

**INSTITUTO FEDERAL GOIANO – CAMPUS CERES  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
ADRIANO XAVIER NOBRE JÚNIOR**

**FINOPS: Práticas de Padronização de Rótulos para Otimização de  
Custos**

**CERES – GO  
2024**

**ADRIANO XAVIER NOBRE JÚNIOR**

**FINOPS: Práticas de Padronização de Rótulos para Otimização de  
Custos**

Trabalho de curso apresentado ao curso de Sistemas de Informação do Instituto Federal Goiano – Campus Ceres, como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação, sob orientação do Prof. Ms. Roitier Campos Gonçalves.

**CERES – GO  
2024**





SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

### ATA DE DEFESA DE TRABALHO DE CURSO

Aos 09 dia(s) do mês de agosto do ano de dois mil e 2024, realizou-se a defesa de Trabalho de Curso do acadêmico ADRIANO XAVIER NOBRE JÚNIOR, do Curso de BACHARELADO EM SISTEMAS DE INFORMAÇÃO, matrícula 2020103202030372, cujo título é "FINOPS: Práticas de padronização de rótulos para otimização de custos". A defesa iniciou-se às 14 horas e 10 minutos, finalizando-se às 15 horas e 15 minutos. A banca examinadora considerou o trabalho APROVADO com média 7,2 no trabalho escrito, média 8,93 no trabalho oral, apresentando assim média aritmética final de 8,06 pontos, estando o estudante APTO para fins de conclusão do Trabalho de Curso.

Após atender às considerações da banca e respeitando o prazo disposto em calendário acadêmico, o(a) estudante deverá fazer a submissão da versão corrigida em formato digital (.pdf) no Repositório Institucional do IF Goiano – RIIF, acompanhado do Termo Ciência e Autorização Eletrônico (TCAE), devidamente assinado pelo autor e orientador.

Os integrantes da banca examinadora assinam a presente.

(Assinado Eletronicamente)  
Roitier Campos Gonçalves

(Assinado Eletronicamente)  
Hugo de Moura Campos

(Assinado Eletronicamente)  
Livia Mancine Coelho de Campos

Documento assinado eletronicamente por:

- Livia Mancine Coelho de Campos, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 16/08/2024 14:02:35.
- Hugo de Moura Campos, TECNICO DE LABORATORIO AREA, em 12/08/2024 17:14:47.
- Roitier Campos Goncalves, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 12/08/2024 17:02:08.

Este documento foi emitido pelo SUAP em 12/08/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 622316  
Código de Autenticação: 7239f97342



INSTITUTO FEDERAL GOIANO  
Campus Ceres

Rodovia GO-154, Km 03, SN, Zona Rural, CERES / GO, CEP 76300-000



## RESUMO

Durante o desenvolvimento de projetos de aplicações é muito comum que seja aplicado os modelos de *Cloud* (nuvem), atualmente pela popularização desses serviços em *Cloud* mas, apesar da praticidade e baixo custo financeiro em comparação com outros modelos de pode-se encontrar possíveis problemas de mapeamento de recursos e isso gera uma infraestrutura em *Cloud* que é altamente provisionada sem saber de fato os ofensores dos recursos provisionados na *Cloud*.

Devido a esse alto provisionamento, surge a necessidade de implementar projetos de *FinOps* dentro dos projetos com o objetivo de reduzir custos e mapear os responsáveis pelos recursos seguindo um padrão, será realizado uma exemplificação de projeto e trabalhado com ferramentas.

Aplicando esses padrões serão feitas buscas e filtros com as informações existentes e dentro do contexto de *FinOps* deverá ser utilizada como uma busca otimizada para os recursos presentes em *Cloud* e trará maior visibilidade do problema trazendo uma análise dos resultados obtidos que surgem como uma alternativa viável a ser implementada.

**Palavras-chave:** *FinOps*, *Cloud*, Otimização, Padronização.

## **ABSTRACT**

During the development of application projects, it is very common for Cloud models to be applied, currently due to the popularization of these Cloud services, but despite the practicality and low financial cost compared to other models, possible problems may be encountered. resource mapping and this generates a Cloud infrastructure that is highly provisioned without actually knowing the offenders of the resources provisioned in the Cloud.

Due to this high provisioning, there is a need to implement FinOps projects within projects with the aim of reducing costs and mapping those responsible for resources following a standard. A project example will be carried out and worked with tools.

Applying these standards, searches and filters will be carried out with the existing information and within the context of FinOps it should be used as an optimized search for the resources present in the Cloud and will bring greater visibility of the problem by bringing an analysis of the results obtained that emerge as a viable alternative to be implemented.

**Keywords:** *FinOps, Cloud, Optimization, Standardization.*

## LISTA DE ILUSTRAÇÕES

|  |    |
|--|----|
| Figura 1 - Estratégia de operações de <i>FinOps</i> .....  | 14 |
| Figura 2 - Estratégia de divisão de Times.....   | 21 |
| Figura 3 – Relação estratégica dos recursos com os times.....  | 21 |
| Figura 4 – Execução do primeiro comando no terminal.....   | 23 |
| Figura 5 – Resposta do primeiro comando no terminal.....   | 23 |
| Figura 6 – Comando com os filtros aplicados para <i>tags</i> e identificadores das máquinas.....         | 24 |
| Figura 7 – Comando com filtros e tratamentos de apenas valores específicos.<br>.....                     | 25 |
| Figura 8 – Comando com filtros e tratamentos de apenas valores específicos e diferentes do anterior..... | 25 |



## SUMÁRIO

|   |           |
|---|-----------|
| <b>1. INTRODUÇÃO.....</b>                   | <b>9</b>  |
| 1.1 Cloud Computing.....                    | 10        |
| 1.2 FinOps.....                             | 12        |
| 1.3 Justificativa.....                      | 14        |
| 1.4 Motivação.....                          | 16        |
| 1.5 Objetivos.....                          | 17        |
| 1.6 Referencial Teórico.....                | 17        |
| 1.6.1 Amazon Web Services(AWS).....         | 18        |
| 1.6.1.1 AWS Elastic Compute Cloud(EC2)..... | 18        |
| 1.6.2 Jq(Json Processor).....               | 19        |
| 1.6.3 Kebab Case.....                       | 19        |
| 1.6.4 Pascal Case.....                      | 19        |
| <b>2. METODOLOGIA.....</b>                  | <b>20</b> |
| 2.1 Experimento.....                        | 21        |
| 2.2 Análises.....                           | 22        |
| 2.3 Resultados.....                         | 26        |
| <b>3. CONCLUSÃO.....</b>                    | <b>26</b> |
| <b>4. REFERÊNCIAS.....</b>                  | <b>27</b> |

## 1. INTRODUÇÃO

Durante os últimos anos a grande adoção de *Cloud Computing* (Computação em Nuvem) e seus recursos por instituições têm gerado uma grande popularização dos termos relacionados ao termo, e tendo isso como base, o surgimento de novas tecnologias e *Frameworks* veio crescendo junto com as necessidades que surgiam em paralelo com a *Cloud*.

Portanto, têm sido constantes os projetos que envolvem a *Cloud*, mas, mesmo que a utilização da *Cloud* seja atrativa para as instituições, esses *Clouds* podem acabar sendo um problema.

Apesar de se sustentarem muito bem tendo pouca necessidade de se realizar manutenções à medida que os projetos vão melhorando e aumentando os recursos em *Cloud*, há pontos que podem passar despercebidos como o custo de gerenciamento e provisionamento da *Cloud* (*FinOps Foundation, 2023*).

O *FinOps*, que é o conceito que envolve a construção e desenvolvimento de um novo time responsável pelo trabalho de parte do time Financeiro e o time Técnico de Operações, busca atingir uma melhor eficiência na projeção de valores em relação a esses recursos usados em *Cloud*, vindo como uma possível solução ao problema.

Com isso o surgimento da ferramenta *FinOps* se tornou algo muito importante para os projetos, visto que o mesmo tem como base o gerenciamento dos custos e melhoria dos recursos levando em consideração o custo e a performance do recurso.

Apesar de *FinOps* estar bem definido, um dos problemas que surge durante a aplicação do *Framework* é a não padronização da estratégia de Rotulamento, acarretando em atrasos e problemas no funcionamento geral do projeto (*Importância de Tags para a Comunidade de FinOps, 2023*).

O presente trabalho visa criar um modelo base de estratégia de Rotulamento para esses projetos com o objetivo de fornecer uma possível solução para esse problema.

### **1.1 *Cloud Computing***

Antes dos serviços de *Cloud Computing* se tornarem populares, as instituições usavam sistemas *on-premises* para recursos computacionais, onde toda a infraestrutura era mantida por seus funcionários ou por parceiros, o que gerava uma série de transtornos em casos de problemas, atualizações ou até procedimentos cotidianos, por se tratar em especial de sistemas físicos, levantando questões geográficas durante o processo de montagem de um sistema.

Para a expansão e manutenção da capacidade computacional destes sistemas, também era demandado muito tempo, uma vez que o processo envolve diversas etapas, tais como a de compra dos equipamentos, que deve ser aprovado pelo setor financeiro da empresa, além de comunicação prévia com todas as partes interessadas dos sistemas e das aplicações que seriam impactadas com indisponibilidades decorrentes destas ações.

Durante a primeira década dos anos 2000 percebeu-se que houve uma crescente difusão de termos como *Cloud* e *Cloud Computing*. Embora essas palavras fossem novidades para o público de maneira geral, já eram conhecidas nos últimos anos de 1990, difundidas principalmente como “redes de computadores” e “servidores interconectados”, que tinham como objetivo oferecer disponibilidade e acessibilidade a redes de informações (Cloud: Guia da computação em nuvem, 2023).

De acordo com os autores Mell & Grance (2011) a definição de *Cloud Computing* definida no documento do *National Institute for Standards in Technology* (NIST) é a seguinte:

*Cloud Computing* é autosserviço sob demanda de amplo acesso à rede e possui três modelos de serviço, tais como, software como serviço; plataforma como serviço; infraestrutura como serviço, e quatro modelos de implementação, como se segue: nuvens privadas; nuvens comunitárias; nuvens públicas; nuvens híbridas.

A definição descrita acima apresenta de maneira clara o que presenciamos na última década e a década atual, pois com o constante crescimento dos serviços de Tecnologia da Informação (TI) os investimentos na área de *Cloud Computing* cresceram de forma considerável (Convergência Digital, 2022).

Como exemplo da definição mencionada, podemos considerar um serviço e modelo que define a criação de ambientes hospedados em *Cloud* para que haja disponibilidade de aplicações, esse ambiente pode ser hospedado por exemplo na *Amazon Web Services(AWS)* que é um provedor com vários serviços que se encaixam nas categorias acima, por exemplo, que pode fazer ou não todo o gerenciamento dessa aplicação.

Os projetos ou instituições que consomem serviço de *Cloud Computing*, geralmente não têm uma visão para controlar e projetar de forma clara os servidores de *Cloud Computing*, e podem sofrer com gastos acima do previsto e surge então a necessidade de aplicar novos projetos gerando mais gastos.

Um exemplo desse tipo de situação pode ser verificado no estudo de caso apresentado pela empresa Avanade<sup>1</sup> o qual após a execução de um projeto para redefinir esse escopo, houve uma redução média de 30% do custo anual do cliente e isso sem prejudicar o desempenho dos produtos finais do cliente (Estudo de caso da Bauducco FinOps, Avanade).

O artigo também traz em si o conceito de um *Framework* que é uma estrutura de modelos a serem seguidos obtendo melhores resultados e maior eficiência na produção e execução de projetos, no contexto, ele surge como *Framework FinOps* que descreve como analistas de negócios, financeiros e de suporte técnico podem

executar para solucionar os problemas relacionados com os gerenciamento de custo da *Cloud* (Leite, 2023).

## 1.2 FinOps

O *FinOps* surgirá com o objetivo de gerenciar e otimizar os gastos de projetos de *Cloud* podendo proporcionar melhores desenvolvimentos de níveis de negócios em instituições que geralmente se encontram em "*Foundations Levels*" quando passam por esse tipo de projeto e também proporciona as tomadas de decisões de gastos baseados em dados (*FinOps Foundation*, 2023).

O *FinOps* pode ser apresentado com princípios bastante significantes para a engenharia de *software* como a colaboração de times, especialização de times focados em *FinOps*, o ato de reportar problemas que devem ser acessíveis e tratados o mais rápido possível, as decisões que devem ser dirigidas pelo valor comercial de *Cloud* e pela obtenção de vantagens pelo modelo de gastos variável de *Cloud* (*FinOps Foundation*, 2023).

Há de ser considerado o fato de que instituições com estruturas de "*Foundations Levels*" precisam de um maturidade para lidar com esses problemas, para isso o *FinOps* argumenta sobre.

A prática da metodologia poderá melhorar o desempenho dos colaboradores com o tempo, portanto usa-se de termos como "*Crawl, Walk, Run*" que em tradução se tornam "Rastejar, Andar, Correr" uma analogia para executar *FinOps* que permite que as organizações iniciem pequenas e cresçam em escala, escopo e complexidade conforme o valor comercial que garantirá o amadurecimento de uma atividade mais funcional e menos fundamental (*FinOps Foundation*, 2023).

Isso significa que os times de *FinOps* podem começar na etapa de rastejar para conseguir alcançar a corrida de forma gradual, passando de um perfil

profissional menos reativo quanto a tratativa de problemas e sendo um profissional mais preventivo (*FinOps Foundation, 2023*).

O *FinOps* apesar de ter um escopo bem definido jamais irá andar isolado, os seus participantes são em sua maioria o Analista *FinOps*, Engenheiros de Operações que em sua maioria cuidam da manutenção e sustentação dos serviços em *Cloud*, *Product Owner* que geralmente busca os interesses de todos os envolvidos a fim de satisfazer todas as necessidades do produto, Financeiro e Executivos que produzem em sua esfera gerenciamento dos gastos (*FinOps Foundation, 2023*).

No geral o *FinOps* sugere que haja uma equipe somente de para cuidar da implementação do *framework* e que irá fazer grande parte do trabalho gerando até mesmo relatórios (*FinOps Foundation, 2023*).

Esses relatórios contêm relações de provisionamento e determina quais recursos podem ser subdimensionados ou superdimensionados, são feitos de modo que seja possível identificar acréscimo ou redução nos custos (*FinOps Foundation, 2023*).

Na etapa de adoção de *FinOps* a empresa vai adaptar 3 fases:

1ª - Informar: etapa onde o profissional faz um levantamento com ferramentas para que seja possível monitorar os gastos da *Cloud* e com alocação de um limite de gastos para a mesma, tendo visão dos gastos a serem feitos e provisionar um painel de custos, isso seguido do Rotulamento dos recursos.

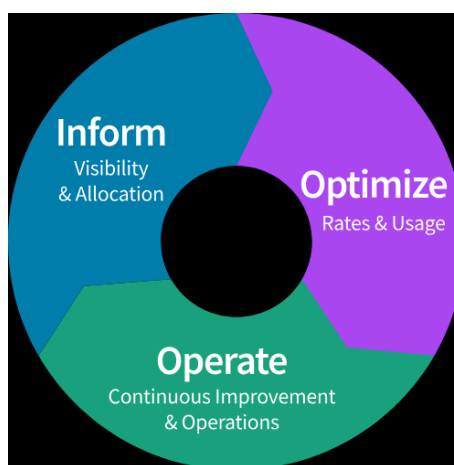
O processo de Rotulamento pode ser considerado um dos mais importantes visto que ele proporcionará a visibilidade de quais recursos e quanto desses recursos estão sendo gastos.

2ª - Otimizar - etapa após realizado o controle e a adoção da *Cloud* deve-se promover então a otimização de custos e configurar o uso apenas do necessário com a política do “*pay for what i need*”(pagar pelo que preciso) em detrimento da

“pay for what i use”(pagar pelo que uso), tendo planos de *savings* que serão resultantes na redução de custos e adoção de *stacks*(ferramentas) de automação.

3ª - Operar - etapa onde se mantém o que já fora aplicado para continuar tendo descontos e buscando formas de ter observabilidade dos gastos e com o objetivo de ter um ciclo que sempre retorna ao ponto inicial.

**Figura 1** - Estratégia de operações de *FinOps*.



**Fonte:** *FinOps Foundation*.

Tendo isso em vista, o profissional transita entre áreas e conceitos importantes que podem vir a trazer grande valor para projetos e produtos, além de existir uma constante reorganização de gastos, podendo trazer mais projetos com o que foi economizado.

### **1.3 Justificativa**

O modo como esses problemas financeiros se expressam em projetos fazem com que o *FinOps* seja necessário como solução e como justificativa do desenvolvimento deste trabalho trazendo casos relatados que se inserem na métrica

de projetos bem planejados mas que se perde no quesito maturidade e visibilidade de gastos.

Aproveitando-se dessa métrica de projetos que não usam nenhuma prática de e são obrigados a repassar por uma redefinição de arquitetura como foi apresentado no trabalho “Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%.” é reforçada a ideia de se trabalhar em padronizações de construção de infraestrutura.

O *FinOps* pode entrar na área de infraestrutura para que faça a adaptação necessária para evitar tais problemas. Trazendo a maturidade e a visibilidade financeira necessária para as instituições que estão em passos iniciais de projetos que envolvem *Cloud* e podem futuramente passar pelo problema.

O presente trabalho tem como objetivo aplicar um método de Rotulamento durante a 1ª etapa do *Framework* esse método será feito para que se tenha melhor visibilidade dos gastos, uma sugestão de dicionário de padronização será apresentado para que se tenha mapeado os recursos de *Cloud* de um projeto.

A complexidade deste projeto surge no momento onde cada desenvolvedor pode sugerir como deve ser rotulado seu recurso, tendo em vista os projetos e as diferenças de metodologia de trabalho entre os desenvolvedores (LUNA et al., 2010).

O Rotulamento também pode ser associado com dicionário de valores onde quando se insere uma chave, obtém-se um valor esperado. Então uma das formas de tratar esse problema no momento de Rotulamento seria passar uma padronização, tendo em vista que uma mesma chave pode ter diversas variações, assim como os valores. Observe o exemplo na Tabela 1 para o problema:



Tabela 1 - Quadro de exemplo de Rotulamentos.

| <b>Chave</b>       | <b>Valor</b>     |
|--------------------|------------------|
| <i>application</i> | <i>ecommerce</i> |
| <i>Aplication</i>  | <i>ecommerce</i> |
| <i>Application</i> | <i>Ecommerce</i> |

Levando esse exemplo apresentado na Tabela 1, observa-se que é necessário uma padronização pois pode-se ter uma falta de entendimento de onde pode vir esses valores rotulados. Com isso tem-se o objetivo de ter um modelo inicial para o uso do *Framework* trazendo como uma hipótese de aplicação de um dos processos que pode ser bastante complexo e pouco aproveitado por todos os times.

#### **1.4 Motivação**

O presente trabalho visando criar um modelo base de estratégia de Rotulamento para esses projetos, se motiva partindo do ponto principal que seria maturidade de projeto a fim de obter o melhor nível em etapas iniciais que são importantes estarem consolidadas, construindo um modelo de padronização e demonstrando como ele pode ser útil no dia-a-dia (Maturidade em gestão de projetos: o que é, como definir e exemplos de matriz de maturidade, 2022).

Buscando trazer a visibilidade necessária do assunto, um dos objetivos a ser alcançado seria a utilização desse modelo para fazer o mapeamento de recursos baseados em Rótulos, uma estratégia que pode trabalhar em conjunto com o *Framework FinOps*, trazendo de forma simples o gerenciamento básico de recursos.

## **1.5 Objetivos**

Com o que foi demonstrado pode-se buscar o *Framework* para o desenvolvimento de estratégias que vão proporcionar novas modelagens para definições em projetos, aos quais já prevêem a empresa de um novo projeto focado nesse tema.

Sendo possível aderir uma padronização no escopo do Rotulamento e nos processos de desenvolvimento no decorrer do projeto, isso vai definir também uma relação harmoniosa entre os times de engenharia, arquitetura e financeiro do projeto promovendo em especial o mapeamento dos recursos de uma forma inicial básica.

De forma prática será apresentado uma forma de implementação de um dos processos de *FinOps* dentro do processo de desenvolvimento e da implantação do projeto, ao qual será usado um exemplo de infraestrutura e uma prática do *Framework*, será feito um levantamento e uma identificação de diferentes recursos criados em *Cloud* e localizá-los buscando facilitar para uso futuro o mapeamento desses gastos com a funcionalidade de rotulação de aplicações.

Essa prática será feita por meio da padronização já comentada e levantada anteriormente que será o objetivo principal do trabalho com ênfase em definir valores padrões de exemplo e seguir regras de nomenclaturas.

## **1.6 Referencial Teórico**

Nessa seção serão apresentados ferramentas e estratégias que serão usadas na realização do trabalho, as ferramentas utilizadas e escolhidas foram selecionadas pois fazem parte das principais ferramentas utilizadas no mercado de trabalho atual e tendo grandes quantidades de materiais disponíveis sobre.

### **1.6.1 Amazon Web Services(AWS)**

A AWS é um provedor de serviços de *Cloud* que começou suas atividades oficialmente em 2006 e com o tempo se tornou o mais conhecido do mundo, um dos motivos seria o fato de ser a mais barata e com uma enorme gama de serviços para uso. Será utilizado o serviço de *Elastic Compute Cloud*(EC2) que cria computador virtual usando os servidores como origem, é possível especificar RAM, ROM, CPU e Sistema Operacional (“O que é AWS? Como funciona Amazon Web Services”, 2023).

A AWS fornece uma ferramenta para terminais de Sistemas Operacionais que serve para gerenciar todos os seus recursos de nuvem que o usuário possa ter em sua conta, e é mantida pela mesma, conhecida como *Amazon Web Service Command Line Interface* (AWS CLI) (“AWS CLI - Interface de linha de comando - Amazon Web Services”, 2023) (APÊNDICE A).

Sendo a AWS um dos provedores mais utilizados na atualidade, além de oferecer uma grande gama de serviços com baixos custos, é uma das provedoras que mais têm materiais disponíveis (Computação em nuvem em 2024 e os 10 principais fornecedores de serviços em nuvem, 2024).

#### **1.6.1.1 AWS Elastic Compute Cloud(EC2)**

O Serviço web de EC2 pode ser usado para criar, gerenciar e redimensionar máquinas virtuais em servidores AWS provisionados para atender ao público necessário e pensando em possíveis alterações no decorrer do tempo, sendo o principal e mais básico recurso fornecido pela AWS aos consumidores dos seus serviços. O serviço pode ser utilizado via `aws-cli` especificamente com o termo “`aws ec2`” (“Elastic Compute Cloud - Amazon EC2 - AWS”, 2023) (APÊNDICE B).

### **1.6.2 Jq(Json Processor)**

Jq é um formatador de dados em formato Json usado geralmente nos terminais de Sistemas Operacionais, sua utilização é muito comum em scripts para fazer tratamentos de dados pois possui a vantagem de ser uma ferramenta a parte que não depende de uma linguagem de programação ou de um pacote na mesma para ser realizado seus processos, sendo conhecido por ser muito rápido na execução via terminal e bem leve visto que foi desenvolvido em *Portable-C*(Jq, 2024).

### **1.6.3 Kebab Case**

O *Kebab Case* é uma convenção de nomenclatura que define boas práticas de identificadores como recomendação, que traz consigo a ideia de se ter uma maior clareza e melhorar a forma como é visualizada no código, tornando o mesmo mais limpo e até mesmo tornar identificadores autoexplicativos para os recursos que o utilizam. Nesse caso o *Kebab Case* é usado com letras minúsculas e sendo usado “-” para a combinação de palavras(“Convenções de nomenclatura: Camel, Pascal, Kebab e Snake case”, 2024).

### **1.6.4 Pascal Case**

O *Pascal Case* é mais uma convenção de nomenclatura que define-se da mesma forma que a anterior porém ressaltando a diferença de uso das duas que nesse caso a primeira letra do identificador deve ser maiúscula e em caso de identificador composto, onde começaria o segundo identificador, deve-se usar outra letra maiúscula ao invés do “-”(“Convenções de nomenclatura: Camel, Pascal, Kebab e Snake case”, 2024).

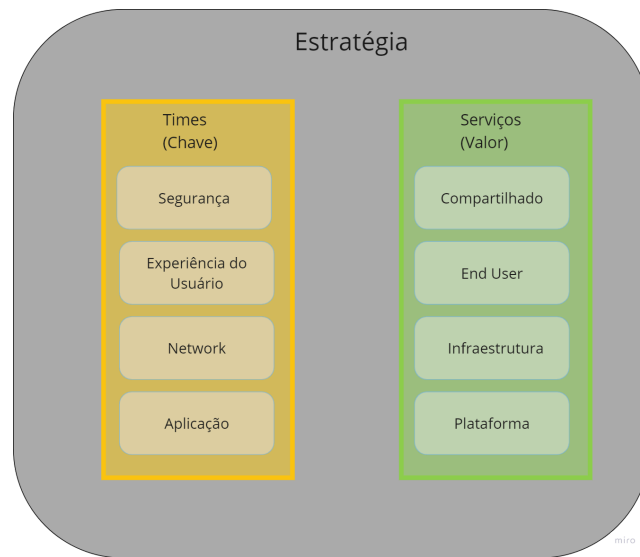
## 2. METODOLOGIA

Para que seja realizada uma boa aplicação de *Cloud* e boa gestão de custos de *Cloud*, o *FinOps* se propõe como solução conhecida, e como visto anteriormente, uma das estratégias que o *Framework* emprega é o Rotulamento de recursos para ser implementado um tipo de mapeamento de custos dos recursos. Portanto surge a possibilidade de encontrar os times responsáveis e tratar uma forma de melhorar ou redirecionar aquele custo.

Será utilizado um método de Rotulamento como descrito nos objetivos que devem fornecer uma separação das aplicações, a qual terá o objetivo de minimizar aplicações sem times definidos, trazendo um ar menos reativo do *FinOps* e mais sendo mais preventivo na sua atuação.

A estratégia de padronização será feita por meio de uma divisão de times e serviços, onde os times compartilharam os serviços por meio da estratégia de rótulos, como pode ser visto na Figura 2:

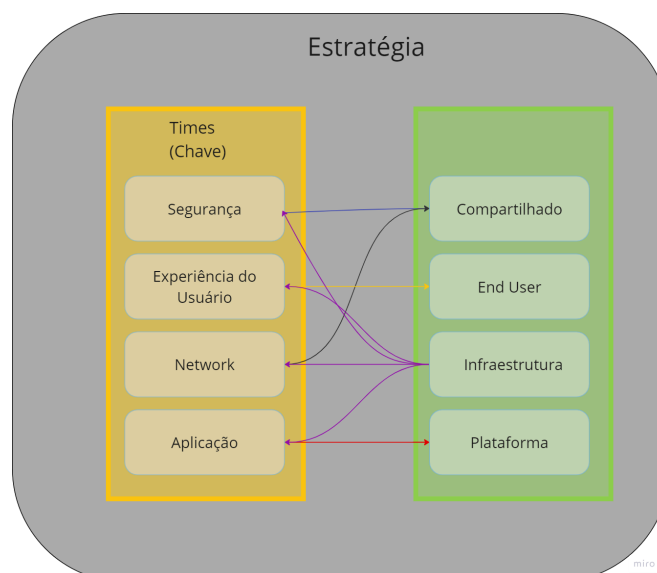
**Figura 2 - Estratégia de divisão de Times.**



**Fonte:** Do Autor.<sup>1</sup>

Esses recursos terão relacionamentos entre si que poderão ajudar a localizar e identificar recursos dos diferentes times, como pode ser visto Figura 3:

**Figura 3 – Relação estratégica dos recursos com os times.**



**Fonte:** Do Autor.

---

<sup>1</sup> Esquema de exemplo montada pelo autor via Miro, uma plataforma de gerenciamento de projetos.

## 2.1 Experimento

Tendo isso abordado, foi realizada a criação de uma conta na AWS e configurada para ser acessível do terminal da máquina de acesso seguindo a documentação e recomendação da própria provedora AWS com o objetivo de ser mais objetivo (APÊNDICE A).

Em seguida, foram criados servidores na AWS com rótulos específicos, no padrão *Kebab Case* e *Pascal Case*, novamente seguindo as recomendações e documentação da provedora (APÊNDICE B).

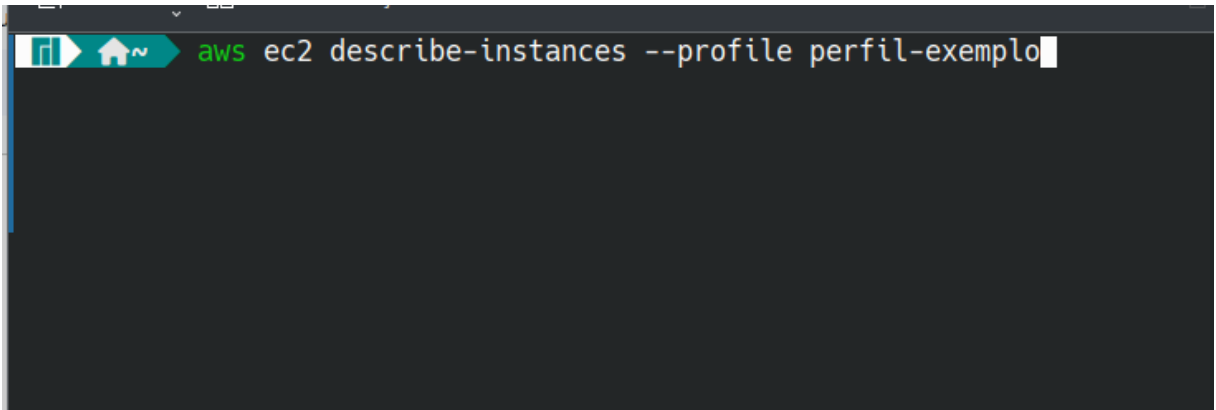
## 2.2 Análises

As análises foram realizadas considerando que os apêndices devem ser respeitados. Posteriormente foi iniciado o procedimento, foram usados diversos comandos no terminal com o AWS CLI, para isso foram executados os comandos “aws --profile perfil-exemplo ec2 describe-instances” na Figura 4, que vai dar um retorno em *json* de todas as instâncias na região, mas será necessário informações específicas para o tratamento, como estamos vendo o Rotulamento, vamos querer filtrar o *json* usando o *jq*.

A saída sendo em *json* existem pontos importantes a serem considerados, como os dados são separados nas seções, há a necessidade de se realizar o filtro do que realmente queremos encontrar. Sabe-se que a seção onde ficam os rótulos, chamados de *tags* em inglês, é um objeto dentro da lista *Instance* que é populado com diversos outros objetos, pode ser observado na Figura 5.

Para chegarmos nesse filtro, torna-se necessário executar outro comando com a saída do primeiro porém não será exibido somente o elemento *tags*, então é realizado um novo filtro com o *jq* como apresentado na Figura 6, com essa nova saída de dados:

**Figura 4** – Execução do primeiro comando no terminal.



```
aws ec2 describe-instances --profile perfil-exemplo
```

Fonte: Do Autor.

**Figura 5** – Resposta do primeiro comando no terminal.



```
{
  "Reservations": [
    {
      "Groups": [],
      "Instances": [
        {
          "AmiLaunchIndex": 0,
          "ImageId": "ami-07caf09b362be10b8",
          "InstanceId": "i-0bde69b97877299d4",
          "InstanceType": "t2.micro",
          "LaunchTime": "2024-05-07T02:37:26+00:00",
          "Monitoring": {
            "State": "disabled"
          },
          "Placement": {
            "AvailabilityZone": "us-east-1d",
            "GroupName": "",
            "Tenancy": "default"
          },
          "PrivateDnsName": "ip-172-31-22-171.ec2.internal",
          "PrivateIpAddress": "172.31.22.171",
          "ProductCodes": [],
          "PublicDnsName": "",
          "State": {
            "Code": 80,
            "Name": "stopped"
          },
          "StateTransitionReason": "User initiated (2024-05-09 02:14:25 GMT)",
          "SubnetId": "subnet-086e51a38df786fa3",
          "VpcId": "vpc-02ba19ea3ca9b0c6b",
          "Architecture": "x86_64",
          "BlockDeviceMappings": [
            {
              "DeviceName": "/dev/xvda",
              "Ebs": {
                "AttachTime": "2024-05-07T02:37:27+00:00",
                "DeleteOnTermination": true,
                "Status": "attached",
                "VolumeId": "vol-02b17309a0bc412b1"
              }
            }
          ]
        }
      ]
    }
  ],
}
```

Fonte: Do Autor.



**Figura 6** – Comando com os filtros aplicados para *tags* e identificadores das máquinas.

```
aws ec2 describe-instances --profile perfil-exemplo | jq '.Reservations[].Instances[] | jq -s '.[] | {InstanceId, Tags}'
```

```
{
  "InstanceId": "i-0bde69b97877299d4",
  "Tags": [
    {
      "Key": "Name",
      "Value": "Seguranca"
    },
    {
      "Key": "Compartilhado",
      "Value": "true"
    },
    {
      "Key": "infraestrutura",
      "Value": "true"
    }
  ]
}
{
  "InstanceId": "i-0886d7ea6579b3e9d",
  "Tags": [
    {
      "Key": "Compartilhado",
      "Value": "true"
    },
    {
      "Key": "Infraestrutura",
      "Value": "true"
    },
    {
      "Key": "Name",
      "Value": "Network"
    }
  ]
}
{
  "InstanceId": "i-04d7ff109e3514a06",
  "Tags": [
    {
      "Key": "Infraestrutura",
      "Value": "true"
    }
  ]
}
```

**Fonte:** Do Autor.

Para ser específico os valores padrões sugeridos foram “infraestrutura” apresentado na Figura 7 e “Infraestrutura” apresentado na Figura 8. Essa diferença irá trazer o principal objetivo levantado, o problema em localizar servidores com rótulos e a necessidade de ser um conhecimento unificado. Serão testadas chamadas com dois valores diferentes para identificação do problema tratado.

**Figura 7** – Comando com filtros e tratamentos de apenas valores específicos.

```
aws ec2 describe-instances --profile perfil-exemplo | jq '.Reservations[].Instances[]' | jq -s '[]' | {InstanceId, Tags} | {InstanceId, Tags: .Tags[] | select(.Key == "infraestrutura") | .Value}'
{
  "InstanceId": "i-0bde69b97877299d4",
  "Tags": "true"
}
```

Fonte: Do Autor.

**Figura 8** – Comando com filtros e tratamentos de apenas valores específicos e diferentes do anterior.

```
aws ec2 describe-instances --profile perfil-exemplo | jq '.Reservations[].Instances[]' | jq -s '[]' | {InstanceId, Tags} | {InstanceId, Tags: .Tags[] | select(.Key == "Infraestrutura") | .Value}'
{
  "InstanceId": "i-0886d7ea6579b3e9d",
  "Tags": "true"
}
{
  "InstanceId": "i-04d7ff109e3514a06",
  "Tags": "true"
}
{
  "InstanceId": "i-076651df9c7ad9a91",
  "Tags": "true"
}
```

Fonte: Do Autor.

### 2.3 Resultados

Pode ser observado que com essa análise que as práticas de padronização devem ser implementadas de forma unificada e alinhada com os times para que o

processo de Rotulamento não seja atrapalhado por esses impedimentos que podem impactar em entregas de projetos e atrapalhar em futuras melhorias.

Esses resultados obtidos nesta análise, mostra que seguindo a metodologia obtém-se uma a exclusão da necessidade de tratamento de outros possíveis valores de chaves fazendo-se buscas mais otimizadas e abrindo possibilidades para tratamentos de visualização das informações apresentadas, o que esse trabalho não se propõe a fazer.

Usar Convenções de Nomenclatura também pode trazer uma maior maturidade ao projeto o que torna a gestão de projetos dentro da empresa mais madura e mais preocupada em outros pontos apresentados e possíveis melhorias posteriores (Maturidade em gestão de projetos: o que é, como definir e exemplos de matriz de maturidade, 2022).

### **3. CONCLUSÃO**

Como abordado, o trabalho visou aplicar um exemplo de padronização no processo de Rotulamento que faz parte da metodologia de *FinOps*, sendo existente em um ponto que ainda pode ser falho dentro dos projetos e recursos, onde os mesmos geralmente não têm padronização desses Rotulamentos e criam esses despadronizados ou sem rótulos.

A metodologia aplicada no presente trabalho buscou uma alternativa de solução ao problema e atingir uma maturidade dentro dos projetos ainda não alcançada para que se alcance maiores melhorias e gerenciamento de custos com a ferramenta de *FinOps*, esses passos iniciais sendo a base para todo o desenvolvimento futuro de *FinOps*.

Nas análises, vimos que o uso de uma convenção de nomenclatura como padrão deve ser adotado e seguido para que se tenha um mapeamento inicial de recursos bem definido e fácil de ser transformado em rotinas automatizadas como

*scripts* de execução em tempo real para que seja feito esse mapeamento rápido e confiante.

Portanto o uso e promoção desses métodos devem ser constantes dentro das organizações com departamentos de Tecnologia da Informação e nas organizações educacionais, que está responsabilizado por trazer e ser um importante foco para disseminação de ferramentas e processos bem estabelecidos e estruturados no mercado.

#### 4. REFERÊNCIAS

**ALM, Cloud & DevOps.** Disponível em: <<https://leonardo-matsumota.com/category/alm-cloud-devops/>>. Acesso em: 1 maio. 2023.

**A Bauducco obtém uma economia de custo média de 30%.** Disponível em: <<https://www.avanade.com/pt-br/clients/bauducco-fin-ops>>. Acesso em: 24 abr. 2023.

AVINTE, E.; NASCIMENTO, M. H.; DO NASCIMENTO, A. **Cloud computing: reducing costs in small and medium business.** ITEGAM-JETIA, v. 5, n. 19, p. 41-47, 6 set. 2019.

**AWS CLI - Interface de linha de comando - Amazon Web Services.** Disponível em: <<https://aws.amazon.com/pt/cli/>>.

**Cloud: Guia da computação em nuvem – ATRA Informática.** Disponível em: <<https://www.atrainformatica.com.br/2023/06/02/cloud-guia-da-computacao-em-nuvem/>>. Acesso em: 25 fev. 2024.

**Convenções de nomenclatura: Camel, Pascal, Kebab e Snake case.** Disponível em: <<https://www.alura.com.br/artigos/convencoes-nomenclatura-camel-pascal-kebab-snake-case>>.

**Elastic Compute Cloud - Amazon EC2 - AWS.** Disponível em: <[https://aws.amazon.com/pt/ec2/?did=ap\\_card&trk=ap\\_card](https://aws.amazon.com/pt/ec2/?did=ap_card&trk=ap_card)>. Acesso em: 21 jun. 2024.

FIGUEIREDO, L. B. de. **O impacto da gestão de ativos de software em uma organização**. app.uff.br, 2021.

Finops Day (@finopsday). **Importância de Tags para a Comunidade de FinOps**. Medium.com, 14 ago. 2024. <https://medium.com/tag/finops>

**FinOps permite uso eficiente da nuvem pelas empresas**. Disponível em: <<https://www.segs.com.br/info-ti/294647-finops-permite-uso-eficiente-da-nuvem-pelas-empresas>>. Acesso em: 24 abr. 2023.

**Gastos com nuvem crescem 22% no 2º trimestre**. Disponível em: <<https://www.convergenciadigital.com.br/Cloud-Computing/Gastos-com-nuvem-crescem-22%25-no-2%BA-trimestre-61622.html?UserActiveTemplate=site#:~:text=Para%20todo%20o%20ano%20de>>. Acesso em: 23 abr. 2023.

GIHR, C. M. T. **Cloud computing na construção de infraestruturas de TI**. comum.rcaap.pt, 1 fev. 2022.

**jq**. Disponível em: <<https://jqlang.github.io/jq/>>.

LEITE, B. **Sistema de apoio ao projeto e implantação de infraestrutura em nuvem**. dspace.unipampa.edu.br, 1 fev. 2023.

LUNA, A. et al. **Agile governance in information and communication technologies: shifting paradigms**. *JISTEM Journal of Information Systems and Technology Management*, v. 7, n. 2, p. 311–334, 30 ago. 2010.

MELL, P.; GRANCE, T. **The NIST definition of Cloud Computing**. National Institute of Standards and Technology, v. 53, n. 6, p. 50, 2011.

**O que é a computação em nuvem?** Disponível em: <<https://aws.amazon.com/pt/what-is-cloud-computing/>>. Acesso em: 13 jun. 2023.

**O que é CLOUD? Explicação COMPLETA para INICIANTES**. Disponível em: <<https://www.youtube.com/watch?v=uhXr9L57hOM>>. Acesso em: 24 abr. 2023.

**O que é FinOps? | IBM**. Disponível em: <<https://www.ibm.com/br-pt/topics/finops#citation1>>. Acesso em: 24 abr. 2023.

**O que é AWS? Como funciona Amazon Web Services.** Disponível em: [<https://aws.amazon.com/pt/what-is-aws/>](https://aws.amazon.com/pt/what-is-aws/).

Rabello, Guilherme. **Maturidade em gestão de projetos: o que é, como definir e exemplos de matriz de maturidade.** Siteware, Disponível em: [<https://www.siteware.com.br/blog/projetos/maturidade-em-gestao-de-projetos/>](https://www.siteware.com.br/blog/projetos/maturidade-em-gestao-de-projetos/), 22 fev. 2022.

**Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%.** Disponível em: <https://www.primevideotech.com/video-streaming/scaling-up-the-prime-video-audio-video-monitoring-service-and-reducing-costs-by-90>>. Acesso em: 11 maio. 2023.

SOUZA, S. L. L. de. **Proposta de melhoria de um processo de solicitação de mudanças em um produto de software.** Disponível em: <https://repositorio.ufpe.br/handle/123456789/47549>>. Acesso em: 24 abr. 2023.

STORMENT, J. R.; FULLER, M. **Cloud FinOps: collaborative, real-time cloud financial management.** O'Reilly, 2019.

Tridens. **Computação em nuvem em 2024 e os 10 principais fornecedores de serviços em nuvem.** 17 nov. 2023. Disponível em: <https://tridenttechnology.com/ar/>.

**Utilizando as Ferramentas da AWS para Implementar FinOps na sua Estratégia de Nuvem.** Disponível em: <https://www.selectsolucoes.com.br/2023/01/27/ferramentas-aws-finops/>>. Acesso em: 1 maio. 2023.

## **APÊNDICES**

## APÊNDICE A - COMO CRIAR UMA CONTA NA AWS E CONFIGURAR PARA EXECUTAR NO TERMINAL

O procedimento de criação de conta na AWS se dá por acessar o link da própria aws <https://aws.amazon.com/>, clicar em “Criar conta da AWS”, preencher o formulário com as informações tais como Nome, Email, Senha, Telefone, País e Método de pagamento, ler e concordar com os termos e condições, clicar em “Continuar”, ir no seu email e verificar a conta criada pela AWS clicando no link de ativação da conta.

Para usar o AWS CLI no terminal é necessário baixar a ferramenta pelo link <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html> depois segue o procedimento para fazer as configurações do ambiente de acesso a conta, para isso é necessário criar uma chave de acesso na AWS pelo perfil IAM, podendo criar após logar no console da AWS <https://aws.amazon.com/console/>, ir na ferramenta de IAM presente no menu de serviços, no painel à esquerda, clicar em “Usuários”; clicar em “Criar usuário”, adicionar o nome do usuário, clicar em “Próximo: Permissões” selecionar a opção de marcar as políticas desejadas em “Anexar políticas diretamente”, nesse caso em “Políticas de permissões” devo marcar a opção “AdministratorAccess”, pois quero ter acesso para gerenciar custos e criar novos usuários mas é uma boa opção estudar os melhores níveis de acesso para cada caso, clicar em “Próximo”; Revisar as políticas e adicionar rótulos(etiquetas) caso necessário e finalizar a criação.

Após criado o usuário precisará criar uma chave de acesso para ele, indo em “Usuários”, selecionando nosso usuário e clicando em “Criar chave de acesso”, selecionando a opção de *Command Line Interface*, marcando que compreende as quer criar uma chave de acesso e “Próximo”, se quiser, informe um rótulo, nesse caso apenas criei as chaves clicando em “Criar chaves de acesso”, quando criar ele informará uma chave de acesso e chave de acesso secreta, copie as duas chaves e com o aws cli instalado poderemos seguir configurando a conexão.

No terminal execute o comando “aws configure --profile perfil-exemplo”, nesse caso vou usar o “perfil-exemplo” como nome de usuário, o mesmo vai solicitar a chave de acesso do usuário, inserir o valor e ir para o próximo, que será a chave de



acesso secreta, o mesmo procedimento deve ser feito e será solicitado a região padrão que vai usar as ferramentas, no exemplo foi usado “*us-east-1*”, em seguida vai solicitar o formato de saída, para o exemplo foi escolhido “*Json*”.

Após isso poderemos usar o “aws login –profile perfil-exemplo” para podermos conseguir fazer o login na conta e usar os comandos sem futuros problemas. Também vamos instalar a ferramenta Jq, usando a documentação <https://jqlang.github.io/jq/download/>.

Concluído o processo de criação e configuração dos recursos usados para o presente trabalho é necessário executar os comandos apenas com “aws –profile perfil-exemplo” e o mesmo será executado normalmente.

## APÊNDICE B - COMO CRIAR UMA MÁQUINA EC2 NA AWS

Será usado a documentação <https://docs.aws.amazon.com/ec2/> como base para as orientações a seguir, como descrito, ir na console aws logado, acessar a opção de “Instâncias”, clicar em “Executar Instância”, informar um nome para ela, e escolher o tipo, para esse caso foi usado uma t2.micro, com uma Imagem da AWS mesmo.

Para esse caso, como queremos ter rótulos, precisamos clicar em “Adicionar mais *tags*” na seção “Nome e Rótulos”, informar os valores para os rótulos que serão usados, para esse caso, usamos o exemplo dos rótulos já presentes no trabalho no capítulo de Metodologia, depois é só clicar em “Executar Instância”.