem de monólito primeiro, porque é muito rápido para começar a se mover rapidamente. Mas com o tempo, à medida que o projeto amadurece, mais desenvolvedores são adicionados, ele cresce e a base de código fica maior e a arquitetura fica mais complexa, esse monólito começará a adicionar sobrecarga ao seu processo, e o ciclo de desenvolvimento de software começará a desacelerar.

Diante da necessidade de refatorar seu sistema a partir do zero, a Amazon quebrou seus aplicativos monolíticos em pequenos aplicativos específicos de serviço, executados de forma independente. O uso de microsserviços mudou imediatamente o funcionamento da empresa. Ele foi capaz de alterar características e recursos individuais, o que trouxe melhorias massivas e imediatas à funcionalidade do site.

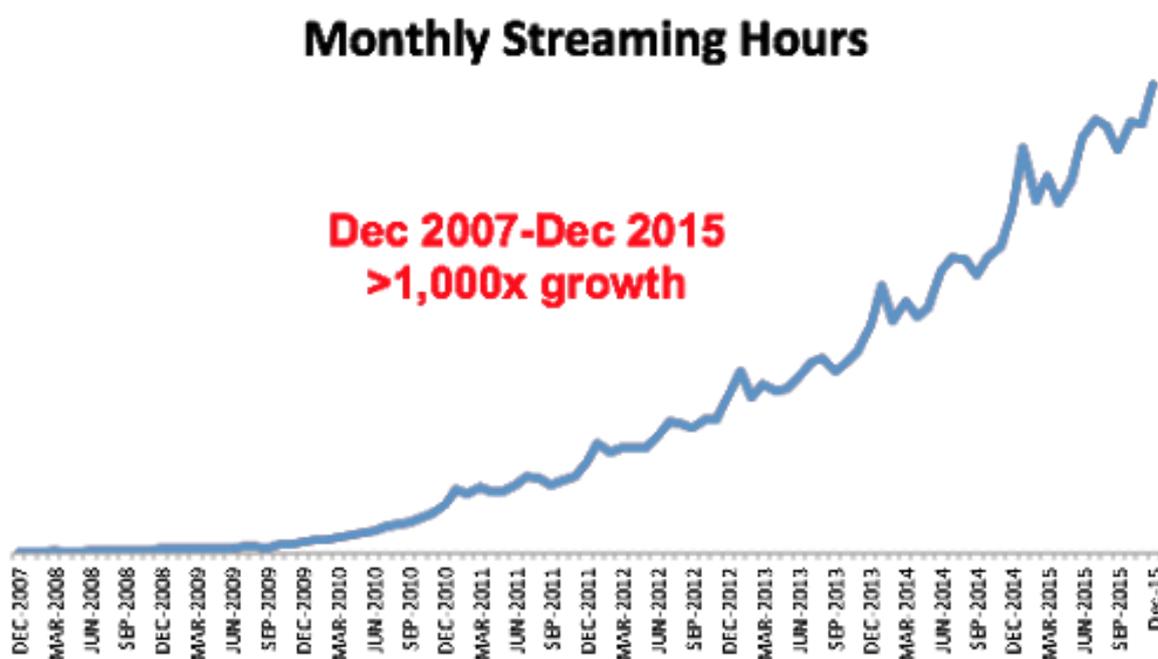
“A mudança de uma arquitetura monolítica para uma arquitetura de microsserviços é uma mudança cultural, não apenas técnica” (NEWMAN, 2015, p.77). A citação de Newman ressalta que a migração da Amazon para uma arquitetura de microsserviços não se limitou apenas a aspectos técnicos, mas exigiu uma transformação cultural em toda a organização. A empresa precisou adotar uma mentalidade ágil e colaborativa, promovendo a autonomia das equipes e a responsabilidade pelos serviços individuais, para obter os benefícios da arquitetura de microsserviços.

A Netflix iniciou seu serviço de streaming de filmes em 2007. Em 2008, ela estava sofrendo com interrupções de serviço e desafios de dimensionamento; por três dias, não foi possível enviar DVDs aos membros.

O engenheiro sênior da Netflix, Hahn (2015) detalha que a jornada para a nuvem na Netflix começou em agosto de 2008, quando sofremos uma grande corrupção de banco de dados e por três dias não pudemos enviar DVDs para nossos membros. Foi quando percebemos que precisávamos mudar de pontos únicos de falha dimensionados verticalmente, como bancos de dados relacionais em nosso datacenter, para sistemas distribuídos altamente confiáveis e escalonáveis horizontalmente na nuvem. Escolhemos a AWS da Amazon como nosso provedor de nuvem porque nos forneceu a maior escala e o conjunto mais amplo de serviços e recursos.

A refatoração para microsserviços permitiu que a Netflix superasse seus desafios de dimensionamento e interrupções de serviço. Em 2013, conforme a própria empresa comenta o gateway de API da Netflix estava lidando com dois bilhões de solicitações diárias, gerenciadas por mais de 500 microsserviços hospedados na nuvem. Em 2017, sua arquitetura consistia em mais de 700 microsserviços fracamente acoplados. Hoje, a Netflix fatura cerca de US\$ 8 bilhões por ano e transmite aproximadamente seis bilhões de horas de conteúdo semanalmente para mais de 220 milhões de assinantes em 190 países, e continua a crescer.

Gráfico 1 – Representação visual do crescimento da Netflix de 2007 a 2015



Fonte: Netflix (2016)

Conforme Fowler (2014), à medida que um aplicativo cresce, a arquitetura monolítica se torna uma limitação para sua escalabilidade. A migração para microsserviços possibilitou que a Netflix lidasse com o aumento exponencial do volume de solicitações, fornecendo um serviço de streaming de filmes confiável e escalável.

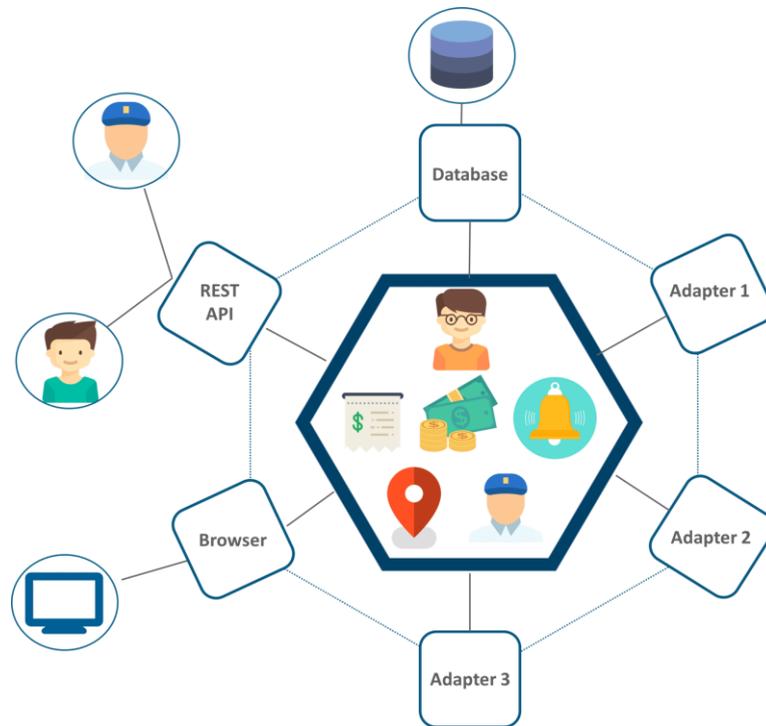
Mas isso não é tudo. A Netflix recebeu outro benefício dos microsserviços: redução de custos. De acordo com a empresa, “seus custos de nuvem por início de streaming acabaram sendo uma fração daqueles no data center, um benefício colateral bem-vindo”. A empresa também é responsável por mudar a forma como as pessoas assistem a programas de televisão. Hoje em dia, você precisa se inscrever em um site específico para receber os programas que deseja, graças ao Netflix iniciar a guerra do streaming. No entanto, a arquitetura de microsserviços está por trás de tudo.

Como uma startup, a Uber precisava apenas de alguns recursos: Conectar motoristas e passageiros, cobranças e pagamentos. O aplicativo precisava oferecer um único serviço em uma única cidade: San Francisco que pertence ao estado da Califórnia nos Estados Unidos. É por isso que foi construído como um aplicativo monolítico: naquela época, esse aplicativo era capaz de atender a todos os requisitos de negócios, implementando as principais funcionalidades de negócios e oferecendo uma base de código limpa.

Quando a Uber se expandiu para mais cidades, eles começaram a introduzir novos produtos e serviços. A aplicação começou a crescer rapidamente e foi aí que manter o sistema monolítico se tornou um verdadeiro desafio.

Conforme relata o engenheiro sênior da Uber, Ranney (2016) implantar a base de código significava implantar tudo de uma vez, o que dificultava a integração contínua. A outra desvantagem era que apenas desenvolvedores que trabalhavam há muito tempo na Uber podiam fazer alterações no sistema: uma única alteração se tornava uma responsabilidade enorme por causa das dependências entre os componentes do aplicativo. Adicionar novos recursos, corrigir bugs e resolver dívidas técnicas em um único repositório tornou-se extremamente difícil. O conhecimento geral da aplicação era necessário antes de tentar fazer uma única mudança.

Figura 5 - Representação da Arquitetura Monolítica da Uber

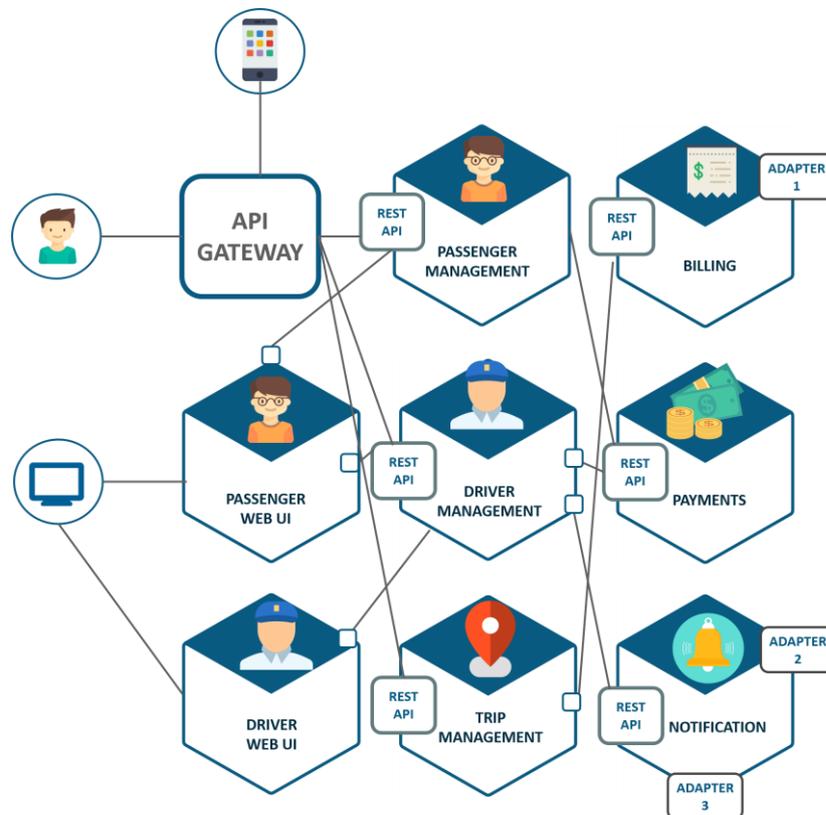


Fonte: Kappagantula (2018)

A Uber decidiu dividir o monólito em várias bases de código para formar uma arquitetura de microsserviços.

“Os microsserviços tornam mais fácil evoluir e dimensionar partes individuais do sistema sem afetar o todo” (RICHARDSON, 2018). Como o software construído como microsserviços é dividido em pequenos serviços independentes, cada componente pode ser escrito em sua própria linguagem, usar sua própria estrutura e ter seu próprio banco de dados. Isso ajudou a Uber a resolver todos os seus problemas no processo de desenvolvimento, criar entrega contínua e acelerar o crescimento dos negócios.

Figura 6 - Representação da Arquitetura de Microsserviços da Uber



Fonte: Kappagantula (2018)

Os microsserviços ajudaram a Uber a desbloquear oportunidades de crescimento de negócios e superar restrições.

As descobertas revelaram que a adoção de microsserviços por empresas como Amazon, Netflix e Uber trouxe benefícios significativos em termos de escalabilidade, agilidade no desenvolvimento e redução de custos de infraestrutura. Essas empresas enfrentaram desafios relacionados ao crescimento, desempenho e complexidade de suas aplicações monolíticas, o que levou à busca por uma abordagem mais modular e flexível, como os microsserviços.

Ao adotar microsserviços, a Amazon conseguiu melhorar a funcionalidade de seu site ao permitir alterações específicas em componentes individuais. A Netflix superou desafios de dimensionamento e interrupções de serviço, alcançando uma arquitetura composta por centenas de microsserviços hospedados na nuvem. O Uber resolveu problemas de integração contínua e acelerou o crescimento dos negócios ao migrar para uma arquitetura de microsserviços.

No entanto, também foi identificada algumas contradições teóricas em relação aos microsserviços. Alguns estudos sugerem que a fragmentação em microsserviços pode aumentar a complexidade da gestão e a sobrecarga de comunicação entre os serviços. Além disso, questões relacionadas à segurança e resiliência podem surgir devido às várias conexões de rede e APIs utilizadas nos aplicativos baseados em microsserviços.

Apesar dessas contradições, as aplicações práticas dos microsserviços têm sido amplamente adotadas pelas grandes corporações, como evidenciado pelos casos de sucesso da Amazon, Netflix e Uber. Essas empresas alcançaram melhorias

significativas em seus processos de desenvolvimento de software, permitindo a entrega contínua de valor aos clientes e aumentando sua competitividade no mercado.

Ao comparar os resultados e interpretações com trabalhos já publicados, é confirmado que a adoção de microsserviços pode trazer benefícios, como a melhoria da escalabilidade, a agilidade no desenvolvimento e a redução de custos. Essa arquitetura permite que as empresas se adaptem às demandas do mercado e superem os desafios associados ao crescimento e à complexidade de suas aplicações.

10 CONSIDERAÇÕES FINAIS

Nesta pesquisa, foi abordada a adoção da arquitetura de microsserviços por grandes corporações, no contexto do desenvolvimento ou migração de aplicações. O objetivo geral deste estudo foi analisar os impactos e benefícios dessa arquitetura nas empresas.

Ao avaliar os principais resultados obtidos a partir da análise dos casos de sucesso, fica evidente que a adoção de microsserviços proporcionou melhorias significativas às empresas investigadas. As corporações foram capazes de superar desafios de escalabilidade, melhorar o desempenho e a funcionalidade de suas aplicações, além de reduzir os custos de infraestrutura.

No caso da Amazon, a migração para microsserviços possibilitou um ambiente mais ágil e escalonável para recursos individuais do site, onde é mais fácil gerenciar e atualizar esses recursos em comparação a uma plataforma monolítica. Na Netflix, a arquitetura de microsserviços permitiu lidar com um volume massivo de solicitações diárias, garantindo um serviço de streaming confiável e escalável. Já o Uber resolveu problemas de integração contínua e acelerou o crescimento do negócio ao adotar microsserviços.

Em resposta ao problema levantado, a pesquisa identificou que a adoção da arquitetura de microsserviços tem um impacto positivo nas grandes corporações. Essa arquitetura permite que as empresas alcancem maior agilidade, flexibilidade e eficiência em seus processos de desenvolvimento e manutenção de sistemas. Além disso, ela oferece a capacidade de escalar e evoluir de forma mais eficiente, possibilitando a integração contínua e a entrega de software de alta qualidade.

Ao confrontar o resultado com a hipótese levantada, as evidências obtidas confirmam a hipótese de que a adoção da arquitetura de microsserviços pelas grandes corporações traz um impacto positivo. Os benefícios identificados, como a rapidez no desenvolvimento, a melhoria da qualidade e a redução dos custos operacionais, sustentam a ideia de que essa arquitetura é uma opção viável e vantajosa para as empresas.

Desta forma, este trabalho contribui para o avanço do conhecimento em arquitetura de software e microsserviços, fornecendo uma compreensão mais profunda dos conceitos, princípios e benefícios associados à adoção de microsserviços pelas grandes corporações. Essas percepções e evidências são valiosas para as empresas interessadas em explorar essa arquitetura como uma estratégia para impulsionar a inovação, a eficiência e a competitividade.

REFERÊNCIAS

AISWARYA, S. Advantages of Microservices Architecture. 2022. Disponível em: <https://www.atatus.com/blog/microservices-architecture/>. Acesso em: 21/04/2023.

ASHTARI, H. What Are Microservices? 2022. Disponível em: <https://www.spiceworks.com/tech/devops/articles/what-are-microservices/>. Acesso em: 25/04/2023.

BEZNOS, M. Microservices vs Monolith: Which Architecture is the Best Choice for Your Business? 2023. Disponível em: <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/>. Acesso em: 03/04/2023.

DIGITEUM. Advantages of Microservices Architecture. 2022. Disponível em: <https://www.digiteum.com/advantages-microservices-architecture/>. Acesso em: 08/05/2023.

ELDRIDGE, I. Container Orchestration Explained. 2018. Disponível em: <https://newrelic.com/blog/best-practices/container-orchestration-explained>. Acesso em: 25/06/2023.

ELLIOTT, R. Monolithic vs Microservice Architecture. 2022. Disponível em: <https://www.digitalocean.com/blog/monolithic-vs-microservice-architecture>. Acesso em: 02/06/2023.

FOWLER, M. Microservices. 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 28/01/2023.

FOWLER, S. J. Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization. O'Reilly Media, 2016.

HARRIS, C. Microservices vs. Monolith. 2020. Disponível em: <https://www.atlassian.com/br/microservices/microservices-architecture/microservices-vs-monolith>. Acesso em: 27/05/2023.

KLUG, C.; GARDLER, R. Communication in Microservice Architecture. 2022. Disponível em: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/communication-in-microservice-architecture>. Acesso em: 11/03/2023.

KLUG, C.; GARDLER, R. Integration Event-Based Microservice Communications. 2023. Disponível em: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/integration-event-based-microservice-communications>. Acesso em: 11/03/2023.

LYPCHENKO, S. Monolith vs. Microservices Architecture: Which is the Best Choice for Your Business? 2023. Disponível em: <https://gowombat.team/blog/posts/monolith-vs-microservices-architecture-which-is-the-best-choice-for-your-business>. Acesso em: 05/04/2023.

MW TEAM. Microservices Architecture. 2023. Disponível em: <https://middleware.io/blog/microservices-architecture/>. Acesso em: 01/05/2023.

NEWMAN, S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Inc., 2015.

NETFLIX. Netflix Cloud Migration. 2016. Disponível em: <https://about.netflix.com/en/news/completing-the-netflix-cloud-migration>. Acesso em: 17/06/2023.

OZKAYA, M. Microservices: Asynchronous, Message-Based Communication. 2021. Disponível em: <https://medium.com/design-microservices-architecture-with-patterns/microservices-asynchronous-message-based-communication-6643bee06123>. Acesso em: 22/05/2023.

RICHARDSON, C. Microservices. 2018. Disponível em: <https://microservices.io/>. Acesso em: 17/02/2023.

RICHARDSON, C. Microservices Patterns. 2018. Disponível em: <https://microservices.io/patterns/microservices.html>. Acesso em: 16/02/2023.