Associação Propagadora Esdeva

Centro Universitário Academia – UniAcademia

Curso de xxxxxxxx

Trabalho de Conclusão de Curso – Artigo\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Migração de um monólito para uma arquitetura de microsserviços *serverless* na nuvem

*Rogério Tostes[[1]](#footnote-0)*

*Centro Universitário Academia, Juiz de Fora, MG*

*Tassio Ferenzini Martins Sirqueira[[2]](#footnote-1)*

*Centro Universitário Academia, Juiz de Fora, MG*

Linha de Pesquisa: xxxxxxxxxxxx

# RESUMO

# [Contexto] ... [Objetivo] ... [Metodologia] ... [Resultado] ... [Considerações]

(Fonte: Arial 12; espaço simples; em português; com mínimo de 100 e máximo de 250 palavras).

**Palavras-chave:** Mínimo de 3 e máximo de 5, separadas por ponto final, sendo iniciadas com letra maiúscula e finalizada com ponto. Se for palavra composta, segunda palavra inicia com letra minúscula. Diferentes das palavras do título.

***ABSTRACT***

*Resumo em inglês*

# INTRODUÇÃO

A tecnologia, tal qual conhecemos hoje, passou por diversas transformações ao longo dos tempos e continua avançando de forma vertiginosa; conforme afirma Joana Neves, vivemos em: “ um processo dinâmico de melhorias, crescimento e evolução” (NEVES, 2002, p.276).

Por trás dos diferentes aplicativos, das ferramentas e sites – entre outros serviços – que usamos sempre que acessamos o mundo digital, existe uma engenhosa estrutura responsável pelo bom desempenho dessas operações.

Elaine Figueiredo (2013) reforça tais observações em seu artigo *Arquiteturas de sistemas Web 3.0.* da seguinte forma, são: “inovações tecnológicas, inovações causadas pelo surgimento de novos negócios, novos frameworks, novas linguagens e novos protocolos que surgem em um intervalo cada vez menor de tempo”; o que caracteriza um cenário marcado por uma “moderna rede de computadores sociáveis e interoperáveis”.

Neste aspecto, é necessário pensar na arquitetura dos sistemas e em suas estruturas com frequência. *Softwares* são criados e desenvolvidos como solução para diversas questões, entre elas (aperfeiçoar processos, melhorar experiências digitais e ampliar mercados). A maior parte destes programas é desenvolvida em monólitos, no entanto, não há como precisar a extensão de serviços que esses programas podem alcançar.

Monólito, conforme o dicionário *Priberam* significa “obra construída em um único bloco”, é por isso que esse termo é utilizado para definir a arquitetura de alguns sistemas. Sendo assim, uma arquitetura monolítica é um sistema único, não dividido, que roda em um único processo, uma aplicação de software em que diferentes componentes estão ligados a um único programa dentro de uma única plataforma. Com o tempo, as estruturas monolíticas vão sofrendo mudanças, suas escalabilidade se tornam inflexíveis e sua manutenção mais complexa. Nesse processo, é interessante pensar em uma arquitetura cujo sistema responda a essas funções com mais flexibilidade.

A arquitetura de microsserviços apresenta um software diferente do monólito, no qual sistemas grandes e complexos são transformados em componentes individuais. As vantagens dos microsserviços são a sua escalabilidade distribuída, ou seja, não é preciso crescer a aplicação inteira para solucionar um problema, podemos escalar apenas recursos específicos, contrário ao monólito cujo auto acoplamento dos sistemas dificultam a manutenção e evolução dos sistemas. Nesse ponto, vale pensar em uma arquitetura de microsserviços que se comunicam entre si por meio de redes e, como opção de arquitetura,oferecem diversos modos de resolver problemas que você poderá enfrentar. (NEWMAN. 14, 2020)

*Serverless* é um termo que traduzido literalmente do inglês, significa “sem servidor”, o que não é bem verdade. A arquitetura *serverless* ainda é baseada em servidores. Nela, porém, o desenvolvedor não precisa configurar ou se preocupar com a maioria dos aspectos da infraestrutura em que sua aplicação será rodada. Essas decisões dinâmicas de infraestrutura ficam ocultas para o desenvolvedor ou operador.

A arquitetura *serverless* é um tema bastante em alta nos tempos atuais por questão de computação em nuvem, mas pudemos observar que, principalmente os sistemas disponibilizados pelo governo ficaram defasados pelo formato monolítico do sistema, por falta de manutenção, pelas dificuldades na evolução dos sistemas e até mesmo por questão de segurança, uma vez que softwares pouco flexíveis são mais vulneráveis à ataques.

Isto posto, a proposta do presente trabalho é mostrar a relevância de se construir uma arquitetura de *software* conhecido como *serverless,* partindo da ideia de evolução da arquitetura de um *monólito*. Para a construção dessa arquitetura neste trabalho está sendo usada a AWS (Amazon Web System) que é uma plataforma de sistema em nuvem.

1.2 OBJETIVOS

Esta seção aborda os objetivos a serem alcançados através deste projeto, citando o objetivo geral e os objetivos específicos, a saber:

1.2.1 Objetivo Geral

* Criar uma arquitetura de microsserviços em *serverless*, partindo da ideia de evolução da arquitetura de um *monólito*, a fim de viabilizar um sistema menos inflexível e com possibilidades de ajustes. Com tal perspectiva, o objetivo geral se desdobra em objetivos “menores” mais específicos, a saber:

1.2.2 Objetivos específicos

* Possibilitar a criação de um software resiliente, visando assim a garantia de que o sistema funcione, mesmo quando outros sistemas dependentes não operem bem;
* Propiciar um software escalável de modo a destacar sua capacidade de expansão sem aumentar de forma drástica os custos com recursos técnicos, e como aplicá-lo;
* Oferecer um software seguro e diminuir a vulnerabilidade de dados dos usuários.

1.3 JUSTIFICATIVA

Atualmente grande parte das aplicações utiliza a arquitetura monolítica como padrão para desenvolvimento de software, uma vez que ela parte do princípio de que todos os processos dentro de uma mesma solução estão ligados. Isso quer dizer que são acoplados, que estão dentro de uma base de dados similar e executam seus processos como um único serviço. Contudo, em determinados casos esta arquitetura se mostra inviável e propensa a falhas, tais como: a dificuldade de escalar a aplicação; o alto custo com a manipulação e manutenção, entre outras falhas.

Em 2007 o governo federal lançou o software Público Brasileiro (SPB), utilizado como alicerce para definir a política de desenvolvimento, distribuição e uso de software pelo setor público do Brasil. Dentre esses sistemas, destaque para o Controle de Marcas e Sinais (CMS) que auxilia o registro e consulta por marcas e sinais utilizados por produtores rurais.

O CMS é um sistema monolítico e a ideia do trabalho, conforme já afirmado, é redesenhar esse programa e trasnsformá-lo em uma arquitetura de microserviços serverless e servir como base para a migração de outros quaisquer sitemas do tipo monolítico, pois os conceitos servem como base de inspiração e estudos.

Em resposta a isso surgiu uma nova arquitetura de software pensada a partir da lógica de microsserviços, tornando possível modularizar serviços menores responsáveis por determinada área da aplicação e que podem funcionar de forma independente.

Com base nessas necessidades foi pensado em um *software serverless* fundamentado no modelo de microsserviços que viabiliza sua escalabilidade, resiliência e evolutiva; compreendendo os termos da seguinte forma:

* Escalabilidade:é a capacidade de aumentar o tamanho do software ou do seu uso. Em outras palavras, é uma característica desejável em todo sistema, rede ou um processo, que indica sua habilidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer.
* Resiliente: é aquele sistema capaz de cumprir sua missão mesmo diante de crises e adversidades. Isto é, fornecer os recursos necessários e continuar operante, apesar do estresse excessivo da operação.
* Evolutiva: é a arquitetura que suporta mudanças contínuas e incrementais como um primeiro princípio por meio de vários aspectos.

Com isso, nota-se que a arquitetura *Serverless* apresenta com frequência, “maior flexibilidade e são passíveis de mudança” (Newman, 2015) ou atualizações, tanto como um todo, quanto por meio de atualizações independentes de seus componentes, conforme o autor:

A natureza independente das implantações da abertura a novos modelos para aumentar a escala e robustez dos sistemas, além de permitir que você misture e combine diferentes tecnologias. Como podemos trabalhar em serviços em paralelo, é possível incluir mais desenvolvedores para resolver um problema sem que atrapalhem uns aos outros. (2015, p. 20).

A possibilidade supracitada se dá pelo fato de a arquitetura ser distribuída e permitir que enquanto seus componentes são atualizados, não seja comprometido o funcionamento da aplicação. Além das vantagens citados anteriormente,

O isolamento de processos também torna possível diversificar as opções de tecnologia de escolhermos, talvez misturando diferentes linguagens e estilos de programação, plataforma de implantação ou bancos de dados para encontrar a combinação correta. (2015, p.20).

Pode-se inferir, portanto, que as arquiteturas de microsserviços também podem oferecer uma maior flexibilidade.

1. **REFERENCIAL TEÓRICO**

Todo sistema operacional parte de uma organização, de uma estrutura. Essa organização é também conhecida como uma arquitetura de software que é responsável por definir quais componentes farão parte de um projeto, quais suas características, funções e a forma como devem interagir entre si e com outros softwares.

Dentro dessa estrutura, é possível entender as diferenças entre as linguagens, sistemas operacionais e ambientes da computação, e por isso mesmo, torna-se uma ideia difícil de se conceituar. Microsserviços e *serverless* são temas caros, por serem relativamente novos em se tratando de discussões teóricas, percebe-se a ausência de fontes mais aprofundadas. Em contrapartida, há muitas referências empíricas no universo tecnológico.

Do mesmo modo, migrar de um sistema para outro também é uma escolha complexa, que envolve a análise de diversos fatores. Desenvolver um software mais flexível que visa promover certa “mobilidade” e agregar novas tecnologias exige planejamento e estudo. Sam Newman em seu livro “Migrando sistemas monolítico, para microsserviços” (2020), traz conceitos, vantagens e desvantagens da migração e referências de melhores práticas para fazer essa migração.

Eric Evans fala no desenvolvimento de domínios, em seu livro “Domain-Drive Design” (DDD) (2016) traz a modelagem de domínio para o software, visando tirar melhor proveito para o negócio. Pensar em DDD significa desenvolver sistemas de acordo com o domínio relacionado ao problema que estamos propondo resolver. Na criação de uma arquitetura, é fundamental conhecer o domínio, pois é onde reside um conjunto de conceitos, princípios e técnicas necessárias ao desenvolvimentos de sistemas.

Um dos primeiros artigos sobre arquitetura de software publicado foi de autoria de Edsger Dijkstra, em 1968. Mas, o termo “Arquitetura de software” se consolidou no meio científico no final da década de 1980 (GIROLDO, 2022). Contudo, vale ressaltar que o foco abordado aqui, neste trabalho, debruça-se sobre a questão da criação de uma arquitetura de microsserviços em um sistema *serverless.*

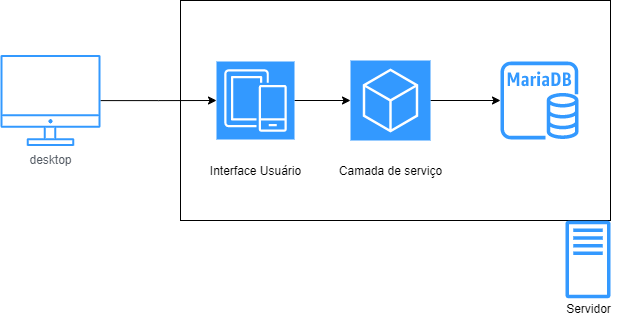
Os estudiosos Alexandre Davi Zanelatto (2019) e Gustavo Matozinho Lima (2022)

apresentam trabalhos relevantes com relação ao tema de uma arquitetura *serverless*, entretanto eles somente definem conceitos e não apresentam propostas pragmáticas, ou seja, não entregam uma solução evolutiva para um projeto já existente. Buscamos aqui propiciar uma solução de nível corporativo.

1. **METODOLOGIA**

A estratégia de pesquisa deste trabalho é qualitativa e, para realizar esse projeto, como o autor já possuía experiência com grande parte das tecnologias envolvidas, primeiro se definiu o contexto do projeto e o que ele precisaria conter de tecnologias, após essa definição se realizou buscas de referências para entender melhor os conceitos e suas nuances, e também foi efetuada a etapa de análise.

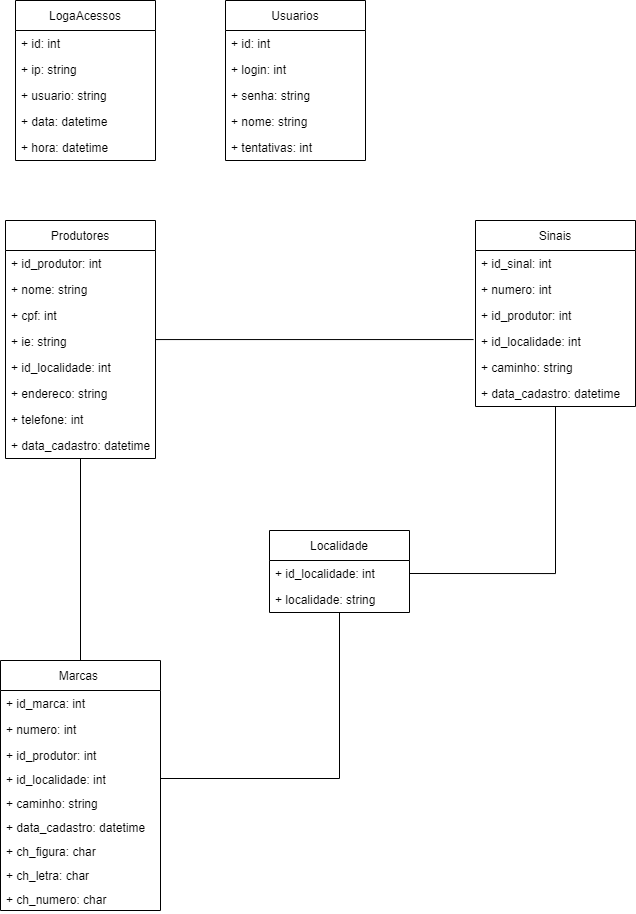
**Figura 1.**



**Fonte: Elaboração Própria.**

Na Figura 1 está representada a arquitetura do tipo monolítica, do sistema CMS em que vamos fazer a migração.

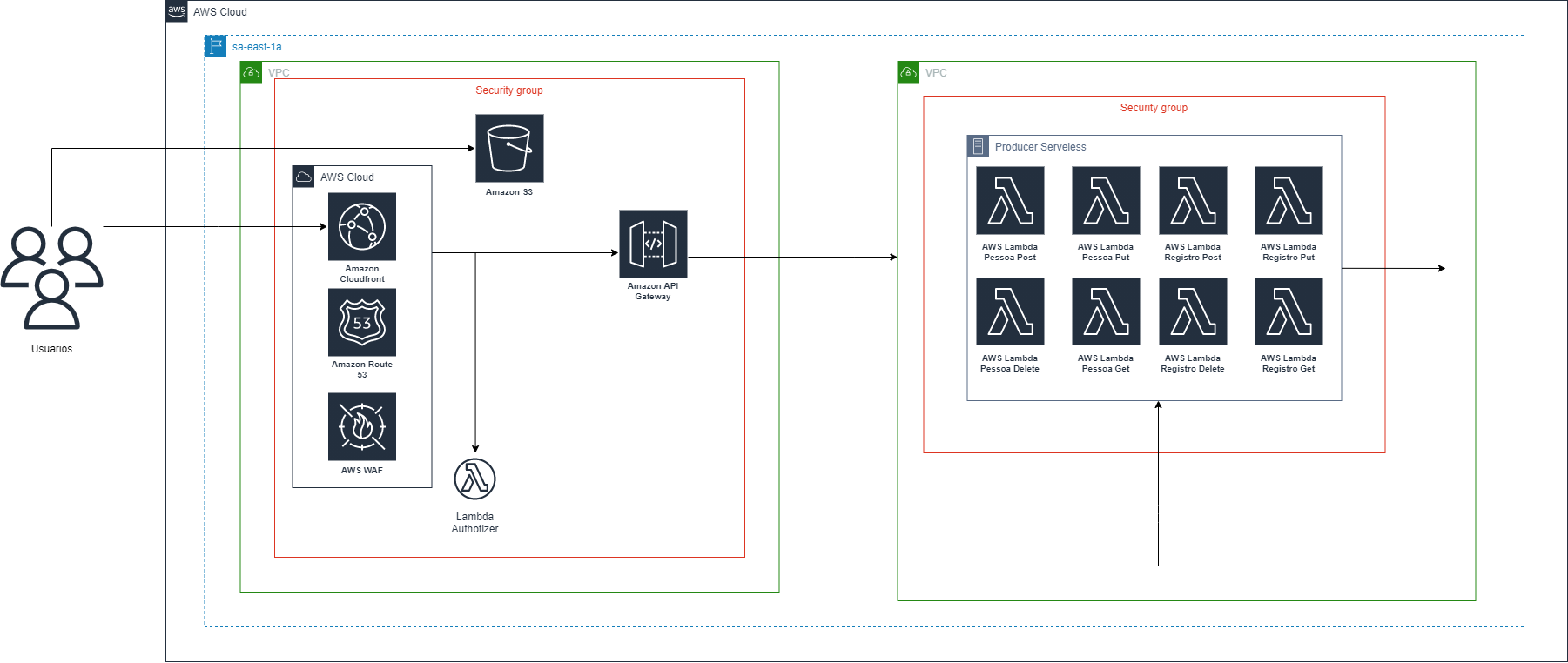
**Figura 2.**

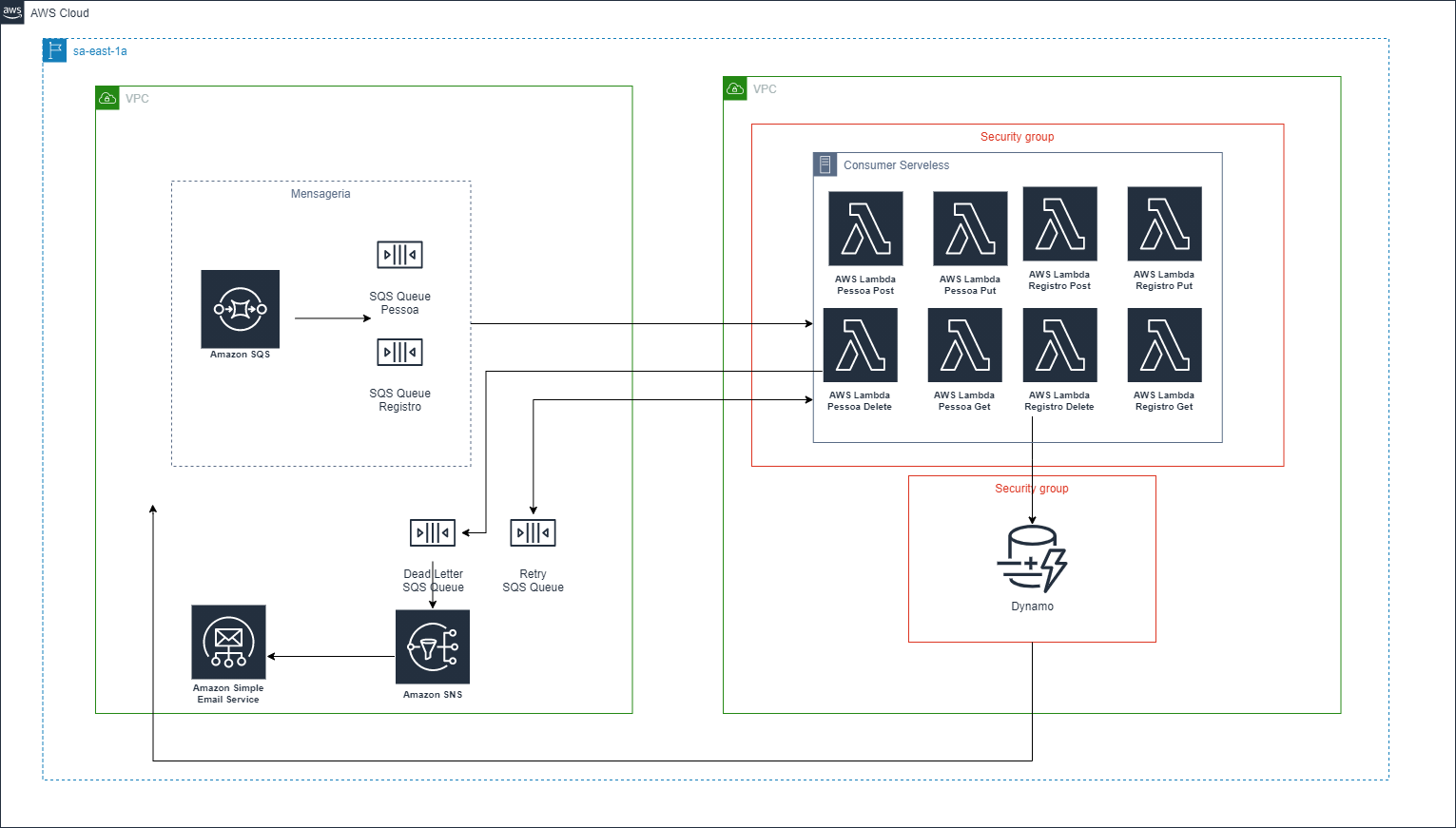


**Fonte: Elaboração Própria.**

Na Figura acima temos UML das entidades de negócio sistema CMS.

**Figura 3.**





**Fonte: Elaboração Própria.**

Na Figura acima a nova arquitetura serverless em micro serviços.

1. **RESULTADOS E DISCUSSÃO**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean luctus ligula ut ex accumsan sagittis. Ut consectetur, eros at lacinia vehicula, metus sem blandit magna, id accumsan lectus tellus eget arcu. Nullam vitae nibh at justo rhoncus congue. Suspendisse a est ut libero semper tincidunt eu sit amet purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Integer porta gravida viverra. Morbi non auctor ex. Donec in ex vel nisi efficitur laoreet rutrum id quam. Curabitur efficitur, arcu id lacinia interdum, urna quam efficitur ante, quis egestas sem diam vitae ante. Praesent placerat blandit ligula quis pretium. Proin accumsan, massa a consectetur eleifend, metus turpis rhoncus tellus, vel convallis mi arcu nec velit. Aliquam nec dictum erat.

Nulla augue diam, mattis sit amet posuere et, hendrerit quis libero. Vivamus eros mi, semper et aliquet non, feugiat vitae neque. Nulla convallis venenatis tellus. Pellentesque faucibus dolor dui, sit amet convallis risus pharetra sit amet. Ut metus erat, dictum in aliquam id, semper et nibh. Sed interdum rutrum orci in efficitur. Maecenas non lobortis purus, sed tincidunt lacus. Aliquam quis dapibus tellus.

Etiam id facilisis orci. Donec ultricies fermentum euismod. Donec gravida feugiat nisl, at commodo ligula viverra at. Praesent sollicitudin tempor lobortis. Cras dapibus diam urna, et pretium nunc tincidunt ac. Nullam vitae placerat lorem. Vivamus metus magna, mattis at urna vitae, feugiat rutrum nisi. Ut eu justo ullamcorper arcu suscipit dictum ut eu libero. Sed suscipit accumsan lacus eget vehicula. Fusce in sapien velit.

1. **CONSIDERAÇÕES FINAIS**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean luctus ligula ut ex accumsan sagittis. Ut consectetur, eros at lacinia vehicula, metus sem blandit magna, id accumsan lectus tellus eget arcu. Nullam vitae nibh at justo rhoncus congue. Suspendisse a est ut libero semper tincidunt eu sit amet purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Integer porta gravida viverra. Morbi non auctor ex. Donec in ex vel nisi efficitur laoreet rutrum id quam. Curabitur efficitur, arcu id lacinia interdum, urna quam efficitur ante, quis egestas sem diam vitae ante. Praesent placerat blandit ligula quis pretium. Proin accumsan, massa a consectetur eleifend, metus turpis rhoncus tellus, vel convallis mi arcu nec velit. Aliquam nec dictum erat.

Nulla augue diam, mattis sit amet posuere et, hendrerit quis libero. Vivamus eros mi, semper et aliquet non, feugiat vitae neque. Nulla convallis venenatis tellus. Pellentesque faucibus dolor dui, sit amet convallis risus pharetra sit amet. Ut metus erat, dictum in aliquam id, semper et nibh. Sed interdum rutrum orci in efficitur. Maecenas non lobortis purus, sed tincidunt lacus. Aliquam quis dapibus tellus.

Etiam id facilisis orci. Donec ultricies fermentum euismod. Donec gravida feugiat nisl, at commodo ligula viverra at. Praesent sollicitudin tempor lobortis. Cras dapibus diam urna, et pretium nunc tincidunt ac. Nullam vitae placerat lorem. Vivamus metus magna, mattis at urna vitae, feugiat rutrum nisi. Ut eu justo ullamcorper arcu suscipit dictum ut eu libero. Sed suscipit accumsan lacus eget vehicula. Fusce in sapien velit.

**REFERÊNCIAS**

BERTALANFY, L. Von. **Teoria Geral dos Sistemas**. Rio de janeiro: Ed. Vozes, 1975.

CARDOSO; MEFFE; MARTINS, S.P. O software Público Brasileiro. Revista Linux Magazine, nº6, 2011.

FERNANDES, Henrique Marques. **O que é sistema/aplicação Monolito/Monolítica?** Disponível em:https://marquesfernandes.com/tecnologia/o-que-e-um-sistema-aplicacao-monolito-monolitica/ Acesso em:

FIGUEIREDO, Elaine G. M. de. **Arquiteturas de sistemas Web 3.0**. 2013. Disponível em: https://www.devmedia.com.br/arquiteturas-de-sistemas-web-3- 0. Acesso em:

GERTEL, Lucas. **Padrões de Arquitetura Web**: Monolítica ou Micro Serviços?.

2019. Disponível em: https://medium.com/@lgertel/padrões-de-arquitetura-web-monolítica-ou-micro-serviços-7b3f0c9394fe. Acesso em:

# Giroldo, Bruna.  Mercado da arquitetura de software. 17, Novembro 2020

# Disponível em: https://posdigital.pucpr.br/blog/mercado-da-arquitetura-de-software

HAQ, Siraj Ul. **Introduction to Monolithic Architecture and MicroServices** **Architecture**. 2018. Disponível em: https://medium.com/koderlabs/introduction-to-

monolithic-architecture-and-microservices-architecture-b211a5955c63. Acesso em:

LEWIS, James; FOWLER, Martin. **Microservices: a definition of this new**

**architectural term.** Disponível em:

https://martinfowler.com/articles/microservices.html. Acesso em:

MENDES, Antonio. **Arquitetura de Software**: Desenvolvimento orientado para arquitetura. Disponível em: https://www.devmedia.com.br/arquitetura-de-software-

desenvolvimento-orientado-para-arquitetura/8033. Acesso em:

NEVES, Joana. **História geral** – a construção de um mundo globalizado. Ed. Saraiva, 2020.

NEWMAN, Sam. **Building Microservices**. 1. Ed. O’Reilly Media, Inc, Gravenstein Highway North, Sebastopol, USA, 2015. Cap. 2, p. 19.

NEWMAN, Sam. Migarndo sistemas monolítico, para micro serviços. Ed.Nova tec, 2020.

# Revista de engenharia de software magazine 63. Disponível em:

# https://www.devmedia.com.br/revista-engenharia-de-software-magazine-63/29391 Acesso em:

**ANEXOS ou APÊNDICES (opcional)**

1. Discente do Curso de Xxxxxx do Centro Universitário Academia – UniAcademia. Endereço: xxxx. Celular: (xx ). E-mail: xxxx. [↑](#footnote-ref-0)
2. Docente do Curso de Xxxxxx do Centro Universitário Academia. Orientador. [↑](#footnote-ref-1)