

UNIVERSIDADE FEDERAL DO PAMPA

Adriano Prado Cavalheiro

# Explorando a Estratégia Serverless First no Desenvolvimento de Aplicativos em Nuvem

Alegrete  
2024



Adriano Prado Cavalleiro

# Explorando a Estratégia Serverless First no Desenvolvimento de Aplicativos em Nuvem

Dissertação apresentado ao Programa de Pós-graduação Stricto Sensu em Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia de Software.

Orientador: Prof. Dr. Claudio Schepke

Alegrete  
2024

Ficha catalográfica elaborada automaticamente com os dados fornecidos pelo(a) autor(a) através do Módulo de Biblioteca do Sistema GURI (Gestão Unificada de Recursos Institucionais) .

C376e Cavalheiro, Adriano Prado  
Explorando a Estratégia Serverless First no Desenvolvimento de Aplicativos em Nuvem / Adriano Prado Cavalheiro.  
51 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa, MESTRADO EM ENGENHARIA DE SOFTWARE, 2024.  
"Orientação: Claudio Schepke".

1. Serverless First. 2. Desenvolvimento de Aplicativos em Nuvem. 3. Arquitetura Serverless. 4. Desenvolvimento de Software. 5. Computação em Nuvem. I. Título.

**ADRIANO PRADO CAVALHEIRO**

**EXPLORANDO A ESTRATÉGIA SERVERLESS FIRST NO DESENVOLVIMENTO DE  
APLICATIVOS EM NUVEM**

Dissertação apresentada ao  
Programa de Engenharia de  
Software da Universidade Federal  
do Pampa, como requisito parcial  
para obtenção do Título de  
Mestre em Engenharia de Software

Dissertação defendida e aprovada em: 19/08/2024

Banca examinadora:

---

Prof. Dr. Claudio Schepke

Orientador

Unipampa

---

Prof. Dr. Diego Luis Kreutz

Unipampa

---

Prof. Dr. Rodrigo Brandão Mansilha  
Unipampa

---

Prof. Dr. Adriano José Vogel  
Johannes Kepler Universität Linz, Austria

---



Assinado eletronicamente por **CLAUDIO SCHEPKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 19/08/2024, às 16:26, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **RODRIGO BRANDAO MANSILHA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 19/08/2024, às 16:26, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **DIEGO LUIS KREUTZ, PROFESSOR DO MAGISTERIO SUPERIOR**, em 19/08/2024, às 22:30, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



Assinado eletronicamente por **Adriano Vogel, Usuário Externo**, em 20/08/2024, às 17:21, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

---



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1516494** e o código CRC **38DED575**.

---

Às meninas da minha vida, Kelly e Rebeca.



## AGRADECIMENTOS

Expresso aqui os meus sinceros agradecimentos a todas as pessoas e instituições que contribuíram direta ou indiretamente para a realização deste trabalho.

Primeiramente, gostaria de agradecer ao meu orientador, Claudio Schepke, pela orientação dedicada, apoio e incentivo ao longo de todo o processo. Sua expertise e comprometimento foram fundamentais para o desenvolvimento desta dissertação. Também sou grato a UNIPAMPA e ao programa de mestrado PPGES da UNIPAMPA por me proporcionar a oportunidade de realizar este estudo. Agradeço à banca de seleção do programa por aprovar minha entrada no curso.

Além disso, expresso minha gratidão aos meus colegas de trabalho do NTI-DIADS UFCSPA, principalmente a Cristina Almeida. Estendo os meus agradecimentos aos professores que ministraram as disciplinas ao longo do programa, bem como aos meus colegas de curso. O compartilhamento de conhecimento e encorajamento contribuíram para o meu crescimento acadêmico e profissional.

Em especial, minha profunda gratidão à minha esposa Kelly Daiana Vargas Serpa Cavaleiro. Sua paciência, apoio e compreensão foram essenciais durante todo o período de elaboração desta dissertação. Além disso, agradeço por me presentear com a maior dádiva de toda a minha existência, nossa amada filha Rebeca Serpa Cavaleiro, que veio ao mundo em abril de 2023, trazendo uma imensa alegria e iluminando nossas vidas.

Muito obrigado a todos.



“O Universo não parece nem benigno nem hostil,  
mas meramente indiferente às preocupações  
de criaturas tão insignificantes como nós ”

- Carl Sagan



## RESUMO

A evolução do desenvolvimento em nuvem tem sido marcada pela emergência da computação *Serverless*, uma abordagem que modifica as práticas tradicionais de desenvolvimento de *software*. Esta dissertação explora as potenciais vantagens da adoção da estratégia *Serverless First*, com um foco particular na experiência do desenvolvedor, abordando a facilidade de implementação e a eficiência do processo de desenvolvimento. Por meio de uma abordagem multidisciplinar que inclui revisão sistemática da literatura, revisão integrativa e um experimento controlado, este estudo investiga como a computação *Serverless* pode simplificar o desenvolvimento ao reduzir as responsabilidades de gerenciamento de infraestrutura. Os resultados, baseados em dados qualitativos obtidos das respostas dos participantes, revelam que a estratégia *Serverless First* permite uma maior concentração na lógica de aplicação, melhorando a produtividade e possibilitando inovação. Os resultados fornecem uma compreensão sobre os impactos da computação *Serverless* nas práticas de desenvolvimento de *software*, destacando tanto os benefícios quanto os desafios associados.

**Palavras-chave:** Serverless First. Desenvolvimento de Aplicativos em Nuvem. Arquitetura Serverless. Aplicações sem Servidor. Desenvolvimento de Software. Computação em Nuvem.



## ABSTRACT

The emergence of serverless computing marks the evolution of cloud development, an approach that alters traditional software development practices. This dissertation explores the potential advantages of adopting the Serverless First strategy, focusing on the developer's experience, the ease of implementation, and the efficiency of the development process. This study investigates how serverless computing can simplify development through a multidisciplinary approach that includes a systematic literature review, an integrative review, and a controlled experiment. The results, based on qualitative data obtained from participant responses, reveal that the Serverless First strategy allows a focus on application logic by reducing infrastructure management responsibilities, improving productivity, and enabling innovation. The findings provide insights into the impacts of serverless computing on software development practices, highlighting both the associated benefits and challenges.

**Key-words:** Serverless First. Cloud Application Development. Serverless Architecture. Serverless Applications. Software Development. Cloud Computing.



## LISTA DE FIGURAS

Figura 1 – Representação esquemática de um exemplo da arquitetura <i>Serverless</i> . . . . .	25
Figura 2 – Resultados da Percepção dos Participantes . . . . .	36



## LISTA DE TABELAS

Tabela 1 – Termos . . . . .	44
Tabela 2 – Bases de dados . . . . .	44
Tabela 3 – Estudos recuperados . . . . .	45
Tabela 4 – Fase de Seleção . . . . .	46
Tabela 5 – Fase de Extração . . . . .	46
Tabela 6 – Impactos nos Custos . . . . .	48



## LISTA DE SIGLAS

**API** *Application Programming Interface*

**AWS** Amazon Web Services

**CPU** *Central Processing Unit*

**FaaS** *Function as a Service*

**GCP** Google Cloud Platform

**I/O** *Input/Output*

**IaaS** *Infrastructure as a Service*

**JSON** *JavaScript Object Notation*

**RSL** Revisão Sistemática da Literatura

**SAM** *Serverless Application Model*

**VM** *Virtual Machine*

**Web** *World Wide Web*



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>23</b>
1.1	Objetivo . . . . .	23
1.2	Justificativa . . . . .	24
1.3	Organização do Trabalho . . . . .	24
<b>2</b>	<b>ASPECTOS CONCEITUAIS . . . . .</b>	<b>25</b>
2.1	Computação Serverless . . . . .	25
2.2	Benefícios da Computação Serverless . . . . .	25
2.3	Desafios da Computação Serverless . . . . .	26
2.4	Estratégia Serverless First . . . . .	26
2.5	Considerações Finais sobre o Capítulo . . . . .	26
<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>27</b>
3.1	Análise dos Trabalhos Relacionados . . . . .	27
3.2	Benefícios de <i>Serverless Computing</i> . . . . .	29
3.3	Desafios Serverless . . . . .	30
3.4	Considerações Finais sobre o Capítulo . . . . .	30
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>31</b>
4.1	Visão Geral dos Frameworks Serverless . . . . .	31
4.1.1	Amazon Web Services (AWS) <i>Serverless Application Model</i> (SAM) . . . . .	31
4.1.2	Serverless Framework . . . . .	31
4.1.3	Pulumi . . . . .	31
4.2	Design Experimental . . . . .	32
4.3	Variáveis . . . . .	32
4.4	Participantes . . . . .	32
4.5	Procedimento . . . . .	32
4.6	Coleta de Dados . . . . .	33
4.7	Análise de Dados . . . . .	33
4.8	Considerações Éticas . . . . .	33
4.9	Ameaças à Validade . . . . .	33
4.10	Considerações Finais sobre o Capítulo . . . . .	34
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>35</b>
5.1	Percepção da Dificuldade da Implementação . . . . .	35
5.2	Benefícios Percebidos da Estratégia Serverless First . . . . .	35
5.3	Diminuição do Tempo de Produção . . . . .	35
5.4	Implementação Simplificada . . . . .	35
5.5	Recomendação Serverless First . . . . .	36

5.6	Sugestões para Melhorias . . . . .	36
5.7	Considerações Finais sobre o Capítulo . . . . .	37
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>39</b>
6.1	Contribuição . . . . .	39
6.2	Sugestões para Trabalhos Futuros . . . . .	39
6.3	Publicação . . . . .	40
	<b>REFERÊNCIAS . . . . .</b>	<b>41</b>
	<b>ANEXO A – REVISÃO DA LITERATURA . . . . .</b>	<b>43</b>
A.1	Definição do Protocolo . . . . .	43
A.1.1	Revisão Sistemática da Literatura . . . . .	43
A.1.2	Revisão Integrativa . . . . .	43
A.1.3	Questões de Pesquisa . . . . .	43
A.1.4	Estratégia de Busca . . . . .	44
A.1.5	Critérios de Inclusão e Exclusão . . . . .	44
A.1.6	Questões de Qualidade . . . . .	45
A.2	Análise e Síntese dos Dados . . . . .	45
A.2.1	Fase de Seleção dos Estudos . . . . .	45
A.2.2	Fase de Extração dos Dados Relevantes . . . . .	45
A.3	Considerações Finais sobre o Capítulo . . . . .	46
	<b>ANEXO B – RELATÓRIO: IMPACTOS DA ABORDAGEM SERVERLESS EM EMPRESAS DA AMÉRICA LATINA . . . . .</b>	<b>47</b>
B.1	Impactos nos Custos . . . . .	47
B.2	Impacto no Desempenho e Disponibilidade . . . . .	47
B.3	Facilidade de Implementação . . . . .	47
B.4	Manutenção e Redução de Pessoal . . . . .	47
B.5	Serviços Serverless Mais Utilizados . . . . .	48
B.6	Tipos de Aplicações Mais Adequadas . . . . .	48
B.7	Conclusão . . . . .	49
	<b>Índice . . . . .</b>	<b>51</b>

## 1 INTRODUÇÃO

O desenvolvimento em nuvem tem se consolidado como uma estratégia eficaz para empresas que buscam otimizar seus processos de criação de aplicativos, aproveitando os recursos e serviços da computação em nuvem (RUPARELIA, 2016). Essa abordagem não só promove eficiência e economia na gestão de dados, como também impulsiona a inovação e a agilidade empresarial (CRONAPP, 2019).

Nesse contexto, a computação sem servidor, ou *serverless computing*, caracteriza-se por permitir o desenvolvimento e a implantação de código sem a necessidade de gerenciar a infraestrutura subjacente, o que resulta em maior flexibilidade e eficiência (CLOUDFLARE, 2024). A arquitetura *serverless* se diferencia por seu modelo de execução baseado em eventos, onde os recursos são alocados dinamicamente conforme a demanda, eliminando a necessidade de provisionamento prévio de servidores. A estratégia *Serverless First* destaca-se como uma metodologia que prioriza a computação sem servidor como a escolha principal para o desenvolvimento em nuvem, maximizando os benefícios deste modelo de computação no desenvolvimento de *software* (VAUGHAN, 2023).

### 1.1 Objetivo

Esta dissertação investiga as vantagens da estratégia *Serverless First*, com foco na experiência do desenvolvedor. O estudo avalia a facilidade de implementação e a eficiência do desenvolvimento ao adotar essa estratégia (GILBERT, 2024). A hipótese central é de que a adoção da estratégia *Serverless First* proporciona uma experiência de desenvolvimento mais simplificada e eficiente, em comparação com o modelo tradicional de desenvolvimento de *software* que envolve a gerência de servidores.

Para alcançar esse objetivo, foi adotada uma abordagem multidisciplinar, que inclui revisão da literatura e experimento controlado. Este estudo envolve a implementação e avaliação de uma *Application Programming Interface* (API) de gestão de tarefas, que abrange operações básicas de manipulação de dados e persistência em banco de dados. A aplicação foi desenvolvida utilizando três *frameworks serverless*: AWS SAM<sup>1</sup>, Serverless Framework<sup>2</sup> e Pulumi<sup>3</sup>. O experimento foi conduzido com 18 estudantes de graduação em Ciência da Computação, avaliando métricas qualitativas relacionadas à experiência do desenvolvedor ao utilizar a estratégia *Serverless First*.

Este trabalho contribui para o contexto da arquitetura *Serverless*, fornecendo uma compreensão sobre essa estratégia no desenvolvimento de *software*. Os resultados servem como base para profissionais de tecnologia, tomadores de decisão e pesquisadores interessados em explorar e adotar a estratégia *Serverless First* em seus projetos.

Este estudo limita-se à estrutura de Amazon Web Services (AWS) para a imple-

<sup>1</sup> <https://aws.amazon.com/pt/serverless/sam/>

<sup>2</sup> <https://www.serverless.com/framework>

<sup>3</sup> <https://www.pulumi.com/>

mentação *Serverless*. A AWS é uma das principais plataformas de computação em nuvem e oferece uma gama de serviços para suportar arquiteturas *Serverless*, tornando-a uma escolha adequada para esta dissertação (Gartner, 2023).

## 1.2 Justificativa

Este trabalho fundamenta-se na necessidade de aprofundar o entendimento sobre a computação sem servidor. Para isso, foram avaliados os benefícios e desafios da adoção da estratégia *Serverless First* no desenvolvimento de aplicativos em nuvem, com foco na experiência do desenvolvedor. A investigação fornece uma análise sobre como essa abordagem pode simplificar e tornar mais eficiente o processo de desenvolvimento, em comparação com o modelo que envolve a gerência de servidores.

## 1.3 Organização do Trabalho

O restante do documento está estruturado da seguinte forma. O Capítulo 2 discute conceitos relacionados à computação sem servidor e detalhes sobre a estratégia *Serverless First*. O Capítulo 3 apresenta os trabalhos relacionados ao objeto da pesquisa. O Capítulo 4 detalha a metodologia empregada na condução do experimento. Os resultados do experimento e sua análise são apresentados no Capítulo 5. Por fim, o Capítulo 6 apresenta as considerações finais e sugere direções para pesquisas futuras.

## 2 ASPECTOS CONCEITUAIS

Neste capítulo apresentamos conceitos relacionados à computação *Serverless* e, em particular, abordaremos a estratégia *Serverless First*. É importante compreender esses conceitos para contextualizar o estudo e sua relevância para a indústria e meio acadêmico.

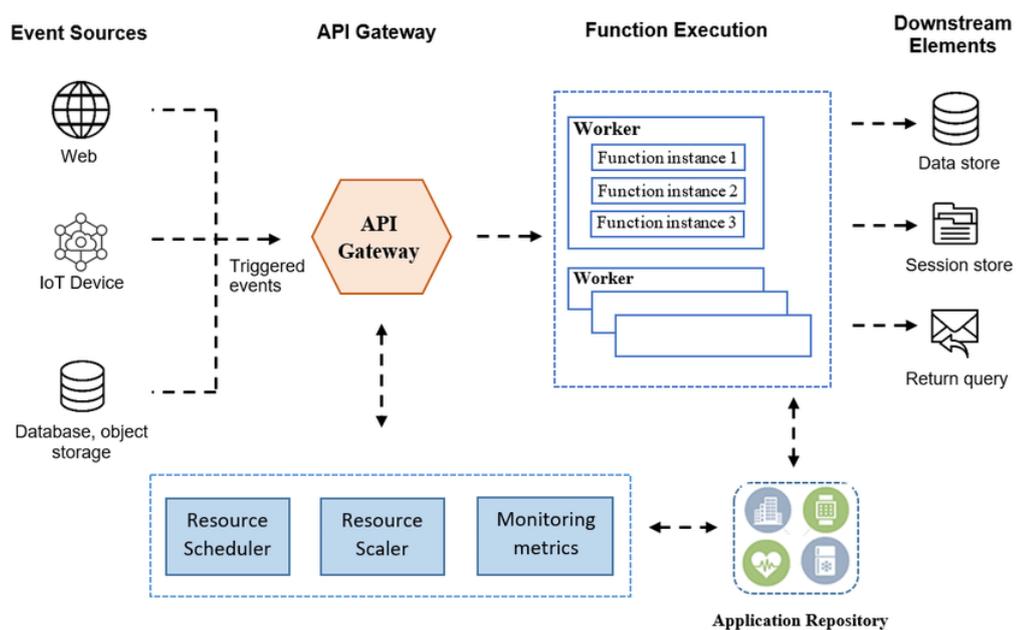
### 2.1 Computação Serverless

A computação *Serverless* é um modelo de computação em nuvem que se caracteriza pela execução de código sob demanda, onde o custo é baseado no tempo de execução efetivo, desvinculando o usuário da gestão de servidores ou infraestrutura física (EISMANN et al., 2021; SCHLEIER-SMITH et al., 2021). Neste modelo, os provedores de serviços de nuvem administram a infraestrutura, permitindo que desenvolvedores concentrem-se exclusivamente na lógica de aplicação. A Figura 1 mostra uma arquitetura de computação *serverless*, mostrando a interação entre componentes como fontes de eventos, execução de funções e gerenciamento de recursos, que juntos possibilitam o processamento de solicitações e a alocação de recursos (MAMPAGE; KARUNASEKERA; BUYYA, 2022).

### 2.2 Benefícios da Computação Serverless

Estudos tanto no âmbito industrial quanto acadêmico sugerem vantagens associadas à computação *Serverless*, entre as quais destacam-se:

Figura 1 – Representação esquemática de um exemplo da arquitetura *Serverless*.



Fonte: adaptado de (MAMPAGE; KARUNASEKERA; BUYYA, 2022)

- **Redução de Custos Operacionais:** pesquisas indicam que a computação *Serverless* reduz os custos, sobretudo em aplicações com demandas variáveis de carga de trabalho (NDAY; KUSUMA; FREDYAN, 2023; ADZIC; CHATLEY, 2017).
- **Agilidade no Desenvolvimento:** a literatura destaca que esta abordagem permite aos desenvolvedores concentrar-se na lógica de negócios, agilizando o ciclo de desenvolvimento e na entrega de funcionalidades (SCHLEIER-SMITH et al., 2021).
- **Escalabilidade e Elasticidade Automáticas:** estudos apontam que *Serverless* proporciona escalabilidade e elasticidade automáticas, ajustando os recursos conforme necessário, sem intervenção manual (IVAN; VASILE; DADARLAT, 2019).

### 2.3 Desafios da Computação Serverless

A adoção da computação *Serverless* não está isenta de desafios, como:

- **Problemas de Latência** como *Cold starts*<sup>1</sup> podem impactar negativamente o desempenho em aplicações que requerem alta disponibilidade (BARDSLEY; RYAN; HOWARD, 2018).
- **Dependência de Provedores** limitam a portabilidade entre diferentes plataformas devido à funcionalidades específicas dos provedores (WEN et al., 2021).
- **Curva de Aprendizado** exige que as equipes de projeto e operações adaptem-se a novas ferramentas e práticas (SCHLEIER-SMITH et al., 2021; Elastic, 2024).

### 2.4 Estratégia Serverless First

A estratégia *Serverless First* sugere que as soluções *Serverless* sejam consideradas como o ponto de partida para o desenvolvimento de novas funcionalidades e serviços, focando na lógica de negócios ao invés de infraestrutura (VAUGHAN, 2023).

### 2.5 Considerações Finais sobre o Capítulo

Este capítulo discutiu conceitos da computação *Serverless* e a estratégia *Serverless First*, salientando os benefícios e desafios associados. A computação *Serverless* oferece benefícios como elasticidade e eficiência, embora traga desafios como dependência de fornecedores. A abordagem *Serverless First* prioriza esse modelo de computação para desenvolvimento de aplicativos na nuvem.

---

<sup>1</sup> *Cold starts:* referem-se ao atraso inicial que ocorre quando uma função *serverless* é invocada pela primeira vez ou após ficar inativa por algum tempo. Durante este período, a infraestrutura de execução (por exemplo, contêineres ou máquinas virtuais) precisa ser inicializada, o que pode levar a um aumento temporário no tempo de resposta.

### 3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma revisão dos trabalhos relacionados à pesquisa desenvolvida nesta dissertação. Através dessa revisão, buscamos obter uma visão sobre a computação sem servidor, principalmente sobre seus benefícios e desafios no desenvolvimento de aplicações em nuvem.

#### 3.1 Análise dos Trabalhos Relacionados

Nesta seção, apresentamos uma análise dos trabalhos relacionados identificados durante a revisão da literatura, conforme descrito em Apêndice A. Os estudos selecionados abordam diferentes aspectos da computação sem servidor, desde sua implementação em sistemas médicos até a análise de desempenho em plataformas de processamento de vídeo. Essa análise é importante para situar a pesquisa desenvolvida neste trabalho e destacar as contribuições relevantes no campo da computação sem servidor. A seguir, listamos cada trabalho juntamente com um breve resumo e suas fontes correspondentes.

- **Estudo 01:** *Design and Implementation of Medical Searching System Based on Microservices and Serverless Architectures.* Este estudo apresenta o design e a implementação do sistema de busca médica ScanMedicine, que visa fornecer acesso fácil a inovações em tecnologias de saúde para profissionais da área, pacientes e pesquisadores. O sistema é implementado utilizando funções AWS Lambda e uma arquitetura baseada em microsserviços. Os autores adotaram um estilo arquitetônico em camadas, onde cada camada opera em *hardware* separado e utiliza diferentes técnicas arquitetônicas, destacando a eficiência e a escalabilidade proporcionadas por essas tecnologias (SADEK; CRAIG; TRENELL, 2022).
- **Estudo 02:** *Serverless Utilization in Microservice E-learning platform.* O artigo propõe uma solução híbrida que alterna entre a utilização de *Infrastructure as a Service* (IaaS) e *Function as a Service* (FaaS) em uma plataforma de microsserviços. A abordagem visa otimizar custos ao alternar entre servidores privados virtuais em horários de alta demanda e funções *serverless* em períodos de baixa demanda. O estudo conclui que essa estratégia pode reduzir os custos operacionais em pelo menos 25%, mantendo um desempenho comparável ao de aplicativos de microsserviços implantados em instâncias (NDAY; KUSUMA; FREDYAN, 2023).
- **Estudo 03:** *A FaaS File System for Serverless Computing.* Este estudo apresenta um sistema de arquivos compartilhado projetado para funções *serverless*, que oferece suporte a gerenciamento de estado com semântica POSIX familiar. Os autores exploram como o sistema aproveita a resiliência inerente às funções *serverless* para alcançar escalabilidade e desempenho, permitindo que aplicações com estado se

beneficiem da simplicidade e escalabilidade da computação *serverless*, muitas vezes com pouca ou nenhuma modificação (SCHLEIER-SMITH et al., 2020).

- **Estudo 04:** *Building a Chatbot with Serverless Computing*. Este artigo apresenta a arquitetura e o protótipo de um *chatbot* construído em uma plataforma *serverless*, onde funções sem estado são compostas para realizar ações e coordenar microsserviços cognitivos. Os autores demonstram como a arquitetura *serverless* melhora a extensibilidade e a escalabilidade do *chatbot*, facilitando a integração com serviços externos e a manutenção contínua (YAN et al., 2016).
- **Estudo 05:** *Evaluation of Production Serverless Computing Environments*. Este artigo avalia ambientes de computação *serverless* em produção, focando em métricas como *throughput*<sup>1</sup>, largura de banda de rede, desempenho de *Input/Output* (I/O) de arquivos e desempenho computacional em invocações simultâneas. Os autores comparam a computação *serverless* com VMs, destacando a elasticidade, a eficiência de custos e a utilização de recursos para aplicações dinâmicas e processamento distribuído (LEE; SATYAM; FOX, 2018).
- **Estudo 06:** *Serverless Computing: Economic and Architectural Impact*. Este artigo apresenta dois estudos de caso industriais de empresas que adotaram a arquitetura AWS Lambda, destacando como a migração para essa arquitetura reduziu os custos de hospedagem entre 66% e 95%. Os autores discutem como a adoção da computação *serverless* pode influenciar as práticas comuns de *design* de *software*, potencialmente mudando a forma como as aplicações cliente/servidor são projetadas, desenvolvidas e operadas (ADZIC; CHATLEY, 2017).
- **Estudo 07:** *An Empirical Study on Challenges of Application Development in Serverless Computing*. Este artigo apresenta um estudo abrangente sobre os desafios enfrentados pelos desenvolvedores na criação de aplicações baseadas em computação *serverless*. Através da análise de 22.731 questões do Stack Overflow<sup>2</sup>, os autores identificam uma tendência crescente de popularidade da computação *serverless*, mas também destacam a alta dificuldade envolvida. O estudo constrói uma taxonomia dos principais desafios enfrentados e fornece implicações práticas para desenvolvedores, pesquisadores e provedores de nuvem, visando melhorar a compreensão e o desenvolvimento de aplicações *serverless* (WEN et al., 2021).
- **Estudo 08:** *Serverless computing: An Investigation of Deployment Environments for Web APIs*. Este artigo investiga o uso da computação *serverless* para a implementação Web APIs, comparando-a com a implantação em VM. Os autores

<sup>1</sup> *Throughput*: refere-se à quantidade de dados ou transações que um sistema é capaz de processar em um determinado período de tempo. É um indicador de desempenho que mede a eficiência com que um sistema processa ou transmite informações.

<sup>2</sup> <https://stackoverflow.com/>

analisam o desempenho, tempos de resposta e custos em diferentes cenários, incluindo variações no número de usuários simultâneos. O estudo conclui que, embora a computação *serverless* ofereça escalabilidade automática e seja eficaz em lidar com aumentos inesperados de usuários, a escolha entre *serverless* e VM depende de uma análise cuidadosa dos custos e das necessidades específicas da aplicação (IVAN; VASILE; DADARLAT, 2019).

- **Estudo 09:** *Serverless computing provides on-demand high performance computing for biomedical research.* Este artigo explora o uso da computação *serverless* para pesquisa biomédica, destacando sua capacidade de fornecer acesso sob demanda a centenas de CPUs com escalabilidade instantânea e baixo custo. Os autores demonstram, por meio de um exemplo prático, que é possível realizar comparações par-a-par entre todas as proteínas humanas em aproximadamente 2 minutos, a um custo inferior a 1 dólar, utilizando o AWS Lambda. Isso representa uma aceleração de 250 vezes em comparação com a execução da mesma tarefa em um *laptop* típico (KUMANOV et al., 2018).
- **Estudo 10:** *Serverless performance and optimization strategies.* Este trabalho avalia o perfil de desempenho de arquiteturas *serverless* utilizando AWS Lambda em contextos de baixa latência e alta disponibilidade. Os autores discutem os desafios relacionados a *cold starts* e características de latência, além de propor um conjunto diversificado de estratégias e técnicas para otimizar o desempenho dessas arquiteturas. O objetivo é fornecer padrões de *design* que ampliem a aplicabilidade da computação *serverless* a um conjunto mais amplo de domínios (BARDSLEY; RYAN; HOWARD, 2018).
- **Estudo 11:** *Video processing with serverless computing: A measurement study.* Este estudo investiga a configuração e implementação de funções *serverless* para processamento de vídeo, destacando como essas escolhas influenciam o tempo de execução e o custo monetário. Os autores enfatizam a importância da configuração adequada de memória e sugerem que, em alguns casos, implementar o processamento de vídeo diretamente nas funções *serverless* pode ser competitivo em relação ao uso de APIs externas. Além disso, o estudo identifica que o desempenho dessas aplicações pode ser impactado pela infraestrutura subjacente (ZHANG et al., 2019).

### 3.2 Benefícios de Serverless Computing

Com base nos trabalhos relacionados, pode-se destacar as seguintes vantagens:

- **Eliminação da necessidade de gerenciar infraestrutura:** a computação *Serverless* permite que os desenvolvedores concentrem-se na lógica de negócios e na

implementação de funcionalidades, reduzindo a necessidade de gerenciar servidores e infraestrutura (SADEK; CRAIG; TRENELL, 2022).

- **Escalabilidade automática e elasticidade:** os sistemas *Serverless* oferecem escalabilidade automática, ajustando os recursos conforme necessário para lidar com variações na carga de trabalho e picos de demanda (NDAY; KUSUMA; FREDYAN, 2023; IVAN; VASILE; DADARLAT, 2019).
- **Redução de custos operacionais:** a alternância entre servidores tradicionais e funções *Serverless* pode resultar em economia de custos, dependendo da aplicação, especialmente em casos com variação na carga de trabalho (NDAY; KUSUMA; FREDYAN, 2023; ADZIC; CHATLEY, 2017).
- **Facilidade de implementação e modularização:** a abordagem *Serverless* facilita a implementação de funções autocontidas e modulares, o que pode simplificar o desenvolvimento e a manutenção de aplicações (YAN et al., 2016; SCHLEIER-SMITH et al., 2020).

### 3.3 Desafios Serverless

A computação *Serverless* também apresenta desafios, que incluem:

- **Desempenho em ambientes de baixa latência:** a computação *Serverless* pode enfrentar desafios relacionados a *cold starts* e latência aumentada, que podem impactar o desempenho em sistemas com altos requisitos de disponibilidade e baixa latência (BARDSLEY; RYAN; HOWARD, 2018).
- **Configuração e gerenciamento de recursos:** a configuração adequada dos recursos é essencial para garantir o desempenho das aplicações. Gerenciar efetivamente a comunicação entre componentes distribuídos e o uso de memória são aspectos críticos para otimização (ZHANG et al., 2019; LEE; SATYAM; FOX, 2018).
- **Dependência de provedores e portabilidade:** a utilização de provedores de nuvem para funções *Serverless* pode limitar a portabilidade das aplicações, uma vez que cada provedor pode oferecer diferentes APIs e funcionalidades específicas (WEN et al., 2021).

### 3.4 Considerações Finais sobre o Capítulo

Este capítulo proporcionou uma síntese dos trabalhos relacionados à pesquisa sobre computação *Serverless*, destacando tanto os benefícios quanto os desafios associados a essa abordagem. As informações apresentadas contribuem para a contextualização e fundamentação teórica do estudo realizado.

## 4 METODOLOGIA

Este estudo envolve a implementação e avaliação de uma aplicação de gestão de tarefas utilizando três *frameworks serverless* distintos. O objetivo foi avaliar a experiência do desenvolvedor ao utilizar a estratégia *Serverless First*.

### 4.1 Visão Geral dos Frameworks Serverless

Os *frameworks serverless* selecionados para a implementação da aplicação foram: AWS *Serverless Application Model* (SAM), Serverless Framework e Pulumi. Cada um desses *frameworks* oferece diferentes abordagens e características para o desenvolvimento de aplicações *serverless*.

#### 4.1.1 AWS SAM

O AWS SAM é um *framework* nativo da AWS que facilita a definição e a implantação de aplicações *serverless*. Ele permite que os desenvolvedores descrevam a infraestrutura necessária utilizando um arquivo *template* em formato *YAML*<sup>1</sup>. O SAM é integrado ao ecossistema da AWS, oferecendo suporte para serviços como Lambda<sup>2</sup>, API Gateway<sup>3</sup> e DynamoDB<sup>4</sup>.

#### 4.1.2 Serverless Framework

O *Serverless Framework* é uma ferramenta de código aberto que oferece suporte para o desenvolvimento e a implantação de aplicações *serverless* em múltiplas plataformas de nuvem, incluindo AWS, Google Cloud Platform (GCP) e Azure. Ele utiliza um arquivo de configuração (*serverless.yml*) para definir os recursos e as funções que compõem a aplicação. O Serverless Framework permite o uso de *plugins* para adicionar funcionalidades.

#### 4.1.3 Pulumi

O Pulumi é um *framework* que permite a definição de infraestrutura utilizando linguagens de programação como Python, JavaScript ou TypeScript, em vez de arquivos *template YAML* ou *JavaScript Object Notation* (JSON). Ele oferece suporte para múltiplas plataformas de nuvem, incluindo AWS, Azure e GCP. O Pulumi permite que a infraestrutura seja definida juntamente com a lógica de aplicação.

---

<sup>1</sup> <https://yaml.org/>

<sup>2</sup> <https://aws.amazon.com/pt/lambda/>

<sup>3</sup> <https://aws.amazon.com/pt/api-gateway/>

<sup>4</sup> <https://aws.amazon.com/dynamodb/>

## 4.2 Design Experimental

Os participantes do experimento seguiram um tutorial específico para cada *framework*, que serviu como nivelador antes de começarem a implementação. Eles desenvolveram uma aplicação de gestão de tarefas, que abrange operações básicas de manipulação de dados e persistência em banco de dados, utilizando a estratégia *Serverless First*, onde a API foi inteiramente construída em *serverless* na AWS. O propósito foi estimar a utilidade dessa estratégia sob a perspectiva da experiência do desenvolvedor, com foco na facilidade de implementação e percepção da eficiência.

## 4.3 Variáveis

- **Variável Independente:** estratégia de implementação (*Serverless First*).
- **Variáveis Dependentes:** facilidade de implementação, eficiência percebida no desenvolvimento, experiência geral do desenvolvedor.

## 4.4 Participantes

Os participantes foram 18 estudantes de graduação em Ciência da Computação da Unipampa, que implementaram uma aplicação de gestão de tarefas utilizando os três *frameworks serverless* definidos. Este grupo foi selecionado para proporcionar uma visão prática e diversificada da experiência do desenvolvedor ao adotar a estratégia *Serverless First*. O experimento foi realizado no primeiro semestre letivo de 2024.

## 4.5 Procedimento

O experimento foi conduzido através das seguintes etapas:

1. Introdução aos participantes sobre os objetivos do experimento e a estratégia *Serverless First*.
2. Treinamento dos participantes através de tutoriais específicos para cada *framework*.
3. Desenvolvimento da aplicação de gestão de tarefas utilizando cada *framework* em um ambiente *serverless*.
4. Aplicação de um questionário aos participantes para avaliar suas percepções sobre a estratégia *Serverless First*.
5. Coleta de dados qualitativos sobre a facilidade de implementação, percepção de eficiência e experiência geral com cada *framework*.

## 4.6 Coleta de Dados

O estudo coletou dados qualitativos por meio de questionários aplicados aos participantes após a implementação da aplicação.

- Percepção sobre a facilidade de instalação e configuração inicial.
- Percepção sobre a clareza da documentação e utilidade dos tutoriais.
- Dificuldades enfrentadas na definição de infraestrutura e recursos.
- Percepção sobre a simplicidade e legibilidade do código.
- Benefícios percebidos da estratégia *Serverless First*.
- Recomendações sobre o uso de *Serverless First* para outros desenvolvedores.
- Impacto na experiência do desenvolvedor ao utilizar *Serverless First*.

## 4.7 Análise de Dados

A análise dos dados envolveu:

- **Análise Qualitativa:** Avaliação das percepções dos participantes para identificar as vantagens, desvantagens e experiências associadas à estratégia *Serverless First*, em comparação a sua experiência prévia com outras abordagens de desenvolvimento de *software*.

## 4.8 Considerações Éticas

Este experimento foi projetado de acordo com diretrizes éticas. Todos os participantes foram informados sobre os objetivos do estudo e emitiram seu consentimento para participar. Importante ressaltar que a participação no experimento não era obrigatória; os alunos tinham a opção de não participar sem que isso implicasse em qualquer prejuízo acadêmico ou pessoal.

## 4.9 Ameaças à Validade

Ameaças à validade do estudo identificadas:

- **Tamanho da Amostra:** o estudo foi realizado com uma amostra limitada a 18 estudantes, o que pode restringir a generalização dos resultados.
- **Evolução Tecnológica:** a rápida evolução das tecnologias *serverless* pode limitar a relevância dos resultados ao longo do tempo.

- **Falta de Aplicação Base de Comparação:** a ausência de uma aplicação de comparação pode afetar a análise dos resultados. No entanto, essa ameaça foi mitigada considerando a experiência prévia dos participantes em desenvolvimento de software, que lhes proporcionou um nível básico de referência.
- **Experiência dos Usuários:** a diversidade no nível de experiência em desenvolvimento e com tecnologias *serverless* pode variar entre os participantes. Essa ameaça foi mitigada através de um tutorial de treinamento que buscou nivelar o conhecimento prévio.
- **Exigências da Plataforma AWS:** o uso da AWS exigiu que os participantes criassem contas e fornecessem informações de cartão de crédito. Apesar de a AWS oferecer um nível gratuito, isso gerou preocupações entre os participantes sobre possíveis cobranças. Essas preocupações foram abordadas no tutorial, que esclareceu o uso do nível gratuito e garantiu que não haveria cobranças inesperadas.

#### 4.10 Considerações Finais sobre o Capítulo

Este capítulo descreveu o protocolo do experimento controlado, incluindo o *design*, os procedimentos, a coleta e análise de dados, bem como as considerações éticas e as ameaças à validade. O experimento foi projetado para avaliar a estratégia *Serverless First* na implementação de uma aplicação de gestão de tarefas, com foco na experiência do desenvolvedor. O próximo capítulo apresentará os resultados do experimento.

## 5 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da implementação de uma aplicação de gestão de tarefas utilizando a estratégia *Serverless First*. As respostas dos participantes ao questionário foram analisadas para avaliar a dificuldade de implementação, os benefícios percebidos e a experiência do desenvolvedor com a referida estratégia.

### 5.1 Percepção da Dificuldade da Implementação

A maioria dos participantes avaliou positivamente a implementação da estratégia *Serverless First*, conforme a classificação disponível na Figura 2. Metade dos participantes considerou a dificuldade de implementação como razoável, 1/3 como fácil, enquanto uma minoria de 1/6 a considerou difícil.

Essa variação pode estar relacionada à familiaridade prévia com a tecnologia *serverless* e ao ambiente de desenvolvimento utilizado. Em relação a uma autodefinição dos participantes, 1/3 se considera desenvolvedor, enquanto o restante se auto-classifica como estudante de TI. Dos participantes, 77,8% nunca haviam usado *serverless* anteriormente.

### 5.2 Benefícios Percebidos da Estratégia Serverless First

Os participantes reconheceram benefícios na adoção da estratégia *Serverless First*, conforme resultado apresentado na Figura 2. Entre os benefícios percebidos, destacaram-se a elasticidade e a escalabilidade automáticas e a simplificação do desenvolvimento. Essas vantagens permitem que os desenvolvedores se concentrem na lógica do negócio e nas funcionalidades da aplicação, em vez de se preocuparem com a gestão da infraestrutura.

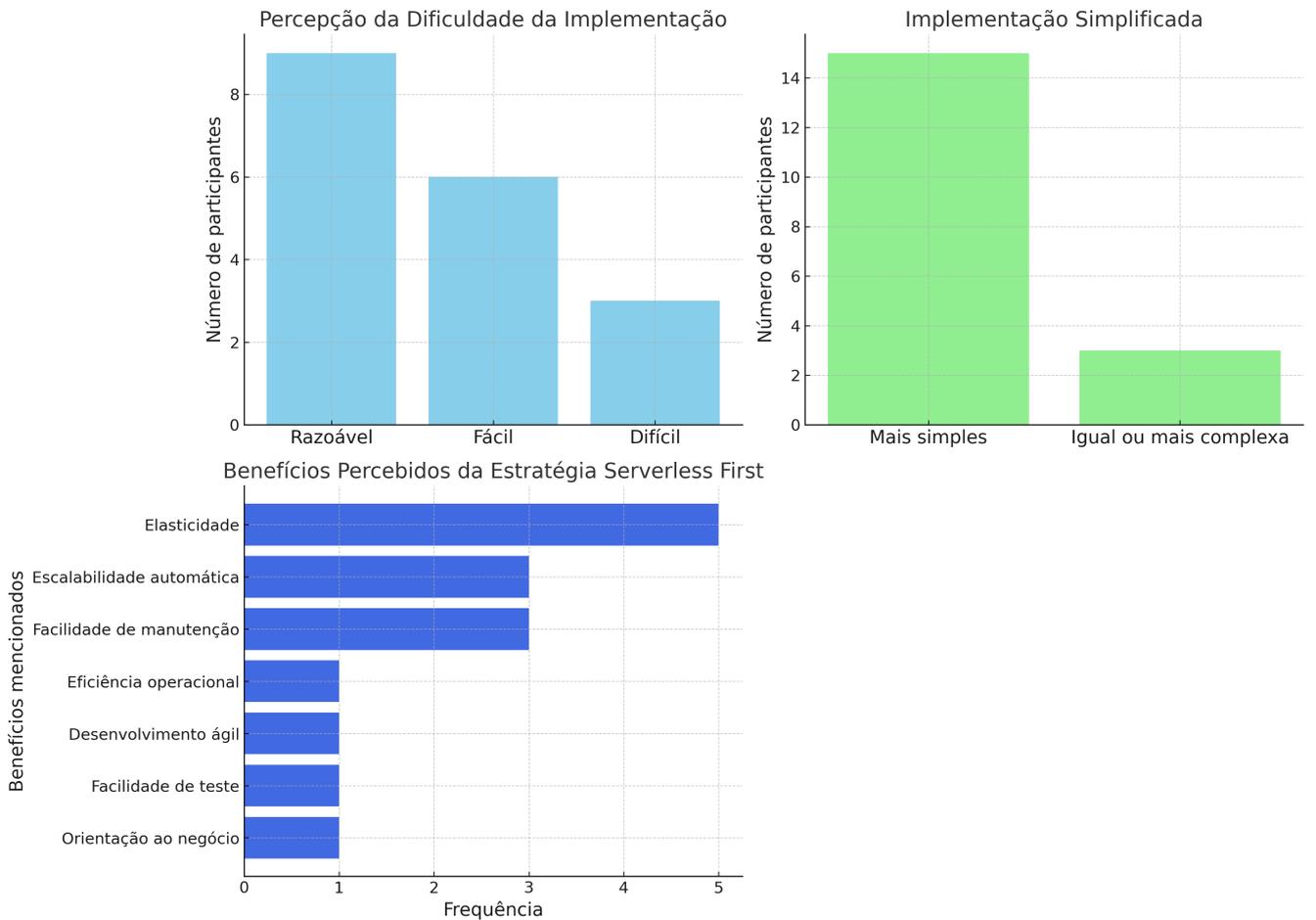
### 5.3 Diminuição do Tempo de Produção

Todos os participantes concordaram que a estratégia *Serverless First* resulta em diminuição do tempo necessário para a produção de *software*. Dentre os fatores descritivos, foi apontada a eliminação da necessidade de gerenciar a infraestrutura e a possibilidade de focar exclusivamente no código, o que contribui para uma maior agilidade no desenvolvimento da aplicação.

### 5.4 Implementação Simplificada

A implementação da estratégia *Serverless First* foi considerada mais simples por 83,3% dos participantes quando comparada ao modelo tradicional de desenvolvimento de software, que requer gerenciamento direto de servidores e infraestrutura, conforme os resultados apresentados na Figura 2. De acordo com os participantes, a gestão da infraestrutura pelo provedor de nuvem no modelo *serverless* reduz a complexidade operacional. No entanto, aqueles que apresentaram ressalvas indicaram que a simplicidade percebida

Figura 2 – Resultados da Percepção dos Participantes



Fonte: Elaborado pelo autor.

no uso de *serverless* pode variar dependendo da complexidade e dos requisitos específicos da aplicação em desenvolvimento.

## 5.5 Recomendação Serverless First

Todos os participantes recomendariam a estratégia *Serverless First* para outros desenvolvedores. Os principais motivos incluíram a simplificação do desenvolvimento, a redução de custos com infraestrutura e a escalabilidade automática. A estratégia foi considerada eficiente e ágil para a criação e manutenção de aplicações.

## 5.6 Sugestões para Melhorias

Os participantes sugeriram melhorias na documentação e nos tutoriais disponíveis, especialmente na configuração inicial e na definição de infraestrutura. Comentários ex-

plicativos e tutoriais mais detalhados podem facilitar a adoção da estratégia *Serverless First* por desenvolvedores com diferentes níveis de experiência.

## 5.7 Considerações Finais sobre o Capítulo

Neste capítulo foram apresentadas as percepções dos participantes sobre a implementação da estratégia *Serverless First* em uma aplicação de gestão de tarefas. Os resultados destacaram que a maioria dos desenvolvedores considerou a implementação dessa abordagem razoavelmente fácil, atribuindo benefícios como redução de custos operacionais, escalabilidade automática e simplificação do desenvolvimento. Além disso, houve consenso sobre a diminuição do tempo necessário para a produção de software e a simplificação na implementação em comparação com modelos tradicionais que requerem gerenciamento de servidores. Todos os participantes recomendaram a estratégia *Serverless First* para outros desenvolvedores, embasados na eficiência percebida e nos ganhos em agilidade. As sugestões de melhorias focaram na documentação e nos tutoriais disponíveis, visando facilitar a adoção da estratégia por uma audiência diversificada de desenvolvedores.



## 6 CONCLUSÃO

Este estudo investigou as vantagens da estratégia *Serverless First*, com foco na experiência do desenvolvedor, avaliando a facilidade de implementação e a eficiência do desenvolvimento. Os resultados da pesquisa foram obtidos a partir do experimento controlado e da análise qualitativa e quantitativa das percepções dos participantes.

### 6.1 Contribuição

Os resultados demonstraram que a estratégia *Serverless First* proporciona uma experiência de desenvolvimento eficiente e simplificada. A maioria dos participantes destacou a redução de custos operacionais e a escalabilidade automática como os principais benefícios dessa abordagem. Adicionalmente, a estratégia *Serverless First* permite que os desenvolvedores se concentrem na lógica do negócio, ao invés de se preocuparem com a infraestrutura subjacente, o que foi evidenciado pela diminuição do tempo necessário para a produção de *softwares*, conforme relatado pelos participantes.

No entanto, também foram relatados desafios relacionados à configuração inicial e à definição de infraestrutura, especialmente no que diz respeito à configuração de credenciais. Esses desafios indicam a necessidade de melhorias na documentação e nos tutoriais disponíveis, a fim de facilitar a adoção da estratégia *Serverless First* por desenvolvedores com diferentes níveis de experiência.

A análise dos estudos relacionados e os resultados do experimento controlado indicam que a estratégia *Serverless First* é uma abordagem promissora para o desenvolvimento de aplicações em nuvem. A adoção desta estratégia pode impulsionar a competitividade e a inovação no mercado, promovendo uma prática de desenvolvimento mais ágil e eficiente.

### 6.2 Sugestões para Trabalhos Futuros

Para pesquisas futuras, recomenda-se:

- Ampliar o tamanho da amostra, incluindo desenvolvedores com diferentes níveis de experiência para obter uma visão mais abrangente das vantagens e desafios associados a estratégia.
- Incluir métricas quantitativas, como tempo de desenvolvimento, desempenho das aplicações e custo operacional, poderia fornecer uma avaliação mais detalhada.
- Investigar estratégias para mitigar os desafios relacionados à configuração inicial e à dependência de provedores, especialmente no que se refere à configuração de credenciais e gerenciamento de recursos.
- Avaliar a adoção da estratégia *Serverless First* em diferentes contextos de aplicação, incluindo setores variados como saúde, educação e finanças, etc.

- Desenvolver e testar novas ferramentas e *frameworks* que possam facilitar ainda mais a implementação de aplicações *serverless*, incluindo melhorias em documentação e tutoriais acessíveis.
- Realizar estudos comparativos entre a estratégia *Serverless First* e outras abordagens de desenvolvimento, a fim de identificar as melhores práticas e os cenários ideais para cada uma.

### 6.3 Publicação

Além disso, como resultado deste trabalho, foi publicado um artigo intitulado *Exploring the Serverless First Strategy in Cloud Application Development*, apresentado no **2023 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW)**. Este trabalho avança a arquitetura *Serverless*, apresentando a estratégia *Serverless First* no desenvolvimento de *software*. Os resultados obtidos incentivam a exploração e adoção dessa estratégia, contribuindo para a evolução das práticas de desenvolvimento na era da computação em nuvem.

## REFERÊNCIAS

- ADZIC, G.; CHATLEY, R. Serverless computing: economic and architectural impact. In: **Proceedings of the 2017 11th joint meeting on foundations of software engineering**. New York, NY, United States: ACM, 2017. p. 884–889. Citado 3 vezes nas páginas 26, 28 e 30.
- BARDSLEY, D.; RYAN, L.; HOWARD, J. Serverless Performance and Optimization Strategies. In: IEEE. **2018 IEEE International Conference on Smart Cloud (SmartCloud)**. Piscataway, NJ, USA, 2018. p. 19–26. Citado 3 vezes nas páginas 26, 29 e 30.
- CLOUDFLARE. **O que é Serverless?** 2024. Disponível em: <<https://www.cloudflare.com/pt-br/learning/serverless/what-is-serverless/>>. Citado na página 23.
- CRONAPP. **Desenvolvimento em Nuvem**. 2019. Disponível em: <<https://blog.cronapp.io/desenvolvimento-em-nuvem/>>. Citado na página 23.
- EISMANN, S. et al. Serverless Applications: Why, When, and How? **IEEE Software**, v. 38, n. 1, p. 32–39, 2021. Citado na página 25.
- Elastic. **What is Serverless Computing?** 2024. Disponível em: <<https://www.elastic.co/pt/what-is/serverless-computing>>. Citado na página 26.
- Gartner. **Magic Quadrant for Cloud**. 2023. Disponível em: <[https://www.gartner.com/doc/reprints?id=1-2FTDYPQN&ct=231204&st=sb&trk=article-ssr-frontend-pulse\\_little-text-block](https://www.gartner.com/doc/reprints?id=1-2FTDYPQN&ct=231204&st=sb&trk=article-ssr-frontend-pulse_little-text-block)>. Citado na página 24.
- GILBERT, J. **Software Architecture Patterns for Serverless Systems: Architecting for innovation with event-driven microservices and micro frontends**. second. Birmingham: Packt Publishing, 2024. 488 p. Citado na página 23.
- IVAN, C.; VASILE, R.; DADARLAT, V. Serverless Computing: An Investigation of Deployment Environments for Web APIs. **Computers**, MDPI, v. 8, n. 2, p. 50, 2019. Citado 3 vezes nas páginas 26, 29 e 30.
- KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Keele University, 2007. Citado na página 44.
- KUMANOV, D. et al. Serverless Computing Provides On-Demand High Performance Computing for Biomedical Research. **arXiv preprint arXiv:1807.11659**, 2018. Citado na página 29.
- LEE, H.; SATYAM, K.; FOX, G. Evaluation of Production Serverless Computing Environments. In: IEEE. **2018 IEEE 11th International Conference on Cloud Computing (CLOUD)**. Piscataway, New Jersey, 2018. p. 442–450. Citado 2 vezes nas páginas 28 e 30.
- MAMPAGE, A.; KARUNASEKERA, S.; BUYYA, R. A holistic view on resource management in serverless computing environments: Taxonomy and future directions. **ACM Computing Surveys (CSUR)**, ACM New York, NY, v. 54, n. 11s, p. 1–36, 2022. Citado na página 25.

- NDAY, B. A.; KUSUMA, G. P.; FREDYAN, R. Serverless utilization in microservice e-learning platform. **Procedia Computer Science**, Elsevier, v. 216, p. 204–212, 2023. Citado 3 vezes nas páginas 26, 27 e 30.
- OKOLI, C.; SCHABRAM, K. A guide to conducting a standalone systematic literature review. **Communications of the Association for Information Systems**, v. 37, n. 43, p. 879–910, 2015. Citado na página 43.
- RUPARELIA, N. B. **Cloud Computing**. Cambridge, Massachusetts: The MIT Press, 2016. 280 p. ISBN 9780262529099. Citado na página 23.
- SADEK, J.; CRAIG, D.; TRENELL, M. Design and implementation of medical searching system based on microservices and serverless architectures. **Procedia Computer Science**, Elsevier, v. 196, p. 615–622, 2022. Citado 2 vezes nas páginas 27 e 30.
- SCHLEIER-SMITH, J. et al. A FaaS File System for Serverless Computing. **arXiv preprint arXiv:2009.09845**, 2020. Citado 2 vezes nas páginas 28 e 30.
- SCHLEIER-SMITH, J. et al. What serverless computing is and should become: The next phase of cloud computing. **Communications of the ACM**, ACM New York, NY, USA, v. 64, n. 5, p. 76–84, 2021. Citado 2 vezes nas páginas 25 e 26.
- SERVICES, A. W. **Página de Casos de Estudo da AWS**. 2023. Disponível em: <<https://aws.amazon.com/pt/solutions/case-studies/>>. Citado na página 47.
- VAUGHAN, D. **Embracing the 'Serverless First' Approach through Progressive Decomposition**. 2023. Disponível em: <<https://www.linkedin.com/pulse/embracing-serverless-first-approach-through-daniel-vaughan>>. Citado 2 vezes nas páginas 23 e 26.
- WEN, J. et al. An empirical study on challenges of application development in serverless computing. In: **Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering**. Danvers, MA, USA: ACM, 2021. p. 416–428. Citado 3 vezes nas páginas 26, 28 e 30.
- YAN, M. et al. Building a chatbot with serverless computing. In: **Proceedings of the 1st International Workshop on Mashups of Things and APIs**. Danvers, MA, USA: ACM, 2016. p. 1–4. Citado 2 vezes nas páginas 28 e 30.
- ZHANG, M. et al. Video Processing With Serverless Computing: A Measurement Study. In: **Proceedings of the 29th ACM workshop on network and operating systems support for digital audio and video**. Danvers, MA, USA: ACM, 2019. p. 61–66. Citado 2 vezes nas páginas 29 e 30.

## ANEXO A – REVISÃO DA LITERATURA

A Revisão Sistemática da Literatura (RSL) é uma abordagem metodológica rigorosa e transparente, usada para identificar, selecionar e sintetizar estudos relevantes sobre um tema de pesquisa. Seu objetivo é minimizar a influência de vieses e subjetividade na análise dos resultados (OKOLI; SCHABRAM, 2015). Para uma análise mais abrangente, esta pesquisa utilizou uma metodologia combinada, integrando a RSL com uma Revisão Integrativa, que permitiu a inclusão de estudos adicionais ao longo do processo.

### A.1 Definição do Protocolo

Esta seção apresenta o protocolo adotado para a revisão combinada sobre a abordagem *Serverless First* no desenvolvimento de aplicativos. Serão descritos a estratégia de busca, os critérios de inclusão e exclusão dos estudos, e os critérios de avaliação da qualidade dos estudos.

#### A.1.1 Revisão Sistemática da Literatura

A RSL foi realizada para identificar e sintetizar estudos relevantes nas principais bases de dados acadêmicas: Scopus, IEEE Xplore, ACM Digital Library, ScienceDirect e Springer. A estratégia de busca foi desenvolvida utilizando termos específicos relacionados ao tema da revisão, conforme descrito na Tabela 1. Esta abordagem resultou na identificação de 65 estudos potencialmente relevantes, dos quais 6 foram selecionados para a análise detalhada após a aplicação dos critérios de inclusão e exclusão.

#### A.1.2 Revisão Integrativa

Para complementar os resultados da RSL, foi realizada uma Revisão Integrativa. Essa abordagem permitiu a inclusão de estudos adicionais que surgiram ao longo do processo de pesquisa, ampliando o escopo e a profundidade da análise. A Revisão Integrativa seguiu uma estratégia de busca mais flexível, permitindo a incorporação de 5 estudos adicionais que não foram capturados inicialmente pela RSL, mas que são relevantes para o tema em questão.

#### A.1.3 Questões de Pesquisa

Três questões de pesquisa foram formuladas para orientar a revisão:

- **Q1:** Quais são os benefícios da estratégia *Serverless First* no desenvolvimento de aplicativos em nuvem?
- **Q2:** Quais são as ferramentas mais eficientes e apropriadas para o desenvolvimento de aplicativos *Serverless First*?

- **Q3:** Quais são os desafios e limitações associados à adoção da estratégia *Serverless First* no desenvolvimento de aplicativos?

#### A.1.4 Estratégia de Busca

A estratégia de busca foi desenvolvida para identificar estudos relevantes em bases de dados acadêmicas. A *string* de busca combina termos relacionados aos temas da revisão em inglês Tabela 1, utilizando operadores booleanos (OR e AND).

Tabela 1 – Termos

Termo	Sinônimos
Serverless First	Serverless approach, serverless architecture
Benefits	Advantages, positive impact, improvements
Challenges and Limitations	Obstacles, problems, restrictions, disadvantages
Efficient Tools	Effective resources, productive solutions, suitable tools
Application Development	App development, software programming

Fonte: Elaboração própria

Para a análise e seleção dos estudos primários foram definidas as seguintes bases de dados Tabela 2:

Tabela 2 – Bases de dados

Base	Link
ACM	< <a href="http://portal.acm.org/">http://portal.acm.org/</a> >
IEEE	< <a href="http://ieeexplore.ieee.org/">http://ieeexplore.ieee.org/</a> >
ScienceDirect	< <a href="http://www.sciencedirect.com/">http://www.sciencedirect.com/</a> >
Scopus	< <a href="https://www.scopus.com/">https://www.scopus.com/</a> >
Springer	< <a href="http://link.springer.com/">http://link.springer.com/</a> >

Fonte: Elaboração própria

#### A.1.5 Critérios de Inclusão e Exclusão

Os critérios de inclusão e exclusão são fundamentais para garantir que os estudos selecionados sejam relevantes e atendam aos objetivos da revisão (KITCHENHAM; CHARTERS, 2007).

##### Critérios de inclusão:

- **CI1:** Idioma inglês;
- **CI2:** Texto acessível na íntegra;
- **CI3:** Publicações entre 2018 e 2023;
- **CI4:** Estudos que contenham no título ou no resumo pelo menos um dos termos relacionados ao tema;

Tabela 3 – Estudos recuperados

Fonte	Quantidade de estudos
ACM	26
IEEEExplore	1
Science Direct	11
Scopus	16
Springer Link	11
<b>Total</b>	<b>65</b>

Fonte: Elaboração própria

- **CI5:** Estudos que apresentem algum tipo de avaliação empírica, como experimentos ou estudos de caso.

**Critérios de exclusão:**

- **CE1:** Estudos que não apresentem avaliação empírica relevante;

### A.1.6 Questões de Qualidade

As questões de qualidade foram elaboradas para avaliar a contribuição dos estudos selecionados, utilizando três legendas: S (Sim), P (Parcialmente) e N (Não).

- **QQ1:** O estudo está descrito de maneira clara e detalhada, permitindo replicação dos resultados?
- **QQ2:** O estudo está atualizado em relação aos avanços recentes na área?
- **QQ3:** O estudo apresenta alguma contribuição para responder às questões da pesquisa?

## A.2 Análise e Síntese dos Dados

A fase de identificação dos estudos resultou em 65 artigos potencialmente relevantes, conforme mostrado na Tabela 3.

### A.2.1 Fase de Seleção dos Estudos

A fase de seleção, utilizando os critérios estabelecidos, resultou na aceitação de 23 artigos para leitura completa, enquanto 38 foram rejeitados, e 4 foram removidos por duplicidade, conforme Tabela 4.

### A.2.2 Fase de Extração dos Dados Relevantes

Durante a fase de extração, foram analisados 23 artigos, dos quais 6 foram considerados relevantes e aceitos na RSL. Durante a Revisão Integrativa, mais 5 artigos

Tabela 4 – Fase de Seleção

<b>Estatística</b>	<b>Quantidade de estudos</b>
Artigos Aceitos	23
Artigos Rejeitados	38
Artigos Duplicados	4
Artigos Não Classificados	0

Fonte: Elaboração própria

relevantes foram adicionados, totalizando 11 estudos aceitos para análise detalhada, conforme Tabela 5.

Tabela 5 – Fase de Extração

<b>Estatística</b>	<b>Quantidade de estudos</b>
Artigos Aceitos	11
Artigos Rejeitados	17
Artigos Duplicados	0
Artigos Não Classificados	0

Fonte: Elaboração própria

### A.3 Considerações Finais sobre o Capítulo

Este capítulo apresentou o protocolo adotado para a realização da revisão combinada sobre a abordagem *Serverless First* no desenvolvimento de aplicativos. A estratégia de busca, os critérios de inclusão e exclusão, e a avaliação dos estudos selecionados foram descritos. Esta revisão visa fornecer uma visão abrangente sobre a computação sem servidor, explorando seus benefícios, desafios e ferramentas utilizadas.

## ANEXO B – RELATÓRIO: IMPACTOS DA ABORDAGEM SERVERLESS EM EMPRESAS DA AMÉRICA LATINA

A fim de entender o impacto da arquitetura *Serverless*, foram buscados exemplos de migração de arquitetura. Como fonte de pesquisa buscou-se a página de casos de uso da AWS (SERVICES, 2023), tendo como data de consulta o dia 12 de outubro de 2023. Como filtro para os casos selecionados foram considerados:

- Localização: América Latina;
- Setor: Diversos;
- Caso de Uso: Diversos.

Desta forma, este relatório analisa os impactos da abordagem *Serverless* em empresas da América Latina, destacando os benefícios e resultados observados em algumas organizações de diferentes setores. Os indicadores são custos, desempenho, disponibilidade, facilidade de implementação, manutenção, redução de pessoal e os serviços *Serverless* mais utilizados (SERVICES, 2023).

### B.1 Impactos nos Custos

A adoção de tecnologia *Serverless* resultou em uma redução nos custos operacionais da empresas. A Tabela 6 apresenta um resumo dos casos de uso, onde é possível conferir os valores e o percentual de redução. Considerando apenas o percentual, a média de redução de custos foi de 83,3%.

### B.2 Impacto no Desempenho e Disponibilidade

As empresas que adotaram tecnologia *Serverless* também relataram um desempenho notável em termos de disponibilidade de seus serviços, proporcionando uma experiência mais confiável para seus usuários. Os resultados obtidos na Tabela 6 mostram que a média de disponibilidade das empresas consultadas foi de 99,9%.

### B.3 Facilidade de Implementação

A facilidade de implementação é um fator importante para considerar. As empresas estudadas na Tabela 6 relataram que a média do tempo necessário para implementar soluções *Serverless* foi de 9 meses, com variações de 3 a 12 meses.

### B.4 Manutenção e Redução de Pessoal

O tempo dedicado à manutenção das aplicações *Serverless* é consideravelmente baixo, economizando recursos de pessoal. A média de tempo de manutenção foi de 3

Tabela 6 – Impactos nos Custos

Empresa	Sector	Caso de Uso	Custo Anterior (R\$ 1.000)	Custo Atual (R\$ 1.000)	Redução de (%)	Disponibilidade (%)
Quasar	Serviços Públicos	Gestão pública	100	10	90%	100
Britvic	Alimentos e Bebidas	Dispensador de água	50	5	90%	100
TWAICE	Energia e Tecnologia	Simulação de baterias	1.000	100	90%	99,9
Grupo Boticário	Varejo	Aplicativo e Site	1.000	200	80%	99,9
H. Albert Einstein	Saúde	Aplicativo de agendamento	500	100	80%	99,9
Embraer	Aero-espaçial	Simulação de voo	10.000	2.000	80%	99,9
iFood	<i>Delivery</i>	Plataforma de entrega	100.000	20.000	80%	99,9
Mercado Livre	<i>E-commerce</i>	Plataforma de <i>E-commerce</i>	1.000.000	200.000	80%	99,9
Sky	TV por assinatura	Plataforma de <i>Streaming</i>	500.000	100.000	80%	99,9

Fonte: Elaborado pelo autor.

horas por semana. Além disso, houve uma redução de pessoal alocado para gerenciar a infraestrutura em média de 30%.

### B.5 Serviços Serverless Mais Utilizados

As empresas utilizaram uma variedade de serviços *Serverless* para atender às suas necessidades específicas. Isso inclui AWS Lambda, Amazon API Gateway, AWS DynamoDB, Amazon CloudFront, Amazon S3, AWS Rekognition e Amazon SageMaker.

### B.6 Tipos de Aplicações Mais Adequadas

A tecnologia *Serverless* é adequada para diversos tipos de aplicações, incluindo:

- Web: Sites, aplicativos web, APIs;
- Mobile: Aplicativos mobile;
- IoT: Dispositivos IoT;
- Data: Análise de Dados, *Machine Learning*;

- Processamento de Eventos: *Streaming* de dados, processamento de eventos em tempo real.

## B.7 Conclusão

Os casos de uso das empresas consideradas apontam benefícios substanciais da tecnologia *Serverless*. A redução de custos, desempenho, facilidade de implementação, economia de tempo de manutenção e redução de pessoal são destaques. Empresas que buscam eficiência operacional e redução de custos podem considerar a adoção da tecnologia *Serverless* como uma alternativa para suas operações.



## ÍNDICE

API, 23, 28–32

AWS, 23, 24, 31, 32, 47

CPU, 29

FaaS, 27

GCP, 31

I/O, 28

IaaS, 27

JSON, 31

RSL, 43, 45

SAM, 31

VM, 28, 29

Web, 28