

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
INSTITUTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ANDRÉ UZIEL DE SOUZA DANTAS

COMPUTAÇÃO EM NUVEM: como abordar esse tema em uma disciplina eletiva do  
curso de Ciência da Computação da UFRJ

RIO DE JANEIRO  
2025

ANDRÉ UZIEL DE SOUZA DANTAS

COMPUTAÇÃO EM NUVEM: como abordar esse tema em uma disciplina eletiva do curso de Ciência da Computação da UFRJ

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Silvana Rossetto

RIO DE JANEIRO

2025

## CIP - Catalogação na Publicação

D192c Dantas, André Uziel de Souza  
COMPUTAÇÃO EM NUVEM: como abordar esse tema em  
uma disciplina eletiva do curso de Ciência da  
Computação da UFRJ / André Uziel de Souza Dantas. --  
Rio de Janeiro, 2025.  
75 f.

Orientadora: Silvana Rossetto.  
Trabalho de conclusão de curso (graduação) -  
Universidade Federal do Rio de Janeiro, Instituto  
de Computação, Bacharel em Ciência da Computação,  
2025.

1. Computação em nuvem. 2. Sistemas distribuídos.  
3. Educação. 4. Disciplina eletiva. I. Rossetto,  
Silvana, orient. II. Título.


ANDRÉ UZIEL DE SOUZA DANTAS

COMPUTAÇÃO EM NUVEM: como abordar esse tema em uma disciplina eletiva do curso de Ciência da Computação da UFRJ

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.


Aprovado em 25 de fevereiro de 2025

BANCA EXAMINADORA:

Documento assinado digitalmente  
 **SILVANA ROSSETTO**  
Data: 28/02/2025 13:41:12-0300  
Verifique em <https://validar.iti.gov.br>


---

Silvana Rossetto  
D.Sc - Instituto de Computação (UFRJ)

Documento assinado digitalmente  
 **JOAO CARLOS PEREIRA DA SILVA**  
Data: 28/02/2025 14:47:28-0300  
Verifique em <https://validar.iti.gov.br>

---

João Carlos Pereira da Silva  
D.Sc - Instituto de Computação (UFRJ)

Documento assinado digitalmente  
 **MARIA JULIA DIAS DE LIMA**  
Data: 28/02/2025 13:58:57-0300  
Verifique em <https://validar.iti.gov.br>

---

Maria Julia de Lima  
D.Sc - Instituto Tecgraf de Desenvolvimento de Software Técnico-Científico (PUC-Rio)

## **AGRADECIMENTOS**

Primeiramente gostaria de agradecer a todos os professores da UFRJ que me apoiaram durante a minha formação, em especial à professora Silvana pela paciência e apoio na elaboração desse trabalho.

À minha família, pelo papel fundamental de incentivo que exerceram durante toda a minha graduação.

Gostaria também de agradecer aos amigos que fiz dentro da UFRJ, em especial ao meu amigo Ney, pelos momentos que passamos juntos dentro do ambiente acadêmico, e que culminaram na conclusão desse trabalho.

E por último gostaria de agradecer à cantora Taylor Swift pelas músicas que me trouxeram inspiração e me mantiveram focado durante as longas horas de pesquisa.

*"L'essentiel est invisible pour les yeux."*

**Antoine de Saint-Exupéry**

## RESUMO

A computação em nuvem é um modelo de fornecimento de recursos de computação que ganhou muita popularidade desde que começou a ser oferecida comercialmente para o público, em 2006. As vantagens trazidas por esse conjunto de tecnologias fizeram com que muitas empresas e instituições migrassem sua infraestrutura de computação para a nuvem, e utilizassem abordagens *cloud-first* no desenvolvimento de novas aplicações. Com a mudança de currículo do curso de bacharelado em Ciência da Computação da UFRJ em 2022, esse tema ganhou uma disciplina exclusiva no elenco de eletivas condicionadas, porém, até a apresentação deste trabalho, ela ainda não foi oferecida para os alunos. Dessa forma, neste trabalho nos propomos a revisitar os referenciais curriculares e os currículos de outras instituições que já abordam o tema computação em nuvem, de forma a garantir sua completude, e também gerar material didático de apoio para sua oferta. Para facilitar o entendimento de termos técnicos e específicos da computação em nuvem que aparecem nessas referências, um capítulo foi dedicado ao estudo de conceitos da nuvem, e suas aplicações. Ao final, um programa mais detalhado para a disciplina é proposto, o qual leva em conta a proposta original e todo o levantamento e análises dos referenciais curriculares e das propostas de outras instituições, realizadas ao longo do trabalho.

**Palavras-chave:** computação em nuvem; sistemas distribuídos; educação; disciplina eletiva.

## ABSTRACT

Cloud computing is a model for delivering computing resources that has gained a lot of popularity since it was first commercially offered to the public in 2006. The advantages brought by this technology have led many companies and institutions to migrate their computing infrastructure to the cloud and adopt cloud-first approaches in the development of new applications. With the curriculum change in the Bachelor's degree in Computer Science at UFRJ in 2022, this topic gained a dedicated course within the list of conditional electives. However, as of the time of this work, it has not yet been offered to students. Thus, in this paper, we aim to revisit curriculum guidelines and review the curricula of other institutions that already cover cloud computing, to ensure its completeness and also to generate supporting teaching materials for the course's future offering. To facilitate the understanding of technical and specific terms in cloud computing that appear in these references, a chapter is dedicated to the study of cloud concepts and their applications. Finally, a more detailed program for the course is proposed, considering the original proposal along with the research and analyses of curriculum guidelines and proposals from other institutions conducted throughout this work.

**Keywords:** cloud computing; distributed systems; education; conditional elective.

## LISTA DE ILUSTRAÇÕES

Figura 1 – A pilha da computação em nuvem . . . . .	17
Figura 2 – Modelos de implementação da computação em nuvem . . . . .	19
Figura 3 – Convergência de Tecnologias que levaram ao advento da computação em nuvem . . . . .	21
Figura 4 – Exemplo de <i>hipervisor</i> do tipo 1 . . . . .	22
Figura 5 – Comparação entre <i>hipervisores</i> do tipo 1 e 2 . . . . .	23
Figura 6 – Comparação entre máquinas virtuais e contêineres . . . . .	26
Figura 7 – Estrutura conceitual dos Referenciais de Formação em Computação . .	28
Figura 8 – Taxonomia de Bloom . . . . .	48
Figura 9 – Interface de criação de uma instância EC2 na AWS . . . . .	65
Figura 10 – Interface de listagem de instâncias EC2 na AWS . . . . .	66
Figura 11 – Executando comandos remotamente na instância EC2 criada a partir de uma instalação local de Linux . . . . .	66
Figura 12 – Criando uma tabela no DynamoDB . . . . .	68
Figura 13 – Código em Python para realizar operações no DynamoDB . . . . .	69
Figura 14 – Escrita em tabela no DynamoDB . . . . .	69
Figura 15 – Role IAM para acesso ao DynamoDB por funções lambda . . . . .	71
Figura 16 – Função Lambda criada . . . . .	72
Figura 17 – Código na Lambda para persistência no DynamoDB . . . . .	73
Figura 18 – Evento configurado com entrada da aplicação na Lambda . . . . .	73
Figura 19 – Lambda com integração com o API Gateway . . . . .	75
Figura 20 – Utilizando a ferramenta Postman para interagir com a API . . . . .	75

## LISTA DE QUADROS

Quadro 1 – Competências esperadas pelo referencial da SBC x Tópicos da ementa	35
Quadro 2 – Tópicos de estudo e objetivos de aprendizado segundo o referencial da ACM/IEEE x Tópicos na ementa . . . . .	36
Quadro 3 – Quadro comparativo sobre a oferta de conteúdos relacionados à computação em nuvem . . . . .	44
Quadro 4 – Tópicos cobertos no módulo 1 . . . . .	50
Quadro 5 – Tópicos cobertos no módulo 2 . . . . .	51
Quadro 6 – Tópicos cobertos no módulo 3 . . . . .	53
Quadro 7 – Tópicos cobertos no módulo 4 . . . . .	55
Quadro 8 – Tópicos cobertos no módulo 5 . . . . .	57

## LISTA DE ABREVIATURAS E SIGLAS

IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
FaaS	Function as a Service
CaaS	Container as a Service
AWS	Amazon Web Services
GCP	Google Cloud Platform
SLA	Service Level Agreement
SDK	Software Development Kit
REST	REpresentational State Transfer
TI	Tecnologia da Informação
MV	Máquina Virtual
SBC	Sociedade Brasileira de Computação
ACM	Association for Computing Machinery
IEEE	Institute of Electrical and Electronics Engineers
UFRJ	Universidade Federal do Rio de Janeiro
USP	Universidade de São Paulo
UFOP	Universidade Federal de Ouro Preto
UFMG	Universidade Federal de Minas Gerais
UNICAMP	Universidade Estadual de Campinas

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>13</b>
1.1	OBJETIVOS . . . . .	14
1.2	METODOLOGIA . . . . .	15
1.3	ORGANIZAÇÃO DO TEXTO . . . . .	15
<b>2</b>	<b>CONCEITOS BÁSICOS . . . . .</b>	<b>16</b>
2.1	MODELOS DE SERVIÇO DE COMPUTAÇÃO EM NUVEM . . . . .	16
2.1.1	Infraestrutura como serviço (IaaS) . . . . .	16
2.1.2	Plataforma como serviço (PaaS) . . . . .	17
2.1.3	Software como serviço (SaaS) . . . . .	17
2.1.4	Tudo como Serviço (XaaS) . . . . .	18
2.1.4.1	Função como Serviço (FaaS) . . . . .	18
2.1.4.2	Contêiner como Serviço (CaaS) . . . . .	18
2.2	MODELOS DE IMPLANTAÇÃO DE COMPUTAÇÃO EM NUVEM	19
2.2.1	Nuvem Pública . . . . .	19
2.2.2	Nuvem Privada . . . . .	20
2.2.3	Nuvem Híbrida . . . . .	20
2.3	TECNOLOGIAS VIABILIZADORAS . . . . .	20
2.3.1	Virtualização . . . . .	21
2.3.1.1	Máquina virtual . . . . .	22
2.3.1.2	Hipervisor . . . . .	22
2.3.1.3	Hipervisor tipo 1 . . . . .	22
2.3.1.4	Hipervisor tipo 2 . . . . .	23
2.3.1.5	Benefícios da Virtualização . . . . .	23
2.3.2	Arquitetura Orientada a Serviços . . . . .	24
2.3.3	Contêineres . . . . .	25
2.3.3.1	Contêineres x Máquinas Virtuais . . . . .	25
<b>3</b>	<b>REFERENCIAIS CURRICULARES . . . . .</b>	<b>27</b>
3.1	REFERENCIAL CURRICULAR DA SBC . . . . .	27
3.1.1	Eixo de Formação: Desenvolvimento de Sistemas . . . . .	28
3.1.2	Eixo de Formação: Gestão de Infraestrutura . . . . .	29
3.2	REFERENCIAIS CURRICULARES DA ACM/IEEE . . . . .	30
3.2.1	Computer Science Curricula 2013 . . . . .	30
3.2.2	Computer Science Curricula 2023 . . . . .	32

3.3	A DISCIPLINA NO CURSO DE CIÊNCIA DA COMPUTAÇÃO DA UFRJ . . . . .	32
3.3.1	Análise baseada no Referencial da SBC . . . . .	34
3.3.2	Análise baseada nos Referenciais da ACM/IEEE . . . . .	35
4	<b>PROGRAMAS EXISTENTES . . . . .</b>	<b>37</b>
4.1	COMPUTAÇÃO EM NUVEM COMO DISCIPLINA DE GRADUAÇÃO E PÓS GRADUAÇÃO . . . . .	37
4.1.1	Computação em Nuvem na USP . . . . .	37
4.1.2	Computação em Nuvem na UFOP . . . . .	38
4.1.3	Computação em Nuvem na UFMG . . . . .	39
4.1.4	Computação em Nuvem na Carnegie Mellon . . . . .	39
4.1.5	Computação em Nuvem na Stanford University . . . . .	40
4.2	COMPUTAÇÃO EM NUVEM COMO ESPECIALIZAÇÃO . . . . .	40
4.2.1	Computação em Nuvem na UNICAMP . . . . .	40
4.2.2	Computação em Nuvem na Mackenzie . . . . .	41
4.3	AVALIAÇÃO DOS PROGRAMAS . . . . .	42
5	<b>NOSSA PROPOSTA . . . . .</b>	<b>45</b>
5.1	ANÁLISE DA DISCIPLINA . . . . .	45
5.1.1	Pré-requisitos . . . . .	45
5.1.2	Ementa . . . . .	46
5.1.3	Conteúdo Programático . . . . .	47
5.1.4	Referências Bibliográficas . . . . .	47
5.2	PROGRAMA PROPOSTO . . . . .	48
5.2.1	Módulo 1 - Terminologia e Conceitos . . . . .	49
5.2.2	Módulo 2 - Virtualização e Hipervisores . . . . .	51
5.2.3	Módulo 3 - Paradigmas e Modelos de Computação em Nuvem . . . . .	52
5.2.4	Módulo 4 - Ambientes e Plataformas de Computação em Nuvem . . . . .	54
5.2.5	Módulo 5 - Gerenciamento, Monitoração e Governança . . . . .	56
5.2.6	Preparação para o período . . . . .	57
5.2.7	Proposta de avaliação . . . . .	57
6	<b>CONCLUSÃO . . . . .</b>	<b>58</b>
6.1	TRABALHOS FUTUROS . . . . .	59
	<b>REFERÊNCIAS . . . . .</b>	<b>60</b>
	<b>APÊNDICE A – LABORATÓRIO: CRIANDO E CONECTANDO EM UMA INSTÂNCIA EC2 NA AWS . . . . .</b>	<b>64</b>

<b>APÊNDICE B</b>	<b>– LABORATÓRIO: PERSISTINDO E CONSULTANDO DADOS EM UM BANCO DE DADOS NA NUVEM COM AMAZON DYNAMODB . . .</b>	<b>67</b>
<b>APÊNDICE C</b>	<b>– LABORATÓRIO: EXECUTANDO CÓDIGO NA NUVEM COM AWS LAMBDA . . . . .</b>	<b>70</b>
<b>APÊNDICE D</b>	<b>– TRABALHO FINAL: APLICAÇÃO COMPLETA NA NUVEM . . . . .</b>	<b>74</b>

## 1 INTRODUÇÃO

A computação em nuvem é um modelo de computação que permite o acesso a recursos de Tecnologia da Informação (TI) sob demanda por meio da Internet, normalmente com modelos de precificação conforme o uso (*pay-per-use*). Ela elimina os custos com infraestrutura física, permitindo que recursos de hardware e software que dão sustentação a uma aplicação sejam gerenciados no lado do provedor, e abstraídos a nível de serviço para o desenvolvedor ou usuário (AWS, 2024i).

Esse modelo inovador não é usado apenas no contexto de desenvolvimento de software, podendo ser oferecido como um aplicativo completo para o usuário final das mais diversas áreas, como por exemplo as suítes de trabalho da Google e da Microsoft, que oferecem serviços como edição de documentos e troca de mensagens diretamente no navegador, sem a necessidade de instalação de programas locais. Esse modelo de serviço, em que o usuário é completamente alheio à infraestrutura subjacente que sustenta suas atividades é chamado de Software como Serviço (AZURE, 2024g).

No contexto de desenvolvimento de software, a computação em nuvem apresenta várias vantagens em relação ao custo operacional, como a eliminação da necessidade de contratação de especialistas de TI para gerenciar a infraestrutura local (máquinas e redes físicas, refrigeração), a redução do custo de entrada de uma nova empresa no mercado — dado que não há necessidade de comprar servidores físicos — e a troca de despesas fixas por despesas variáveis (precificação conforme uso). Isso permite que uma aplicação lide com picos de uso sem a necessidade de investimento em hardware mais potente, que ficaria subutilizado a maior parte do tempo durante períodos de baixa demanda, gerando ganhos tanto para os gestores quanto para o meio ambiente (AWS, 2024i).

Por outro lado, a uso correto desses recursos apresenta diversos desafios aos desenvolvedores. A natureza altamente escalável aliada à disponibilidade virtualmente ilimitada da computação em nuvem pode resultar em custos significativamente elevados no caso de aplicações que não estejam devidamente otimizadas. Enquanto que em sistemas convencionais o fluxo de execução atingiria limites físicos, como o uso de CPU ou memória, aplicações que operam em ambientes de nuvem continuarão a alocar mais recursos conforme necessário. Portanto, é fundamental que o desenvolvedor compreenda os modelos de precificação de serviços utilizados e seja capaz de identificar problemas em aplicações, por meio de ferramentas de gestão e de monitoramento, e de técnicas como a definição de alertas de uso e de custo.

Dois modelos de serviço se destacam para os desenvolvedores: Infraestrutura como Serviço (IaaS), que oferece mais flexibilidade e controle sobre os recursos de TI, e Plataforma como Serviço (PaaS), na qual não é necessário gerenciar a infraestrutura subjacente, permitindo ao desenvolvedor manter o foco na implantação e no gerenciamento de seus

aplicativos (AWS, 2024i). Assim, torna-se cada vez mais necessário que os profissionais da área de computação saibam escolher e avaliar qual modelo de serviço é mais adequado para solucionar os problemas que lhes são apresentados, de forma a evitar o investimento em soluções redundantes, e incentivar o uso de ferramentas já testadas e consolidadas.

Atualmente, a computação em nuvem é cada vez mais adotada por empresas e indivíduos no desenvolvimento de novas aplicações. Com isso, é muito provável que alunos de cursos da área de computação já terão contato com essa tecnologia durante seu primeiro estágio, e muito provavelmente trabalharão com ela quando formados, seja desenvolvendo uma nova aplicação ou dando manutenção a uma existente, seja fazendo uso de aplicações desenvolvidas para execução em nuvem.

O crescente uso da computação em nuvem traz novas demandas para a formação dos estudantes da área. O conhecimento sobre os conceitos básicos relacionados a esse novo modelo de infraestrutura de hardware e software, seu modelo de funcionamento, benefícios e desafios permitirá que o estudante complemente a sua formação e entre no mercado de trabalho mais bem preparado para os desafios que deverá enfrentar. Além disso, o discente terá mais autonomia para tomar decisões estratégicas e oferecer alternativas arquiteturais mais eficientes para os sistemas em projeto.

No mercado, as principais provedoras de computação na nuvem atualmente são a Amazon (AWS, 2024i), Microsoft (AZURE, 2024g) e Google (GCP, 2024c). Essas empresas costumam fornecer serviços equivalentes, como máquinas virtuais, banco de dados, armazenamento de objetos, mas que podem variar em custo, flexibilidade e capacidade de processamento. O primeiro contato com ao menos uma dessas plataformas já durante a graduação auxiliará a compreensão da aplicabilidade dos paradigmas da nuvem e modelo de precificação.

A disciplina de computação em nuvem já está presente no referencial curricular da ACM e IEEE para o curso de Ciência da Computação (ACM/IEEE, 2013) desde 2013 como disciplina eletiva, e desde então a demanda por esse curso só cresceu. No referencial da SBC (SBC, 2017), a computação em nuvem faz parte do eixo de desenvolvimento de sistemas, como parte da competência de tomada de decisões e inovação, e do eixo de gestão de infraestrutura, como parte da competência de planejamento e manutenção de sistemas de computação.

## 1.1 OBJETIVOS

Na última reforma curricular do curso de bacharelado em Ciência da Computação da UFRJ, a disciplina Computação em Nuvem (ICP025) foi adicionada no elenco de disciplinas eletivas, no entanto até o momento da apresentação desse trabalho, ainda não foi oferecida. Embora já exista uma proposta de programa para essa disciplina, ainda não há conteúdo teórico e prático que viabilize sua oferta.

Assim, esse trabalho tem como objetivo propor um programa mais detalhado para essa disciplina recém criada, com material didático de apoio, aliado a uma metodologia de ensino que favoreça o engajamento dos alunos na realização de atividades práticas. Isso será feito por meio de laboratórios executados dentro do próprio ambiente da nuvem, de forma a permitir que os alunos apliquem o conhecimento adquirido.

## 1.2 METODOLOGIA

Tomaremos a proposta inicial da disciplina como ponto de partida deste trabalho, propondo sugestões e melhorias com o intuito de torná-la mais completa. Para isso, revisitaremos os referenciais curriculares da ACM/IEEE e SBC, a fim de garantir que todas as competências exigidas estão sendo cobertas. Além disso, avaliaremos como os conteúdos de computação em nuvem estão sendo oferecidos em outras instituições de ensino, brasileiras e estrangeiras, para observar as práticas e abordagens de ensino já utilizadas, material e atividades propostas.

Para a parte prática, vamos propor roteiros de atividades que levam em conta o ambiente da AWS, mas minimizando a dependência da plataforma ao oferecer alternativas e serviços equivalentes, de forma a torná-los extensíveis ou adaptáveis para outras plataformas com esforço reduzido. A escolha da nuvem da Amazon para os exemplos se deu exclusivamente pela familiaridade do autor.

## 1.3 ORGANIZAÇÃO DO TEXTO

No capítulo 2 serão explicados conceitos básicos e terminologias adotadas na área de computação em nuvem, de forma a aprofundar o conhecimento sobre esse tema e facilitar a compreensão do restante do texto. No capítulo 3 vamos nos aprofundar sobre os principais referenciais curriculares dos cursos de graduação em Computação, e o que sugerem em termos de conteúdos relacionados à computação em nuvem. No capítulo 4 apresentaremos exemplos de oferta de disciplinas relacionadas à computação em nuvem em instituições de ensino no Brasil e no exterior e as compararemos entre si. No capítulo 5 descreveremos em detalhe nossa proposta, levando em consideração a ementa e programa propostos na reforma curricular do curso, as competências recomendadas pelos referenciais, e a experiência de outras instituições. Finalmente, no capítulo 6 concluiremos o trabalho. Os roteiros das atividades práticas citados na proposta serão descritos nos apêndices.

## 2 CONCEITOS BÁSICOS

Nesse capítulo, vamos nos aprofundar sobre conceitos básicos relacionados à computação em nuvem que serão utilizados nos capítulos posteriores. Começaremos a discussão pelas classificações, primeiramente em modelos de serviço e posteriormente em modelos de implantação. Em seguida, passaremos pelas principais tecnologias que viabilizaram a computação em nuvem como modelo computacional. Finalmente, abordaremos rapidamente o tema de contêineres, que é uma tecnologia amplamente utilizada em aplicações desenvolvidas para a nuvem.

### 2.1 MODELOS DE SERVIÇO DE COMPUTAÇÃO EM NUVEM

A computação em nuvem pode ser classificada tradicionalmente em três modelos de serviço, de acordo com o nível de abstração desejado, oferecendo diferentes níveis de controle, flexibilidade e gerenciamento (AWS, 2024i). Esses níveis de abstração podem ser comparados a uma arquitetura em camadas, em que serviços de camadas superiores podem ser compostos por camadas inferiores (BUY YA JAMES BROBERG, 2011). São eles, em ordem crescente de abstração: Infraestrutura como serviço (IaaS), Plataforma como serviço (PaaS) e Software como serviço (SaaS).

Adicionalmente, novos modelos de serviço mais específicos foram surgindo à medida que a computação em nuvem foi se consolidando no mercado. Essa nova categoria de modelos, que está em constante expansão, pode ser chamada de Tudo como Serviço (XaaS) (IBM, 2024e).

A figura 1 traz exemplos de cada modelo de serviço segundo a classificação tradicional, e a ferramenta utilizada para acessá-lo e gerenciá-lo.

#### 2.1.1 Infraestrutura como serviço (IaaS)

Esse modelo oferece o nível mais alto de flexibilidade e controle de gerenciamento sobre os recursos de TI contratados. O IaaS oferece acesso a recursos de rede, máquinas virtuais e espaço de armazenamento, tornando-o mais semelhante aos recursos tradicionais de TI (AWS, 2024i).

Um exemplo de serviço que segue esse modelo é o EC2 da AWS (AWS, 2024g), que oferece máquinas virtuais com uma camada de software adicional que pode ser configurada de forma análoga a como um servidor físico seria configurado. Esse serviço permite que gestores realizem ações de alto nível, como iniciar e desligar as máquinas, além da instalação de softwares adicionais (BUY YA JAMES BROBERG, 2011). Os serviços equivalentes na Azure e Google Cloud são o Azure Virtual Machines (AZURE, 2024e) e o Google Compute Engine (GCP, 2024a), respectivamente.

Figura 1 – A pilha da computação em nuvem

Classe de Serviço	Principal Ferramenta de Acesso e Gerenciamento	Conteúdo do Serviço
 SaaS	Navegador Web	<b>Aplicações em Nuvem</b> Redes sociais, Suítes de trabalho, Processamento de Vídeo
 PaaS	<b>Ambiente de Desenvolvimento em Nuvem</b>	<b>Plataforma em Nuvem</b> Linguagens de Programação, Frameworks, Dados estruturados
 IaaS	<b>Gerenciador de Infraestrutura Virtual</b>	<b>Infraestrutura em Nuvem</b> Servidores de Computação, Armazenamento de dados, Firewall, Balanceadores de Carga

(BUYYA JAMES BROBERG, 2011) (Traduzido para este trabalho)

### 2.1.2 Plataforma como serviço (PaaS)

Esse modelo intermediário se refere a serviços de computação que oferecem todos os recursos de hardware e software necessários para o desenvolvimento e sustentação de um aplicativo em nuvem. A PaaS permite que os desenvolvedores criem aplicativos mais rapidamente, sem se preocupar com a configuração ou gerenciamento da infraestrutura subjacente de servidores, armazenamento, rede e banco de dados (AZURE, 2024g).

O Heroku é um exemplo de serviço que segue esse modelo, permitindo ao usuário realizar *deploy* de aplicações a partir de um repositório de código (ex: GitHub) (HEROKU, 2024). Uma vez configurada a aplicação, ele se responsabiliza por expor uma *url* para acesso do usuário final, e também monitora alterações no repositório de forma a manter sempre a versão mais atualizada disponível.

As grandes provedoras de computação em nuvem também oferecem serviços classificados como PaaS, como é o caso do AWS Elastic Beanstalk (AWS, 2024c), que possui maior disponibilidade global em relação ao Heroku, por fazer parte da infraestrutura da AWS.

### 2.1.3 Software como serviço (SaaS)

Esse modelo compreende serviços destinados ao usuário final, na forma de um produto completo, como alternativa a um aplicativo desktop, por exemplo. Nesse formato, o

usuário é completamente alheio à camada de software e hardware que dá sustentação à aplicação, sua única preocupação é com o uso do software específico (AWS, 2024i).

Um exemplo de aplicação que segue esse modelo é um serviço de email na nuvem, como Gmail e Outlook, contrapondo-se aplicações de email desktop como o Mozilla Thunderbird. Outro exemplo é o Google Docs (Google Workspace), que surgiu como uma alternativa ao aplicativo desktop do Word (Microsoft Office), que por muito tempo dominou o mercado.

#### 2.1.4 Tudo como Serviço (XaaS)

Os modelos conhecidos como *Everything as a Service* ou XaaS englobam uma nova categoria de fornecimento de serviços na nuvem, que vai além da definição tradicional (IaaS, PaaS e SaaS). Essa nova classificação pode ser usada para descrever especializações ou combinações desses modelos originários, e que continuam a se expandir à medida que novas demandas surgem e mais acrônimos são criados.

Como o XaaS se refere a uma categoria de serviços que está em constante expansão, selecionamos dois exemplares populares para nos aprofundarmos: a Função como Serviço (FaaS) e o Contêiner como Serviço (CaaS).

##### 2.1.4.1 Função como Serviço (FaaS)

Esse modelo oferece uma plataforma para que clientes executem código de forma remota sem a necessidade de gerenciar ou provisionar os servidores em que ele executa. O provedor é responsável por gerir o hardware físico, o sistema operacional, a conexão com a Internet e o *runtime* da linguagem, enquanto cabe ao desenvolvedor somente fazer *upload* do código da aplicação e definir o eventos que disparam sua execução (IBM, 2024b).

Um exemplo desse modelo é o AWS Lambda, que é o serviço da AWS que permite executar código em resposta a eventos, e que torna transparente ao gestor quando a máquina virtual subjacente é ligada ou desligada (AWS, 2024d). A Azure e a Google Cloud também contam com serviços equivalentes, que são o Azure Functions (AZURE, 2024a) e o Cloud Run functions (GCP, 2024b), respectivamente.

##### 2.1.4.2 Contêiner como Serviço (CaaS)

O modelo CaaS se refere a uma categoria de serviços em nuvem que permite que gestores monitorem e implementem aplicações que rodam em contêineres de forma simplificada, segura e transparente. Cabe ao desenvolvedor fornecer as imagens de contêiner e configurar o processo de orquestração, enquanto o provedor de nuvem fica responsável pela implementação, sustentação, controle de acesso e escalabilidade (IBM, 2024d). No final desse capítulo, os contêineres serão explorados com mais profundidade.

Alguns exemplos de CaaS no mercado são o Amazon Elastic Container Service (ECS) (AWS, 2024a) e o Azure Container Instances (ACI) (AZURE, 2024c).

## 2.2 MODELOS DE IMPLANTAÇÃO DE COMPUTAÇÃO EM NUVEM

Embora a computação em nuvem tenha se destacado principalmente por conta das nuvens públicas, outros modelos de implantação, que variam de acordo com sua localização física e distribuição, também são adotados. Nesse sentido, independente de seu modelo de serviço, a nuvem pode ser classificada em três modelos de implantação: nuvem pública, nuvem privada e nuvem híbrida (BUYYYA JAMES BROBERG, 2011). A figura 2 mostra os casos de uso e características de cada um desses modelos.

Figura 2 – Modelos de implementação da computação em nuvem



(BUYYYA JAMES BROBERG, 2011) (Traduzido para este trabalho)

### 2.2.1 Nuvem Pública

As nuvens públicas são administradas por provedores de nuvens terceirizados, e todo o hardware e software que dá sustentação a ela pertence ao provedor. Seus serviços são acessados sob demanda por meio da Internet, coberto por um *Service Level Agreement* (SLA) (AZURE, 2024g).

As nuvens públicas mais conhecidas e utilizadas hoje e sua respectiva participação no mercado são (LU, 2024):

- Amazon Web Services (AWS) [31%]
- Microsoft Azure [24%]
- Google Cloud Platform (GCP) [11%]
- Alibaba Cloud [4%]
- Salesforce [3%]
- IBM Cloud [2%]
- Oracle Cloud [2%]

### 2.2.2 Nuvem Privada

As nuvens privadas são gerenciadas pela própria organização em data centers próprios, e também são conhecidas como “no local” (*on premises*). Esse modelo oferece mais controle, segurança e gerenciamento de dados, na medida em que ela pode ser acessada somente pela rede interna da organização, embora exija mais custo com manutenção (GCP, 2024c).

### 2.2.3 Nuvem Híbrida

Nuvens híbridas combinam características de nuvens públicas e privadas, o que permite que organizações garantam seus requisitos de segurança e conformidade ao mesmo tempo que aproveitam os serviços oferecidos por nuvens públicas (GCP, 2024c).

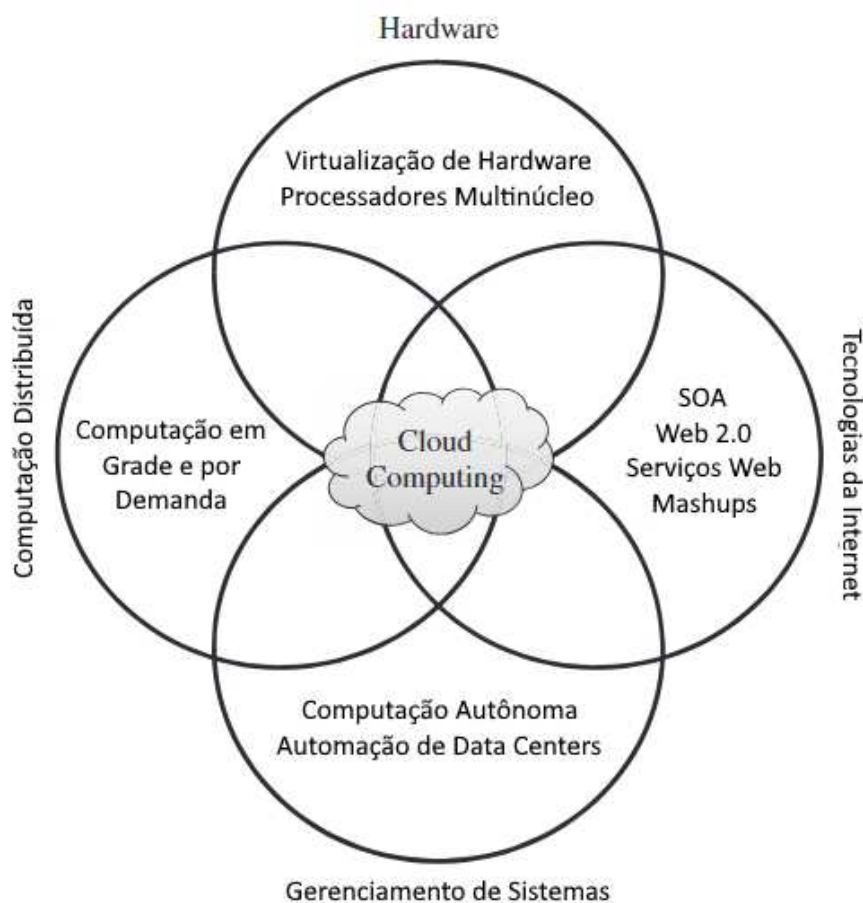
## 2.3 TECNOLOGIAS VIABILIZADORAS

Segundo Buyya et al (BUYYA JAMES BROBERG, 2011), a computação em nuvem pode ser definida da seguinte forma:

A nuvem é um sistema de computação paralelo e distribuído que consiste em uma coleção de computadores interconectados e virtualizados que são dinamicamente provisionados e apresentados como um ou mais recursos de computação baseados em um acordo a nível de serviço (SLA) estabelecido através da negociação entre o provedor de nuvem e os consumidores.

A computação em nuvem é o resultado do avanço de diversas tecnologias, principalmente nas áreas de hardware, redes (*Internet*), computação distribuída e gerenciamento de sistemas, como mostra a figura 3. Nesse capítulo, vamos nos aprofundar mais sobre os temas de virtualização e arquitetura orientada a serviços, por estarem mais diretamente relacionados com esse trabalho, visto que já são assuntos previstos na ementa da disciplina.

Figura 3 – Convergência de Tecnologias que levaram ao advento da computação em nuvem



(BUYYA JAMES BROBERG, 2011) (Traduzido para este trabalho)

### 2.3.1 Virtualização

A tecnologia de virtualização viabiliza a computação em nuvem. Provedores de nuvem configuram e mantêm os seus *datacenters*, os quais são particionados em diferentes ambientes virtuais oferecidos para os consumidores por meio de Interfaces de Programação de Aplicação (APIs). Dessa forma, as necessidades de infraestrutura podem ser atendidas como um serviço totalmente gerenciado (AWS, 2024m). Por conta disso, estudar o avanço das tecnologias de virtualização é o primeiro passo para entender o surgimento da computação em nuvem.

A ideia de virtualizar recursos de um computador, incluindo processadores, memória e dispositivos de entrada e saída já está consolidado há décadas, com o intuito de melhorar o compartilhamento e aproveitamento de sistemas de computação. A virtualização permite a execução de múltiplos sistemas operacionais e pilhas de software em uma única máquina física. Um *hipervisor* é o programa responsável por intermediar o acesso ao hardware da máquina física, fornecendo a cada sistema operacional convidado uma máquina virtual

(BUYYA JAMES BROBERG, 2011).

### 2.3.1.1 Máquina virtual

Uma máquina virtual (MV) é um computador definido por software executado em um computador físico com um sistema operacional e recursos de computação separados. O computador físico é chamado de máquina hospedeira (*host*) e as máquinas virtuais são chamadas de máquinas convidadas (*guests*).

### 2.3.1.2 Hipervisor

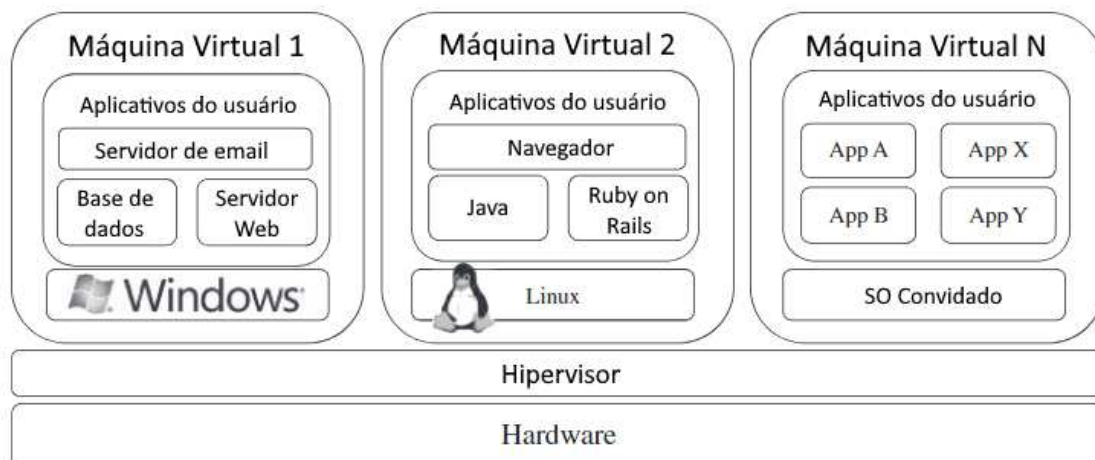
O *hipervisor* é o componente de software que gerencia várias máquinas virtuais em um computador. Sua função é garantir que cada máquina virtual não interfira na operação das outras, e que receba os recursos alocados. Os *hipervisores* podem ser classificados em tipo 1 e tipo 2, dependendo de onde ele executa.

### 2.3.1.3 Hipervisor tipo 1

Esse tipo de *hipervisor*, também chamado de *bare-metal*, é um programa instalado diretamente no hardware do computador, em oposição à instalação no sistema operacional. Por conta disso, os *hipervisores* do tipo 1 costumam ter melhor desempenho e são encontrados mais comumente em aplicações corporativas. Alguns exemplos desse tipo de *hipervisor* são o Microsoft Hyper-V (MICROSOFT, 2024) e o Xen (XEN, 2024).

A figura 4 mostra uma máquina *host* virtualizando três máquinas convidadas com sistemas operacionais diferentes, por meio de um *hipervisor* do tipo 1. Podemos observar que o *hipervisor* opera logo acima do hardware da máquina.

Figura 4 – Exemplo de *hipervisor* do tipo 1



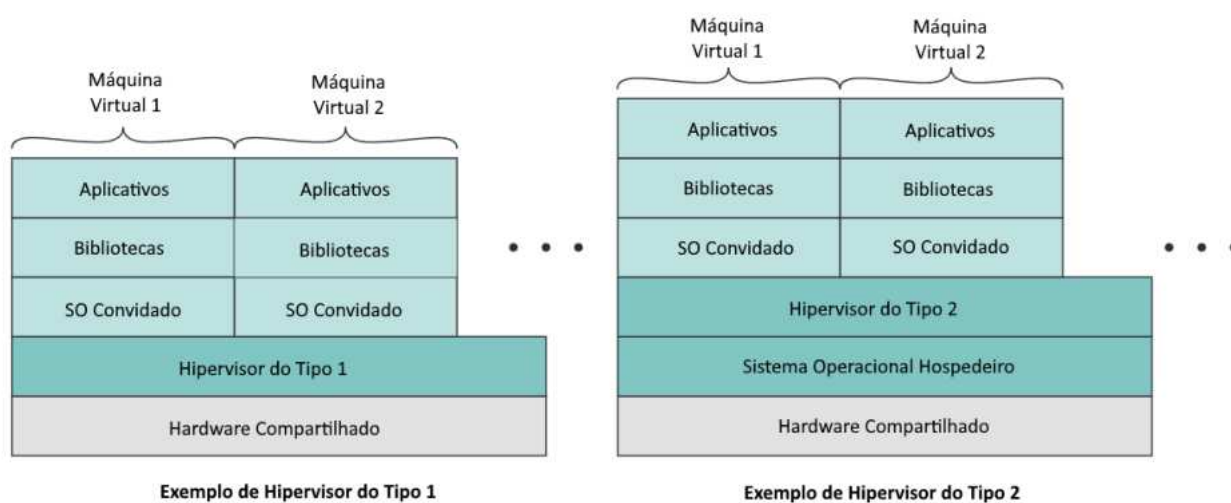
(BUYYA JAMES BROBERG, 2011) (Traduzido para este trabalho)

### 2.3.1.4 Hipervisor tipo 2

Esse tipo de *hipervisor* é instalado como um programa que roda em cima do sistema operacional. Por conta disso, ele é mais encontrado em computadores pessoais. Alguns exemplos desse tipo de *hipervisor* são o Oracle Virtualbox (ORACLE, 2024) e o VMware Workstation (VMWARE, 2024).

A figura 5 mostra uma comparação entre os tipos de hipervisores. Observa-se que o *hipervisor* do tipo 1 opera logo acima do hardware, enquanto que o *hipervisor* do tipo 2 opera acima do SO executando sobre o hardware físico.

Figura 5 – Comparação entre *hipervisores* do tipo 1 e 2



(STALLINGS, 2017) (Traduzido para este trabalho)

### 2.3.1.5 Benefícios da Virtualização

Embora inicialmente os benefícios da virtualização fossem voltados para o compartilhamento de recursos, mais recentemente — com a crescente adoção da virtualização em sistemas de servidores e clientes — pesquisadores e praticantes da área destacam três capacidades básicas no que tange ao gerenciamento da carga de trabalho em sistemas virtualizados: isolamento, consolidação e migração (BUYA JAMES BROBERG, 2011).

O **isolamento** da carga de trabalho é alcançado porque todas as instruções do programa estão totalmente confinadas dentro da máquina virtual, o que leva a uma melhor segurança do sistema. Além disso, o uso de máquinas virtuais também oferece melhor controle de desempenho, pois a execução (ou falhas) de uma MV não afeta o desempenho de outra.

A **consolidação** de várias cargas de trabalho individuais e heterogêneas em uma única máquina física leva a uma melhor utilização do sistema. Essa prática também é empregada

para superar possíveis incompatibilidades de software e hardware em caso de atualizações, dado que é possível executar sistemas operacionais legados e novos simultaneamente.

A **migração** de cargas de trabalho visa facilitar a manutenção de hardware, balanceamento de carga e recuperação de desastres. Isso é feito encapsulando o estado do sistema operacional convidado dentro de uma MV e permitindo que ela seja suspensa, totalmente serializada, migrada para uma plataforma diferente e retomada imediatamente ou preservada para ser restaurada posteriormente. O estado de uma MV inclui uma imagem completa de disco ou partição, arquivos de configuração e uma imagem da memória principal.

Em suma, foi a aplicação dessas características da virtualização que tornou possível a computação em nuvem como a conhecemos hoje, permitindo a oferta de recursos escaláveis e sob demanda, com flexibilidade e eficiência no uso dos recursos físicos. A virtualização não apenas otimizou a infraestrutura, mas também possibilitou a criação de ambientes de computação isolados entre si que atendem de forma mais ágil às necessidades dos desenvolvedores.

### 2.3.2 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (*Service Oriented Architecture* ou SOA) é uma forma de desenvolvimento que usa de componentes de software autônomos, intitulados serviços, para criar aplicações distribuídas. Cada serviço é responsável por um domínio da aplicação, e todos eles devem poder se comunicar entre si, independente da plataforma ou linguagem que estão implementados. Por meio da SOA, é possível reutilizar serviços em sistemas diferentes ou combinar vários serviços independentes para realizar tarefas complexas (AWS, 2024k).

Por exemplo, uma empresa pode desenvolver um serviço de autenticação uma única vez, e reutilizá-lo em todas as suas novas aplicações, ao invés de reimplementar códigos que remetem a esse domínio todas as vezes. Dessa forma, ela aumenta a clareza de novos códigos desenvolvidos, pois não há mistura de funcionalidades, evita a redundância e minimiza riscos, pois utiliza de serviços mais robustos e já consolidados.

O objetivo da SOA é tratar de requisitos como acoplamento fraco, orientação a padrões e interoperabilidade. Cada serviço é um módulo bem definido e auto-contido, que oferece funcionalidades para o negócio que sejam independentes do estado ou contexto de outros serviços. Serviços são descritos em uma linguagem padronizada e com uma interface pública, e embora alguns deles tenham o intuito de servir aplicações para o usuário final, seu verdadeiro poder está em sua interface ser acessível por outros serviços. Na prática, isso faz com que cada serviço ofereça sua própria interface, o que pode tornar a composição de serviços uma tarefa complexa (TANENBAUM, 2023).

No contexto de computação em nuvem, mais especificamente no modelo de Software como Serviço, aplicações para o usuário final que rodam na nuvem podem ser compostas

de outros serviços de diferentes provedores, independente de quem hospeda a aplicação final. Serviços como autenticação, email, calendário, banco de dados e armazenamento de objetos são alguns dos exemplos que podem ser oferecidos, cabendo somente ao desenvolvedor escolher quais serviços oferecem a melhor usabilidade/custo para compor sua aplicação (BUYYA JAMES BROBERG, 2011).

### 2.3.3 Contêineres

Um contêiner é um pacote de software que inclui o código de uma aplicação, suas bibliotecas e todas as outras dependências necessárias para sua execução. Sua função é isolar o código do ambiente em que é executado e garantir que ele executará de maneira uniforme independente da configuração das camadas de software inferiores, como o sistema operacional, e a versão das bibliotecas instaladas. (AZURE, 2024f).

O uso de contêineres traz vários benefícios, como a portabilidade, permitindo que desenvolvedores desenvolvam aplicações com base em um sistema de computação dado, e posteriormente as implantem em sistemas dedicados, sem a preocupação com a compatibilidade.

Enquanto máquinas virtuais utilizam *hipervisores* para intermediar a comunicação entre o sistema host e convidado, os contêineres usam **mecanismos de contêiner** (*Container Engine*), sendo o mais popular o Docker (DOCKER, 2024). Um mecanismo de contêiner é um programa capaz de inicializar contêineres com base em uma imagem de contêiner, e gerenciar os recursos alocados para a execução da imagem (AWS, 2024n).

No contexto de nuvem, é possível executar uma imagem de contêiner criando uma máquina virtual (IaaS) e instalando nela um mecanismo de contêiner, no entanto também é possível utilizar um serviço gerenciado (*managed*) para isso, a orquestração de contêineres, onde as funcionalidades de containerização são abstraídas. No contexto de nuvem, esse serviço também é chamado de CaaS (Contêiner como Serviço), que já foi citado na seção de modelos de serviço.

A AWS tem ainda outro serviço relacionado a contêineres, o Elastic Kubernetes Service (EKS) (AWS, 2024b). Kubernetes (KUBERNETES, 2024) é um serviço de orquestração de contêineres *open-source*, que permite gerenciar uma aplicação em contêiner sem ficar acoplado a um CaaS específico. A adoção desse serviço em oposição ao Elastic Container Service (ECS) traz diversas vantagens, como tornar mais fácil a migração para outro provedor e tornar possível a execução local para desenvolvimento e avaliação.

#### 2.3.3.1 Contêineres x Máquinas Virtuais

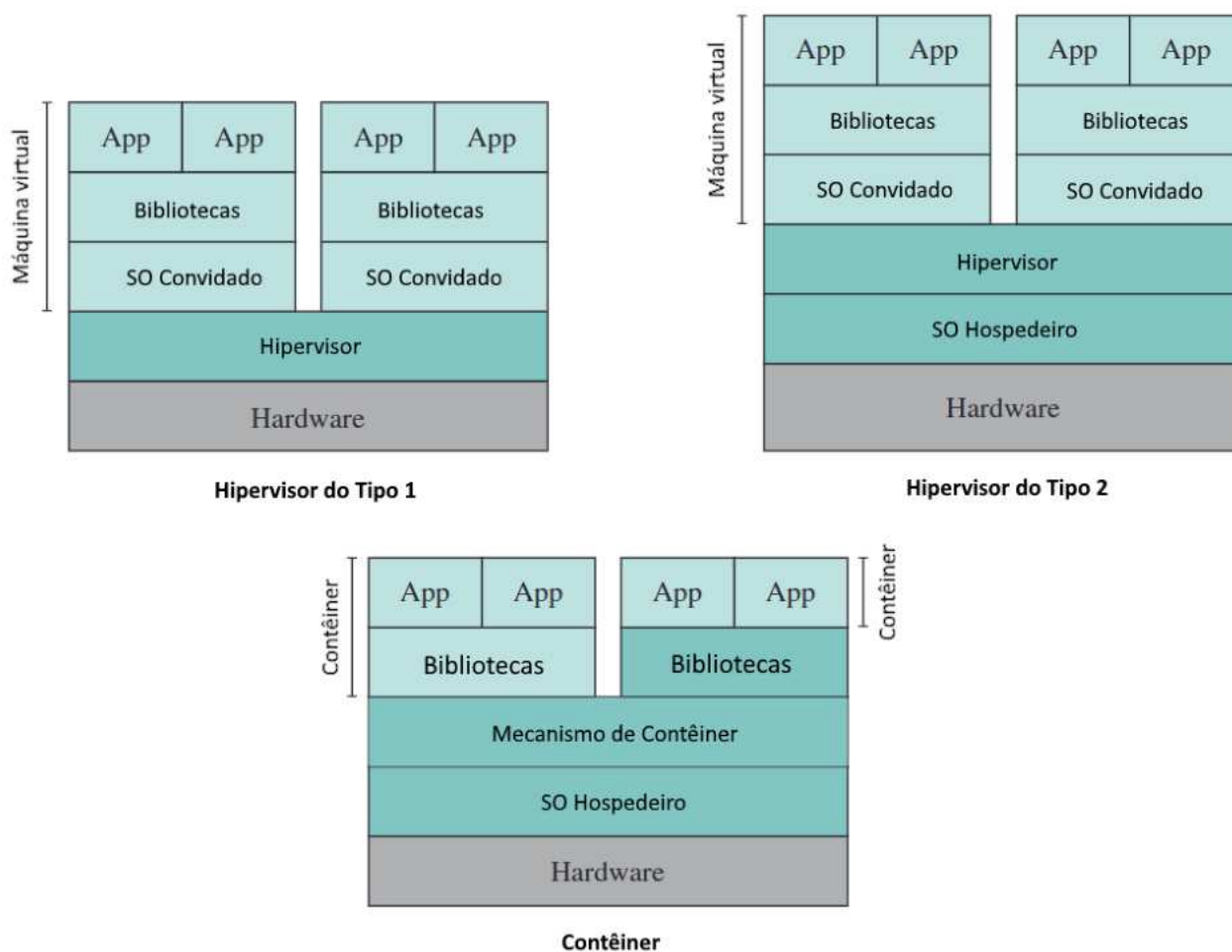
Contêineres são mais leves e eficientes do que máquinas virtuais, pois compartilham dos mesmos recursos da máquina em que são executados, sem a necessidade de sustentar

um sistema operacional para cada um, o que os torna escaláveis e rápidos de iniciar e reiniciar.

Por outro lado, também apresentam muitas semelhanças com máquinas virtuais, principalmente no quesito isolamento, pois cada contêiner executa isoladamente em relação aos outros contêineres e da máquina hospedeira, com espaços de disco e memória isolados, permitindo que falhas em um contêiner específico não interfiram nos demais.

A figura 6 faz uma comparação entre as camadas de software em que se encaixam os contêineres e as máquinas virtuais (*hipervisores* do tipo 1 e 2). É possível reparar que contêineres operam somente com um sistema operacional, e podem compartilhar bibliotecas com a máquina hospedeira.

Figura 6 – Comparação entre máquinas virtuais e contêineres



(STALLINGS, 2017) (Traduzido para este trabalho)

### 3 REFERENCIAIS CURRICULARES

Esse capítulo consiste de uma consolidação das competências relacionadas à computação em nuvem esperadas de cursos de graduação em Ciência da Computação, seguindo os referências curriculares da SBC e da ACM/IEEE. Ao final, avaliaremos como a ementa e o programa inicial propostos no currículo do curso de Ciência da Computação da UFRJ, criado a partir da última reforma curricular de 2022, atende às recomendações desses referenciais.

#### 3.1 REFERENCIAL CURRICULAR DA SBC

O último referencial de formação para os cursos de graduação em computação da SBC foi publicado em 2017. Essa edição contém os referenciais de formação para os cursos de Ciência da Computação, Engenharia de Computação, Engenharia de Software, Licenciatura em Computação, Sistemas de Informação e cursos superiores de tecnologia em computação.

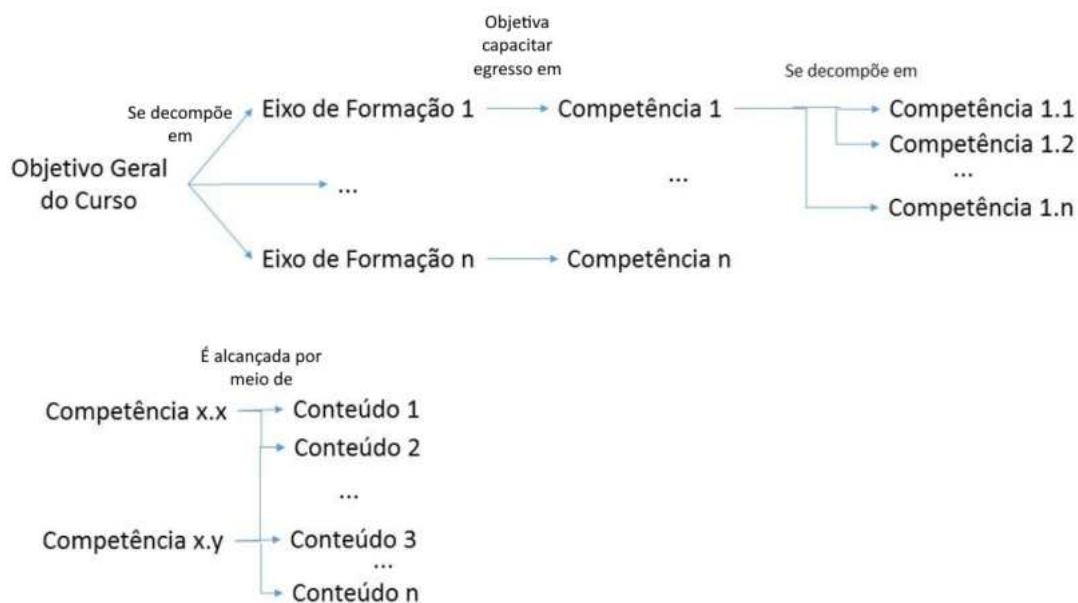
Esse referencial foi feito pela Comissão de Educação da SBC em conjunto com diversos associados da SBC de diferentes universidades do Brasil. Referenciais de Formação (RFs) não são currículos, mas um material de referência para as instituições elaborarem os seus currículos, levando em consideração os diversos aspectos institucionais e regionais (SBC, 2017).

Os Referenciais Curriculares da SBC seguem uma abordagem orientada a competências esperadas do egresso do curso. Para a definição das competências, é utilizado um modelo baseado na Taxonomia de Bloom Revisada, em que uma competência pode expressar o conhecimento, as habilidades ou as atitudes esperadas de um egresso do curso, sob a perspectiva de objetivos de aprendizagem (FERRAZ, 2010).

Seguindo essa abordagem, o referencial de formação para o curso de Ciência da Computação foi elaborado decompondo o objetivo geral do curso em 7 eixos de formação, cada um deles responsável por capacitar o egresso do curso em competências mais genéricas, que podem ser decompostas em competências derivadas, e estas são alcançadas por meio de conteúdos específicos, como mostra a figura 7.

Analisando as competências destacadas, três apresentam conteúdos relacionadas à computação em nuvem, sendo dois deles dentro do referencial curricular para o curso de Ciência da Computação, e o terceiro para o curso de Sistemas de Informação. No caso do curso de Ciência da Computação, a computação em nuvem se encontra dentro dos eixos Desenvolvimento de Sistemas e Gestão de Infraestrutura.

Figura 7 – Estrutura conceitual dos Referenciais de Formação em Computação



Fonte: (SBC, 2017)

### 3.1.1 Eixo de Formação: Desenvolvimento de Sistemas

Esse eixo de formação trata tanto da criação de sistemas quanto da adaptação de sistemas existentes. Suas competências vão desde o levantamento de requisitos funcionais e não-funcionais, até sua análise, modelagem, projeto, implementação e teste.

**COMPETÊNCIA:** Desenvolver sistemas computacionais que atendam qualidade de processo e de produto, considerando princípios e boas práticas de engenharia de sistemas e engenharia de software, incluindo:

- Identificar, analisar, especificar, validar requisitos.
- Projetar soluções computacionais em harmonia com o ambiente social e físico no seu entorno de aplicação.
- Implementar sistemas computacionais utilizando ambientes de desenvolvimento apropriados.
- Testar e manter sistemas computacionais

Competência Derivada C.2.2: Tomar decisões e inovar, com base no conhecimento do funcionamento e das características técnicas de hardware e da infraestrutura de software dos sistemas de computação, consciente dos aspectos éticos, legais e dos impactos ambientais decorrentes.

Fonte: (SBC, 2017)

A computação em nuvem simplifica o trabalho do desenvolvedor, reduzindo o tempo necessário para colocar aplicações em execução. No entanto, para fazer bom uso de seus recursos, é necessário ter consciência crítica sobre eles. A compreensão das melhores práticas de desenvolvimento para este meio e o conhecimento dos diferentes serviços e

provedores no ambiente da nuvem torna possível a construção de software que seja ao mesmo tempo eficiente e fracamente acoplado a serviços específicos.

O desenvolvimento de aplicações para a nuvem não só traz consigo uma série de boas práticas e padrões (como microsserviços, contêineres e automação) que são fundamentais na engenharia de sistemas e software modernos, mas também envolve decisões éticas e ambientais, como a seleção de provedores que utilizam energia renovável ou práticas sustentáveis, além de considerar a proteção de dados e a privacidade. O egresso deve ser capaz de tomar essas decisões já durante as etapas iniciais do desenvolvimento, de forma a evitar complicações futuras.

### 3.1.2 Eixo de Formação: Gestão de Infraestrutura

Esse eixo de formação trata da complexidade de gerir um sistema computacional de forma a fazê-lo operar de acordo com seus requisitos de desempenho, segurança, conectividade, disponibilidade, confiabilidade, custos entre outros. Para isso, devem ser alocados recursos de hardware e software para processamento, armazenamento, comunicação e interação com o meio, por meio de computadores, redes, periféricos e sistemas operacionais etc.

COMPETÊNCIA: Gerenciar infraestrutura computacional em sua plenitude, incluindo projeto, implantação e manutenção, assim definidos:

- Projetar uma infraestrutura computacional a partir das especificações dos sistemas computacionais que irão compartilhar os recursos da infraestrutura e das necessidades adicionais decorrentes desse uso compartilhado.
- Implantar a infraestrutura computacional, com domínio do processo de aquisição ou contratação de componentes de hardware e software, bem como do processo de instalação, configuração e integração desses componentes.
- Manter a infraestrutura computacional em conformidade com a sua especificação na eventual ocorrência de alterações no seu contexto de operação.

Competência Derivada C.5.9. Especificar, projetar, implementar, manter e avaliar sistemas de computação, empregando teorias, práticas e ferramentas adequadas.

Fonte: (SBC, 2017)

Embora a infraestrutura na nuvem abstraia aspectos físicos, como a compra, conexão e a instalação de componentes de hardware, ainda é necessário definir e dar eventual sustentação a diversos aspectos de sistemas em nuvem, como:

**Especificar:** Definir os requisitos de sistemas em nuvem, como escalabilidade, disponibilidade, segurança, e desempenho. É necessário entender como a infraestrutura em nuvem pode atender a esses requisitos, aproveitando as capacidades dos serviços de nuvem,

como balanceamento de carga, armazenamento distribuído, e provisionamento automático de recursos.

**Projetar:** Criar a arquitetura do sistema, considerando as melhores práticas da computação em nuvem, como a arquitetura de microsserviços, uso de contêineres, orquestração com Kubernetes ou serviços proprietários, e a escolha entre diferentes modelos de implantação (IaaS, PaaS, SaaS).

**Implementar:** Desenvolver o sistema utilizando linguagens de programação, paradigmas, e ferramentas que são adequadas para a nuvem, como arquiteturas sem servidor, bancos de dados em nuvem, e SOA. Linguagens de programação modernas (como Python, Java e C#) contam com Software Development Kits (SDKs) que facilitam a integração com os serviços da nuvem (AWS, 2024h).

**Manter:** Gerenciar e monitorar o sistema em nuvem para garantir que ele esteja operando conforme o esperado. Isso inclui a manutenção da infraestrutura, aplicando atualizações, gerenciando quem tem acesso à aplicação, e ajustando a escalabilidade conforme necessário. Conforme o volume cresce, é essencial que administradores mantenham visibilidade sobre o desempenho de suas aplicações. Para isso, pode-se empregar um serviço de monitoramento em nuvem (IBM, 2024c).

**Avaliar:** Analisar o desempenho do sistema em nuvem, verificando se ele atende aos requisitos especificados e se continua a ser eficiente à medida que as demandas mudam e o volume cresce. Isso pode envolver a realização de testes de carga, análise de custo-benefício, e revisão da arquitetura para possíveis melhorias.

## 3.2 REFERENCIAIS CURRICULARES DA ACM/IEEE

A ACM atua desde 1960, juntamente com outras lideranças profissionais e sociedades científicas de computação para adaptar suas recomendações curriculares no cenário altamente volátil das tecnologias da computação. Conforme a área da computação continua a evoluir, e novas disciplinas relacionadas com essa área surgem, relatórios curriculares existentes são atualizados, e novos relatórios são redigidos para acomodá-las.

Dois relatórios se destacam para o curso de Ciência da Computação: o primeiro, publicado em dezembro em 2013 é um trabalho conjunto da ACM com IEEE. O segundo, publicado em maio de 2024, conta com mais uma sociedade de computação, a AAAI (Associação para o Avanço da Inteligência Artificial), mas foi publicado durante o desenvolvimento desse trabalho, e por esse motivo nos limitaremos a destacar as mudanças entre esta última versão e a anterior.

### 3.2.1 Computer Science Curricula 2013

Esse referencial curricular, também conhecido como CS2013, é dividido em 18 áreas do conhecimento (KAs), que correspondem a tópicos de estudo na área de computação

(ACM/IEEE, 2013). As áreas de conhecimento que contêm disciplinas relacionadas à computação em nuvem são:

- Computação Paralela e Distribuída (PD), por conta da própria disciplina de Computação em Nuvem;
- Fundamentos de Sistemas (SF), por conta da disciplina de Virtualização e Isolamento;
- Sistemas Operacionais (SO), por conta da disciplina de Máquinas Virtuais.

A Computação Paralela e Distribuída, há mais de dez anos atrás quando esse relatório foi publicado, já era uma área que crescia expressivamente, tanto pelo crescente uso de processadores com vários núcleos, quanto pelo surgimento de *data centers* distribuídos. Por conta desse crescimento, essa área deixou de ser eletiva para estudo, e tornou-se um componente do núcleo da formação em computação.

Esse referencial aborda disciplinas de paralelismo em duas de suas áreas do conhecimento. Dentro da área de Fundamentos de Sistema, o paralelismo é tratado de forma introdutória, com o intuito de prover ao estudante uma visão geral de como a execução simultânea se comporta em diferentes níveis de abstração. Já na área de Computação Paralela e Distribuída o foco é em entender os desafios e aplicações do uso do paralelismo e concorrência.

Dentro da área de Computação Paralela e Distribuída, a disciplina Computação em Nuvem é considerada eletiva e contempla os seguintes tópicos:

- Computação a nível de *Internet*, que trata do particionamento de tarefas para lidar com o grande volume de dados disponível na *Internet* e acesso a esses dados;
- Serviços na Nuvem, que trata da diferenciação entre os modelos IaaS e SaaS, segurança e gestão de custos;
- Virtualização, com foco no compartilhamento de recursos e migração de processos;
- Armazenamento baseado em nuvem, com ênfase no acesso compartilhado a bases de dados com consistência fraca, sincronização, sistemas de arquivos distribuídos e replicação.

Dentre os objetivos de aprendizado sugeridos para a disciplina, destacam-se:

- Discutir a importância da elasticidade e gerenciamento de recursos na computação em nuvem;
- Explicar estratégias para sincronização de dados entre dispositivos;

- Explicar as vantagens e desvantagens de usar infraestrutura virtualizada;
- Implantar uma aplicação que usa infraestrutura na nuvem para computação e/ou recursos de dados;
- Separar corretamente uma aplicação entre cliente e recursos.

A outra disciplina que nos interessa, Máquinas Virtuais, está dentro da área de Sistemas Operacionais, e embora esse assunto seja apresentado com menos detalhes, somente como pré-requisito para compreender os fundamentos da computação em nuvem, os seguintes tópicos são recomendados para serem usados como referência:

- Tipos de virtualização (hardware/software, sistemas operacionais, servidores, serviços, redes);
- Sistemas de arquivo virtuais;
- *Hipervisores*;
- Emulação portátil; emulação vs isolamento;
- Custo de virtualização.

### 3.2.2 Computer Science Curricula 2023

Em comparação com o referencial de 2013, a revisão publicada em 2023 deixou de ter o foco no que deve ser ensinado (modelo de conhecimento) e passou a ser no que deve ser aprendido (modelo de competências), tornando-se mais parecido com o referencial da SBC (ACM/IEEE/AAAI, 2024).

Seguindo essa nova abordagem, a computação em nuvem ganhou maior destaque na área de gerenciamento de dados, particularmente na unidade de armazenamento distribuído, consultas e transações. Dentro desse tópico, também foi incluído o modelo de processamento MapReduce, que consiste em distribuir o processamento em muitas máquinas para processar grandes volumes de dados em um tempo aceitável.

## 3.3 A DISCIPLINA NO CURSO DE CIÊNCIA DA COMPUTAÇÃO DA UFRJ

Na última reforma curricular do curso de Ciência da Computação foi criada a disciplina eletiva de Computação em Nuvem, com código ICP025. Ela tem como pré-requisitos as disciplinas Arquitetura de Computadores e Sistemas Operacionais e Redes de Computadores I, e período previsto de 7, 8 ou 9.

A disciplina tem como objetivo ensinar aos alunos aspectos teóricos e práticos do desenvolvimento de aplicações distribuídas segundo o modelo de computação em nuvem,

focando nos aspectos de virtualização, arquiteturas orientadas a serviços, escalonamento e interfaces de programação para computação em nuvem.

A ementa da disciplina inclui os seguintes tópicos:

- Objetivos, conceitos e definições sobre computação em nuvem;
- Abordagem geral de computação em nuvem e sua aplicabilidade para o desenvolvimento de aplicações distribuídas;
- Virtualização e migração de máquinas virtuais;
- Infraestrutura como serviço (IaaS);
- Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;
- Software como serviço (SaaS);
- Plataformas e frameworks populares na nuvem;
- Plataforma como serviço (PaaS);
- Gerenciamento e monitoração do ambiente em nuvem;
- Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem;
- Segurança e privacidade no contexto de computação em nuvem

Já a proposta inicial de programa está dividida em 4 módulos: **Terminologia e Conceitos, Virtualização e Hipervisores, Paradigmas e plataformas da computação em nuvem e Gerenciamento, monitoração e governança.**

O primeiro módulo introduz o conceito de computação em nuvem comparando-a com outras plataformas de computação distribuída, e também apresenta a arquitetura orientada a serviços (SOA). O segundo módulo é dedicado à virtualização, e se propõe a apresentar conceitos básicos, como a diferenciação entre virtualização, paravirtualização e emulação, e também o estudo e comparação entre os diferentes tipos de hipervisores (1 e 2) com exemplos. O terceiro módulo trata das plataformas e modelos da nuvem, apresentando os três maiores provedores (Amazon, Google e Microsoft) e também a alternativa *open-source*, o Eucalyptus (EUCALYPTUS, 2024). Nesse módulo, é feita a diferenciação entre os modelos de serviço (IaaS, PaaS e SaaS) passando pelo processo de construção de cada um, e com exemplos de aplicações. O quarto e último módulo é destinado a gestão da nuvem, passando por ferramentas de gerenciamento e de monitoração, segurança na nuvem e análise de desempenho.

### 3.3.1 Análise baseada no Referencial da SBC

Começando pelo primeiro eixo de formação, desenvolvimento de sistemas, destacamos a ausência da parte de princípios e boas práticas de engenharia de sistemas e de software aplicado ao contexto da nuvem. Esses princípios incluem, por exemplo, arquitetura baseada em microsserviços, escalabilidade horizontal e cache, e auxiliam na criação de aplicações confiáveis, escalonáveis e seguras na nuvem (AZURE, 2024d).

Entendemos que tanto a ementa quanto o programa já estão de acordo com o que o referencial espera quando ele cita a “utilização dos ambientes de desenvolvimento apropriados”, pois no contexto da nuvem, isso se traduz na diferenciação e escolha dos modelos de serviço, tópico que já está sendo tratado com bastante ênfase.

Por outro lado, poderia ser incluído na ementa um tópico de testes de aplicações em nuvem, que exemplificasse e diferenciasses testes de carga, desempenho, *stress* e de integração, para estar em conformidade com a parte do referencial que destaca “Testar e manter sistemas computacionais”.

Quando o referencial menciona tomada de decisões, também entendemos que a ementa atual já está consoante com o recomendado, por apresentar as diferentes soluções para hospedagem de recursos em nuvem (grandes provedores e alternativas *open-source*) e ter bastante destaque nos modelos de serviço. Essas são escolhas que o aluno deverá tomar e que envolvem aspectos éticos, legais e ambientais que o referencial cita.

Passando para o segundo eixo, gestão de infraestrutura, acreditamos que grande parte já está sendo atendida dentro da unidade 4 do programa, (Gerenciamento, monitoração e governança), que inclui ferramentas de monitoração para aplicativos em nuvem. No entanto, para deixar o programa ainda mais em conformidade com o referencial, poderia ser acrescentado um tópico que trate da implantação de aplicações em nuvem (*cloud deployment*), seguindo boas práticas como Infraestrutura como Código, que propõe a substituição de configurações e processos manuais por documentos declarativos armazenados em sistemas de controle de versão, como Git, junto ao código da aplicação (AWS, 2024j).

Dentro desse eixo, o referencial destaca a importância de “Manter a infraestrutura computacional em conformidade com a sua especificação na eventual ocorrência de alterações no seu contexto de operação”. No contexto de nuvem, essas alterações no contexto de operação se traduzem em variações nos acordos a nível de serviço (SLA), que podem envolver alterações no custo, disponibilidade e privacidade. É essencial que o aluno como desenvolvedor seja capaz de desenvolver soluções com fraco acoplamento a serviços específicos para facilitar a migração em caso de troca de um serviço por outro, e esse ponto deve ser destacado na parte do programa que cobre a arquitetura orientada a serviços (módulo 1).

No quadro 1, relacionamos as competências associadas à computação em nuvem pre-

sententes no referencial da SBC com os tópicos da ementa da disciplina que as contemplam.

Quadro 1 – Competências esperadas pelo referencial da SBC x Tópicos da ementa

Competência esperada	Tópicos que a contemplam
Identificar, analisar, especificar, validar requisitos	Abordagem geral de computação em nuvem e sua aplicabilidade para o desenvolvimento de aplicações distribuídas;
Projetar soluções computacionais em harmonia com o ambiente social e físico no seu entorno de aplicação	Segurança e privacidade no contexto de computação em nuvem; Plataformas e frameworks populares na nuvem;
Implementar sistemas computacionais utilizando ambientes de desenvolvimento apropriados	Plataformas e frameworks populares na nuvem; Infraestrutura como serviço (IaaS); Software como serviço (SaaS); Plataforma como serviço (PaaS);
Testar e manter sistemas computacionais	<b>Não há tópicos na ementa que contemplem essa competência, no entanto existe uma disciplina dedicada a testes no curso.</b>
Tomar decisões e inovar, com base no conhecimento do funcionamento e das características técnicas de hardware e da infraestrutura de software dos sistemas de computação, consciente dos aspectos éticos, legais e dos impactos ambientais decorrentes	Virtualização e migração de máquinas virtuais; Objetivos, conceitos e definições sobre computação em nuvem;
Projetar uma infraestrutura computacional a partir das especificações dos sistemas computacionais que irão compartilhar os recursos da infraestrutura e das necessidades adicionais decorrentes desse uso compartilhado	Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem; Virtualização e migração de máquinas virtuais;
Implantar a infraestrutura computacional, com domínio do processo de aquisição ou contratação de componentes de hardware e software, bem como do processo de instalação, configuração e integração desses componentes	Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;
Manter a infraestrutura computacional em conformidade com a sua especificação na eventual ocorrência de alterações no seu contexto de operação	Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem;
Especificar, projetar, implementar, manter e avaliar sistemas de computação, empregando teorias, práticas e ferramentas adequadas.	Gerenciamento e monitoração do ambiente em nuvem; Infraestrutura como serviço (IaaS); Software como serviço (SaaS); Plataforma como serviço (PaaS);

### 3.3.2 Análise baseada nos Referenciais da ACM/IEEE

Passando para os referenciais da ACM e IEEE, o currículo do BCC atende à maioria dos tópicos de estudo propostos, o que inclui computação a nível de Internet, modelos de

serviço na nuvem e virtualização. Dentre estes, apenas o tópico armazenamento baseado em nuvem não está presente na ementa.

A ausência desse tópico nessa disciplina pode ser justificado pela presença dele em outra disciplina eletiva: Sistemas Distribuídos. No entanto, a vantagem de trazê-lo aqui é que se torna possível explorar a parte prática utilizando um serviço na nuvem.

Seguindo com os objetivos de aprendizado do referencial, novamente o currículo do BCC já atende a grande maioria dos pontos, que devem fluir naturalmente com o programa apresentado, mas falha ao não contemplar conceitos de sincronização de dados, que se encontra no mesmo tópico de estudo citado anteriormente.

No quadro 2, criamos um paralelo entre os tópicos de estudos e objetivos de aprendizado descritos nos referenciais da ACM/IEEE e os tópicos abordados na ementa.

Quadro 2 – Tópicos de estudo e objetivos de aprendizado segundo o referencial da ACM/IEEE x Tópicos na ementa

<b>Tópico de estudo ou objetivo</b>	<b>Tópicos que a contemplam</b>
Computação a nível de Internet	Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;
Serviços na Nuvem	Infraestrutura como serviço (IaaS); Software como serviço (SaaS); Plataforma como serviço (PaaS); Plataformas e frameworks populares na nuvem; Segurança e privacidade no contexto de computação em nuvem; Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem;
Virtualização	Virtualização e migração de máquinas virtuais;
Armazenamento baseado em nuvem	<b>Não há tópicos na ementa que contemplem esse tópico de estudo.</b>
Discutir a importância da elasticidade e gerenciamento de recursos na computação em nuvem	Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem;
Explicar estratégias para sincronização de dados entre dispositivos	<b>Não há tópicos na ementa que contemplem esse objetivo de aprendizado.</b>
Explicar as vantagens e desvantagens de usar infraestrutura virtualizada	Virtualização e migração de máquinas virtuais;
Implantar uma aplicação que usa infraestrutura na nuvem para computação e/ou recursos de dados	Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;
Separar corretamente uma aplicação entre cliente e recursos	Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;

## 4 PROGRAMAS EXISTENTES

Dentre as universidades brasileiras que já possuem disciplinas de computação em nuvem, selecionamos a Universidade de São Paulo (campus capital e São Carlos), Universidade Federal de Ouro Preto, Universidade Federal de Minas Gerais, UNICAMP e MACKENZIE. Dentre as estrangeiras, selecionamos a Carnegie Mellon e a Stanford University.

Na maioria dessas universidades, a computação em nuvem está presente na forma de uma única disciplina que cobre todo o conteúdo — de forma similar a como ela será oferecida no curso de Ciência da Computação da UFRJ — que vai desde as definições e conceitos básicos até o desenvolvimento, monitoramento e avaliação de desempenho de uma aplicação construída para a nuvem.

No caso da UNICAMP e MACKENZIE, existem especializações dedicadas à computação em nuvem que consistem em várias disciplinas, cada uma focada em um aspecto da nuvem, o que permite tratar cada tópico contemplado de forma mais aprofundada.

### 4.1 COMPUTAÇÃO EM NUVEM COMO DISCIPLINA DE GRADUAÇÃO E PÓS GRADUAÇÃO

Nessa seção, vamos nos aprofundar sobre como as instituições levantadas abordam o tema computação em nuvem no formato de disciplina, tanto em cursos de graduação quanto pós graduação. Para isso, destacaremos o conteúdo oferecido por cada uma, e o formato de avaliação dos alunos.

#### 4.1.1 Computação em Nuvem na USP

Na Universidade de São Paulo, há duas disciplinas diferentes relacionadas a computação em nuvem: no campus da capital e no campus de São Carlos.

No campus da capital, a disciplina se chama *Computação em Nuvem e Arquitetura Orientadas a Serviços*, com código SSC0158. Ela é oferecida como disciplina optativa para o curso de Bacharelado em Ciência de Computação (USP, 2024a). Já no campus de São Carlos, ela se chama *Computação em Nuvem*, com código SSC5973, e faz parte do programa de pós graduação em Ciência da Computação e Matemática Computacional (USP, 2024b). A segunda disciplina tem muito conteúdo em comum com a primeira, no entanto apresenta alguns tópicos adicionais, por se tratar de uma disciplina de mestrado.

Ambas disciplinas tem como objetivo introduzir aspectos teóricos e práticos do desenvolvimento de aplicações distribuídas seguindo o modelo de computação em nuvem, com foco nos seguintes tópicos: aspectos de virtualização, arquiteturas orientadas a serviços, escalonamento e interfaces de programação para computação em nuvem.

A disciplina de graduação inicia abordando conceitos e definições básicas para o entendimento da computação em nuvem, e sua aplicabilidade para o desenvolvimento de aplicações distribuídas. Logo depois, seu foco passa para as arquiteturas orientadas a serviço (SOA), introduzindo o modelo arquitetural, configuração, desenvolvimento e implantação de aplicações que seguem esse modelo. Na parte de virtualização, são apresentados as características e tipos de virtualizadores, vantagens e desvantagens. Ao final, essa disciplina aborda uma parte mais prática do desenvolvimento de aplicações em nuvem, o que envolve ferramentas, modelos e técnicas para construir e monitorar ambientes na nuvem.

Já a disciplina de mestrado conta com os seguintes tópicos adicionais em relação a de graduação: carga de trabalho, segurança, infraestrutura computacional, middlewares para infraestrutura como serviços, APIs para o consumo de aplicações e testes de carga em ambiente de nuvem. O programa dessa disciplina ainda contempla discussões sobre o modelo de implantação de nuvem privado, bem como a interseção entre computação em nuvem, computação em névoa e computação de borda.

Em ambos os campus, os alunos são avaliados por meio de exercícios e trabalhos práticos executados fora de classe, além de provas sobre os assuntos do programa.

#### 4.1.2 Computação em Nuvem na UFOP

Na Universidade Federal de Ouro Preto, a disciplina se chama *Computação nas Nuvens* e tem código BCC422. Ela faz parte do curso de bacharelado em Ciência da Computação como disciplina eletiva (UFOP, 2024).

Essa disciplina inicia com uma parte conceitual, em que aborda temas como os fundamentos necessários para compreender a computação em nuvem como um todo, como terminologias e conceitos específicos desse domínio, e depois aborda virtualização. Dentro desse tópico, são tratadas as diferenças entre os tipos de hipervisores, dando como exemplo o XEN (XEN, 2024) e o QEMU.

Dentro do assunto de modelos de implantação, é abordado o Eucalyptus, que é um software *open-source* projetado para construir nuvens que podem ser tanto privadas quanto híbridas. Isso porque esse software permite mover instâncias entre sua nuvem privada e a AWS, permitindo a coexistência entre as duas (EUCALYPTUS, 2024).

Para exemplificar os tipos de computação em nuvem, são apresentadas soluções na nuvem que usam o Ruby on Rails (RAILS, 2024), como exemplo de PaaS, o roteamento definido por Software como exemplo de IaaS e a arquitetura orientada a serviço como exemplo de SaaS. Ao final, são apresentados conceitos como acordo a nível de serviço e ferramentas para análise de desempenho de aplicações na nuvem.

Não encontramos detalhamento sobre a forma como os alunos são avaliados nessa disciplina.

### 4.1.3 Computação em Nuvem na UFMG

Na Universidade Federal de Minas Gerais, a disciplina se chama *Tópicos em Redes de Computadores: Virtualização e Computação em Nuvem*. Ela é oferecida tanto para graduação quanto para pós-graduação (UFMG, 2024).

A primeira metade dessa disciplina é composta por aulas expositivas, com a introdução dos conceitos principais da área para os alunos. Já a segunda metade é conduzida pelos próprios alunos, por meio da leitura e discussão de artigos recentes selecionados pelo professor, que devem ser lidos por todos, cabendo a cada aluno submeter um pequeno resumo sobre sua visão do artigo, objetivos e resultados antes da aula.

O conteúdo da disciplina inicia com aspectos de virtualização, e em sua aplicação para ambientes na nuvem, passando por problemas de escalonamento que surgem em máquinas virtuais, e terminando com desenvolvimento de aplicações distribuídas segundo o modelo de computação em nuvem.

Os tópicos cobertos pelo programa são: tipos de virtualização (servidores, redes, desktops), ambientes de virtualização (OpenStack (OPENSTACK, 2024) e CloudStack (APACHE, 2024)), relacionamento entre virtualização e nuvens, modelos de serviço na nuvem (SaaS, PaaS, IaaS), confiabilidade, localização de recursos, consistência de dados na nuvem, datacenters verdes (eficiência energética) e desafios do IaaS.

Além dos resumos, os alunos apresentam um seminário sobre um artigo recente da literatura apresentado nos principais congressos da área nos últimos 2 ou 3 anos. Há ainda uma prova no formato de um estudo dirigido, para ser entregue pelo aluno com intervalo de alguns dias.

### 4.1.4 Computação em Nuvem na Carnegie Mellon

Na universidade de Carnegie Mellon, em Pittsburgh, Pensilvânia, a disciplina se chama *Computação em Nuvem (Cloud Computing)*, com códigos 15-319 / 15-619, e é oferecida de forma remota para alunos de graduação em Ciência da Computação e pós-graduação (MELLON, 2024a).

Esse curso conta com o apoio de concessões (*grants*) da AWS (AWS Educate), Azure (Microsoft Azure Educator Grant Award) e da Google (Google Cloud Platform). Ou seja, o provedor de serviços disponibiliza um valor para ser gasto com o uso de seus serviços na nuvem, o que isenta o aluno de ter que arcar com essa despesa.

O objetivo do curso é oferecer ao aluno uma experiência prática no ambiente da nuvem, resolvendo problemas reais que envolvem tópicos como *cloud analytics*, elasticidade, armazenamento e *frameworks*. As soluções devem ser implementadas utilizando ferramentas e serviços públicos de computação em nuvem. Nessa disciplina, os alunos tem acesso ao conteúdo teórico e posteriormente aplicam-o em projetos que envolvem todas as três principais plataformas de nuvem: AWS, Azure e GCP.

Para a parte conceitual, são cobertos os seguintes tópicos: infraestrutura na nuvem, virtualização, redes e armazenamento definido por software, armazenamento na nuvem, modelos de programação (frameworks analíticos), benefícios e dificuldades na nuvem, modelos de serviço na nuvem, acordos de nível de serviço (SLA), segurança na nuvem, Docker, Kubernetes e *serverless* (Função como Serviço).

A avaliação dos alunos é realizada por meio de projetos, que correspondem a 80% da nota, sendo os outros 20% compostos por questionários que são disponibilizados por módulo coberto da disciplina.

#### 4.1.5 Computação em Nuvem na Stanford University

Na universidade de Stanford, em Palo Alto, Califórnia, a disciplina se chama *Tecnologia de Computação em Nuvem* (Cloud Computing Technology), com código CS349d (UNIVERSITY, 2024).

O conteúdo dessa disciplina enfatiza conceitos sobre Machine Learning (MLaaS) e Deep Learning na nuvem, destacando os benefícios do processamento desses modelos usando a computação em nuvem. Para abrir o assunto de cada aula, o professor propõe a leitura de um artigo pré-selecionado relacionado ao tema.

Como forma de avaliação, os alunos desenvolvem um projeto ao decorrer das aulas que corresponde a 60% da nota, sendo outros 20% de participação e os 20% restantes compostos pela apresentação e resumo de artigos relacionados a nuvem selecionados pelo professor.

## 4.2 COMPUTAÇÃO EM NUVEM COMO ESPECIALIZAÇÃO

Nessa seção, vamos explorar como o tema computação em nuvem é tratado no formato de especialização, principalmente em como esse assunto foi dividido em disciplinas, e a profundidade com qual cada uma delas aborda os tópicos da área.

### 4.2.1 Computação em Nuvem na UNICAMP

Na Universidade Estadual de Campinas, o conteúdo de computação em nuvem é oferecido na forma de um curso de extensão com nome *Nuvens e Virtualização de Redes*, tendo como pré-requisito nível superior completo. O curso é pago e consiste em oito disciplinas, porém somente 4 delas são relevantes para esse levantamento, pois as demais são relacionadas mais especificamente com o conteúdo de redes de computadores. As disciplinas que envolvem virtualização ou computação em nuvem são: Redes Definidas por Software, Virtualização de Redes e Sistemas Computacionais e Computação em Nuvem I e II (UNICAMP, 2024).

A disciplina de *Redes Definidas por Software* tem carga horária prevista de 40 h/a, e cobre os fundamentos de redes definidas por software (SDN), virtualização de funções de redes (NFV) e fatiamento de redes (network slicing).

Na disciplina de *Virtualização de Redes e Sistemas Computacionais*, a carga horária prevista é de 27 h/a (horas de aula) e são cobertos os tópicos de redes virtualizadas, interfaces e *bridges* virtuais, tipos e sistemas de virtualização e ferramentas de gerenciamento para virtualização e containerização.

A computação em nuvem em si está dividida em duas disciplinas. A primeira disciplina, chamada *Computação em Nuvem I*, tem carga horária prevista de 18 h/a, e caráter introdutório — cobrindo a evolução da computação, modelos de serviço, virtualização de servidores e aplicações de serviços em nuvens. Já a segunda disciplina, *Computação em Nuvem II*, tem carga horária de 23 h/a, e cobre os tópicos de sistemas de armazenamento distribuído, migração de máquinas virtuais, gerenciamento e monitoramento de recursos e de tarefas computacionais.

#### 4.2.2 Computação em Nuvem na Mackenzie

Na universidade Mackenzie, o Curso de *Especialização em Computação em Nuvem* é composto de 12 disciplinas divididas em 4 módulos, totalizando 432h/a (MACKENZIE, 2024).

O módulo 1 se chama Computação em Nuvem, com carga horária estimada de 128h/a. Ele contém disciplinas que tratam dos fundamentos da computação em nuvem, dos modelos de negócio e da governança de TI na nuvem. Nesse módulo, são apresentados os modelos de implantação e de serviço, as tecnologias e padrões da indústria e o ciclo de vida de uma aplicação em nuvem, o que envolve escalabilidade, balanceamento e monitoramento. Na disciplina de Modelos de Negócio, são abordados os aspectos econômicos e estratégicos para adoção da computação em nuvem, que inclui avaliação de riscos e temas legais, fiscais e tributários. Já na disciplina de Governança de TI na Nuvem, são apresentados conceitos de controle, métricas de desempenho e *frameworks* de governança de TI para a nuvem.

O módulo 2 se chama Plataforma e Infraestrutura para Computação em Nuvem, e tem carga horária estimada de 128h/a. Ele aborda temas como virtualização, infraestrutura para nuvem, Big Data e IoT. A primeira disciplina desse módulo se chama Virtualização e Arquiteturas Dedicadas de Software e Hardware, e trata de temas como a diferença entre hardware dedicado e virtualizado, disponibilidade, confiabilidade e escalabilidade em arquiteturas virtualizadas. A segunda disciplina, Infraestrutura e Telecomunicações para Nuvem, trata dos modelos de conectividade e de rede, protocolos e arquiteturas de comunicação. As outras duas disciplinas desse módulo fogem um pouco do tema desse trabalho e não serão aprofundadas.

O módulo 3 se chama Projeto e Desenvolvimento para Nuvem, com carga horária estimada de 128h/a. Suas disciplinas envolvem temas mais práticos, como engenharia de software, DevOps, qualidade, segurança e mobilidade voltado para a nuvem.

Finalmente, o módulo 4 é o único remoto, e contém uma única disciplina prática, com carga horária de 48h/a. Ela consiste na definição e resolução de um desafio pessoal ou institucional utilizando os conceitos aprendidos ao longo da especialização.

### 4.3 AVALIAÇÃO DOS PROGRAMAS

Em todas os programas levantados, há uma parte teórica de introdução que inclui conceitos de virtualização, mas com variação na profundidade do conteúdo. Enquanto as universidades americanas se limitam a tratar somente os conceitos básicos de máquinas virtuais, como pré-requisito para a compreensão da computação em nuvem, as brasileiras se propõem a explorar mais esse assunto, o que inclui a diferenciação entre os tipos de hipervisores e exemplos de virtualizadores, algumas até com atividades práticas nesse assunto. Por outro lado, as universidades americanas trazem conceitos sobre Docker e Kubernetes, que embora não sejam propriamente tecnologias de virtualização, funcionam como uma alternativa moderna e eficiente. Nesse contexto, embora o conteúdo previsto na proposta inicial da disciplina do BCC sobre virtualização já se mostre bastante completo quando comparado com as demais instituições, seria muito interessante acrescentar na disciplina proposta para o curso de Ciência da Computação da UFRJ uma comparação com tecnologias de containerização, destacando suas diferenças, vantagens e desvantagens em relação às máquinas virtuais, principalmente pelo seu crescente uso na computação em nuvem.

Os conceitos considerados básicos para a compreensão da computação em nuvem são consistentes em todos os programas avaliados, como a diferenciação entre os modelos de implementação e a importância de tecnologias como SOA e dos SLAs para a consolidação dos modelos de computação em nuvem. Para exemplificar os diferentes modelos, de forma geral recorre-se a exemplos práticos, selecionando algum serviço em nuvem público e comparando-o a outros que seguem paradigmas diferentes. Abordaremos alguns desses serviços com mais profundidade nos laboratórios propostos.

Algumas universidades, como a UFMG e Stanford, organizam suas aulas com base na discussão de artigos relacionados a temas da computação em nuvem, que são selecionados previamente pelo professor, e devem ser obrigatoriamente lidos por todos. Os tópicos de pesquisa incluem o uso da nuvem aplicado para aprendizado de máquina, arquitetura *serverless* e frameworks de segurança na nuvem.

Em relação à parte prática dos cursos, há bastante variação nas ferramentas utilizadas. Na universidade Carnegie Mellon, os alunos desenvolvem projetos que envolvem serviços nas três principais provedoras de serviços em nuvem, como o Amazon EC2 (AWS, 2024g),

o Azure HDInsight (AZURE, 2024b) e o GCP Dataproc (GCP, 2024b). Além disso, também utilizam ferramentas que não são específicas de nenhum provedor, como o MongoDB e o Apache Kafka (MELLON, 2024b). Na UFMG, o conteúdo é ainda mais abrangente, cobrindo serviços de armazenamento específicos na AWS e na GCP, como o DynamoDB e o BigTable. E na UFOP, o diferencial é o estudo do software Eucalyptus para construção de nuvens privadas.

O quadro 1 faz uma comparação entre os conteúdos abordados por cada instituição, incluindo o programa proposto no currículo do curso de Ciência da Computação da UFRJ. Os tópicos foram agrupados levando em consideração 3 categorias: conteúdos de virtualização, conteúdos de nuvem e formas de avaliação. A escolha desses tópicos para a comparação das instituições se deu considerando os conteúdos que mais recebem destaque em cada programa avaliado.

Quadro 3 – Quadro comparativo sobre a oferta de conteúdos relacionados à computação em nuvem

INSTITUIÇÃO	VIRTUALIZAÇÃO	NUVEM	AVALIAÇÃO
USP São Paulo	Tipos de virtualizadores, vantagens, desvantagens	SOA, web Services, tipos de nuvem	Provas, trabalhos e exercícios práticos
USP São Carlos	Tipos de virtualizadores, vantagens, desvantagens	SOA, web services, implantação de aplicações, micros serviços	Provas, trabalhos e exercícios práticos
UFOP	Exemplos de virtualizadores: XEN, QEMU	Modelos e tipos de Nuvem, elasticidade, resiliência, uso medido	Sem informação.
UFMG	Tipos de virtualização, Ambientes de virtualização: OpenStack e CloudStack, migração de VMs	Tipos de nuvem, confiabilidade, localização, consistência	Participação na apresentação dos artigos, seminários e trabalhos
Carnegie Mellon	Redes e armazenamento definidos por software	MapReduce, função como serviço, modelos de serviço, SLA, segurança, Apache Spark	Questionários e projetos práticos
Stanford	Sem conteúdo.	Aprendizado de máquina na nuvem, data lakes, gerenciamento de recursos, computação sem servidor	Projetos práticos, participação e apresentação de artigos
UNICAMP	Redes virtualizadas, ferramentas de gerenciamento, tipos de virtualização, sistemas de virtualização, confinamento de processos, pilhas de redes, virtualização de redes	Evolução da computação (grades, SOA, utility computing), modelos de serviço, big data, map-reduce, web-scale arquiteturas	Varia de acordo com a disciplina. (especialização)
Mackenzie	Hardware e software dedicado x virtualizado, disponibilidade, confiabilidade, escalabilidade, conceitos de datacenter definido por software	Tecnologias e padrões de indústria, ciclo de vida, modelos de referência, escalabilidade, balanceamento de carga	Varia de acordo com a disciplina. (especialização)
UFRJ	Tipos de hipervisores, paravirtualização, emulação, estudo e comparação entre hipervisores: Xen, KVM / QEMU, VMWare, VirtualBox	Modelos de serviço e de implantação, estudo das plataformas de nuvem, softwares de gerenciamento de nuvem, ferramentas de monitoração	Sem informação.

## 5 NOSSA PROPOSTA

Nesse capítulo, vamos consolidar todo o levantamento apresentado nos capítulos anteriores de forma a construir uma análise crítica à proposta da disciplina de Computação em Nuvem do curso de Ciência da Computação da UFRJ criada a partir da reforma curricular de 2022. Ao final, vamos propor um programa mais detalhado e completo para a disciplina, tomando como base o programa atual e complementando-o conforme necessário com as discussões levantadas anteriormente.

### 5.1 ANÁLISE DA DISCIPLINA

Nessa seção analisaremos a proposta atual da disciplina levando em consideração o levantamento apresentado, passando pelos pré-requisitos, ementa, conteúdo programático e referências bibliográficas.

#### 5.1.1 Pré-requisitos

A disciplina *Computação em Nuvem* apresenta dois pré-requisitos: *Redes de Computadores I* e *Arquitetura de Computadores e Sistemas Operacionais*.

A primeira traz em sua ementa competências importantes para a compreensão da comunicação entre serviços em nuvem, como os protocolos que são amplamente utilizados na *Internet* (ex: HTTP, TCP/IP). Por exemplo, para o estudo da arquitetura orientada a serviços (SOA), é fundamental conhecer os protocolos de comunicação, como HTTP RESTful ou SOAP, que são baseados nos protocolos da camada de aplicação. Já o conhecimento sobre os protocolos da camada de rede será utilizado, por exemplo, ao configurar um IP estático a uma máquina virtual na nuvem, ou expor uma porta nessa máquina para acesso à aplicação.

A segunda disciplina é importante por conta da parte de virtualização de hardware. Embora esse tópico não esteja listado explicitamente na ementa de *Arquitetura de Computadores e Sistemas Operacionais*, ele deriva de outros conceitos que são apresentados, como gerenciamento de tarefas e de memória e sistemas de armazenamento e de arquivos.

Uma outra disciplina que está amplamente relacionada com a computação em nuvem é *Sistemas Distribuídos*. Embora não esteja listada como pré-requisito de *Computação em Nuvem*, os alunos que já a tiverem feito terão uma vantagem em relação aos demais, visto que ela explora a parte prática de muitos conceitos vistos na disciplina de *Redes de Computadores I*, e também apresenta conteúdo em comum com a área da computação em nuvem.

Para desenvolver as competências relacionadas à testagem de aplicações projetadas para execução no ambiente da nuvem, o conteúdo coberto na disciplina *Projeto de teste*

*de software* (ICP640) pode ser útil. Nessa disciplina são apresentadas técnicas e ferramentas de teste, como por exemplo o Desenvolvimento Dirigido a Testes (TDD), que podem auxiliar o aluno a projetar cenários e casos de teste que contemplem os requisitos e especificidades das aplicações em nuvem.

Dessa forma, entendemos que as disciplinas listadas como pré-requisito estão adequadas para a disciplina de *Computação em Nuvem*, e fornecem ao aluno a base teórica necessária para a compreensão dos conceitos desta disciplina. Adicionalmente, caso o aluno tenha feito a disciplina eletiva *Sistemas Distribuídos*, ele estará ainda mais preparado, por se tratar de uma disciplina cujo conteúdo apresenta interseção significativa com a computação em nuvem, além de prover ao aluno experiência prática de comunicação entre sistemas independentes, nos exercícios de laboratório e trabalhos.

### 5.1.2 Ementa

A ementa da disciplina, já apresentada na **seção 3.3**, é:

- Objetivos, conceitos e definições sobre computação em nuvem;
- Abordagem geral de computação em nuvem e sua aplicabilidade para o desenvolvimento de aplicações distribuídas;
- Virtualização e migração de máquinas virtuais;
- Infraestrutura como serviço (IaaS);
- Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services;
- Software como serviço (SaaS);
- Plataformas e frameworks populares na nuvem;
- Plataforma como serviço (PaaS);
- Gerenciamento e monitoração do ambiente em nuvem;
- Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem;
- Segurança e privacidade no contexto de computação em nuvem

Conforme discutido nos **capítulos 3 e 4**, os tópicos da ementa apresentados estão em conformidade com as competências esperadas dos egressos do curso, e são suficientemente amplos para cobrir o conteúdo proposto. Além disso, estão bastante alinhados com a

ementa proposta pelas instituições de ensino brasileiras levantadas, mostrando-se até mais completa em tópicos como virtualização.

No entanto, ao analisar o referencial da SBC, percebemos a ausência de um tópico importante, dedicado a testes de aplicações em nuvem, e que já está presente na ementa da USP. Embora não possamos alterar a ementa da disciplina, ressaltamos que existem técnicas específicas para a testagem desses tipos de aplicações, que visam garantir que estas possam lidar com o aumento do volume de usuários e com picos de uso e que talvez possa ser incluído em reformas curriculares futuras.

### 5.1.3 Conteúdo Programático

O programa inicial, conforme já descrito na **seção 3.3**, está dividido em quatro módulos: **Terminologia e Conceitos, Virtualização e Hipervisores, Paradigmas e plataformas da computação em nuvem e Gerenciamento, monitoração e governança**.

De forma geral, esse programa está bem detalhado e em conformidade com a ementa da disciplina. Os quatro módulos previstos estão bem definidos e dividem o conteúdo da disciplina em unidades que se complementam cumulativamente.

No entanto, um dos pontos que nos chamou atenção foi que a carga horário do 3<sup>o</sup> módulo é muito superior a dos demais, e por conta disso optamos por dividi-lo em dois em nossa proposta: **Paradigmas e Modelos de Computação em Nuvem e Ambientes e Plataformas de Computação em Nuvem**.

### 5.1.4 Referências Bibliográficas

A proposta da disciplina considera dois livros como bibliografia principal: *Cloud Computing: Principles and Paradigms* (BUYAYA JAMES BROBERG, 2011) e *Operating Systems - Internals and Design Principles* (capítulo 14 - virtual machines) (STALLINGS, 2017).

O primeiro deles foi publicado em 2011, e também é utilizado como livro-texto principal na disciplina equivalente da USP. Nesse sentido, embora apresente os conceitos fundamentais desse tema, a computação em nuvem ainda é uma área que está em constante expansão, e que evoluiu muito desde sua publicação. Por conta disso, alguns conceitos do livro podem ser considerados datados, e não refletir mais a realidade dos especialistas dessa área. Por exemplo, o livro não menciona a arquitetura sem servidor (*serverless*), que é um padrão amplamente utilizado hoje em dia na construção de aplicações em nuvem, se limitando aos modelos tradicionais de serviço. Assim, sugerimos que a leitura desse livro seja complementada com a apresentação de artigos mais recentes, de forma a oferecer ao aluno contato com ferramentas mais modernas da computação em nuvem.

Já o segundo livro foi publicado em 2014, e também é utilizado na disciplina de *Arquitetura de Computadores e Sistemas Operacionais*. Como não houve mudança significativa nos conceitos de virtualização, por se tratar de uma área mais consolidada, acreditamos que esse livro está adequado para cobrir essa parte da disciplina.

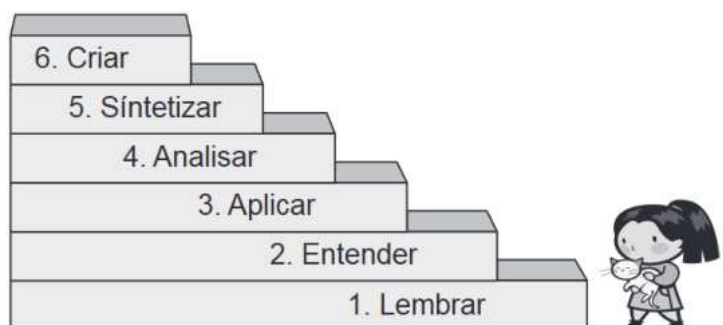
Em relação aos artigos que podem servir de bibliografia complementar aos alunos, levantamos dois que foram utilizados na *Carnegie Mellon*. O primeiro, publicado em 2023 com título *On-demand Container Loading in AWS Lambda* (BROOKER MIKE DANILOV, 2023), explica como os engenheiros da AWS conseguiram adicionar suporte para maiores pacotes de código em funções *Lambda* (FaaS), e ao mesmo tempo atender os requisitos de alta escalabilidade desse serviço. O segundo, com nome *Everywhere All at Once: Co-Location Attacks on Public Cloud FaaS* (ZHAO ADAM MORRISON, 2024), foi publicado em 2024, e é mais voltado para a área de segurança, abordando riscos e técnicas de prevenção para ataques em nuvens públicas que oferecem serviços FaaS.

## 5.2 PROGRAMA PROPOSTO

Nessa seção, vamos propor um programa mais detalhado para a disciplina de *Computação em Nuvem*. Para isso, vamos dividi-lo em módulos de forma similar ao programa inicial, e utilizaremos a taxonomia de Bloom (FERRAZ, 2010) para listar as competências esperadas dos alunos após cada módulo.

A Taxonomia de Bloom é um instrumento que auxilia na identificação e na declaração de objetivos educacionais de forma a facilitar o planejamento do processo de ensino e aprendizagem. A definição clara de objetivos instrucionais direciona o processo de ensino para a escolha adequada de estratégias, métodos, instrumentos de avaliação e, consequentemente, para uma aprendizagem efetiva e duradoura. A figura 8 exibe a versão mais atual das categorias do domínio cognitivo, como descritas em sua revisão de 2001 da Taxonomia de Bloom.

Figura 8 – Taxonomia de Bloom



Fonte: (FERRAZ, 2010)

Nessa seção, também apresentaremos um breve resumo de cada laboratório no momento em que ele será apresentado aos alunos, mas o roteiro completo estará descrito nos apêndices. Embora os laboratórios propostos nesse trabalho levem em consideração a plataforma da AWS, a escolha do provedor de nuvem pelo professor deve levar em conta diversos fatores, como o custo de utilização, a quantidade de crédito oferecida, e a qualidade dos serviços prestados. Para facilitar essa escolha, os roteiros para cada laboratório ou trabalho serão descritos utilizando termos mais genéricos e apontando os serviços equivalentes nas demais plataformas (sempre que possível), facilitando assim a troca da plataforma caso seja necessário. Os códigos utilizados nas atividades estão disponíveis no GitHub (UZIEL, 2025).

### 5.2.1 Módulo 1 - Terminologia e Conceitos

O objetivo deste módulo é fazer uma introdução a conceitos e termos relacionados a computação em nuvem, destacando diferenças e similaridades com outras plataformas de computação distribuída. Também deve contemplar uma parte dedicada a *Internet*, lembrando conceitos de redes como o protocolo HTTP e apresentando o padrão *REST* como base para a arquitetura orientada a serviços (SOA). Esse módulo se estenderá por 5 aulas (10 horas), sendo as 4 primeiras teóricas e a última prática.

Ao final deste módulo, os discentes deverão ser capazes de:

**Lembrar** conceitos e termos fundamentais da computação em nuvem, como tipos de serviços (IaaS, PaaS, SaaS) e modelos de implantação (nuvem pública, privada, híbrida), reconhecendo-os em aplicações que utilizam a computação em nuvem.

**Entender** as diferenças entre os diferentes níveis de abstração e responsabilidades de serviços em nuvem, interpretando-os como uma arquitetura em camadas.

**Aplicar**, de forma consciente, os princípios da arquitetura orientada a serviços (SOA), implementando a comunicação entre serviços por meio de APIs REST.

**Analisar** as diferenças e similaridades da computação em nuvem com outras plataformas de computação distribuída, diferenciando-as de acordo com características como escalabilidade, flexibilidade, modelo de custo, e gerenciamento de recursos.

**Avaliar** as implicações do uso dos diferentes modelos de serviço e de implantação para atender a necessidades específicas de uma organização, estimando o impacto na manutenção, privacidade e na segurança.

O quadro 4 apresenta um mapeamento dos tópicos da ementa com tópicos mais detalhados do conteúdo, e as aulas do módulo em que devem ser estudados.

A primeira aula da disciplina deve conter um breve histórico da computação, começando pela popularização dos computadores pessoais nos anos 80, da *Internet* nos anos 90, e a difusão da computação distribuída nos anos 2000. A partir daí devem ser citadas algumas motivações para o estudo da computação em nuvem, como o aumento anual do

Quadro 4 – Tópicos cobertos no módulo 1

Tópico da ementa	Tópicos detalhados	Aulas
Objetivos, conceitos e definições sobre computação em nuvem	Modelos de serviço e de implantação; Plataformas de computação em nuvem; Exemplos de serviços em nuvem; Boas práticas e padrões;	1 e 2
Abordagem geral de computação em nuvem e sua aplicabilidade para o desenvolvimento de aplicações distribuídas	Evolução da computação; Tecnologias que viabilizaram a adoção da computação a nuvem; Comparação da computação em nuvem com outras plataformas de computação distribuída;	1 e 2
Introdução ao modelo arquitetural, instalação, configuração e desenvolvimento de aplicações orientadas a serviços: arquitetura orientada a serviços (SOA) e web-services	Revisão de conceitos de <i>Redes</i> , como protocolos da camada de aplicação; Introdução à arquitetura REST; Desenvolvimento de aplicações seguindo os paradigmas da arquitetura orientada a serviços	3 e 4

volume de informação produzida, o processamento de dados científicos e o volume de dados gerado por dispositivos IoT.

Na segunda aula, devem ser mencionados temas que serão abordados com mais profundidade em módulos posteriores, como as diferentes plataformas de nuvem, alguns exemplos de serviços em nuvens, e os diferentes modelos de serviço e de implantação. É interessante que já nesse momento o professor apresente exemplos em alguma plataforma de nuvem para demonstrar como esses serviços são oferecidos, e também para oferecer aos alunos um primeiro contato com a computação em nuvem.

A terceira aula deve ser mais voltada à evolução de serviços da web, destacando como o protocolo HTTP e a linguagem HTML se tornaram um padrão global para troca de dados na *Internet*. Nesse momento, devemos citar a evolução das tecnologias da web, desde a abstração de chamada remota de procedimento (RPC) até o surgimento dos *Web Services*, como solução para os problemas de integração entre sistemas diferentes.

Já na quarta aula, continuando com o tema de arquitetura orientada a serviços, será apresentado os serviços RESTful (*REpresentational State Transfer*), que é o estilo arquitetural mais popular para a construção de APIs da atualidade. Deve ser destacada a importância que essa arquitetura teve para a padronização da comunicação em redes complexas como a *Internet*, destacando regras como a interface uniforme, a ausência de estado e a capacidade de armazenamento em cache (AWS, 2024l).

Na quinta aula, haverá o primeiro laboratório da disciplina. Nesse laboratório, o aluno deverá criar uma máquina virtual dentro de um serviço IaaS de uma plataforma em nuvem, e desenvolver uma aplicação simples de “*hello world*”, de forma similar a como é feito em outras disciplinas do curso. A plataforma escolhida para esse laboratório foi a AWS, e o serviço é o EC2 (Elastic Container Service) (AWS, 2024g). O roteiro completo desse laboratório está descrito no **Apêndice A**.

### 5.2.2 Módulo 2 - Virtualização e Hipervisores

O objetivo desse módulo é abordar o tema de virtualização, e destacar sua importância para a computação em nuvem. Deverão ser apresentados conceitos básicos nessa área, o que inclui máquinas virtuais, os tipos de *hipervisores*, e a diferença entre virtualização, paravirtualização e emulação. Além disso, devem ser cobertos exemplos de *hipervisores* de cada tipo, como o Xen, KVM, VMWare e o VirtualBox. O módulo se estenderá por 4 aulas teóricas (8 horas).

Ao final deste módulo, os discentes deverão ser capazes de:

**Lembrar** as definições de termos específicos de virtualização e suas diferenças, reproduzindo-as tanto na instalação de máquinas virtuais locais quanto na configuração de aplicações em nuvem, principalmente as que seguem o paradigma IaaS.

**Entender** a importância das técnicas de virtualização para a computação em nuvem, explicando a tecnologia subjacente que está sendo utilizada em serviços em nuvem.

**Analisar** os diferentes tipos e exemplos de *hipervisores*, organizando-os de acordo a camada em que atuam na máquina hospedeira e entendendo em quais circunstâncias cada um deles é mais adequado.

**Avaliar** a necessidade da migração de máquinas virtuais entre servidores ou para a nuvem, estimando os custos associados a curto e longo prazo.

No quadro 5 fazemos um paralelo dos tópicos da ementa com tópicos mais detalhados do conteúdo, e as aulas do módulo em que devem ser estudados.

Quadro 5 – Tópicos cobertos no módulo 2

Tópico da ementa	Tópicos detalhados	Aulas
Virtualização e migração de máquinas virtuais	Motivações para o uso da virtualização: compatibilidade, isolamento, versatilidade; Definições: máquina virtual, máquina hospedeira e máquina convidada; Diferenciação entre virtualização, paravirtualização e emulação; Migração de máquinas virtuais;	1 e 2
—	O que são <i>hipervisores</i> ; Tipos de <i>hipervisores</i> ; Exemplos de <i>hipervisores</i> : Xen, KVM, VMWare, Virtualbox;	3
—	Definições sobre contêineres; Diferenças e similaridades entre contêineres e máquinas virtuais;	4

Para introduzir o assunto de virtualização, na primeira aula desse módulo devemos abordar a problemática do uso compartilhado de uma máquina física por vários programas — prática também conhecida como multiprogramação. Embora essa técnica seja ideal para computadores pessoais, caso desejássemos particionar nossos recursos para uso de terceiros — como é o caso das provedoras de serviços em nuvem — logo encontraríamos problemas de segurança, privacidade e compatibilidade.

Como solução para esse problemas, devem ser apresentados os conceitos de virtualização, paravirtualização e emulação, suas vantagens e desvantagens. Em seguida, explicamos o que é uma máquina virtual, como máquinas hospedeiras se diferem das máquinas convidadas, e como elas facilitam a migração de recursos entre servidores.

A terceira aula será voltada para o estudo dos *hipervisores*, seus tipos e as diferenças entre eles. A instalação de *hipervisores* do tipo 1 (*bare-metal*) é bem mais complexa, pois é feita a nível de hardware, sendo necessário gravar sua imagem em um dispositivo removível (como um CD/DVD ou pen-drive) e iniciar o computador a partir dele (XEN, 2024). Por outro lado, os *hipervisores* do tipo 2 podem ser instalados como qualquer outro programa, em cima do sistema operacional. Dessa forma, é possível realizar uma demonstração utilizando o VirtualBox ou o VMWare, instalando uma distribuição de Linux em cima de outra.

Para finalizar o assunto, na última aula deve-se abordar a diferença entre contêineres e máquinas virtuais. Esse assunto é importante nessa disciplina pois tecnologias relacionadas a contêineres ganharam muita popularidade recentemente, principalmente quando combinadas a serviços de computação em nuvem, como o CaaS (IBM, 2024d). O segundo livro texto da disciplina, *Arquitetura de Computadores e Sistemas Operacionais* (STALLINGS, 2017) já trata desse tópico no mesmo capítulo de máquinas virtuais. Desta forma, não é necessário a inclusão de material didático adicional para tratar esse tema.

### 5.2.3 Módulo 3 - Paradigmas e Modelos de Computação em Nuvem

Esse módulo será dedicado ao estudo dos diferentes paradigmas da computação em nuvem, o que inclui os modelos de serviço e de implantação. Cada modelo de serviço (IaaS, PaaS, SaaS, XaaS) será explorado a fundo, passando pelo seu processo de construção, implantação e exemplos. O módulo se estenderá por 7 aulas, sendo 6 teóricas e uma prática, totalizando 14 horas.

No programa inicial da disciplina, esse módulo e o próximo formavam um só. Todavia nesta proposta, optamos por dividi-lo em dois, dado que sua carga horária era muito alta e se destoava significativamente quando comparada aos demais módulos.

Ao final deste módulo, os discentes deverão ser capazes de:

**Lembrar** a definição teórica de cada modelo, reconhecendo-os em serviços oferecidos por provedoras de computação em nuvem.

**Entender** as diferentes estratégias para o desenvolvimento de aplicações em nuvem, exemplificando como cada uma delas pode levar a solução desejada, listando vantagens e desvantagens.

**Aplicar**, de forma consciente, os paradigmas escolhidos para solucionar o problema proposto, implementando uma aplicação que utilize o serviço escolhido.

**Analisar** as diferentes abordagens para a solução de problemas utilizando a computação em nuvem, atribuindo critérios de avaliação como custo, desempenho, escalabilidade

e segurança e concluindo as circunstâncias que as tornam mais adequadas.

**Avaliar** os resultados obtidos por meio do emprego de cada abordagem, estimando o impacto a curto e longo prazo no custo, manutenção e eficiência.

**Criar** soluções inovadoras utilizando diferentes paradigmas, integrando serviços e plataformas de forma criativa e projetando arquiteturas que atendam às necessidades específicas de cada aplicação.

No quadro 6 fazemos um paralelo dos tópicos da ementa com tópicos mais detalhados do conteúdo, e as aulas do módulo em que devem ser estudados.

Quadro 6 – Tópicos cobertos no módulo 3

Tópico da ementa	Tópicos detalhados	Aulas
Infraestrutura como serviço (IaaS)	Processo de construção e implantação do IaaS; Exemplos de serviços IaaS; Vantagens e desvantagens em relação a outros modelos; Casos de uso do IaaS;	1 e 2
Plataforma como serviço (PaaS)	Processo de construção e implantação do PaaS; Especialização do PaaS em serviços XaaS; Exemplos de serviços PaaS e XaaS; Casos de uso do PaaS e XaaS;	1, 3, 5 e 6
Software como serviço (SaaS)	Casos de Uso do SaaS; Exemplos de serviços SaaS; Modelo de negócio do Saas;	1 e 4

A primeira aula desse módulo deverá servir como uma recapitulação das definições de cada modelo, que já terão sido apresentadas no módulo de *Terminologias e Conceitos*, mas dessa vez com mais profundidade. O objetivo dessa aula é que fique clara a distinção entre cada modelo, como pré-requisito para a compreensão das próximas, o que pode ser atingido por meio de discussões e exemplos. A partir daí, vamos apresentar cada modelo em uma aula separada, em ordem crescente de abstração.

Na aula de IaaS (Infraestrutura como Serviço), apresentaremos o modelo que oferece maior flexibilidade para o desenvolvedor. Nesse momento da disciplina, os alunos já terão concluído o primeiro laboratório, que envolveu o uso de um serviço IaaS para a criação de máquinas virtuais, o que pode ser usado como introdução para o assunto. É importante ressaltar ao aluno que, por oferecer maior controle ao programador, esse modelo também é o mais trabalhoso de se configurar, o que inclui aspectos como poder de processamento, armazenamento e rede.

A terceira aula será dedicada ao PaaS (Plataforma como Serviço), modelo que os alunos ainda não tiveram contato dentro da disciplina, mas que talvez já tenham utilizado em alguma outra disciplina, por meio de serviços como o *Heroku* (HEROKU, 2024). Nessa aula, é importante destacar que serviços dessa categoria abstraem muitas das complexi-

dades associadas ao IaaS, no entanto são menos flexíveis e mais específicos, permitindo foco no desenvolvimento e implantação rápida de aplicações.

Na aula de SaaS (Software como Serviço), discutiremos sobre como as aplicações oferecidas diretamente no navegador tomaram o lugar de softwares tradicionalmente instalados na máquina do usuário, citando aspectos como praticidade, acessibilidade e compatibilidade. Embora na maioria das vezes aplicações que seguem esse modelo de serviço sejam destinadas ao usuário final, e não a desenvolvedores de software, é essencial que alunos de computação conheçam seu modelo de negócio, normalmente mantido por assinatura.

Na aula de XaaS (Tudo como Serviço), falaremos das demandas que levaram ao surgimento de novos modelos de serviço mais especializados, que fogem à classificação tradicional. Para isso, podemos citar o exemplo de instituições que estão migrando sua infraestrutura existente para a nuvem, e querem utilizar somente alguns recursos de computação em nuvem, como banco de dados ou mensageria, e manter o processamento do seu lado (*on premises*). Como exemplo de serviços que se encaixam no XaaS, podem ser citados o CaaS (Contêiner como Serviço) e o FaaS (Função como Serviço), que também será explorado mais a fundo em um dos laboratórios.

Para o segundo laboratório da disciplina, optamos por um serviço de banco de dados em nuvem. Se consideramos somente a classificação tradicional dos modelos em nuvem, esse serviço seria classificado como um PaaS, mas se levamos em conta as classificações mais específicas do XaaS, é mais apropriado classificá-lo como um DBaaS (Database as a Service) (IBM, 2024a). Nesse laboratório, o objetivo do aluno é interagir com uma tabela desse serviço a partir de um código rodando localmente em sua máquina, no entanto esse processo envolve aspectos mais complexos, como a autenticação na máquina local com as credenciais da AWS e o uso de uma SDK (*Software Development Kit*) (AWS, 2024h) para facilitar a comunicação com o serviço a partir de uma linguagem de programação. O roteiro completo desse laboratório está descrito no **Apêndice B**.

Para fechar o módulo, devem ser apresentados os diferentes modelos de implantação da computação em nuvem (nuvens públicas, privadas e híbridas). Devemos enfatizar suas diferenças e benefícios, o que envolve questões como segurança, flexibilidade e controle. Por fim, será estudado o software *Eucalyptus*, que permite construir nuvens privadas e híbridas dinamicamente.

#### 5.2.4 Módulo 4 - Ambientes e Plataformas de Computação em Nuvem

Esse módulo abordará as diferentes plataformas de computação em nuvem, o que inclui vantagens e desvantagens de cada provedora, serviços oferecidos e zonas de disponibilidade. Serão estudadas com mais detalhes as três maiores provedoras (AWS, Azure e Google Cloud), mas também podem ser citadas algumas menores, como a IBM Cloud e o Heroku. Esse módulo será composto por 5 aulas, sendo 4 teóricas e uma prática, totalizando 10 horas.

Ao final deste módulo, os discentes deverão ser capazes de:

**Lembrar** as diferentes opções de provedoras e seus principais serviços de computação em nuvem disponíveis para o público, reproduzindo-os na formulação de soluções técnicas para problemas propostos.

**Entender** as diferenças entre as plataformas de computação em nuvem, comparando-as entre si em termos de desempenho, custo, variedade de serviços disponíveis e zonas de disponibilidade.

**Analisar** as vantagens e desvantagens que cada plataforma oferece em seus serviços, concluindo quais delas melhor atendem às necessidades do problema que está sendo solucionado.

**Avaliar** como o uso de cada plataforma impacta na solução desenvolvida, criticando aspectos como modelo de precificação, comprometimento com disponibilidade (por meio de *SLAs*) e facilidade de integração entre serviços.

No quadro 7 fazemos um paralelo dos tópicos da ementa com tópicos mais detalhados do conteúdo, e as aulas do módulo em que devem ser estudados.

Quadro 7 – Tópicos cobertos no módulo 4

Tópico da ementa	Tópicos detalhados	Aulas
Plataformas e frameworks populares na nuvem	AWS; Microsoft Azure; Google Cloud; Heroku; Eucalyptus; Principais serviços oferecidos por cada plataforma; Modelo de precificação; Zonas de disponibilidade;	1 e 2
Desafios e soluções que envolvem o escalonamento, provisionamento e migração de recursos na nuvem	Segurança e conformidade de dados; Integração e suporte com sistemas legados; Gestão de custos e orçamento na nuvem; Escolha do provedor de serviços em nuvem;	1, 3 e 5

Na primeira aula serão discutidos aspectos mais gerais, como popularidade no mercado, público alvo (se focam mais em empresas grades, médias ou desenvolvedores individuais) e modelos de cobrança. Também serão examinados os diferenciais de cada provedora, como a flexibilidade em escalabilidade, o suporte oferecido e os serviços que mais se destacam. O objetivo é fornecer uma visão inicial de como essas plataformas se posicionam no mercado e em quais situações cada uma delas pode ser a melhor escolha para diferentes perfis de usuários.

Na segunda aula, devem ser apresentados detalhes mais específicos de cada plataforma, como o histórico de cada uma no mercado, recursos de segurança e o alcance de suas redes globais, incluindo o número de zonas de disponibilidade e regiões atendidas. Serão explorados em mais detalhes os principais serviços oferecidos por cada uma delas em áreas como processamento, armazenamento e aprendizado de máquina, para entendermos melhor onde cada plataforma se destaca.

No terceiro laboratório da disciplina, o aluno novamente terá contato com um serviço PaaS, mas que oferece um maior grau de flexibilidade em comparação com o do último

laboratório. Isto porque esse serviço permite que o desenvolvedor execute seu próprio código na nuvem, de forma similar ao IaaS, mas sem a complexidade de gerenciar uma máquina virtual. Faremos uso desse serviço para processar um *input* e persistir o resultado no banco de dados utilizado no laboratório anterior. O roteiro completo desse laboratório está descrito no **Apêndice C**.

### 5.2.5 Módulo 5 - Gerenciamento, Monitoração e Governança

O último módulo da disciplina será dedicado ao estudo de ferramentas de administração na nuvem, o que inclui softwares de gerenciamento de nuvem, como *OpenStack* (OPENSTACK, 2024) e *CloudStack* (APACHE, 2024) e ferramentas de monitoração. Também estudaremos de forma mais aprofundada sobre os acordos a nível de serviço (SLA) e abordaremos aspectos como segurança de dados, análise de desempenho na nuvem e testagem de aplicações em nuvem. O módulo será composto por 3 aulas teóricas e duas aulas de apresentação de trabalho, totalizando 10 horas.

Ao final deste módulo, os discentes deverão ser capazes de:

**Lembrar** as diferentes ferramentas de gerenciamento e monitoração de aplicações de computação em nuvem, reconhecendo-as pelo seu propósito e escopo de atuação.

**Entender** os principais conceitos de Acordos a Nível de Serviço (*SLAs*), interpretando suas principais cláusulas, como disponibilidade, tempo de resposta e penalidades para prever possíveis impactos no funcionamento de aplicações em nuvem.

**Aplicar** ferramentas de monitoração e gerenciamento de nuvem, executando diagnósticos que identifiquem falhas e problemas de desempenho em aplicações hospedadas no ambiente de nuvem.

**Analisar** os dados coletados pelas ferramentas de monitoração de nuvem, diferenciando métricas de desempenho para identificar padrões que indiquem problemas ou oportunidades de melhoria.

**Avaliar** os termos e condições de *SLAs* de diferentes provedoras de computação em nuvem, checando requisitos de segurança, desempenho e confiabilidade e julgando se atendem às necessidades da aplicação.

No quadro 8 fazemos um paralelo dos tópicos da ementa com tópicos mais detalhados do conteúdo, e as aulas do módulo em que devem ser estudados.

Na primeira aula, o professor deve explorar os conceitos básicos de administração de nuvem, começando com uma visão geral sobre as funções e recursos oferecidos pelo softwares *OpenStack* e *CloudStack*. Deve ser discutido como essas ferramentas se diferenciam das soluções de nuvem pública, entendendo os processos de orquestração, automação e monitoração nessas plataformas.

Na segunda aula, o foco será nas práticas de monitoramento e gerenciamento contínuo em ambientes de nuvem. Serão introduzidas ferramentas de monitoramento que ajudam a rastrear o uso de recursos, o desempenho das aplicações e a detectar possíveis falhas. Para

Quadro 8 – Tópicos cobertos no módulo 5

Tópico da ementa	Tópicos detalhados	Aulas
Gerenciamento e monitoração do ambiente em nuvem	<i>OpenStack</i> ; <i>CloudStack</i> ; Amazon Cloudwatch; Gerenciamento de logs e alertas; Acordos a nível de serviço ( <i>SLA</i> ); Testagem de aplicações em nuvem;	1 e 2
Segurança e privacidade no contexto de computação em nuvem	Gestão de identidade e acesso; Proteção de dados sensíveis; Criptografia;	1, 3 e 5

isso, serão utilizadas técnicas de testagem de carga e *stress*, com o objetivo de entender o comportamento das aplicações sob diferentes condições e melhorar sua resiliência.

Na terceira aula o professor entrará em detalhes sobre acordos de nível de serviço (*SLA*). Serão estudadas cláusulas comuns dos *SLAs*, como métricas de desempenho, tempos de resposta e disponibilidade, e como esses acordos impactam a experiência do usuário final.

Para o trabalho final da disciplina, os alunos deverão desenvolver uma aplicação com tema livre utilizando serviços em nuvem. Para isso, poderão fazer uso dos serviços utilizados nos últimos laboratórios, ou incorporar algum que não foi utilizado. Esse trabalho também contará com uma apresentação final de cada grupo, o que deve levar em torno de duas aulas. O roteiro do trabalho final e um exemplo de aplicação pode ser encontrado no **Apêndice D**.

### 5.2.6 Preparação para o período

Antes do período letivo, recomenda-se que o professor inscreva a turma no programa de educação da plataforma de nuvem escolhida; Todas as três grandes provedoras (*AWS*, *Azure* e *GCP*) apresentam programas similares, e disponibilizam crédito dentro da plataforma para os alunos realizarem laboratórios e se familiarizarem com a plataforma. Esse crédito viabiliza a parte prática do curso pois, embora nem todos os serviços na plataforma sejam pagos até certa cota, é necessário inserir um cartão de crédito para caso ela seja atingida.

### 5.2.7 Proposta de avaliação

Como forma de avaliação dos alunos, propomos que a média final seja composta pelos laboratórios e pelo trabalho final da disciplina. Para os laboratórios, o aluno deverá submeter um relatório da atividade realizada e um repositório com os códigos desenvolvidos, se houver. Já o trabalho final contará também com a apresentação dos grupos, e deverá levar em conta aspectos como a complexidade do projeto desenvolvido, o uso correto dos serviços em nuvem escolhidos e a resiliência da aplicação em caso de falhas.

## 6 CONCLUSÃO

Este trabalho se propôs a fazer uma análise crítica e construir um programa mais detalhado para a disciplina de *Computação em Nuvem* do curso de Ciência da Computação da UFRJ, levando em conta a ementa aprovada na reforma curricular de 2022. Para isso, revisitamos os referenciais curriculares da SBC e da ACM/IEEE, e levantamos as ementas e programas de outras instituições, brasileiras e estrangeiras, que já contam com disciplinas equivalentes em sua grade curricular. Além disso, propomos roteiros para os laboratórios e trabalhos práticos da disciplina, com o objetivo de alinhar as atividades práticas aos conceitos teóricos abordados no programa proposto, proporcionando uma formação mais completa e preparando os alunos para os desafios na área de computação em nuvem.

A computação em nuvem é um conjunto de tecnologias que ganhou muita adesão desde que começou a ser oferecida comercialmente em 2006, sendo utilizada por praticantes das mais diversas áreas, sejam elas envolvidas diretamente com computação ou não. A inclusão de uma disciplina dedicada a esse assunto no currículo do curso de Ciência da Computação da UFRJ beneficiará o aluno, pois permitirá que ele adquira experiência prática com tecnologias amplamente utilizadas no mercado, e desenvolva competências necessárias para a construção de aplicações modernas e eficientes.

Ao revisar os referenciais curriculares, percebemos que a ementa da disciplina está suficientemente ampla para cobrir a grande maioria das competências curriculares recomendadas, com algumas exceções pontuais. Para esses casos, destacamos a importância de incluí-las na ementa da disciplina em reformas futuras, mas não modificando-a neste trabalho.

Durante o levantamento sobre como esse assunto é oferecido em outras instituições, concluímos que a ementa da disciplina no nosso curso está bastante alinhada com as demais universidades brasileiras, inclusive se demonstrando mais completa em vários aspectos. No entanto, ao compará-la com as instituições estrangeiras levantadas, percebemos diferenças significativas no conteúdo apresentado, pois estas não se aprofundam tanto em temas como virtualização, enquanto que abordam contêineres e frameworks (aspectos menos teóricos e mais práticos).

Por fim, analisamos os pré-requisitos, ementa e referências bibliográficas da disciplina e construímos um programa mais detalhado a ser utilizado em sua primeira oferta, que levou em conta a ementa aprovada, a proposta inicial e os pontos levantados nos capítulos anteriores. Os roteiros de laboratório apresentados poderão ser utilizados como base para a construção de novas atividades práticas, mesmo que o professor não opte por utilizar a mesma plataforma utilizada nesse trabalho (AWS).

Dado que ainda não foi possível avaliar o programa elaborado nesse trabalho na prá-

tica — uma vez que a disciplina ainda não foi oferecida no curso — sugerimos que sejam preenchidos formulários de avaliação pelos alunos e pelo professor ao final dos períodos em que ela for oferecida. Esse formulário terá como objetivo avaliar se o conteúdo apresentado está sendo oferecido de forma didática, se as tecnologias apresentadas ainda estão condizentes com as demandas atuais da computação em nuvem, por se tratar de uma área da computação ainda em expansão, e propor melhorias para o próximos períodos.

## 6.1 TRABALHOS FUTUROS

Ao utilizar soluções de nuvem pública, é necessário levantar questões éticas relacionadas à privacidade e às liberdades individuais, principalmente quando a computação em nuvem é utilizada para treinar modelos de inteligência artificial. Outra preocupação é com a segurança de dados: dado que servidores terceirizados estão sendo utilizados para processar e armazenar bases possivelmente sensíveis, é essencial manter controle estrito sobre quem tem acesso a eles, e selecionar provedoras com políticas de privacidade rigorosas. Dessa forma, propomos que esses tópicos sejam discutidos antes do oferecimento da disciplina, para definir se são cabíveis no programa, e com qual profundidade devem ser abordados.

Uma boa prática em aplicações desenvolvidas para nuvem é manter a configuração da infraestrutura versionada em um sistema de controle de versão como *GIT*, junto ao código. Esses arquivos de configuração normalmente são encontrados no formato de um template *JSON* ou *YAML*, e permitem que itens de infraestrutura da aplicação, como filas de mensagens ou bancos de dados, sejam rapidamente restaurados de forma precisa, da mesma forma como é feito com o código. Embora esse assunto não esteja previsto no programa, é essencial estudar a viabilidade de incluí-lo, por contribuir para a automação e a reprodutibilidade do ambiente.

Outro tema fundamental que deve ser considerado para esta disciplina é a distinção entre processamento síncrono e assíncrono. Enquanto o processamento síncrono na nuvem ocorre por meio de APIs REST ou SOAP, o assíncrono é frequentemente implementado por meio de filas de mensagens. É essencial que o aluno desenvolva a capacidade de identificar operações que não exigem resposta imediata ao usuário, permitindo assim um sistema mais responsivo e resiliente. Além disso, aspectos como volume, escalabilidade e custo devem ser cuidadosamente considerados já na fase de desenvolvimento, garantindo que a arquitetura escolhida suporte variações de carga de forma eficiente, otimize o uso de recursos e mantenha a viabilidade da aplicação a longo prazo.

## REFERÊNCIAS

ACM/IEEE. **Computer Science Curricula 2013**. 1. ed. ACM/IEEE, 2013. Disponível em: [https://www.acm.org/binaries/content/assets/education/cs2013\\_web\\_final.pdf](https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf).

ACM/IEEE/AAAI. **Computer Science Curricula 2023**. 1. ed. ACM/IEEE, 2024. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/3664191>.

APACHE. **Apache CloudStack**. 2024. <https://cloudstack.apache.org/>. Accessed: 2024.

AWS. **Amazon Elastic Container Service**. 2024. <https://aws.amazon.com/pt/ecs/>. Accessed: 2024.

AWS. **Amazon Elastic Kubernetes Service**. 2024. <https://aws.amazon.com/pt/eks/>. Accessed: 2024.

AWS. **AWS Elastic Beanstalk**. 2024. <https://aws.amazon.com/pt/elasticbeanstalk/>. Accessed: 2024.

AWS. **AWS Lambda**. 2024. <https://aws.amazon.com/pt/lambda/>. Accessed: 2024.

AWS. **AWS Lambda Functions Powered by AWS Graviton2 Processor – Run Your Functions on Arm and Get Up to 34x Performance Improvement**. Accessed: 2024.

AWS. **Boto3 documentation**. 2024. <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/dynamodb.html>. Accessed: 2024.

AWS. **Elastic Compute Cloud - Amazon EC2 - AWS**. 2024. <https://aws.amazon.com/pt/ec2/>. Accessed: 2024.

AWS. **Ferramentas para criar com a AWS**. 2024. <https://aws.amazon.com/pt/developer/tools/>. Accessed: 2024.

AWS. **O que é a computação em nuvem?** 2024. <https://aws.amazon.com/pt/what-is-cloud-computing/>. Accessed: 2024.

AWS. **O que é infraestrutura como código?** 2024. <https://aws.amazon.com/pt/what-is/iac/>. Accessed: 2024.

AWS. **O que é SOA (arquitetura orientada a serviços)?** 2024. <https://aws.amazon.com/pt/what-is/service-oriented-architecture/>. Accessed: 2024.

AWS. **O que é uma API RESTful?** 2024. <https://aws.amazon.com/pt/what-is/restful-api/>. Accessed: 2024.

AWS. **O que é virtualização?** 2024. <https://aws.amazon.com/pt/what-is/virtualization/>. Accessed: 2024.

AWS. **Qual a diferença entre contêineres e máquinas virtuais?** 2024. <https://aws.amazon.com/pt/compare/the-difference-between-containers-and-virtual-machines/#:~:text=Cont%C3%AAsineres%2C%20por%20outro%20lado%2C%20usam,de%20c%C3%B3digo%20aberto%20mais%20popular>. Accessed: 2024.

- AZURE. **AWS Lambda**. 2024. <https://azure.microsoft.com/pt-br/products/functions>. Accessed: 2024.
- AZURE. **Azure HDInsight**. 2024. <https://azure.microsoft.com/pt-br/products/hdinsight>. Accessed: 2024.
- AZURE. **Instâncias de contêiner**. 2024. <https://azure.microsoft.com/pt-br/products/container-instances>. Accessed: 2024.
- AZURE. **Melhores práticas para aplicativos de nuvem**. 2024. <https://learn.microsoft.com/pt-br/azure/architecture/best-practices/index-best-practices>. Accessed: 2024.
- AZURE. **Máquinas Virtuais do Azure**. 2024. <https://azure.microsoft.com/pt-br/products/virtual-machines>. Accessed: 2024.
- AZURE. **O que é um contêiner?** 2024. <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-container>. Accessed: 2024.
- AZURE, M. **O que é computação em nuvem?** 2024. <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-cloud-computing>. Accessed: 2024.
- BROOKER MIKE DANILOV, C. G. P. P. M. **On-demand Container Loading in AWS Lambda**. 2023. <https://www.usenix.org/system/files/atc23-brooker.pdf>. Accessed: 2025.
- BUYYA JAMES BROBERG, A. M. G. R. **Cloud Computing: Principles and Paradigms**. 1. ed. Canada: WILEY, 2011.
- DOCKER. **Docker Homepage**. 2024. <https://www.docker.com/>. Accessed: 2024.
- EUCALYPTUS. **Eucalyptus**. 2024. <https://www.eucalyptus.cloud/>. Accessed: 2024.
- FERRAZ, R. V. B. Ana Paula do C. M. Taxonomia de bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. **scielo**, São Carlos, v. 1, 2010. Disponível em: <https://www.scielo.br/j/gp/a/bRkFgcJqbGCDp3HjQqFdqBm#>.
- GCP. **Compute Engine | Google Cloud**. 2024. [https://cloud.google.com/products/compute?hl=pt\\_br](https://cloud.google.com/products/compute?hl=pt_br). Accessed: 2024.
- GCP. **Funções do Cloud Run**. 2024. <https://cloud.google.com/functions?hl=pt-BR>. Accessed: 2024.
- GCP. **O que é computação na nuvem?** 2024. <https://cloud.google.com/learn/what-is-cloud-computing?hl=pt-br>. Accessed: 2024.
- HEROKU. **The Heroku Platform**. 2024. <https://www.heroku.com/platform>. Accessed: 2024.
- IBM. **O que é Database-as-a-Service (DBaaS)?** 2024. <https://www.ibm.com/br-pt/topics/dbaas>. Accessed: 2024.
- IBM. **O que é FaaS (Função como Serviço)?** 2024. <https://www.ibm.com/br-pt/topics/faas>. Accessed: 2024.

- IBM. **What is cloud monitoring?** 2024. <https://www.ibm.com/topics/cloud-monitoring>. Accessed: 2024.
- IBM. **What is containers as a service (CaaS)?** 2024. <https://www.ibm.com/topics/containers-as-a-service>. Accessed: 2024.
- IBM. **What is XaaS (anything as a service?)**. 2024. <https://www.ibm.com/topics/xaas>. Accessed: 2024.
- KUBERNETES. **Sobre o Kubernetes**. 2024. <https://kubernetes.io/pt-br/>. Accessed: 2024.
- LU, M. **The World's Biggest Cloud Computing Service Providers**. 2024. <https://www.visualcapitalist.com/worlds-biggest-cloud-computing-service-providers/>. Accessed: 2024.
- MACKENZIE. **Curso de Especialização em Computação em Nuvem (Cloud Computing)**. 2024. [https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/6-pos-graduacao/upm-higienopolis/1PÃSS\\_E\\_MBA/computacao-nuvem/ComputaÃ§Ã£o\\_em\\_nuvem\\_\\_Cloud\\_Computing\\_\\_\\_.pdf](https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/6-pos-graduacao/upm-higienopolis/1PÃSS_E_MBA/computacao-nuvem/ComputaÃ§Ã£o_em_nuvem__Cloud_Computing___.pdf). Accessed: 2024.
- MELLON, C. **Cloud Computing**. 2024. <https://www.cs.cmu.edu/~seth/15619-f20/>. Accessed: 2024.
- MELLON, C. **Cloud Computing: Course Description and Syllabus**. 2024. [https://www.cs.cmu.edu/~seth/15619-f20/15319\\_15619\\_f20\\_Syllabus.pdf](https://www.cs.cmu.edu/~seth/15619-f20/15319_15619_f20_Syllabus.pdf). Accessed: 2024.
- MICROSOFT. **Visão geral da tecnologia Hyper-V**. 2024. <https://learn.microsoft.com/pt-br/windows-server/virtualization/hyper-v/hyper-v-overview>. Accessed: 2024.
- OPENSTACK. **openstack**. 2024. <https://www.openstack.org/>. Accessed: 2024.
- ORACLE. **Oracle VirtualBox**. 2024. <https://www.virtualbox.org/>. Accessed: 2024.
- RAILS, R. on. **Ruby on Rails**. 2024. <https://rubyonrails.org/>. Accessed: 2024.
- SBC. **Referenciais de Formação para os Cursos de Graduação em Computação**. 1. ed. [S.l.]: SBC, 2017.
- STALLINGS, W. **Operating Systems: Internals and Design Principles**. 9. ed. US: Pearson Education, Inc, 2017.
- TANENBAUM, M. V. S. A. S. **Distributed Systems**. 4. ed. US: Pearson Education, Inc, 2023.
- UFMG. **Tópicos em Redes de Computadores: Virtualização e Computação em Nuvem (Cloud Computing)**. 2024. <https://homepages.dcc.ufmg.br/~dorgival/cursos/vcn/curso.pdf>. Accessed: 2024.
- UFOP. **Computação nas Nuvens**. 2024. [http://www3.decom.ufop.br/decom/disciplina\\_ementa/bcc422/](http://www3.decom.ufop.br/decom/disciplina_ementa/bcc422/). Accessed: 2024.
- UNICAMP. **Nuvens e Virtualização de Redes**. 2024. <https://nuvens.ic.unicamp.br/>. Accessed: 2024.

UNIVERSITY, S. **Cloud Computing Technology**. 2024. <https://web.stanford.edu/class/cs349d/>. Accessed: 2024.

USP. **Disciplina: SSC0158 - Computação em Nuvem e Arquitetura Orientadas a Serviços**. 2024. <https://uspdigital.usp.br/jupiterweb/obterDisciplina?sgldis=SSC0158&verdis=2>. Accessed: 2024.

USP. **Disciplina: SSC5973 Computação em Nuvem**. 2024. <https://www.icmc.usp.br/pos-graduacao/disciplinas?programa=55134&disciplina=SSC5973>. Accessed: 2024.

UZIEL, A. **Andreusd/tcc**. 2025. <https://github.com/Andreusd/tcc>. Accessed: 2025.

VMWARE. **vmware workstation**. 2024. <https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion>. Accessed: 2024.

XEN. **Xen Project**. 2024. <https://xenproject.org/>. Accessed: 2024.

ZHAO ADAM MORRISON, C. W. F. J. T. Z. N. **Everywhere All at Once: Co-Location Attacks on Public Cloud FaaS**. 2024. [https://iacomaweb.web.engr.illinois.edu/iacomaweb/papers/asplos24\\_1.pdf](https://iacomaweb.web.engr.illinois.edu/iacomaweb/papers/asplos24_1.pdf). Accessed: 2025.

## APÊNDICE A – LABORATÓRIO: CRIANDO E CONECTANDO EM UMA INSTÂNCIA EC2 NA AWS

**Objetivo:** Esse laboratório visa prover ao aluno uma introdução prática à computação em nuvem, explorando o modelo de Infraestrutura como Serviço (IaaS) por meio da criação e configuração de uma máquina virtual hospedada remotamente.

**Resumo:** Nesse laboratório, o aluno irá inicializar uma instância EC2 no console da AWS, conectar-se a ela via protocolo SSH e escrever um programa simples em Python diretamente pelo terminal, utilizando um editor de texto como Nano ou Vim.

O *Amazon Elastic Compute Cloud* (EC2) é a solução da Amazon para criar máquinas virtuais na nuvem, permitindo ao usuário se conectar e executar comandos via interface de linha de comando (CLI). Esse serviço se encaixa no paradigma Infraestrutura como Serviço, pois oferece seus recursos no menor nível de abstração possível, permitindo configurar o hardware da máquina virtualizada, o sistema operacional e o estado da máquina (ligado/desligado). O serviço equivalente na plataforma Azure é o *Azure Virtual Machines*, e na Google Cloud é o *Google Compute Engine*.

O EC2 trabalha com o conceito de instâncias, que são máquinas virtuais pré-configuradas com uma imagem de máquina da Amazon, contendo o sistema operacional e uma camada de software adicional. As instâncias do EC2 se diferenciam em aspectos como capacidade de CPU, memória e armazenamento. Neste laboratório, utilizaremos a instância *t2.micro*, disponível no nível gratuito da AWS (*Free Tier*), que inclui 750 horas de uso por mês. Essa configuração oferece 1 CPU virtual e 1 GB de memória RAM, sendo suficiente para os objetivos deste exercício.

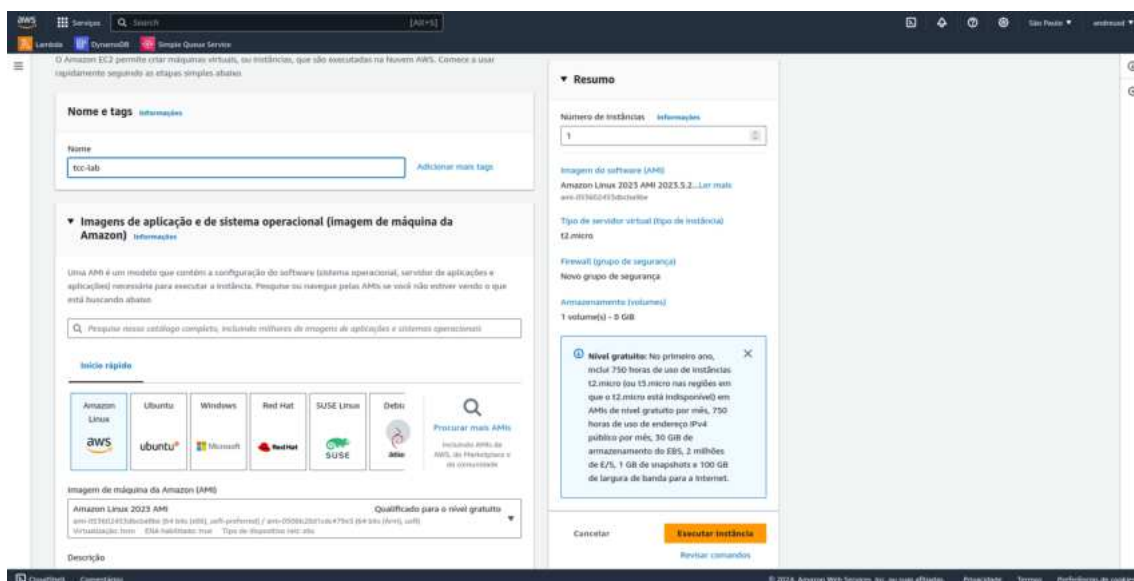
### Roteiro:

#### Passo 1 - Configurar instância

- Acesse o serviço EC2 buscando por ele na barra de pesquisa do console da AWS;
- Dentro do serviço do EC2, clique em executar instância;
- Preencha um nome para a instância e selecione uma imagem que tenha na descrição “Qualificado para nível gratuito” e preferencialmente que já venha com um interpretador de Python instalado (exemplo: Amazon Linux). A figura 9 mostra a interface de criação de uma instância EC2, com alguns dos sistemas operacionais disponíveis;
- Selecione o tipo de instância *t2.micro*;
- Crie um par de chaves novo. Esse par de chaves será usado para conexão com a instância via protocolo SSH. Se ele for perdido, será necessário recriar a instância com um par novo;

- Selecione executar instância.

Figura 9 – Interface de criação de uma instância EC2 na AWS



Material produzido para esse trabalho

## Passo 2 - Conectando na instância

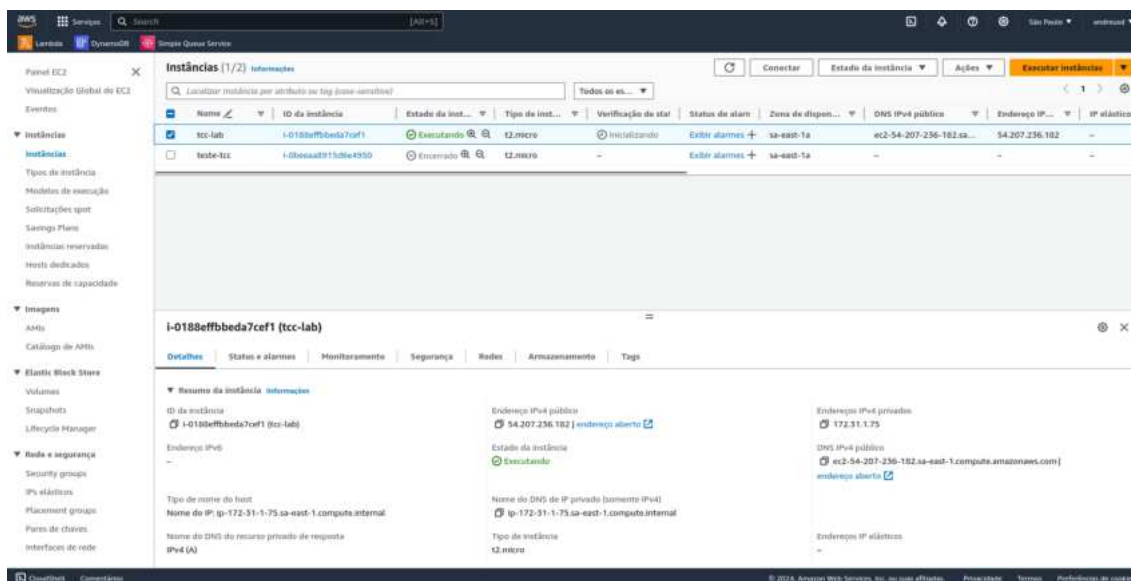
As instruções deste passo são destinadas a serem executadas em máquinas Linux, por ser o sistema operacional dos computadores dos laboratórios do IC.

- Selecione a instância criada e copie o endereço IPv4 público. A figura 10 mostra a interface que lista todas as instâncias EC2 que pertencem a sua conta;
- Em sua máquina Linux, abra o emulador de terminal no diretório em que salvou o par de chaves;
- Digite `chmod 400 "{nome-do-par-de-chaves}.pem"` para dar permissão de leitura somente para o usuário (essa é uma medida de segurança do protocolo SSH);
- Digite `ssh -i "{nome-do-par-de-chaves}.pem {endereço-ipv4-copiado}"` e confirme a conexão digitando `yes`.

## Passo 3 - Executando comandos na instância

- Crie um arquivo na linguagem Python em um editor de linha de comando de sua preferência, como *vim* ou *nano*;
- Inicialmente escreva um programa simples, como um `“Hello World”`;

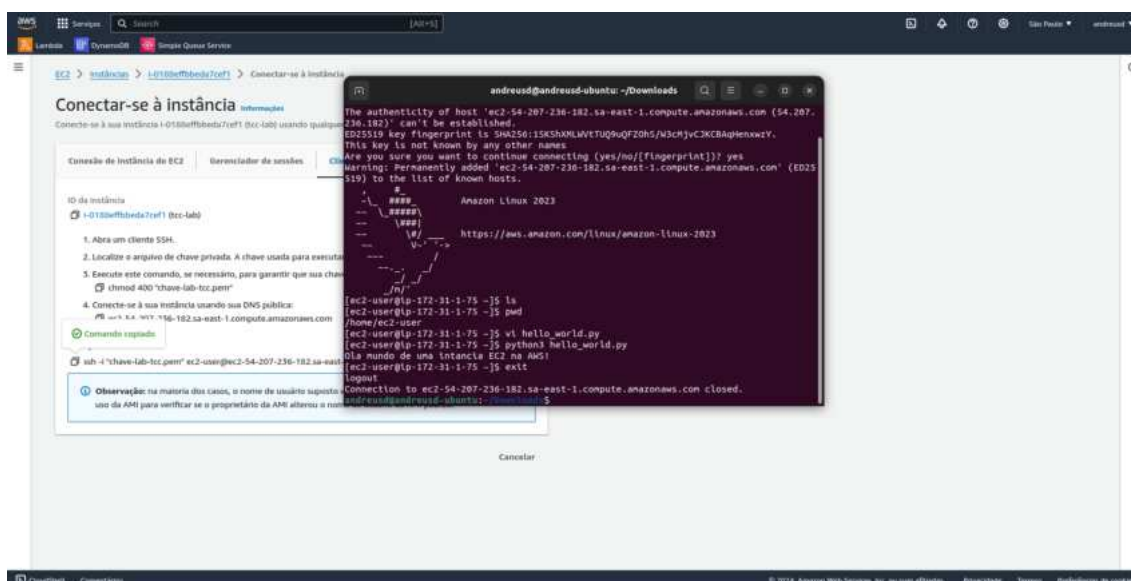
Figura 10 – Interface de listagem de instâncias EC2 na AWS



Material produzido para esse trabalho

- Execute o script chamando “python3 nome-do-arquivo.py” (Python já vem instalado nessa imagem, mas pode ser que seja necessário instalá-lo pelo gerenciador de pacotes). A figura 11 mostra a saída do meu programa;
- Como atividade adicional, propomos utilizar a biblioteca *requests* do Python para fazer uma chamada HTTP a algum serviço REST público, e exibir a resposta.

Figura 11 – Executando comandos remotamente na instância EC2 criada a partir de uma instalação local de Linux



Material produzido para esse trabalho

## APÊNDICE B – LABORATÓRIO: PERSISTINDO E CONSULTANDO DADOS EM UM BANCO DE DADOS NA NUVEM COM AMAZON DYNAMODB

**Objetivo:** Prover ao aluno uma primeira experiência com o modelo de Plataforma como Serviço, aplicado a um banco de dados em nuvem não relacional.

**Resumo:** Nesse laboratório, o aluno deverá criar uma tabela no *DynamoDB*, configurar localmente as credenciais para conexão com a AWS, e interagir com essa tabela com operações de escrita, leitura e exclusão a partir de sua máquina.

O Amazon DynamoDB é um banco de dados não relacional sem servidor com foco em desempenho e escalabilidade. Seu modelo de negócio consiste em pagar apenas por utilização, sem valor mínimo ou assinatura mensal, com garantia de 99,999% de disponibilidade. O Dynamo é ideal para arquiteturas orientadas a evento, permitindo definir *triggers* para que aplicações tratem um evento na tabela, seja criação, atualização ou exclusão, sempre que ocorrer uma mudança. O nível gratuito da AWS oferece 25 GB de armazenamento, além de capacidade de gravação e leitura suficiente para processar até 200 milhões de solicitações gratuitas.

Alternativamente, também poderíamos utilizar o MongoDB Atlas, que é uma opção de banco de dados não relacional que não pertence a nenhuma grande provedora de nuvem. Na plataforma Azure, o serviço mais parecido com o DynamoDB é o Cosmos DB.

Tabelas no DynamoDB podem possuir uma ou duas chaves, a chave de partição e chave de classificação (opcional). A combinação dessas duas chaves constitui a chave forte da tabela, que não pode se repetir. A utilização de duas chaves pode trazer ganhos de desempenho na consulta, por exemplo podemos salvar objetos que estão intimamente relacionados com a mesma chave de partição (PK), mas com chave de classificação diferentes. Dessa forma, podemos fazer uma única consulta por PK e trazer todos esses objetos relacionados de uma única vez. Para fins práticos, nesse laboratório utilizaremos somente uma chave.

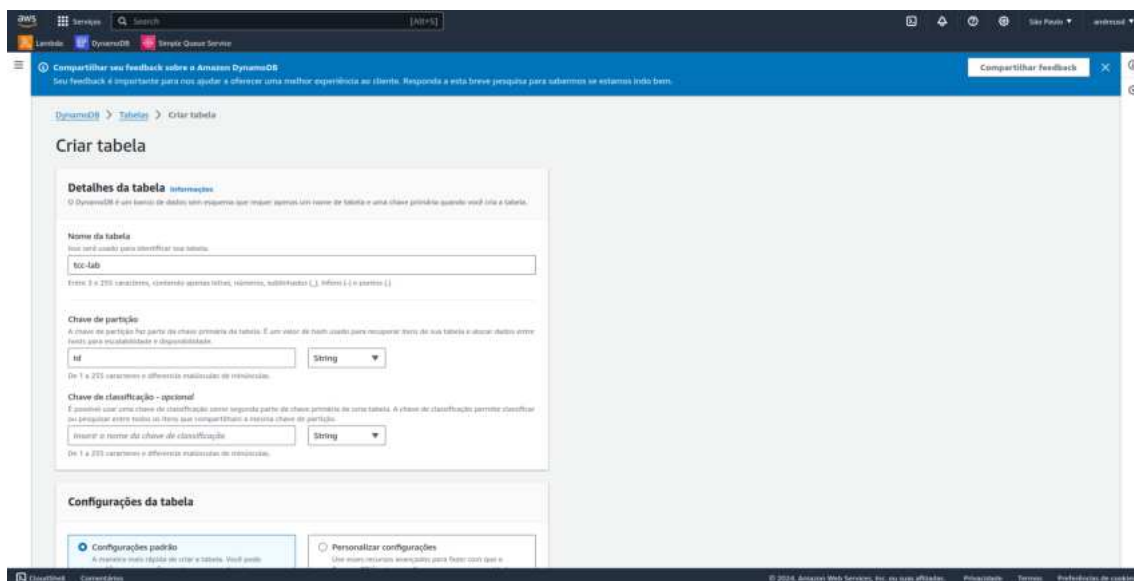
### **Roteiro:**

#### **Passo 1 - Criação da Tabela**

- Abrir console da AWS e pesquisar por DynamoDB;
- No painel inicial, clicar em criar tabela;
- Escolher um nome para a tabela e um para a chave de partição;
- Confirmar a criação da tabela. A figura 12 mostra essa interface de criação.

#### **Passo 2 - Autenticação Local**

Figura 12 – Criando uma tabela no DynamoDB



Material produzido para esse trabalho

Para conseguir realizar operações na AWS via código, será necessário configurar as credenciais de acesso à conta na máquina local. Para isso, é necessário instalar a interface de linha de comando da AWS (AWS CLI).

- Para instalar o AWS CLI pelo gerenciador de pacotes do Linux, execute o comando "sudo apt install awscli";
- Para obter as credenciais de acesso, volte ao console da AWS, aperte no seu nome de usuário e entre em "Credenciais de segurança";
- Aperte em Criar chave de acesso e confirme o checkbox. Como usuário raiz da conta, não há problema em fazer isso;
- De volta ao terminal, digite "aws configure", e cole a chave de acesso e a chave secreta. Para a região, utilizamos "sa-east-1", que é a com menos latência;

### Passo 3 - Operações

Para interagir com o DynamoDB usando Python, recomenda-se utilizar a SDK da AWS para Python, o Boto3.

- Para instalar a biblioteca Boto3 utilizando o pip, digite "pip install boto3";
- A partir daí, pode-se consultar a documentação do boto3 para realizar operações (AWS, 2024f);

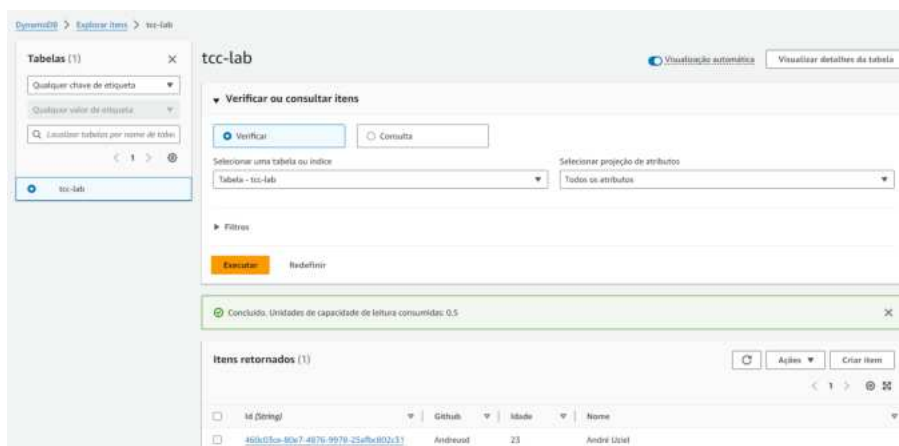
- A figura 13 exibe um código em Python que consiste em escrever, ler e excluir um objeto de uma tabela do DynamoDB. A figura 14 mostra o resultado da operação de escrita, no console da AWS.

Figura 13 – Código em Python para realizar operações no DynamoDB

```
operacoes-dynamo.py src X
operacoes-dynamo.py > ...
1 import uuid
2 import boto3
3
4 NOME_TABELA = "tcc-lab"
5
6 dynamodb = boto3.resource("dynamodb")
7 table = dynamodb.Table(NOME_TABELA)
8
9 # Gera um identificador aleatório
10 id = str(uuid.uuid4())
11
12 # Escrita
13 result = table.put_item(
14     Item={
15         "Id": id,
16         "Nome": "André Uziel",
17         "Github": "Andreusd",
18         "Idade": 23,
19     }
20 )
21
22 # Leitura
23 item = table.get_item(Key={"Id": id})
24 print(item["Item"])
25
26 # Exclusão
27 table.delete_item(Key={"Id": id})
28
```

Material produzido para esse trabalho

Figura 14 – Escrita em tabela no DynamoDB



The screenshot shows the AWS DynamoDB console interface. On the left, there's a sidebar with 'Tabelas (1)' and 'tcc-lab' selected. The main area shows the 'tcc-lab' table details. Under 'Verificar ou consultar itens', the 'Verificar' radio button is selected. The table name is 'Tabela - tcc-lab' and the projection is 'Todos os atributos'. A green status bar indicates 'Concluído. Unidades de capacidade de leitura consumidas: 0.5'. Below, the 'Itens retornados (1)' section shows a table with one item:

Id (String)	Nome	Idade	GitHub
45b6f5e-80e7-4876-9579-25af8e102131	André Uziel	23	Andreusd

Material produzido para esse trabalho

## APÊNDICE C – LABORATÓRIO: EXECUTANDO CÓDIGO NA NUVEM COM AWS LAMBDA

**Objetivo:** O objetivo desse laboratório é apresentar ao aluno um formato um pouco mais genérico do modelo de Plataforma como Serviço, que permite ao desenvolvedor executar qualquer código de forma remota na nuvem, sem se preocupar com o gerenciamento de servidores: a arquitetura Sem Servidor (*Serverless*). Esse formato é considerado por algumas fontes como um modelo próprio, por ter características tanto do PaaS como do IaaS, devido ao grau de liberdade dada ao desenvolvedor, como é o caso da Microsoft. (AZURE, 2024g)

**Resumo:** O aluno deverá criar uma Função Lambda (*Lambda Function*) que, ao receber um evento, deverá realizar algum processamento e persistir a saída no banco de dados utilizado no laboratório anterior, o DynamoDB.

Apesar do nome, ainda há servidores na arquitetura Sem Servidor, mas eles estão completamente abstraídos do desenvolvimento da aplicação, o que permite que o desenvolvedor mantenha seu foco somente no código, enquanto que o provedor de nuvem fica responsável por alocar e fornecer os recursos necessários de computação quando o código é executado, e também por removê-los completamente, quando a execução é interrompida.

As arquiteturas sem servidor são altamente escalonáveis e normalmente controladas por eventos, usando recursos apenas quando ocorre uma função ou um evento que desencadeie esse uso. Isso permite a construção de aplicações que sejam extremamente eficientes, sem desperdício durante períodos ou horários ociosos, e cujo custo varia diretamente de acordo com o volume.

Para esta finalidade, utilizaremos o serviço AWS Lambda, que é a solução da Amazon para arquiteturas Sem Servidor. A alternativa da Microsoft é o *Azure Functions*, e da Google é o *Google Cloud Functions*. Uma vantagem de usar o serviço da AWS, é que não será necessário configurar credenciais para integração com o DynamoDB, por ambas estarem dentro do mesmo ecossistema AWS.

### Roteiro:

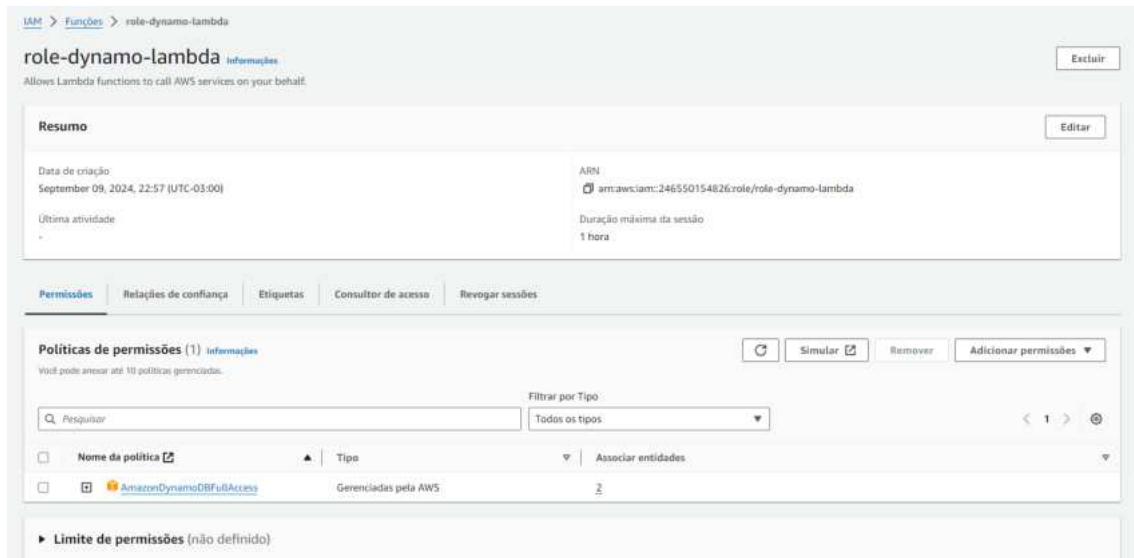
#### Passo 1 - Criando o perfil de acesso ao DynamoDB

Para garantir que nossa função Lambda terá acesso a outros serviços da AWS, como o DynamoDB, é necessário criar um perfil de acesso no Gerenciador de Identidade e Acesso da Amazon, o IAM.

- Acesse o IAM procurando pelo serviço no console da AWS;
- Entre na aba *Roles* e aperte em Criar perfil;
- Dentro de Caso de uso, busque por Lambda e aperte Próximo;

- Busque pela permissão *AmazonDynamoDBFullAccess*, marque e parte Próximo;
- Escolha o nome do perfil (escolhi *role-dynamo-lambda*) e aperte em Criar perfil. A figura 15 exhibe a interface de criação de um novo perfil no IAM.

Figura 15 – Role IAM para acesso ao DynamoDB por funções lambda



Material produzido para esse trabalho

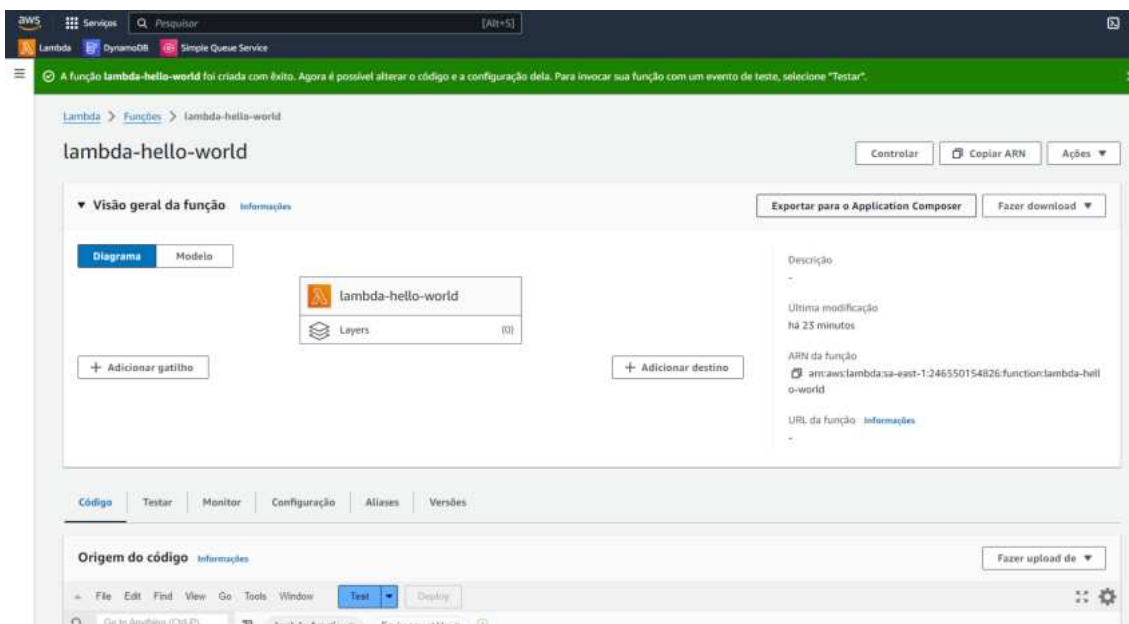
## Passo 2 - Criando a função Lambda

- Acesse o serviço procurando por “lambda” no console da AWS;
- Aperte em criar função e escolha um nome único (escolhi “lambda-hello-world”);
- Em “Tempo de execução” (*Runtime*) selecione Python na versão mais recente;
- Para a arquitetura, recomenda-se escolher arm64, pois a AWS fez otimizações para seu CPU Graviton2 que a torna 19% mais eficiente e 20% mais barata que a x86. (AWS, 2024e);
- No “Papel de execução”, selecione “Usar uma função existente” e escolha o role que criamos no passo 1 (*role-dynamo-lambda*);
- Aperte em criar função. A figura 16 exhibe o detalhamento da *Lambda* criada.

## Passo 3 - Invocando a função

- A função lambda que criamos já vem pré configurada com um código padrão muito simples, que consiste em retornar a string “Hello from Lambda!” quando invocada;

Figura 16 – Função Lambda criada



Material produzido para esse trabalho

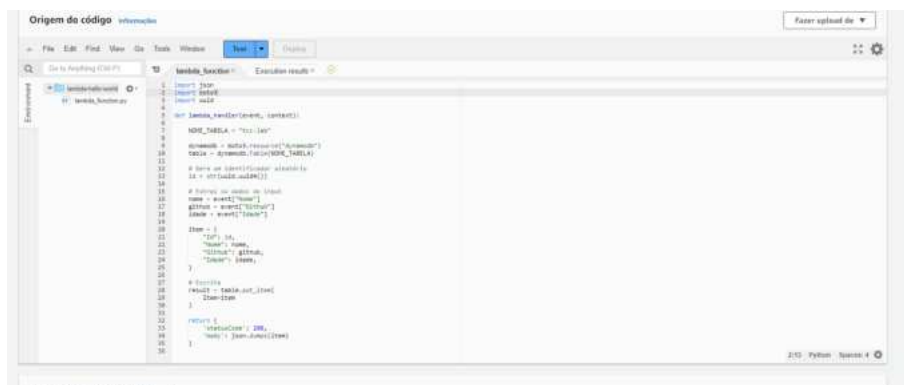
- Para invocar uma execução da Lambda, basta apertar no botão azul *Test* e em seguida “Invocar”. Essa interface nos permite configurar a entrada (*input*) da função, mas nesse código padrão ele não é usado;
- Uma nova guia na IDE integrada será aberta, com o nome *Execution results*, e vai conter a saída (*output*) da função.

#### Passo 4 - Alterando o código

Agora que já testamos a função, podemos alterar o código para interagir com outros serviços da AWS, no nosso caso o DynamoDB. A implementação será muito parecida com a do laboratório anterior, a diferença é que receberemos os dados do usuário que desejamos salvar no banco de dados como entrada da função. Dessa forma, é possível integrar nossa função em algum outro serviço, interno (ex: outra *Lambda*) ou externo (ex: *API*) à AWS e encapsular o acesso ao BD.

- Voltando à IDE integrada da lambda, basta editar o código e realizar a implementação (*deploy*) das alterações. A figura 17 mostra a IDE integrada com o código escrito para a *Lambda*.
- Em seguida, configura-se um novo evento de testes, que contém os dados de entrada adequados para nossa aplicação e apertamos em Invocar. A figura 18 mostra um exemplo de dados de entrada da *Lambda*, que será processado pela aplicação;
- Ao abrir o serviço do DynamoDB, a nova entidade de usuário deve estar persistida, e deve refletir os dados do evento que configuramos.

Figura 17 – Código na Lambda para persistência no DynamoDB



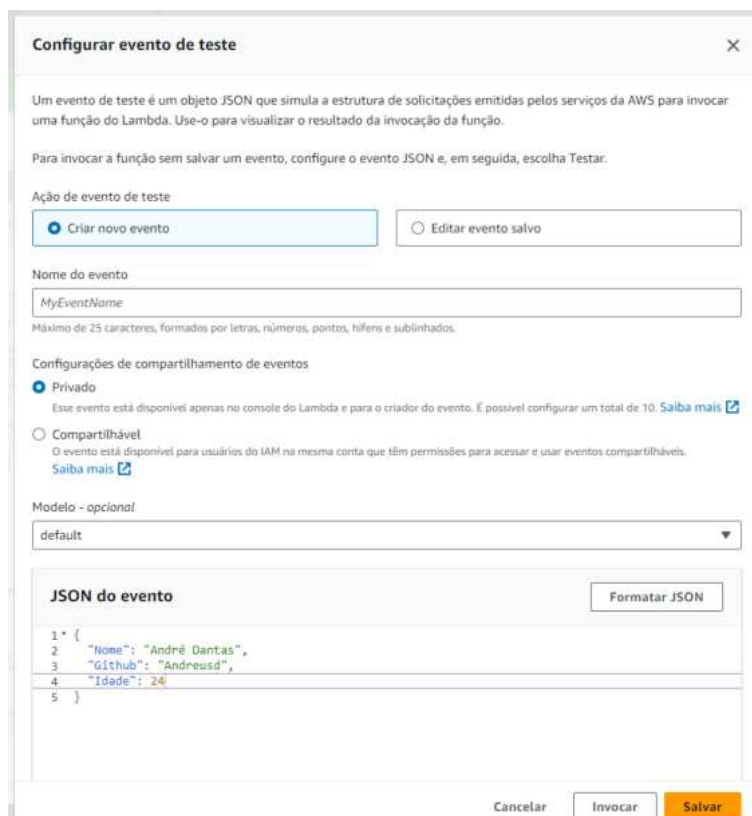
```

1 import boto3
2 import os
3 import json
4
5 def lambda_handler(event, context):
6     # Nome do usuário
7     nome = event["nome"]
8     # Nome do GitHub
9     github = event["github"]
10    # Idade do usuário
11    idade = event["idade"]
12
13    # Cria o cliente do DynamoDB
14    dynamodb = boto3.client('dynamodb')
15
16    # Cria o item a ser salvo no DynamoDB
17    item = {
18        'nome': {'S': nome},
19        'github': {'S': github},
20        'idade': {'N': idade}
21    }
22
23    # Salva o item no DynamoDB
24    dynamodb.put_item(
25        TableName=os.environ['TABLE_NAME'],
26        Item=item
27    )
28
29    # Retorna o resultado da invocação
30    return {
31        'statusCode': 200,
32        'body': json.dumps('Item salvo com sucesso!')
33    }

```

Material produzido para esse trabalho

Figura 18 – Evento configurado com entrada da aplicação na Lambda



**Configurar evento de teste**

Um evento de teste é um objeto JSON que simula a estrutura de solicitações emitidas pelos serviços da AWS para invocar uma função do Lambda. Use-o para visualizar o resultado da invocação da função.

Para invocar a função sem salvar um evento, configure o evento JSON e, em seguida, escolha Testar.

Ação de evento de teste

Criar novo evento  Editar evento salvo

Nome do evento

Máximo de 25 caracteres, formados por letras, números, pontos, hífens e sublinhados.

Configurações de compartilhamento de eventos

Privado  
Este evento está disponível apenas no console do Lambda e para o criador do evento. É possível configurar um total de 10. [Saiba mais](#)

Compartilhável  
O evento está disponível para usuários do IAM na mesma conta que têm permissões para acessar e usar eventos compartilháveis. [Saiba mais](#)

Modelo - opcional

**JSON do evento**

```

1 {
2   "nome": "André Dantas",
3   "github": "Andreusd",
4   "idade": 24
5 }

```

Material produzido para esse trabalho

## APÊNDICE D – TRABALHO FINAL: APLICAÇÃO COMPLETA NA NUVEM

**Objetivo:** O objetivo deste trabalho é aplicar os conhecimentos adquiridos nas aulas teóricas e práticas ao longo da disciplina para construir uma aplicação completa na nuvem. Essa aplicação deve integrar diferentes serviços de computação em nuvem de forma a oferecer valor para o usuário.

**Resumo:** O aluno deverá criar uma aplicação funcional com tema livre que esteja hospedada na nuvem. Para isso, ele poderá utilizar serviços que já teve contato nos laboratórios, ou novos serviços que julgar úteis para sua aplicação. Não é necessário interface gráfica pois este não é o foco da disciplina.

**Requisitos:** A aplicação final deverá utilizar ao menos três serviços na nuvem, sendo um deles para persistência em banco de dados, como o DynamoDB. Também deve ser entregue um desenho da arquitetura da aplicação, demonstrando as interações entre os serviços e usuários.

**Avaliação:** Os alunos serão avaliados de acordo com a qualidade da apresentação de seu trabalho para a turma, criatividade para resolver o problema e profundidade do relatório, o que inclui casos de uso, cenários de teste e diagramas conceituais.

### **Exemplo de possível trabalho:**

Como um exemplo muito simples de trabalho final, serão utilizados alguns dos serviços que já foram apresentados em laboratórios anteriores, aliado a um serviço que não foi utilizado ainda para montar uma aplicação que pode ser acessada via *HTTP*, por usuários não autenticados na AWS.

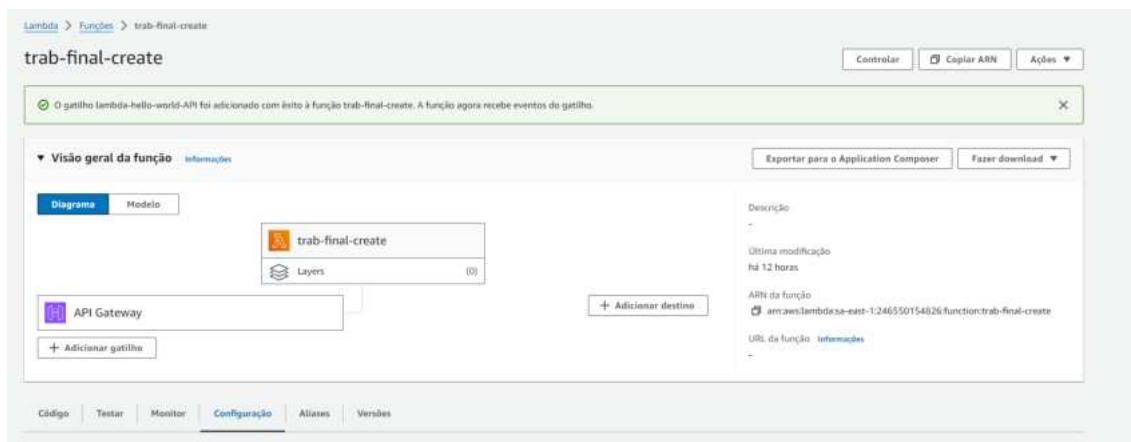
Para isso, utilizamos um serviço da Amazon chamado *API Gateway*. Esse serviço permite criar *endpoints HTTP* e integrá-los com outros serviços da AWS. Essa aplicação consistirá de um *CRUD* (*Create, Read, Update, Delete*) para cadastrar e consultar usuários em uma base de dados.

Desta forma, foram criadas quatro funções Lambda (uma para cada operação), com integração com o *API Gateway*. Ao criar o *Gateway*, a AWS fornece uma chave de API, que deve ser enviada como header em todas as requisições HTTP e funciona como uma forma muito simples de autenticação. Cada função Lambda tem um código diferente que realiza uma operação no DynamoDb. A figura 19 mostra a *Lambda* criada, já com a integração com o *API Gateway* configurada.

A ferramenta que utilizamos para fazer chamadas HTTP é o *Postman*. A rota que cria um usuário aceita o método PUT e recebe um nome, um usuário e a idade. A rota que atualiza um usuário aceita o método POST e recebe também o id de um usuário existente no corpo da requisição. A rota que consulta os usuários cadastrados é um GET e não precisa de nenhum parâmetro, mas poderia ser expandida para aceitar filtros. E a rota que apaga um usuário usa o método DELETE e recebe somente o id do usuário. A

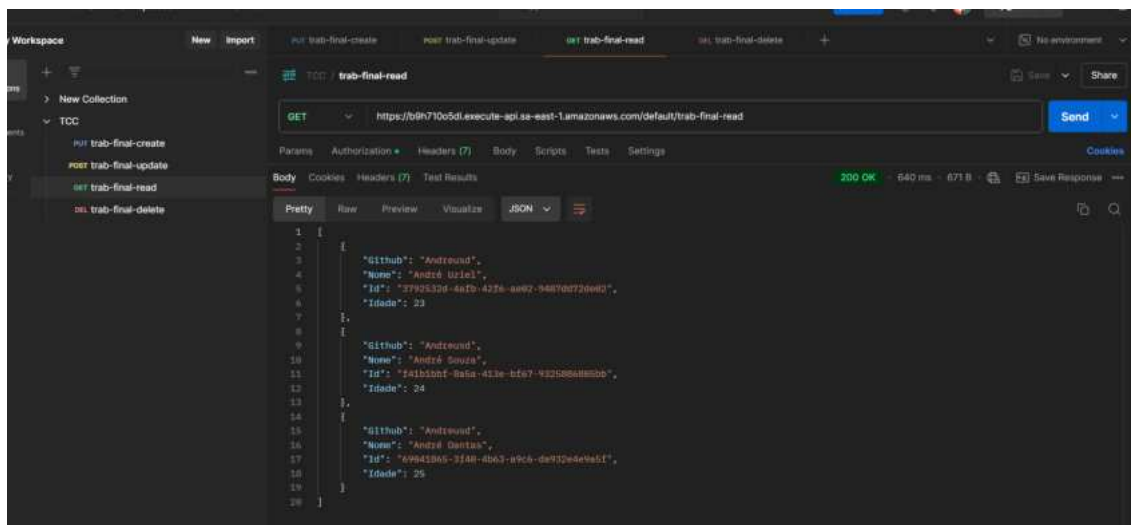
figura 20 mostra a ferramenta *Postman* com as rotas configuradas.

Figura 19 – Lambda com integração com o API Gateway



Material produzido para esse trabalho

Figura 20 – Utilizando a ferramenta Postman para interagir com a API



Material produzido para esse trabalho