

INSTITUTO FEDERAL DE SANTA CATARINA

MARIA LUIZA THEISGES

Mineração de bancos de dados distribuídos usando inteligência artificial para análise de log de nuvem privada de contêineres

São José - SC

junho/2018

**MINERAÇÃO DE BANCOS DE DADOS DISTRIBUÍDOS USANDO
INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE DE LOG DE NUVEM PRIVADA
DE CONTÊINERES**

Trabalho de conclusão de curso apresentado à Coordenação do Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Ederson Torresini

Coorientador: Clayrton Monteiro Henrique e Fernando Rodrigues Santos

São José - SC

junho/2018

Maria Luiza Theisges

Mineração de bancos de dados distribuídos usando inteligência artificial para análise de log de nuvem privada de contêineres/ Maria Luiza Theisges. – São José - SC, junho/2018-

31 p. : il. (algumas color.) ; 30 cm.

Orientador: Ederson Torresini

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC

Campus São José

Engenharia de Telecomunicações, junho/2018.

1. Inteligência artificial. 2. Banco de dados. 2. Contêiner. I. Ederson Torresini. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Mineração de bancos de dados distribuídos usando inteligência artificial para análise de log de nuvem privada de contêineres

MARIA LUIZA THEISGES

**MINERAÇÃO DE BANCOS DE DADOS DISTRIBUÍDOS USANDO
INTELIGÊNCIA ARTIFICIAL PARA ANÁLISE DE LOG DE NUVEM PRIVADA
DE CONTÊINERES**

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 25 de junho de 2018:

Ederson Torresini, Me.
Orientador
Instituto Federal de Santa Catarina

Cleber Jorge Amaral, Me.
Instituto Federal de Santa Catarina

Marcos Moecke, Dr. Eng.
Instituto Federal de Santa Catarina

*“Que os vossos esforços desafiem as impossibilidades,
lembrai-vos de que as grandes coisas do homem
foram conquistadas do que parecia impossível”.*
(Charles Chaplin)

RESUMO

A computação em nuvem propiciou que os aplicativos fossem utilizados em nuvem por meio de uma conexão de internet e trouxe o aumento da escalabilidade para diversos tipos de demanda. A Coordenadoria de Tecnologia da Informação e Comunicação (CTIC) do Instituto Federal de Santa Catarina câmpus São José (IFSC/SJ) vem trabalhando de forma a disponibilizar todas as aplicações oferecidas pelo câmpus em nuvem de contêineres, visando diminuir a redundância destes serviços e aumentar a eficiência e escalabilidade do sistema, na forma de microsserviços. Através do uso de aprendizado de máquina, pretende-se, a partir da avaliação dos *logs* armazenados no banco de dados, desenvolver uma ferramenta que auxilie na análise e na obtenção de indicadores de eficiência da nuvem do Instituto.

Palavras-chave: Inteligência artificial. Banco de dados. Contêiner.

LISTA DE ABREVIATURAS E SIGLAS

API <i>Application Programming Interface</i>	21
CPD Centro de Processamento de Dados	15
CTIC Coordenadoria de Tecnologia da Informação e Comunicação	14
DCaaS <i>Data Center as a Service</i>	16
IaaS <i>Infrastructure as a Service</i>	18
IFSC/SJ Instituto Federal de Santa Catarina câmpus São José	13
IA Inteligência Artificial	14
IP <i>Internet Protocol</i>	21
PaaS <i>Platform as a Service</i>	18
SaaS <i>Software as a Service</i>	18
SDC <i>Software Defined Computing</i>	16
SDDC <i>Software Defined Data Center</i>	16
SDN <i>Software Defined Network</i>	16
SDS <i>Software Defined Storage</i>	16
TI <i>Tecnologia da Informação</i>	14
VM <i>Virtual Machine</i>	13

SUMÁRIO

1	INTRODUÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Centro de Processamento de Dados Definido por <i>Software</i>	15
2.2	Rede Definida por <i>Software</i>	16
2.3	Computação em Nuvem	17
2.3.1	Modelos de Serviço	18
2.3.2	Modelos de Implantação	19
2.4	Contêiner	20
2.4.1	O Kubernetes	20
2.4.2	Coleta de <i>logs</i>	22
2.5	Inteligência Artificial	23
2.5.1	Aprendizado de Máquina	24
3	PROPOSTA	27
3.1	Justificativa	27
3.2	Objetivos	27
3.2.1	Objetivo Geral	27
3.2.2	Objetivos Específicos	27
3.3	Metodologia	27
	REFERÊNCIAS	29

1 INTRODUÇÃO

Desde o primeiro computador até os sistemas atuais, é possível perceber o avanço obtido nas tecnologias utilizadas pelas empresas, instituições e pelo público em geral. Cada vez mais, o *software*, antes executado de forma local, está sendo migrado para a nuvem, e nesta são processados e armazenados grande quantidade de dados de forma remota e de modo transparente ao usuário.

A virtualização foi uma tecnologia que impulsionou a disponibilização dos recursos através da nuvem e o desenvolvimento da computação em nuvem (*cloud computing*). Ao virtualizar os recursos de *hardware*, vários ambientes podem ser criados a partir de um único *hardware* físico (Red Hat, 2018). Isto é, as chamadas máquinas virtuais (*Virtual Machine (VM)*) são criadas através de *software* e abstraem um ambiente de uma máquina física (CARISSIMI, 2009, p. 40). Segundo Microsoft Azure (2018b):

A virtualização cria um ambiente de computação simulado ou virtual em vez de um ambiente físico. A virtualização costuma incluir versões de *hardware*, sistemas operacionais, dispositivos de armazenamento e outros, todas geradas por computador. Isso permite que as organizações particionem um único computador físico ou servidor em diversas máquinas virtuais.

As vantagens obtidas ao se criar diversos recursos de um computador ou servidor em diversas máquinas virtuais são: melhorar a escalabilidade, gastar menos energia e reduzir os custos de infraestrutura e manutenção (Microsoft Azure, 2018b). Além disso, é possível fazer a migração de um ambiente computacional para outro sem que haja conflitos de dependência de *hardware*, por exemplo.

Com a evolução da computação em nuvem foi necessário desenvolver uma arquitetura de *software* para propiciar a criação de aplicações baseadas em microsserviços, permitindo o aumento da escalabilidade destas para atender aos diversos tipos de demanda existentes (KUSZKA, 2017). Com o crescimento de aplicações executando em nuvem foram surgindo novas tecnologias mais fáceis de serem criadas e mantidas como, por exemplo, os contêineres. Os contêineres podem ser considerados um novo tipo de virtualização e são mais versáteis que as VMs (MANDIC, 2018). Um contêiner disponibiliza os recursos básicos necessários para o funcionamento de uma aplicação, sendo esta executada em um ambiente isolado. Com a abstração do *hardware* e do sistema operacional, e com o auxílio do microsserviço, cria-se a portabilidade do contêiner a qual possibilita a execução deste em uma máquina física ou virtual, e em nuvem privada ou pública, além de prover o desenvolvimento e infraestrutura mais ágeis (KUSZKA, 2017). Isto é possível uma vez que, em contêineres, as aplicações geralmente não fazem uso de *hardware* específico, podendo ser facilmente portadas.

O Instituto Federal de Santa Catarina câmpus São José (IFSC/SJ) é um exemplo que atende a diversas pessoas diariamente, entre professores, alunos e funcionários, oferecendo a cada um destes públicos aplicações locais, como *Matlab* e *Quartus*, e também aplicações *Web*, como blog e *Wiki* (REIS, 2017b; FILHO, 2012). Porém, existem serviços redundantes, ocasionando pouca eficiência no uso dos recursos e má alocação dos mesmos, que são divididos em diversas máquinas virtuais nas quais nenhum recurso é compartilhado. Com isso, acontecem gargalos de recurso na rede. Reis (2017b, p. 31) afirma que “Para que esses serviços funcionem em nuvem com qualidade de serviço, é preciso analisar a infraestrutura do IFSC em seus três componentes básicos: rede, processamento e armazenamento de dados.”

No IFSC/SJ ainda existe um problema de recurso limitado no que se refere à infraestrutura física, principalmente em termos de processamento e armazenamento, o que não permite entregar ao usuário um sistema que permaneça sempre disponível quando requisitado por qualquer pessoa da Instituição.

Pode-se citar como por exemplo o Matlab, que é um *software* de desenvolvimento muito usado por alunos e professores, o qual precisa ser ofertado na nuvem, permitindo que ele possa ser usado de qualquer lugar, apenas requisitando o *login* e a senha.

Estas aplicações utilizadas diariamente pelo IFSC/SJ muitas vezes são entregues aos usuários através de uma máquina virtual, onde cada uma executa uma instância de um sistema operacional de forma isolada. Porém, isto se torna complexo de manter e operar, pois o modelo atual do câmpus contempla uma máquina virtual de tamanho fixo para atender todos os alunos. A proposta da Coordenadoria de Tecnologia da Informação e Comunicação (CTIC), é de criar contêineres por demanda para atender, de forma elástica, os acessos simultâneos dos usuários. Com isto, estes serviços podem ser entregues através de aplicações *Web*, de maneira transparente para o usuário que não necessita conhecer onde estes estão sendo executados. Assim, através de gerenciadores específicos de aplicações *Web*, a garantia de manter todos estes serviços sempre disponíveis será muito maior.

Com a adoção do modelo de serviço SaaS (do inglês *Software as a Service*) pelo IFSC/SJ, poderá ser entregue ao usuário uma solução *Web*, capaz de fornecer ao mesmo um ambiente de desenvolvimento com as aplicações necessárias, sem a preocupação com restrição de *hardware* e *software* existentes no câmpus. Com isso, é possível atingir alta disponibilidade usando uma nuvem de contêineres para que seja respondido de forma rápida e eficiente os picos de demanda ou problemas existentes na infraestrutura como queda da rede, entre outros. Porém, uma dificuldade encontrada pela CTIC no fornecimento e melhoria desses serviços está justamente na análise e tomada de decisão frente aos *logs* gerados pela infraestrutura.

Muitas das decisões tomadas hoje em dia estão relacionadas a percepção e experiência dos profissionais da área de *Tecnologia da Informação (TI)*. Mas, com quando se trabalha com *Big Data*, para extração de informações e o reconhecimento de padrões se faz necessário o uso de técnicas de Inteligência Artificial (IA) para mineração de dados. Dessa forma, este trabalho se propõe a utilizar aprendizado de máquina para análise de *log* da nuvem privada do IFSC/SJ.

2 FUNDAMENTAÇÃO TEÓRICA

Quando o homem quis exceder a capacidade da aritmética prática tradicional, começaram a surgir as primeiras máquinas de somar e as caixas registradoras. Desde então, a evolução da computação, realizada através de máquinas, cresceu e na Segunda Guerra Mundial já começaram a surgir os primeiros computadores eletrônicos. Inicialmente eram máquinas extremamente grandes e ocupavam salas inteiras, mas já eram capazes de processar todo tipo de informação (CAMPBELL-KELLY, s.d).

2.1 Centro de Processamento de Dados Definido por *Software*

Com o passar dos anos, a evolução tecnológica e a difusão dos computadores na sociedade foi se tornando cada vez maior, estes começaram a ser adotados tanto por empresas quanto pelas pessoas. Campbell-Kelly (s.d) afirma que:

Talvez o desenvolvimento mais notável tenha sido que o computador – originalmente projetado para cálculos matemáticos – acabou se tornando infinitamente adaptável a diversas aplicações, do processamento de dados de empresas à computação pessoal e à construção de uma rede global de informação.

Nas empresas, estes grandes computadores ocupavam amplas salas chamadas de Centro de Processamento de Dados (CPD), e nelas eram feitos todo o processamento de dados necessário de forma centralizada. Porém, estes computadores eram complexos, de difícil manutenção e consumiam muita energia. Além disso, para fazer a ligação destes sistemas era necessário conectar muitos cabos entre os componentes de *hardware* e também necessitavam de métodos específicos para organizar estes equipamentos, como armários (ou *racks*, do inglês) e piso elevado. No CPD havia um computador de grande porte (o *mainframe*) que era responsável por processar um grande volume de informações nestes primeiros centros de processamento de dados (GARCIA, 2012).

Um CPD opera de maneira centralizada que, segundo Damasceno (2015, p. 32), contém em sua infraestrutura “componentes de hardware, como computadores, sistemas de armazenamento, sistemas energéticos; e componentes de software, como aplicações, sistemas operacionais e software de gerenciamento.”. Ainda, essa infraestrutura é composta por “[...] sistemas tecnológicos como: controle de acesso físico, combate a incêndios, refrigeração de precisão, geradores de energia e UPS (*no-breaks*).” (DAMASCENO, 2015, p. 32). O intuito do *Data Center* é o de armazenar e processar grandes volumes de informações, através de diversos servidores e banco de dados, sendo parte importante de empresas e instituições (Canal Westcon, 2016).

Neste modelo de CPD ainda não há total abstração de *hardware* e *software*. Uma aplicação ainda é executada em máquinas com características específicas tanto físicas (*hardware*) quanto de sistema (*software*), gerando dependências entre a aplicação e o ambiente na qual ela é executada. Isto se torna um problema maior quando é preciso fazer a migração da aplicação de uma máquina para outra, pois nem sempre se encontram as mesmas características em ambas as máquinas. Mas, com o início da virtualização este problema ganha um fim (ZUCCHI; AMÂNCIA, 2013).

A virtualização pode ser definida como uma técnica que permite a abstração dos recursos de *hardware*, sistema operacional, sistemas de armazenamento e de rede, propiciando, com isso, “uma versão virtual de vários tipos de recursos computacionais.” (REIS, 2017a, p. 8). Através dela é possível entregar

um CPD como um serviço (do inglês *Data Center as a Service (DCaaS)*), ou também conhecido como CPD Definido por *Software (Software Defined Data Center (SDDC))*.

O SDDC é um CPD onde a infraestrutura é virtualizada. Isto é, toda parte de recursos computacionais como armazenamento, rede e segurança são virtualizados e entregues como um serviço. Com isso, é possível ampliar os níveis de automação e flexibilidade dos serviços através da adoção dos mesmos disponibilizados em nuvem (WEBSTER, s.d.). Segundo Ferreira (2014) este modelo de *Data Center* virtualizado cresce “[...] principalmente pela possibilidade de armazenamento de dados em uma nuvem que pode crescer elasticamente para atender às demandas das empresas.”, porém o autor ressalta que há problemas com aplicações mais antigas e que sejam muito customizadas, isso se dá através da incompatibilidade existente entre o *hardware* e *software* atuais, gerando uma alta complexabilidade no que se refere a integração desses ambientes.

Já existem no mercado empresas que vendem soluções de SDDC, as mais conhecidas são a VMware e a Microsoft. E, de solução *Open Source*, dentre as mais conhecidas, destaca-se o Openstack (WEBSTER, s.d.).

Como já mencionado, hoje em dia a maioria dos componentes de *hardware* vem sendo virtualizados, principalmente os componentes existentes na arquitetura do SDDC. É possível ter a parte de armazenamento (*Software Defined Storage (SDS)*), processamento (*Software Defined Computing (SDC)*) e de rede (*Software Defined Network (SDN)*) virtualizados.

2.2 Rede Definida por Software

A rede é um componente da arquitetura do SDDC que pode ser virtualizada. A virtualização da rede consiste em abstrair alguns recursos e também possibilitar que os componentes de uma rede física possam compartilhar sua capacidade realizando várias funções. Sendo assim, segundo Costa (2013, p. 16)

A virtualização de redes passou a permitir que os componentes de uma rede física compartilhassem sua capacidade de maneira que realize simultaneamente múltiplas funções, estabelecendo infraestruturas lógicas distintas e mutuamente isoladas provendo um método para que múltiplas arquiteturas de rede distintas compartilhem o mesmo substrato físico.

Dessa forma, as redes virtuais permanecem isoladas umas das outras, possuindo cada uma a sua pilha de protocolos e gerenciamento individual (COSTA, 2013).

Com esta nova característica das redes, surgiram propostas de redes programadas sob demanda e que são flexíveis, permitindo um controle e expansão mais fácil. As Redes Definidas por *Software (SDN)* podem ser desenvolvidas de maneira simples e desprendida da arquitetura dos equipamentos de rede, onde nas SDNs há separação do plano de controle em relação ao plano de dados. Os equipamentos de rede existentes no mercado possuem uma arquitetura interna dividida em duas partes principais: o plano de controle e o plano de dados. O plano de controle é o encarregado pelos protocolos e também pelas tomadas de decisão que resultam nas tabelas de roteamento. Já o plano de dados, é responsável pela comutação e repasse dos pacotes de rede (COSTA, 2013).

A SDN, segundo Costa (2013), tem como princípio básico a capacidade de programar os equipamentos contidos em uma rede de computadores, sendo esta programação uma maneira simples de manipular os pacotes em um fluxo, e estes são oferecidos pela interface de programação. Ele afirma que

[...] as redes SDN tem a capacidade de controlar o plano de encaminhamento de pacotes por meio de uma interface bem definida. Sem dúvida uma das interfaces mais conhecidas deste paradigma, desde o início é o *OpenFlow*. O principal objetivo do

OpenFlow é permitir que se utilize equipamentos de rede comerciais para pesquisas de novos protocolos de rede em paralelo ao tráfego real de produção da rede. (COSTA, 2013, p.26).

Basicamente, a SDN é composta por três camadas distintas, sendo elas: as camadas de aplicação, de controle e de infraestrutura. A camada de aplicação contém as aplicações em nuvem e de gerenciamento, e é desta que surgem as demandas de rede para a camada de controle. A camada de controle analisa o comportamento do encaminhamento de rede e recebe as demandas de serviço vindas da camada de aplicação. Enquanto a camada de infraestrutura são os elementos da rede que encaminham os pacotes de dados (JUNIOR, 2015; CIENA, s.d.).

2.3 Computação em Nuvem

Nuvem é um termo utilizado para representar a Internet e sua infraestrutura de comunicação, de modo a ocultar a complexidade embutida na infraestrutura que, neste caso, não necessariamente precisa ser conhecida. Segundo Somavila (2015, p. 18)

A nuvem é uma representação para a internet ou infraestrutura de comunicação entre os componentes arquiteturais, baseada em uma abstração que oculta à complexidade de infraestrutura. Cada parte desta infraestrutura é provida como um serviço e estes serviços são normalmente alocados em *datacenters*, utilizando *hardware* compartilhado para computação e armazenamento, com isto, não é mais necessário ocupar espaço de armazenamento do computador local, os dados estarão sendo acessados pela internet.

Já a computação em nuvem (do inglês *cloud computing*) pode ser definida como a capacidade de acessar arquivos e aplicações de qualquer dispositivo com o uso da Internet. Diferente da virtualização é basicamente, dentre outras características, o compartilhamento de recursos de TI através de infraestruturas virtuais, a computação em nuvem refere-se a entrega de soluções como um serviço através da nuvem (Canal Westcon, 2017). A computação em nuvem também traz a característica de manter os serviços disponíveis em Nuvem para que possam ser utilizados quando necessário e pelo tempo desejado (DAMASCENO, 2015).

As características importantes da computação em nuvem são: autoatendimento sob demanda, amplo acesso à rede, grupo de recursos, elasticidade rápida e serviço sob demanda. O autoatendimento sob demanda refere-se ao acesso de recursos da nuvem pelo usuário que, de acordo com a sua necessidade e demanda, a quantidade necessária de recursos será disponibilizada. O amplo acesso à rede refere-se a disponibilidade que deve ser dada ao usuário, isto é, a nuvem deve ter a capacidade de receber acessos simultâneos advindos de locais diferentes. Já o grupo de recursos trata de disponibilizar os recursos computacionais da nuvem de forma transparente para o usuário. A elasticidade rápida, uma das principais características da computação em nuvem, é a capacidade de a nuvem disponibilizar mais recursos ao usuário conforme for sendo requisitado pelo mesmo ou por sua aplicação. E, por fim, como a computação em nuvem baseia-se no modelo *pay-for-usage* (pague pelo uso) então, como forma de cobrar o que foi usado, o serviço sob demanda é a forma pela qual são contabilizados os recursos e serviços utilizados pelo usuário (DAMASCENO, 2015).

No que diz respeito a arquitetura da computação em nuvem pode-se dizer que esta é dividida em quatro partes principais, como mostra a Figura 1.



Figura 1 – Arquitetura da Computação em Nuvem (Adaptado de ZHANG; CHENG; BOUTABA (2010)).

A camada de *Hardware* contém os equipamentos de rede que compõem a nuvem, como servidores, roteadores, *switches*, sistemas de resfriamento e energia. Esta camada, geralmente, é implementada em um *Data Center* e nela são feitos o gerenciamento de *hardware* e de tráfego, o gerenciamento de energia e resfriamento, e também o controle de tolerância a falhas (ZHANG; CHENG; BOUTABA, 2010; BRAGA; SILVA; BARROS, 2012).

Na camada de Infraestrutura é onde são criados os recursos computacionais necessários pela computação em nuvem, realizando o particionamento dos recursos físicos através da virtualização. ZHANG; CHENG; BOUTABA afirmam que esta camada é importante para a *Cloud Computing* uma vez que, através da virtualização, é possível alocar dinamicamente os recursos da nuvem.

A camada de Plataformas (*Platforms*) contém os sistemas operacionais e também os *frameworks* das aplicações. Esta camada tem o intuito de simplificar a implementação das aplicações em Máquinas Virtuais (VM) (BRAGA; SILVA; BARROS, 2012).

E a última camada é a de aplicação, sendo a mais alta da hierarquia, contém as aplicações da *Cloud Computing*, estas que irão permitir, através da escalabilidade automática, alcançar melhor desempenho, disponibilidade e menor custo operacional para executar aplicações ao usuário final (ZHANG; CHENG; BOUTABA, 2010).

2.3.1 Modelos de Serviço

A computação em nuvem opera com um modelo de serviço sob demanda, ou seja, cada recurso da nuvem fica disponível como um serviço e este é disponibilizado sob demanda para o usuário. Braga, Silva e Barros (2012) afirmam que, conceitualmente, cada uma das camadas da arquitetura da *Cloud Computing* pode ser implementada como um serviço para a camada de cima. E, estas podem ser agrupadas em três categorias sendo a de *Software* como um Serviço (*Software as a Service (SaaS)*), a Plataforma como um Serviço (*Platform as a Service (PaaS)*) e a Infraestrutura como um Serviço (*Infrastructure as a Service (IaaS)*) (BRAGA; SILVA; BARROS, 2012).

O *IaaS* é a camada que fornece a infraestrutura necessária para as camadas de cima, sendo elas: a *PaaS* e a *SaaS*. Nesta camada os serviços, como os de servidores e roteadores, são reunidos e entregues, por meio de uma interface disponível em nuvem, ao usuário ou à aplicação (SOMAVILA, 2015). Damasceno (2015, p. 52) diz que

Em geral, serviços neste modelo proveem uma Interface de Programação de Aplicativos (do inglês *Application Programming Interface - API*) de gerenciamento para a utilização

de um conjunto de recursos físicos de forma que o usuário tenha acesso a funcionalidades como configurações automatizadas do sistema operacional, escalabilidade sob demanda e armazenamento.

O **PaaS** oferece um ambiente com sistema operacional, linguagens de programação, banco de dados, servidores e outras ferramentas que permitem o desenvolvimento de aplicações para a nuvem. O usuário tem a possibilidade de implementar e testar aplicações em nuvem através do ambiente de desenvolvimento que é disponibilizado nesta camada. Com isso, são eliminadas a complexidade e os gastos obtidos com a compra de *softwares* e elementos de infraestrutura, uma vez que estes serviços são pagos conforme foi a demanda de utilização na nuvem (SOMAVILA, 2015; DAMASCENO, 2015; Microsoft Azure, 2018a).

A **SaaS** é a camada onde o usuário utiliza as aplicações disponíveis na nuvem. Nesta camada, os usuários têm acesso aos serviços prestados através de uma interface *Web* e *online* (SOMAVILA, 2015; DAMASCENO, 2015). Segundo Somavila (2015, p. 29)

[...] os consumidores estão, cada vez mais, mudando de programas instalados localmente para serviços on-line, que oferecem até mais funcionalidades. Aplicações tradicionais de *desktops*, tais como processamento de texto e planilhas eletrônicas, já podem ser acessados como serviços na *web*.

2.3.2 Modelos de Implantação

A nuvem pode ser implementada de diversas formas, baseada na necessidade de cada empresa ou instituição. Geralmente, são avaliados os quesitos de redução de custos, segurança, elasticidade, entre outros (ZHANG; CHENG; BOUTABA, 2010). Há quatro formas de se implementar a nuvem, que podem ser: privadas, públicas, comunitária e híbrida.

A nuvem privada oferece serviço propriamente para as organizações. Neste caso, ela pode ser interna ou externa. No primeiro caso, tem-se a nuvem implementada e gerenciada pelo departamento de **TI** da própria organização, sendo o caso mais adequado quando se deseja obter um maior controle sobre a infraestrutura, configurações de recursos, dados, segurança e outras características. No segundo caso, a nuvem privada é hospedada de maneira externa e, desse modo, o provedor deste serviço irá gerenciar a estrutura e prover um ambiente único para a organização, e a conexão nesta será feita preferencialmente através de uma rede segura (DAMASCENO, 2015; ZHANG; CHENG; BOUTABA, 2010).

A nuvem pública é disponibilizada para o público em geral e, normalmente, os serviços disponíveis nela não são cobrados ou são cobrados de maneira reduzida se comparados ao primeiro modelo (FELIPE et al., 2012). Segundo Felipe et al. (2012) “[...] neste caso o usuário ou organização que venha a buscar seus serviços não saberá onde seus dados estão sendo armazenados e/ou manipulados de maneira que não poderão intervir na maneira como é feita a gestão da nuvem nem quais as práticas adotadas pela a empresa contratada quanto à segurança da informação.”

Na nuvem comunitária integrantes de diversas organizações se reúnem para criar uma nuvem comum a todos e que atenda às exigências que são comuns entre os mesmos. Neste caso os serviços são compartilhados para todos os proprietários da nuvem comunitária e, dentre os modelos citados, este é o que tem nível de segurança intermediário (FELIPE et al., 2012).

A nuvem híbrida acaba por ser uma junção dos modelos de Nuvem Pública e Privada, citadas acima. Neste caso é possível ampliar os recursos de uma nuvem privada através dos recursos de uma nuvem pública, a vantagem obtida com isso é a de garantir os níveis de serviços quando houver flutuações rápidas na utilização dos recursos (SOMAVILA, 2015). Somavila (2015, p. 27) afirma que “Utilizando este modelo de nuvem, as organizações podem executar serviços que não são necessariamente exigidos pela

empresa, ou seja, que podem ser repassados para terceiros, hospedando eles em uma nuvem pública e mantendo os aplicativos principais e dados sensíveis da empresa dentro da própria organização em uma nuvem privada [...]”.

Dentre os modelos de nuvem existentes e citados anteriormente, neste trabalho, será utilizado o modelo de nuvem Privada.

2.4 Contêiner

A transformação que vem acontecendo no meio digital está mudando a forma de consumo dos recursos computacionais. Cada vez mais, o setor TI precisa ter uma abordagem mais ágil, adequando sua infraestrutura, melhorando os sistemas e serviços utilizados neste ambiente. O porquê de se depender de uma infraestrutura mais ágil se justifica com o uso de contêineres, que agilizam a entrega de recursos (ESPINHA, 2018). Um contêiner é baseado em técnicas de virtualização em nível de sistemas operacionais e pode ser definido como um ambiente isolado que possibilita a execução de uma ou mais aplicações dentro desse ambiente restrito (MONTEIRO; XAVIER; VALENTE, 2017).

No Linux, o uso de contêineres começou com o projeto VServer, sendo este desenvolvido por Jacques Gélinas no ano de 2001. Com isto, pretendia-se separar o espaço de usuário em unidades singulares, como forma de representar servidores reais para os processos existentes em cada uma destas unidades. Em 2006, Paul B. Manage, engenheiro da Google, sugeriu mudanças para que fossem elevados os níveis de utilização dos contêineres, o que resultou, posteriormente, na implementação do *cgroups* no kernel do Linux. O *cgroups* possibilita a junção de processos em grupos, e disponibiliza a cada um destes uma parte da memória, da CPU e de recursos de Entrada/Saída (E/S). Assim, é possível evitar a monopolização destes pelos contêineres. Já no ano de 2008, foi adicionado ao modelo de contêineres o *namespace*, recurso este que permite ao processo ter um conjunto de usuários, com possibilidade de privilégios de administrador¹, e que são restritos a cada contêiner (ALVES, 2017).

É possível criar diversos contêineres que executam tarefas específicas. E, como forma de facilitar o gerenciamento de um conjunto de contêineres, existem as plataformas de orquestração. Alves (2017, p. 30), afirma que:

[...] uma plataforma de orquestração de *containers* gerencia e coordena, de maneira simples e eficiente, uma rede de recursos computacionais independentes, que executam uma plataforma de execução de *container*.

Dentre as plataformas existentes no mercado, destaca-se a da Google, o Kubernetes. Este é de código aberto e, segundo Alves (2017, p. 30) permite a “implantação, o redimensionamento (*scaling*) e o gerenciamento de aplicações em *containers*”.

2.4.1 O Kubernetes

Conhecido também como *k8s*², o Kubernetes foi desenvolvido pela Google, disponibilizado no ano de 2014, como sendo uma ferramenta que permite gerenciar um conjunto de contêineres (BURNS et al., 2016). O Kubernetes é formado por um conjunto de processos que possuem um controle independente, no qual conduzem o fluxo de trabalho do estado atual para o estado desejado. Isso não é feito através de um controle centralizado, o que resulta em um sistema mais robusto, poderoso e expansível (Kubernetes, 2018).

¹ *Linux capabilities*.

² k + 8 caracteres + s.

Quando se trabalha com o Kubernetes, são usadas *Application Programming Interface (API)*s que irão descrever o estado desejado do *cluster*, como por exemplo quais aplicações e cargas de trabalho que se deseja executar, quais as imagens de contêiner utilizadas, número de réplicas, entre outros. Para isso, o usuário define o estado desejado, que se dá através da criação de objetos usando a API do Kubernetes, e quando o estado do *cluster* se for o desejado, os objetos do *Kubernetes Control Plane* irão trabalhar de forma que o *cluster* permaneça sempre nesse estado. O Kubernetes faz isso através da realização de uma série de tarefas, executadas automaticamente, como por exemplo iniciar ou reiniciar um contêiner e outras tarefas (Kubernetes, 2018). Caso algum processo em execução no Kubernetes apresentar algum tipo de problema, ele o “exclui” e cria novamente outro processo igual e continua a executá-lo (NEWMAN, 2015). Como forma de gerenciar todo esse conjunto de funções, o *Kubernetes Control Plane* contém em uma coleção de processos que ficam em execução no *cluster*, sendo:

- *Kubernetes Master*: é uma coleção de três processos que executam no(s) nó(s) mestre(s). Esses processos são:
 - *kube-apiserver*: tem a função de validar e configurar os dados para *api-objects*, onde incluem os PODs, Serviços, Replication Controllers e outros, e também analisa a demanda que há no sistema. É este processo que faz a interface com o mundo externo para administração da nuvem.
 - *kube-controller-manager*: este processo tem a função de garantir que toda a demanda existente no sistema seja atendida. O *controller-manager* fará o gerenciamento como forma de saber se todas as aplicações estão executando da maneira correta.
 - *kube-scheduler*: a partir de uma solicitação do usuário por determinada aplicação, é o *scheduler* que encontrará os nós físicos, como forma de localizar os dados requeridos pelo usuário. O *scheduler* também irá tentar equalizar a demanda com base nos recursos disponíveis.
- Cada nó que não é mestre executa em seu *cluster* um processo:
 - *kube-proxy*: *proxy* de rede que refletirá os serviços de rede em cada um dos nós. É o *kube-proxy* que irá atender uma demanda, interna ou externa, e a encaminhará para o Pod certo. Em resumo, o *kube-proxy* executa duas funções importantes: ele quem irá falar com o *kube-apiserver* e fará o encaminhamento das requisições no sistema.

O *k8s* é composto por uma série de objetos, *Kubernetes Objects*, que são abstrações cuja função é a de representar o estado do sistema. É possível obter informações sobre o que está sendo feito no *cluster* como aplicativos e cargas de trabalho distribuídos, recursos de rede e disco, entre outras. Alguns dos objetos básicos do Kubernetes são os Pods, Serviço, Volume e o *Namespaces*.

Os Pods são blocos de construção básicos do Kubernetes, a menor e mais simples unidade dentre os objetos. O Pod caracteriza um processo em execução no *cluster*. Ele encapsula um ou mais contêineres de aplicação, recursos de armazenamento, um *Internet Protocol (IP)* de rede único e opções que dizem como o contêiner deve ser executado.

O *Service* deixa visível o Pod em rede criando uma ponte virtual entre o usuário e a aplicação executada no Pod. Em outras palavras, é ele que se encarrega de exportar uma porta de um contêiner permitindo, assim, que a aplicação sendo executada no Pod fique “visível” para o usuário.

No Kubernetes, o Volume representa a parte persistente de dados. Os arquivos no disco dos contêineres não são persistentes, ocasionando um problema. Quando um contêiner apresenta uma falha, o *kube-proxy* o reinicia, mas os arquivos são perdidos. É também ao se executar contêineres juntos em um

Pod estes arquivos precisam ser compartilhados entre eles. O Volume vem como forma de resolver esses dois problemas.

O Kubernetes pode suportar *clusters* virtuais no mesmo *cluster* físico. Os *clusters* virtuais são chamados de *Namespaces*. Estes são utilizados em ambientes com muitos usuários que se espalham em diversas equipes ou em projetos, e fornecem um escopo para nomes. Neste caso, acontece a criação de áreas isoladas dentro do *cluster*.

O Kubernetes também contém abstrações de nível superior que são os Controladores. Os Controladores tem a função de desenvolver os objetos básicos e, além disso, fornecem recursos e funcionalidades adicionais. Eles são: *ReplicaSet*, *Deployment*, *StatefulSet*, *DaemonSet* e *Job*.

O *ReplicaSet* é controlador de replicação, ou seja, é ele que irá garantir a quantidade de réplicas de aplicação que foram especificadas pelo usuário.

O *Deployment* fornece atualizações para Pods . É o objeto que cuida do ciclo de vida de cada aplicação, no que se refere a versão das aplicações e as bibliotecas utilizadas. O *Deployment* que criará o objeto que controla a quantidade de réplicas de Pod, o *ReplicaSet*.

O *StatefulSet* é usado para gerenciamento de aplicações com estado. Segundo o *site* ele, assim como um *Deployment*, “gerencia a implantação e a escala de um conjunto de Pods, e fornece garantias sobre o pedido e a originalidade desses Pods.”³ (Kubernetes, 2018). Isto é, caso um Pod apresente algum tipo de problema, o *StatefulSet* tentará recuperá-lo, fazendo com que ele continue executando conforme estava anteriormente.

O *DaemonSet* irá garantir que todos os nós, ou alguns deles, executem uma cópia de um Pod na máquina física. E, por fim, o *Job* é um Pod que tem “tempo de vida útil” curta. O objetivo é o de atender um tarefa específica e, depois disso, ele deixa de existir.

2.4.2 Coleta de *logs*

Com o crescimento de aplicações sendo executadas em contêineres é preciso monitorar o que acontece em cada instância em execução. Sendo assim, as ferramentas de gerenciamento de contêineres, como o Kubernetes, necessitam de um registro do que acontece e dos erros existentes em cada um dos contêineres (weaveworks, 2018). Para isto, existem os registros em *logs* que fornecem uma visão holística do sistema. Segundo Microsoft Azure (2017), os *logs* “são registros de eventos que ocorrem durante a execução do aplicativo”. Dessa forma, estes são úteis para que, através deles, sejam feitas análises mais detalhadas da causa de um problema, por exemplo. Gerenciar este grande volume de *logs* gerados não é uma tarefa simples. Entretanto, hoje já existem ferramentas que monitoram e administram este conjunto de informações, como é o caso do Prometheus.

O Prometheus foi desenvolvido pela empresa SoundCloud como uma ferramenta para monitoramento e alerta de sistemas, e faz isto através da coleta de métricas dos sistemas monitorados (Prometheus, 2018). As vantagens de utilizá-lo vão desde sua integração com o Kubernetes, o que permite o monitoramento e o agendamento de serviços dinamicamente, até métricas baseadas em *pull*, que possibilitam aos serviços não saberem onde está sendo executado o sistema de monitoramento (weaveworks, 2018).

Os dados do Prometheus são como séries temporais, ou seja, são fluxos de valores que possuem data e hora e pertencem a uma mesma métrica e conjunto de dimensões rotulados. Estas séries são identificadas pelo nome da métrica e um conjunto de pares de chave-valor, conhecidos como rótulos (Prometheus, 2018).

³ “StatefulSet manages Pods that are based on an identical container spec.” (Kubernetes, 2018)

Segundo o site oficial, o Prometheus trabalha bem com séries temporais que sejam exclusivamente numéricas. Ele pode ser usado tanto para o monitoramento centralizado quanto ao de arquiteturas dinâmicas orientadas a serviço. Ainda segundo o site, o ponto forte dele no mundo do microsserviço é a coleta e a consulta de dados multidimensionais (Prometheus, 2018).

O Prometheus pode coletar um volume de *logs* extremamente grande dependendo do tamanho do sistema no qual ele monitora. Entretanto, analisar esta quantidade de dados não é uma tarefa simples. Isto implicaria em um grande conhecimento sobre o sistema em questão, bem como um longo período de tempo para estudar quais as possíveis falhas do sistema, por exemplo. Para isto, podem ser usadas técnicas de IA onde os dados podem ser minerados e o resultado obtido pode ser avaliado com maior facilidade.

2.5 Inteligência Artificial

Fazer com que a máquina não só processasse dados mas que também tivesse inteligência e começasse a tomar as próprias decisões, já é algo que vem sendo estudado há muito tempo. Na década de 1950, o matemático Alan Turing propôs o Teste de Turing. Um humano, que era o interrogador, fazia algumas perguntas escritas ao computador. Se ao fim do teste o humano não soubesse diferenciar se as respostas vinham de um computador ou de outra pessoa, dizia-se que o computador passou no teste e que, na teoria de Turing, o computador podia de alguma forma, pensar (RUSSELL; NORVIG, 2009, p. 2).

Para que este teste fosse possível, o computador deveria ter capacidade de: processamento de linguagem natural, representação de conhecimento, raciocínio automatizado e aprendizado de máquina. O processamento de linguagem natural serve para que o computador possa conseguir se comunicar. A representação de conhecimento serve para que o computador possa conseguir armazenar o que aprende. O raciocínio automatizado para que possa, com as informações já armazenadas, responder novas perguntas e tirar conclusões. E o aprendizado de máquina permite que o computador se adapte a novas eventualidades (RUSSELL; NORVIG, 2009, p. 2).

O teste acabou passando por modificações, adicionando a máquina mais algumas capacidades. Da primeira vez, o interrogador não tinha nenhum contato com a máquina mas da segunda vez isso já se tornou possível. Russell e Norvig (2009) comentam sobre o Teste de Turing total, dessa vez no teste há um sinal de vídeo que permite ao interrogador a possibilidade de testar as habilidades de percepção do indivíduo e também é possível repassar objetos através de uma “janelinha”, segundo os autores. Para isto, o computador precisa, além das capacidades já comentadas, ter visão computacional como forma de perceber os objetos, e robótica para manusear os mesmos e se movimentar. Estas habilidades adicionadas à máquina fazem parte de uma área de estudo maior, denominado de Inteligência Artificial (IA).

A IA é um ramo da ciência da computação e “Situa-se no cruzamento da cognição, das neurociências e da informática, buscando compreender o pensamento e o comportamento humanos, a fim de reproduzi-los artificialmente.” (PESSIS-PASTERNAK, 1992, p. 193). Busca-se, com ela, solucionar problemas cognitivos que são relacionados à inteligência humana como o aprendizado, a solução de problemas e o reconhecimento de padrões e, com isto, fazer com que o computador se comporte de forma inteligente (AWS, 2018; MONARD; BARANAUSKAS, s.d.a).

Por ser um campo que envolve muitos conhecimentos, a IA se divide em diversas sub-áreas indo desde a matemática à psicologia. Ela, ainda, se divide em segmentos de pesquisa sendo os sistemas baseados em casos, robótica, aprendizado de máquina, entre outros. Assim, possibilita ao computador desempenhar diversas funções que antes eram possíveis apenas para os humanos (MONARD; BARANAUSKAS, s.d.a).

A IA já é aplicada em diversas áreas, indo da culinária a medicina. A IBM é uma empresa que vem

inovando nessas áreas com o uso da inteligência aplicada em máquinas. O supercomputador, denominado Watson, baseado em um conjunto de mais de 30 mil receitas, tabelas contendo o perfil molecular dos compostos responsáveis pelo aroma e sabor dos alimentos e estudos acadêmicos na área da culinária, foi capaz de criar receitas novas que foram colocadas no livro *Cognitive Cooking with Chef Watson*, numa parceria entre a IBM e o *Institute of Culinary Education* (Endeavor Brasil, 2016). Já na área da medicina, a IBM possui uma ferramenta, denominada *Watson for Oncology*, que auxilia no diagnóstico de câncer em pessoas. No *site* da empresa, eles afirmam que, na grande maioria dos casos, a ferramenta é compatível com os diagnósticos feitos por médicos. E não é só a IBM que trabalha com IA, os *smartphones*, por exemplo, da Apple e da Samsung, grandes marcas de *smartphones*, já possuem assistentes virtuais que respondem aos usuários e os ajudam em algumas tarefas, como abrir aplicativos ou até mesmo em pesquisas na Internet (Endeavor Brasil, 2016).

Para fazer com que o computador tenha habilidade de reconhecer e diagnosticar uma pessoa com câncer, por exemplo, ele precisa de uma base de dados e de técnicas que o permitam aprender com as informações disponíveis. O aprendizado de máquina é um campo da IA que foca no desenvolvimento de técnicas e na construção de sistemas possíveis de adquirir conhecimento (MONARD; BARANAUSKAS, s.d.b).

2.5.1 Aprendizado de Máquina

Ter a habilidade de aprender é fundamental para um comportamento inteligente. O cérebro humano consegue fazer generalizações através de alguns fatos ou descobre padrões por meio de observações que faz sobre diversas situações, e isto se dá através do aprendizado (CARRIJO, 2004, p. 86). Em computadores, a forma de fazer com estes aprendam é através de um campo da IA chamado aprendizado de máquina. O aprendizado de máquina é, segundo Oliveira (s.d.), “[...] uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática”.

Os humanos, com base em seus conhecimentos e experiências, são capazes de tirar suas próprias conclusões. Isto é chamado de indução e é um modo de inferência lógica que possibilita as pessoas concluírem a respeito de um conjunto de exemplos, inicialmente ele é originado de em um conceito específico e é generalizado posteriormente para um todo. Através do conhecimento indutivo é possível obter novos conhecimentos e antever eventos futuros (MONARD; BARANAUSKAS, s.d.b, p. 40). Este tipo de “ferramenta”, muito utilizado pelo cérebro humano, foi repassado às máquinas e foi chamado de aprendizado indutivo. Este é realizado através dos exemplos proporcionados por processos externos aos sistemas de aprendizado. Este aprendizado é dividido em supervisionado e não-supervisionado. No aprendizado supervisionado o algoritmo de aprendizado, ou indutor, recebe um conjunto de treinamento no qual os rótulos e atributos de cada classe são conhecidos. Neste caso, o objetivo é o de criar um classificador que irá definir a classe dos novos exemplos que não possuem rótulos ainda. Quando a classe possui rótulos de valores discretos, o problema é conhecido como classificação, e quando possui valores contínuos é chamado de regressão. Já no aprendizado não-supervisionado o indutor não tem conhecimento prévio sobre as classes. Neste caso, indutor analisa os exemplos e vê a forma como estes podem ser agrupados, gerando assim agrupamentos ou *clusters*.

O aprendizado de máquina é aplicado em diversas áreas. As empresas Netflix e Spotify, por exemplo, usam o aprendizado de máquina para criarem seus sistemas de recomendação de séries e músicas tendo como base o que os usuários acessam (BARROS, 2016). A empresa Facebook também vem investindo no aprendizado de máquina. Numa conferência, a F8, que aconteceu na Califórnia, a empresa divulgou que está construindo modelos de aprendizado de máquina para reconhecimento facial. Segundo Saturno

(2018), neste dia foi revelada uma pesquisa de como o Facebook, com base em mais de 3,5 bilhões de fotos públicas do Instagram e mais de 17 mil *hashtags* que acompanham as fotos, estava desenvolvendo a sua própria tecnologia de reconhecimento imagem. Saturno (2018) diz que

O material que cedemos ao Facebook quando postamos uma foto no Instagram vai além da imagem dos nossos rostos: quando combinamos a fotografia com o uso das *hashtags*, criamos um contexto para aquela imagem. Assim, a fase inicial da pesquisa teve foco em hierarquizar as *hashtags* que fossem mais relevantes. Termos que eram usados como sinônimos foram mapeados, de forma que foi possível determinar quais *hashtags* eram mais específicas e quais eram mais gerais, levando ao que os pesquisadores chamaram de "large-scale hashtag prediction model", ou algo como "modelo de previsão de *hashtag* em grande escala", em tradução livre.

O aprendizado de máquina envolve a construção de sistemas capazes de extrair informações a partir de um conjunto de dados conhecidos e a utilização desse comportamento aprendido para a resolução de novos problemas. Este trabalho, busca gerar os indicadores de eficiência para o problema do IFSC São José, através da técnica de aprendizado de máquina, usando ferramentas como por exemplo o [TensorFlow](#) da Google⁴.

⁴ Segundo o site oficial, ele é uma biblioteca para computação numérica de alto desempenho e também oferece suporte para o aprendizado de máquina.

3 PROPOSTA

3.1 Justificativa

A motivação deste trabalho é ajudar a equipe da [CTIC](#) a mensurar a eficiência da infraestrutura atual, no que diz respeito a nuvem privada de contêineres, do [IFSC/SJ](#). O uso desta nuvem é um conceito recente e é preciso analisar esse cenário para compreender a sua taxa de eficiência no que tange ao uso de recursos computacionais - comparado com os antigos modelos (máquinas virtuais, grade computacional, etc.). Também, o intuito é o de auxiliar a equipe da [CTIC](#) nas decisões estratégicas de implantação de soluções e aplicativos, e futuras aquisições de equipamentos e ativos de rede. Com os resultados obtidos a partir deste trabalho, a aplicação deste poderá se estender a empresas para que consigam fazer a mesma mensuração de eficiência de sua infraestrutura, como forma de otimizar a aquisição de novos ou uso em nuvem, por exemplo.

3.2 Objetivos

3.2.1 Objetivo Geral

Com este trabalho pretende-se criar um ferramenta ou um método que auxilie na análise da eficiência da nuvem privada de contêineres utilizada atualmente na [CTIC](#) do câmpus.

3.2.2 Objetivos Específicos

- Estudar a infraestrutura da [CTIC](#);
- Conhecer as técnicas de inteligência artificial que serão aplicadas no trabalho;
- Analisar os *logs* da nuvem;
- Gerar os indicadores de eficiência;
- Analisar com a equipe da [CTIC](#) se os indicadores de eficiência obtidos através da mineração de dados são compatíveis com a realidade atual.

3.3 Metodologia

O trabalho se inicia com estudos sobre a infraestrutura da [CTIC](#), como forma de conhecer o cenário atual do câmpus. Posteriormente, serão analisadas as técnicas de [IA](#) para entender sobre o funcionamento de cada uma. O passo seguinte será selecionar as que vão ser utilizadas no trabalho na parte de mineração de dados. No desenvolvimento do trabalho, serão analisados os *logs* da nuvem do [IFSC/SJ](#) como forma de criar indicadores de eficiência de infraestrutura e, com isto, averiguar se atualmente ela está sendo eficiente. As técnicas, previamente selecionadas, serão aplicadas em conjunto com os *logs* armazenados em um banco de dados não relacional. Os resultados obtidos a partir da mineração destes dados, serão avaliados com a equipe da [CTIC](#) para determinar se são, de fato, compatíveis com a realidade da infraestrutura atual do Instituto.

Tabela 1 – Cronograma de Atividades

	Jul	Ago	Set	Out	Nov	Dez
Estudo sobre a infraestrutura da CTIC	X					
Definição das técnicas IA a serem utilizadas	X	X				
Desenvolvimento		X	X	X	X	
Análise dos indicadores obtidos com a mineração dos dados com a equipe da CTIC					X	
Apresentação						X

REFERÊNCIAS

- ALVES, T. H. C. R. *Uma Arquitetura Baseada em Containers para Workflows de Bioinformática em Nuvens Federadas*. Dissertação (Mestrado) — Universidade de Brasília, 2017. Disponível em: <http://repositorio.unb.br/bitstream/10482/30994/1/2017_TiagoHenriqueCostaRodriguesAlves.pdf>. Acesso em: 28 de maio de 2018. Citado na página 20.
- AWS. *O que é inteligência artificial?* 2018. Disponível em: <<https://aws.amazon.com/pt/machine-learning/what-is-ai/>>. Acesso em: 19 de junho de 2018. Citado na página 23.
- BARROS, P. *Aprendizagem de Máquina: Supervisionada ou Não Supervisionada?* 2016. Disponível em: <<https://medium.com/opensanca/aprendizagem-de-maquina-supervisionada-ou-n%C3%A3o-supervisionada-7d01f78cd80a>>. Acesso em: 24 de junho de 2018. Citado na página 24.
- BRAGA, A. S.; SILVA, G. M.; BARROS, M. C. Cloud computing. *Instituto de Computação - Universidade Estadual de Campinas*, 2012. Disponível em: <<http://www.ic.unicamp.br/~ducatte/mo401/1s2012/T2/G08-079713-079740-820650-t2.pdf>>. Acesso em: 4 de abril de 2018. Citado na página 18.
- BURNS, B. et al. Borg, omega, and kubernetes. *Commun. ACM*, ACM, New York, NY, USA, v. 59, n. 5, p. 50–57, abr. 2016. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/2890784>>. Citado na página 20.
- CAMPBELL-KELLY, M. *A origem da Computação: Era da informação começou ao se perceber que máquinas poderiam imitar o poder da mente*. s.d. Disponível em: <http://www2.uol.com.br/sciam/reportagens/a_origem_da_computacao.html>. Acesso em: 10 de abril de 2018. Citado na página 15.
- Canal Westcon. *PLANEJAMENTO DO SDDC: FOCO EM INTEGRAÇÃO E ESCALABILIDADE PARA O SEU CLIENTE*. 2016. Disponível em: <<http://blogbrasil.westcon.com/planejamento-do-sddc-foco-em-integracao-e-escalabilidade-para-o-seu-cliente>>. Acesso em: 1 de abril de 2018. Citado na página 15.
- Canal Westcon. *VIRTUALIZAÇÃO E CLOUD COMPUTING ESTÃO RELACIONADAS?* 2017. Disponível em: <<http://blogbrasil.westcon.com/virtualizacao-e-cloud-computing-estao-relacionadas>>. Acesso em: 3 de abril de 2018. Citado na página 17.
- CARISSIMI, A. Virtualização: Princípios básicos e aplicações. *ERAD*, p. 39–69, 2009. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/erad/2009/004.pdf>>. Acesso em: 25 de abril de 2018. Citado na página 13.
- CARRIJO, I. B. *Extração de regras operacionais ótimas de sistemas de distribuição de água através de algoritmos genéticos multiobjetivo e aprendizado de máquina*. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, 2004. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18138/tde-01032005-154332/pt-br.php>>. Acesso em: 24 de junho de 2018. Citado na página 24.
- CIENA. *O que é SDN?* s.d. Disponível em: <http://www.ciena.com.br/insights/what-is/What-is-SDN_pt_BR.html>. Acesso em: 3 de abril de 2018. Citado na página 17.
- COSTA, L. R. Openflow e o paradigma de redes definidas por software. 2013. Disponível em: <http://bdm.unb.br/bitstream/10483/5674/1/2013_LucasRodriguesCosta.pdf>. Acesso em: 2 de abril de 2018. Citado 2 vezes nas páginas 16 e 17.
- DAMASCENO, J. C. *U-CLOUD: uma abordagem para implantação de nuvem privada para a administração pública federal*. Tese (Doutorado) — Universidade Federal de Pernambuco, ago 2015. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/17341>>. Acesso em: 2 de abril de 2018. Citado 4 vezes nas páginas 15, 17, 18 e 19.
- Endeavor Brasil. *Como a inteligência artificial já muda mercados, da gastronomia ao petróleo*. 2016. Disponível em: <<https://endeavor.org.br/inteligencia-artificial-watson/>>. Acesso em: 23 de junho de 2018. Citado na página 24.

- ESPINHA, R. *Microserviços, contêineres e o data center na era digital*. 2018. Disponível em: <https://www.cisco.com/c/m/pt_br/articles/microservicos.html>. Acesso em: 21 de maio de 2018. Citado na página 20.
- FELIPE, D. S. S. et al. Cloud computing: definições e análise para além das vantagens. In: *V Congresso Tecnológico de TI e Telecom InfoBrasil*. [s.n.], 2012. Disponível em: <http://infobrasil.inf.br/userfiles/16-S3-2-97220-Cloud%20computing____.pdf>. Acesso em: 4 de abril de 2018. Citado na página 19.
- FERREIRA, J. *Rumo à consolidação do Data Center Definido por Software*. 2014. Disponível em: <<http://cio.com.br/opiniao/2014/01/21/rumo-a-consolidacao-do-data-center-definido-por-software/>>. Acesso em: 2 de abril de 2018. Citado na página 16.
- FILHO, A. A. G. M. *Infraestrutura de redes de computadores focada em SaaS*. [S.l.], 2012. Disponível em: <https://wiki.sj.ifsc.edu.br/wiki/images/4/4b/TCC_AlexandreAugusto.pdf>. Acesso em: 18 de abril de 2018. Citado na página 13.
- GARCIA, D. Centro de dados (datacenters). *Intr@ciência*, 2012. Disponível em: <http://uniesp.edu.br/sites/_biblioteca/revistas/20170531143315.pdf>. Acesso em: 10 de abril de 2018. Citado na página 15.
- JUNIOR, L. V. *REDES DEFINIDAS POR SOFTWARE: Um guia compilado sobre a evolução das redes, data centers e suas novas arquiteturas, tecnologias e paradigmas*. 2015. Disponível em: <<http://blog.leomarviegas.com/wp-content/uploads/2015/12/Redes-Definidas-por-Software-Leomar-Viegas.pdf>>. Acesso em: 3 de abril de 2018. Citado na página 17.
- Kubernetes. *What is Kubernetes?* 2018. Disponível em: <<https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>>. Citado 3 vezes nas páginas 20, 21 e 22.
- KUSZKA, B. *Desmistificando os containers*. 2017. Disponível em: <<https://canaltech.com.br/infra/desmistificando-os-containers/>>. Acesso em: 25 de abril de 2018. Citado na página 13.
- MANDIC. *O que é IaaS, SaaS, tipos de Cloud e Virtualização*. 2018. Disponível em: <<https://www.mandic.com.br/artigos/o-que-e-iaas-tipos-cloud-virtualizacao/>>. Acesso em: 25 de abril de 2018. Citado na página 13.
- Microsoft Azure. *Projetando microserviços: registro em log e monitoramento*. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/logging-monitoring>>. Acesso em: 5 de junho de 2018. Citado na página 22.
- Microsoft Azure. *O que é PaaS?* 2018. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-paas/>>. Acesso em: 4 de abril de 2018. Citado na página 19.
- Microsoft Azure. *O que é virtualização?* 2018. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-virtualization/>>. Acesso em: 24 de abril de 2018. Citado na página 13.
- MONARD, M. C.; BARANAUSKAS, J. A. Aplicações de inteligência artificial: Uma visão geral. s.d. Disponível em: <<http://dcm.ffclrp.usp.br/~augusto/publications/2000-laptec.pdf>>. Acesso em: 19 de junho de 2018. Citado na página 23.
- MONARD, M. C.; BARANAUSKAS, J. A. *Conceitos sobre Aprendizado de Máquina*. s.d. Disponível em: <<http://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf>>. Acesso em: 20 de junho de 2018. Citado na página 24.
- MONTEIRO, L.; XAVIER, L.; VALENTE, M. T. *Uma Caracterização em Larga Escala da Arquitetura de Sistemas Docker*. 2017. Disponível em: <<http://homepages.dcc.ufmg.br/~mtov/pub/2017-vem-lucas.pdf>>. Acesso em: 20 de maio de 2018. Citado na página 20.
- NEWMAN, S. *Building Microservices: Designing Fine-Grained Systems*. [S.l.]: O'Reilly Media, 2015. ISBN 1491950358. Citado na página 21.
- OLIVEIRA, S. R. M. *Introdução à Aprendizagem de Máquina*. s.d. Disponível em: <<https://www.ime.unicamp.br/~wanderson/Aulas/Aula9/MT803-AprendizadoMaquina-ArvoreDecisao.pdf>>. Acesso em: 23 de junho de 2018. Citado na página 24.

- PESSIS-PASTERNAK, G. *Do caos à inteligência artificial: quando os cientistas se interrogam*. [S.l.]: Unesp, 1992. Citado na página 23.
- Prometheus. *OVERVIEW*. 2018. Disponível em: <<https://prometheus.io/docs/introduction/overview/>>. Acesso em: 6 de junho de 2018. Citado 2 vezes nas páginas 22 e 23.
- Red Hat. *Introdução à virtualização*. 2018. Disponível em: <<https://www.redhat.com/pt-br/topics/virtualization>>. Acesso em: 24 de abril de 2018. Citado na página 13.
- REIS, D. C. dos S. *Execução e Gestão de Aplicações Containerizadas*. Dissertação (Mestrado) — Universidade do Porto, 2017. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/109650/2/236222.pdf>>. Acesso em: 16 de março de 2018. Citado na página 15.
- REIS, R. B. *Teste de desempenho de sistemas de arquivos distribuídos em nuvem computacional privada no IFSC câmpus São José*. [S.l.], 2017. Disponível em: <https://wiki.sj.ifsc.edu.br/wiki/images/c/cd/TCC_Raphael_Brito_Reis.pdf>. Acesso em: 18 de abril de 2018. Citado na página 13.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence*. 3. ed. [S.l.]: Pearson, 2009. ISBN 0136042597. Citado na página 23.
- SATURNO, A. *Fotos do Instagram são usadas para aprendizado de máquina do Facebook*. 2018. Disponível em: <<https://canaltech.com.br/redes-sociais/fotos-do-instagram-sao-usadas-para-aprendizado-de-maquina-do-facebook-112998/>>. Acesso em: 24 de junho de 2018. Citado na página 25.
- SOMAVILA, C. Comparativo de serviços de hospedagem de arquivos em nuvem. 2015. Disponível em: <<http://bibliodigital.unijui.edu.br:8080/xmlui/bitstream/handle/123456789/3286/TCC.pdf?sequence=1>>. Acesso em: 3 de abril de 2018. Citado 3 vezes nas páginas 17, 18 e 19.
- weaveworks. *Monitoring Kubernetes with Prometheus*. 2018. Disponível em: <<https://www.weave.works/technologies/monitoring-kubernetes-with-prometheus/>>. Acesso em: 5 de junho de 2018. Citado na página 22.
- WEBSTER, L. da S. Software defined datacenter – uma proposta de entrega Ágil para os serviços de negócio. s.d. Disponível em: <https://www.riuni.unisul.br/bitstream/handle/12345/3658/SDDC_LEONARDO_DA_SILVEIRA_WEBSTER.pdf?sequence=1&isAllowed=y>. Acesso em: 2 de abril de 2018. Citado na página 16.
- ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *J Internet Serv Appl*, p. 7–18, 2010. Disponível em: <<https://link.springer.com/content/pdf/10.1007/s13174-010-0007-6.pdf>>. Acesso em: 4 de abril de 2018. Citado 2 vezes nas páginas 18 e 19.
- ZUCCHI, W. L.; AMÂNCIA, A. B. Construindo um data center. *Revista USP*, n. 97, p. 43–58, 2013. Disponível em: <<https://www.revistas.usp.br/revusp/article/view/61684/64573>>. Acesso em: 11 de abril de 2018. Citado na página 15.