



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Gabryella Mika Quitakava Tanigawa

Estudo Comparativo De Plataformas De Computação Em Nuvem

São José do Rio Preto
2025

Gabryella Mika Quitakava Tanigawa

Estudo Comparativo De Plataformas De Computação Em Nuvem

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Orientador:

Prof^a. Dr^a. Renata Spolon Lobato

**São José do Rio Preto
2025**

Gabryella Mika Quitakava Tanigawa

Estudo Comparativo De Plataformas De Computação Em Nuvem

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Comissão Examinadora:

Prof^a. Dr^a. Renata Spolon Lobato

Prof. Dr. Aleardo Manacero Junior

Prof. Dr. Eng. Rodrigo Capobianco
Guido

São José do Rio Preto
2025

T164e

Tanigawa, Gabryella Mika Quitakava

Estudo Comparativo De Plataformas De Computação Em Nuvem /
Gabryella Mika Quitakava Tanigawa. -- São José do Rio Preto, 2026
77 f. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da
Computação) - Universidade Estadual Paulista (UNESP), Instituto de
Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientadora: Renata Spolon Lobato

1. Ciência da Computação. 2. Computação sem Servidor. 3.
Computação em Nuvem. I. Título.

GABRYELLA MIKA QUITAKAVA TANIGAWA

ESTUDO COMPARATIVO DE PLATAFORMAS DE COMPUTAÇÃO EM NUVEM

Trabalho de Conclusão de Curso (TCC) apresentado à Universidade Estadual Paulista (UNESP), Instituto de Biociências, Letras e Ciências Exatas, Câmpus de São José do Rio Preto, para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração: Computação em Nuvem

Data de defesa: 27/01/2026

BANCA EXAMINADORA

Prof^ª. Dr^ª. Renata Spolon Lobato
UNESP – IBILCE – São José do Rio Preto

Prof. Dr. Aleardo Manacero Junior

Prof. Dr. Eng. Rodrigo Capobianco Guido

São José do Rio Preto
2025

Agradecimentos

Agradeço a minha família e amigos, à minha orientadora Renata e a todo o corpo docente e funcionários da Unesp.

Resumo

A computação em nuvem tornou-se de grande importância com o avanço tecnológico e a globalização da internet, uma vez que os dados passaram a ter escala global, o que tornou inviável o seu tratamento em servidores locais. Diante disso, foram criados os serviços de computação em nuvem, baseados em sistemas distribuídos, os quais se mostraram tão eficientes e seguros que, atualmente, poucas são as aplicações que não utilizam tais serviços. Essa evolução resultou não apenas na criação de diversas plataformas de computação em nuvem, mas também no aumento de sua relevância acadêmica e comercial. Este trabalho, portanto, concentra-se na comparação entre dois provedores — AWS e Azure — que irão executar aplicações sem servidor, utilizando exclusivamente recursos disponibilizados gratuitamente para estudantes. Ao serem iniciadas, as funções da aplicação retornarão o tempo necessário para executar as aplicações.

Palavras-chave: Computação sem Servidor, Computação em Nuvem.

Abstract

Cloud computing has become highly significant with technological advancements and the globalization of the internet, as data has reached a global scale, making its processing on local servers unfeasible. In response, cloud computing services based on distributed systems were created, proving to be so efficient and secure that nowadays few applications do not rely on such services. This evolution has resulted not only in the emergence of various cloud computing platforms but also in an increase in their academic and commercial relevance. This work, therefore, focuses on comparing two providers — AWS and Azure — which will run serverless applications, using only resources made available for free to students. When triggered, the application's functions will return the time required by the algorithm.

Keywords: Serverless Computing, Cloud Computing.

Lista de Figuras

2.1	Cifra de César	7
3.1	Desempenho da aplicação de assinaturas digitais às 7h00 na <i>Azure</i>	16
3.2	Desempenho da aplicação de assinaturas digitais às 15h00 na <i>Azure</i>	17
3.3	Desempenho da aplicação de assinaturas digitais às 21h00 na <i>Azure</i>	18
3.4	Desempenho da aplicação de funções hash às 7h00 na <i>Azure</i>	19
3.5	Desempenho da aplicação de funções hash às 15h00 na <i>Azure</i>	20
3.6	Desempenho da aplicação de funções hash às 21h00 na <i>Azure</i>	21
3.7	Desempenho da aplicação de compressão GZIP às 7h00 na <i>Azure</i>	22
3.8	Desempenho da aplicação de compressão GZIP às 15h00 na <i>Azure</i>	23
3.9	Desempenho da aplicação de compressão GZIP às 21h00 na <i>Azure</i>	24
3.10	Desempenho da aplicação de assinaturas digitais às 7h00 na <i>AWS</i>	26
3.11	Desempenho da aplicação de assinaturas digitais às 15h00 na <i>AWS</i>	27
3.12	Desempenho da aplicação de assinaturas digitais às 21h00 na <i>AWS</i>	28
3.13	Desempenho da aplicação de funções hash às 7h00 na <i>AWS</i>	29
3.14	Desempenho da aplicação de funções hash às 15h00 na <i>AWS</i>	30
3.15	Desempenho da aplicação de funções hash às 21h00 na <i>AWS</i>	31
3.16	Desempenho da aplicação de compressão GZIP às 7h00 na <i>AWS</i>	32
3.17	Desempenho da aplicação de compressão GZIP às 15h00 na <i>AWS</i>	33
3.18	Desempenho da aplicação de compressão GZIP às 21h00 na <i>AWS</i>	34
3.19	Tempos de execução das verificações de assinatura da às 7:00	35
3.20	Tempos de execução das verificações de assinatura às 15:00	36
3.21	Tempos de execução das verificações de assinatura às 21:00	37

3.22	Execução da verificação de assinaturas em diferentes horários na <i>Azure</i>	38
3.23	Execução da verificação de assinaturas em diferentes horários na <i>AWS</i>	39
3.24	Execução das funções hash às 7h00	40
3.25	Execução das funções hash às 15:00	41
3.26	Execução das funções hash às 21:00	42
3.27	Execução de funções hash em diferentes horários na <i>Azure</i>	43
3.28	Execução de funções hash em diferentes horários na <i>AWS</i>	44
3.29	Tempos de execução das compressões GZIP às 7:00	44
3.30	Tempos de execução das compressões GZIP às 15:00	45
3.31	Tempos de execução das compressões GZIP às 21:00	46
3.32	Execução de funções hash em diferentes horários na <i>Azure</i>	47
3.33	Execução de funções hash em diferentes horários na <i>AWS</i>	48

Lista de Tabelas

3.1	Comparação das médias de velocidade entre <i>Azure</i> e <i>AWS</i>	48
3.2	Comparação dos desvios padrão de velocidade entre <i>Azure</i> e <i>AWS</i>	49
3.3	Desempenho da <i>AWS Lambda</i> por horário	49
3.4	Desempenho da <i>Azure Functions</i> por horário	50
4.1	Comparação geral entre <i>Azure</i> e <i>AWS</i>	52
4.2	Comparação do uso das plataformas <i>AWS</i> e <i>Azure</i>	53
A.1	Tempo de execução da aplicação de verificação de assinaturas digitais às 7h00	56
A.2	Tempo de execução da aplicação de verificação de assinaturas digitais às 15h00	57
A.3	Tempo de execução da aplicação de verificação de assinaturas digitais às 21h00	57
A.4	Tempo de execução da aplicação de funções hash às 7h00	57
A.5	Tempo de execução da aplicação de funções hash às 15h00	57
A.6	Tempo de execução da aplicação de funções hash às 21h00	58
A.7	Tempo de execução da aplicação de compressão GZIP às 7h00	58
A.8	Tempo de execução da aplicação de compressão GZIP às 15h00	58
A.9	Tempo de execução da aplicação de compressão GZIP às 21h00	58
B.1	Tempo de execução da aplicação de verificação de assinaturas digitais às 7h00	59
B.2	Tempo de execução da aplicação de verificação de assinaturas digitais às 15h00	60
B.3	Tempo de execução da aplicação de verificação de assinaturas digitais às 21h00	60
B.4	Tempo de execução da aplicação de funções hash às 7h00	60
B.5	Tempo de execução da aplicação de funções hash às 15h00	60
B.6	Tempo de execução da aplicação de funções hash às 21h00	61

B.7	Tempo de execução da aplicação de compressão GZIP às 7h00	61
B.8	Tempo de execução da aplicação de compressão GZIP às 15h00	61
B.9	Tempo de execução da aplicação de compressão GZIP às 21h00	61

Sumário

1	Introdução	1
1.1	Objetivos	1
1.2	Justificativa	2
2	Revisao Bibliográfica	3
2.1	Computação em Nuvem	3
2.1.1	Modelos de Serviços de Computação em Nuvem	4
2.2	Computação sem Servidor	5
2.3	Criptografia	6
2.3.1	Cifra de César	6
2.3.2	Criptografia RSA	7
2.3.3	Criptografia Hash	8
2.4	GZIP	8
2.5	Trabalhos Relacionados	9
2.5.1	<i>Desvendando as Nuvens: Uma Análise Comparativa de Desempenho de Aplicações Serverless e Containers em Provedores de Computação em Nuvem</i>	9
2.5.2	<i>Cloud Computing and Comparison based on Service and Performance between Amazon AWS, Microsoft Azure, and Google Cloud</i>	9
2.5.3	<i>Análise Comparativa de Escalabilidade e Disponibilidade de Bancos de Dados em Ambientes de Nuvem: um estudo dos serviços oferecidos por AWS e Azure</i>	10

2.5.4	<i>Comparative Analysis of Virtual Machine Performance and Security Test Across Leading Cloud Providers</i>	10
2.5.5	<i>A Comparative Analysis of Cloud Computing Services: AWS, Azure, and GCP</i>	10
3	Desenvolvimento	12
3.1	Aplicações	12
3.1.1	Bibliotecas do <i>Python</i>	12
3.2	Testes de Desempenho	13
3.2.1	<i>Azure Functions</i>	13
3.2.2	<i>AWS Lambda</i>	25
3.3	Resultado dos Testes Comparativos	35
3.3.1	Assinaturas Digitais	35
3.3.2	Funções Hash	39
3.3.3	Compressão GZIP	43
3.3.4	Visão geral	47
4	Conclusão	51
A	APÊNDICE A – Resultados detalhados dos experimentos na plataforma Azure Functions	56
B	APÊNDICE B – Resultados detalhados dos experimentos na plataforma AWS Lambda	59
	Referências Bibliográficas	62

Capítulo 1

Introdução

A computação está em constante evolução por ser uma área que resolve ou facilita tarefas muito complexas ou de grande demanda. A internet, por exemplo, tornou mais simples a troca de informações entre dois pontos distantes e, posteriormente, entre pessoas e sistemas em todo o mundo. Consequentemente, o volume de dados também se expandiu em escala global, o que tornou inviável o uso exclusivo de servidores locais. Assim, surgiu a computação em nuvem, que, baseada em sistemas distribuídos, viabilizou o processamento de grandes quantidades de dados.

Por serem facilmente escaláveis, flexíveis e transparentes, muitas empresas migraram para essas plataformas com o objetivo de atender às suas necessidades sem se preocuparem com grandes volumes de hardware, atualizações, falhas em equipamentos ou compatibilidade entre software e hardware.(FILHO; LOBATO, 2022)

Diante desse cenário, diversas plataformas passaram a oferecer serviços em nuvem. Assim, o presente trabalho realiza uma comparação entre duas das maiores provedoras desse tipo de serviço.

1.1 Objetivos

O objetivo deste trabalho é comparar duas plataformas — *Amazon Web Services (AWS)* e *Microsoft Azure* — quanto aos serviços que disponibilizam gratuitamente para estudantes, utilizando, para isso, aplicações serverless.

Serão considerados, para a comparação entre AWS Lambda e Azure Function App, fatores como velocidade de processamento, comportamento das variações e disponibilidade de serviços.

1.2 Justificativa

Como justificativa para o presente trabalho, destaca-se a possibilidade de apoiar estudantes que têm como objetivo utilizar plataformas de serviços em nuvem, auxiliando-os a identificar a melhor opção para seus projetos sem a necessidade de gastos adicionais.

Vale ressaltar que, ao auxiliar os estudantes na escolha das plataformas, também se promove o aprendizado e o desenvolvimento de habilidades relacionadas ao uso desses serviços.

Capítulo 2

Revisão Bibliográfica

2.1 Computação em Nuvem

A computação em nuvem redefiniu a forma como recursos computacionais são utilizados, substituindo o modelo tradicional baseado na aquisição de infraestrutura física por um modelo orientado à contratação de serviços. Nesse contexto, os usuários passam a acessar recursos distribuídos em *datacenters* de provedores como Microsoft e Amazon, que oferecem capacidades de hardware, software, plataformas de desenvolvimento e diversos serviços de forma escalável e sob demanda (FILHO, 2024). De acordo com (CALIXTO, 2024), esse modelo permite que empresas e desenvolvedores utilizem estruturas robustas sem a necessidade de investir em servidores próprios, reduzindo tanto gastos de capital (CAPEX) quanto despesas operacionais (OPEX).

A computação em nuvem é estruturada a partir de um conjunto de características essenciais que possibilitam o uso eficiente e flexível de recursos computacionais. (CALIXTO, 2024) destaca que esses elementos definem a forma como os serviços são organizados e consumidos no ambiente em nuvem, enquanto (FILHO, 2024) reforça a importância da escalabilidade e do provisionamento ágil nesse modelo.

Entre as características fundamentais, destacam-se:

- **Serviço sob demanda / autoatendimento:** o usuário pode provisionar ou liberar capacidade computacional — como processamento, armazenamento ou execução de apli-

cações — a qualquer momento, sem intervenção direta do provedor (FILHO, 2024; CALIXTO, 2024).

- **Acesso amplo à rede:** os serviços de nuvem podem ser acessados por diversos dispositivos conectados à internet. (CALIXTO, 2024) ressalta que essa disponibilidade garante maior flexibilidade ao usuário, permitindo o gerenciamento de recursos em qualquer lugar e momento.
- **Aglomerado ou agrupamento de recursos:** os provedores utilizam um modelo multilocatário, distribuindo recursos físicos e virtuais em diversos datacenters ao redor do mundo. (FILHO, 2024) destaca que essa distribuição global possibilita escalabilidade, diferentes latências e diversidade de serviços.
- **Elasticidade rápida:** os recursos podem ser expandidos ou reduzidos automaticamente conforme a demanda. Essa capacidade é enfatizada tanto por (CALIXTO, 2024), ao tratar da automação do processo, quanto por (FILHO, 2024), ao abordar o impacto da elasticidade no desempenho das aplicações.
- **Serviço mensurado / medição do consumo:** os sistemas em nuvem monitoram e registram continuamente o uso de recursos, permitindo tomada de decisões baseada em métricas (CALIXTO, 2024). Para (FILHO, 2024), esse monitoramento também contribui para a eficiência da alocação de recursos.

Com base nessas características, a computação em nuvem consolidou-se como uma solução eficiente para organizações e indivíduos que buscam flexibilidade, escalabilidade e economia, permitindo o desenvolvimento e a disponibilização de aplicações sem a necessidade de manter uma infraestrutura própria complexa.

2.1.1 Modelos de Serviços de Computação em Nuvem

Os provedores de serviços em nuvem disponibilizam diferentes modelos de serviço, que variam de acordo com o nível de controle concedido ao cliente sobre suas aplicações e com

o grau de abstração oferecido pela plataforma. Consequentemente, os preços também podem variar conforme o modelo escolhido.

- **SaaS – Software as a Service – Software como Serviço:**

Nesse modelo, o cliente possui o menor nível de controle sobre o serviço oferecido em nuvem, utilizando-o por meio de uma interface gráfica, sem possibilidade de interferir na infraestrutura do provedor. Exemplos comuns incluem serviços de e-mail. Esse modelo implica alta escalabilidade, ampla acessibilidade pela internet independentemente do dispositivo utilizado e ausência de preocupações relacionadas à infraestrutura (RASHID; CHATURVEDI, 2019).

- **PaaS – Platform as a Service – Plataforma como Serviço:**

O nível de abstração desse modelo oferece ao cliente um pouco mais de controle, permitindo o uso de linguagens de programação, bibliotecas, serviços e ferramentas suportadas pela plataforma. No entanto, não há controle sobre o sistema operacional, o armazenamento ou o hardware subjacente. Entre seus benefícios, destaca-se a possibilidade de desenvolver e implementar aplicações próprias sem a necessidade de gerenciar atualizações ou manutenções da infraestrutura (RASHID; CHATURVEDI, 2019).

- **IaaS – Infrastructure as a Service – Infraestrutura como Serviço:**

É o modelo que proporciona o maior nível de controle ao cliente. Nesse caso, o usuário tem acesso a recursos computacionais para gerenciar processamento, armazenamento, rede e sistema operacional. Entre suas vantagens estão a redução de investimentos em bens de capital, a possibilidade de pagamento conforme o uso, o maior controle sobre o ambiente e o acesso a recursos computacionais de alto desempenho (RASHID; CHATURVEDI, 2019).

2.2 Computação sem Servidor

As aplicações convencionais utilizam máquinas virtualizadas hospedadas nos *datacenters* do provedor, que são alocadas integralmente para uso do cliente. Entretanto, em muitos

casos, a aplicação não utiliza todos os recursos da máquina virtual, aproveitando apenas uma parte dela. Isso faz com que o cliente pague pela máquina inteira, mesmo sem utilizá-la completamente.

Com o intuito de solucionar essa limitação, surgiu a computação *serverless*, que permite que a aplicação utilize apenas uma fração dos recursos de uma máquina virtual durante um período muito menor do que nas aplicações convencionais. Dessa forma, em vez de pagar pela infraestrutura completa, o cliente paga apenas pelo tempo de execução e pelos recursos consumidos pela aplicação em si (FILHO, 2024).

2.3 Criptografia

Algoritmos criptográficos são métodos destinados a impedir que informações sigilosas sejam acessadas por pessoas não autorizadas, ou seja, por todos aqueles que não possuem a chave secreta necessária para decifrá-las (TERADA, 2008).

2.3.1 Cifra de César

Esse é um exemplo simples e clássico de criptografia, criado por Júlio César, do Império Romano. Tal algoritmo consiste em escrever uma mensagem substituindo cada letra por seu equivalente em um alfabeto deslocado; a chave é justamente esse deslocamento. A Figura 2.1 demonstra a letra original na primeira linha e seu equivalente na segunda. No caso da Figura 2.1, ao escrever a palavra SATURNO, obtém-se ZHABYUV, sendo a chave igual a 19, pois o alfabeto foi deslocado 19 letras à frente. Portanto, ao enviar essa mensagem, o receptor precisa conhecer o tipo de criptografia utilizada e a chave correspondente para realizar a descryptografia. (TERADA, 2008).

Figura 2.1: Cifra de César

A	B	C	D	E	F	G	H	I	J	K	L	M
H	I	J	K	L	M	N	O	P	Q	R	S	T
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
U	V	W	X	Y	Z	A	B	C	D	E	F	G

Fonte: TERADA, R. *Segurança de dados: Criptografia em rede de computador*. 2008.

2.3.2 Criptografia RSA

Entretanto, a Cifra de César é extremamente simples de ser descriptografada. Como possui apenas 25 chaves possíveis, um indivíduo sem a chave secreta pode testar todas as combinações em um curto período de tempo, especialmente com o auxílio de um computador. Diante disso, os algoritmos de criptografia modernos precisam oferecer um nível de segurança significativamente superior.

Entre esses algoritmos, destaca-se o RSA (Rivest–Shamir–Adleman), amplamente utilizado em assinaturas digitais por sua confiabilidade. Criado por Ron Rivest, Adi Shamir e Leonard Adleman, o RSA é um algoritmo de criptografia assimétrica, ou seja, utiliza duas chaves distintas: uma para criptografar e outra para descriptografar. A chave utilizada para a criptografia pode ser pública, pois, sem a chave correspondente de descriptografia — que é privada —, a decodificação torna-se inviável com as tecnologias computacionais atualmente disponíveis (RAKHRA et al., 2024).

A verificação consiste na aplicação da chave pública sobre a assinatura, possibilitando a validação da autenticidade e da integridade da mensagem. Este trabalho não se aprofundará no detalhamento matemático do algoritmo, uma vez que o foco do estudo está na análise comparativa de desempenho entre serviços de computação em nuvem. Uma explicação mais detalhada pode ser encontrada em (ANDRADE; SILVA, 2012).

2.3.3 Criptografia Hash

A função criptográfica de *hash*, ou simplesmente função de *hash*, é um tipo de mecanismo criptográfico caracterizado por ser unidirecional, isto é, uma vez realizado o cálculo do *hash*, não é possível obter o texto original a partir do valor gerado. Além disso, independentemente do tamanho da mensagem de entrada, o valor de *hash* resultante possui tamanho fixo.

Entretanto, como a mensagem de entrada pode ter comprimento arbitrário, o número de entradas possíveis para essa função é infinito, enquanto o conjunto de valores de *hash*, embora extremamente grande, é finito. Essa característica implica a possibilidade de ocorrência de colisões, isto é, diferentes mensagens gerarem o mesmo valor de *hash*. Dessa forma, funções de *hash* criptográficas são projetadas para serem resistentes a colisões, mas não totalmente imunes a elas (SERAFIM, 2012) (IBM Data and AI Team, 2025).

2.4 GZIP

O *GZIP* é um projeto do *GNU* que realiza a compressão de dados por meio do algoritmo *DEFLATE*, o qual combina dois métodos: o *Lempel–Ziv 77 (LZ77)* e a codificação de *Huffman*. Essa combinação permite a compressão eficiente de uma ampla variedade de dados, explorando tanto redundâncias estruturais quanto estatísticas presentes na informação.

Como esses algoritmos não constituem o foco principal deste trabalho, seus funcionamentos não serão detalhados em profundidade; entretanto, uma explicação completa pode ser encontrada em (KADOOKA, 2025).

O algoritmo *DEFLATE* inicia o processo de compressão utilizando o *LZ77*, responsável por identificar padrões e sequências repetidas nos dados de entrada, substituindo-as por referências, como pares de deslocamento e comprimento, o que reduz significativamente a redundância estrutural. Em seguida, é aplicada a codificação de *Huffman* com árvores dinâmicas, a fim de compactar ainda mais os dados, atribuindo códigos menores aos símbolos mais frequentes.

2.5 Trabalhos Relacionados

2.5.1 *Desvendando as Nuvens: Uma Análise Comparativa de Desempenho de Aplicações Serverless e Containers em Provedores de Computação em Nuvem*

O trabalho intitulado *Desvendando as Nuvens: Uma Análise Comparativa de Desempenho de Aplicações Serverless e Containers em Provedores de Computação em Nuvem*”, de Valter, utiliza uma aplicação web *serverless* desenvolvida em JavaScript com Node.js. Essa aplicação é responsável por receber dados e imagens dos usuários para a realização de cadastro em um banco de dados, possibilitando posteriormente a utilização dessas informações. O estudo emprega essa aplicação como base para a comparação entre as plataformas Amazon AWS, Microsoft Azure e Google Cloud, analisando o custo-benefício de cada provedor em termos de velocidade de execução e custo monetário, por meio da realização de testes de carga incremental e de pico de carga (FILHO, 2024).

2.5.2 *Cloud Computing and Comparison based on Service and Performance between Amazon AWS, Microsoft Azure, and Google Cloud*

Outro trabalho que realiza uma comparação entre essas três plataformas é *Cloud Computing and Comparison based on Service and Performance between Amazon AWS, Microsoft Azure, and Google Cloud*. Nesse estudo, a comparação é conduzida por meio da análise de desempenho das três plataformas em ambientes virtuais idênticos, bem como da avaliação dos serviços oferecidos por cada provedor. Para a obtenção dos dados de desempenho, foi utilizada a aplicação de *benchmarking Phoronix Test Suite* versão 10.4, cujos resultados foram analisados e discutidos ao longo do trabalho (KAUSHIK et al., 2021).

2.5.3 Análise Comparativa de Escalabilidade e Disponibilidade de Bancos de Dados em Ambientes de Nuvem: um estudo dos serviços oferecidos por AWS e Azure

O trabalho *Análise Comparativa de Escalabilidade e Disponibilidade de Bancos de Dados em Ambientes de Nuvem: um estudo dos serviços oferecidos por AWS e Azure* realiza uma comparação entre a *Amazon AWS* e a *Microsoft Azure* no que diz respeito à escalabilidade e à disponibilidade oferecidas por serviços gerenciados de bancos de dados *MySQL*, especificamente o *Amazon RDS* e o *Azure Database for MySQL* (AMBROZIO; VICENTE, 2024).

2.5.4 Comparative Analysis of Virtual Machine Performance and Security Test Across Leading Cloud Providers

O trabalho intitulado *Comparative Analysis of Virtual Machine Performance and Security Test Across Leading Cloud Providers* apresenta uma análise comparativa do desempenho de máquinas virtuais das plataformas *Amazon Web Services (AWS)*, *Microsoft Azure* e *Google Cloud Platform (GCP)*, considerando aspectos como eficiência computacional, desempenho de rede e segurança em ambientes controlados. Para a avaliação do poder de processamento da CPU, da taxa de transferência de dados da memória, das operações de entrada e saída em disco, bem como da identificação de vulnerabilidades, foram utilizadas as ferramentas de *benchmarking* *Geekbench*, *Sysbench*, *Apache JMeter* e *OWASP ZAP* (ELEGBE; WIMMER, 2025).

2.5.5 A Comparative Analysis of Cloud Computing Services: AWS, Azure, and GCP

Assim como no trabalho anteriormente mencionado, o estudo intitulado *A Comparative Analysis of Cloud Computing Services: AWS, Azure, and GCP* também apresenta uma análise comparativa das máquinas virtuais oferecidas pelos provedores *Amazon Web Services (AWS)*,

Microsoft Azure e Google Cloud Platform (GCP). Nesse estudo, foram realizados cinco tipos de *benchmarking*: latência, *framework*, memória RAM, CPU e GPU. Além disso, são avaliados os preços e os serviços disponibilizados por cada provedor (ALKHATIB; SHAHEEN; ALBUSTANJI, 2024).

Capítulo 3

Desenvolvimento

3.1 Aplicações

As aplicações desenvolvidas para as análises compreendem a verificação de assinaturas digitais, a aplicação de funções de *hash* e a realização de compressões *GZIP* de um arquivo de 5,5KB. Todas foram implementadas utilizando a linguagem de programação *Python*, juntamente com o formato de dados *JSON* para o retorno dos resultados.

3.1.1 Bibliotecas do *Python*

Para o desenvolvimento da aplicação, foram utilizadas as bibliotecas *Cryptography*, *Hashlib*, *Gzip* e *Time*, da linguagem *Python*.

- ***Cryptography***

É um pacote da linguagem *Python* que pode ser dividido em dois níveis. O primeiro fornece receitas criptográficas seguras e de fácil utilização, como o *Fernet*. O segundo é composto por primitivas criptográficas de baixo nível, conhecidas como *hazmat*, que exigem maior cuidado em sua utilização, pois são mais suscetíveis a usos incorretos (The Cryptography Developers, 2025).

No contexto das assinaturas digitais, que utilizam criptografia RSA, foi empregada a camada *hazmat* para a geração das chaves e para a realização do processo de verificação.

- ***Hashlib***

O *hashlib* é um módulo da linguagem *Python* que provê uma interface comum e simplificada para diversos algoritmos de *hash* criptográfico seguros. Neste trabalho, foi utilizado o algoritmo de *hash SHA-256*, o qual está sempre disponível nesse módulo (Python Software Foundation, 2026b).

- ***GZIP***

A biblioteca *gzip* da linguagem *Python* disponibiliza funções que realizam a compressão e a descompressão de arquivos de maneira equivalente aos programas do *GNU*, *gzip* e *gunzip*. Dessa forma, esse módulo foi utilizado para a aplicação das compressões *GZIP* analisadas neste trabalho (Python Software Foundation, 2026a).

- ***Time***

Módulo da linguagem *Python* que provê diversas funções relacionadas ao tempo. Dessa forma, foi utilizado para medir o intervalo necessário para a aplicação realizar as verificações (Python Software Foundation, 2025).

3.2 Testes de Desempenho

O projeto será executado nas plataformas *AWS Lambda* e *Azure Functions*, ambas localizadas no Brasil. Os códigos realizarão 1, 10, 100, 1 000 e 10 000 verificações de assinaturas digitais, aplicações de funções de *hash* e compressões *GZIP*, retornando o tempo de execução e a quantidade de operações por segundo.

3.2.1 *Azure Functions*

Para a utilização do *Azure Functions*, foi empregada uma conta *Azure for Students*, que permite o acesso gratuito a mais de 65 serviços até determinados limites, conforme o plano de hospedagem escolhido, além da disponibilização de US 100 em créditos não cumulativos a cada 12 meses, tudo isso sem a necessidade de cadastro de cartão de crédito. Contas pagas também podem utilizar os mesmos recursos gratuitamente dentro desses limites; entretanto, caso esses limites sejam ultrapassados, os valores excedentes serão cobrados.

O Azure disponibiliza quatro principais planos de hospedagem para aplicativos de funções, cada um projetado para atender a necessidades distintas de desempenho, custo e escalabilidade: Plano de Consumo, Plano de Consumo Flexível, Plano Premium e Plano Dedicado.

O Plano de Consumo caracteriza-se por um modelo de cobrança baseado exclusivamente no tempo de execução das funções, no qual os recursos computacionais são alocados apenas durante o processamento das requisições. Nesse modelo, o escalonamento ocorre de forma automática, com a adição ou remoção dinâmica de instâncias conforme o volume de eventos de entrada, configurando uma abordagem totalmente *serverless* e eficiente, inclusive em cenários de alta carga.

Uma variação desse modelo é o Plano de Consumo Flexível, que mantém a cobrança conforme o uso, porém amplia as opções de configuração de computação e conectividade. Esse plano permite ajustar dinamicamente o número de instâncias não apenas em função da quantidade de eventos, mas também com base em parâmetros de simultaneidade definidos por instância. Além disso, possibilita o pré-provisionamento de instâncias, reduzindo significativamente o impacto das inicializações a frio (*cold starts*). O suporte à integração com redes virtuais adiciona uma camada extra de segurança, tornando esse plano adequado para aplicações que exigem maior controle sobre o ambiente de execução sem comprometer a elasticidade automática.

O Plano Premium foi desenvolvido para cenários que demandam desempenho consistente e execução contínua ou quase contínua. Ele utiliza instâncias pré-ativadas, eliminando atrasos decorrentes de períodos de inatividade e garantindo respostas imediatas às requisições. Esse plano opera em ambientes com maior capacidade de processamento e memória, oferece suporte a redes virtuais e possibilita o uso de imagens personalizadas do sistema operacional Linux. Dessa forma, torna-se apropriado para aplicações com alto volume de execuções, uso intensivo de recursos computacionais ou requisitos específicos de conectividade e isolamento.

Por fim, o Plano Dedicado executa funções dentro de um plano tradicional do Serviço de Aplicativo, com cobrança baseada em valores fixos. Esse modelo é indicado para aplicações com execuções prolongadas, como aquelas que utilizam *Durable Functions*, as quais não são compatíveis com outros planos. Além disso, permite o aproveitamento de máquinas virtuais

subutilizadas e oferece suporte tanto ao escalonamento manual quanto ao automático. O Plano Dedicado possibilita a hospedagem de múltiplos aplicativos web e funções em um mesmo ambiente, disponibilizando maior capacidade de CPU, memória elevada e isolamento completo de computação, sendo especialmente adequado para cenários corporativos mais exigentes.

O plano escolhido foi o Plano de Consumo Flexível, por adotar um modelo de cobrança conforme o uso, assim como o Plano de Consumo tradicional, porém oferecendo maior controle sobre o desempenho e os custos.

Considerando esse plano, as restrições para a utilização gratuita da ferramenta correspondem a 100 000 GB/s de tempo de execução sob demanda e a um total de 250 000 execuções sob demanda por mês, sem o uso de instâncias sempre prontas (Microsoft Corporation, 2025).

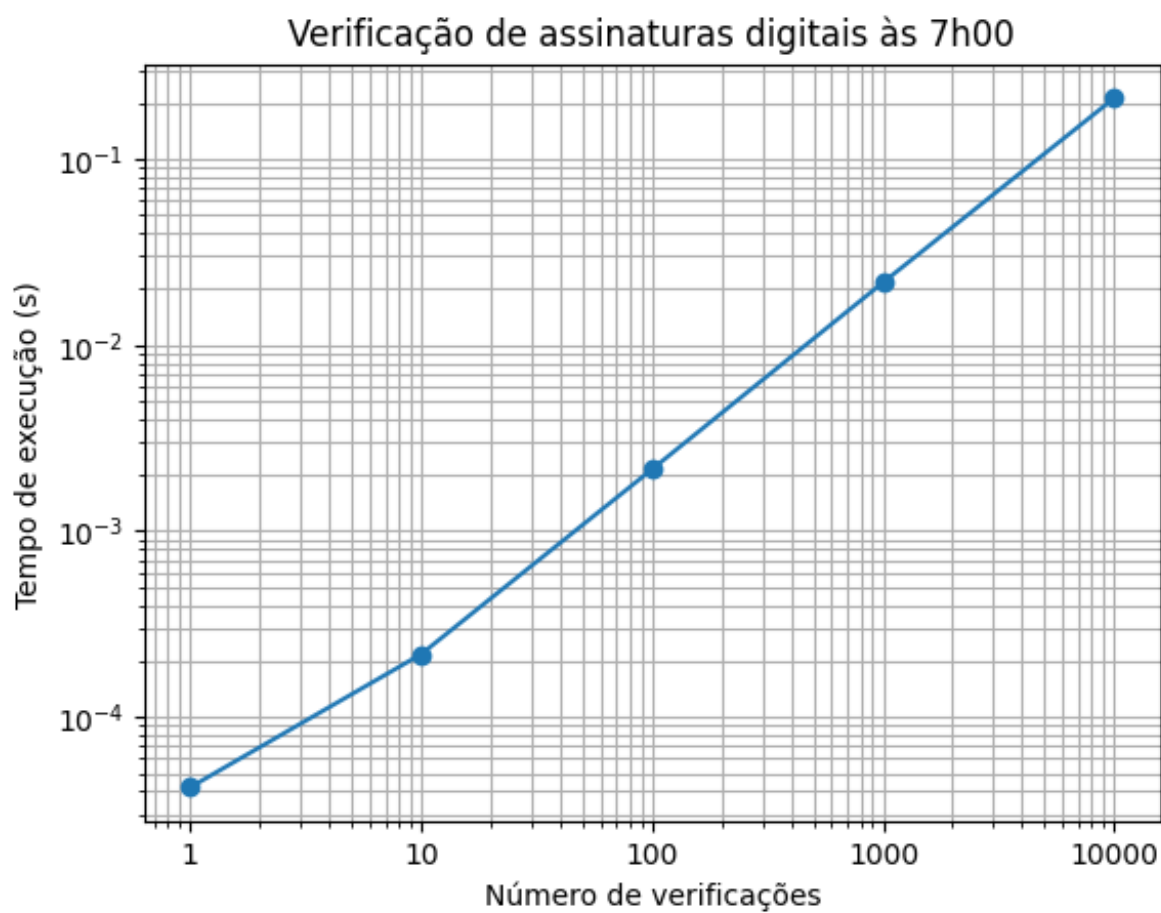
Após a execução das aplicações em três horários distintos, os resultados obtidos, utilizando uma CPU virtual na região *Brazil South*, serão apresentados nos itens a seguir. Figura 2.1.

- **Assinaturas Digitais**

A partir da realização das verificações das assinaturas digitais, obtiveram-se os resultados apresentados no Gráfico 3.1, às 7h00, no Gráfico 3.2, às 15h00, e no Gráfico 3.3, às 21h00.

A Figura 3.1 foi feita a partir dos resultados obtidos pela execução da aplicação de verificação de assinaturas digitais às 7h00 na plataforma Azure, considerando 1, 10, 100, 1.000 e 10.000 execuções. Observa-se que o gráfico possui comportamento estritamente crescente, evidenciando que o desempenho da aplicação aumenta de forma contínua conforme se eleva o número de execuções. A partir de 10 execuções, o crescimento passa a apresentar um comportamento aproximadamente linear, indicando maior estabilidade e previsibilidade do desempenho.

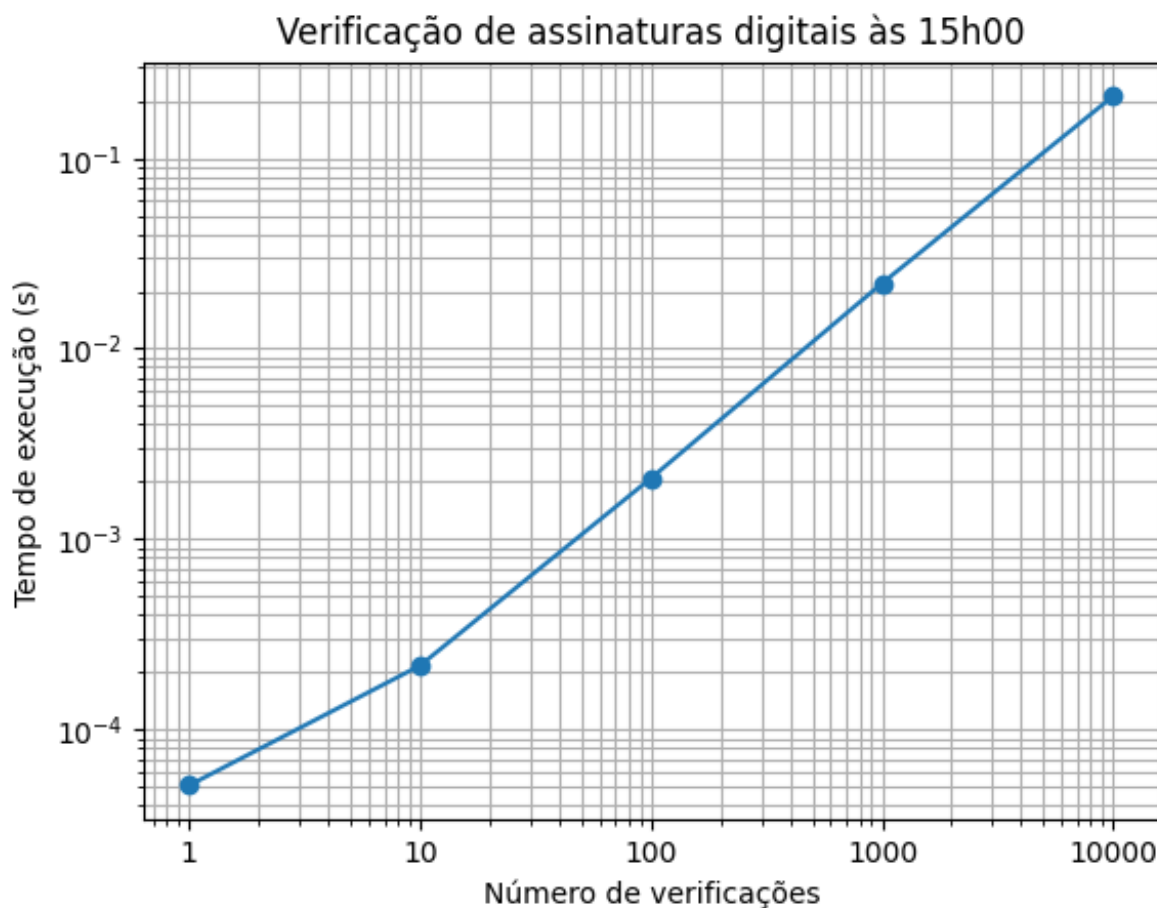
Na Figura 3.2 está ilustrado o desempenho da aplicação de verificação de assinaturas digitais às 15h00 na plataforma Azure, sob as mesmas quantidades de execução analisadas. Assim como observado no horário anterior, o gráfico é estritamente crescente, demonstrando consistência nos resultados obtidos. Verifica-se que, após a fase inicial,

Figura 3.1: Desempenho da aplicação de assinaturas digitais às 7h00 na *Azure*.

Fonte: Elaborado pela autora.

o comportamento do gráfico tende à linearidade a partir de 10 execuções.

Figura 3.2: Desempenho da aplicação de assinaturas digitais às 15h00 na *Azure*.



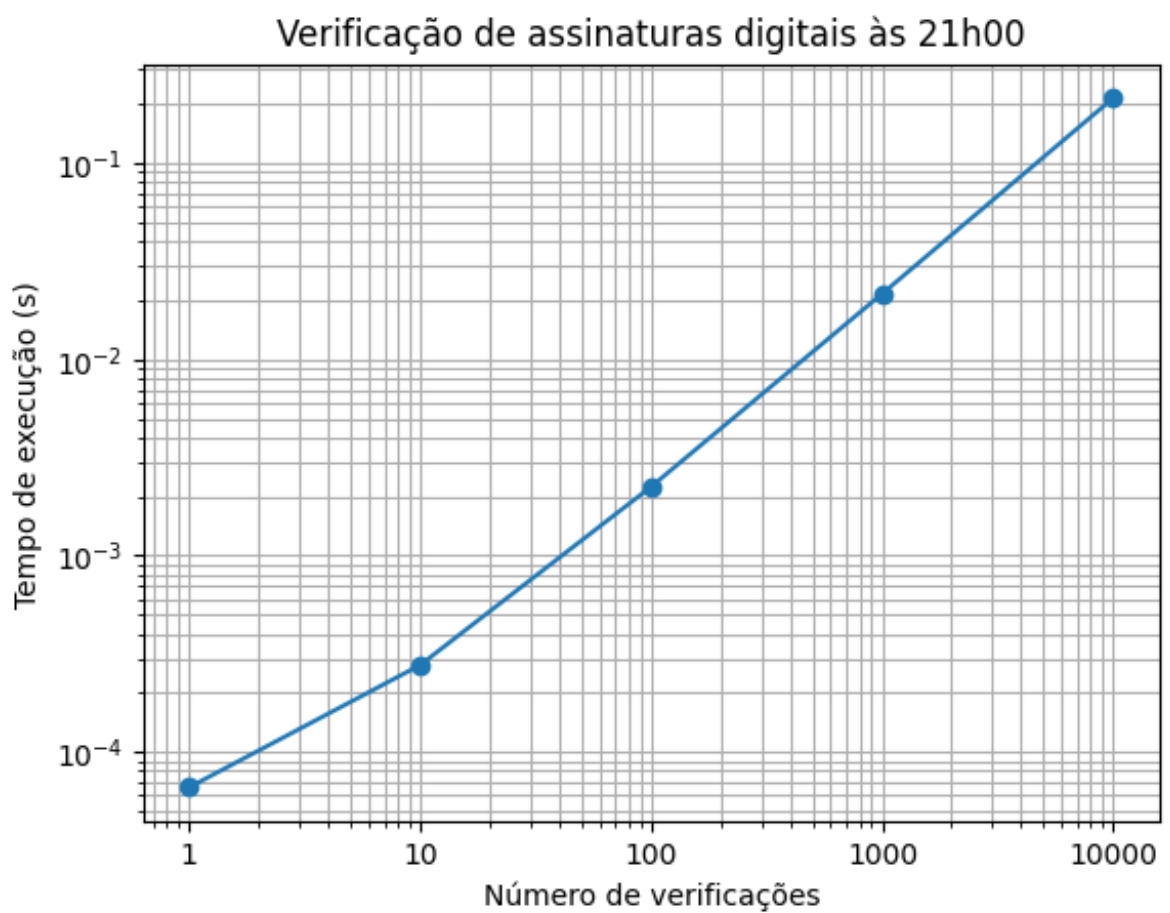
Fonte: Elaborado pela autora.

Por fim, na Figura 3.3 observa-se os resultados referentes às execuções realizadas às 21h00 na plataforma Azure. O gráfico mantém o comportamento estritamente crescente observado nos demais horários, reforçando a repetibilidade e a confiabilidade dos experimentos. Nota-se novamente que, a partir de 10 execuções, o desempenho da aplicação passa a seguir uma tendência linear.

• Funções Hash

A partir da execução das funções de hash, obtiveram-se os resultados apresentados no Gráfico 3.4, às 7h00, no Gráfico 3.5, às 15h00, e no Gráfico 3.6, às 21h00.

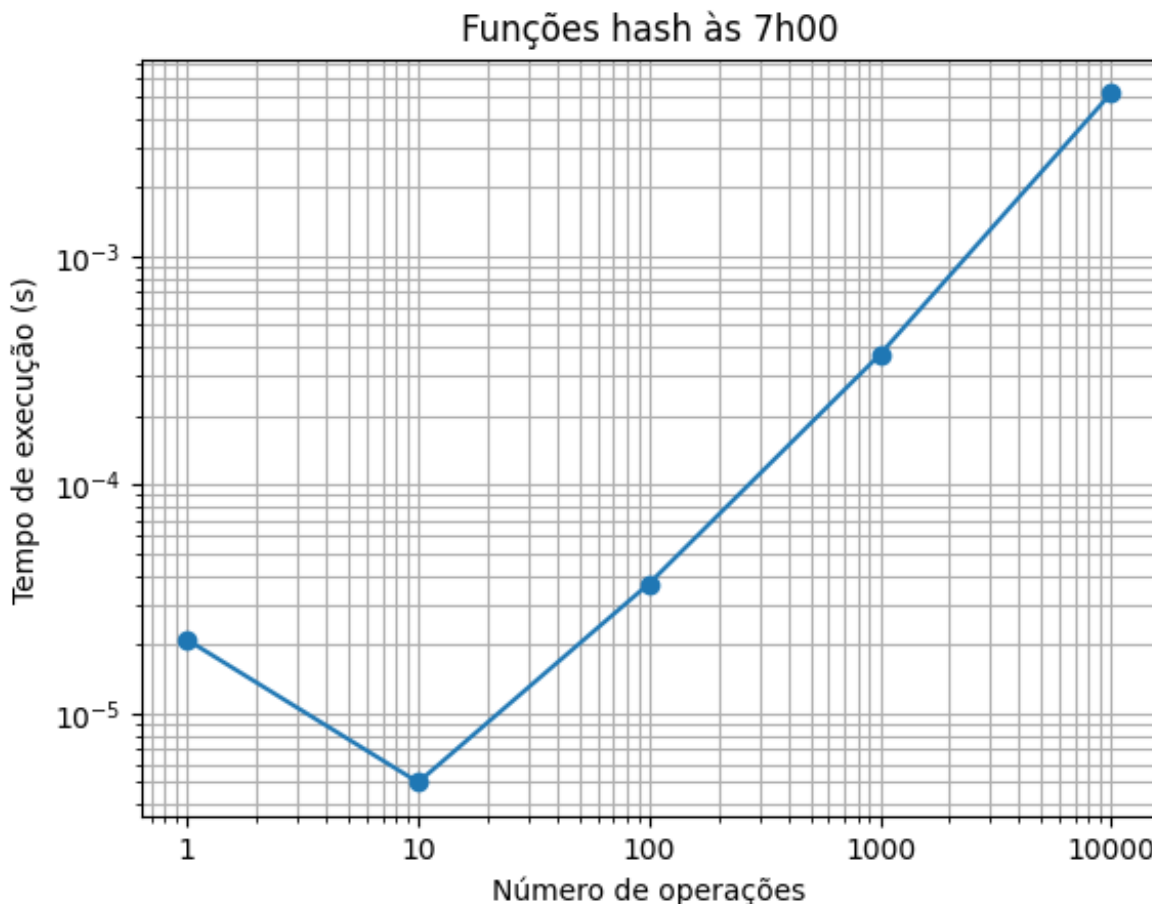
A Figura 3.4 foi elaborada a partir dos resultados obtidos pela execução da aplicação de funções hash às 7h00 na plataforma Azure, considerando 1, 10, 100, 1.000 e 10.000

Figura 3.3: Desempenho da aplicação de assinaturas digitais às 21h00 na *Azure*.

Fonte: Elaborado pela autora.

execuções. Observa-se que, diferentemente do comportamento estritamente crescente verificado em outras aplicações, o gráfico apresenta uma redução inicial entre 1 e 10 execuções. A partir de 10 execuções, o desempenho passa a crescer de forma contínua, apresentando uma tendência aproximadamente linear.

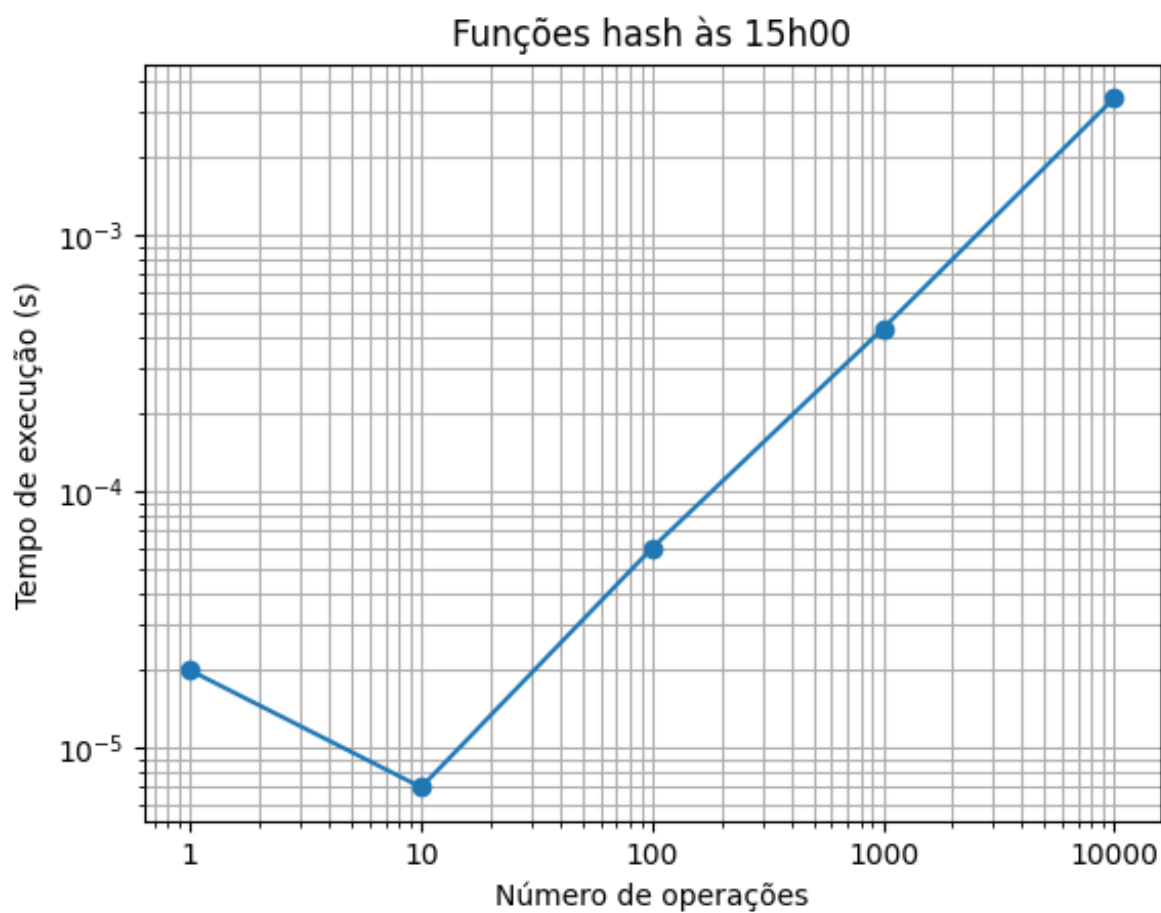
Figura 3.4: Desempenho da aplicação de funções hash às 7h00 na Azure.



Fonte: Elaborado pela autora.

Na Figura 3.5, está ilustrado o desempenho da aplicação de funções hash às 15h00 na plataforma Azure, observa-se comportamento semelhante ao identificado no horário anterior. O gráfico apresenta um decréscimo inicial entre 1 e 10 execuções, seguido por um crescimento progressivo à medida que o número de execuções aumenta, aproximando-se de um comportamento linear.

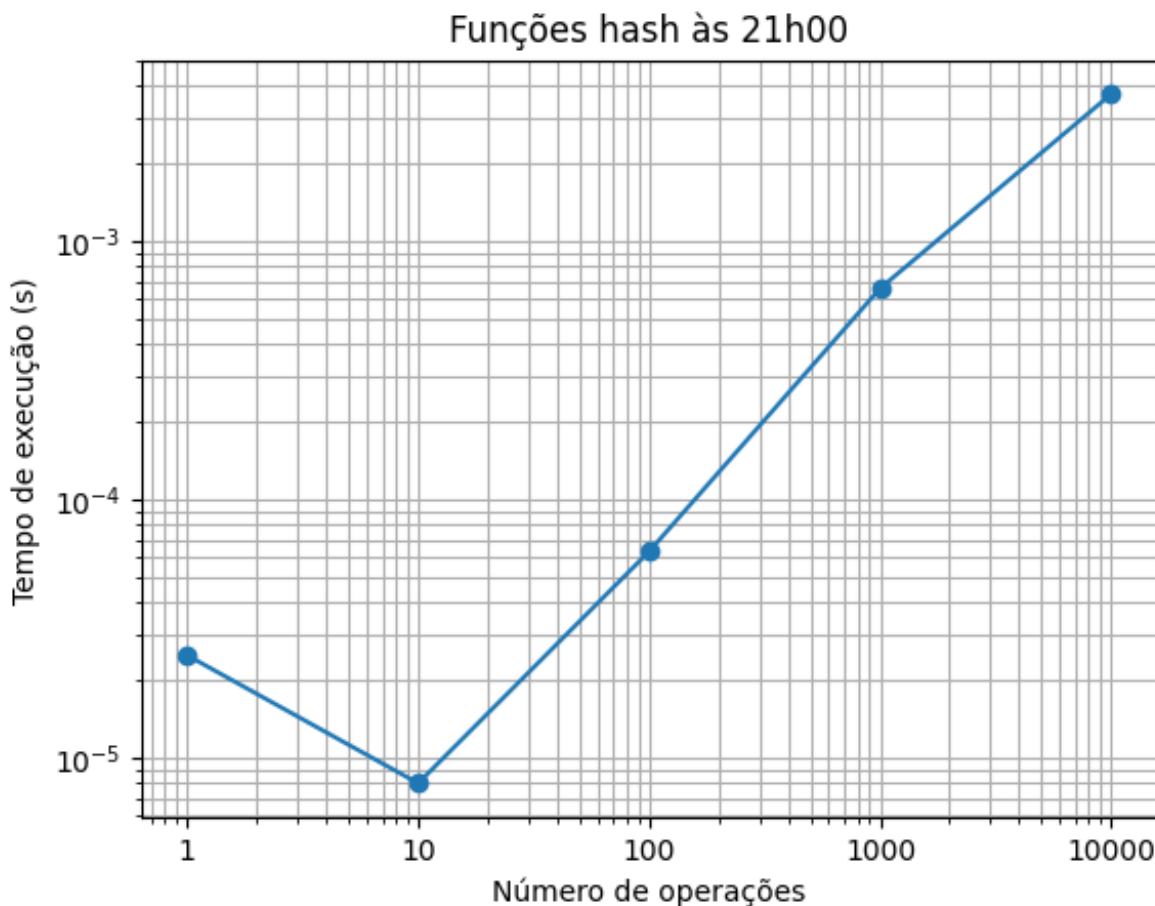
Por fim, na Figura 3.6 apresenta-se os resultados referentes às execuções da aplicação de funções hash realizadas às 21h00 na plataforma Azure. Observa-se novamente um comportamento inicial decrescente entre 1 e 10 execuções, seguido por um crescimento

Figura 3.5: Desempenho da aplicação de funções hash às 15h00 na *Azure*.

Fonte: Elaborado pela autora.

contínuo e relativamente linear nas execuções subsequentes.

Figura 3.6: Desempenho da aplicação de funções hash às 21h00 na *Azure*.



Fonte: Elaborado pela autora.

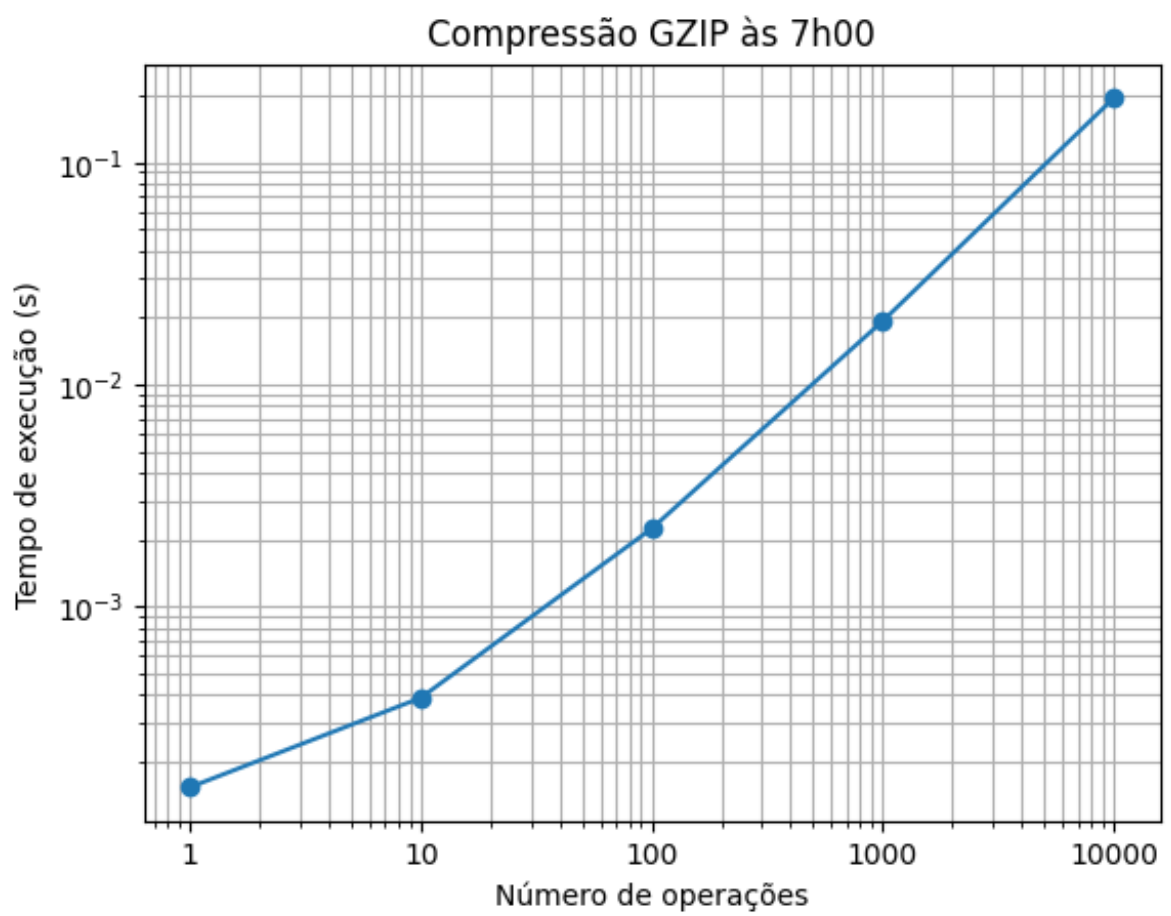
- **Compressões GZIP**

A partir da realização das compressões GZIP, obtiveram-se os resultados apresentados no Gráfico 3.7, às 7h00, no Gráfico 3.8, às 15h00, e no Gráfico 3.9, às 21h00.

Na Figura 3.7 observa-se os resultados obtidos pela execução da aplicação de compressão GZIP às 7h00 na plataforma Azure, considerando 1, 10, 100, 1.000 e 10.000 execuções. Observa-se que o gráfico apresenta comportamento estritamente crescente, indicando aumento contínuo do desempenho conforme se eleva o número de execuções. A partir de 10 execuções, o comportamento do gráfico torna-se aproximadamente linear.

A Figura 3.8 foi obtida com os resultados do desempenho da aplicação de compressão

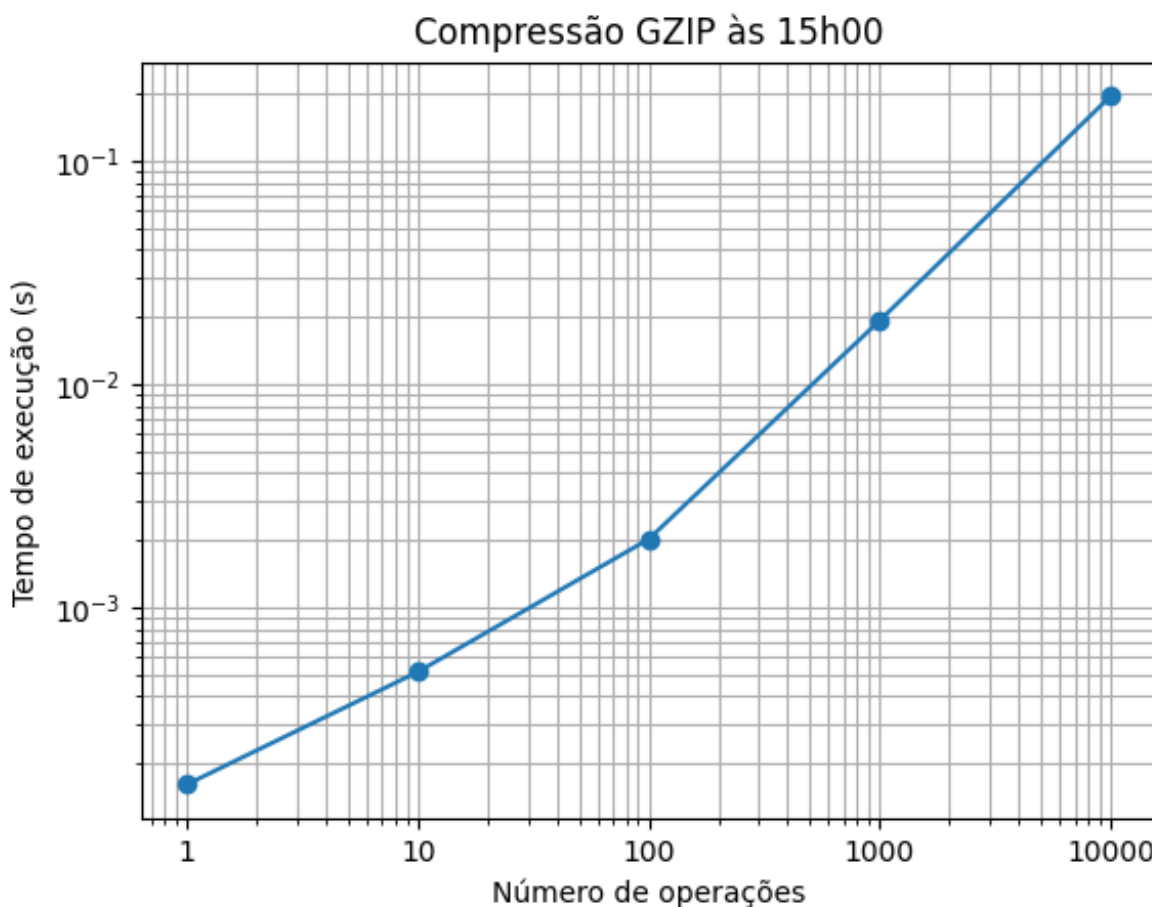
Figura 3.7: Desempenho da aplicação de compressão GZIP às 7h00 na Azure.



Fonte: Elaborado pela autora.

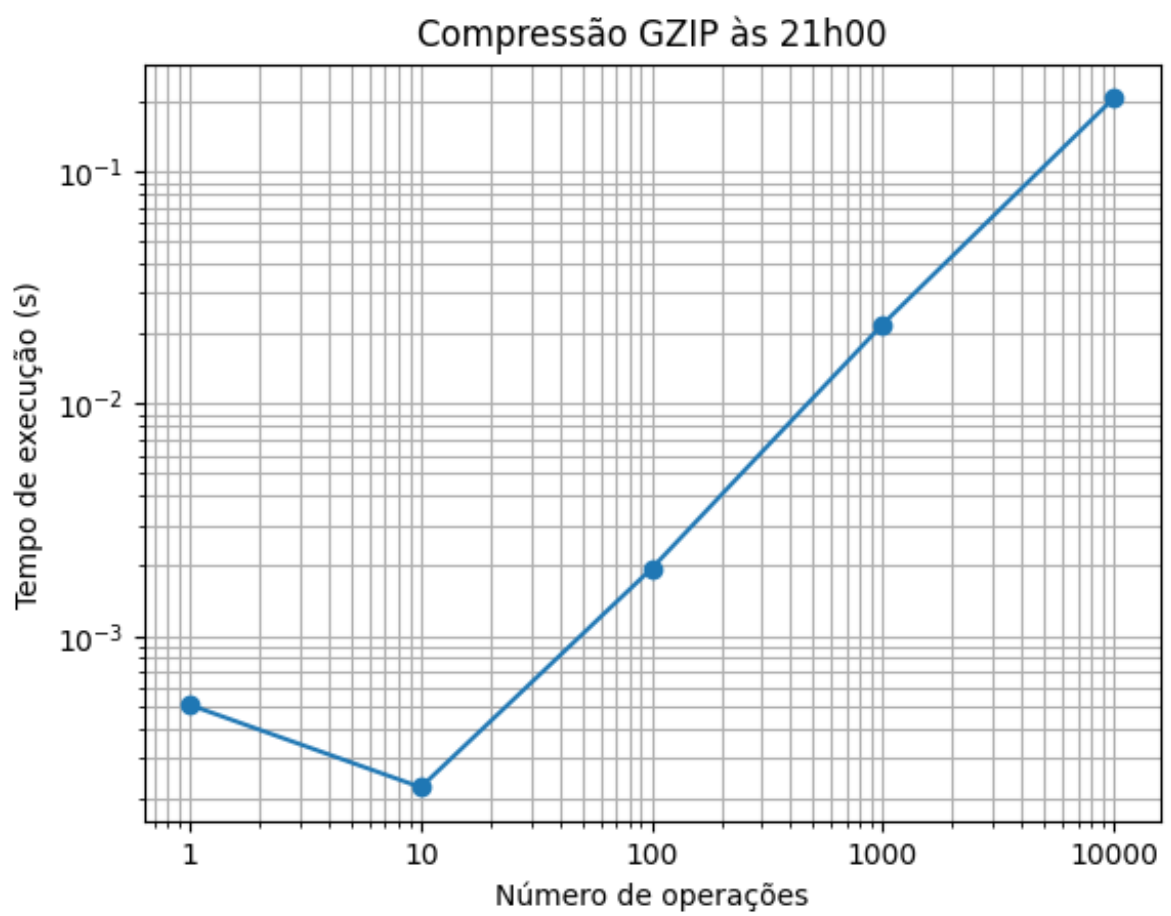
GZIP às 15h00 na plataforma Azure. Assim como observado no horário das 7h00, o gráfico mantém um comportamento estritamente crescente ao longo de todas as quantidades de execução analisadas. Nota-se que, após a fase inicial, o desempenho passa a crescer de forma mais regular, apresentando tendência aproximadamente linear a partir de 10 execuções.

Figura 3.8: Desempenho da aplicação de compressão GZIP às 15h00 na *Azure*.



Fonte: Elaborado pela autora.

Por fim, na Figura 3.9 observa-se os resultados referentes às execuções realizadas às 21h00 na plataforma Azure. Diferentemente dos horários anteriores, observa-se um comportamento inicial decrescente entre 1 e 10 execuções, seguido por um crescimento contínuo nas execuções subsequentes. A partir de 10 execuções, o gráfico passa a apresentar um comportamento aproximadamente linear, indicando maior estabilidade no desempenho da aplicação. Esse padrão é semelhante ao observado na aplicação de funções hash.

Figura 3.9: Desempenho da aplicação de compressão GZIP às 21h00 na *Azure*.

Fonte: Elaborado pela autora.

3.2.2 *AWS Lambda*

O tipo de conta utilizada para o *AWS Lambda* foi o nível gratuito da *AWS*. Embora a *AWS* também disponha de contas voltadas para estudantes, essas são mais focadas em atividades de ensino, apresentando ferramentas limitadas aos conteúdos de aprendizado (Amazon Web Services, Inc., 2025a).

A conta gratuita oferece US 100 em créditos e permite o uso de aproximadamente 97 ferramentas da *AWS* até limites específicos para cada serviço, sem qualquer cobrança durante os primeiros seis meses; entretanto, é necessário cadastrar um cartão de crédito. Após esse período, é possível continuar utilizando os serviços dentro das mesmas limitações em uma conta paga; contudo, caso os limites sejam ultrapassados, serão aplicadas as cobranças correspondentes.

Assim como o Plano de Consumo Flexível da *Azure*, o *AWS Lambda* também é *serverless*, possui escalonamento automático e adota cobrança conforme o uso. Conseqüentemente, esse serviço foi selecionado para a execução da aplicação.

Os limites para o uso gratuito desse serviço são de até 400 000 GB/s de tempo de computação ou 3,2 milhões de segundos de execução, além de 1 000 000 de solicitações gratuitas por mês (Amazon Web Services, Inc., 2025b).

Ao executar a aplicação utilizando uma CPU virtual na região *South America (Sao Paulo)*, obtiveram-se os resultados abaixo.

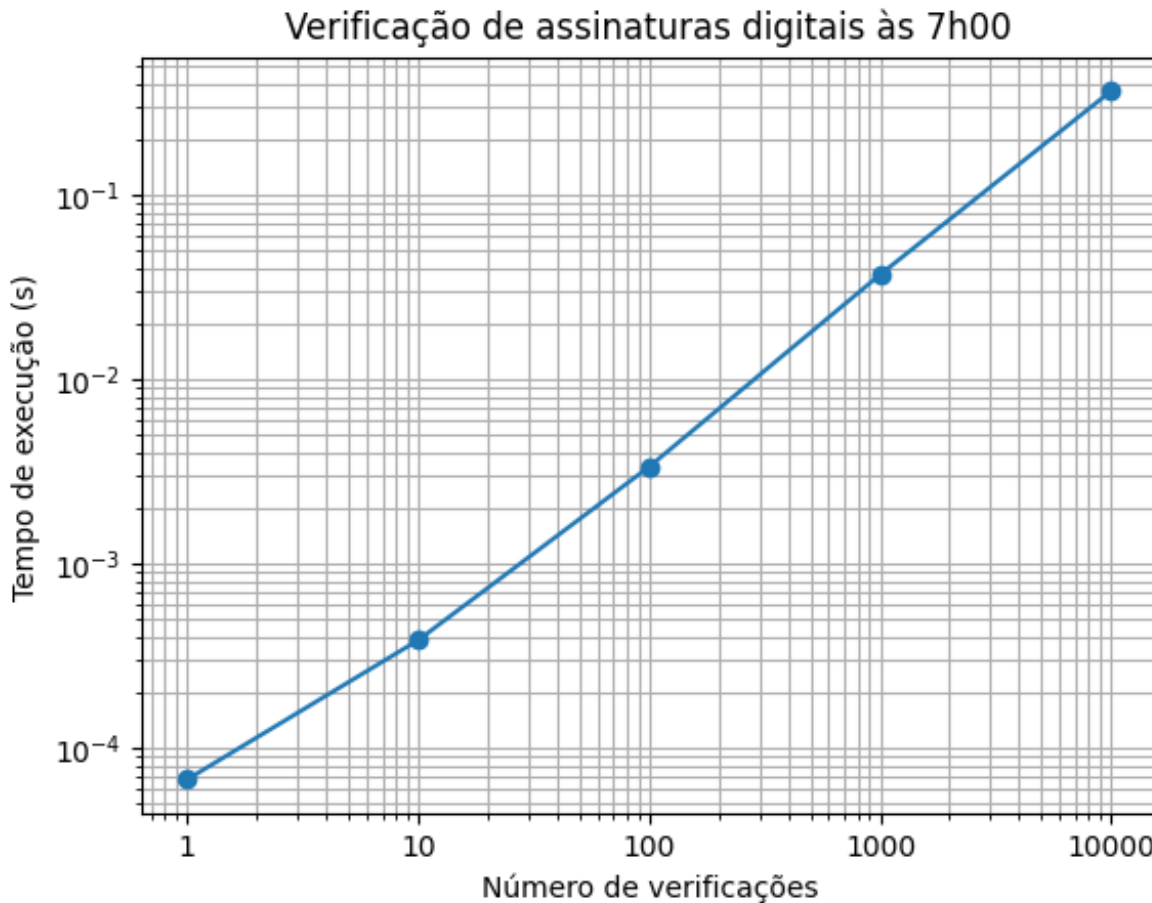
- **Assinaturas Digitais**

A partir da realização das verificações das assinaturas digitais, obtiveram-se os resultados apresentados no Gráfico 3.10, às 7h00, no Gráfico 3.11, às 15h00, e no Gráfico 3.12, às 21h00.

A Figura 3.10 foi elaborada a partir dos resultados obtidos pela execução da aplicação de verificação de assinaturas digitais às 7h00 na plataforma *AWS*, considerando 1, 10, 100, 1.000 e 10.000 execuções. Observa-se que o gráfico apresenta comportamento estritamente crescente, indicando aumento contínuo do desempenho à medida que o número de execuções é ampliado. A partir de 10 execuções, o crescimento torna-se

aproximadamente linear, evidenciando maior estabilidade e previsibilidade no desempenho da aplicação.

Figura 3.10: Desempenho da aplicação de assinaturas digitais às 7h00 na AWS.

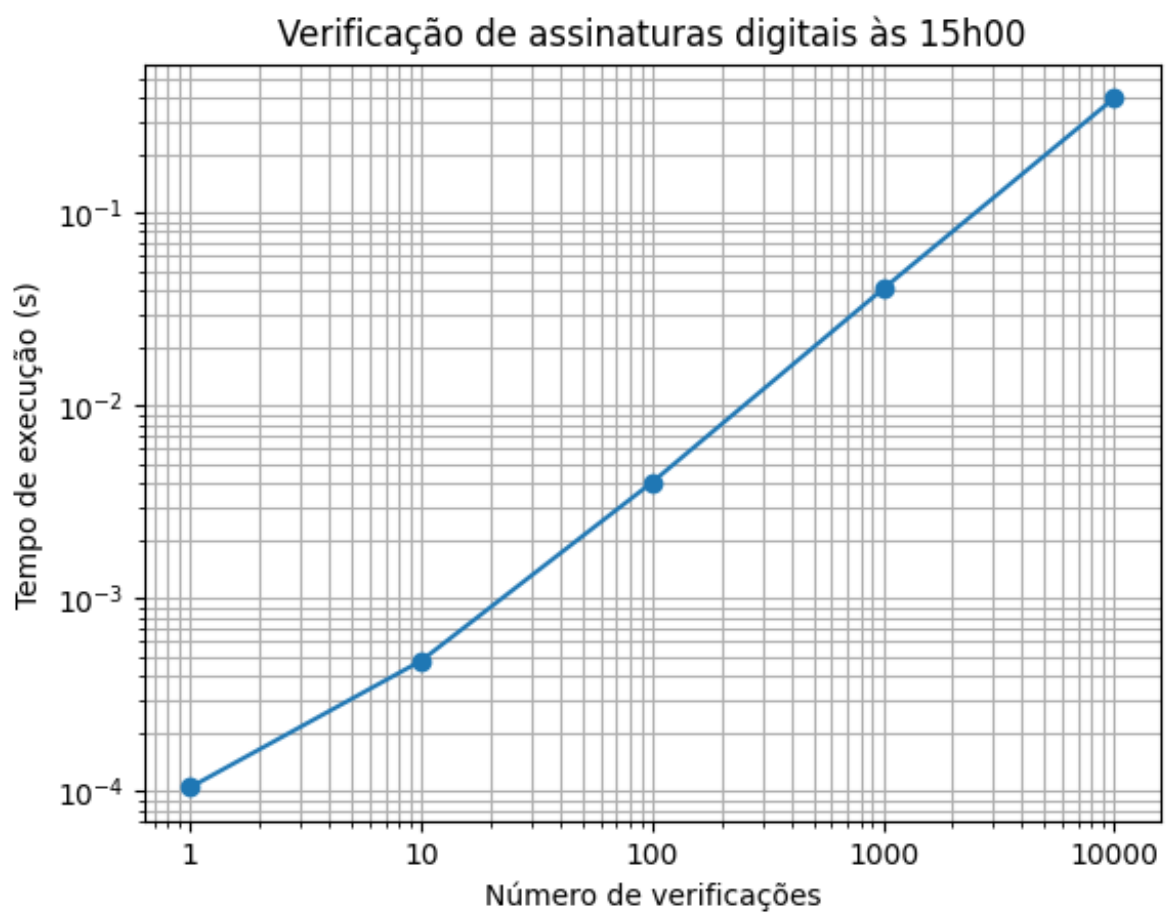


Fonte: Elaborado pela autora.

A Figura 3.11 representa o desempenho da aplicação de verificação de assinaturas digitais às 15h00 na plataforma AWS. Assim como observado no horário das 7h00, o gráfico mantém comportamento estritamente crescente ao longo de todas as quantidades de execução analisadas. Nota-se que, após a fase inicial, o desempenho passa a crescer de forma mais regular, apresentando tendência aproximadamente linear a partir de 10 execuções.

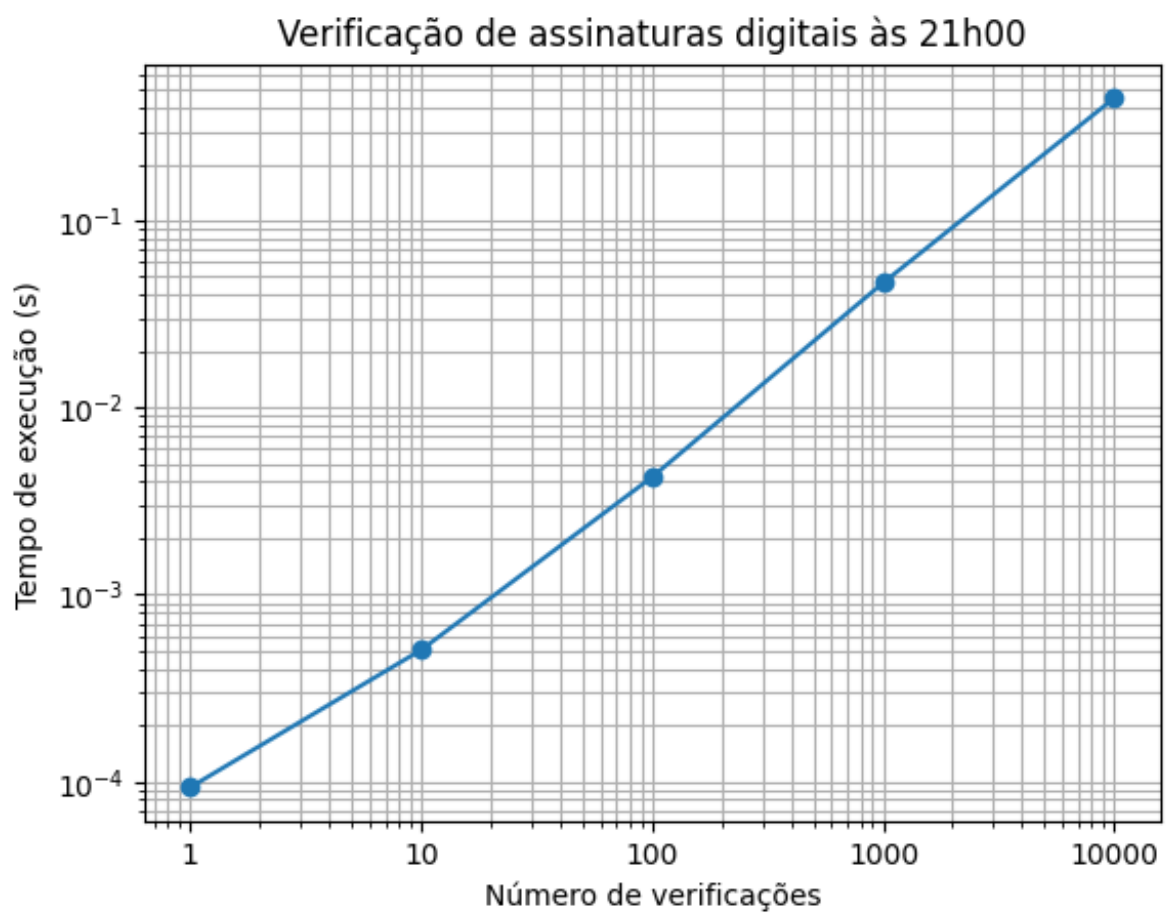
Por fim, a Figura 3.12 é obtido dos resultados referentes às execuções realizadas às 21h00 na plataforma AWS. O gráfico mantém o comportamento estritamente crescente observado nos demais horários. A partir de 10 execuções, observa-se novamente uma tendência de crescimento aproximadamente linear.

Figura 3.11: Desempenho da aplicação de assinaturas digitais às 15h00 na AWS.



Fonte: Elaborado pela autora.

Figura 3.12: Desempenho da aplicação de assinaturas digitais às 21h00 na AWS.



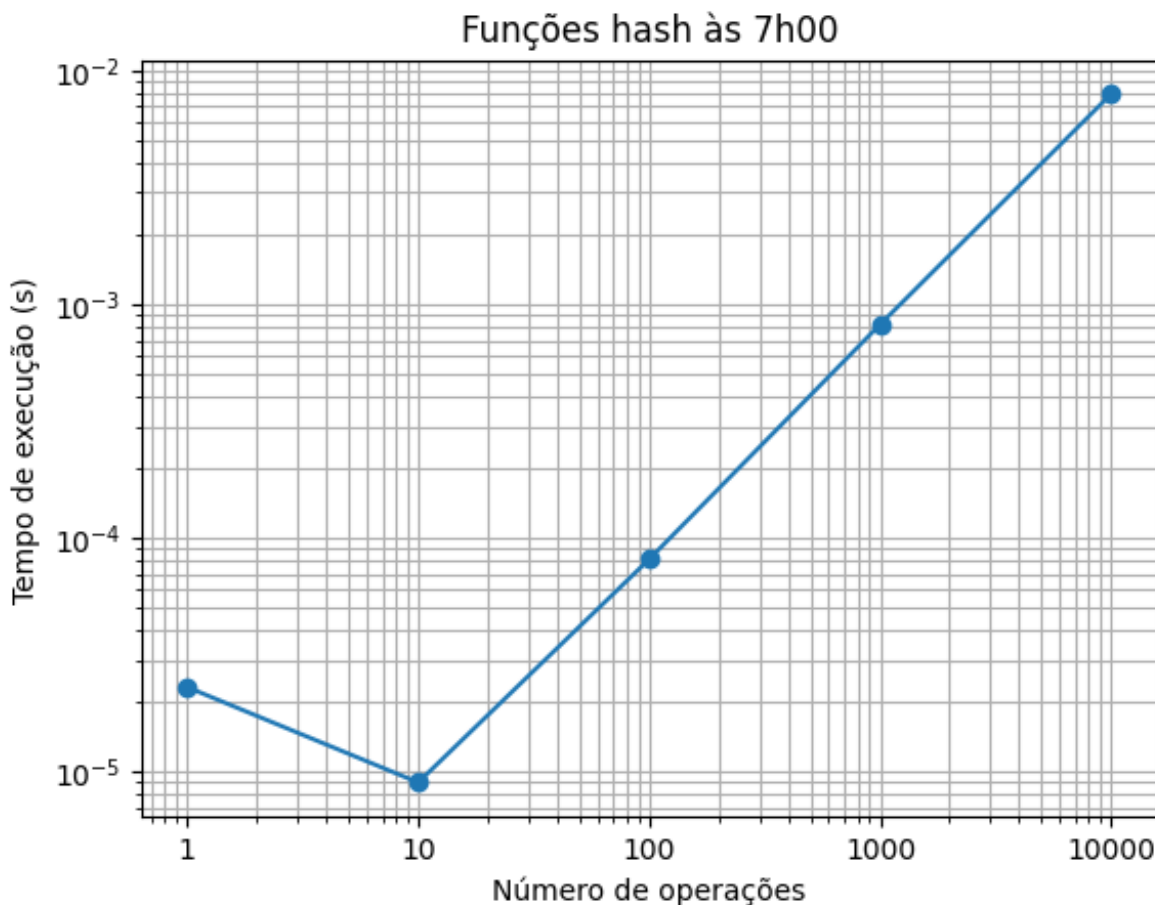
Fonte: Elaborado pela autora.

- **Funções Hash**

A partir da execução das funções de hash, obtiveram-se os resultados apresentados no Gráfico 3.13, às 7h00, no Gráfico 3.14, às 15h00, e no Gráfico 3.15, às 21h00.

A Figura 3.13 foi obtida dos resultados obtidos pela execução da aplicação de funções hash às 7h00 na plataforma AWS, considerando 1, 10, 100, 1.000 e 10.000 execuções. Observa-se que o gráfico apresenta um comportamento inicial decrescente entre 1 e 10 execuções. A partir de 10 execuções, o desempenho passa a crescer de forma contínua, apresentando uma tendência aproximadamente linear.

Figura 3.13: Desempenho da aplicação de funções hash às 7h00 na AWS.

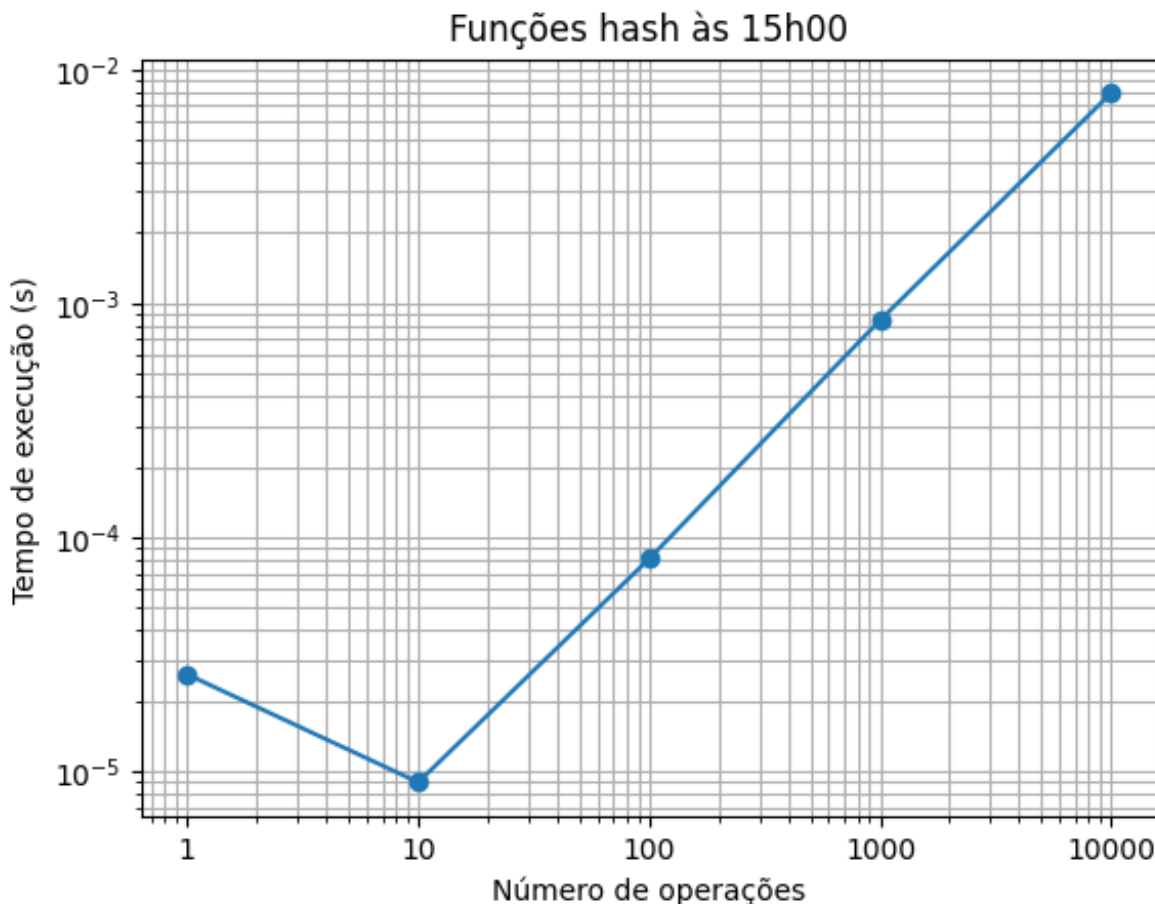


Fonte: Elaborado pela autora.

Na Figura 3.14 observa-se o desempenho da aplicação de funções hash às 15h00 na plataforma AWS. De forma semelhante ao observado às 7h00, o gráfico apresenta um decréscimo inicial entre 1 e 10 execuções, seguido por um crescimento progressivo conforme o número de execuções aumenta. Após esse ponto, o comportamento do

gráfico torna-se mais estável e previsível, aproximando-se de uma relação linear a partir de 10 execuções.

Figura 3.14: Desempenho da aplicação de funções hash às 15h00 na AWS.



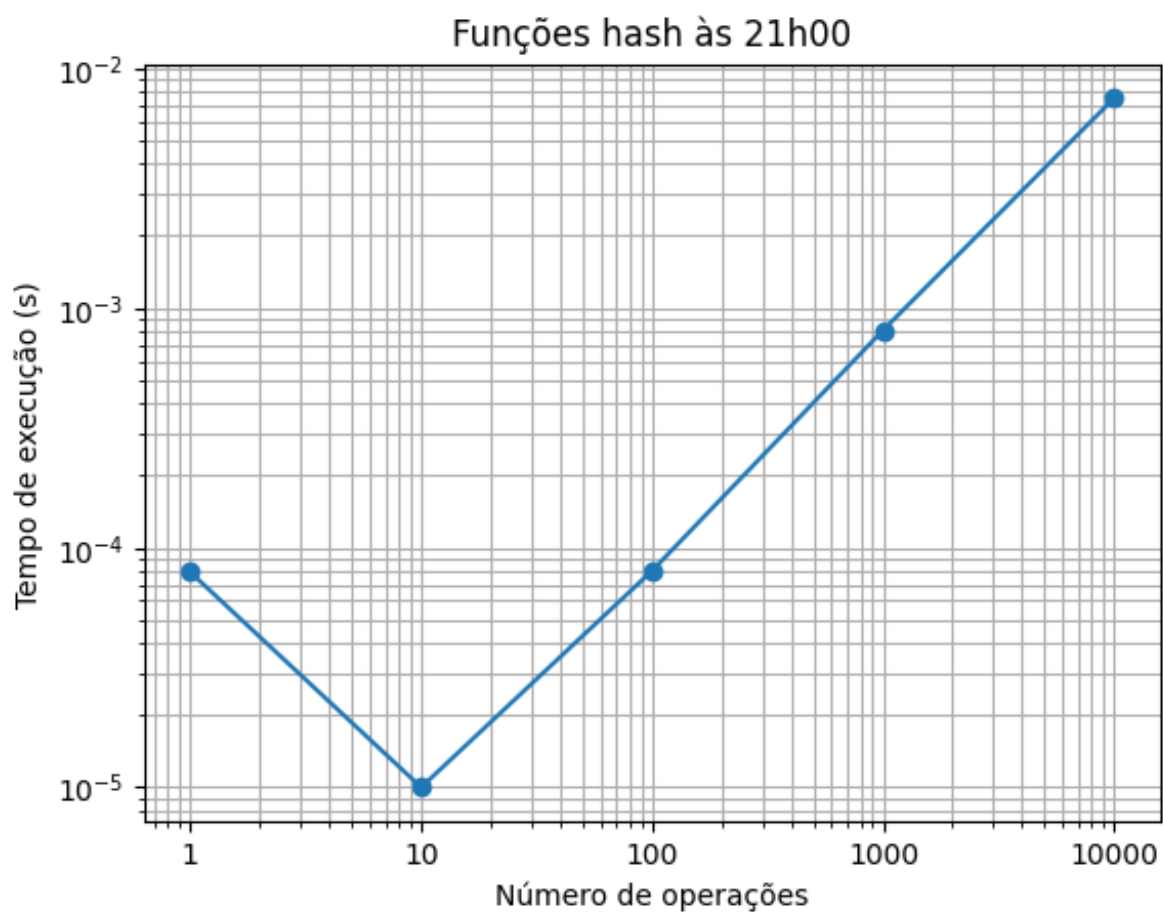
Fonte: Elaborado pela autora.

Por fim, a Figura 3.15 elaborada com os resultados referentes às execuções realizadas às 21h00 na plataforma AWS. Observa-se novamente um comportamento inicial decrescente entre 1 e 10 execuções, seguido por um crescimento contínuo nas execuções subsequentes. A partir de 10 execuções, o gráfico passa a apresentar um comportamento aproximadamente linear, indicando maior estabilidade no desempenho da aplicação. Esse resultado é consistente com o comportamento observado para a aplicação de funções hash na plataforma Azure.

- **Compressões GZIP**

A partir da realização das compressões GZIP, obtiveram-se os resultados apresentados

Figura 3.15: Desempenho da aplicação de funções hash às 21h00 na AWS.

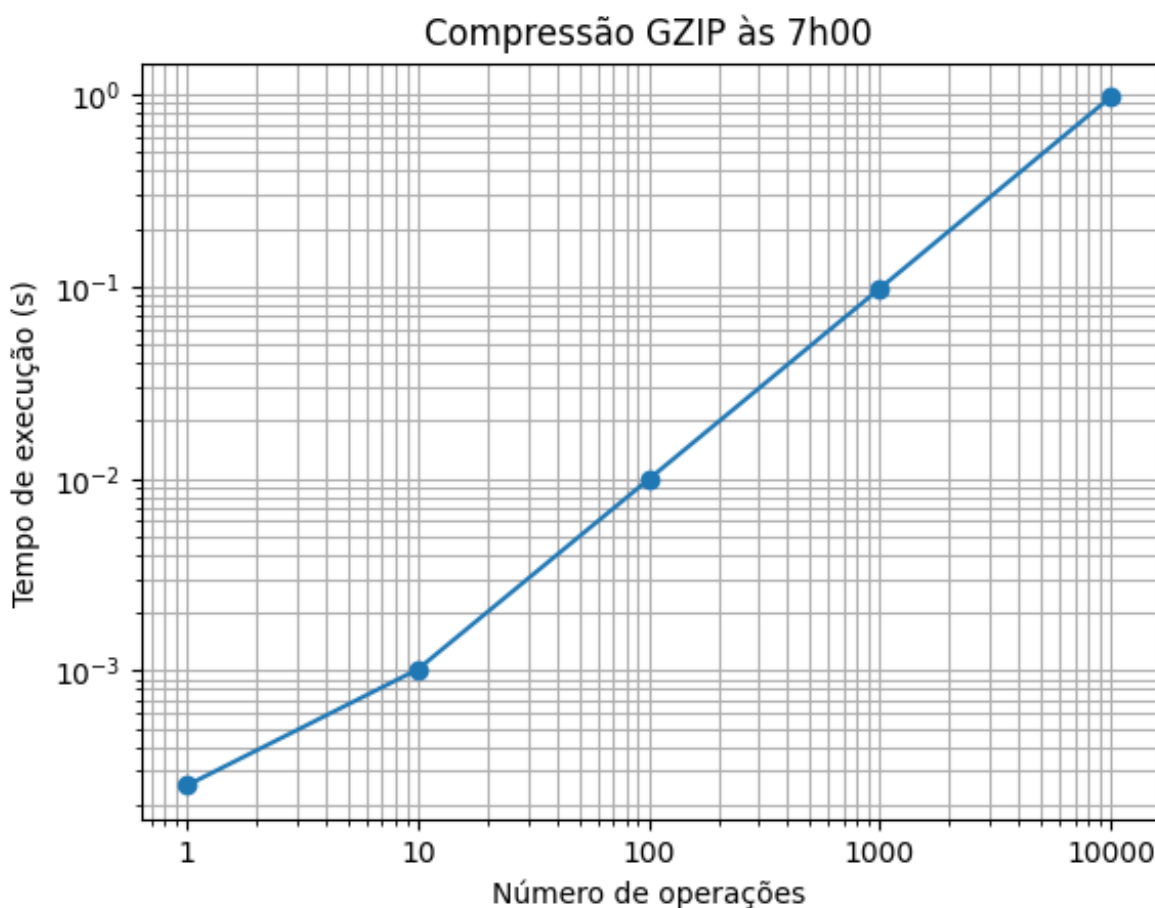


Fonte: Elaborado pela autora.

no Gráfico 3.16, às 7h00, no Gráfico 3.17, às 15h00, e no Gráfico 3.18, às 21h00.

Na Figura 3.16 observa-se os resultados obtidos pela execução da aplicação de compressão GZIP às 7h00 na plataforma AWS, considerando 1, 10, 100, 1.000 e 10.000 execuções. Observa-se que o gráfico apresenta comportamento estritamente crescente ao longo de todas as quantidades de execução analisadas, indicando aumento contínuo do desempenho conforme o número de execuções é ampliado. A partir de 10 execuções, o crescimento passa a apresentar comportamento aproximadamente linear, evidenciando maior estabilidade no desempenho da aplicação.

Figura 3.16: Desempenho da aplicação de compressão GZIP às 7h00 na AWS.

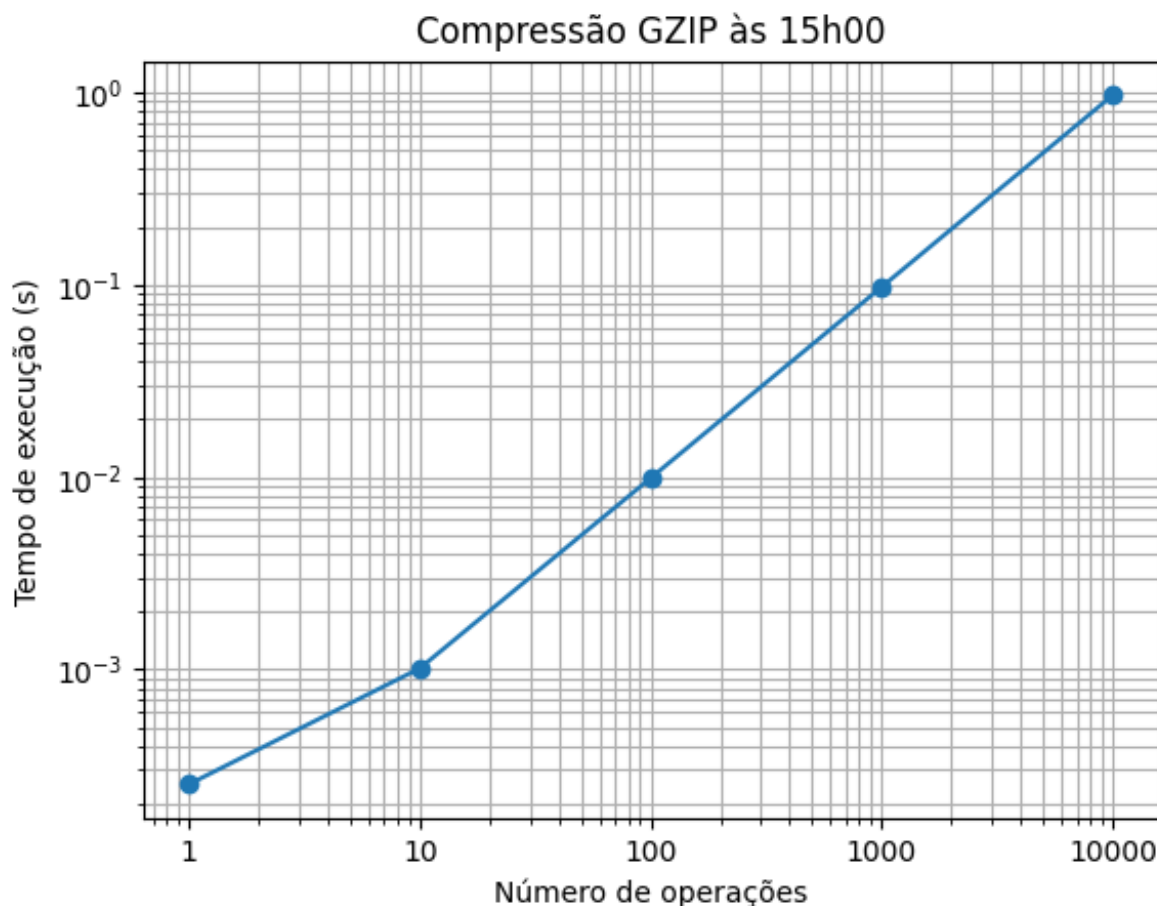


Fonte: Elaborado pela autora.

A Figura 3.17 elaborada com os resultados do desempenho da aplicação de compressão GZIP às 15h00 na plataforma AWS. Assim como observado no horário das 7h00, o gráfico mantém um comportamento estritamente crescente, sem apresentar decréscimos iniciais. Nota-se que, após as primeiras execuções, o desempenho passa a crescer

de forma mais regular, apresentando tendência aproximadamente linear a partir de 10 execuções.

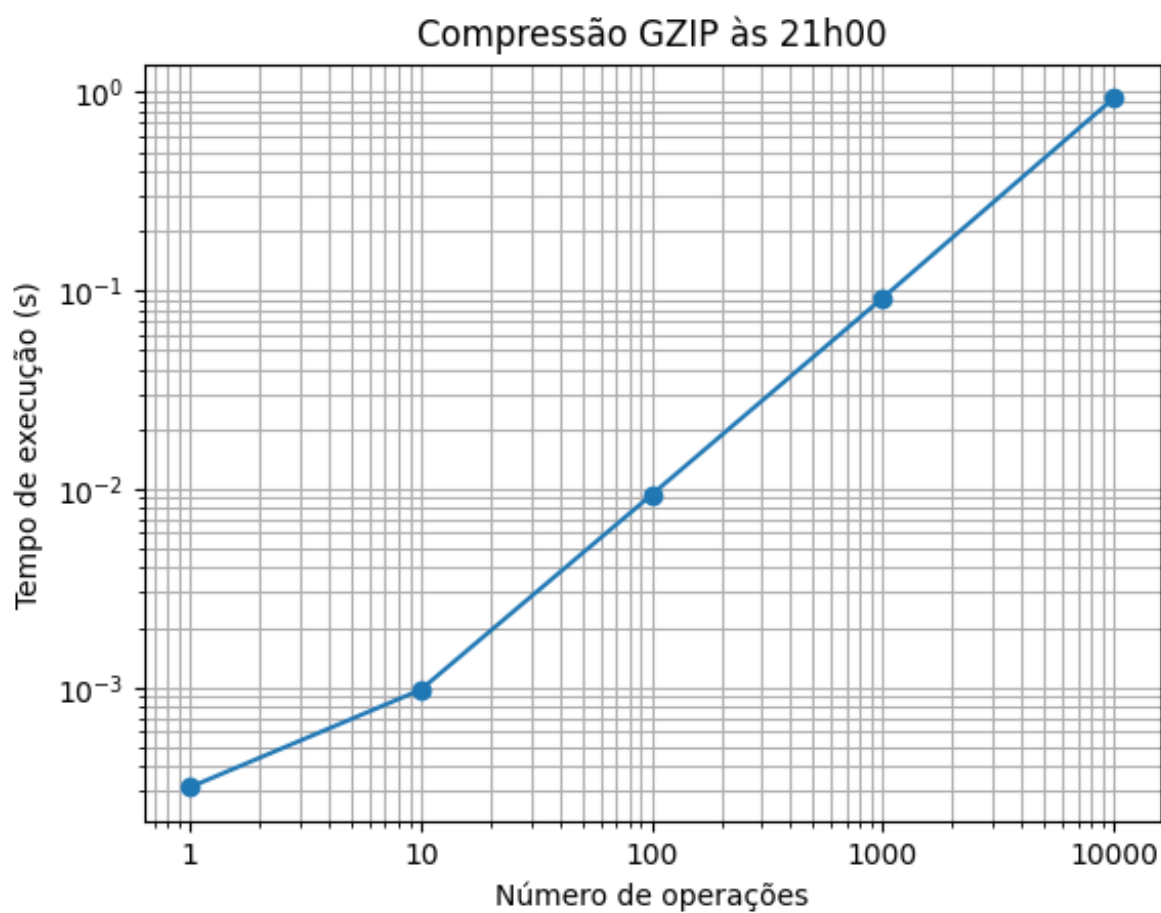
Figura 3.17: Desempenho da aplicação de compressão GZIP às 15h00 na AWS.



Fonte: Elaborado pela autora.

Por fim, a Figura 3.18 apresenta os resultados referentes às execuções realizadas às 21h00 na plataforma AWS. Diferentemente do comportamento observado na plataforma Azure para esse mesmo horário, o gráfico na AWS mantém comportamento estritamente crescente desde a primeira execução. A partir de 10 execuções, observa-se novamente uma tendência de crescimento aproximadamente linear, indicando estabilidade e previsibilidade no desempenho da aplicação. Esse resultado evidencia uma diferença relevante entre as plataformas analisadas, uma vez que, na Azure, o gráfico das 21h00 apresentou comportamento semelhante ao da aplicação de funções hash, com decréscimo inicial seguido de crescimento linear.

Figura 3.18: Desempenho da aplicação de compressão GZIP às 21h00 na AWS.



Fonte: Elaborado pela autora.

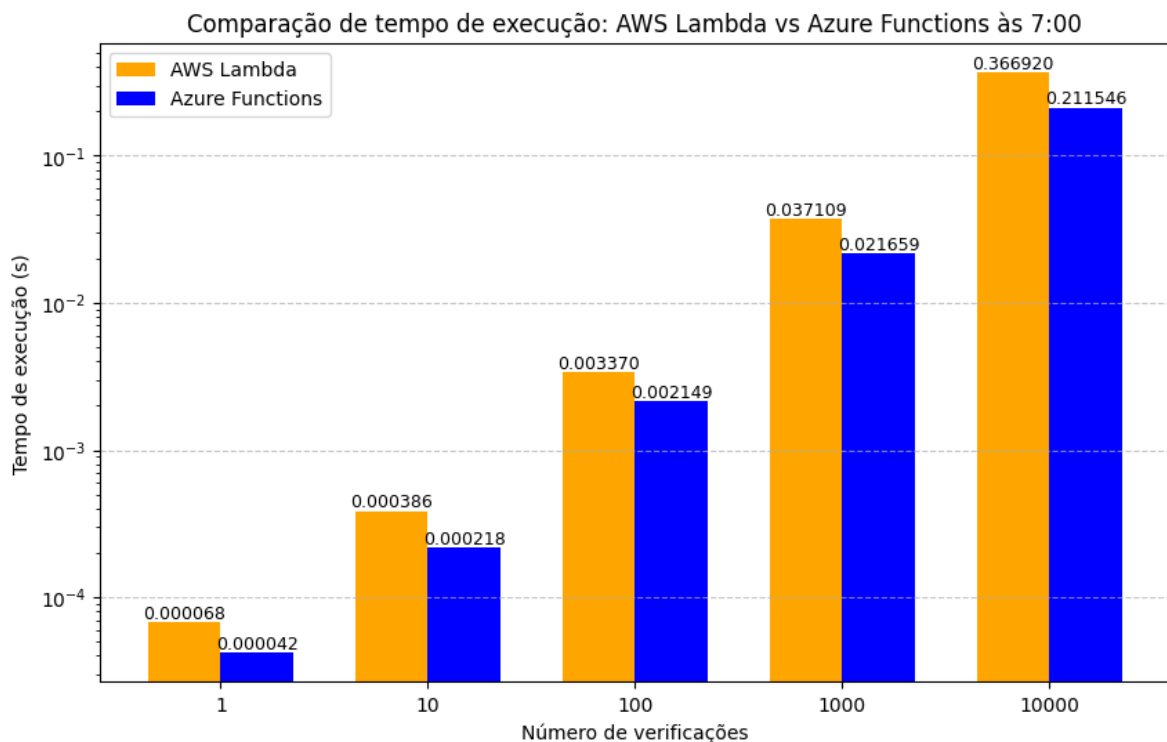
3.3 Resultado dos Testes Comparativos

Considerando os dados apresentados na seção anterior, observa-se que a plataforma *Azure* apresentou, de modo geral, desempenho superior, conforme evidenciado nas análises a seguir.

3.3.1 Assinaturas Digitais

Com base nos resultados obtidos na seção anterior, foi elaborado o gráfico apresentado na figura a seguir, que compara os tempos de execução da aplicação de verificação de assinaturas digitais para 1, 10, 100, 1.000 e 10.000 verificações nas plataformas *Azure* e *AWS* às 7h00.

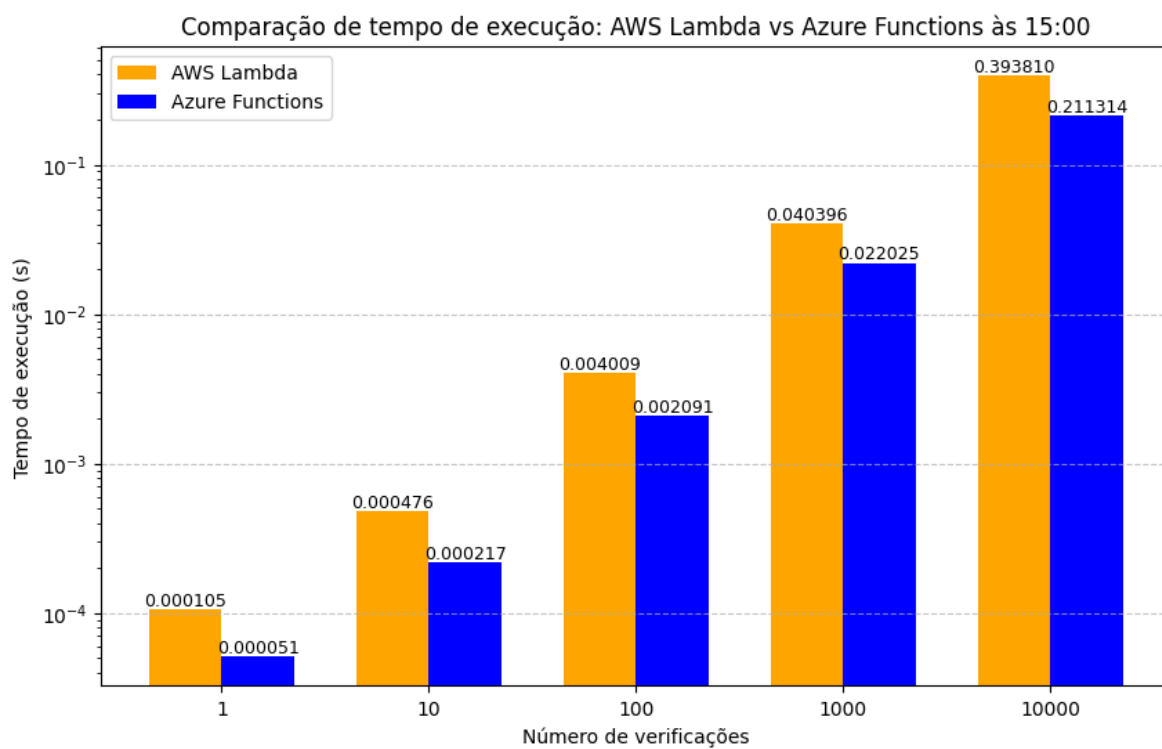
Figura 3.19: Tempos de execução das verificações de assinatura da às 7:00



Fonte: Elaborado pela autora.

Outro gráfico elaborado é apresentado na figura a seguir, o qual compara os tempos de execução da aplicação de verificação de assinaturas digitais para 1, 10, 100, 1.000 e 10.000 verificações nas plataformas *Azure* e *AWS*, realizadas às 15h00.

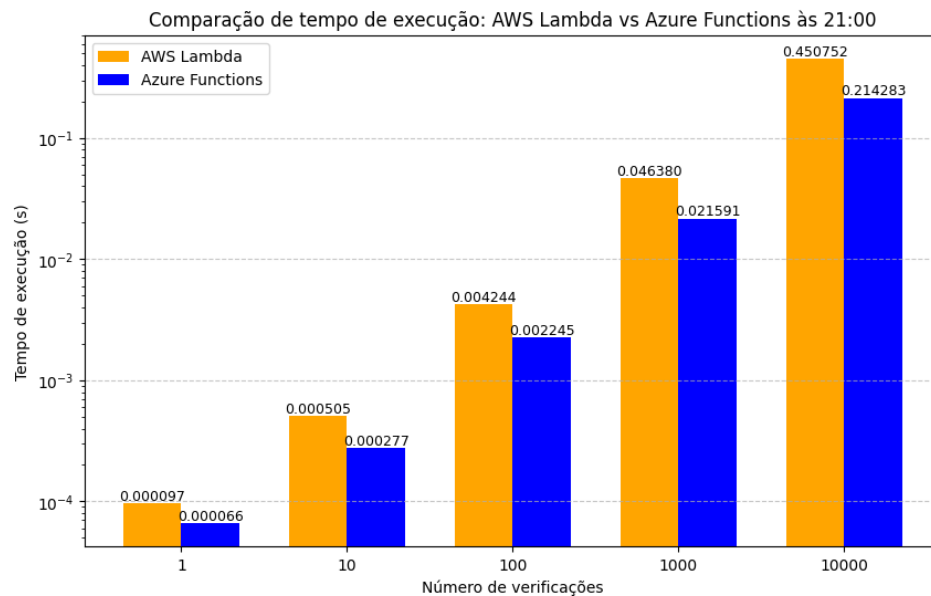
Figura 3.20: Tempos de execução das verificações de assinatura às 15:00



Fonte: Elaborado pela autora.

A figura a seguir apresenta um gráfico comparativo entre as plataformas *Azure* e *AWS*, considerando os tempos de execução da aplicação de verificação de assinaturas digitais para 1, 10, 100, 1.000 e 10.000 operações, realizadas às 21h00.

Figura 3.21: Tempos de execução das verificações de assinatura às 21:00



Fonte: Elaborado pela autora.

Observando-se os gráficos apresentados nas Figuras 3.19, 3.20 e 3.21, constata-se que, em todos os cenários analisados, a plataforma *Azure Functions* apresentou desempenho superior em relação à *AWS Lambda*.

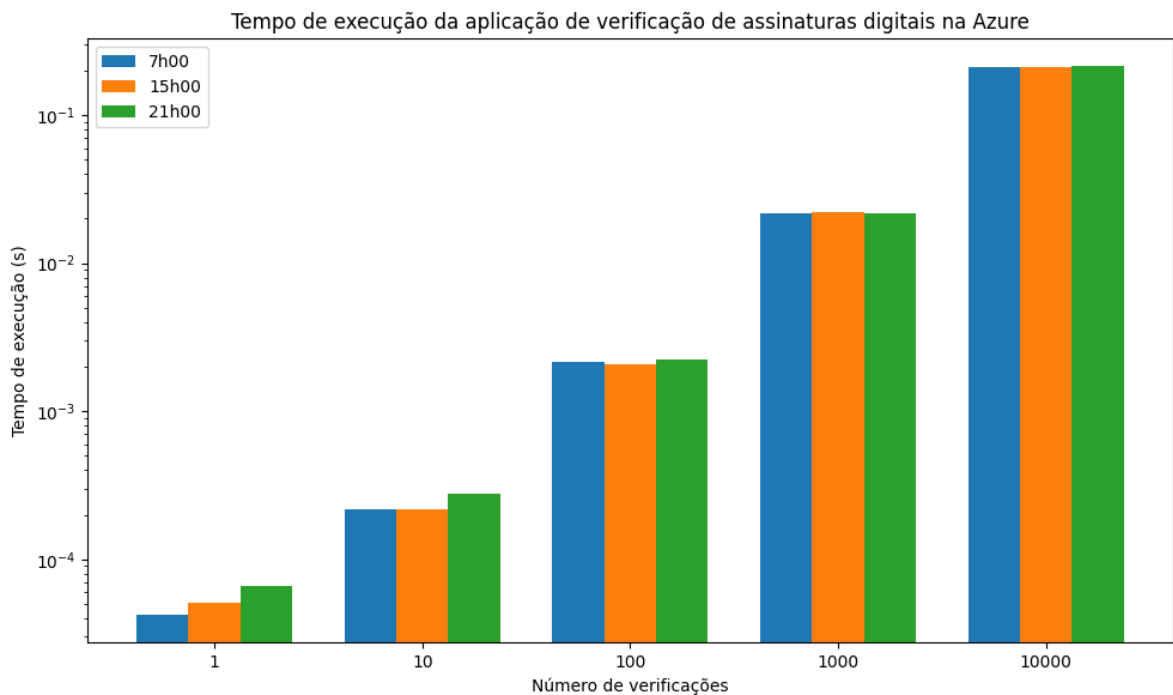
Outro aspecto relevante a ser destacado é que, a partir de 10 verificações, o tempo de execução passou a apresentar um comportamento aproximadamente linear. Em outras palavras, ao aumentar o número de operações em uma ordem de grandeza, observa-se um aumento proporcional no tempo de execução, da mesma ordem, evidenciando a previsibilidade do desempenho das plataformas para cargas maiores.

Além das análises previamente apresentadas, adota-se como abordagem complementar a avaliação dos tempos de execução obtidos em diferentes horários para cada plataforma de computação em nuvem, conforme ilustrado no gráfico da aplicação executada na plataforma *Azure* 3.22 e no gráfico correspondente à plataforma *AWS* 3.23.

Na Figura 3.22, é apresentada a comparação dos tempos de execução da aplicação de ve-

rificação de assinaturas digitais em diferentes horários na plataforma Azure. Observa-se que os tempos registrados às 7h00, 15h00 e 21h00 apresentam comportamentos semelhantes, não sendo identificadas variações significativas entre os horários analisados. As diferenças observadas são pouco expressivas, indicando que o desempenho da aplicação se mantém relativamente estável ao longo do dia, o que sugere que o horário de execução não exerce influência relevante sobre o tempo de processamento nessa plataforma.

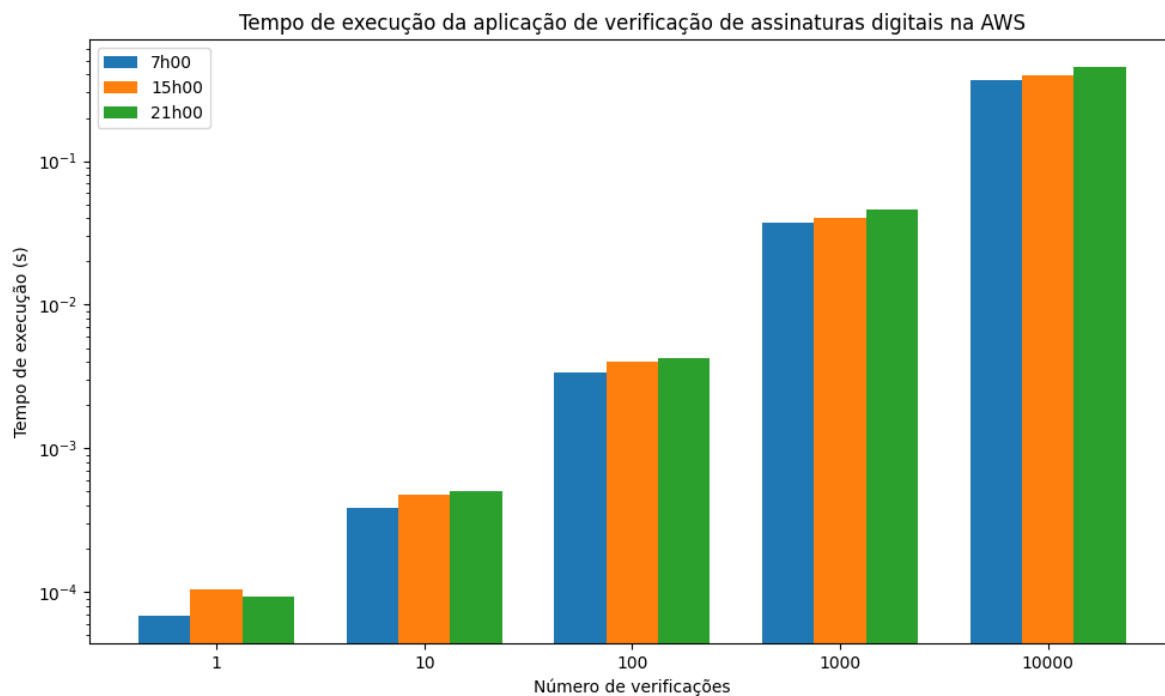
Figura 3.22: Execução da verificação de assinaturas em diferentes horários na *Azure*.



Fonte: Elaborado pela autora.

Na Figura 3.23, é apresentada a comparação dos tempos de execução da aplicação de verificação de assinaturas digitais em diferentes horários na plataforma AWS. De forma semelhante ao observado na plataforma Azure, verifica-se que os tempos de execução nos três horários analisados apresentam comportamentos próximos entre si, com diferenças pouco acentuadas. Esse resultado indica que o desempenho da aplicação permanece consistente ao longo do dia também na AWS, reforçando que a variação temporal não impacta de maneira significativa o tempo de execução da aplicação analisada.

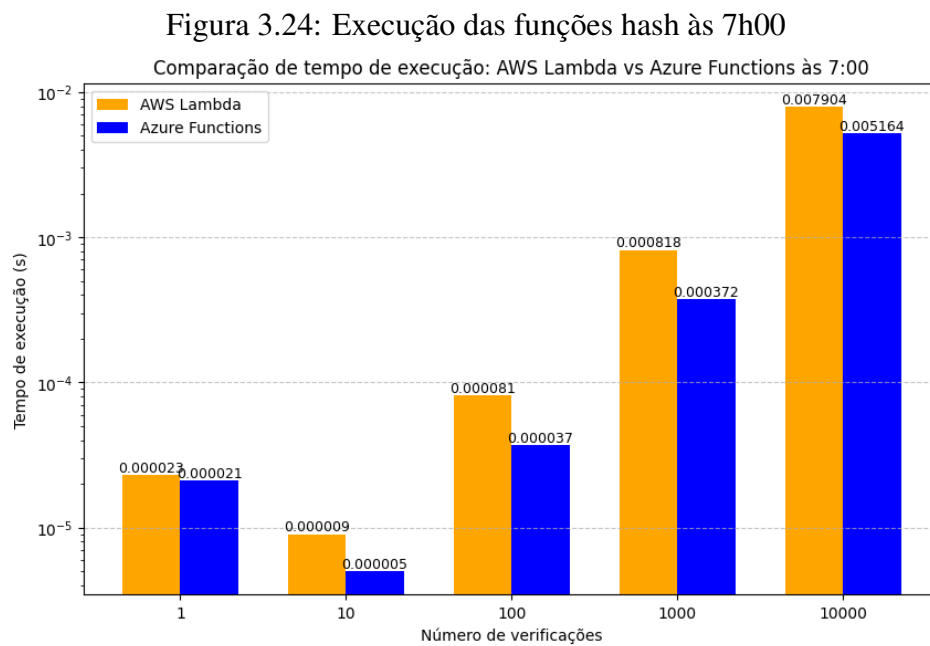
Figura 3.23: Execução da verificação de assinaturas em diferentes horários na AWS.



Fonte: Elaborado pela autora.

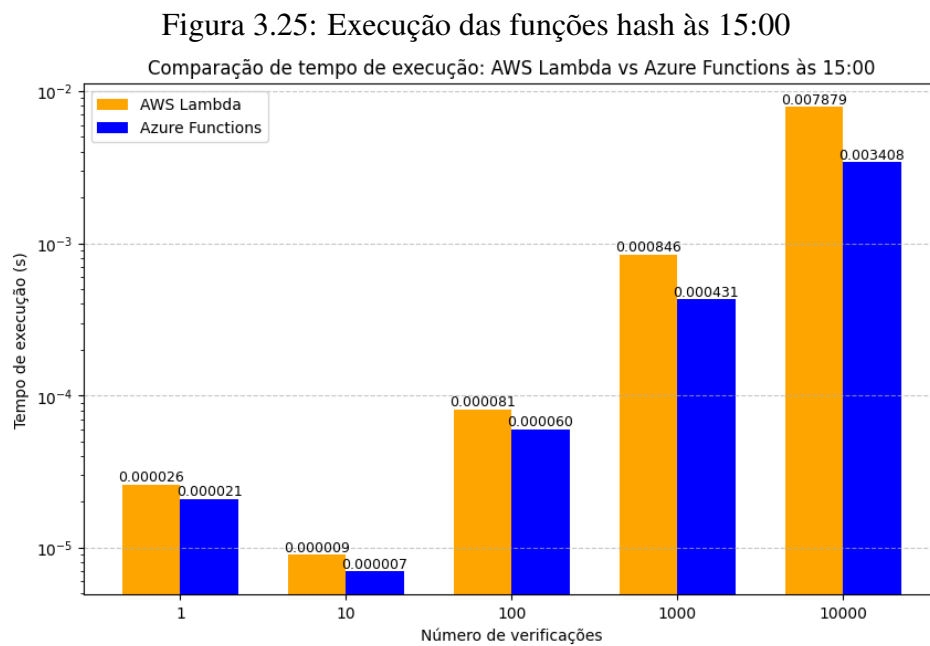
3.3.2 Funções Hash

A figura a seguir apresenta um gráfico comparativo entre as plataformas *Azure* e *AWS*, considerando os tempos de execução da aplicação de funções hash para 1, 10, 100, 1.000 e 10.000 operações realizadas às 7:00.



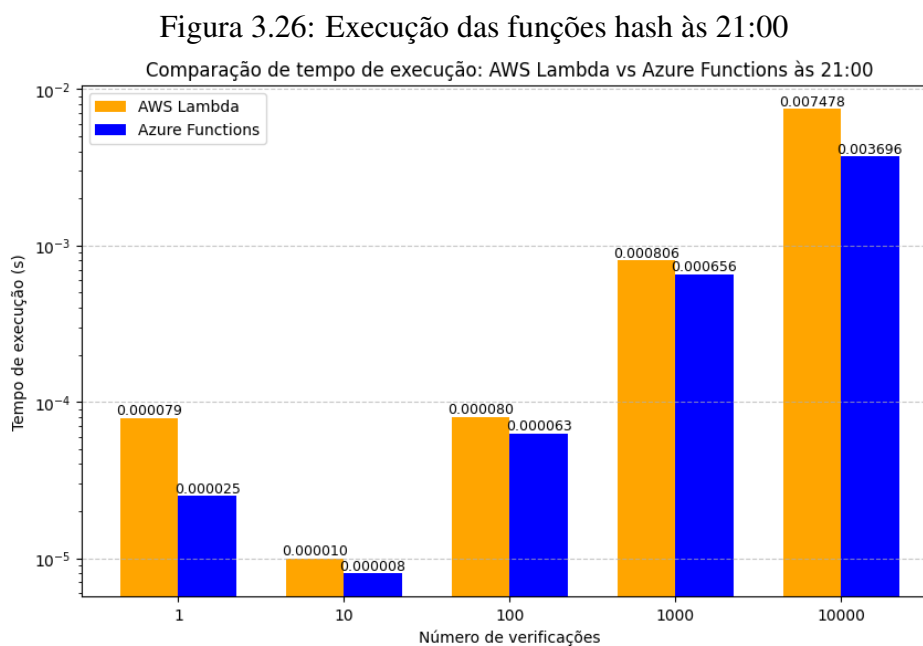
Fonte: Elaborado pela autora.

A figura a seguir apresenta uma análise comparativa do desempenho das plataformas *Azure* e *AWS*, com base nos tempos de execução observados na aplicação de funções hash. A comparação considera diferentes volumes de processamento — 1, 10, 100, 1.000 e 10.000 operações — executados às 15h00.



Fonte: Elaborado pela autora.

A figura apresentada a seguir ilustra a comparação de desempenho entre as plataformas *Azure* e *AWS*, considerando os tempos de execução registrados na aplicação de funções hash. A análise abrange diferentes cargas de processamento, correspondentes a 1, 10, 100, 1.000 e 10.000 operações, realizadas às 21h00.



Fonte: Elaborado pela autora.

Considerando os gráficos apresentados nas Figuras 3.24, 3.25 e 3.26, observa-se que, de maneira semelhante ao comportamento verificado na aplicação de verificação de assinaturas digitais, a plataforma *Azure* apresentou desempenho superior em relação à *AWS*.

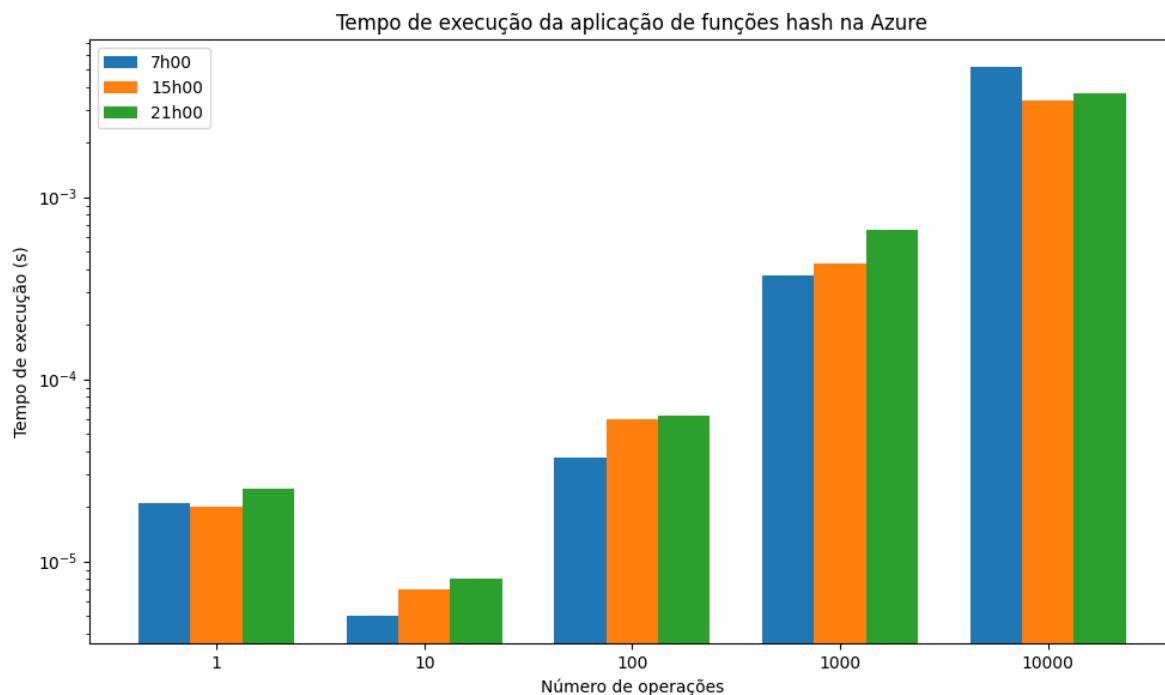
Uma diferença relevante desta aplicação em comparação às demais reside no comportamento dos tempos de execução. Enquanto nas outras aplicações os gráficos apresentaram crescimento estritamente crescente, os resultados referentes à aplicação das funções hash demonstraram um comportamento decrescente entre 1 e 10 operações. A partir desse ponto, observa-se um crescimento aproximadamente linear do tempo de execução.

Além das análises anteriormente apresentadas, adotou-se, como abordagem complementar, a avaliação dos tempos de execução obtidos em diferentes horários para cada plataforma de computação em nuvem. Essa análise é ilustrada pelo gráfico da aplicação executada na plataforma *Azure* Figura 3.27 e pelo gráfico correspondente à plataforma *AWS* 3.28.

Na Figura 3.27, é apresentada a comparação dos tempos de execução da aplicação de funções hash em diferentes horários na plataforma *Azure*. Observa-se que os tempos registrados às 7h00, 15h00 e 21h00 apresentam comportamentos semelhantes, embora sejam identificadas variações mais perceptíveis quando comparadas às observadas na aplicação de verificação

de assinaturas digitais. Ainda assim, tais diferenças não comprometem a uniformidade geral dos resultados.

Figura 3.27: Execução de funções hash em diferentes horários na *Azure*.



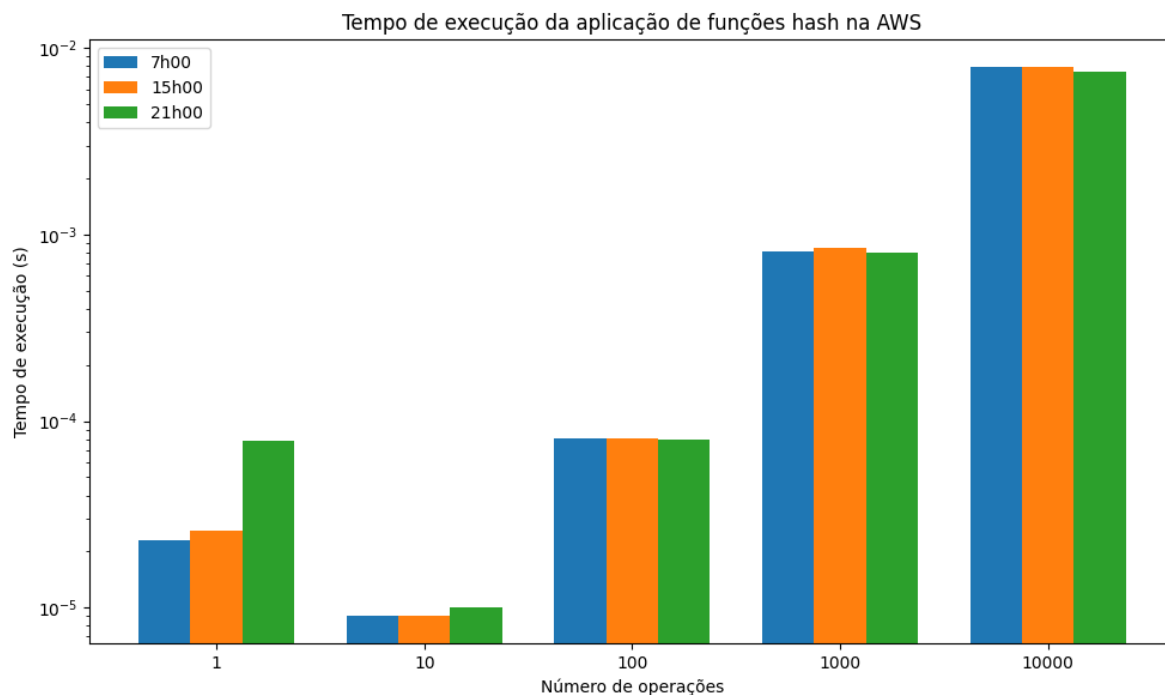
Fonte: Elaborado pela autora.

Na Figura 3.28, é apresentada a comparação dos tempos de execução da aplicação de funções hash em diferentes horários na plataforma AWS. De forma análoga ao observado na Azure, verifica-se que os tempos obtidos nos três horários analisados apresentam comportamentos próximos entre si, embora com variações mais acentuadas em relação às aplicações de assinaturas digitais. Esse resultado indica que, mesmo com maior dispersão dos valores, a aplicação mantém desempenho relativamente consistente ao longo do dia.

3.3.3 Compressão GZIP

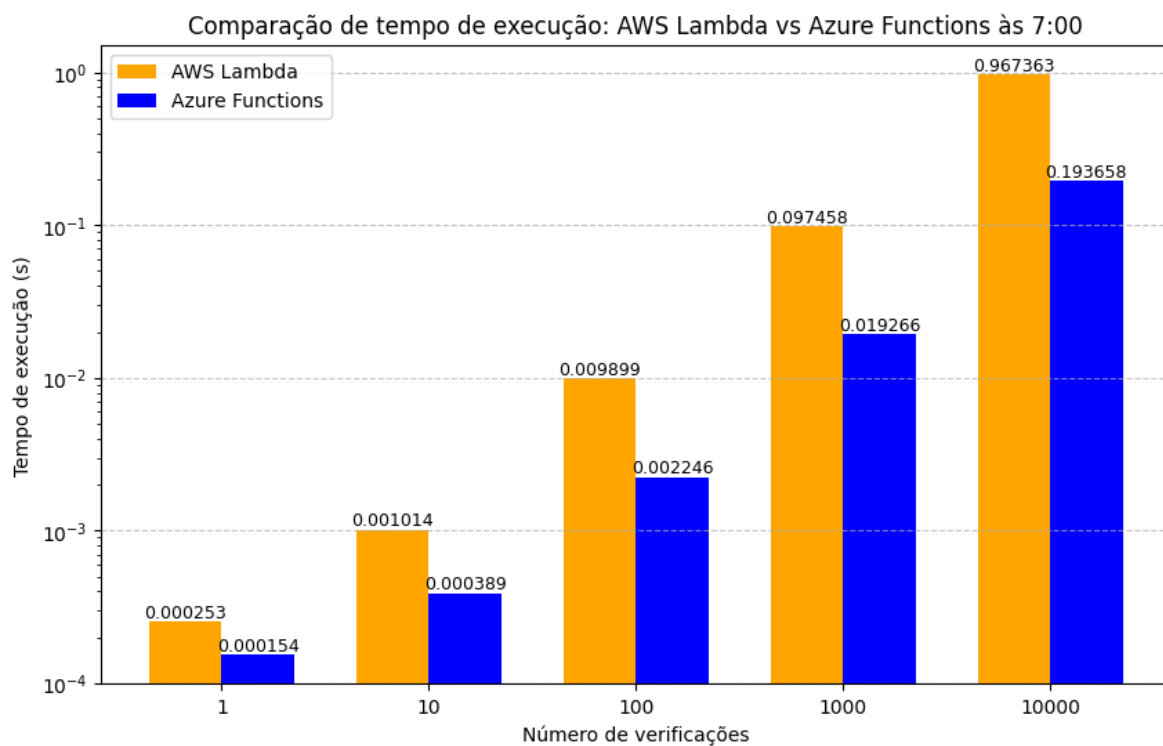
A figura a seguir apresenta uma comparação do desempenho das plataformas *Azure* e *AWS* a partir dos tempos de execução obtidos na aplicação de compressão GZIP. O gráfico considera diferentes níveis de carga, correspondentes a 1, 10, 100, 1.000 e 10.000 operações, executadas às 7h00.

Figura 3.28: Execução de funções hash em diferentes horários na AWS.



Fonte: Elaborado pela autora.

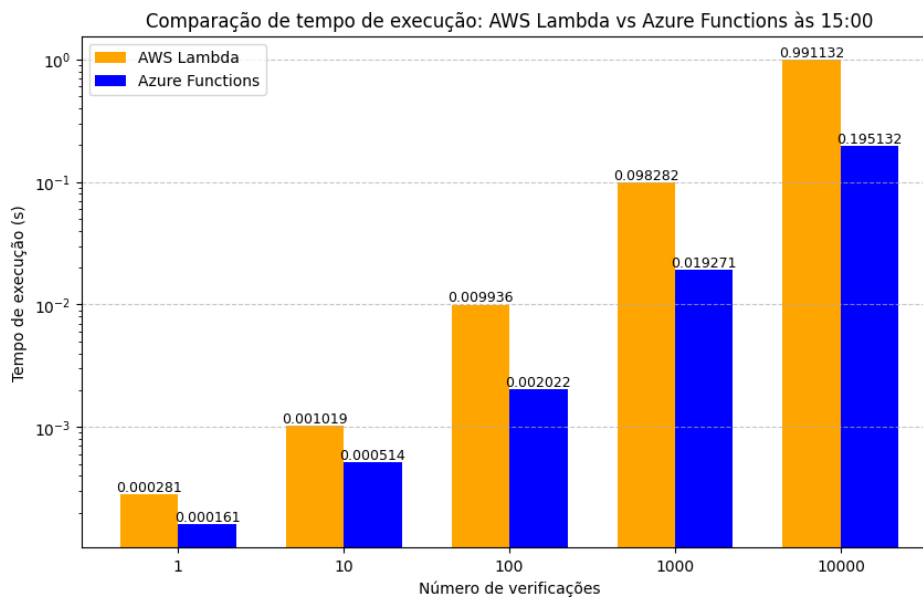
Figura 3.29: Tempos de execução das compressões GZIP às 7:00



Fonte: Elaborado pela autora.

A figura apresentada a seguir evidencia a comparação de desempenho entre as plataformas *Azure* e *AWS*, considerando os tempos de execução da aplicação de compressão GZIP. A análise contempla diferentes volumes de processamento, variando de 1 a 10.000 operações, referentes aos experimentos realizados às 15h00.

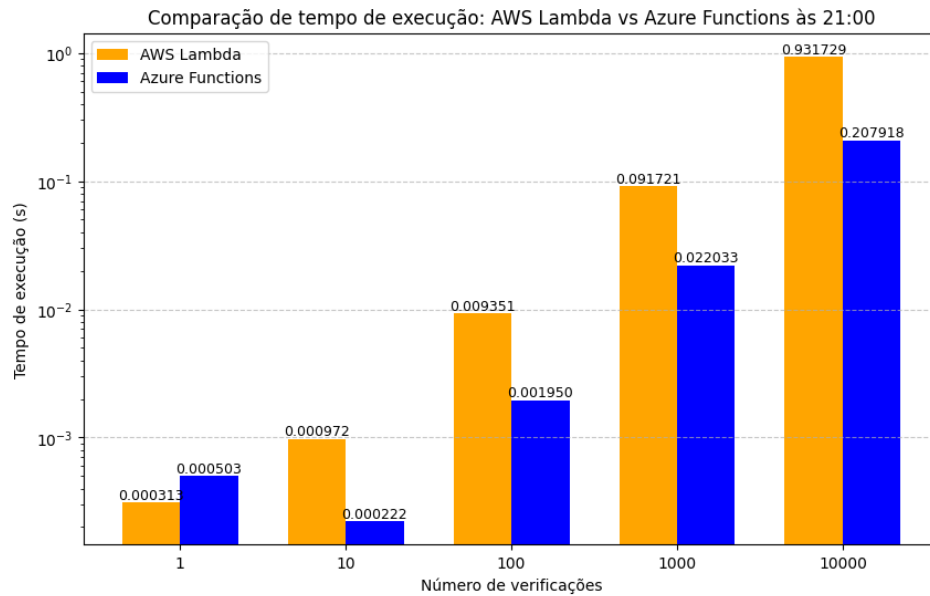
Figura 3.30: Tempos de execução das compressões GZIP às 15:00



Fonte: Elaborado pela autora.

A figura a seguir demonstra o comportamento comparativo das plataformas *Azure* e *AWS* quanto ao desempenho da aplicação de compressão GZIP, com base nos tempos de execução registrados. O gráfico abrange diferentes níveis de carga computacional, de 1 a 10.000 operações, relativos aos testes conduzidos às 21h00.

Figura 3.31: Tempos de execução das compressões GZIP às 21:00



Fonte: Elaborado pela autora.

Os gráficos nas Figuras 3.29, 3.30 e 3.31 evidenciam a mesma tendência observada nas demais aplicações analisadas, na qual a plataforma Azure apresentou, de modo geral, melhor desempenho em comparação à AWS.

Entretanto, observa-se uma exceção na execução de uma compressão realizada às 21h00, na qual a Azure apresentou tempo de execução superior ao da AWS. Esse comportamento torna esse gráfico o único caso não estritamente crescente que não corresponde à aplicação de funções hash.

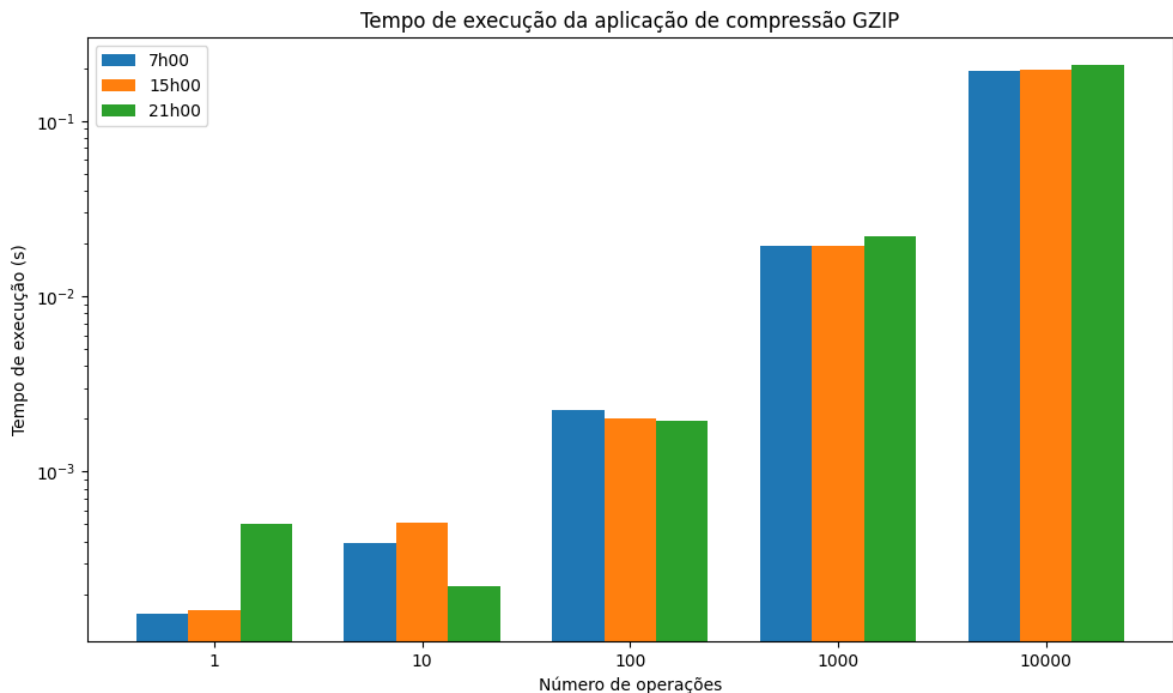
Ainda assim, assim como nos demais cenários avaliados, os gráficos também demonstram uma tendência de crescimento aproximadamente linear a partir de 10 operações, indicando comportamento estável do desempenho para volumes maiores de processamento.

A avaliação dos tempos de execução em diferentes horários foi utilizada como abordagem complementar às análises previamente realizadas para cada plataforma de computação em nuvem. Os resultados obtidos são apresentados por meio do gráfico da aplicação executada na plataforma *Azure* 3.32 e do gráfico correspondente à plataforma *AWS* 3.33.

Na Figura 3.32, é apresentada a comparação dos tempos de execução da aplicação de compressão GZIP em diferentes horários na plataforma *Azure*. Observa-se que os tempos

obtidos às 7h00 e 15h00 apresentam comportamento crescente ao longo das execuções, indicando aumento progressivo do tempo de processamento. Em contraste, o horário das 21h00 apresenta um comportamento distinto, caracterizado por um decréscimo inicial seguido de crescimento nas execuções subsequentes.

Figura 3.32: Execução de funções hash em diferentes horários na *Azure*.



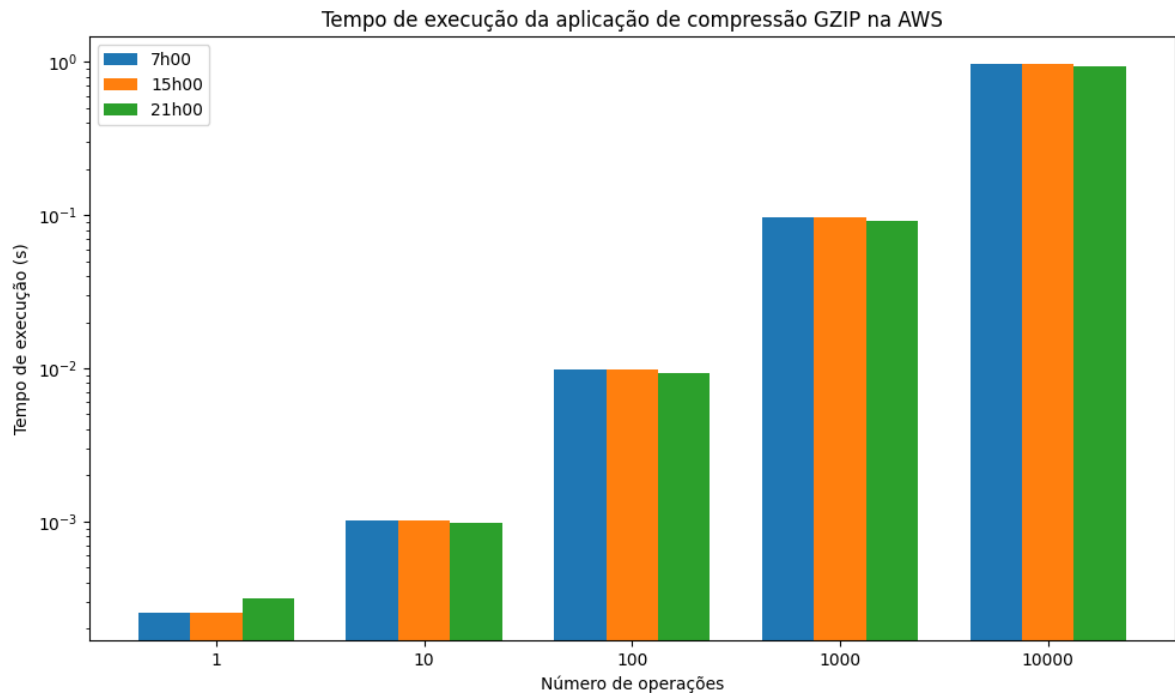
Fonte: Elaborado pela autora.

Na Figura 3.33, é apresentada a comparação dos tempos de execução da aplicação de compressão GZIP em diferentes horários na plataforma AWS. Verifica-se que os resultados obtidos nos três horários analisados apresentam comportamentos semelhantes entre si, sem variações expressivas. As diferenças observadas entre os valores são pouco significativas, indicando que o desempenho da aplicação mantém relativa estabilidade ao longo do dia nessa plataforma. Esse resultado contrasta com o comportamento observado na Azure, evidenciando maior uniformidade temporal da aplicação de compressão GZIP na AWS.

3.3.4 Visão geral

A partir do cálculo da média e do desvio padrão das velocidades de execução de cada aplicação, foram obtidas as Tabelas 3.1 e 3.2, respectivamente.

Figura 3.33: Execução de funções hash em diferentes horários na AWS.



Fonte: Elaborado pela autora.

Observando-se essas tabelas, verifica-se que a *Azure Functions* apresentou, em média, maior desempenho que a *AWS Lambda* em todos os casos analisados e que esta, por sua vez, apresentou menor desvio padrão nas velocidades de execução.

A tabela a seguir compara a média das velocidades de execução, em operações por segundo, da *AWS Lambda* e da *Azure Functions* para cada uma das aplicações analisadas: verificação de assinaturas digitais, funções de hash e compressões GZIP.

Tabela 3.1: Comparação das médias de velocidade, em operações por segundo, obtidas nas plataformas *Azure Functions* e *AWS Lambda*

Operação	AWS – Média	Azure – Média
Assinaturas Digitais (RSA)	21830.748905	40337.117199
Hash	965625.965419	1660894.002396
GZIP	8990.499948	36690.317213

Fonte: Elaborado pela autora.

A tabela a seguir apresenta uma comparação do desvio padrão das velocidades de execução, expressas em operações por segundo, entre as plataformas *Azure Functions* e *AWS*

Lambda, considerando as aplicações de verificação de assinaturas digitais, funções de hash e compressão GZIP.

Tabela 3.2: Comparação dos desvios padrão da velocidade, em operações por segundo, obtidos nas plataformas *Azure Functions* e *AWS Lambda*

Operação	AWS – Desvio Padrão	Azure – Desvio Padrão
Assinaturas Digitais	5813.857723	10817.772467
Hash	474665.140500	948506.300273
GZIP	2663.541642	18388.347199

Fonte: Elaborado pela autora.

A partir desses cálculos, obtiveram-se as Tabelas 3.3 e 3.4, a partir das quais é possível observar que a aplicação de verificação de assinaturas digitais apresentou melhor desempenho às 7h00 em ambas as plataformas. A aplicação de funções hash, por sua vez, obteve melhor desempenho às 21h00 na *AWS Lambda* e às 7h00 na *Azure Functions*. Já na aplicação de compressão GZIP, ambas as plataformas apresentaram melhor desempenho às 21h00.

A tabela a seguir apresenta a média dos valores de velocidade de execução, em operações por segundo, de cada aplicação — verificação de assinaturas digitais, funções de hash e compressões GZIP — em três horários distintos (7h00, 15h00 e 21h00) na plataforma *AWS*.

Tabela 3.3: Velocidade média, em operações por segundo, da *AWS Lambda* para os algoritmos RSA, Hash e GZIP em diferentes horários

Aplicação	7:00 (ops/s)	15:00 (ops/s)	21:00 (ops/s)
Assinaturas Digitais	24885.874311	21130.021145	19476.351258
Hash	963621.735375	963973.726342	969282.434540
GZIP	8904.728641	8904.728641	9162.042564

Fonte: Elaborado pela autora.

A tabela a seguir reúne as médias das velocidades de execução, expressas em operações por segundo, das aplicações de verificação de assinaturas digitais, funções de hash e compressão GZIP, considerando os três horários de execução (7h00, 15h00 e 21h00) na plataforma *Azure*.

Tabela 3.4: Velocidade média, em operações por segundo, da Azure Functions para os algoritmos RSA, Hash e GZIP em diferentes horários

Aplicação	7:00 (ops/s)	15:00 (ops/s)	21:00 (ops/s)
Assinaturas Digitais	42001.330605	41244.226730	37765.794261
Hash	1875898.886669	1672322.549183	1434460.571335
GZIP	36058.744552	35650.361283	38361.845803

Fonte: Elaborado pela autora.

Capítulo 4

Conclusão

Este trabalho apresentou uma análise comparativa de desempenho entre as plataformas de computação serverless *AWS Lambda* e *Azure Functions*, considerando três aplicações representativas: verificação de assinaturas digitais por meio do algoritmo RSA, cálculo de funções de hash e compressão de dados utilizando GZIP. Os experimentos foram realizados variando-se a carga de trabalho — 1, 10, 100, 1.000 e 10.000 operações — e os horários de execução — 7h00, 15h00 e 21h00. Ambas as plataformas foram configuradas para utilizar uma CPU virtual, sendo os testes executados na região de São Paulo, de modo a garantir maior uniformidade nas condições experimentais.

O estudo utilizou exclusivamente recursos gratuitos disponíveis para estudantes. Nesse contexto, apenas a *Azure Functions* foi acessada por meio de uma conta de estudante, sem necessidade de cadastro de cartão de crédito. A *AWS Lambda* foi utilizada por meio de uma conta gratuita padrão, não vinculada a programas educacionais. As plataformas foram escolhidas por se tratarem de provedores de grande porte e por disponibilizarem planos acessíveis a usuários acadêmicos e iniciantes, permitindo a realização de experimentos de forma econômica e segura.

A Tabela 4.1 apresenta uma comparação geral entre as plataformas *Azure Functions* e *AWS Lambda*, considerando a média e o desvio padrão das velocidades de execução, em operações por segundo, de três aplicações distintas: verificação de assinaturas digitais, funções de hash e compressão GZIP. Os valores correspondem à execução de cargas de trabalho variáveis — 1, 10, 100, 1.000 e 10.000 operações — em três horários distintos do dia (7h00, 15h00 e

21h00).

Tabela 4.1: Comparação geral de desempenho e características das plataformas *Azure Functions* e *AWS Lambda*

Critério	Azure	AWS
Média da velocidade da aplicação de verificação de assinaturas	40337.117199	21830.748905
Média da velocidade da aplicação de funções de hash	1660894.002396	965625.965419
Média da velocidade da aplicação de compressões GZIP	36690.317213	8990.499948
Desvio padrão da velocidade da aplicação de verificação de assinaturas	10817.772467	5813.857723
Desvio padrão da velocidade da aplicação de funções de hash	948506.300273	474665.140500
Desvio padrão da velocidade da aplicação de compressões GZIP	18388.347199	2663.541642

Fonte: Elaborado pela autora.

Além dos indicadores quantitativos de desempenho, realizou-se também uma análise qualitativa referente à facilidade de uso de cada plataforma, aspecto relevante no contexto acadêmico e no desenvolvimento de aplicações *serverless* por estudantes e pesquisadores. Com o objetivo de apresentar essa avaliação, elaborou-se a Tabela 4.2, na qual é apresentada a comparação entre as duas plataformas analisadas.

A Tabela 4.2 apresenta uma comparação entre as plataformas *AWS Lambda* e *Azure Functions* sob a perspectiva de usabilidade e processo de desenvolvimento. Observa-se que a *AWS Lambda* disponibiliza um ambiente de codificação diretamente em seu portal web, o que possibilita maior agilidade tanto no desenvolvimento quanto no processo de *deploy*, além de facilitar a utilização de bibliotecas externas por meio do recurso de *Layers*. Em contrapartida, a *Azure Functions* exige que a codificação e a instalação de bibliotecas sejam realizadas localmente, demandando o posterior envio dos arquivos da aplicação para a plataforma, o que pode resultar em maior tempo de implantação. Dessa forma, as diferenças evidenciam abordagens distintas quanto à flexibilidade e praticidade no desenvolvimento e na implantação de aplicações *serverless*.

Tabela 4.2: Comparação das características de uso das plataformas

Critério	AWS Lambda	Azure Functions
Ambiente de codificação	Disponibiliza interface para codificação diretamente no portal web da plataforma	A codificação é realizada localmente, em arquivos no ambiente do desenvolvedor
Uso de bibliotecas	Permite a utilização de bibliotecas externas por meio do uso de camadas (<i>Layers</i>)	As bibliotecas devem ser instaladas diretamente no projeto local antes do envio para a plataforma
Processo de <i>deploy</i>	O <i>deploy</i> da aplicação pode ser realizado diretamente pelo portal web	O <i>deploy</i> exige o envio dos arquivos da aplicação para a plataforma
Tempo de implantação	Implantação geralmente mais rápida, por ser realizada diretamente no ambiente web	Pode apresentar atraso devido ao processo de envio dos arquivos para <i>deploy</i>

Fonte: Elaborado pela autora.

A análise dos dados apresentados evidencia que a *Azure Functions* obteve, em todas as aplicações avaliadas, médias de velocidade de execução superiores às observadas na *AWS Lambda*, indicando maior eficiência no processamento das cargas de trabalho consideradas. Em contrapartida, a *AWS Lambda* apresentou menores valores de desvio padrão nas velocidades de execução, o que aponta para uma maior consistência e previsibilidade do desempenho ao lidar com diferentes quantidades de operações. No que se refere à facilidade de uso, observa-se que a AWS disponibiliza uma interface web que permite o desenvolvimento e a edição do código diretamente no navegador, simplificando o processo de configuração e reduzindo a necessidade de manipulação local de arquivos. A *Azure Functions*, por sua vez, privilegia o desenvolvimento local, exigindo a criação, manipulação e posterior envio dos arquivos para a nuvem, sendo assim mais complexa, embora eficiente em termos de desempenho.

De forma geral, os resultados obtidos demonstraram que a *Azure Functions* apresentou, em média, melhor desempenho que a *AWS Lambda* na maioria dos cenários analisados, tanto em termos de tempo de execução quanto de velocidade de processamento. A única exceção observada ocorreu na aplicação de compressão GZIP, ao executar uma única operação às 21h00, na qual a *AWS Lambda* apresentou desempenho superior. No entanto, esse compor-

tamento não se manteve à medida que o número de operações aumentou, reforçando a superioridade média da Azure Functions nos demais testes. Observou-se ainda que, a partir de 10 execuções, os tempos de execução passaram a apresentar um comportamento relativamente linear em ambas as plataformas, indicando uma estabilização dos custos de inicialização e do ambiente de execução. Além disso, as aplicações de funções de hash e de compressão GZIP às 21h00 foram as únicas a não apresentarem crescimento estritamente crescente do tempo de execução, exibindo um decréscimo entre 1 e 10 operações. Esse comportamento pode estar associado a fatores como otimizações internas das plataformas, mecanismos de cache ou variações momentâneas na alocação de recursos computacionais.

A análise considerando os horários de execução evidenciou padrões distintos entre as aplicações. A verificação de assinaturas digitais apresentou melhor desempenho às 7h00 em ambas as plataformas. A aplicação de funções de hash obteve melhor desempenho às 21h00 na AWS Lambda e às 7h00 na Azure Functions. Já a aplicação de compressão GZIP apresentou melhor desempenho às 21h00 em ambas as plataformas, sugerindo que a carga global dos serviços ao longo do dia pode influenciar significativamente o desempenho das aplicações serverless.

Diante dos resultados apresentados, conclui-se que a escolha entre AWS Lambda e Azure Functions deve levar em consideração não apenas o desempenho médio, mas também a variabilidade, o tipo de aplicação, o volume de operações, o horário de execução e a facilidade de uso. Embora a Azure Functions tenha se destacado em termos de desempenho geral, a AWS Lambda demonstrou maior estabilidade em determinados cenários e maior simplicidade na configuração e execução do código.

Como trabalhos futuros, considera-se viável a ampliação deste estudo por meio da inclusão de um número maior de plataformas de computação em nuvem, como Google Cloud Platform e Oracle Cloud Infrastructure, ou da exploração de diferentes tipos de contas e configurações de recursos dentro de um mesmo provedor. Outra possibilidade consiste na realização de experimentos com um volume significativamente maior de operações ou aplicações que demandem maior poder de CPU, permitindo avaliar o comportamento das plataformas sob cargas mais intensas e próximas de cenários reais de produção. Além disso, uma análise

mais detalhada da experiência de uso e da integração com ferramentas de desenvolvimento poderia fornecer subsídios adicionais para a escolha da plataforma mais adequada ao contexto acadêmico ou profissional.

Apêndice A

APÊNDICE A – Resultados detalhados dos experimentos na plataforma Azure Functions

Este apêndice apresenta os resultados completos dos experimentos de desempenho das aplicações de verificação de assinaturas digitais, funções hash e compressão GZIP executadas na plataforma *Azure Functions*, considerando diferentes cargas de processamento e horários de execução.

Tabela A.1: Tempo de execução da aplicação de verificação de assinaturas digitais às 7h00

Número de verificações	Tempo de execução (s)
1	0,000042
10	0,000218
100	0,002149
1.000	0,021659
10.000	0,211546

Fonte: Elaborado pela autora.

Tabela A.2: Tempo de execução da aplicação de verificação de assinaturas digitais às 15h00

Número de verificações	Tempo de execução (s)
1	0,000051
10	0,000217
100	0,002091
1.000	0,022025
10.000	0,211314

Fonte: Elaborado pela autora.

Tabela A.3: Tempo de execução da aplicação de verificação de assinaturas digitais às 21h00

Número de verificações	Tempo de execução (s)
1	0,000066
10	0,000277
100	0,002245
1.000	0,021591
10.000	0,214283

Fonte: Elaborado pela autora.

Tabela A.4: Tempo de execução da aplicação de funções hash às 7h00

Número de operações	Tempo de execução (s)
1	0,000021
10	0,000005
100	0,000037
1.000	0,000372
10.000	0,005164

Fonte: Elaborado pela autora.

Tabela A.5: Tempo de execução da aplicação de funções hash às 15h00

Número de operações	Tempo de execução (s)
1	0,000020
10	0,000007
100	0,000060
1.000	0,000431
10.000	0,003408

Fonte: Elaborado pela autora.

Tabela A.6: Tempo de execução da aplicação de funções hash às 21h00

Número de operações	Tempo de execução (s)
1	0,000025
10	0,000008
100	0,000063
1.000	0,000656
10.000	0,003696

Fonte: Elaborado pela autora.

Tabela A.7: Tempo de execução da aplicação de compressão GZIP às 7h00

Número de operações	Tempo de execução (s)
1	0,000154
10	0,000389
100	0,002246
1.000	0,019266
10.000	0,193658

Fonte: Elaborado pela autora.

Tabela A.8: Tempo de execução da aplicação de compressão GZIP às 15h00

Número de operações	Tempo de execução (s)
1	0,000161
10	0,000514
100	0,002022
1.000	0,019271
10.000	0,195133

Fonte: Elaborado pela autora.

Tabela A.9: Tempo de execução da aplicação de compressão GZIP às 21h00

Número de operações	Tempo de execução (s)
1	0,000503
10	0,000222
100	0,001950
1.000	0,022033
10.000	0,207918

Fonte: Elaborado pela autora.

Apêndice B

APÊNDICE B – Resultados detalhados dos experimentos na plataforma AWS Lambda

Este apêndice apresenta os resultados completos dos experimentos de desempenho das aplicações de verificação de assinaturas digitais, funções hash e compressão GZIP executadas na plataforma *AWS Lambda*, considerando diferentes cargas de processamento e horários de execução.

Tabela B.1: Tempo de execução da aplicação de verificação de assinaturas digitais às 7h00

Número de verificações	Tempo de execução (s)
1	0,000068
10	0,000387
100	0,003370
1.000	0,037109
10.000	0,366920

Fonte: Elaborado pela autora.

Tabela B.2: Tempo de execução da aplicação de verificação de assinaturas digitais às 15h00

Número de verificações	Tempo de execução (s)
1	0,000105
10	0,000476
100	0,004009
1.000	0,040396
10.000	0,393810

Fonte: Elaborado pela autora.

Tabela B.3: Tempo de execução da aplicação de verificação de assinaturas digitais às 21h00

Número de verificações	Tempo de execução (s)
1	0,000093
10	0,000505
100	0,004244
1.000	0,046380
10.000	0,450752

Fonte: Elaborado pela autora.

Tabela B.4: Tempo de execução da aplicação de funções hash às 7h00

Número de operações	Tempo de execução (s)
1	0,000023
10	0,000009
100	0,000081
1.000	0,000818
10.000	0,007904

Fonte: Elaborado pela autora.

Tabela B.5: Tempo de execução da aplicação de funções hash às 15h00

Número de operações	Tempo de execução (s)
1	0,000026
10	0,000009
100	0,000081
1.000	0,000846
10.000	0,007879

Fonte: Elaborado pela autora.

Tabela B.6: Tempo de execução da aplicação de funções hash às 21h00

Número de operações	Tempo de execução (s)
1	0,000079
10	0,000010
100	0,000080
1.000	0,000806
10.000	0,007478

Fonte: Elaborado pela autora.

Tabela B.7: Tempo de execução da aplicação de compressão GZIP às 7h00

Número de operações	Tempo de execução (s)
1	0,000252
10	0,001014
100	0,009899
1.000	0,097458
10.000	0,967363

Fonte: Elaborado pela autora.

Tabela B.8: Tempo de execução da aplicação de compressão GZIP às 15h00

Número de operações	Tempo de execução (s)
1	0,000253
10	0,001014
100	0,009897
1.000	0,097458
10.000	0,967363

Fonte: Elaborado pela autora.

Tabela B.9: Tempo de execução da aplicação de compressão GZIP às 21h00

Número de operações	Tempo de execução (s)
1	0,000313
10	0,000972
100	0,009351
1.000	0,091720
10.000	0,931729

Fonte: Elaborado pela autora.

Referências Bibliográficas

ALKHATIB, A.; SHAHEEN, A.; ALBUSTANJI, R. N. A comparative analysis of cloud computing services: AWS, Azure and GCP. *Genesis*, v. 4, 2024.

Amazon Web Services, Inc. *AWS Educate*. [S.l.], 2025. Disponível em: <<https://aws.amazon.com/pt/education/awseducate/>>. Acesso em: 13 dez. 2025. Disponível em: <<https://aws.amazon.com/pt/education/awseducate/>>.

Amazon Web Services, Inc. *AWS Lambda*. [S.l.], 2025. Disponível em: <<https://aws.amazon.com/pt/pm/lambda/>>. Acesso em: 13 dez. 2025. Disponível em: <<https://aws.amazon.com/pt/pm/lambda/>>.

AMBROZIO, T. d. A.; VICENTE, J. G. Análise comparativa de escalabilidade e disponibilidade de banco de dados em ambientes de nuvem: um estudo dos serviços oferecidos por aws e azure. *Repositório Institucional do Centro Paula Souza*, dez. 2024. Artigo científico. Disponível em: <<https://ric.cps.sp.gov.br/handle/123456789/30351>>.

ANDRADE, R. S.; SILVA, F. d. S. Algoritmo de criptografia rsa: análise entre a segurança e velocidade. *Eventos Pedagógicos*, v. 3, n. 3, p. 438–457, 2012. Disponível em: <<https://periodicos.unemat.br/index.php/rep/article/view/9329>>. Acesso em: 12 dez. 2025. Disponível em: <<https://periodicos.unemat.br/index.php/rep/article/view/9329>>.

CALIXTO, C. C. *CLOUD COMPARE: Uma Ferramenta de Comparação de Custos de Implantação de Software em Nuvens do Tipo Plataforma como Serviço (PaaS)*. 2024. Disponível em: <<https://repositorio.ifpb.edu.br/handle/177683/4029>>.

ELEGBE, I.; WIMMER, H. Comparative analysis of virtual machine performance and security test across leading cloud providers. In: *International IoT, Electronics and Mechatronics Conference*. [S.l.]: Springer, 2025. p. 257–284.

FILHO, V. C. *Desvendando as nuvens: uma análise comparativa de desempenho de aplicações serverless e containers em provedores de computação em nuvem*. 2024. Disponível em: <<https://hdl.handle.net/11449/256373>>.

FILHO, V. C.; LOBATO, R. S. *Serverless Computing - O data center agora é o computador*. 2022. 25–28 p. In: Escola Regional de Alto Desempenho de São Paulo (ERAD-SP), 13., 2022, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação. Disponível em: <<https://doi.org/10.5753/eradsp.2022.222115>>.

IBM Data and AI Team. *Tipos de criptografia*. 2025. IBM Think. Disponível em: <<https://www.ibm.com/br-pt/think/topics/cryptography-types>>.

KADOOKA, G. Y. S. *Explorando algoritmos de compressão de dados: teoria, implementação e desempenho*. 2025. Disponível em: <<https://hdl.handle.net/11449/315442>>.

KAUSHIK, P. et al. Cloud computing and comparison based on service and performance between amazon aws, microsoft azure, and google cloud. In: *2021 International Conference on Technological Advancements and Innovations (ICTAI)*. [S.l.: s.n.], 2021. p. 268–273.

Microsoft Corporation. *Azure Functions Pricing*. [S.l.], 2025. Disponível em: <<https://azure.microsoft.com/pt-br/pricing/details/functions/#pricing>>. Acesso em: 13 dez. 2025. Disponível em: <<https://azure.microsoft.com/pt-br/pricing/details/functions/#pricing>>.

Python Software Foundation. *time — Time access and conversions*. [S.l.], 2025. Disponível em: <<https://docs.python.org/pt-br/dev/library/time.html>>. Acesso em: 12 dez. 2025. Disponível em: <<https://docs.python.org/pt-br/dev/library/time.html>>.

Python Software Foundation. *gzip — Suporte para arquivos gzip*. 2026. Documentação Python 3.14 (pt-BR). Disponível em: <<https://docs.python.org/pt-br/3.14/library/gzip.html>>.

Python Software Foundation. *hashlib — Resumos de mensagens e hashes seguros*. 2026. Documentação Python 3.14 (pt-BR). Disponível em: <<https://docs.python.org/pt-br/3.14/library/hashlib.html>>.

RAKHRA, M. et al. Digital signature verification in cloud computing. In: *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [S.l.: s.n.], 2024. p. 1–6.

RASHID, A.; CHATURVEDI, A. Cloud computing characteristics and services: A brief review. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*, v. 7, p. 421–426, 02 2019.

SERAFIM, V. d. S. *Funções de Hash*. 2012. Online. Disponível em: <http://www.serafim.eti.br/academia/recursos/Roteiro_08-Funcoes_de_Hash.pdf>.

TERADA, R. *Segurança de dados: Criptografia em rede de computador*. Editora Blucher, 2008. ISBN 9788521215400. Disponível em: <<https://books.google.com.br/books?id=Ile6DwAAQBAJ>>.

The Cryptography Developers. *Cryptography Documentation*. [S.l.], 2025. Disponível em: <<https://cryptography.io/en/latest/>>. Acesso em: 12 dez. 2025. Disponível em: <<https://cryptography.io/en/latest/>>.