



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO DE INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

Maria Angelica da Costa Ciriaco

**Uma Intervenção Metodológica no Ensino de Computação em Nuvem, utilizando
Contêineres Docker, Kubernetes e AWS de Infraestrutura**

PAU DOS FERROS - RN

2025

Maria Angelica da Costa Ciriaco

**Uma Intervenção Metodológica no Ensino de Computação em Nuvem, utilizando
Contêineres Docker, Kubernetes e AWS de Infraestrutura**

Monografia apresentada à Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Tecnologia da Informação.

Orientador: Reudismam Rolim de Sousa, Prof.
Dr.

PAU DOS FERROS - RN

2025

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

C578i Ciriaco, Maria Angelica da Costa.
Uma intervenção metodológica no ensino de
computação em nuvem, utilizando contêineres
docker, kubernetes e AWS de infraestrutura /
Maria Angelica da Costa Ciriaco. - 2025.
81 f. : il.

Orientador: Reudismam Rolim de Sousa.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Tecnologia da
Informação, 2025.

1. Computação em nuvem. 2. Docker. 3.
Kubernetes. 4. AWS. 5. Metodologias ativas. I.
Sousa, Reudismam Rolim de , orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

Maria Angelica da Costa Ciriaco

**Uma Intervenção Metodológica no Ensino de Computação em Nuvem, utilizando
Contêineres Docker, Kubernetes e AWS de Infraestrutura**

Monografia apresentada à Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Tecnologia da Informação.

Defendida em: **05/08/2025**.

BANCA EXAMINADORA

Reudismam Rolim de Sousa, Prof. Dr. - (UFERSA)
Presidente,

Hortência Pessoa Rêgo Gomes, Prof. Me. - (UFERSA)
Membro Examinador

Leandro de Almeida Melo, Prof. Dr. - (UFERSA)
Membro Examinador

AGRADECIMENTOS

Gostaria de agradecer à minha mãe, dona Raimunda, por ter me incentivado a estudar e por sempre acreditar na minha capacidade de realizar sonhos por meio da educação. Ela é o meu maior exemplo de mulher e mãe. Esta conquista também é dela.

Agradeço, com muito carinho aos professores que marcaram profundamente minha trajetória: à professora Edileuza, que me ensinou a ler e despertou em mim a busca pelo conhecimento; ao professor Benedito Maia, que acompanhou minha adolescência e sempre acreditou no meu potencial, mesmo diante das dificuldades de aprendizagem que enfrentei no Ensino Fundamental; à professora Sônia Aires, por confiar em mim e apoiar meu crescimento acadêmico e pessoal. Sou, também, profundamente grata ao professor Walber José, que me apresentou ao mundo da computação em nuvem (*cloud*). Hoje, atuo como júnior em uma empresa parceira da AWS e posso afirmar que estou vivendo um sonho: fazer o que gosto, ser reconhecida pelo meu trabalho e ser certificada pela AWS.

Gostaria de agradecer ao Prof. Dr. Leandro Melo, meu primeiro orientador, que inspirou o início de uma ideia que se tornou real com a possibilidade de ajudar outras pessoas.

Quero também expressar minha sincera gratidão ao meu orientador, Prof. Dr. Reudismam Rolim, meu primeiro professor de Algoritmos, na UFERSA. Sua paciência e sabedoria foram essenciais para que eu conseguisse aprender tecnologia em um ambiente totalmente novo, no qual entrei literalmente de paraquedas e decidi permanecer.

Sou imensamente grata por ter tido ao meu lado educadores incríveis que, mais do que exercer seu papel profissional, deixaram marcas profundas na minha história acadêmica e pessoal, marcas que levarei comigo por toda a vida.

RESUMO

Este trabalho apresenta uma intervenção metodológica com o objetivo de aplicar conceitos de Computação em Nuvem no ensino de contêineres Docker e Kubernetes, utilizando a plataforma AWS como infraestrutura para o desenvolvimento de um minicurso. Para embasar a proposta, foi realizada uma Revisão Sistemática da Literatura (RSL), conforme os procedimentos metodológicos descritos por Kitchenham (2004), com o intuito de identificar metodologias de ensino e formas avaliativas compatíveis com os objetivos do curso. A RSL busca responder a três questões de pesquisas principais: “QP₁: Quais são as metodologia aplicadas em relatos de experiência sobre a adoção de cursos de ensino/extensão?”, “QP₂: Como os cursos de ensino/extensão são avaliados nos relatos de experiências apresentados?” e “QP₃: Quais são os principais resultados buscados nos relatos de experiência sobre a execução dos cursos de ensino/extensão?”. Os resultados da revisão evidenciaram a efetividade do uso de metodologias ativas, como a sala de aula invertida e a aprendizagem baseada em problemas, no contexto do ensino de tecnologias através da prática (RQ₁), apontando também o interesse por avaliações que busquem capturar o desenvolvimento de habilidades e competências (RQ₁). Os resultados principais buscados pelas ações estão relacionados ao desenvolvimento da aprendizagem significativa e ao desenvolvimento do pensamento crítico como pontos a serem desenvolvidos pelos cursos (RQ₃).

Palavras-chave: computação em nuvem; docker; kubernetes; AWS; metodologias ativas.

ABSTRACT

This work presents a methodological intervention that applies Cloud Computing concepts to the teaching of Docker and Kubernetes containers, utilizing the AWS platform as the infrastructure for developing a mini-course. To support the proposal, a Systematic Literature Review (SLR) was conducted, following the methodological procedures described by Kitchenham (2004), to identify teaching methodologies and assessment methods compatible with the course objectives. The SLR seeks to answer three main research questions: "RQ1: What are the methodologies applied in experience reports on the adoption of teaching/extension courses?", "RQ2: How are teaching/extension courses evaluated in the experience reports presented?", and "RQ3: What are the main outcomes sought in the experience reports on the implementation of teaching/extension courses?" The review results highlighted the effectiveness of active methodologies, such as the flipped classroom and problem-based learning, in the context of teaching technology through practice (RQ1). They also highlighted the interest in assessments that seek to capture the development of skills and competencies (RQ1). The main outcomes sought by the initiatives are related to the development of meaningful learning and critical thinking, areas that the courses aim to develop (RQ3).

Keywords: cloud computing; docker; kubernetes; AWS; active learning.

LISTA DE FIGURAS

Figura 1 - Host Management Configurado.....	37
Figura 2 - Configurações de Roles no IAM.....	38
Figura 3 - Conexão de Instância.....	39
Figura 4 - Testando Conexão de Instância.....	39
Figura 5 - Principais Comandos básicos do Linux.....	40
Figura 6 - Primeiros Comandos Docker.....	41
Figura 7 - Comando para Interagir com Contêineres.....	42
Figura 8 - Testando no Navegador.....	42
Figura 9 - Comandos Docker para Gerenciar Ciclo de Vida do Contêiner.....	43
Figura 10 - Clonando Repositório GitHub.....	44
Figura 11 - Construindo Imagem no Dockerfile.....	44
Figura 12 - Construção da Imagem Docker.....	45
Figura 13 - Verificar e Executar a Imagem Docker.....	45
Figura 14 - Imagem no Repositório Docker Hub.....	46
Figura 15 - Comandos para Criar Diretório e Arquivo.....	49
Figura 16 - Editor de Texto com o Manifesto YAML.....	49
Figura 17 - Comando para Criar Cluster.....	50
Figura 18 - Credenciamento com Access keys.....	52
Figura 19 - Configuração no Terminal do AWS configure.....	52
Figura 20 - Cluster EKS no Console AWS.....	53
Figura 21 - Nodes no Console AWS.....	53
Figura 22 - Comando do Deployment.....	54
Figura 23 - Comando kubectl get all.....	54
Figura 24 - DNS do Load Balancer no Navegador.....	55
Figura 25 - Experiência com Cloud Computing.....	57
Figura 26 - Experiência com Provedores de Nuvem.....	57

Figura 27 - Nível de Conhecimento em AWS.....	58
Figura 28 - Nível de Conhecimento em Docker.....	58
Figura 29 - Nível de Conhecimento em Kubernetes.....	59
Figura 30 - Confiança para aplicar os conteúdos aprendidos na prática.....	60
Figura 31 - Realização de atividades práticas durante o curso.....	60
Figura 32 - Avaliação da Didática do Curso.....	61
Figura 33 - O conteúdo atendeu às suas expectativas.....	61
Figura 34 - Avaliação do Ritmo das Aulas.....	62
Figura 35 - Intenção de uso do conteúdo aprendido.....	62
Figura 36 - Indicação do curso para outras pessoas.....	63

LISTA DE QUADROS

Quadro 1 - Relação de trabalhos que apresentam diferentes resultados buscados nas ações de formação.....	29
Quadro 2 - Sugestão de estrutura para o minicurso com base no artigo.....	74
Quadro 3 - Sugestão de atividades práticas no estilo do artigo.....	77

LISTA DE ABREVIATURAS E SIGLAS

UFERSA	Universidade Federal Rural do Semi-Árido.
BITI	Bacharelado Interdisciplinar em Tecnologia da Informação.
EC2	Amazon Elastic Compute Cloud
IAM	Identity and Access Management
EKS	Amazon Elastic Kubernetes Service
SO	Sistema Operacional
AWS	Amazon Web Services

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Justificativa.....	16
1.2	Objetivo.....	16
1.3	Organização do trabalho.....	17
2	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	Contêineres.....	18
2.2	Docker.....	18
2.3	Docker hub.....	19
2.4	Kubernetes.....	19
2.5	Cloud computing.....	20
2.6	Amazon web services (AWS).....	21
3	REVISÃO SISTEMÁTICA DA LITERATURA.....	22
3.1	Quais são as metodologias aplicadas pelos cursos de ensino na educação superior?.....	23
3.1.1	Metodologias Ativas Tradicionais.....	23
3.1.2	Metodologias Ativas Personalizadas.....	24
3.1.2	Modelo Tecnológico.....	25
3.1.3	Abordagem Mista Freiriana.....	26
3.1.4	Lúdico.....	26
3.1.5	Pesquisa-Ação.....	26
3.1.6	Método do Caso de Ensino.....	27
3.1.7	Modelo Didático.....	27
3.2	RQ2 - Como os cursos de ensino são avaliados na educação superior?.....	27

3.2.1	Desenvolvimento de Habilidades e Competências.....	27
3.2.2	Percepção dos Alunos.....	28
3.2.3	Adaptação de Práticas Pedagógicas.....	29
3.2.4	Avaliação Continuada.....	29
3.2.5	Avaliação Mista.....	30
3.2.6	Sistemas de Avaliações Autênticos.....	30
3.2.7	Análise do Discurso.....	31
3.3	RQ3 - Quais são os principais resultados buscados pelos cursos de ensino na educação superior?.....	31
4	INTERVENÇÃO METODOLÓGICA.....	34
4.1	Estrutura do minicurso.....	34
4.2	Módulo 1: Introdução à containerização.....	34
4.2.1	Conceito de Contêiner	35
4.2.2	Diferença Entre Máquina Virtual e Contêiner.....	35
4.2.3	Contêineres do Ponto de Vista do Sistema Operacional.....	35
4.2.4	Benefícios da Utilização de Contêineres.....	36
4.2.5	Seção 1 – Teoria: Contêineres Docker.....	36
4.2.6	Conceito de Docker.....	37
4.2.7	Arquitetura do Docker.....	37
4.2.8	Funcionamento do Docker.....	37
4.4	Módulo 2: Práticas com Docker na AWS.....	38
4.4.1	Seção 1 – Primeiros Passos na AWS: Instância EC2.....	39
4.4.2	Seção 2 – Configurando Contêiner Docker na Instância EC2.....	42
4.4.3	Seção 3 – Projeto Docker.....	46
4.5	Módulo 3: Kubernetes.....	50
4.5.1	Seção 1 – Introdução ao Kubernetes	50

4.5.2	Seção 2 – Implementando Kubernetes do Zero ao Deploy.....	51
4.5.3	Seção 3 – Kubernetes na AWS.....	54
5	RESULTADOS E DISCUSSÕES.....	59
5.1	Formulário diagnóstico.....	59
5.1.1	Respostas ao Questionário.....	57
5.2	Formulário final.....	62
5.2.1	Respostas ao Questionário.....	62
6	CONSIDERAÇÕES FINAIS.....	68
6.1	Trabalhos futuros.....	68
	REFERÊNCIAS.....	70
	APÊNDICE - METODOLOGIAS DOS ARTIGOS INSPIRADAS PARA A	
	REALIZAÇÃO DA INTERVENÇÃO SOBRE O USO DE CONTÊINERES E AWS...73	

1 INTRODUÇÃO

Durante o processo de ensino-aprendizagem são necessárias estratégias para o desenvolvimento pleno dos discentes. Em especial, na realização de ações como cursos de ensino e ou de extensão, é essencial um planejamento adequado para atender aos objetivos de aprendizagem propostos, incluindo a definição de metodologia e avaliações adequadas, assim como o estabelecimento de resultados a serem alcançados ao se realizar essas ações. É nesse contexto que o presente trabalho se insere. O objetivo do trabalho é realizar uma intervenção metodológica utilizando Computação em Nuvem no ensino de Contêineres Docker, Kubernetes, utilizando a AWS como infraestrutura. Para buscar melhores formas de realizar o curso, foi realizada uma Revisão Sistemática da Literatura (RSL), seguindo os procedimentos metodológicos definidos por Kitchenham (2004).

A RSL buscou responder a três Questões de Pesquisas (QPs) principais: “QP₁: Quais são as metodologia aplicadas em relatos de experiência sobre a adoção de cursos de ensino/extensão?”, “QP₂: Como os cursos de ensino/extensão são avaliados nos relatos de experiências apresentados?” e “QP₃: Quais são os principais resultados buscados nos relatos de experiência sobre a execução dos cursos de ensino/extensão?”, buscando nas bases de dados Portal de Periódicos da Capes, *ACM Digital Library*, *ScienceDirect* e *IEEE Xplore*.

A RSL apontou usos de diferentes metodologias (RQ₁), tais como metodologias ativas, a exemplo da sala de aula invertida e aprendizagem baseada em projetos. Também foram identificadas diferentes formas de avaliação das ações (RQ₂), tais como a análise do desempenho dos estudantes referente ao desenvolvimento de habilidades e competência e a adaptação às práticas pedagógicas. Os principais resultados investigados nas ações (RQ₃) incluem a avaliação da aprendizagem e o desenvolvimento de profissionais críticos.

Baseado no resultado da RSL, a intervenção foi desenvolvida. Ela busca levar temas atuais, tais como Contêineres Docker e Kubernetes com infraestrutura AWS. O curso foi desenvolvido no período de 08 de julho a 22 de julho de 2025 e contou com a realização de 5 encontros formativos. Nos encontros formativos, os conceitos teóricos foram apresentados, com o uso de práticas sobre a temática. Ao final de cada aula era disponibilizado um vídeo com o tema abordado em sala de aula. Os alunos poderiam utilizar esse material para desenvolver um projeto proposto. Durante a execução do curso, os alunos poderiam também tirar dúvidas do professor ministrante, na aula posterior, em um grupo formativo ou pelo agendamento de momento com o professor ministrante, seguindo elementos da metodologia

ativa sala de aula invertida. Em uma aprendizagem ativa, o discente assume o papel de protagonista no processo de ensino-aprendizagem, em que, seja por meio de atividades individuais ou em grupo, o processo de ensino-aprendizagem propicia a descoberta pelos estudantes (Gemelli; Santos; Nunes, 2018, p. 131). O aluno, enquanto protagonista, enriquece seu aprendizado por meio da execução de atividades práticas.

A temática da computação em nuvem foi escolhida por está amplamente utilizada no mercado de trabalho. O ensino de Contêineres Docker, orquestração com Kubernetes, utilizando a AWS como plataforma de infraestrutura pode funcionar como um meio no processo de ensino para alunos, uma vez que essa tecnologia é bastante empregada em componentes curriculares da Universidade Federal Rural do Semi-Árido (UFERSA), Campus Pau dos Ferros.

1.1 Justificativa

O ensino de novas tecnologias como contêineres é importante, uma vez que está cada vez mais presente no mercado de trabalho. Dessa forma, conhecer o tema pode enriquecer os currículos dos alunos, possibilitando novas perspectivas de atuação profissional, conhecimento e habilidades técnicas, além de ser uma temática utilizada em uma ampla gama de componentes curriculares dos cursos de Bacharelado Interdisciplinar em Tecnologia da Informação (BITI), Engenharia de Software (ES) e Engenharia de Computação (EC), tais como Sistemas Distribuídos, Dependabilidade e Segurança e Desenvolvimento de Aplicações Web, podendo atuar para reduzir insucessos nessas componentes, embora a estrutura curricular desses cursos não disponha de uma componente curricular voltada para computação em nuvem, uma área em constante expansão e com alta demanda no mercado, no momento da escrita desse trabalho, o que reforça a necessidade de iniciativas como a apresentada.

1.2 Objetivos

O objetivo deste trabalho tem como objetivo desenvolver uma intervenção metodológica para o ensino de Computação em Nuvem, com os Contêineres Docker, Kubernetes. O conteúdo do minicurso aborda o uso do sistema operacional *Linux*, além de

tecnologias como o *GitHub* e aplicações em *Python*, que já estão presentes em projetos de componentes curriculares do curso de BITI, ES e EC, introduzindo também novas tecnologias capazes de enriquecer o currículo dos alunos, tais como *Registry Docker Hub*, *Kubernetes*, comandos avançados e serviços da *AWS*.

1.3 Organização do trabalho

O presente trabalho está estruturado da seguinte maneira: no Capítulo 1, foi introduzido o trabalho, apresentando os objetivos e justificativas para o desenvolvimento do trabalho; no Capítulo 2, é apresentado o referencial teórico; no Capítulo 3, detalha-se a RSL, que busca levantar relatos de experiências sobre cursos de ensino/extensão, buscando entender as metodologias utilizadas em cursos de ensino/extensão, como os procedimentos avaliativos ocorrem, assim como os principais resultados buscados; no Capítulo 4, é apresentada a intervenção metodológica, apresentando o minicurso desenvolvido; no Capítulo 5, são apresentados os resultados e discussões, incluindo os resultados dos formulários aplicados antes e após o minicurso; por fim, o Capítulo 6, reúne as considerações finais do projeto, bem como sugestões de melhorias e direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão discutidos alguns conceitos e termos importantes para o entendimento deste trabalho.

2.1 Contêineres

Contêineres são unidades executáveis de *software* que empacotam o código da aplicação juntamente com suas bibliotecas e dependências. Eles permitem que o código seja executado em qualquer ambiente de computação, seja *desktop* ou infraestrutura de nuvem (Stephanie Susnjara; Ian Smalley).

O contêiner permite a execução de um processo ou grupo de processos de forma totalmente isolada dentro de um mesmo sistema operacional. Do ponto de vista do sistema operacional (SO), trata-se de um processo isolado que executa com recursos (CPU, memória, rede, disco) separados, como se fosse um sistema independente, mas ainda compartilhando o mesmo *kernel do host*.

A implementação de contêiner utiliza diversos recursos do kernel Linux, incluindo *cgroups* e *namespaces*.

As principais tecnologias do contêiner incluem:

- *Namespaces*: isolam recursos (processos, rede, sistema de arquivos) para o contêiner.
- *cGroups (Control Groups)*: limitam e controlam o uso de recursos (CPU, memória, disco).
- *Capabilities*: Permissões granulares do root.
- *OverlayFS*: sistema de arquivos que permite compartilhar camadas entre contêineres, economizando espaço e facilitando atualizações.

2.2 Docker

O Docker é uma plataforma de código aberto que permite ao desenvolvedor construir, implementar, executar, atualizar e gerenciar contêineres em ambientes isolados e portáteis. Essa tecnologia facilita o desenvolvimento ao garantir que os aplicativos funcionem de

maneira consistente em diferentes ambientes, desde máquinas de desenvolvimento até servidores de produção. É importante destacar que contêiner não é sinônimo de Docker. Docker é uma plataforma que utiliza a tecnologia de contêineres, mas existem outras ferramentas e *runtimes* que também fazem uso desse conceito de contêiner.

Dentre sua arquitetura destacam-se:

- Imagem: O resultado da construção, um modelo que pode ser usado para criar contêineres.
- Dockerfile: Um *script* que define como construir uma imagem.
- Contêiner: Uma instância em execução da imagem, que pode ser executada em qualquer ambiente com Docker instalado.
- Registry: É um local onde ficam armazenadas as imagens Docker, como um repositório.

2.3 Docker hub

O Docker Hub é um serviço fornecido pelo Docker para localizar e compartilhar imagens de contêineres. Trata-se de um repositório de imagens que permite hospedar um número ilimitado de repositórios públicos e privados, facilitando contribuições de código aberto e o engajamento da comunidade.

O Docker Hub oferece a capacidade de acessar facilmente os conteúdos mais relevantes para dar suporte a aplicativos mais confiáveis, auxiliando na distribuição e no gerenciamento eficiente de imagens de contêineres ao longo de todo o processo de desenvolvimento.

2.4 Kubernetes

Kubernetes é uma plataforma de orquestração de contêineres de código aberto. Ela agenda e automatiza tarefas essenciais para o gerenciamento de arquiteturas baseadas em contêineres, incluindo a execução de vários contêineres em múltiplos servidores. Entre essas tarefas, destacam-se: a implementação de contêineres, atualizações, descoberta de serviços,

provisionamento de armazenamento, balanceamento de carga, monitoramento de integridade, substituição automática de contêineres com falhas, escalabilidade automática e atualizações contínuas sem indisponibilidade (*rolling updates*).

Principais funcionalidades:

- Distribuição de contêineres entre máquinas (nós).
- Substituição automática de contêineres que falharam.
- Escalonamento automático de aplicações.
- Atualizações contínuas (*rolling updates*) sem interrupções.
- Monitoramento e auto recuperação do estado desejado.

Principais componentes do Kubernetes:

- Scheduler: Responsável por decidir em qual nó (máquina) um novo contêiner será executado, com base na disponibilidade de recursos e outras restrições.
- Controller Manager: Monitora o estado do cluster e realiza ajustes para garantir que ele esteja em conformidade com as especificações desejadas (por exemplo, mantendo o número correto de réplicas de um pod).
- etcd: Banco de dados distribuído que armazena o estado de todo o *cluster* Kubernetes.
- Pods: Unidade mínima de execução no Kubernetes; é onde os contêineres realmente rodam.
- Nodes (nós): Máquinas (físicas ou virtuais) responsáveis por executar os pods.
- Control Plane: Conjunto de componentes que gerencia o estado geral do *cluster*, incluindo o agendamento e o controle dos nós e *pods*.
- Service: Abstração que permite expor a aplicação com um IP fixo e balanceamento de carga entre os *pods*.

2.5 Cloud computing

A computação em nuvem é a entrega de recursos de TI sob demanda por meio da *Internet* com definição de preço de pagamento conforme o uso. Em vez de comprar, ter e manter *datacenters* e servidores físicos, você pode acessar serviços de tecnologia, como

capacidade computacional, armazenamento e bancos de dados, conforme a necessidade, usando um provedor de nuvem como a *Amazon Web Services* (AWS).

2.6 Amazon web services (AWS)

O Amazon Web Services (AWS) é uma plataforma de serviços em nuvem da Amazon que fornece infraestrutura tecnológica sob demanda, oferecendo uma gama de serviços como computação em nuvem, armazenamento, bancos de dados e segurança. Sua arquitetura escalável e flexível possibilita o desenvolvimento e hospedagem de aplicações de diversos portes com alta disponibilidade e desempenho (AWS, 2024).

Os serviços essenciais da AWS para este projeto incluem:

- Amazon EC2 (Elastic Compute Cloud): fornece servidores virtuais escaláveis.
- Amazon EKS (Elastic Kubernetes Service): permite executar, escalar e gerenciar clusters kubernetes.
- AWS IAM (Identity and Access Management): gerencia com segurança as identidades e o acesso a serviços e recursos da AWS.
- AWS Systems Manager Agent (SSM Agent): permite o gerenciamento automático de instâncias privadas EC2.
- Application Load Balancer: balanceador de carga de tráfego HTTP e HTTPS com roteamento avançado.
- Amazon Virtual Private Cloud (VPC): fornece uma rede virtual isolada logicamente.

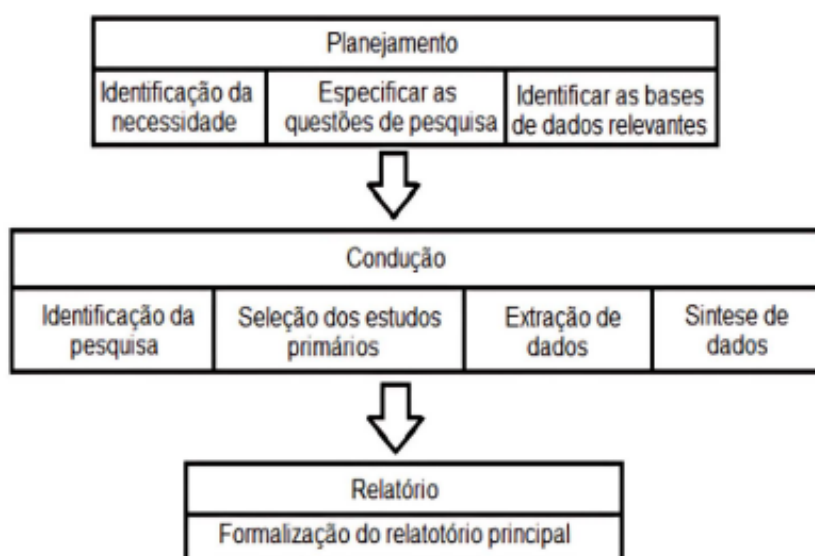
A adoção da AWS contribui diretamente para a confiabilidade, segurança e escalabilidade da implementação de contêineres na nuvem .

Essa fundamentação teórica oferece uma base sólida para o desenvolvimento, implementação e manutenção eficiente do curso na UFERSA, alinhando as melhores práticas tecnológicas com os objetivos específicos do projeto.

3 REVISÃO SISTEMÁTICA DA LITERATURA

Neste capítulo é apresentada a revisão sistemática da literatura sobre relatos de experiências sobre a aplicação de cursos introdutórios. A revisão sistemática desenvolvida foi baseada em Kitchenham (2004). Conforme apresentado em Feitoza (2024), uma RSL é dividida em três partes principais (Figura 1): Planejamento, Condução e Relatório.

Figura 1 - Etapas de uma RSL



Fonte: Feitoza (2024)

Na fase de Planejamento são identificadas as necessidades para a revisão, a especificação das questões de pesquisa e a identificação das bases de dados relevantes. A necessidade da RSL é a de fomentar um melhor planejamento da execução de ações acadêmicas, em formato de curso. Para atender a esse objetivo, foram elencadas as Questões de Pesquisa (QPs), a seguir:

- QP₁: Quais são as metodologia aplicadas em relatos de experiência sobre a adoção de cursos de ensino/extensão?
- QP₂: Como os cursos de ensino/extensão são avaliados nos relatos de experiências apresentados?
- QP₃: Quais são os principais resultados buscados nos relatos de experiência sobre a execução dos cursos de ensino/extensão?

Para responder às questões de pesquisa, foi elaborada a *string* de busca a seguir:

string de busca: (relato) AND (curso OR disciplina) AND (inicia OR introdut)

A string de busca foi aplicada às bases de dados do Portal de Periódicos da Capes, da ACM Digital Library, da ScienceDirect e da IEEE Xplore.

Na condução da RSL, foram selecionados relatos de experiências que oferecessem respostas para as questões de pesquisa. Durante a extração dos dados, foram coletadas informações básicas sobre os trabalhos, tais como autores, veículo de publicação, ano de publicação, etc. Além dessas informações básicas, foram levantadas as respostas para cada uma das questões de pesquisa. Ao total 16 relatos de experiência se encaixaram dentro do escopo do trabalho e foram utilizados para responder às questões de pesquisas, detalhadas nas próximas subseções.

3.1 RQ1 - Quais são as metodologias aplicadas pelos cursos de ensino na educação superior?

3.1.1 Metodologias Ativas Tradicionais

A literatura vem apresentando o uso constante de metodologias ativas de aprendizagem. Algumas dessas metodologias já são comumente aplicadas como metodologia de ensino, a exemplo da Sala de Aula Invertida (*Flipped Classroom*) e aprendizagem baseada em problemas (ABP). Almeida e Miguel (2020) defendem o uso dessas metodologias, destacando que elas favorecem a formação crítica e reflexiva dos estudantes, proporcionando a autonomia dos discentes, a participação ativa no processo de ensino-aprendizagem e podem envolver o uso de tecnologias digitais que estimulam a criatividade, o trabalho em equipe e a resolução de problemas reais.

Dentre eles, Silveira *et al.* (2020) empregam a sala de aula invertida, propondo o estudo antecipado pelo aluno dos conteúdos a serem abordados em sala de aula, por meio de materiais como vídeo-aulas, fazendo com que os encontros com o professor sejam destinados para atividades práticas, esclarecimento de dúvidas, trabalhos em grupo e avaliações. Essa metodologia valoriza o protagonismo do aluno na construção do conhecimento, promovendo a interação social e o uso de tecnologias digitais de informação e comunicação para apoiar o aprendizado (Silveira *et al.*, 2020). Vasconcelos *et al.* (2019) apontam que essa estratégia visa estimular a participação do discente de forma mais ativa no processo de ensino-aprendizagem,

tornando-o protagonista do próprio aprendizado, invertendo a lógica da sala de aula tradicional, em que o estudante acessa conteúdos fora da sala (vídeo-aulas, leituras) e utiliza o momento presencial para discussões, resolução de problemas e atividades práticas, com o professor atuando como mediador e orientador.

Nessa perspectiva, Azemi, Bodek e Chinn (2013) empregaram uma sala de aula invertida. No processo de ensino e recursos, o curso incluiu um livro-texto para referência e tarefas de casa e vídeos curtos pré-gravados disponibilizados semanalmente. Os estudantes deveriam assistir aos vídeos, ler o livro e realizar uma avaliação de prontidão antes de cada aula síncrona (para medir o aprendizado da parte assíncrona). As sessões síncronas foram desenhadas para enfatizar tópicos que 40% dos estudantes erraram na primeira entrega, empregando também questionários semanais de programação *online* (assíncronos).

Ainda nessa perspectiva, Colvara e Santo (2019) empregam uma metodologia exploratória e descritiva, com abordagem qualitativa, tendo como procedimento de coleta de dados um levantamento com estudantes do turno noturno de uma instituição de ensino superior (IES) localizada na Região Sul do Brasil. O estudo investiga a implementação da metodologia ativa da sala de aula invertida, fundamentada nos princípios do ensino híbrido, que combina momentos presenciais e virtuais mediados pelas tecnologias digitais de informação e comunicação (TDIC). Os dados foram obtidos por meio da aplicação de questionários com questões abertas e fechadas a 55 estudantes de distintos cursos e semestres da IES.

Já Oliveira-Barreto *et al.* (2017) empregam abordagem da aprendizagem baseada em problemas (ABP), uma metodologia centrada no estudante e baseada na solução de problemas. O curso estudado (de Fonoaudiologia) visa analisar mudanças metodológicas, pautadas em uma discussão sobre as demandas surgidas nos processos de reestruturação de currículos, utilizando estratégias que exigem habilidades comunicativas orais e escritas, bem como a assunção de diferentes papéis no processo de aprendizagem.

3.1.2 Metodologias Ativas Personalizadas

Fernandes *et al.* (2020) aplicam uma abordagem centrada em práticas que promovem a integração entre teoria e prática, utilizando metodologias ativas, como estudos de casos clínicos, práticas orientadas, simulações, dinâmicas de grupo e uso de recursos tecnológicos para facilitar o aprendizado. Elencam também a monitoria acadêmica, uma estratégia que

estimula a aprendizagem colaborativa, valorizando a mediação do ensino para além da simples transmissão de conteúdo, buscando desenvolver habilidades e competências através de uma interação dinâmica entre alunos, monitores e professores (Fernandes *et al.*, 2020).

Já Garzke e Steinwand (2024) utilizam a metodologia ativa pedagogia CREATE (*Consider, Read, Elucidate hypotheses, Analyze and interpret the data, and Think of the next Experiment*), que se baseia em atividades centradas no aluno, como mapeamento de conceitos, anotação de artigos científicos, transformação de dados e interpretação de descobertas. Essa abordagem é aplicada de forma interativa e colaborativa, com foco em engajar os alunos nas práticas reais dos cientistas, mesmo em um ambiente de aprendizagem *online*. A implementação é feita por uma comunidade de aprendizagem de instrutores, que se reúne semanalmente para discutir desafios, sucessos e objetivos pedagógicos comuns.

Por sua vez, Moreira e Vieira (2020) buscam romper com o ensino tradicional e promover a autonomia do aluno, o uso de tecnologias da informação e comunicação e a integração entre teoria e prática. Adotar o modelo “aprender vendo e aprender fazendo”, em que o aluno é estimulado a criar e executar projetos práticos, com base nos conteúdos teóricos. Essa abordagem considera o aluno protagonista de sua aprendizagem, proporcionando atividades personalizadas que respeitem sua individualidade, ritmo e interesses, com acompanhamento de mentores e uso de Ambiente Virtual de Aprendizagem (AVA) com ferramentas como *webcast*, *podcast* e comunicação escrita.

3.1.2 Modelo Tecnológico

Hernandez Neto (2020) emprega o modelo tecnológico, implementado no contexto da Tecnologia da Educação, que é o conjunto de estratégias desenvolvidas com o propósito de tornar o processo de ensino e aprendizagem mais eficaz. A metodologia do modelo tecnológico é caracterizada por: planejamento de ensino baseado em objetivos claramente definidos; uso de múltiplas mídias e recursos didáticos diversificados; ênfase na interação ativa do aluno; valorização dos aspectos cognitivos da aprendizagem, como o pensamento crítico e a resolução de problemas. É sugerido, também, a reestruturação dos currículos para adaptar-se à mudança do ensino de ações do professor para o aluno, promovendo a autonomia e a responsabilidade dos estudantes em sua própria formação.

3.1.3 Abordagem Mista Freiriana

Ferreira *et al.* (2025) aplicam uma metodologia mista, combinando abordagens teóricas e práticas, com forte influência freireana, valorizando o contexto sociocultural dos estudantes e incentivando sua curiosidade epistemológica, para condução de um curso na área de programação. O curso adotou dinâmicas lúdicas, atividades de computação desplugada, uso de ferramentas acessíveis como o Scratch e o VisuAlg e metáforas do cotidiano para facilitar a compreensão dos conceitos de programação. Além disso, priorizou-se um ambiente de ensino acolhedor e colaborativo, no qual os instrutores mantiveram uma postura não autoritária, incentivando a autonomia e a participação ativa dos estudantes.

3.1.4 Lúdico

Grossi (2017) apresentam uma metodologia fundamentada no uso do lúdico como recurso pedagógico, por meio de atividades planejadas conscientemente como jogos didáticos, dinâmicas de grupo, exibição de filmes com debates e recursos interativos como caça-palavras. Essas estratégias são utilizadas para proporcionar uma forma diferente de ensinar e aprender, tornando o processo educativo mais envolvente, reflexivo e humanizado, respeitando as diferenças cognitivas dos alunos e valorizando o desenvolvimento integral.

3.1.5 Pesquisa-Ação

Freitas D'estillac Leal e Amparo (2023) empregam a metodologia pesquisa-ação, com base em práticas formativas interativas e reflexivas. Foi desenvolvido um curso de extensão *online* com foco em professores dos Anos Iniciais, utilizando fundamentos teóricos (como os de Cosson, Petit, Morin e Pinheiro) e atividades síncronas e assíncronas. O curso seguiu o modelo da Sequência Básica de letramento literário, com etapas de motivação, introdução, leitura e interpretação, priorizando o aprendizado sensível, o diálogo e a troca de experiências.

3.1.6 Método do Caso de Ensino

Gemelli, Santos e Nunes (2018) empregam o método do caso de ensino, que utiliza situações reais discutidas abertamente para promover a aprendizagem ativa. Essa metodologia envolve o aluno como protagonista no processo, estimulando a reflexão, a tomada de decisão e a análise de diferentes perspectivas. Também são usadas ferramentas pedagógicas como textos, imagens e dramatizações para enriquecer o ensino-aprendizagem.

3.1.7 Modelo Didático

Machado *et al.* (2021) utilizam uma metodologia baseada no modelo didático, que inclui três fases principais: planejamento, execução e avaliação (Nérici, 1992). A escolha dos métodos depende da unidade entre objetivos, conteúdos e métodos, bem como das condições concretas das situações didáticas.

3.2. RQ2 - Como os cursos de ensino são avaliados na educação superior?

3.2.1 Desenvolvimento de Habilidades e Competências

Muitos artigos realizam avaliações, no sentido do desenvolvimento de habilidades e competências produzidas pela ação. Vasconcelos *et al.* (2019) apontam que os cursos da educação superior são avaliados por meio de diretrizes e indicadores que envolvem não apenas a quantidade de aulas, mas também a qualidade das metodologias empregadas, o papel do professor como facilitador e a participação efetiva dos estudantes no processo de construção do conhecimento. A avaliação considera se o curso promove o desenvolvimento de habilidades e competências compatíveis com os desafios da sociedade.

Para Hernandez Neto (2020), as ações são avaliadas em termos mais amplos, em que os cursos de ensino são avaliados conforme as exigências da comunidade acadêmica, que esperam um processo com capacidade de analisar, sintetizar e tomar decisões. O processo de avaliação educacional deve abranger aspectos como: objetivos de ensino, seleção e organização dos conteúdos, métodos e técnicas de ensino, bem como o desempenho do aluno e do professor.

Já Fernandes *et al.* (2020) apontam que os cursos são avaliados por meio do desempenho acadêmico dos alunos, que inclui atividades teóricas e práticas. A avaliação considera o processo de ensino aprendizagem, o envolvimento dos alunos em atividades complementares, como monitorias e práticas laboratoriais e indicadores institucionais, como conceitos atribuídos pelo Ministério da Educação (MEC). No caso do curso de Enfermagem da FACENE/RN, por exemplo, o conceito quatro atribuído pelo MEC reflete a qualidade do ensino oferecido.

3.2.2 Percepção dos Alunos

Outras abordagens são baseadas na percepção dos alunos sobre si mesmos ou sobre os cursos. Em seu trabalho, Garzke e Steinwand (2024) apontam que os cursos são avaliados por meio de pesquisas de autoavaliação dos alunos, especialmente, com o instrumento SAAB (*Survey of Attitudes and Abilities in Biology*). Esse instrumento mede a confiança e auto percepção dos alunos em áreas como: Leitura e análise de literatura científica primária, interpretação de dados, leitura ativa, visualização de experimentos, pensar como cientista, compreensão do processo de pesquisa. Os dados são analisados com modelos de regressão ordinal repetida, comparando os resultados pré e pós-período, considerando também o tipo de instrutor.

Já em Colvara e Santo (2019), os cursos são avaliados a partir das percepções dos estudantes sobre o uso da metodologia da sala de aula invertida, considerando elementos como: Tempo disponível para estudo prévio, Acesso às ferramentas tecnológicas, Competência e fluência digital, Retomada de conteúdos pelo professor, Participação ativa nas atividades presenciais e virtuais. A avaliação também contempla se os estudantes entendem e concordam com as técnicas do método ou se enfrentam dificuldades relacionadas ao tempo e aos recursos. Além disso, o artigo considera critérios avaliativos sugeridos pelo *Flipped Classroom Field Guide* (2016), como devolutiva imediata, estruturação adequada do material didático, e participação como parte do processo avaliativo.

Por sua vez, em Silveira *et al.* (2020), a avaliação é realizada por meio de instrumentos de pesquisa que coletam a percepção dos alunos, como questionários com escala Likert que medem o grau de concordância dos estudantes em relação a vários aspectos da metodologia aplicada, incluindo a qualidade dos materiais didáticos, a participação nas atividades presenciais, a interação social e a preferência pelo modelo de ensino. Também se

avalia a capacidade dos alunos de aplicar os conceitos estudados e desenvolver competências no ambiente presencial.

3.2.3 Adaptação de Práticas Pedagógicas

Alguns trabalhos utilizam a avaliação para adaptar as práticas pedagógicas ao longo do curso. No estudo de Machado *et al.* (2021), a terceira fase do método de ensino é a avaliação, a qual consta de provas de verificação ou de outras técnicas avaliativas que forneçam ao professor dados que o permitam uma avaliação do estudo efetuado pela classe e pelos educandos separadamente”. Essa avaliação visa identificar se os objetivos de aprendizagem foram atingidos, e possibilita retificações ou recuperação da aprendizagem, sempre que necessário.

Por sua vez, no estudo de Azemi, Bodek e Chinn (2013), a avaliação do curso foi baseada em múltiplos componentes: participação nas sessões síncronas, tarefas de casa, projetos, apresentações, *quizzes online*, avaliações de prontidão (*readiness assessments*) e exames. Além disso, no artigo é mencionado que "a parte síncrona foi ajustada com base na avaliação dos estudantes", o que mostra que o *feedback* dos alunos também foi usado para avaliar e melhorar o curso.

Já no trabalho de Gemelli, Santos e Nunes (2018), os cursos são avaliados pela participação e envolvimento dos alunos no processo de aprendizagem, pelo desenvolvimento do protagonismo estudantil e pela capacidade de aplicar conhecimentos em situações práticas. A avaliação também considera o *feedback* do docente, que reflete sobre os resultados das ações didático-pedagógicas e adapta suas práticas conforme as experiências das turmas.

3.2.4 Avaliação Continuada

Outra forma de avaliação é a continuada. Em Grossi (2017), a avaliação é deste tipo e processual, sendo baseada na participação ativa dos alunos durante as atividades lúdicas, no desempenho nas respostas às perguntas dos jogos, na capacidade de argumentação nos debates e também na interação e reflexão sobre os conteúdos propostos. As atividades

avaliativas são construídas para ir além da memorização de conceitos, focando na compreensão, na criatividade e no pensamento crítico.

Similarmente, em Ferreira *et al.* (2025), a avaliação foi feita de forma processual e contínua, com acompanhamento individual de alguns estudantes pelos instrutores ao longo das aulas. A cada encontro, os instrutores analisavam o desenvolvimento das atividades, identificando dificuldades e ajustando o ritmo do ensino conforme a necessidade da turma. Foi utilizado a ferramenta de avaliação *Moodle*, permitindo aos estudantes e professores comunicação e extensão de aprendizado.

3.2.5 Avaliação Mista

A avaliação mista também é aplicada, a exemplo do trabalho de Moreira e Vieira (2020), em que a avaliação acontece de forma diversificada e participativa, dividida em duas etapas principais: na sala de aula virtual 1, o aluno realiza atividades teóricas propostas, estuda os conteúdos indicados e é avaliado segundo métodos mais tradicionais, correspondendo a até 50% da nota. Na sala de aula virtual 2, o estudante elabora e desenvolve um projeto prático, definindo o próprio plano de estudo, ferramentas e formas de avaliação, com base em autoavaliação e comprovação da aprendizagem. Essa avaliação está diretamente ligada à relevância do projeto para a formação profissional do aluno e sua contribuição para a comunidade. O processo é acompanhado por um mentor que fornece *feedbacks* contínuos, valorizando o desenvolvimento individual e coletivo.

Por sua vez, em Oliveira-Barreto *et al.* (2017) são apresentadas duas formas principais de avaliação: Avaliação formativa, que ocorre ao longo do processo de aprendizagem, de forma diária e processual, com critérios que consideram desempenho, participação, interação, conhecimento prévio, responsabilidade e atitude crítica. Avaliação somativa, que ocorre ao final de um processo, por meio de provas teóricas e teórico-práticas, com objetivos de mensuração de resultados. Essas avaliações são aplicadas em duas atividades: Sessões Tutoriais e Habilidades Práticas.

3.2.6 Sistemas de Avaliações Autênticos

Os sistemas de avaliações autênticos também são utilizados. Em Almeida e Miguel (2020) são empregados sistemas de avaliação autênticos que mostram o que os alunos têm realmente aprendido. Além disso, o professor deve possuir grande autonomia e critério profissional e a escola deve ter currículos atualizados, flexíveis e capazes de se ligar às necessidades dos alunos. Isso demonstra que a avaliação deve ir além da simples memorização, buscando evidenciar a real construção do conhecimento.

3.2.7 Análise do Discurso

Diferentemente, dos anteriores, em Freitas D'estillac Leal e Amparo (2023), a avaliação ocorreu por meio da análise qualitativa das participações, falas e respostas dos professores em atividades práticas e reflexivas. Utilizou-se a Análise do Discurso (AD) para interpretar os sentidos atribuídos pelas participantes ao conceito de letramento literário e às práticas pedagógicas. A avaliação considerou também o engajamento, a apropriação teórica e a transformação das percepções docentes ao longo do curso.

3.3. RQ3 - Quais são os principais resultados buscados pelos cursos de ensino na educação superior?

Os resultados buscados pelos cursos são diversos. No Quadro 1, é possível ver uma sumarização daqueles encontrados ao longo do trabalho.

Quadro 1 - Relação de trabalhos que apresentam diferentes resultados buscados nas ações de formação

Resultado buscado	Trabalhos
Aprendizagem	Azemi, Bodek e Chinn (2013), Freitas D'estillac Leal e Amparo (2023), Grossi (2017), Moreira e Vieira (2020), Silveira <i>et al.</i> (2020), Fernandes <i>et al.</i> (2020), Machado <i>et al.</i> (2021) e Garzke e Steinwand (2024).
Pensamento crítico	Vasconcelos <i>et al.</i> (2019), Hernandez Neto (2020), Fernandes <i>et al.</i> (2020), Garzke e Steinwand (2024), Machado <i>et al.</i> (2021) e Freitas D'estillac Leal e

	Amparo (2023).
Autonomia	Fernandes <i>et al.</i> (2020), Machado <i>et al.</i> (2021), Moreira e Vieira (2020), Gemelli, Santos e Nunes (2018) e Silveira <i>et al.</i> , (2020).
Trabalho em equipe	Hernandez Neto (2020), Garzke e Steinwand (2024), Gemelli, Santos e Nunes (2018), Silveira <i>et al.</i> (2020) e Vasconcelos <i>et al.</i> (2019).
Transformação social e humana	Machado <i>et al.</i> (2021), Almeida e Miguel (2020), Freitas D'estillac Leal e Amparo (2023), Machado et al. (2021) e Fernandes <i>et al.</i> (2020).
Engajamento	Garzke e Steinwand (2024), Ferreira <i>et al.</i> (2025), Azemi, Bodek e Chinn (2013) e Grossi (2017).
Mercado de trabalho	Fernandes <i>et al.</i> (2020), Moreira e Vieira (2020), Gemelli, Santos e Nunes, 2018) e Hernandez Neto (2020).
Domínio de ferramentas tecnológicas	Almeida e Miguel (2020), Colvara e Santo (2019), Azemi, Bodek e Chinn (2013) e Fernandes et al. (2020)
Protagonismo	Vasconcelos <i>et al.</i> (2019), Gemelli, Santos e Nunes (2018) e Silveira <i>et al.</i> (2020).
Resolução de problemas	Hernandez Neto (2020) e Fernandes <i>et al.</i> (2020).
Aumentar a interatividade	Azemi, Bodek e Chinn (2013)
Formação contínua que vá ao encontro da realidade e das necessidades atuais dos professores	Almeida e Miguel (2020).
Desenvolvimento da criatividade	Silveira et al. (2020)
Ressignificar essa abordagem metodológica para que se torne mais inclusiva e equitativa	Colvara e Santo (2019)
Melhoria da interação social entre alunos e professores	Silveira <i>et al.</i> (2020)
Construir um curso de	Azemi, Bodek e Chinn (2013)

programação	
Comparar a avaliação somativa e a formativa	Oliveira-Barreto <i>et al.</i> (2017)
Desenvolver o raciocínio lógico em programação	Ferreira <i>et al.</i> (2025).
Pertencimento para a área STEM.	Garzke e Steinwand, 2024).
Desenvolver práticas inovadoras	Freitas D'estillac Leal e Amparo (2023).
Desenvolver a habilidade de análise científica	Garzke e Steinwand (2024)
Percepção dos alunos	Colvara e Santo (2019).
Compreender a realidade dos estudantes	Colvara e Santo (2019).
Desenvolvimento contínuo	Fernandes <i>et al.</i> (2020).
Adaptabilidade	Hernandez Neto (2020).
Responsabilidade	Silveira <i>et al.</i> (2020).
Desenvolvimento da confiança	Garzke e Steinwand (2024).
Qualidade de ensino	Vasconcelos <i>et al.</i> (2019).

Fonte: Autoria própria (2025)

Como pode ser visto no Quadro 1, os trabalhos têm focado consideravelmente em desenvolver a aprendizagem. Na sequência, outra habilidade bastante buscada pela literatura é o desenvolvimento do pensamento crítico. A autonomia e o trabalho em equipe são também bastante buscados, assim como a formação social e humana, o engajamento, o desenvolvimento de competências para atuação no mercado de trabalho. Além desses, também há interesse considerável da literatura para a formação para uso de ferramentas tecnológicas e o protagonismo do estudante no centro da aprendizagem.

4 INTERVENÇÃO METODOLÓGICA

Este trabalho descreve a intervenção pedagógica realizada por meio do desenvolvimento e aplicação de um minicurso voltado aos alunos da UFERSA. O minicurso teve como foco introduzir conceitos como Docker, Kubernetes e AWS entre outros serviços, para o ensino de metodologias ativas de aprendizagem. O desenvolvimento do curso tomou como base a RSL apresentada no Capítulo 3. No Apêndice A, pode ser visto como o curso foi baseado, considerando por base informações de diferentes trabalhos.

Este capítulo está estruturado da seguinte forma: a Seção 4.1 descreve a estrutura geral do minicurso; a Seção 4.2 apresenta a preparação dos alunos por meio do formulário diagnóstico; a Seção 4.3 detalha os módulos teóricos e práticos desenvolvidos ao longo das aulas; e a Seção 4.4 apresenta a finalização do curso com aplicação de formulário avaliativo e considerações pedagógicas.

4.1 Estrutura do minicurso

O minicurso foi desenvolvido com o objetivo de apresentar conceitos fundamentais e práticos sobre containerização, Docker, Kubernetes e sua aplicação em ambientes de computação em nuvem, especialmente, na AWS. Foi realizado um estudo sobre a metodologia ativa de aprendizagem denominada Sala de Aula Invertida (*Flipped Classroom*) (Vasconcelos *et al.*, 2018).

No contexto da Sala de Aula Invertida, os alunos assistiam a vídeos gravados com o conteúdo teórico e levavam suas dúvidas para os encontros síncronos realizados por meio do Google Meet. Nessas aulas ao vivo, eram apresentados os laboratórios práticos. Ao final de cada encontro, o material da aula era disponibilizado aos participantes a gravação da aula, conteúdos teóricos e atividades práticas. O curso teve duração de duas semanas, com aulas síncronas às terças e quintas-feiras.

A seguir, detalha-se a estrutura adotada nas aulas e nas atividades práticas.

4.2 Módulo 1: Introdução à containerização

Nesta etapa inicial do minicurso, foi apresentada aos alunos uma base conceitual sobre containerização, com foco nos fundamentos necessários para a compreensão e uso do Docker. A abordagem foi pautada na exposição teórica e em atividades práticas que incentivam a autoavaliação contínua por parte dos participantes. Inicialmente, alguns conceitos básicos foram apresentados, destacados na sequência.

4.2.1 Conceito de Contêiner

Contêineres são unidades executáveis que empacotam o código de uma aplicação juntamente com suas bibliotecas e dependências, permitindo que ela seja executada de forma isolada, consistente e padronizada em qualquer ambiente, independentemente do sistema operacional subjacente. Essa tecnologia facilita a replicação de ambientes de desenvolvimento e produção, reduzindo erros e incompatibilidades durante o ciclo de vida da aplicação.

4.2.2 Diferença entre Máquina Virtual e Contêiner

Embora ambos proporcionam isolamento, contêineres diferem das máquinas virtuais por compartilharem o mesmo *kernel* do sistema operacional *host*, enquanto as máquinas virtuais requerem a execução de sistemas operacionais independentes por meio de hipervisores. Essa característica torna os contêineres mais leves e rápidos de inicializar, com menor consumo de recursos computacionais.

4.2.3 Contêineres do Ponto de Vista do Sistema Operacional

Do ponto de vista do sistema operacional, um contêiner é um processo isolado que executa com recursos (como CPU, memória, rede e armazenamento) segregados, mas ainda compartilha o mesmo *kernel* do *host*. Essa execução isolada é possível graças a recursos nativos do Linux, como:

- *Namespaces*: responsáveis por isolar recursos como processos, rede, sistema de arquivos e usuários.

- *cGroups (Control Groups)*: permitem controlar e limitar o uso de recursos do sistema por parte dos contêineres.
- *Capabilities*: oferecem permissões granulares ao processo containerizado, substituindo o conceito tradicional de privilégios de superusuário.
- *OverlayFS*: sistema de arquivos que permite empilhar múltiplas camadas, otimizando o uso de espaço e facilitando atualizações de imagem.

4.2.4 Benefícios da Utilização de Contêineres

Durante a aula, foram discutidos os benefícios da adoção de contêineres no desenvolvimento e implantação de aplicações, tais como:

- Isolamento: evita conflitos entre diferentes aplicações e suas dependências.
- Portabilidade: possibilita que aplicações funcionem da mesma forma em ambientes distintos (desenvolvimento, testes e produção).
- Agilidade: os contêineres iniciam rapidamente, otimizando o tempo de execução e resposta.
- Escalabilidade: facilitam o gerenciamento de múltiplas instâncias de aplicação, especialmente em ambientes de orquestração como o Kubernetes.

4.2.5 Seção 1 – Teoria: Contêineres Docker

Ainda na primeira aula do minicurso, o objetivo foi apresentar aos alunos os fundamentos teóricos sobre Docker e containerização. A aula teórica foi ministrada de forma expositiva, com espaço para dúvidas e intervenções práticas, visando à construção do conhecimento.

Nesse momento, os estudantes foram incentivados a responder a questionamentos com o intuito de expressar seu nível de compreensão sobre o conteúdo e as atividades que seriam realizadas em seguida, reforçando, assim, a proposta de metodologias ativas.

4.2.6 Conceito de Docker

O Docker é uma plataforma de containerização que permite empacotar, distribuir e executar aplicações em ambientes isolados e consistentes. A principal característica do Docker é o uso compartilhado do *kernel* do sistema operacional *host*, o que o torna mais leve e eficiente em comparação com máquinas virtuais tradicionais, que requerem a emulação de sistemas operacionais completos.

4.2.7 Arquitetura do Docker

A arquitetura do Docker foi apresentada aos alunos por meio de uma estrutura composta por três elementos principais:

- **Imagem (*Image*):** Representa um modelo estático contendo todos os recursos necessários para a execução de uma aplicação, incluindo bibliotecas, dependências, arquivos e configurações.
- **Contêiner:** É a instância em execução de uma imagem. A partir de uma imagem previamente construída, um contêiner é iniciado para rodar a aplicação de forma isolada.
- ***Registry* (Registro):** Repositório de imagens Docker, responsável por armazenar e distribuir essas imagens. Foram apresentados aos alunos exemplos como:
 - Docker Hub (registro público e gratuito);
 - AWS Elastic Container Registry (ECR);
 - Azure Container Registry (ACR);
 - Google Container Registry (GCR);
 - Docker Trusted Registry (DTR) – utilizado em ambientes corporativos on-premises ou em nuvem.

4.2.8 Funcionamento do Docker

O funcionamento do Docker foi explorado a partir da criação de imagens e contêineres. A imagem é construída com base em um Dockerfile, um *script* que define, de forma declarativa, os passos necessários para montar o ambiente da aplicação.

- Dockerfile: Arquivo de configuração que especifica instruções para a criação de uma imagem.
- Imagem: Resultado da construção baseada no Dockerfile, armazenável localmente ou em um *registry*.
- Contêiner: Execução ativa de uma imagem, podendo ser replicado em diferentes ambientes com Docker instalado.

Também foi demonstrado como utilizar o Docker Hub, o principal repositório público de imagens, para buscar, baixar e reutilizar imagens de aplicações populares, além de como publicar imagens criadas localmente.

Essa aula teve como foco fornecer a base necessária para as etapas práticas subsequentes, permitindo que os alunos compreendessem como a tecnologia funciona antes de aplicá-la em laboratório.

4.4 Módulo 2: Práticas com docker na AWS

Dando continuidade à primeira etapa do minicurso, foi gravada uma aula assíncrona disponibilizada previamente à aula síncrona, com o objetivo de orientar os alunos na configuração do ambiente AWS, possibilitando que levassem suas dúvidas para o momento ao vivo.

Os alunos foram orientados a configurar instâncias EC2 utilizando o AWS *Systems Manager* (SSM) para acesso via *Session Manager*, eliminando a necessidade de IP público, o que também permitiu abordar boas práticas de segurança em ambientes de nuvem.

Foi disponibilizado aos participantes usuários do IAM para que os mesmos pudessem configurar as permissões necessárias e utilizar uma VPC preconfigurada. Durante a atividade, surgiram questionamentos espontâneos sobre a criação de VPCs do zero, o que evidenciou o engajamento dos alunos e possibilitou discussões adicionais sobre redes na nuvem.

Todos os serviços foram configurados nas regiões us-east-1 (N. Virginia) e us-east-2 (Ohio).

4.4.1 Seção 1 – Primeiros passos na AWS: Instância EC2

Nesta atividade, os alunos realizaram a configuração de uma instância EC2 em uma *subnet* privada, com o suporte do *AWS Systems Manager* (SSM) para o acesso remoto. Foram trabalhados os seguintes conceitos e serviços:

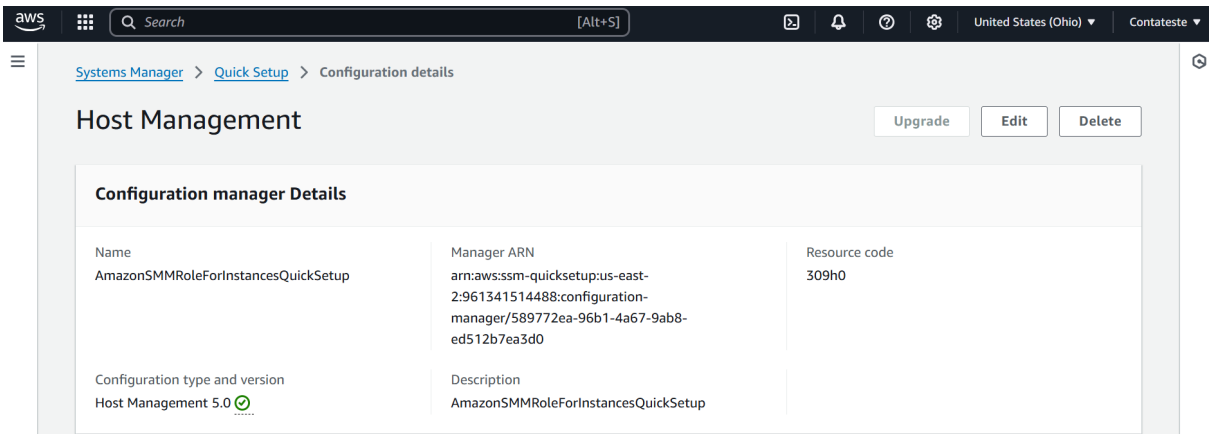
- Configuração do AWS Systems Manager (SSM);
- Criação de Role IAM para gerenciamento de permissões;
- Implantação de NAT Gateway;
- Edição de tabelas de rotas;
- Lançamento e acesso de uma instância EC2 privada.

AWS SYSTEMS MANAGER (SSM)

O *AWS Systems Manager* é um serviço que permite o gerenciamento centralizado e seguro de recursos da nuvem, como instâncias EC2. O *Session Manager*, recurso do SSM, foi utilizado para conexão remota às instâncias sem a necessidade de IP público ou chave SSH, reforçando práticas modernas de segurança em cloud computing.

A configuração foi realizada via *Quick Setup*, com a seleção do perfil de *Host Management*, e instalação automatizada do *CloudWatch Agent* para monitoramento da instância. Os detalhes da configuração podem ser vistos na Figura 1.

Figura 1 - Host Management Configurado

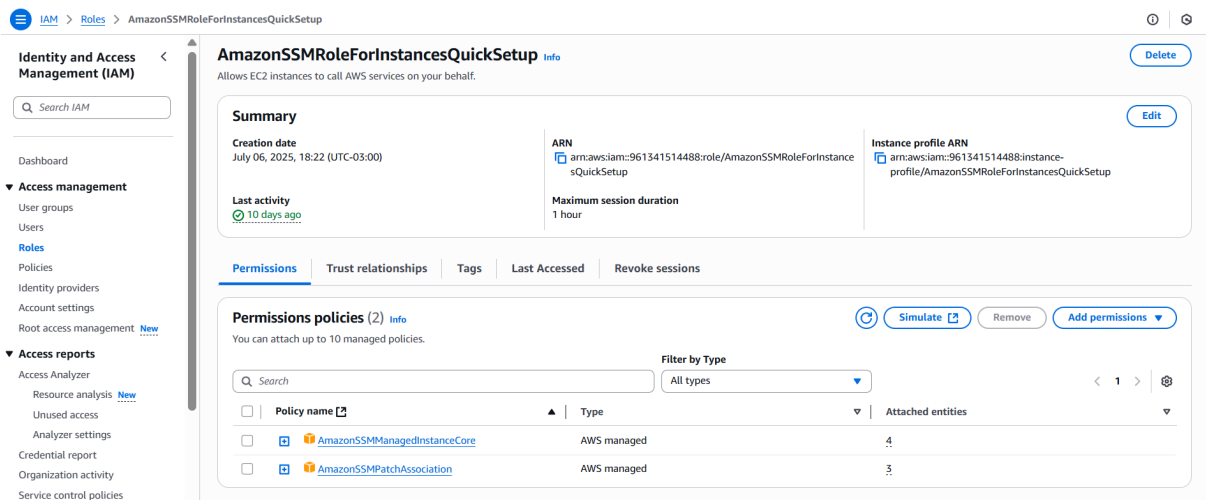


Fonte: Adaptado do console AWS (2025)

IAM ROLE PARA SSM

Foi demonstrada a criação de uma *Role IAM* com permissões adequadas para que a instância EC2 pudesse se registrar no *Systems Manager*. A política *AmazonSSMManagedInstanceCore* e *AmazonSSMPatchAssociation* foram atribuídas à *role*, nomeada como *AmazonSMMRoleForInstancesQuickSetup*, respeitando o princípio do menor privilégio e possibilitando que as instâncias fossem gerenciadas de forma segura (Figura 2).

Figura 2 - Configurações de *Roles* no IAM



Fonte: Adaptado do console AWS (2025)

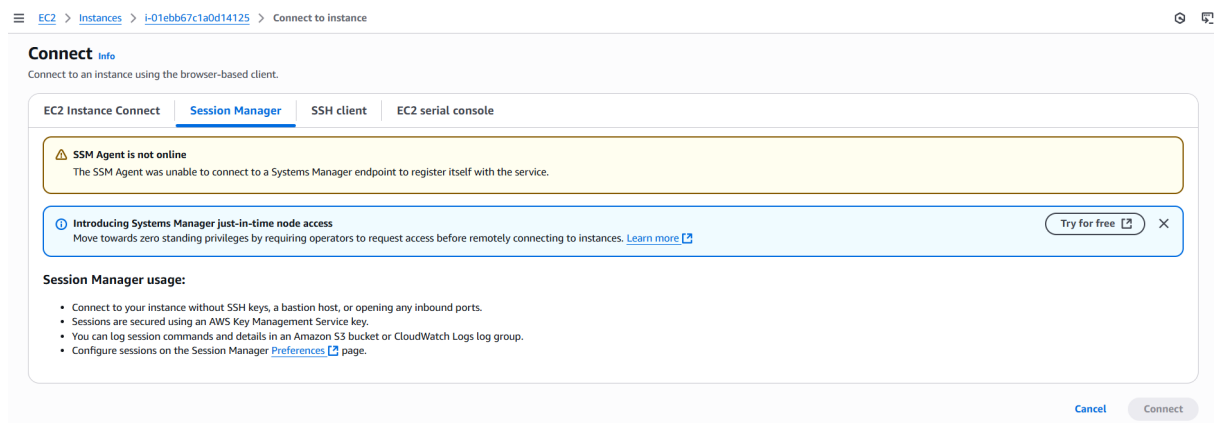
CRIAÇÃO DA INSTÂNCIA EC2 EM *SUBNET* PRIVADA

A criação da instância seguiu as seguintes etapas:

- Seleção da *Amazon Machine Image (AMI)*, Ubuntu;
- Escolha do tipo de instância (t2.micro);
- Associação à *VPC* e subnet privada;
- Desativação da atribuição de *IP público*;
- Seleção da *IAM Role* configurada anteriormente;
- Criação de *Security Groups* com regras mínimas necessárias;
- Definição de armazenamento e criptografia com o tipo de *volume gp3*

Após o lançamento, observou-se que a conexão via *Session Manager* ainda não estava disponível, evidenciando a necessidade de configurar o *NAT Gateway* e ajustar as tabelas de rotas da VPC (Figura 3).

Figura 3 - Conexão de Instância



Fonte: Adaptado do console AWS (2025)

NAT GATEWAY E TABELAS DE ROTAS

Para possibilitar que a instância privada acessasse a internet (por exemplo, para instalar pacotes Docker), foi criado um *NAT Gateway* em uma *subnet* pública, associado a um

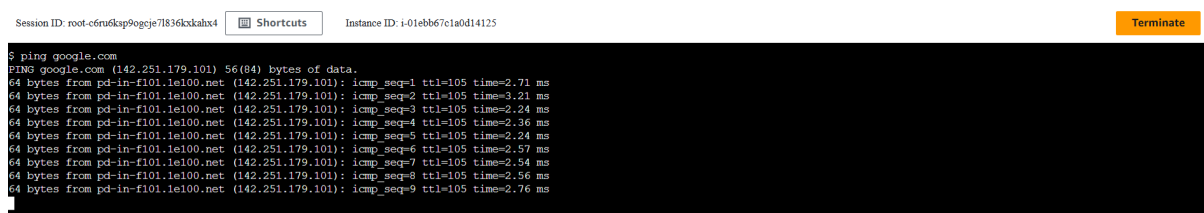
Elastic IP. Em seguida, a tabela de rotas da *subnet* privada foi editada para direcionar o tráfego externo (0.0.0.0/0) ao NAT Gateway criado.

As *subnets* privadas foram associadas à nova tabela de rotas, completando a configuração de saída para a internet de forma segura.

VERIFICAÇÃO E CONECTIVIDADE

Por fim, os alunos acessaram a instância privada via *Session Manager* e testaram a conectividade com a internet por meio do comando `ping google.com`. A conexão foi bem-sucedida, demonstrando o funcionamento do *NAT Gateway* e permitindo a continuação das atividades com instalação do Docker (Figura 4).

Figura 4 - Testando Conexão de Instância



The screenshot shows a terminal window with the following text:

```
Session ID: root-cfrufkap9egeje7l836kxkax4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate
```

```
$ ping google.com
PING google.com (142.251.179.101) 56(84) bytes of data:
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=1 ttl=105 time=2.71 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=2 ttl=105 time=3.21 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=3 ttl=105 time=2.24 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=4 ttl=105 time=2.36 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=5 ttl=105 time=2.24 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=6 ttl=105 time=2.57 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=7 ttl=105 time=2.54 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=8 ttl=105 time=2.56 ms
64 bytes from pd-in-f101.1e100.net (142.251.179.101): icmp_seq=9 ttl=105 time=2.76 ms
```

Fonte: Adaptado do console AWS (2025)

4.4.2 Seção 2 – Configurando Contêiner Docker na Instância EC2

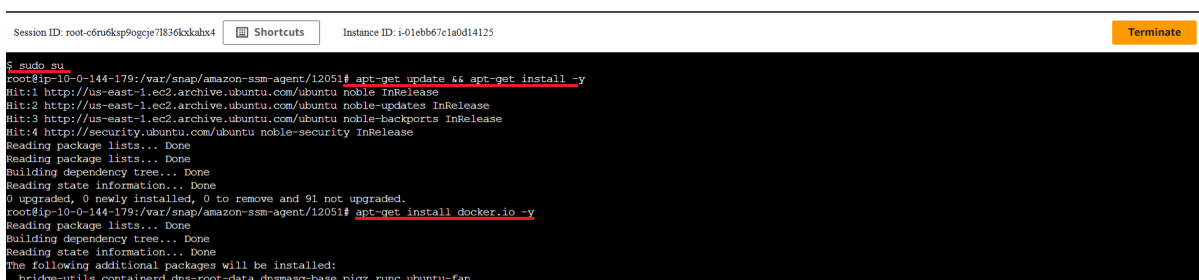
Nesta etapa do minicurso, os alunos realizaram a instalação do Docker na instância EC2 previamente configurada e exploraram os comandos principais para gerenciar imagens e contêineres. O objetivo central da atividade foi permitir o contato prático com a execução de aplicações containerizadas, consolidando os conhecimentos teóricos abordados anteriormente.

A atividade foi conduzida diretamente no terminal da instância. A execução prática incluiu desde a instalação do Docker até a criação de um contêiner Nginx e personalização de sua página principal, com o intuito de demonstrar a operacionalização de serviços containerizados em ambiente de nuvem.

INSTALAÇÃO E CONFIGURAÇÃO DO DOCKER

Para iniciar, os alunos utilizaram o comando `sudo su` para acessar o terminal como super usuário, possibilitando a instalação do Docker com permissões administrativas (Figura 5). A instalação foi realizada utilizando o gerenciador de pacotes `apt-get`:

Figura 5 - Principais Comandos básicos do Linux



```
Session ID: root-c6n6ksp9ogcje7l836kckahx4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate
$ sudo su
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# apt-get update && apt-get install -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# apt-get install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
```

Fonte: Adaptado do console AWS (2025)

COMANDOS BÁSICOS DE GERENCIAMENTO DE CONTÊINERES

Com o Docker instalado, os estudantes passaram a executar uma sequência de comandos básicos que permitiram visualizar, executar, interromper e remover **contêineres**, além de manipular imagens (Figura 6). Os principais comandos trabalhados foram:

- Listar **contêineres** em execução :**docker ps**
- Listar todos os **contêineres**, inclusive os finalizados: **docker ps -a**
- Visualizar todas as imagens disponíveis localmente: **docker images**
- Fazer o download de uma imagem do Docker Hub: **docker pull nginx**
- Executar um contêiner com a imagem baixada: **docker run -itd -p 80:80 --name maria <ID da imagem>**

O comando acima cria um contêiner chamado maria, em segundo plano, mapeando a porta 80 do host para a porta 80 do contêiner, o que possibilita o acesso via navegador.

Figura 6 - Primeiros Comandos Docker

Session ID: root-c6ru6ksp9ogcje7l836kxkx4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

```
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
59e22667830b: Pull complete
140da4f89dcb: Pull complete
96e47e70491e: Pull complete
2ef442a3816e: Pull complete
4b1e45a9989f: Pull complete
1d9f51194194: Pull complete
f30ffbee4c54: Pull complete
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker run -itd -p 80:80 --name maria nginx
84562e9c7fc637276875e9d1ac8a2dfca7db217abf1f93d9e7273e7f7a89af0
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051#
```

Fonte: Adaptado do console AWS (2025)

PERSONALIZAÇÃO DO CONTÊINER NGINX

Um dos destaques da atividade foi a modificação da página principal do servidor Nginx rodando dentro do contêiner. Para isso, os alunos acessaram o terminal do contêiner com o seguinte comando: **docker container exec -it maria bash**.

Dentro do contêiner, o conteúdo da página inicial foi modificado para exibir a mensagem: **echo "Olá alunos UFERSA" > /usr/share/nginx/html/index.html**.

Essa ação permitiu observar, em tempo real, como alterações no ambiente do contêiner refletem no comportamento do serviço web (Figura 7).

Figura 7 - Comando para Interagir com Contêineres

Session ID: root-c6ru6ksp9ogcje7l836kxkx4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

```
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker container exec -it maria bash
root@84562e9c7fc6:/# echo Ola alunos UFERSA > /usr/share/nginx/html/index.html
root@84562e9c7fc6:/# read escape sequence
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051#
```

Fonte: Adaptado do console AWS (2025)

IMPLEMENTAÇÃO DE UM APPLICATION LOAD BALANCER (ALB) PARA TESTES PRÁTICOS

Foi criado um *Application Load Balancer (ALB)*, associado ao *Security Group* configurado anteriormente e vinculado à instância EC2. O ALB foi integrado a um *Target Group*, permitindo o roteamento automático das requisições para o contêiner Nginx, operando na porta 80.

O ALB disponibilizou um domínio público (DNS) fornecido pela AWS, permitindo que os alunos testassem a personalização da página do contêiner por meio de requisições diretas via navegador.

Essa integração com o ALB possibilitou aos alunos compreender de forma prática, como aplicações containerizadas podem ser expostas à internet com segurança e escalabilidade. A simulação de uma arquitetura, com balanceamento de carga e isolamento de rede, contribuiu significativamente para ampliar a visão dos estudantes sobre o ciclo completo de implantação de seus projetos na nuvem. A Figura 8 mostra um teste no navegador.

Figura 8 - Testando no Navegador



Fonte: Adaptado do navegador Google (2025)

GERENCIAMENTO DO CICLO DE VIDA DO CONTÊINER

Também foram explorados comandos para controle do ciclo de vida do contêiner (Figura 9):

- Parar o contêiner: **docker stop maria**
- Reiniciar o contêiner: **docker start maria**
- Remover o contêiner: **docker rm -f maria**

Figura 9 - Comandos Docker para Gerenciar Ciclo de Vida do Contêiner

```
Session ID: root-q2glpjvcrldxpqtqapz9387u Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
84562e9c7fc6   nginx    "/docker-entrypoint..." 3 hours ago Up 3 hours 0.0.0.0:80->80/tcp, :::80->80/tcp   maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker images
REPOSITORY    TAG       IMAGE ID       CREATED    SIZE
nginx         latest   2cd1d97f893f   10 days ago 192MB

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker stop maria
maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
84562e9c7fc6   nginx    "/docker-entrypoint..." 3 hours ago Exited (0) 20 seconds ago      maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker start maria
maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
84562e9c7fc6   nginx    "/docker-entrypoint..." 3 hours ago Up 5 seconds 0.0.0.0:80->80/tcp, :::80->80/tcp   maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker rm -f maria
maria

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051#
```

Fonte: Adaptado do console AWS (2025)

4.4.3 Seção 3 – Projeto Docker

A última aula prática sobre Docker teve como objetivo permitir que os alunos aplicassem os conhecimentos adquiridos sobre containerização desenvolvendo e publicando uma aplicação real. Para isso, os alunos trabalharam na criação de um contêiner personalizado, utilizando um projeto em Python hospedado no GitHub. A atividade incluiu desde a escrita de um Dockerfile, construção da imagem, até o envio dessa imagem para o Docker Hub.

Conteúdo trabalhado desta etapa foi permitir aos alunos:

- Clonar um projeto Python real do GitHub;
- Criar uma imagem Docker personalizada;
- Executar a aplicação localmente via contêiner;
- Publicar essa imagem no Docker Hub para reuso e compartilhamento.

CLONAGEM DO REPOSITÓRIO

Também foi exemplificado a clonagem de um repositório no GitHub (Figura 10).

Figura 10 - Clonando Repositório GitHub

```
Session ID: root-49ujjetudac22367telhxxpi4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# git clone https://github.com/KubeDev/conversao-distancia
Cloning into 'conversao-distancia'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 16 (delta 3), reused 1 (delta 1), pack-reused 6 (from 1)
Receiving objects: 100% (16/16), 4.81 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (3/3), done.
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051# cd conversao-distancia/
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia# ls
README.md  app.py  requirements.txt  templates
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia#
```

Fonte: Adaptado do console AWS (2025)

CRIAÇÃO DO DOCKERFILE

O Dockerfile foi criado com base na estrutura do projeto, conforme o exemplo abaixo (Figura 11):

Figura 11 - Construindo Imagem no Dockerfile

```
Session ID: root-49ujjetudac22367telhxxpi4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

GNU nano 7.2 Dockerfile *
FROM python:3.13.0
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . /app/
EXPOSE 5000
CMD [ "unicorn", "--bind", "0.0.0.0:5000", "app:app" ]
```

Fonte: Adaptado do console AWS (2025)

Esse arquivo define como a imagem deve ser construída, incluindo dependências, diretório de trabalho, porta exposta e comando de inicialização da aplicação.

CONSTRUÇÃO DA IMAGEM DOCKER

Na Figura 12 é exemplificado como foi feita a construção da Imagem Docker.

Figura 12 - Construção da Imagem Docker

```
Session ID: root-49ujjetcudac22367telhxxpi4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia# nano Dockerfile
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia# docker build -t conversao-distancia -f Dockerfile .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 79.36kB
Step 1/7 : FROM python:3.13.0
3.13.0: Pulling from library/python
fd894e782a2: Pull complete
5bd71677db44: Pull complete
551df7f94f9c: Pull complete
ce82e98d553d: Pull complete
07db14c4e2ab: Pull complete
7fd26ac3f7b8: Pull complete
ff09d2a676dd: Pull complete
Digest: sha256:e95be020750503923c5d4f51a56ab8f5b21e40cdce66fb7000e270df68d04f8e
Status: Downloaded newer image for python:3.13.0
```

Fonte: Adaptado do console AWS (2025)

VERIFICAÇÃO E EXECUÇÃO DA IMAGEM

Na Figura 13 pode ser visto um exemplo da aplicação de comandos para verificar e executar a imagem Docker.

Figura 13 - Verificar e Executar a Imagem Docker

```
Session ID: root-49ujjetcudac22367telhxxpi4 Shortcuts Instance ID: i-01ebb67c1a0d14125 Terminate

root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia# docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
conversao-distancia latest      f02c4c33c2dc  2 minutes ago 1.04GB
nginx                latest      2cd1d97f893f  11 days ago  192MB
python               3.13.0      058afcc56095  9 months ago  1.02GB
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia# docker run -d -p 5000:5000 conversao-distancia
0f24d3d056b904fd092b0a988e1cdd54c8629bf8c57b1d38909ba23da992081
root@ip-10-0-144-179:/var/snap/amazon-ssm-agent/12051/conversao-distancia#
```

Fonte: Adaptado do console AWS (2025)

Ao executar o contêiner, a aplicação ficou disponível localmente na porta 5000, permitindo o teste via navegador ou comandos *curl*.

PREPARAÇÃO PARA O DOCKER HUB

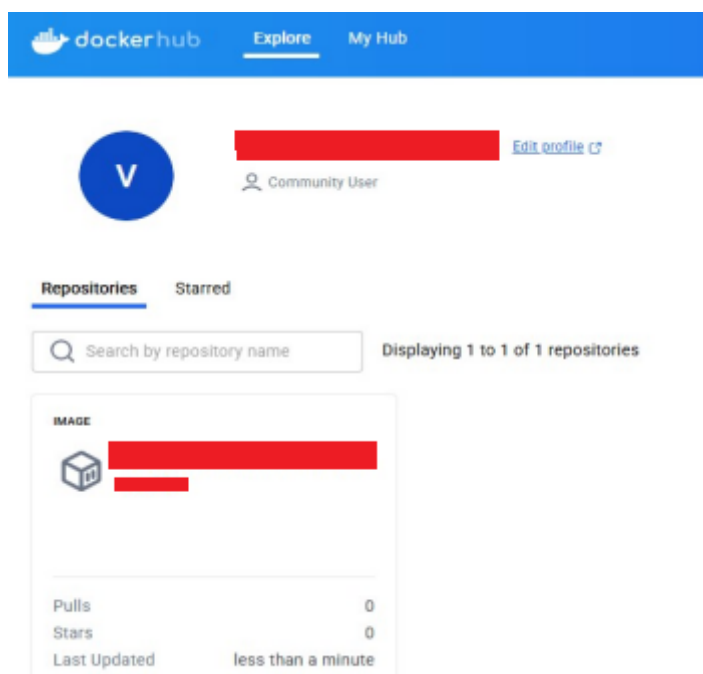
Foi criada uma nova versão da imagem com a tag **mariaciriaco/conversao-distancia:v1**, utilizando o comando: **docker build -t mariaciriaco/conversao-distancia:v1**.

Em seguida, os alunos realizaram o *login* em suas contas no Docker Hub com o comando: **docker login**.

Por fim, a imagem foi enviada para o Docker Hub por meio do comando: **docker push mariaciriaco/conversao-distancia:v1**.

Essa atividade permitiu aos participantes vivenciar o processo completo de criação, versionamento e publicação de uma imagem Docker em um repositório público, permitindo que outros usuários da comunidade UFERSA (e além) pudessem utilizar a imagem diretamente. A Figura 14 mostra a imagem do repositório de um dos alunos.

Figura 14 - Imagem no Repositório Docker Hub



Fonte: Adaptado do Docker Hub (2025)

Durante a atividade, os alunos compreenderam na prática:

- Como construir um ambiente de execução isolado para aplicações web;
- A importância de versionar e distribuir imagens padronizadas;
- Como reutilizar contêineres em ambientes de nuvem ou locais;
- Como um projeto simples pode ser transformado em um serviço acessível e utilizável via Docker Hub.

4.5 Módulo 3: Kubernetes

Nesta etapa do minicurso, os alunos foram introduzidos à prática com Kubernetes, com foco em um ambiente real utilizando o Amazon EKS (*Elastic Kubernetes Service*). O objetivo foi proporcionar uma experiência prática de orquestração de contêineres em nuvem, utilizando as ferramentas discutidas teoricamente na aula anterior.

4.5.1 Seção 1 – Introdução ao Kubernetes

Esta aula teve como objetivo apresentar os conceitos fundamentais da orquestração de contêineres utilizando Kubernetes. Inicialmente, foi realizado um comparativo entre Docker e Kubernetes, destacando seus papéis complementares no ciclo de vida das aplicações containerizadas.

- Docker é utilizado para empacotar e executar aplicações em contêineres de forma isolada.
- Kubernetes é responsável por orquestrar múltiplos contêineres, distribuindo-os em clusters de servidores, garantindo alta disponibilidade, escalabilidade automática e tolerância a falhas.

COMPONENTES E ARQUITETURA DO KUBERNETES

Foram explorados os principais componentes que compõem a arquitetura do Kubernetes:

- *Control Plane*: Responsável pela gerência central do cluster (*API Server*, *etcd*, *Scheduler*, *Controller Manager*).
- *Worker Nodes*: Executam as cargas de trabalho reais (Pods) e interagem com o plano de controle via *kubelet*, *kube-proxy* e *Container Runtime*.

Além disso, os alunos foram introduzidos às interfaces de interação com o *cluster*, como:

- `kubectl` (linha de comando);
- *Dashboard* (interface gráfica);
- API RESTful do Kubernetes.

OBJETOS FUNDAMENTAIS NO KUBERNETES

Durante a aula, foram explicados e praticados os seguintes conceitos:

- *Pods*: Unidades mínimas de execução (um ou mais contêineres).
- *ReplicaSet*: Garante a replicação de *Pods*.
- *Deployments*: Define a forma como aplicações são implementadas, atualizadas e escaladas.
- *Services*: Abstraem o acesso aos *Pods*, permitindo balanceamento de carga e descoberta de serviços.
- *Namespaces*: Permitem a segmentação de recursos dentro de um cluster.

Os alunos também foram apresentados ao uso do *YAML*, a linguagem declarativa de configuração utilizada no Kubernetes.

4.5.2 SEÇÃO 2 – Seção 2 – Implementando Kubernetes do Zero ao Deploy

Nesta aula, foi realizada uma demonstração prática do *deploy* de uma aplicação containerizada utilizando um cluster Kubernetes local, criado com a ferramenta K3d. Para garantir melhor desempenho, foi utilizada uma nova instância EC2 configurada de forma semelhante à anterior, porém com a família de instância *t3.medium*, substituindo a instância anterior da família *t2.micro*. O esperado era que a aplicação não travasse.

O projeto utilizado foi da aplicação de Conversão de Distância, que já havia sido desenvolvida e containerizada com Docker nas aulas anteriores.

CRIAÇÃO DO MANIFESTO KUBERNETES

Dando continuidade ao projeto anterior, foi implantado na aplicação um manifesto *YAML*. Para isso, foi criado um diretório chamado *k8s* e, dentro dele, um arquivo chamado *deployment.yaml* contendo as definições dos recursos a serem implantados (Figura 15 e Figura 16):

- Deployment: cria um *pod* com um contêiner baseado na imagem **mariaciriaci/conversao-distancia:v1**, definindo uma réplica e expondo a porta 5000.
- Service: expõe a aplicação na porta 30000 do cluster, permitindo acesso externo via IP do node.

Figura 15 - Comandos para Criar Diretório e Arquivo

```
Session ID: root-xh7j4drfqg64iceli2k2n82x28 Shortcuts Instance ID: i-04e6705688cc3c26f Terminate

root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051# ls
conversao-distancia
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051# cd conversao-distancia/
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia# mkdir k8s
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia# ls
README.md app.py k8s requirements.txt templates
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia# cd k8s/
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia/k8s# ls
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia/k8s# nano deployment.yaml
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia/k8s#
```

Fonte: Adaptado do console AWS (2025)

Figura 16 - Editor de Texto com o Manifesto YAML

```
Session ID: root-xh7j4drfqg64iceli2k2n82x28 Shortcuts Instance ID: i-04e6705688cc3c26f Terminate

GNU nano 7.2 deployment.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: conversao-distancia
spec:
  replicas: 1
  selector:
    matchLabels:
      app: conversao-distancia
  template:
    metadata:
      labels:
        app: conversao-distancia
    spec:
      containers:
        - name: conversao-distancia
          image: mariaciriaci/conversao-distancia:v1
          ports:
            - containerPort: 5000
---
apiVersion: v1
```

Fonte: Adaptado do console AWS (2025)

INSTALAÇÃO DO KUBERNETES LOCAL COM K3D

Para facilitar o desenvolvimento e testes, o Kubernetes foi instalado localmente utilizando a ferramenta K3d, que cria *clusters* Kubernetes leves e locais dentro de contêineres Docker. Com comandos simples, K3d permite configurar clusters rapidamente, sendo ideal para quem precisa experimentar Kubernetes de forma prática e sem a complexidade de uma instalação completa.

Comandos principais para instalação e verificação:

- Para instalar o *kubect* usando o gerenciador de pacotes *snap*: **snap install kubect --classic**
- Para mostrar a versão do *client kubect* que está instalada na instância: **kubect version --client**
- Para fazer o download e executa o script de instalação do *k3d*, que permite criar clusters Kubernetes com Docker: **curl -s https://raw.githubusercontent.com/k3d-io/k3d/main/install.sh | bash**
- Exibe a versão instalada: **k3d version**

CRIAÇÃO E GERENCIAMENTO DO CLUSTER KUBERNETES

Com K3d, foi criado um cluster chamado *mycluster* com 1 servidor e 2 agentes, mapeando a porta *8080* do host para a porta *30000* do cluster para acesso externo (Figura 17).

Figura 17 - Comando para Criar Cluster



The screenshot shows an AWS console terminal window. At the top, it displays 'Session ID: root-xh7j4drfqg64iceli2k2n82x28' and 'Instance ID: i-04e6705688cc3e26f'. There is a 'Shortcuts' button and a 'Terminate' button. The terminal output shows the command 'k3d cluster create mycluster --servers 3 --agents 3' being executed. The output includes several informational messages: 'INFO[0000] Prep: Network', 'INFO[0000] Created network 'k3d-mycluster'', 'INFO[0000] Created image volume k3d-mycluster-images', 'INFO[0000] Starting new tools node...', 'INFO[0000] Creating initializing server node', and 'INFO[0000] Creating node 'k3d-mycluster-server-0''.

```
root@ip-10-0-144-197:/var/snap/amazon-ssm-agent/12051/conversao-distancia/k8s# k3d cluster create mycluster --servers 3 --agents 3
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-mycluster'
INFO[0000] Created image volume k3d-mycluster-images
INFO[0000] Starting new tools node...
INFO[0000] Creating initializing server node
INFO[0000] Creating node 'k3d-mycluster-server-0'
```

Fonte: Adaptado do console AWS (2025)

DEPLOY DA APLICAÇÃO NO CLUSTER

Foi aplicado o manifesto Kubernetes para criar os recursos no *cluster*, através do comando: **kubectl create -f k8s/deployment.yaml**

Verificações realizadas:

- Listagem dos pods em execução: **kubectl get pods**
- Detalhes de um pod específico: **kubectl describe pod <nome-do-pod>**
- Monitoramento contínuo dos pods: **kubectl apply -f k8s/deployment.yaml && watch 'kubectl get pods'**
- Visualização dos recursos do cluster: **kubectl get all**

TESTE DE ACESSO À APLICAÇÃO

Para testar o acesso externo, a porta 30000 do cluster foi mapeada para a porta 8080 do host, permitindo acessar a aplicação via navegador no endereço, através do comando: **kubectl get svc fakeshop**.

4.5.3 Seção 3 – Kubernetes na AWS

Nesta atividade, os alunos aprenderam a implementar um novo projeto do zero usando uma imagem já configurada e disponível no Docker Hub, seguindo o aprendizado das aulas anteriores. O objetivo foi explorar como configurar permissões via CLI no terminal e criar um cluster EKS utilizando linha de comando.

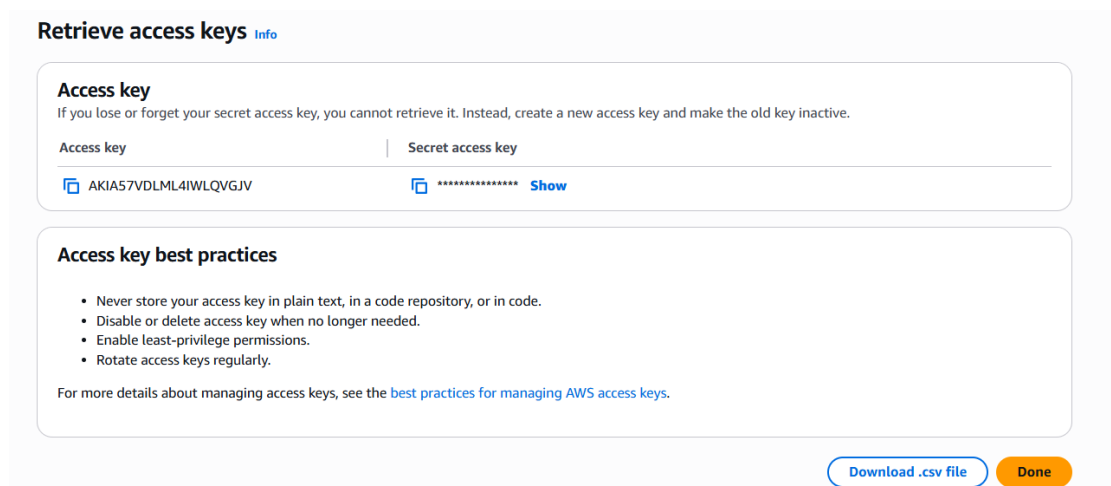
A ferramenta principal utilizada foi *eksctl*, uma ferramenta de linha de comando oficial para criar e gerenciar clusters do Amazon EKS (*Elastic Kubernetes Service*) com simplicidade, e o AWS CLI para configurar acesso a serviços AWS.

CONFIGURAÇÃO DE ACESSO À AWS VIA CLI

Inicialmente, os alunos foram orientados a gerar uma *Access Key* no serviço IAM para criação de credenciais de acesso programático (*Access Key ID* e *Secret Access Key*). Essas credenciais foram então utilizadas para configurar a AWS CLI com o comando: **aws configure**.

Essa configuração permitiu que os participantes pudessem interagir com os serviços da AWS, especialmente, o Amazon EKS, diretamente pelo terminal da instância EC2 (Figura 18).

Figura 18 - Credenciamento com Access keys

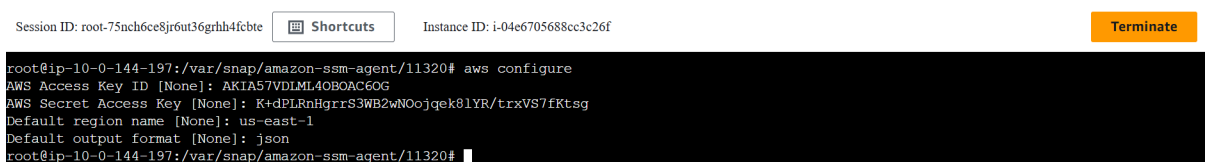


Fonte: Adaptado do console AWS (2025)

PREPARAÇÃO DA INSTÂNCIA EC2

Na etapa seguinte, os alunos acessaram o terminal da instância EC2 por meio do Session Manager (Figura 19), em que realizaram a instalação das ferramentas *eksctl*, *curl*, *tar* e da *AWS CLI*. Em seguida, configuraram a *Access Key* na *CLI* com o objetivo de estabelecer a conexão com os serviços da AWS.

Figura 19 - Configuração no Terminal do AWS *configure*



Fonte: Adaptado do console AWS (2025)

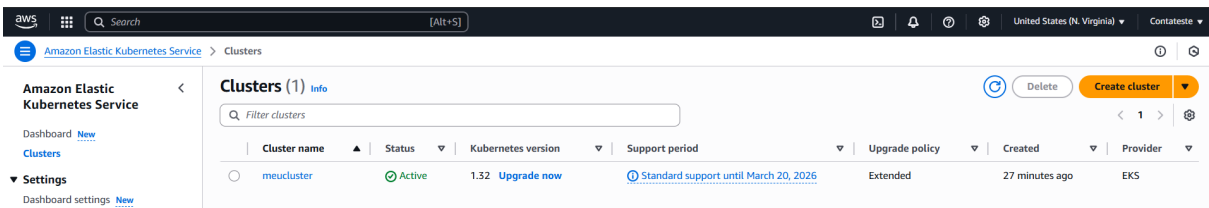
CRIAÇÃO DO CLUSTER KUBERNETES NO AMAZON EKS

Com os ambientes configurados e com o devido permissionamento, foi demonstrado como criar um cluster Kubernetes, utilizando o *eksctl* no terminal, por meio do seguinte comando: `eksctl create cluster --name meucluster --nodes-min 1 --nodes-max 2 --region us-east-1`.

Esse processo levou alguns minutos. Enquanto isso, foi explicado aos alunos o que cada parte do comando representa, como por exemplo: *--nodes-min* e *--nodes-max*, que definem, respectivamente, o número mínimo e máximo de nós EC2 (servidores) configurados na região N. Virginia.

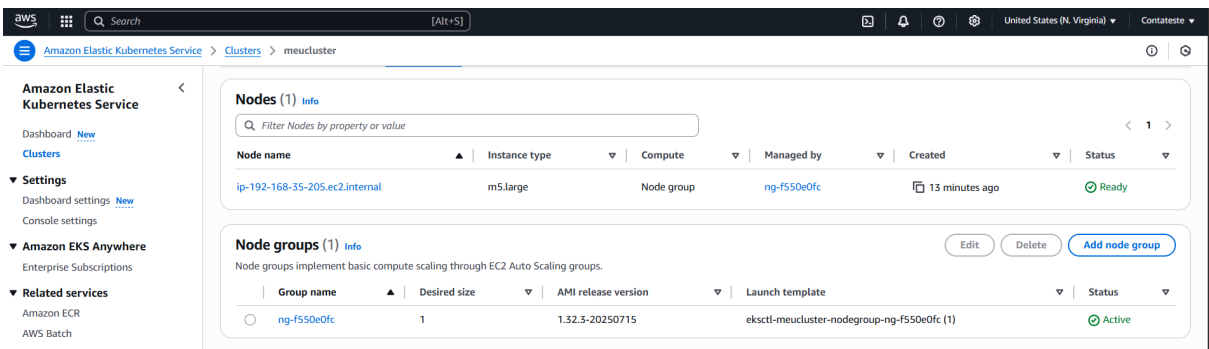
Ao final desse processo, foi apresentado como listar os *clusters* criados na região configurada, verificar a quantidade de nós (*nodes*) do cluster Kubernetes e visualizar o cluster configurado automaticamente no console do Amazon EKS (Figura 20 e Figura 21).

Figura 20 - Cluster EKS no Console AWS



Fonte: Adaptado do console AWS (2025)

Figura 21 - Nodes no Console AWS



Fonte: Adaptado do console AWS (2025)

CONFIGURANDO O DEPLOYMENT NO KUBERNETES

Nessa etapa, foi utilizado um projeto em Python com uma aplicação web disponível em um repositório no GitHub chamado fake-shop, mantido pelo professor Fabrício Veronez. A partir desse projeto, foi criada uma imagem Docker, que posteriormente foi registrada no Docker Hub. Os alunos tiveram acesso a essa imagem, criada seguindo o mesmo processo ensinado nas aulas práticas de Docker.

Com a imagem no Docker Hub, foi realizado o *push* para a instância EC2 (Figura 22). Em seguida, criou-se um diretório chamado k8s, em que foi adicionado o arquivo *deployment.yaml*, contendo o manifesto responsável por criar o *deployment* da aplicação web. Após salvar as configurações, o manifesto foi aplicado com o comando **kubectl apply -f k8s/deployment.yaml**.

Figura 22 - Comando do *Deployment*

```
root@ip-192-168-23-217:~/fake-shop# kubectl apply -f ~/fake-shop/k8s/deployment.yaml
deployment.apps/postgre created
service/postgre created
deployment.apps/fakeshop created
service/akeshop created
root@ip-192-168-23-217:~/fake-shop#
```

Fonte: Adaptado do console AWS (2025)

Para permitir o acesso à aplicação, foi necessário abrir regras de entrada (*inbound rules*) no *Security Group* associado à instância EC2. Com o manifesto aplicado, a aplicação foi configurada dentro do cluster EKS.

Por meio do terminal, utilizando o comando **kubectl get all**, os alunos puderam observar os recursos do Kubernetes relacionados à aplicação (Figura 23), como *pods*, *services*, *deployments* e *replicasets*, conceitos abordados nas aulas teóricas.

Figura 23 - Comando *kubectl get all*

```
root@ip-192-168-23-217:~/fake-shop# kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/fakeshop-85df69cf6c-mt2xp       1/1      Running   0           103s
pod/postgre-b66685bbc-r52jx         1/1      Running   0           103s

NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
service/akeshop                     LoadBalancer       10.100.61.156    ac67e0b8b59b24ae386c0ccdc749396-1919710508.us-east-1.elb.amazonaws.com  80:32494/TCP     103s
service/kubernetes                   ClusterIP            10.100.0.1       <none>            443/TCP          102m
service/postgre                      ClusterIP            10.100.206.60    <none>            5432/TCP         103s

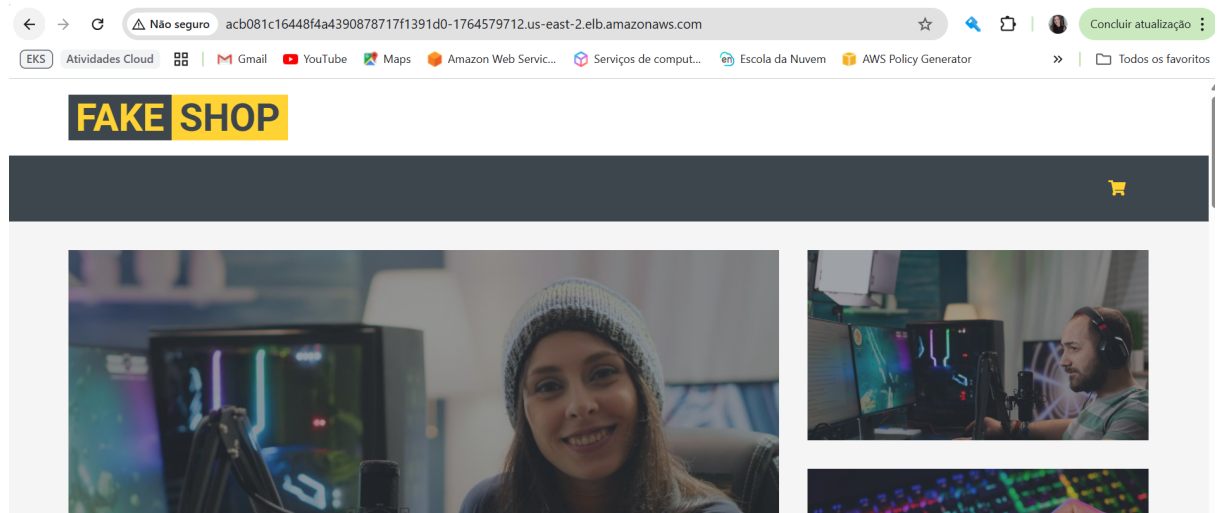
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/fakeshop             1/1      1              1            103s
deployment.apps/postgre              1/1      1              1            103s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/fakeshop-85df69cf6c  1          1          1        103s
replicaset.apps/postgre-b66685bbc     1          1          1        103s
root@ip-192-168-23-217:~/fake-shop#
```

Fonte: Adaptado do console AWS (2025)

Por fim, a aplicação foi testada com sucesso por meio do endereço DNS fornecido pelo *Load Balancer* do EKS (Figura 24).

Figura 24 - DNS do Load Balancer no Navegador



Fonte: Adaptado do navegador Google (2025)

Ao final de cada aula, os materiais utilizados eram disponibilizados aos alunos em formato PDF e em vídeo, por meio do Google Sala de Aula e também no grupo do WhatsApp. O objetivo era permitir que os alunos pudessem revisar as aulas gravadas e estudar os materiais teóricos e tutoriais para praticarem no console da AWS. Durante a realização dessas atividades, surgiram dúvidas, e alguns alunos solicitaram atendimentos individuais por meio de chamadas (*calls*) para obter auxílio na execução das tarefas propostas.

5 RESULTADOS E DISCUSSÕES

Neste capítulo, são apresentados os resultados e as discussões obtidos a partir da análise de formulários aplicados com o objetivo de compreender as expectativas dos alunos durante o desenvolvimento e após a finalização do curso. A Seção 5.1 apresenta o formulário diagnóstico, aplicado no início da experiência. A Seção 5.2 discute os resultados do formulário final, utilizado para avaliar se as expectativas dos alunos foram atendidas e para entender o que esperam em relação aos próximos projetos.

5.1 Formulário diagnóstico

Antes do início da primeira aula síncrona do minicurso, foi aplicado um formulário diagnóstico com o objetivo de avaliar o nível de conhecimento prévio dos alunos sobre os temas abordados, como Docker, Kubernetes e AWS. O formulário foi elaborado por meio da ferramenta Google Forms, com prazo estipulado para preenchimento até um dia após o início das aulas síncronas.

O questionário foi composto por perguntas objetivas, contemplando os seguintes aspectos:

- Nome completo;
- Já teve alguma experiência com computação em nuvem?
- Qual provedor de nuvem você já utilizou?
- Como você classifica seu conhecimento sobre AWS?
- Como você classifica seu conhecimento sobre Docker?
- Como você classifica seu conhecimento sobre Kubernetes?

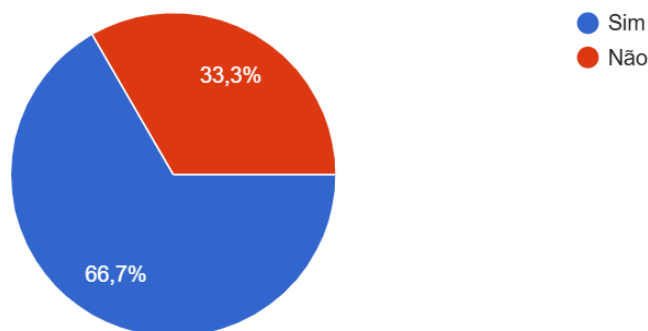
5.1.1 Respostas ao Questionário

Analisando as respostas dos usuários, os resultados apontam que a maioria dos discentes (66,7%) já possuíam alguma experiência com Computação em Nuvem (*Cloud Computing*), conforme pode ser visto na Figura 25. Dos 12 discentes que responderam ao formulário, 8 afirmaram possuir experiência com *Cloud Computing*, enquanto 4 indicaram

não ter tido contato anterior com essa tecnologia. Esses resultados podem se dar por essa tecnologia ser abordada dentro de componentes curriculares, tais como Sistemas Distribuídos, Dependabilidade e Segurança e Programação Web, de cursos como BITI, ES e EC.

Figura 25 - Experiência com Cloud Computing

12 respostas

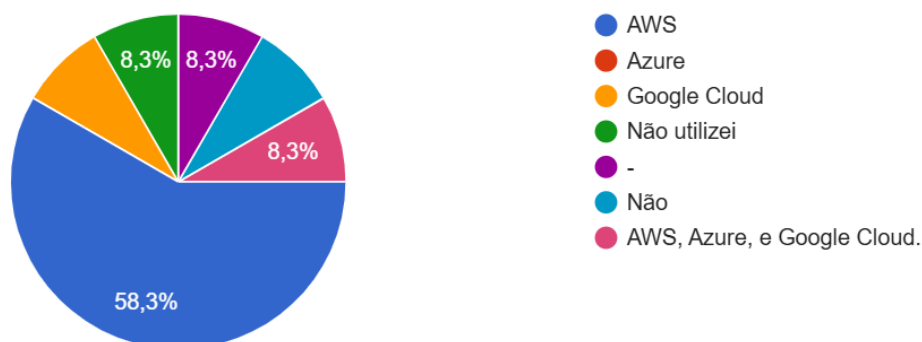


Fonte: Adaptado do Google Formulário (2025)

Por sua vez, referente a experiência com provedores em nuvem, a resposta foi diversificada (Figura 26). Dos 12 alunos, 7 possuíam experiência com a AWS, 1 com o Google Cloud, e 1 com os três provedores: AWS, Azure e Google Cloud. Já 3 alunos informaram que não possuíam experiência com nenhum provedor de nuvem.

Figura 26 - Experiência com Provedores de Nuvem

12 respostas

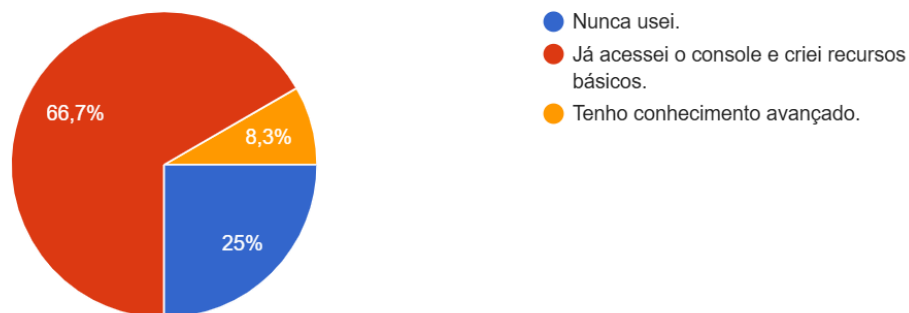


Fonte: Adaptado do Google Formulário (2025)

Por sua vez, no que se refere ao nível de conhecimento em AWS, dos 12 alunos, 8 já acessaram o console e criaram recursos básicos, 3 nunca utilizaram a plataforma e 1 possuía conhecimento avançado em AWS (Figura 27).

Figura 27 - Nível de Conhecimento em AWS

12 respostas



Fonte: Adaptado do Google Formulário (2025)

Referente ao nível de conhecimento com Docker, os resultados do gráfico da Figura 28 apontam que, dos 12 alunos, 5 conheciam os conceitos, mas nunca utilizaram na prática; 4 já criaram contêineres simples; 2 nunca utilizaram; e 1 já escreveu *Dockerfiles*.

Figura 28 - Nível de Conhecimento em Docker

12 respostas



Fonte: Adaptado do Google Formulário (2025)

Já no tocante ao nível de conhecimento com Kubernetes, conforme Figura 19, dos 12 participantes, 8 já haviam ouvido falar sobre Kubernetes, mas nunca utilizaram; e 4 nunca haviam ouvido falar da tecnologia.

Figura 29 - Nível de Conhecimento em Kubernetes

12 respostas



Fonte: Adaptado do Google Formulário (2025)

5.2 Formulário final

Ao final do minicurso, foi aplicado um formulário com o objetivo de avaliar a experiência dos alunos e identificar pontos de melhoria para os próximos cursos. O instrumento de avaliação foi elaborado por meio da ferramenta Google Forms, contendo as seguintes perguntas:

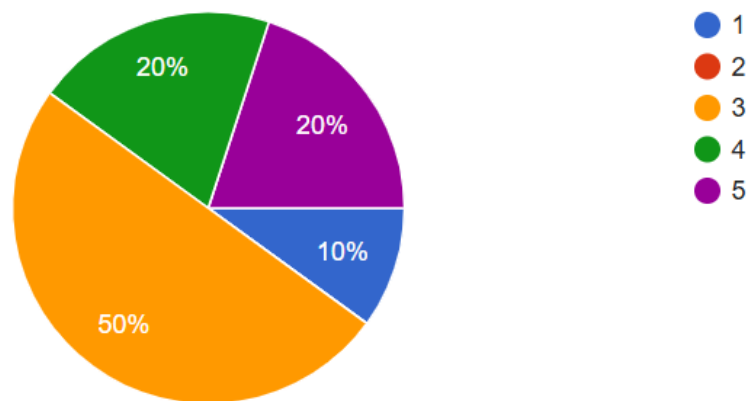
- Me sinto confiante para aplicar os conteúdos aprendidos na prática;
- Consegui realizar atividades práticas durante o curso como;
- Como você avalia a didática do curso?
- O conteúdo atendeu às suas expectativas?
- Se não atendeu às suas expectativas, justifique;
- Como foi o ritmo das aulas ?
- O que você mais gostou no curso?
- Você indicaria esse curso para outras pessoas?
- O que você pretende fazer com o que aprendeu no curso?
- Quais tópicos você gostaria de ver em um curso mais avançado?

5.2.1 Resposta ao Questionário

No que se refere a confiança dos participantes em aplicar os conteúdos aprendidos na prática (Figura 30), dos 10 alunos que responderam ao questionário, avaliando seu nível de confiança para aplicar os conteúdos aprendidos na prática em uma escala de 1 a 5 (sendo 1 "muito baixo" e 5 "muito alto"), 5 atribuíram nível de confiança 3; 2 alunos atribuíram nível 4; outros 2, nível 5; e 1 aluno avaliou com nível 1.

Figura 30 - Confiança para aplicar os conteúdos aprendidos na prática

10 respostas

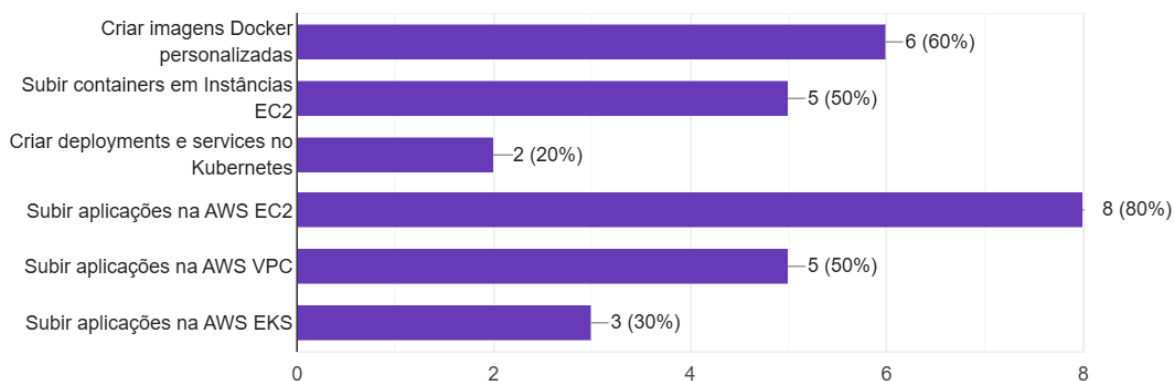


Fonte: Adaptado do Google Formulário (2025)

Os resultados do gráfico apresentado na Figura 31 apontam que as atividades mais realizadas pelos participantes foram: subir aplicações na AWS EC2, com 80% das indicações; criar imagens Docker, com 60% das respostas; e subir containers em instâncias EC2 e subir aplicações AWS VPC, ambas com 50% das indicações dos respondentes.

Figura 31 - Realização de atividades práticas durante o curso

10 respostas

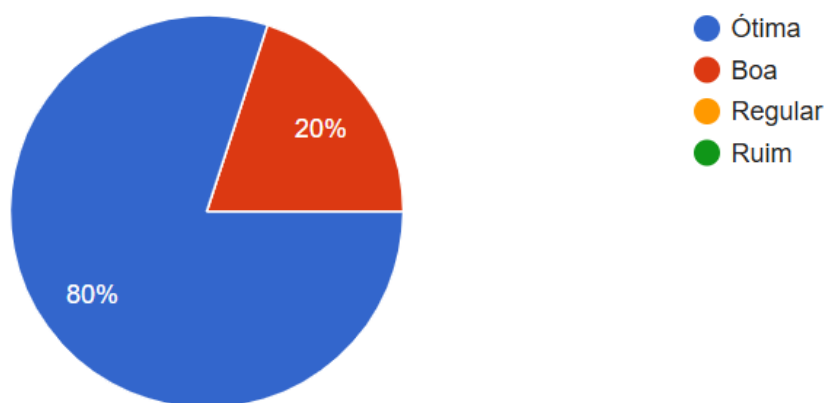


Fonte: Adaptado do Google Formulário (2025)

Referente a avaliação da didática do curso (Figura 32), dos 10 participantes, 8 avaliaram que a didática do curso foi ótima, enquanto 2 consideraram boa.

Figura 32 - Avaliação da Didática do Curso

10 respostas

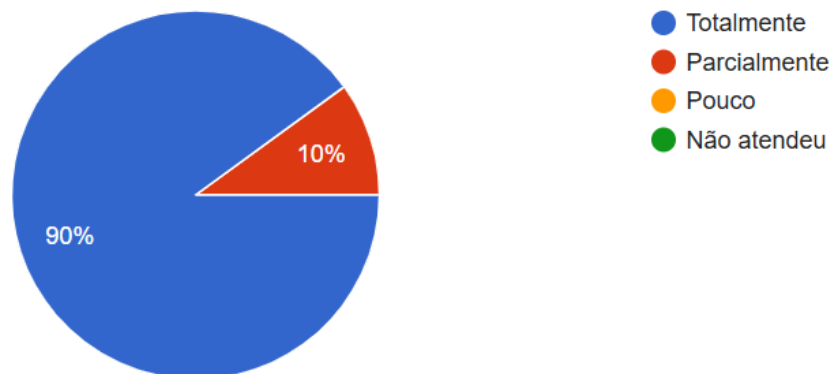


Fonte: Adaptado do Google Formulário (2025)

No que concerne ao atendimento às expectativas do conteúdo, os resultados apresentados na Figura 33 apontam que, dos 10 alunos que responderam à pergunta sobre o atendimento às expectativas em relação ao conteúdo, 9 afirmaram que o curso atendeu totalmente às suas expectativas, enquanto 1 respondeu que foi parcialmente atendido.

Figura 33 - O conteúdo atendeu às suas expectativas

10 respostas

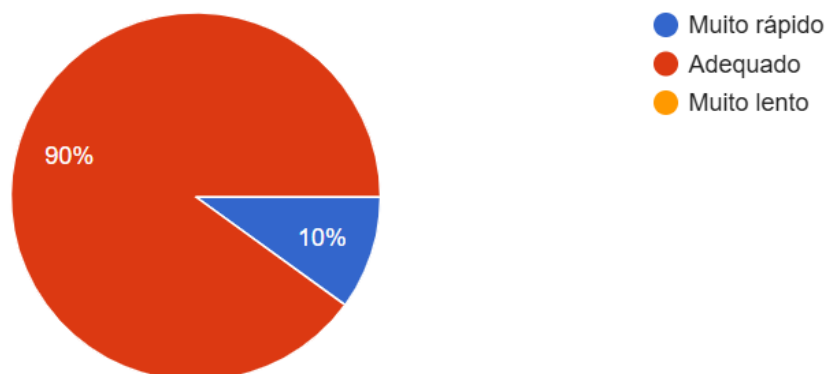


Fonte: Adaptado do Google Formulário (2025)

No que concerne a avaliação dos ritmo das aulas pelos alunos, no gráfico da Figura 34, é possível ver que, dos 10 alunos que responderam à pergunta sobre o ritmo das aulas, 9 consideraram o ritmo adequado e 1 aluno avaliou como muito rápido.

Figura 34 - Avaliação do Ritmo das Aulas

10 respostas

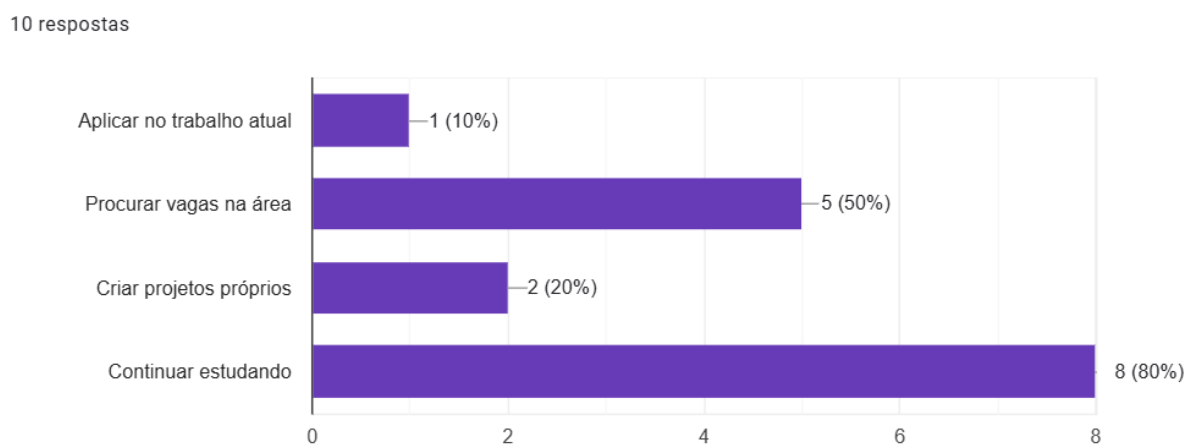


Fonte: Adaptado do Google Formulário (2025)

Referente aos aspectos, mais apreciados nos cursos (pergunta aberta), foram citados pelos participantes o Docker, a didática, a chance de aprender com atividades prática e a oportunidade de conhecer coisas novas.

Em relação à pergunta "O que você mais gostou no curso?", os 10 alunos forneceram respostas diversas, mas com ênfase em aspectos relacionados à didática e à aplicação prática dos conteúdos, conforme pode ser observado no gráfico da Figura 35.

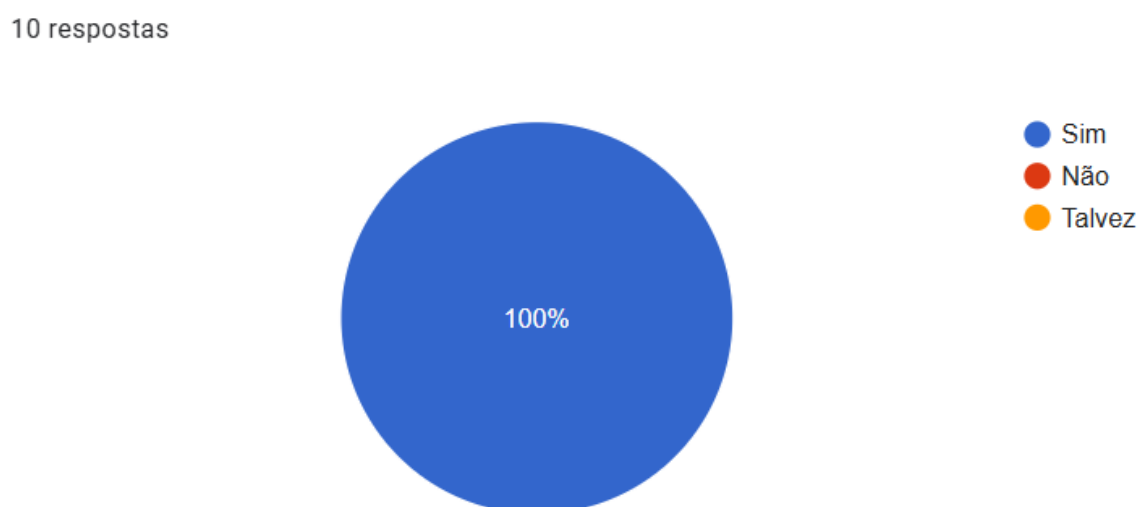
Figura 35 - Intenção de uso do conteúdo aprendido



Fonte: Adaptado do Google Formulário (2025)

Conforme pode ser visto na Figura 36, os respondentes foram unânimes referente a indicação do curso para outras pessoas.

Figura 36: Indicação do curso para outras pessoas



Fonte: Adaptado do Google Formulário (2025)

Em uma pergunta aberta, foi questionado sobre os tópicos que os participantes desejariam ter em cursos mais avançados. Entre as citações, encontram-se pontos, tais como saber Kubernetes com exemplos práticos, como conseguir estágio na área de Computação em Nuvem, Autenticação com Cogniti e Docker.

Os questionários aplicados foram importantes para avaliar o nível de conhecimento dos alunos, bem como para compreender suas expectativas ao longo do curso e após a sua conclusão. Os resultados do questionário final demonstraram que o minicurso foi bem avaliado pelos participantes, atendendo às suas expectativas. Vale destacar que os mesmos alunos que responderam ao primeiro questionário, aplicado no início da primeira aula, também participaram da avaliação final, o que garante consistência na análise dos dados.

6 CONSIDERAÇÕES FINAIS

Este trabalho abordou o desenvolvimento de um minicurso voltado para o ensino de conceitos teóricos e práticos sobre tecnologias como Docker e Kubernetes, utilizando a AWS como infraestrutura de nuvem, destinado aos alunos da Universidade Federal Rural do Semi-Árido (UFERSA). Para isso, foi realizada uma RSL, para levantar estratégias utilizadas em relatos de experiência sobre a aplicação de cursos de ensino e ou extensão, no tocante às metodologias utilizadas, as avaliações aplicadas e os resultados esperados com a realização do curso. Essas tecnologias são fundamentais no cenário atual do mercado de trabalho, e o principal objetivo do projeto foi apresentar os conceitos essenciais que os alunos podem aplicar em projetos acadêmicos, enriquecendo seus currículos.

Os objetivos propostos foram alcançados, considerando que foi possível:

- Criar um minicurso utilizando metodologias ativas, que priorizam a aprendizagem por meio da prática, tornando o aluno protagonista do seu próprio processo de ensino-aprendizagem, ao mesmo tempo em que dominam tecnologias amplamente utilizadas no mercado;
- Aplicar a metodologia da sala de aula invertida, disponibilizando videoaulas previamente gravadas, permitindo que os alunos acessassem o conteúdo quantas vezes fossem necessárias e levassem suas dúvidas para os encontros assíncronos.
- A participação dos alunos foi essencial para o desenvolvimento do minicurso. Além do ensino das tecnologias propostas, os próprios alunos contribuíram com discussões relevantes, como a inclusão de tópicos sobre redes, o que enriqueceu ainda mais as aulas.

6.1 Trabalhos futuros

A análise dos questionários respondidos pelos alunos, bem como a vivência ao longo do desenvolvimento do curso, permitiu identificar pontos de melhoria para futuras edições do minicurso. Pretende-se dar continuidade a este trabalho, com a aplicação de novas turmas no início de cada período letivo. A proposta é desenvolver uma abordagem mais aprofundada dos conteúdos, com foco nas demandas do mercado de trabalho, aprimorando continuamente o formato e a carga horária do curso com base no retorno dos participantes.

REFERÊNCIAS

ALMEIDA, Carlos Eugenio de Castro; MIGUEL, Joelson Rodrigues. Metodologias ativas e recursos tecnológicos nos processos de ensino e aprendizagem. Id on Line: **Revista Multidisciplinar e de Psicologia**, Salvador, v. 14, n. 50, p. 422–438, 2020.

AMAZON WEB SERVICES. **O que é a computação em nuvem?**. AWS. 2025a. Disponível em: <<https://aws.amazon.com/pt/what-is-cloud-computing/>>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **Gerenciar instâncias privadas do EC2 sem acesso à Internet (gerar endpoints da Amazon VPC para Systems Manager)**. AWS. 2025b. Disponível em: <<https://aws.amazon.com/pt/premiumsupport/knowledge-center/ec2-systems-manager-vpc-endpoints/>>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **Visão geral do agente do AWS Systems Manager**. AWS. 2025c. Disponível em: <https://docs.aws.amazon.com/pt_br/systems-manager/latest/userguide/ssm-agent.html>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **Introdução ao Amazon EC2**. AWS. 2025d. Disponível em: <https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/EC2_GetStarted.html>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **Configurar permissões de instância para o Systems Manager**. AWS. 2025e. Disponível em: <https://docs.aws.amazon.com/pt_br/systems-manager/latest/userguide/setup-instance-permissions.html>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **O que é o AWS Systems Manager**. AWS. 2025f. Disponível em: <https://docs.aws.amazon.com/pt_br/systems-manager/latest/userguide/what-is-systems-manager.html>. Acesso em: 30 jul. 2025.

AMAZON WEB SERVICES. **Qual é a diferença entre o Docker e uma VM?**. AWS. 2025g. Disponível em: <<https://aws.amazon.com/pt/compare/the-difference-between-docker-vm/>>. Acesso em: 30 jul. 2025.

AZEMI, Asad; BODEK, Mathew; CHINN, Gary. Teaching an introductory programming course using hybrid e-learning approach. *In: IEEE Frontiers in Education Conference (FIE)*. IEEE, 2013. p. 1911-1913.

COLVARA, Jonas dos Santos; SANTO, Eniel do Espírito. Metodologias ativas no ensino superior: o hibridismo da sala de aula invertida. **Revista Brasileira de Aprendizagem Aberta e a Distância**, [S.l.], v. 18, n. 1, 2019.

DOCKER INC. **Docker Hub**. Docker. 2025a. Disponível em: <<https://hub.docker.com/>>. Acesso em: 30 jul. 2025.

DOCKER INC. **Docker Hub API Reference**. Docker. 2025b. Disponível em: <<https://docs.docker.com/reference/api/hub/latest/>>. Acesso em: 30 jul. 2025.

EBAC ONLINE. **O que é GitHub?**. EbacOnline. 2025. Disponível em: <<https://ebaconline.com.br/blog/o-que-e-github>>. Acesso em: 30 jul. 2025.

FEITOZA, Alisson Castro. **Iniciativas para o ensino de computação/informática em outras áreas: uma revisão sistemática da literatura**. Trabalho de Conclusão de Curso (Graduação em Bacharelado Interdisciplinar em Tecnologia da Informação), Universidade Federal Rural do Semi-Árido, Campus Pau dos Ferros. 2025.

FERNANDES, Daniele Cristina Alves; FERNANDES, Helder Matheus Alves; BARBOSA, Elane da Silva; CHAVES, Márcia Jaíne Campelo; NÓBREGA-THERRIÉN, Sílvia Maria. Contribuições da monitoria acadêmica na formação do aluno-monitor do curso de Enfermagem: relato de experiência. **Revista da Universidade Federal de Alagoas**, v. 12, n. 27, 2020.

FERREIRA, Ana Carolina; SANTOS, Juliana Oliveira dos; SILVA, Raul; OLIVEIRA, Allan Thales Ramos; ZABOT, Diego; ABDALLA, Débora; MATOS, Ecivaldo. Hello World: relato de experiência de um curso de iniciação à programação. **Workshop sobre Computação na Educação**, v. 1, 2016.

FREITAS D'ESTILLAC LEAL, Ana Claudia de Macena; AMPARO, Flávia Vieira da Silva do. Poesia e letramento literário: relato de curso de extensão online para professores dos anos iniciais. **E-mosaicos**, Rio de Janeiro, v. 12, n. 30, 2023.

GARZKE, Jessica; STEINWAND, Blaire J. CREATE'ing improvements in first-year students' science efficacy via an online introductory course experience. **Journal of Microbiology and Biology Education**, v. 25, n. 1, p. e00079-23, 2024.

GEMELLI, Cátia Eli; SANTOS, Genéia Lucas dos; NUNES, Adriana Paz. Relato de experiência: casos de ensino como fomento de ensino-aprendizagem – relato de experiência no curso de Administração. **Revista Professare**, v. 7, n. 1, 2018.

GROSSI, Márcia Gorett Ribeiro. O lúdico na aprendizagem de alunos de cursos técnico, graduação e mestrado: relatos de experiências. **Revista Ibero-Americana de Estudos em Educação**, v. 12, n. 3, 2017.

HERNANDEZ NETO, Alberto. Avaliação de metodologia de ensino centrada no aluno. 2000, Anais. Ouro Preto: **ABENGE**, 2000.

INTERNATIONAL BUSINESS MACHINES – IBM. **O que é Docker?**. IBM. 2025. Disponível em: <<https://www.ibm.com/br-pt/think/topics/docker>>. Acesso em: 30 jul. 2025.

KITCHENHAM, B. Procedures for performing systematic reviews. Keele, UK, **Keele University**, v. 33, n. 2004, p. 1-26, 2004.

MACHADO, Adelcio; TWARDOWSKI, Rita Márcia; ALVES, Audete; CAETANO, Santos; LEFFER, Danielle Martins; ESCHER, Alisson André. Introdução a métodos de ensino. **International Journal of Development Research**, [S.l.], v. 11, n. 06, p. 47981–47982, jun. 2021.

MOREIRA, Sandra Maria do Nascimento; VIEIRA, Camila de Oliveira. Projeto “aprender vendo e aprender fazendo”: um relato de experiência do curso de Educação Física – Uniube. **Revista de Pesquisa em Educação**, v. 20, n. 44, 2020.

OLIVEIRA-BARRETO, Aline Cabral de et al. Métodos de avaliação discente em um curso de graduação baseado em metodologias ativas. **Revista Interfaces Científicas – Saúde e Ambiente, Lagarto: Universidade Federal de Sergipe**, 2017.

SILVEIRA, Sidnei Renato; VIT, Antônio Rodrigo Delepiane de; FORTES, Patrícia Rodrigues; FRANCISCATTO, Roberto. Sala de aula invertida: um relato de experiência em um curso de bacharelado em Sistemas de Informação. **Educa – Revista Multidisciplinar em Educação**, v. 8, p. 1–21, jan./dez. 2021.

VASCONCELOS, Andreza Cavalcanti; SOUZA, Gabrielly Laís de Andrade; BRAINER, Sâmara Aline Brito; SOARES, Raianne Monteiro; BARBOSA, Luciana Dilane dos Santos; CAMPOS, Paulo Isaac de Souza. As estratégias de ensino por meio das metodologias ativas / Teaching strategies through active methodologies. **Brazilian Journal of Development**, [S.l.], v. 5, n. 5, p. 5093–5105, 2019. ISSN 2525-8761.

APÊNDICE - METODOLOGIAS DOS ARTIGOS INSPIRADAS PARA A REALIZAÇÃO DA INTERVENÇÃO SOBRE O USO DE CONTÊINERES E AWS

A1. POESIA E LETRAMENTO LITERÁRIO: RELATO DE CURSO DE EXTENSÃO ONLINE PARA PROFESSORES DOS ANOS INICIAIS (FREITAS D'ESTILLAC LEAL, 2025)

Sequência Básica de Cosson (adaptada para a área técnica):

- **Motivação:** Crie um ambiente inicial instigante, com perguntas, desafios do mundo real, vídeos ou demonstrações rápidas sobre o impacto de containers na nuvem.
- **Introdução:** Apresenta conceitos essenciais como "O que é Docker?", "Porque usar Kubernetes no mercado de trabalho?", "Por que usar na AWS?".
- **Leitura/Exploração prática:** Essa é a etapa central. Os alunos manipulam containers, usam EKS, ECS ou Fargate e praticam com comandos Docker e YAML do Kubernetes.
- **Interpretação/Reflexão:** Momento de avaliação e troca. O que funcionou? O que foi mais difícil de entender e executar? Como aplicar isso em cenários reais?

Aprendizagem sensível e significativa:

- Traga casos reais do mercado de trabalho, problemas práticos e incentive os alunos a pensarem em soluções reais com containers, como "Como escalar uma aplicação web com Kubernetes?".
- Usar ambientes visuais, simulações ou painéis como o AWS Console, Cloud9 ou VSCode Web para tornar a experiência mais imersiva.

Roda de memória/experiência prévia (como no início do curso do artigo):

- Qual seu nível de familiaridade com Docker/Kubernetes?
- Você já implantou alguma aplicação na nuvem AWS?
- Quais desafios teve com infraestrutura e escalabilidade?

Avaliação com base em análise qualitativa e participação prática:

- Em vez de provas, avalie pela prática: execução de um pipeline simples com container, configuração de cluster no EKS, ou leitura de logs no CloudWatch.
- Promova reflexões técnicas: O que faria diferente se esse fosse um ambiente de produção?, ou Como otimizar esse deployment para reduzir custos?

Questionamentos que você pode fazer aos participantes:

- O que você entende por containerização?
- Já usou algum serviço AWS para gerenciar aplicações?

Durante a prática (para promover reflexão crítica):

- Qual a vantagem de usar Docker em vez de instalar tudo na máquina?
- Quais os desafios de orquestrar containers com Kubernetes?
- Por que usar Kubernetes no EKS e não localmente?
- Como a infraestrutura como código se encaixa nesse processo?

Ao final (avaliação/reflexão):

- Você se sente mais preparado para usar Docker/Kubernetes no trabalho?
- Quais conhecimentos ainda sente que precisa reforçar?
- Como essa abordagem em nuvem muda sua visão sobre DevOps?

A2. HELLO WORLD: RELATO DE EXPERIÊNCIA DE UM CURSO DE INICIAÇÃO À PROGRAMAÇÃO (FERREIRA *et al.*, 2025)

Metodologias do artigo:

1. Abordagem teórico-prática

- Combinar teoria com aplicação imediata em laboratório ou ambiente de nuvem (ex: AWS Cloud9, EC2, EKS, etc).
- Fazer com que os alunos testem comandos e criem containers e clusters enquanto aprendem.

2. Aprendizagem por projetos (project-based learning)

- Projeto final (ex: uma aplicação containerizada publicada em um cluster Kubernetes no EKS). Para estimular a autonomia, resolução de problemas e senso de realização.

3. Metodologia freireana (contextualização e participação ativa)

- Considerar o contexto dos participantes: têm familiaridade com a AWS? Já viram Docker?
- Incentivar perguntas e a troca de experiências usando analogias do cotidiano (ex: "containers como caixas").

4. Dinâmicas e práticas lúdicas

- Adaptar atividades “desplugadas” para explicar conceitos abstratos (ex: usar blocos físicos para simular containers e pods).
- Usar gamificação leve, como desafios práticos com recompensas simbólicas.

5. Acompanhamento individualizado e avaliação processual

- Estar atento ao ritmo de cada aluno.
- Avaliar por meio da prática e das interações, não só de testes objetivos.

6. Ambientes colaborativos e recursos digitais

- Usar ferramentas como Moodle, GitHub, Notion, Discord, WhatsApp ou Slack para comunicação e apoio entre encontros.
- Estimule os alunos a ajudarem uns aos outros, como no CIProg.

Questionamentos que você deve fazer ao planejar o mini curso:

1. Quem são meus alunos e qual o nível de conhecimento prévio deles?
 - a. Já conhecem AWS, Docker e Kubernetes?
 - b. Têm acesso a computador com recursos mínimos? Já possuem conta AWS?
2. Quais são os objetivos principais do curso?
 - a. Ensinar a criar imagens Docker?
 - b. Implantar aplicações no Kubernetes via EKS?
 - c. Demonstrar CI/CD com contêineres na nuvem?
3. Qual será a estrutura do curso?
 - a. Quantos encontros e qual a carga horária?
 - b. O curso será 100% prático? Haverá desafios/projetos?
4. Como garantir que o conteúdo seja acessível e significativo?
 - a. Posso usar analogias e metáforas para explicar conceitos complexos?
 - b. Vou oferecer tutoriais passo a passo e tempo para dúvidas?
5. Quais ferramentas e plataformas vou usar?
 - a. Usarei o AWS Academy, Cloud9, EC2, EKS, ou outro recurso de baixo custo?
 - b. Como os alunos vão praticar? Haverá um ambiente simulado? Sim, uma conta AWS (vou criar uma conta na AWS, configurar e disponibilizar para os alunos).
6. Como será feita a avaliação da aprendizagem?
 - a. Através de um projeto final?
 - b. Acompanhamento contínuo com desafios práticos?

A3. CONTRIBUIÇÕES DA MONITORIA ACADÊMICA NA FORMAÇÃO DO ALUNO-MONITOR DO CURSO DE ENFERMAGEM: RELATO DE EXPERIÊNCIA (FERNANDES *et al.*, 2025)

Metodologias do artigo:

- Aprendizagem ativa: Usar metodologias que incentivem o aluno a “aprender fazendo”, como labs práticos na AWS com Docker e Kubernetes.

- Monitoria e tutoria colaborativa: Incentivar a interação entre alunos e monitores/instrutores, criando um ambiente de troca de conhecimentos, onde dúvidas possam ser sanadas com proximidade e confiança.
- Estudos em grupo: Organizar grupos de estudo para discutir conceitos, resolver problemas e desenvolver projetos em conjunto, facilitando o aprendizado colaborativo.
- Uso de recursos digitais: Utilizar ferramentas online para comunicação rápida (ex: WhatsApp, Slack) para troca de materiais, esclarecimento de dúvidas e compartilhamento de experiências.
- Simulações e exercícios práticos: Propor exercícios práticos simulando situações reais de implantação e gestão de containers na AWS (ex: deploy de aplicações, escalabilidade com Kubernetes, troubleshooting).
- Autoavaliação e feedback: Promover momentos de reflexão e avaliação dos próprios alunos sobre o que aprenderam, dificuldades encontradas e estratégias para superá-las.

Questionamentos que você pode fazer para estimular o aprendizado

1. Quais as principais vantagens do uso de containers em ambientes de nuvem como AWS?
2. Como o Docker simplifica o desenvolvimento e o deploy de aplicações?
3. Quais são os principais componentes do Kubernetes e qual a função de cada um?
4. Como configurar um cluster Kubernetes na AWS (ex: EKS) e quais cuidados tomar?
5. Quais desafios podem surgir na orquestração de containers em produção? Como resolvê-los?
6. Como garantir alta disponibilidade e escalabilidade de aplicações containerizadas?
7. De que forma a monitoria e o trabalho em grupo podem ajudar na solução de problemas complexos?
8. Que boas práticas de segurança devemos seguir no uso de containers e Kubernetes?

A4. PROJETO “APRENDER VENDO E APRENDER FAZENDO”: UM RELATO DE EXPERIÊNCIA DO CURSO DE EDUCAÇÃO FÍSICA – UNIUBE (MOREIRA; VIEIRA, 2025)

Metodologias do artigo:

1. Aprender vendo e aprender fazendo
 - a. Estruturar o curso com momentos de demonstração (ver) e execução prática (fazer).
Ex.: Mostrar a criação de um cluster no EKS e depois os alunos fazerem o mesmo com apoio de um guia.
2. Protagonismo e autonomia do aluno
 - a. Permitir que os alunos escolham pequenos projetos práticos personalizados dentro do conteúdo apresentado.

Ex.: Implantar uma aplicação containerizada simples (ex: Node.js ou Python) em um cluster EKS com Load Balancer.

3. Integração entre teoria e prática
 - a. Trazer breves explicações teóricas seguidas de atividade prática imediata. Ex.: Após explicar o conceito de container, peça que criem um Dockerfile e buildem a imagem.
4. Projeto prático com relevância real
 - a. Orientar cada aluno (ou grupo) a escolher um pequeno projeto útil para seu contexto. Ex.: Criar um serviço web escalável usando pods no Kubernetes com balanceamento de carga.
5. Mentoria contínua e feedback personalizado
 - a. Manter canais abertos de comunicação (chat, vídeo, fórum) para orientação contínua, assim como o uso de webcast no AVA da Uniube.

Questionamentos que você pode fazer aos alunos para estimular o aprendizado significativo

1. Por que estou aprendendo isso?
 - a. Qual é a importância de entender containers na sua realidade profissional?
 - b. Como isso pode ser aplicado no seu ambiente de trabalho ou estudos?
2. Como posso aplicar isso na prática?
 - a. Você seria capaz de criar e executar uma aplicação real usando Docker?
 - b. Consegue imaginar um sistema que poderia beneficiar-se de um cluster Kubernetes?
3. Quais tecnologias eu quero dominar primeiro?
 - a. Prefere começar com Docker local ou já explorar o deploy em nuvem?
 - b. Deseja focar em EKS, ECS ou usar instâncias EC2 com kubeadm?
4. Que tipo de projeto prático posso desenvolver que seja útil para mim e/ou para a comunidade?
 - a. Você pode construir algo para resolver um problema comum na sua área?
 - b. Consegue criar um tutorial ou script que ajude outras pessoas?
5. Como posso avaliar meu progresso e resultados?
 - a. Que indicadores você vai usar para saber que entendeu o conteúdo?
 - b. Você se sente capaz de ensinar esse conteúdo para alguém depois do minicurso?

Quadro A1 - Sugestão de estrutura para o minicurso com base no artigo

Etapa	Ação do aluno	Papel do instrutor
Teoria rápida	Assiste a demonstrações e anota conceitos-chave	Apresenta temas com foco prático (ex: container, imagem, pod, cluster)
Prática guiada	Executa comandos e pequenos deploys com apoio	Dá feedback e tira dúvidas em tempo real
Projeto individual	Cria plano de ação para um mini deploy real	Atua como mentor e avalia projeto com base em critérios personalizados
Apresentação	Compartilha projeto com colegas (vídeo ou código)	Estimula interação e avaliação entre pares
Autoavaliação	Reflete sobre o que aprendeu e onde pode melhorar	Oferece formulário de reflexão + devolutiva

Fonte: Autoria própria

A5. RELATO DE EXPERIÊNCIA: CASOS DE ENSINO COMO FOMENTO DE ENSINO-APRENDIZAGEM – RELATO DE EXPERIÊNCIA NO CURSO DE ADMINISTRAÇÃO (GEMELLI; SANTOS; NUNES, 2025)

Metodologias do artigo:

- Método do caso de ensino: Apresentar situações reais ou problemas práticos que profissionais enfrentam ao usar Docker e Kubernetes na AWS. Promover discussões abertas para que os alunos possam analisar diferentes soluções e decisões.
- Aprendizagem ativa: Estimular o protagonismo dos alunos com atividades práticas, como montagem de clusters Kubernetes, deploy de containers Docker e resolução de problemas reais.
- Dramatização e simulação: Criar simulações de cenários onde os alunos possam assumir papéis (ex: DevOps, administrador, desenvolvedor) para resolver desafios de orquestração, escalabilidade e segurança na AWS.
- Uso de ferramentas variadas: Combinar textos, vídeos, tutoriais práticos e quizzes para diversificar o aprendizado e atender diferentes perfis de alunos.

Questionamentos para guiar o mini curso:

1. Quais são os principais desafios enfrentados na implementação de containers e orquestração na AWS?
2. Como aplicar o método do caso para resolver problemas reais em ambientes com Docker e Kubernetes?
3. De que forma a aprendizagem ativa pode contribuir para fixar o conhecimento em tecnologias de containers na nuvem?
4. Como garantir o protagonismo do aluno durante as atividades práticas e discussões?
5. Quais práticas e configurações são essenciais para a segurança e escalabilidade dos containers em ambientes AWS?
6. Como adaptar o conteúdo para diferentes níveis de conhecimento e tamanhos de turma?

A6. O LÚDICO NA APRENDIZAGEM DE ALUNOS DE CURSOS TÉCNICO, GRADUAÇÃO E MESTRADO: RELATOS DE EXPERIÊNCIAS (GROSSI, 2017)

Metodologias do artigo:

1. Jogos Didáticos Baseados em Desafios Técnicos
 - Adaptação: Criar desafios gamificados, como: “Corrida dos Contêineres”: um jogo de tabuleiro onde os alunos avançam ao resolverem tarefas como buildar uma imagem, criar um Pod, ou fazer deploy no EKS.
2. Café com Filme + Debate
 - Adaptação: Exibir documentários ou trechos de vídeos técnicos curtos (ex: do canal da AWS ou CNCF), e depois promova um debate guiado com perguntas críticas: O que esse vídeo mostra sobre automação de containers? Quais práticas vistas no vídeo você já utilizou ou pretende utilizar?
3. Caça-palavras ou quizzes
 - Adaptação: Criar caça-palavras ou quizzes com comandos Docker/K8s, terminologias da AWS (ECS, Fargate, EKS, IAM, etc). Eles podem ser usados como revisão divertida no final de cada módulo.

Questionamentos que você deve fazer ao planejar seu curso

Sobre objetivos pedagógicos:

1. Quais competências técnicas e comportamentais meus alunos devem desenvolver?
2. Como posso garantir que eles aprenderão de forma significativa e não apenas decorando comandos?

Sobre o perfil dos alunos:

1. Qual o nível de familiaridade dos alunos com Docker/Kubernetes/AWS?
2. Eles aprendem melhor com prática, teoria, vídeos, desafios?

Sobre metodologia:

1. Como transformar cada conceito técnico (ex: pod, cluster, container registry) em uma atividade prática ou lúdica?
2. Como posso manter o engajamento dos alunos em cada etapa do curso?

Sobre interações:

1. Como promover cooperação e trabalho em grupo, como o artigo sugere?
2. Como posso criar um ambiente seguro para erros e aprendizado?

Sobre avaliação:

1. Como saberei se os alunos realmente entenderam os conceitos?
2. A avaliação pode ocorrer por projetos? Por desafios? Por autoavaliação?

Quadro A2 - Sugestão de atividades práticas no estilo do artigo

Conteúdo	Atividade lúdica sugerida
Conceitos básicos de Docker	Montar um puzzle (quebra-cabeça) com partes de um Dockerfile embaralhadas
Kubernetes	Jogo de tabuleiro com YAMLS: alunos avançam ao identificar falhas de configuração
Deploy na AWS (ECS/EKS)	Simulação em grupo: cada grupo assume um papel (DevOps, QA, Dev) e precisa lançar uma app integrada na AWS
IAM e segurança	Jogo de "controle de acesso": cartões de permissão e negação simulando papéis de usuário

Fonte: Autoria própria

A7. AVALIAÇÃO DE METODOLOGIA DE ENSINO CENTRADA NO ALUNO (HERNANDEZ NETO, 2000)

Metodologia centrada no aluno (modelo tecnológico)

- Utilize recursos como vídeos, tutoriais, ambientes interativos na nuvem, laboratórios práticos, simulações em tempo real, videoconferências e fóruns.
- Conhecimentos: Conceitos de container, imagens Docker, pods, clusters Kubernetes.
- Habilidades: Subir containers na AWS, usar EKS, criar arquivos YAML, configurar serviços.
- Atitudes: Estimular curiosidade, responsabilidade e capacidade de resolver problemas reais em nuvem.

Sistema de controle e avaliação contínua:

- Avalie os alunos com base em atividades práticas (deploys, resolução de erros, automação), feedback contínuo, e projetos aplicados.
- Considere se os alunos têm acesso à Internet, contas AWS, e conhecimento básico em Linux ou terminal. Planeje os pré-requisitos de entrada no curso.

Questionamentos:

1. O que quero que o aluno saiba, saiba fazer e como quero que ele se comporte após o curso?
2. Quais pré-requisitos os alunos precisam ter para acompanhar o curso com aproveitamento?
3. Conhecimentos prévios em computação, terminal, nuvem, Docker básico?
4. Quais meios educacionais são mais eficazes para alcançar meus objetivos?
5. Vale mais a pena usar um laboratório online? Um tutorial guiado? Um vídeo demonstrativo?
6. Como avaliar se o aluno alcançou os objetivos definidos?
7. Usarei um projeto final? Checklists de atividades? Avaliações práticas?
8. O conteúdo e as ferramentas estão de acordo com a realidade dos meus alunos?
9. Eles têm acesso aos recursos da AWS? Sabem usar o terminal? Conseguem acompanhar vídeos gravados?