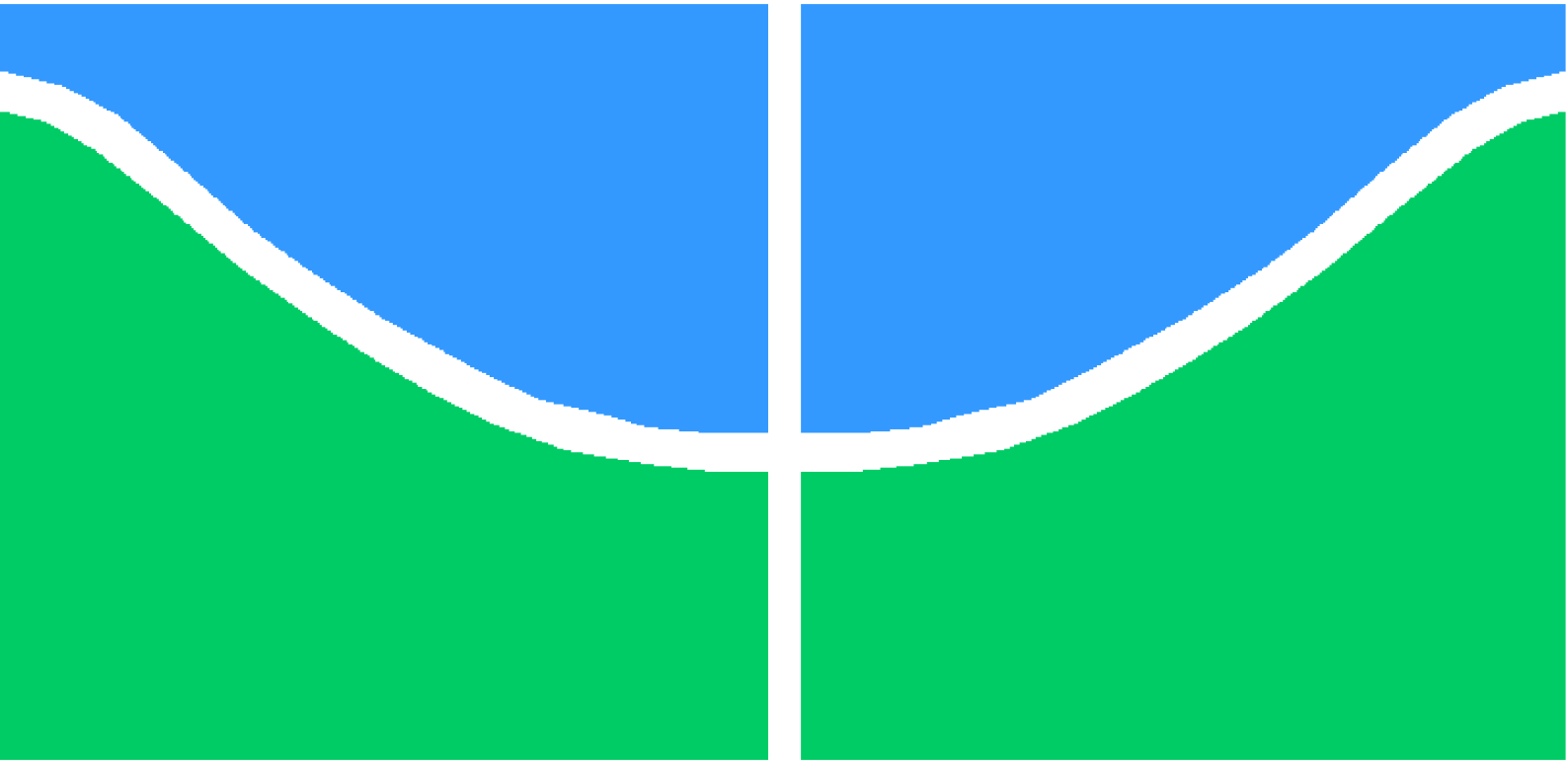

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

Formação de Profissionais DevOps/SRE: processos seletivos e os desafios de ingresso no mercado de trabalho

Autor: Lorrany dos Santos Azevedo

Orientador: Professora Doutora Carla Silva Rocha Aguiar

Brasília, DF

2022

Lorrany dos Santos Azevedo

Formação de Profissionais DevOps/SRE: processos seletivos e os desafios de ingresso no mercado de trabalho

Monografia submetida ao curso de graduação em Engenharia de Softwareda Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Professora Doutora Carla Silva Rocha Aguiar

Brasília, DF

2022

Lorrany dos Santos Azevedo

Formação de Profissionais DevOps/SRE: processos seletivos e os desafios de ingresso no mercado de trabalho/ Lorrany dos Santos Azevedo. – Brasília, DF, 2022-

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

Orientador: Professora Doutora Carla Silva Rocha Aguiar

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Palavra-chave01. 2. Palavra-chave02. I. Professora Doutora Carla Silva Rocha Aguiar. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Formação de Profissionais DevOps/SRE: processos seletivos e os desafios de ingresso no mercado de trabalho

CDU 02:141:005.6

Lorrany dos Santos Azevedo

Formação de Profissionais DevOps/SRE: processos seletivos e os desafios de ingresso no mercado de trabalho

Monografia submetida ao curso de graduação em Engenharia de Softwareda Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

Professora Doutora Carla Silva Rocha Aguiar
Orientador

Titulação e Nome do Professor Convidado 01
Convidado 1

Titulação e Nome do Professor Convidado 02
Convidado 2

Brasília, DF
2022

Agradecimentos

Agradeço a minha família que sempre me apoiou e meu deu forças para seguir, em especial a minha mãe, Márcia dos Santos Silva, pois só estou aqui graças a todo o esforço que ela smepre fez para que eu pudesse ter não só uma boa educação como uma boa vida.

Agradeço ao meu namorado, Marlon Mendes, que me apoiou durante a graduação não só emocionalmente como com questões do cusro de engenharia de software, você foi um grande companheiro.

Resumo

A adoção de práticas DevOps no mercado de tecnologia vem crescendo e ganhando força a cada ano que se passa, isso ocorre pois as empresas de software buscam cada vez mais automatizar seus processos com o intuito de aumentar sua produtividade e eficiência, e assim entregar valor e qualidade ao cliente no menor tempo possível. Apesar da crescente da área no mercado de tecnologia, encontrar, ou se tornar um profissional DevOps Junior pode ser um desafio, devido ao fato de ser uma área relativamente nova, onde se tem pouco material que ajude na formação de profissionais com esse perfil. Neste trabalho iremos analisar o perfil desses profissionais com o intuito de mapear os conhecimentos necessários para o seu ingresso no mercado de trabalho, e assim gerar um material de apoio que facilite a formação desses profissionais, e sirva como um guia do DevOps Júnior.

Palavras-chaves: devops. sre. automação. devops júnior

Abstract

The adoption of DevOps practices in the technology market has been growing and gaining strength over the years, this occuor because software companies increasingly seek to automate their processes in order to increase their productivity and efficiency, and thus deliver value and quality, to the customer in the shortest possible time. Despite the growing area in the technology market, finding or becoming a DevOps Junior professional can be a challenge, due to the fact that it is a relatively a new role, where there is little material that helps in the training of professionals with this profile. In this work, we will analyze the profile of these professionals in order to map the necessary knowledge for their entry into the job market, and thus generate support material that facilitates the training of these professionals, and serves as a guide for DevOps Junior.

Palavras-chaves: devops. sre. automation. junior devops

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CI	Continuous Integration
CD	Continuous Delivery
SRE	Site Reliability Engineering
DevOps	Development Operations
SO	Sistema Operacional
OSI	Open Systems Interconnection
NAT	Network Adress Translation
TCP	Transmission Control Protocol
IP	Internet Protocol
VM	Virtual Machine
IaC	Infra as Code

Sumário

Introdução

Nos últimos anos, houve uma mudança significativa no desenvolvimento de software, empresas que antes mantinham seus próprios servidores e sua infraestrutura, de forma a gerar muitos gastos, começaram a mudar com o surgimento de IaC, cloud services e outras tecnologias.

Com essas mudanças o uso de DevOps no mercado cresceu, e com isso, a demanda por profissionais que sejam qualificados para trabalharem unindo Infraestrutura e Desenvolvimento aumentou. De acordo com uma pesquisa realizada pelo GitLab, cerca de 60% das organizações estão recrutando profissionais de DevOps agora ou no futuro. (GITLAB, 2021)

Entretanto, esse perfil profissional é bem novo no mercado, já que as primeiras discussões sobre DevOps se iniciaram em 2008 na , na Agile Conference – Toronto-C (MELNIK; KRUCHTEN; POPPENDIECK, 2008), desde então a prática DevOps evoluiu substancialmente, e continua a se expandir, incorporando novas ferramentas e tecnologias em resposta às demandas do mercado.

O fato deste perfil profissional ser novo, pode acabar gerando dificuldades na formação desses profissionais, o que acaba acarretando um deficit no mercado em encontrar profissionais que consigam passar pelos difíceis processos seletivos de DevOps , e que sejam qualificados para as vagas ofertadas. (CLARKE, 2021)

O processo seletivo de desenvolvedores já é um processo bem peculiar, este costuma ter diferentes etapas que vão desde a entrevista com o recrutador, até desafios e entrevistas técnicas que abordam diversos conceitos importantes da ciência da computação, além da etapa de alinhamento cultural, entre outras. Um exemplo de processo seletivo que tem todas as etapas citadas anteriormente é o da Amazon. (AMAZON, 2022)

Atualmente é possível encontrar-se com uma certa facilidade materiais que preparam novos desenvolvedores para ingressarem no mercado de trabalho, basta uma busca no google para encontrar diversas publicações como esta: Interviewing at Google: best practices, advice, and tips, além disto, existe uma gama de sites com questões que costumam cair em entrevistas, publicações em blogs dando dicas de material preparatório para os processos seletivos, e até mesmo dicas das próprias empresas de como se preparar para o seu processo seletivo.

Entretanto para os profissionais que desejam ingressar no mundo do DevOps essa mesma facilidade não ocorre, encontrar materiais centralizados que ajudem a ingressar na carreira de DevOps /SRE, muitas vezes pode ser uma tarefa confusa e difícil.

O intuito deste trabalho é, por meio de um estudo de caso, mapear e centralizar os conhecimentos necessários e desejados pelo mercado em um profissional DevOps Júnior, para facilitar a formação e o ingresso destas pessoas na carreira de DevOps /SRE.

Objetivos

O principal objetivo deste trabalho é criar um guia/roadmap de apoio, com os conhecimentos necessários para o ingresso de profissionais juniores na carreira de DevOps/SRE.

O objetivo principal será alcançado através dos seguintes objetivos específicos:

- Traçar o perfil de profissionais DevOps/SRE Junior
- Entender como funcionam os processos seletivos desta área e quais são os seus pré-requisitos
- Mapear os conhecimentos necessários para o ingresso deste profissional no mercado de trabalho
- Listar as principais ferramentas e tecnologias utilizadas em DevOps e cada uma de suas áreas

Estrutura do Trabalho

Este trabalho está organizado em quatro capítulos, são eles:

- Referencial teórico: Onde trabalhos correlatos são analisados e conceitos da área são definidos.
- Metodologia: Aqui iremos apresentar melhor a proposta do trabalho e discorrer sobre as metodologias utilizadas para a sua realização.
- Resultados preliminares: Aqui discutiremos os resultados obtidos.
- Considerações finais: Discutimos as limitações conhecidas e trabalhos futuros.

1 Referêncial Teórico

Neste capítulo falaremos sobre os principais aspectos do desenvolvimento devops encontrados através da análise de trabalhos relevantes da área, com o intuito de termos um maior entendimento sobre os conhecimentos que são necessários para um profissional que deseja ingressar no mercado devops, entendimento desta prática.

1.1 DevOps

De acordo com (DYCK; PENNERS; LICHTER, 2015), DevOps são esforços colaborativos e multidisciplinares em organizações para automatizar a entrega contínua de software. Esses esforços surgem para atender uma demanda do movimento agile de que o software deve ir para produção como pequenas versões.

Antes do surgimento do DevOps , os times que eram responsáveis pela implementação do código do produto, eram separados dos times responsáveis pela implantação do produto. A palavra DevOps surge da junção das palavras "desenvolvimento"(dev) e "operações"(ops).

O conceito de DevOps surge para promover uma maior integração entre essas duas partes, e também uma maior automatização de tarefas operacionais comuns como build, test, release e deploy.

A ideia de unificar desenvolvimento e operações é relativamente nova e teve seu caminho aberto em 2008, na Agile Conference – Toronto-CA (MELNIK; KRUCHTEN; POPPENDIECK, 2008). Desde então a ideia vem ganhando força no mercado e na academia.

O DevOps se tornou uma nova área de atuação, e profissionais DevOps , ou também conhecidos como Site Reliability Engineer (Engenheiro de Confiabilidade de Sites), tiveram sua procura aumentada já que a prática DevOps vem sendo adotada cada vez mais pelas empresas.

1.2 Engenharia de Confiabilidade de Sites

O termo Engenharia de Confiabilidade de Sites (Site Reliability Engineering), ou a sigla SRE (BEYER CHRIS JONES, 2016), trata-se de uma disciplina que engloba práticas de engenharia de software e princípios da ciência da computação aplicados a grandes sistemas distribuídos.

De acordo com o criador do termo, Ben Treynor o aspecto de confiabilidade é crítico e crucial quando se trata de produto de software, sendo assim os esforços de SRE estão voltados para construir sistemas escaláveis, confiáveis e eficientes. E o seu foco pode se dar tanto através de serviços operacionais, quanto da parte de desenvolvimento (BEYER CHRIS JONES, 2016).

Dito isto, é comum que nos refiramos a profissionais ou times que utilizam a prática de DevOps pela sigla ou termo de Engenharia de Confiabilidade de Sites (SRE) já que são conceitos que trabalham com objetivos muito semelhantes.

1.3 Pipeline Devops

Quando falamos de DevOps , falamos principalmente sobre automação, entretanto para que possamos automatizar sistemas e processos de forma a gerarmos eficiência e agilidade, além de precisarmos da colaboração entre as equipes de desenvolvimento e operações, precisamos também das ferramentas e práticas corretas para a implantação desta abordagem. (SHAH; DUBARIA; WIDHALM, 2018)

Dito isto, um Pipeline DevOps é um conjunto de processos e ferramentas automatizadas que garantem a colaboração entre as equipes, e agilidade. No pipeline DevOps , é comum que sejam utilizados os métodos de integração e entrega contínuas, uma infraestrutura que seja escalável, automatizada e que entregue funcionalidades de forma contínua para o cliente. (REDHAT, 2018d).

Dentro deste pipeline que visa automatizar e agilizar entregas, temos práticas como a integração contínua, entrega contínua e testes contínuos. Isto quer dizer que, além de sempre estarmos integrando código novo, estamos sempre testando este código e o levando para produção, como podemos ver na figura 1.

1.4 Runtime

Da perspectiva de engenharia de software os conceitos de runtime (execução) e delivery (entrega) são muito importantes, sendo a parte de *delivery* mais voltada para o desenvolvimento, e a parte de *runtime* mais voltada para operações.

Os conceitos de runtime são fundamentais em DevOps , pois esta prática exige que além de entregarmos código de maneira contínua e versionada, precisamos também entregar software de qualidade e confiável, para isto, precisamos que o produto de software seja performático, escalável, confiável e disponível. (LEITE CARLA ROCHA, 2019)

Para que isto seja possível, precisamos entender os conceitos e as ferramentas certas que farão com que cheguemos ao nosso objetivo. Abaixo iremos explicar todos os

Figura 1 – DevOps Cicle

Fonte: (DUONG, 2021).

conceitos que estão dentro do aspecto de runtime e que tornam possível criar sistemas com as características citadas anteriormente, após isso, na seção de resultados falaremos sobre as ferramentas necessárias para o processo de build, integração contínua, automatização de deploy, monitoramento de sistemas etc, e como escolhemos estas ferramentas.

1.4.1 Integração Contínua

De acordo com (SHAHIN; BABAR; ZHU, 2017), integração contínua é uma prática de desenvolvimento onde membros de uma equipe entregam e mesclam o código desenvolvido diversas vezes ao longo de um dia.

Na integração contínua todo o código integrado deve passar pelo processo de build, onde o código deve ser compilado com o intuito de que se possa identificar os erros o mais rápido possível, o código também passa pela etapa de testes automatizados para garantir o bom funcionamento do sistema.

Sempre que desejamos integrar algum código na ramificação principal, antes o pipeline de integração dispara um conjunto de testes que executa tanto testes unitários, quanto testes end-to-end.

Após isto caso as etapas anteriores não apresentem erros, e o repositório local não apresente nenhum conflito com o remoto que se encontra em alguma plataforma de hospedagem de código fonte (Github, Gitlab), o código poderá ser mergido ao repositório

remoto (FOWLER, 2014).

Vale lembrar que todo o processo de build e testes são automatizados, e para executá-los localmente precisamos apenas executar um comando que irá rodar todo o pipeline de CI de uma só vez (REDHAT, 2018a).

Abaixo podemos ver na figura 2 o fluxo de trabalho de um pipeline de CI, onde uma alteração é subida para o repositório, passa pelos estágios de build e teste, e caso o pipeline seja bem sucedido, as alterações agora ficam disponíveis para outros desenvolvedores terem acesso. Caso contrário, os erros encontrados devem ser corrigidos para que o pipeline funcione como o esperado, e isto não impacte o trabalho de outros membros da equipe.

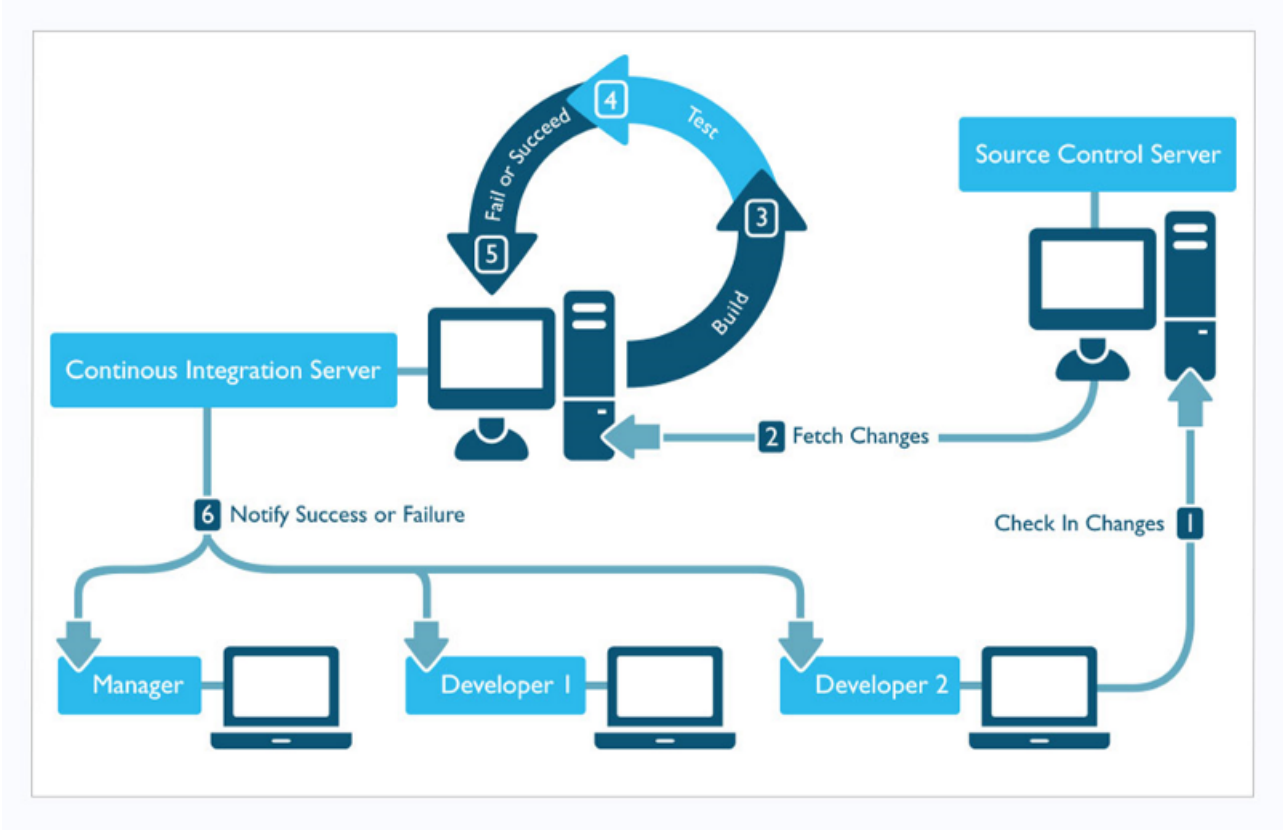

Figura 2 – CI - Workflow

Fonte: (DEVELOPER, 2017).

1.4.2 Controle de versão

Como podemos ver, uma das principais características da integração contínua é a possibilidade de termos uma equipe grande, subindo diversas alterações no código ao mesmo tempo. Para que isso seja possível precisamos manter um bom controle do histórico dessas alterações, além de precisarmos de um jeito seguro de agregar as novas alterações ao código já existente, sem quebrar o seu comportamento.

Existem diversas ferramentas que nos permitem manter esse histórico, mesclar o código, resolver conflitos existentes etc. A ferramenta de gerenciamento de código mais robusta e mais utilizada que temos é o Git, sendo assim é de extrema importância que os desenvolvedores saibam usá-la (STACKOVERFLOW, 2021).

1.4.3 Sistemas Operacionais (SO)

Sistemas operacionais podem ser vistos como programas responsáveis por esconder dos usuários operações relacionadas ao hardware, como o gerenciamento de memória, temporizadores e outros recursos de baixo nível (TANENBAUM, 2008).

Os sistemas operacionais também apresentam para os usuários uma interface amigável, além de servirem como gerenciadores de recursos, coordenando processadores, memória,e dispositivos de entrada e saída, Tanenbaum, dá ainda a seguinte definição para SO:

> "o conceito do sistema operacional como fornecendo principalmente uma interface conveniente para seus usuários é uma visão top-down (de cima para baixo). Uma visão alternativa, bottom-up (de baixo para cima), sustenta que o sistema operacional existe para gerenciar todas as partes de um sistema complexo". (TANENBAUM, 2008, pág 24)

Existem diversos sistemas operacionais, entretanto um que se tornou muito popular na área de TI foram os sistemas operacionais Linux-based, esses SO são baseados no kernel do Linux, que foi desenvolvido por Linus Torvalds e são open source. Alguns exemplos bem populares de distribuições de sistemas operacionais baseados no Linux são: Ubuntu, Debian, Fedora, Manjaro, Arch Linux, entre outros (WIKIPEDIA, 2022b).

Os SO's podem rodar tanto em desktops quanto em servidores, os servidores geralmente são manipulados através de linha de comando, para poupar recursos físicos que seriam consumidos caso utilizássemos uma interface gráfica, já os cloud servers costumam ser Linux-based e também são manipulados via linha de comando.

Através da linha de comando é possível instalar pacotes, fazer o gerenciamento de usuários e a manipulação de arquivos etc. Sendo assim é necessário que o profissionais que pretendem seguir na área de DevOps tenham alguns conhecimentos sobre SO baseados no Linux, como por exemplo:

- Interface de linha de comando (comandos dados para o shell e impressos no terminal)
- Conhecimentos do sistema de diretórios do linux
- Gerenciamento de Pacotes

- Gerenciamento de chaves SSH
- Ferramentas de redes do linux (nmap, ping, iptables, traceroute, netstat)
- Gerenciamento de servidores

Os conceitos de SO, redes e outros que abordaremos logo adiante, são fundamentais para a implantação dos serviços em nuvem, pois, assim como nos servidores "tradicionais", nos servidores em nuvem nós ainda precisamos de um SO para servir as nossas aplicações e quando falamos de infra em nuvem, os SO's que costumam ser mais utilizados são os Linux based (REDHAT, 2019).

1.4.4 Conteinerização e Redes

O Docker é uma ferramenta open source utilizada para a criação e manipulação de contêineres. Atualmente ele é uma das principais ferramentas para se trabalhar com containers, e também um dos principais responsáveis pela popularização da conteinerização. O uso de containers está muito atrelado ao movimento DevOps devido a sua eficiência e capacidade de isolamento.

Apesar de sua popularidade, contêineres linux antecedem o Docker, os sistemas unix por exemplo, já utilizavam contêineres com o intuito de isolar o filesystem.

Um container é um agrupamento de aplicações que rodam em um ambiente isolado do host (máquina física ou virtual), porém compartilham do mesmo kernel (REDHAT, 2018e). Quando falamos em isolamento, isso significa que, se sua máquina física tem uma determinada configuração de desenvolvimento, porém você precisa trabalhar com diferentes configurações e não quer fazer essas configurações no seu host e acabar "poluindo"seu ambiente local, os contêineres vão rodar as aplicações com as configurações desejadas, isoladamente das configurações do seu ambiente local (LINUXTIPS, 2021).

De certa forma, os contêineres se comportam como máquinas virtuais (VM's), para o mundo exterior eles podem parecer seu próprio sistema completo, mas diferentemente de uma VM, ao invés de criar todo um SO virtual, os contêineres precisam replicar apenas os componentes individuais dos quais ele precisa para operar, isso nos dá um aumento significativo de desempenho e reduz o tamanho das imagens. Os contêineres também operam muito mais rápido, pois, diferentemente da virtualização tradicional, o processo está sendo executado de forma nativa em seu host, apenas com uma camada adicional de proteção ao seu redor.

Na figura 3 temos uma ilustração que mostra as diferenças entre se utilizar contêineres e VM´s, como é possível observar, nos contêineres não ocorre uma nova replicação do sistema operacional como ocorrem nas VM´s.

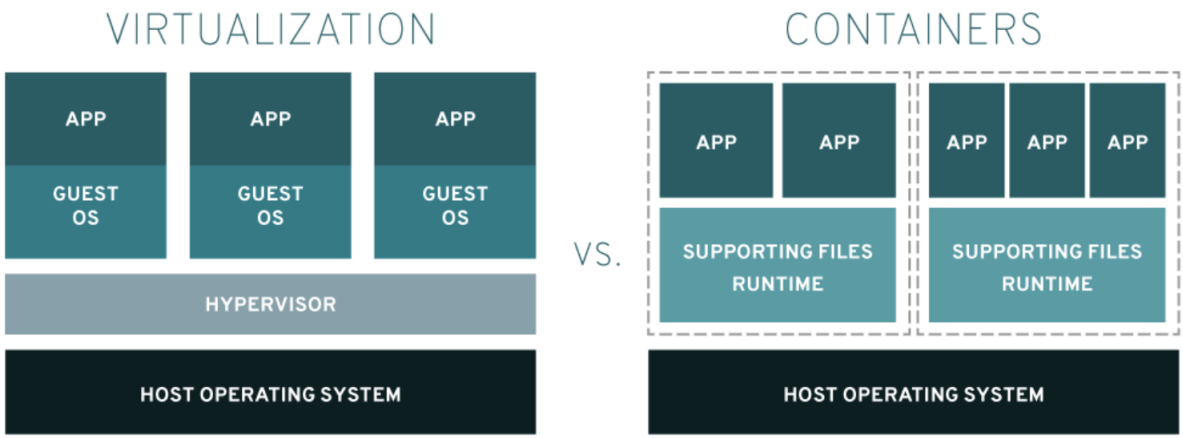

Figura 3 – Container x Máquina Virtual

Fonte: (REDHAT, 2018e).

A engine do Docker é uma aplicação server-cliente, que utilizamos para rodar nossos containers, o docker daemon é o servidor que processa as requisições feitas através de uma interface de linha de comando (cliente), e ambas as partes se comunicam através de uma api rest. O daemon e o cliente podem rodar na mesma máquina, entretanto é possível conectar o cliente a um daemon que roda em máquinas diferentes através de uma interface de rede, a figura 4 nos mostra as camadas envolvidas na engine do Docker (DOCKER, 2022a).

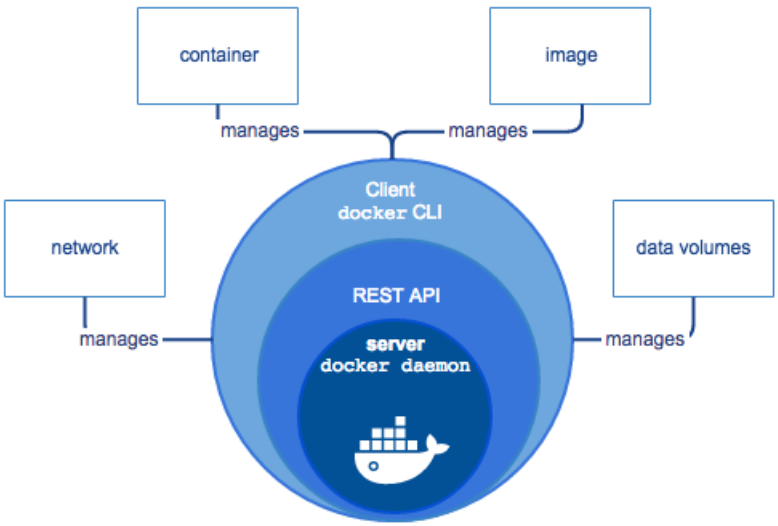

Figura 4 – Docker Engine

Fonte: (AQUA, 2022).

Sendo assim para que você consiga criar e isolar as configurações desejadas de um container, você precisa saber como criar uma imagem docker para que possamos buildá-

las, como será preciso salvar estas imagens em repositórios de artefatos como o Docker Hub, é recomendável que você saiba como criar e gerenciar estes repositórios.

Um conceito muito importante da área da computação é o de redes de computadores, sua definição é dada por um conjunto de dispositivos eletrônicos que estão interligados e trocam dados entre si, esta rede é formada por componentes de hardware (desktops, celulares etc), e software (protocolos de comunicação como o TCP/IP, Http etc) (KUROSE; ROSS, 2014).

Um dos principais exemplos de rede que temos é a internet, a internet interconecta milhares de dispositivos físicos e utiliza os protocolos anteriormente citados para efetuar essa troca de dados. O protocolo mais importante da internet se chama TCP/IP, esse nome é dado pela junção de dois protocolos o TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) e o IP (Internet Protocol - Protocolo de Internet, ou ainda, protocolo de interconexão) (KUROSE; ROSS, 2014).

De acordo com (KUROSE; ROSS, 2014, pág 7) temos a seguinte definição para o que é um protocolo de rede:

> "Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento."

A Organização Internacional de Padrões (ISO) desenvolveu o modelo OSI (Open Systems Interconnection), o modelo OSI divide os protocolos de rede em camadas e cada camada tem sua função, esse modelo surge com o objetivo de facilitar a interconectividade através da especificação das camadas e seus protocolos, abaixo temos uma vista top-down do modelo (WIKIPEDIA, 2022c):

Através da imagem 5 é possível visualizarmos na primeira colunas as 7 camadas do modelo OSI. Já a segunda coluna nos mostra as camadas do protocolo TCP/IP, que de acordo com (BRADEN, 1989), possui quatro camadas diferentes do modelo OSI.

Nos sistemas Unix temos uma ferramenta de interface de usuário chamada IPTables, o IPTables é o responsável por configurar as regras de IP como redirecionamento, filtragem de pacotes, criação de regras de firewall e NAT (Network Address Translation) (WIKIPEDIA, 2022a). O NAT é um protocolo aplicado a camada de rede, e é responsável por traduzir endereços de IP privado[1] e portas de TCP, para um IP público. [2].

[1] Um IP privado é um endereço que identifica um dispositivo em uma rede privada, por exemplo, quando você se conecta a internet da sua casa pelo seu celular e pelo seu computador, cada um desses dispositivos terá um ip privado único. (KUROSE; ROSS, 2014)

[2] Já um IP público te identifica na rede externa de internet e é atribuído pelo seu provedor de internet (KUROSE; ROSS, 2014)

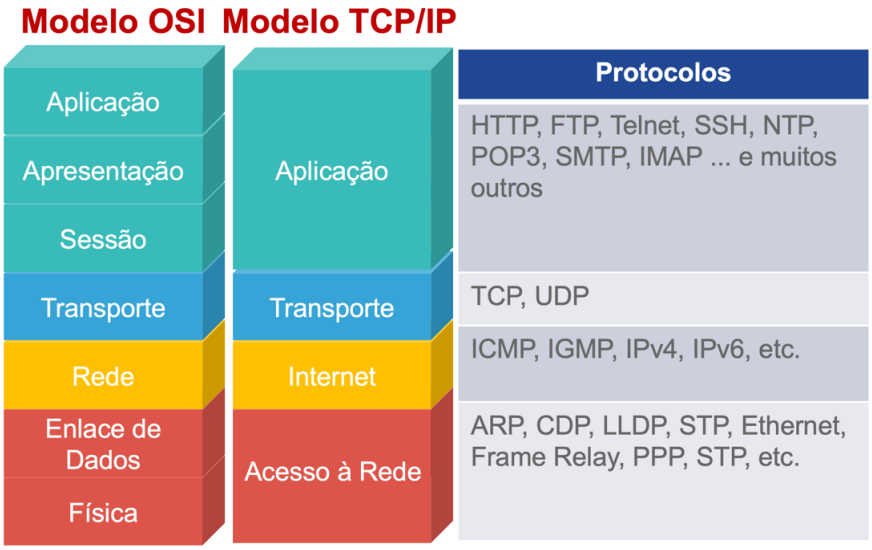

Figura 5 – Modelo Osi - TCP/IP

Fonte: (FURTADO, 2020).

Como já foi citado anteriormente containers são utilizados como uma forma eficiente de isolar diferentes aplicações, sendo assim o docker tem um subsistema de redes manipula o iptables para fazer o isolamento das redes ip do host, entre as redes dos containers (DOCKER, 2022a).

Por default o docker cria três redes no host: a bridge, a host e a none. Quando nós não especificamos qual rede o docker deve usar, ele utiliza a bridge network (rede de ponte). Esse tipo de rede é utilizado quando temos vários containers rodando no mesmo host do docker daemon, sendo assim contêineres que estão conectados a uma mesma rede de ponte podem se comunicar entre si, enquanto estão isolados dos contêineres não conectados a essa rede. Também é possível criar redes de ponte personalizada pelo usuário, e essas irão se sobrepor às redes criadas por padrão (DOCKER, 2022b).

Já a rede host, quando utilizada, diferente da bridge não isola a rede dos contêineres da rede do host onde o docker daemon está sendo executado. Se executar um contêiner que se vincula à porta 80 e usar a rede do host, o aplicativo do contêiner estará disponível na porta 80 do endereço IP do host (DOCKER, 2022c). A terceira rede criada pelo docker no momento da instalação é a none, essa serve para quando você quer desativar todas as redes de um container (DOCKER, 2022a).

Toda a parte de manipulação de redes de containers docker se torna possível devido ao fato do Docker Engine manipular o *iptables* e permitir que o usuário crie suas próprias regras de firewall e redirecionamento. Sendo assim, é necessário conhecimentos prévios em redes, como os citados anteriormente para que possamos fazer essa manipulação de forma correta, abaixo temos uma lista dos conhecimentos necessários para uma melhor utilização das redes do docker:

- Webserver, arquitetura (arquitetura simples back, front, banco, load balancer, DNS)
- Redes públicas e privadas - subredes e como fazemos para se ligarem
- Aplicação: como configurar redes em dockerfiles
- Configurar Firewalls para proteger o aplicativo, entender como endereços IP funcinam, entender os conceitos de portas e DNS
- Load Balancers
- Proxy Server
- HTTP/HTTPS

1.4.5 Computação em Nuvem e Provisionamento

A computação em nuvem (Cloud Computing), é a disponibilização de recursos computacionais sob demanda através da internet. Algumas empresas são responsáveis por disponibilizarem nuvens públicas[3], para que nuvens privadas[4] possam fazer uso dos recursos oferecidos através de pagamento de acordo com a quantidade de recursos utilizados (REDHAT, 2018b).

Na prática o que isto quer dizer é: grandes empresas como a Amazon (Provedor AWS - Amazon Web Services) e a Google (Provedor GCP - Google Cloud Plataform), possuem grandes servidores que tem muita capacidade computacional de armazenamento e processamento, alugam espaços nesses servidores para qualquer pessoa que queira, possa alugar e fazer uso dos recursos disponibilizados.

Os cloud providers disponibilizam diversos recursos de infraestrutura (redes, serviços de banco de dados, armazenamento de dados), plataformas [5], etc. O uso destes vem crescendo, pois comprar máquinas de servidores pode ser muito caro, devido ao alto custo do hardware, alto consumo de energia, etc. As máquinas/servidores também podem ocupar muito espaço físico e caso você queira escalar sua infraestrutura você precisará comprar mais hardware, consumir mais energia e ampliar seu espaço físico, além dos custos de manutenção e outros que podem surgir.

Os cloud providers vem se tornando cada vez mais populares pois além de oferecer servidores potentes, ainda oferecem todos os serviços listados anteriormente, além

[3] "Um ambiente de nuvem criado a partir de recursos sem um proprietário, como um usuário final, que pode ser redistribuído para outros locatários."(REDHAT, 2018b)

[4] "Os provedores de serviços de nuvem privada, também conhecida como nuvem privada gerenciada, oferecem aos clientes uma nuvem que é implantada, configurada e gerenciada por terceiros."(REDHAT, 2018b)

[5] Platform-as-a-service (PaaS): Solução que permite ao usuário desenvolver, executar e gerenciar aplicações sem precisar criar e manter a infraestrutura ou plataforma necessárias para a execução dessas aplicações (REDHAT, 2019b)

do suporte prestado aos contratantes dos serviços (REDHAT, 2018c). Ou seja, utilizar cloud providers pode ser uma boa opção para empresas com poucos funcionários e baixo orçamento, pois isto barateia o custo de se manter uma infraestrutura convencional.

Além disso, temos também a questão da escalabilidade, quando falamos de escalonamento de infraestrutura, temos dois conceitos que precisam ser entendidos, o primeiro é o escalonamento vertical, que consiste basicamente em adicionar mais poder de processamento aos servidores que hospedam os serviços. Já o escalonamento horizontal ocorre através da adição de mais instâncias do servidor (máquinas virtuais por exemplo), que irão rodar cargas de trabalho definidas pelo load balance[6]. Ambas as técnicas possuem prós e contras, abaixo citaremos alguns deles (LIU et al., 2014).

Prós e contras do escalonamento vertical:

- Pró: Centralização dos dados e serviços em um único servidor
- Pró: Comunicação entre processos (Mais rápida)
- Contra: Alto custo de manutenção de infraestrutura
- Pró/Contra: Ponto único de falha, ao mesmo tempo que isto facilita a identificação de falhas devido a centralização dos dados, também é um contra pois caso ocorra algum problema isso pode deixar os serviços fora do ar até que esse problema seja resolvido (SANTANA, 2019).

Prós e contras do escalonamento horizontal:

- Pró: O sistema é capaz de reagir a falhas caso ocorra erro com algum dos processos já que os mesmos são executados em diferentes hosts.
- Pró: Custo mais baixo de manutenção de infraestrutura.
- Contra: Comunicação remota entre processos (por rede), como os processos são executados em diferentes máquinas eles se comunicam através das redes, o que torna o desempenho mais lento.
- Contra: Dificuldade de manter a consistência dos dados, como temos diversos servidores processando diversas requisições, isso torna mais difícil o processo de alocar os dados das requisições de forma não conflituosa e integra (SANTANA, 2019).

Na figura 6 podemos observer na imagem a direita como ocorre o escalonamento vertical (scale up), e a esquerda temos a representação do escalonamento horizontal (scale out).

[6] Load Balance: Distribui as cargas de trabalho entre diferentes máquinas.

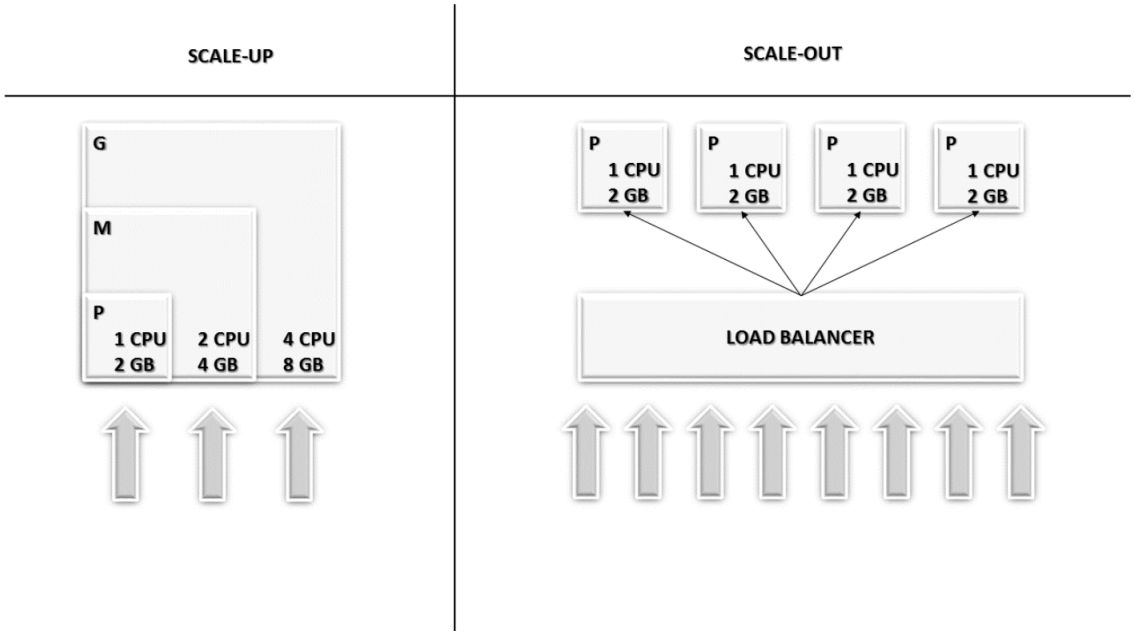

Figura 6 – Escalonamento Vertical (Scale Up) x Escalonamento Horizontal (Scale Out)
Fonte: (AGILITY, 2016).

1.4.6 Infraestrutura como Código

O conceito de infra-as-code, ou infraestrutura como código, está muito ligado aos conceitos abordados no tópico anterior (Cloud Computing e Provisionamento), já que se trata do provisionamento e gerenciamento de uma infraestrutura (física ou em nuvem), através do uso de scripts. Como a configuração da infra se dá através de arquivos, essas configurações podem ser salvas em sistemas de versão de controle como o Git, o que automatiza todo este processo de manutenção infra, já que podemos rastrear, reutilizar e compartilhar essas configurações (TERRAFORM, 2022).

A figura 7 mostra o fluxo de trabalho de IaC.

Benefícios do uso de Infraestrutura como código:

- Manutenir os pipelines criados com atualizações, mudanças de comportamento etc.
- Se atentar a boas práticas de segurança para que credenciais e dados sensíveis não sejam vazados no momento do deploy, ou de subir alterações de código para repositórios.
- Manter as ferramentas utilizadas no pipeline sempre atualizadas para que não tenham falhas de segurança decorrentes de terceiros.

Dificuldades do uso de Infraestrutura como código:

- Automatizar processos de atualização e instalação de pacotes.

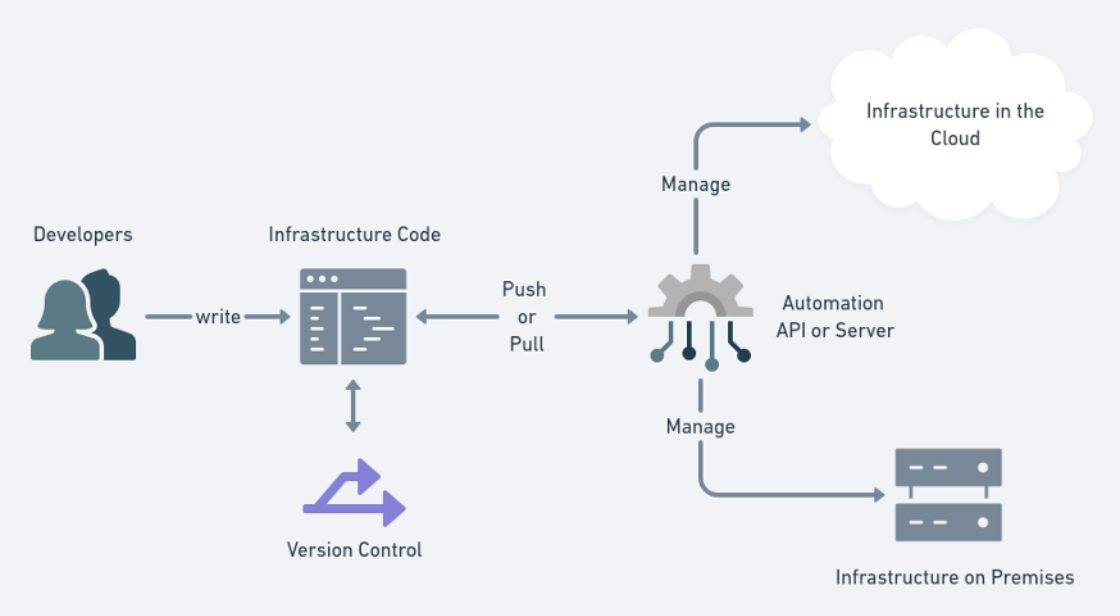

Figura 7 – Fluxo de Trabalho da Infraestrutura como Código

Fonte: (FERNANDEZ, 2020).

- Consistência, já que como mantemos versões de controle das configurações, evitamos erros que poderiam acontecer em processos de configuração manuais.
- Agilidade, já que podemos configurar uma infraestrutura complexa apenas com a execução de scripts.

1.4.7 Gerenciamento de configuração

Para que possamos gerenciar os recursos do sistema como servidores, bancos de dados, máquinas virtuais etc, precisamos de um gerenciador de configuração. Estas ferramentas irão nos ajudar a implantar modificações no nosso sistema,de forma controlada e segura, estas também irão rastrear o estado do sistema, evitando assim comportamentos que possam gerar alguma instabilidade, tudo isto é feito através de código que pode ser versionado e reutilizado, isto faz com que seja fácil alterar comportamentos no servidor sem grandes complicações. No mercado existem diversas ferramentas disponíveis para isso como o Chef, Ansible, Puppet, etc.

O gerenciamento de configuração está fortemente ligado a IaC, devido a parte de automatização e escalabilidade, já que essas ferramentas nos ajudam a manter o ambiente no estado desejado (REDHAT, 2019a).

1.4.8 Monitoramento

O termo monitoramento dentro dos conceitos de DevOps/SRE, trata-se de coletar, processar e agregar dados em tempo real sobre um sistema, com o intuito de observar o

desempenho do sistema (tempo de processamento), quantidade de erros, falhas de rotina etc, e nos retornar dados quantitativos acerca de tudo o que foi monitorado (BEYER CHRIS JONES, 2016).

Os dados observados através das ferramentas de monitoramento são utilizados para que as equipes de DevOps consigam resolver os erros encontrados o mais rápido possível, melhorar questões de desempenho e consumo de hardware nos servidores da aplicação, otimizar os processos, ou até mesmo evitar futuros erros no sistema. Isso faz com que as aplicações se tornem confiáveis, disponíveis e eficientes (BEYER CHRIS JONES, 2016).

2 Metodologia

2.1 Proposta

No presente trabalho temos como principal objetivo identificar o perfil técnico de profissionais DevOps Júnior, e mapear/elaborar um material de apoio com os conhecimentos necessários para a formação de um profissional com este perfil de acordo com as demandas do mercado. Portanto ao final desse trabalho as seguintes questões de pesquisa serão tratadas:

1. Quais os conhecimentos técnicos cobrados em processos seletivos de DevOps / SRE?

2. Quais os conhecimentos técnicos necessários para se preparar para entrevista de DevOps Júnior?

Para que possamos tratar as perguntas acima, iremos definir o perfil de um profissional devops Júnior e fazer um mapeamento dos conhecimentos necessários, através de pesquisas exploratórias, entrevistas não estruturadas com profissionais que já atuam no mercado etc, iremos construir um "guia"que irá auxiliar os profissionais que desejam ingressar na carreira a se prepararem.

2.2 Revisão Bibliográfica

Inicialmente foi realizada uma revisão bibliográfica para identificar os principais conceitos relacionados a DevOps em artigos acadêmicos, com o intuito de identificarmos as principais práticas relacionadas a DevOps para que pudéssemos construir o referencial teórico. Durante as pesquisas e leituras da revisão bibliográfica observou-se uma certa dificuldade em encontrar-se materiais acadêmicos que abordavam a temática dos principais desafios e conceitos associados ao DevOps, sendo assim um dos principais trabalhos utilizados como referência foi o (LEITE CARLA ROCHA, 2019).

A partir daí realizamos uma pesquisa exploratória utilizando a técnica de snowballing em comunidades open source, fóruns de tecnologia, publicações de grandes empresas de tecnologia e outros tipos de literatura cinzenta, para que pudéssemos identificar com ainda mais precisão as práticas relacionadas a DevOps e também quais são as tecnologias que vem sendo mais utilizadas no mundo DevOps de acordo com essas comunidades, algumas publicações como a (OLIVEIRA, 2020), (AMRI, 2017) exemplificam um pouco do que foi buscado na pesquisa exploratória.

2.3 Pesquisa Qualitativa

Além da pesquisa exploratória para identificarmos os principais conceitos DevOps , fizemos também uma pesquisa exploratória para identificarmos o perfil de um profissional DevOps Júnior e o que é exigido deste no mercado de trabalho. Para isso nós buscamos por vagas para profissionais DevOps / SRE Júnior, disponíveis na plataforma Linkedin utilizando algumas strings de busca, primeiro realizamos uma análise das vagas encontradas com o intuito de identificar quais os requisitos necessários para estas vagas e qual a quantidade de vagas para pessoas do perfil buscado. Após este processo nós fizemos a filtragem dos dados que foram recolhidos com o intuito de refinar a nossa analise.Os resultados desta busca podem ser observados nas tabelas 1 e 2 onde relacionamos strings de busca, com o tipo de vaga encontrada e os conhecimentos de cada vaga.

Para que pudéssemos colher ainda mais inputs de tecnologias e principais conceitos DevOps , realizamos uma análise de uma pesquisa quantitativa da empresa StackOverflow, que é uma das maiores comunidades de programadores do mundo. Na pesquisa em questão nós analisamos todos os dados que eram relacionados a DevOps /SRE, no entanto esta pesquisa tem dados acerca de toda a comunidade de software.

Além das pesquisas, realizamos também uma entrevista não estruturada com um profissional que atua como DevOps no mercado. A entrevista em questão foi feita com um profissional DevOps da empresa PicPay, e se deu a partir da seguinte pergunta principal: "Quais os conhecimentos necessários para quem deseja ingressar na carreira de DevOps? ".

Através das pesquisas e entrevistas é notável que conceitos como CI, Iac, SO, cloud computing e outros que são citados diversas vezes neste trabalho aparecem na maioria dos artigos e publicações revisados e são conceitos que muitas vezes são até confundidos com a prática DevOps .

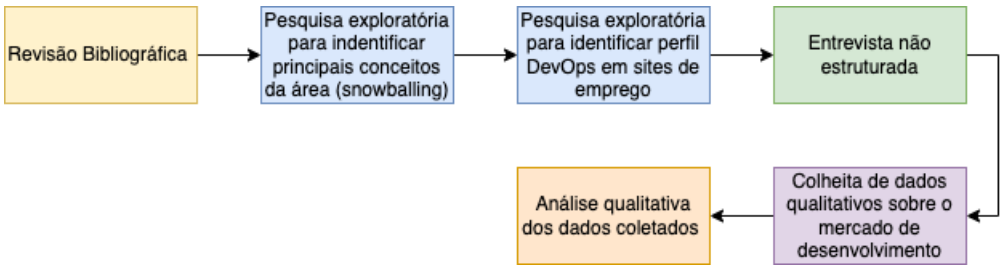

Figura 8 – Processo de pesquisa

Fonte: Autora, 2022.

A imagem 8 mostra o processo de pesquisa que foi seguido para o desenvolvimento desta primeira parte do trabalho e para a coleta dos dados que serão analisados no capítulo de resultados preliminares.

2.4 Escolha de ferramentas

As ferramentas escolhidas para colher os dados da análise qualitativa realizada nesta primeira etapa do trabalho foram: o Linkedin [7], que é uma das maiores redes sociais profissional da atualidade, esta plataforma é focada em gerar conexões entre profissionais e também em divulgar vagas de emprego. Esta plataforma foi escolhida com o intuito de colher dados a respeito de vagas divulgadas por empresas que buscam profissionais DevOps júnior.

A outra ferramenta utilizada neste momento do trabalho foi a pesquisa sobre desenvolvimento do StackOverflow[8], que é uma plataforma de perguntas e respostas voltadas ao desenvolvimento e tecnologias da informação no geral. A plataforma é utilizada no mundo todo e é uma ferramenta muito utilizada pelos desenvolvedores.

[7] https://www.linkedin.com
[8] https://stackoverflow.com/

3 Resultados

Este capítulo apresenta os resultados obtidos a partir das análises qualitativas realizadas através das metodologias adotadas na seção anterior. Aqui apresentaremos os principais conhecimentos e ferramentas necessários para os profissionais que desejam ingressar no mercado, e como se dá o processo seletivo de contratação de pessoas DevOps .

3.1 Etapas de um processo típico de contratação de TI

Os processos seletivos na área de TI costumam ser bem peculiares da área, com diversas etapas que vão desde a entrega do currículo, até a contratação, empresas como a Amazon e a Google, que são grandes empresas de tecnologia cobiçadas por muitos profissionais da área, têm processos seletivos que podem durar até 2 meses, e seus processos seletivos de certa forma moldam os processos de outras empresas de tecnologia.

As etapas que geralmente costumam compor os processos seletivos para desenvolvedores que desejam ingressar em big techs, são as seguintes:

- Seleção de currículo;
- Entrevistas online e/ou por telefone;
- Desafios técnicos que costumam englobar questões de lógica, estrutura de dados, algoritmos, problemas de maratona etc;
- Algumas empresas como a google podem pedir para que o candidato desenvolva uma aplicação (PROCESS, 2022);
- Entrevista de system design;
- Entrevista de cultura para saber se o perfil do candidato se alinha ao perfil da empresa.

Dadas as etapas acima e alta concorrência para os cargos de desenvolvedor nas empresas citadas (que tem os seus processos seletivos considerados os mais difíceis do mercado), é possível encontrar com certa facilidade materiais que preparam os profissionais para esses processos seletivos, abaixo temos alguns exemplos.

- Cracking the Google coding interview: The definitive prep guide

- Interviewing at Google: best practices, advice, and tips
- HackerRank (Site com questões de código que costumam cair nessas entrevistas)
- LeetCode (Site com questões de código que costumam cair nessas entrevistas)

3.2 Busca e análise de vagas DevOps no Linkedin

A tabela 1 mostra a relação de resultados encontrados dada uma string de busca e os filtros do Linkedin aplicados nesta busca, nesta tabela podemos também observar o número de vagas encontradas para cada combinação.

Número da busca	String de Busca	Filtros aplicados	Vagas encontradas
1	"Devops Junior"	Nenhum	9
2	"Devops Junior"	Júnior	6
3	"SRE Junior"	Nenhum	2
4	"SRE"	Junior	57
5	"Devops"	Junior	764

Tabela 1 – Busca por vagas de DevOps Júnior no Linkedin

Na busca de número 2 da tabela 1 foram encontrados 6 resultados de vagas, entretanto, lendo as especificações das vagas encontradas, de 6, apenas uma era de fato Júnior, as outras eram para profissionais sênior/pleno e exigiam alguns anos de experiência na área. Na busca de número 4 foram encontrados 57 resultados, entretanto apenas 6 eram de fato para pessoas DevOps / SRE júnior. Na busca número 5 foram encontrados 764 resultados, entretanto muitas vagas eram para pessoas sênior/pleno.

Todas as buscas citadas acima utilizaram filtros do Linkedin, que se mostraram não serem muito eficientes pois a maioria das vagas encontradas mesmo com a aplicação de filtro "Júnior"eram vagas que exigiam uma senioridade maior em sua descrição, ou experiência de mercado de 3 a 5 anos, o que não condiz com o tempo de experiência de uma pessoa júnior, além disso algumas vagas pediam conhecimentos avançados em determinadas ferramentas como Kubernetes, AWS e outras, o que também não condiz com o nível de conhecimento de uma pessoa que está ingressando no mercado de trabalho.

Na parte de quantidade de pré-requisitos exigidos na vaga foi um pouco mais difícil avaliar o que poderia ser justo, ou não para uma pessoa júnior já que esse quesito é muito subjetivo, mas conhecimentos obrigatórios em monitoramento foram considerados como requisitos que não fazem parte do escopo de uma pessoa júnior pois exigem uma experiência com monitoramento de sistemas reais, diferente de outras ferramentas que podem ser aprendidas e bem compreendidas em aplicações e produtos fora do mercado de trabalho.

Vaga	String utilizada para encontrar a vaga	Conhecimentos	Ferramentas
1	Devops Júnior	Cloud Computing, Microsserviços, IaC, CI/CD, SO (Windows, Linux), Monitoramento, BD, Containers, Pipelines	Rancher, Kubernetes, Docker, Jenkins, Gitlab ci, Azure DevOps, Bamboo, Travis, BitbucketCI AWS, Azure, OCI e GCP Prometheus, Graphana, NewRellic, Datadog, Zabbix MySQL, Postgres, SQL SERVER, Mongo, Git
2	Devops Júnior	CI/CD, Deploy, Manutenção de Infra, Containers, Redes, SO (Linux), Escalabilidade, Deploy,	Docker/Compose/Swarm, Kubernetes, Terraform, AWS, Google Cloud, Azure, Jenkins, Drone, Travis, Phyton e Shellscript (Bash), Git
3	Devops Júnior	Cloud computing, SO (Linux), Linux/Windows Server, Containers	Azure, GitLab, PowerShell/Python shell, Docker, K8s
4	Devops Júnior	Cloud computing, SO (Linux), Redes, BD	AWS
5	Devops Júnior	IaC, Containers, SO (Linux), Microsserviços, Cloud Computing	Docker, AWS, Azure, Terraform, Ansible, Chef, Docker Swarm ou Kubernets
6	Devops Júnior	CI/CD, testes automatizados, IaC, Pipelines, Cloud Computing, SO (Linux), Monitoramento, Microsserviços, BD	Jenkins, Gitlab ci, Azure DevOps, Bamboo, Travis, BitbucketCI, Ansible, Chef, Puppet, OpsWorks,cloudformation, terraform, Groovy, Gradlle, AWS, Azure, OCI e GCP, Prometheus, Graphana, Site24x7,NewRellic, Datadog, Zabbix, Rancher, Kubernetes, Docker, Phyton e Shellscript (Bash).
7	Devops Júnior	SO (Linux), Redes, BD, Monitoramento, Containers	Phyton e Shellscript (Bash), Zabbix, Grafana, Prometheus, Gitlab, AWS (EC2) e RDS, Kubernetes, Dockers, Stack ELK, TomCat, Apache (Zookeeper e Storm) e HAProxy
8	Devops Júnior	IaC, Servidores Linux, Redes, CI/CD, Containers	AWS, AZURE, GCP, kubernetes
9	Devops Júnior	CI/CD, Containers	Kubernetes
10	SRE Júnior	Cloud, SO (Linux), Containers, BD, Monitoramento	Shell (python), Kubernetes, Docker, Azure e AWS, Zabbix, AppDynamics e DataDog

A partir das buscas realizadas na 1 analisamos 14 vagas para DevOps Júnior, com intuito de identificar quais os requisitos e conhecimentos que costumam ser pedidos pelas empresas, quando eles buscam por profissionais com este perfil. As especificações de cada vaga podem ser vistas juntamente com a string de busca que utilizamos para encontrar a vaga na tabela 2.

11	SRE com filtro Júnior do Linkedin	Microsserviços, Cloud Computing, SO (Linux), Containers, CI/CD, provisionamento, monitoramento, BD	AWS, Linux com scripts em Bash / Shell e / ou Python, Docker, Kubernetes, Helm, Jenkins, Github Actions, Flux, GitLab, CircleCI, Terraform, Clouformation, Ansible, Chef, Puppet, Elasticsearch, Logstash, Kibana, New Relic, Prometheus, Grafana,
12	SRE com filtro Júnior do Linkedin	SO (Linux), Redes, CI/CD, Containers, Bash, PowerShell/ Shell Script Python,	Jenkins e Groovy, Docker e Kubernetes, Chef, Puppet, Salt, Ansible e Terraform, AWS, Git
13	SRE com filtro Júnior do LinkedIn	Cloud Computing, CI/CD, gitflow, monitoramento	Docker, Kubernetes, Dynatrace, Grafana/Prometheus
14	SRE com filtro Júnior do Linkedin	Monitoramento, Cloud Computing, (Shell Script / Bash), Containers, SO (Linux), GitFlow, IaC, Provisionamento, CI/CD, TCP/IP Networks (Redes)	CloudFormation ou Terraform, Git, Kubernetes, Docker, Docker swarm, grafana/kibana , Graphite, StatsD, Prometheus, Datadog, Microsoft Azure DevOps

Tabela 2 – Vagas encontradas nas buscas da tabela 1 e requisitos

3.3 Análise dos dados do StackOverflow Developer Survey - 2021

A empresa Stackoverflow publica anualmente uma pesquisa sobre a área de desenvolvimento que conta com participantes de todo o mundo e tem o objetivo de colher dados sobre a comunidade de tecnologia no mundo.

A pesquisa de 2021 contou com a participação de 83.439 pessoas, de 181 países ao redor do mundo, sendo a maioria residente dos Estados Unidos (18.32%), seguido pela Índia (12.6%). No relatório é possível ver dados sobre idade dos participantes, tecnologia em que começaram a programar, cargos em que estão atuando (front-end, back-end, devops, etc), anos de experiência entre outros (STACKOVERFLOW, 2021).

Neste trabalho iremos analisar apenas os dados relacionados aos profissionais que atuam como DevOps , para termos uma ideia melhor do perfil desses profissionais.

De acordo com a pesquisa para se tornar um Especialista em DevOps , o profissional deve ter cerca de 11.26 anos de experiência no mercado de trabalho, já para se tornar um engenheiro de confiabilidade de sites sênior, o profissional leva em torno de 11.33 anos (STACKOVERFLOW, 2021).

A maioria dos participantes dessa pesquisa são profissionais que atuam como Full-Stack Developer (49.47%), e em sexto lugar temos os profissionais Especialistas DevOps (10.62%) (STACKOVERFLOW, 2021).

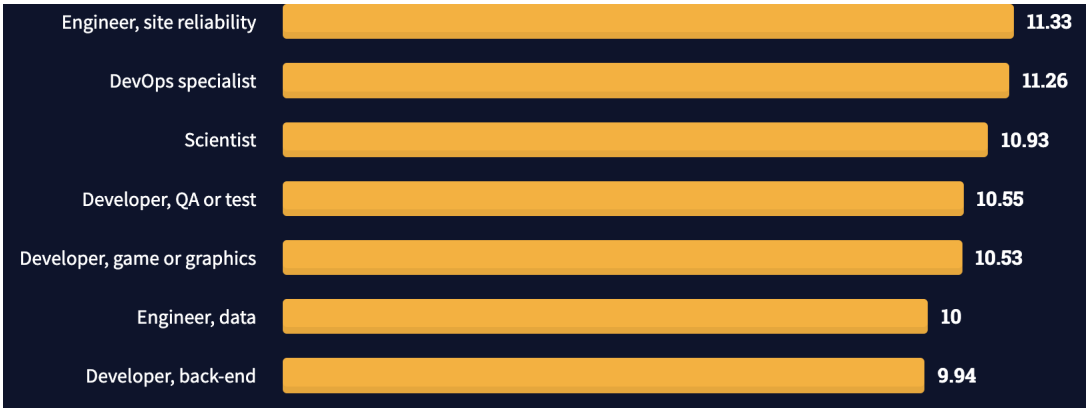

Figura 9 – Anos de experiência profissional em codificação por tipo de desenvolvedor
Fonte: (STACKOVERFLOW, 2021).

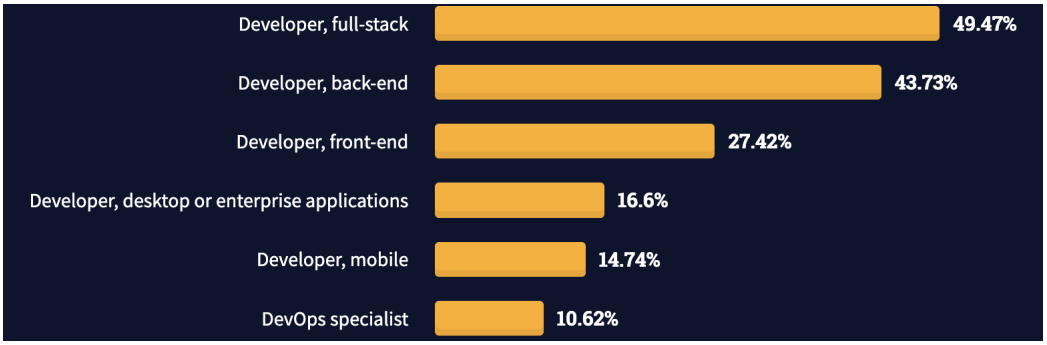

Figura 10 – Perfil dos Desenvolvedores que participaram da pesquisa
Fonte: (STACKOVERFLOW, 2021).

De acordo com a pesquisa as tecnlogias mais populares entre os participantes são as seguintes (STACKOVERFLOW, 2021):

- Linguagens de programação, script e marcação:
  - JavaScript (64.96%)
  - Python (48.24%)
  - SQL (47.08%)
  - Bash/shell (27.13%)
- Bancos de Dados:
  - MySQL (50.18%)
  - PostgresSQL (40.42%)
  - SQLite (32.18%)
  - MongoDB (27.7%)
- Plataformas de nuvem:

  - AWS (54.22%)
  - Google Cloud Platform (31.05%)
  - Microsoft Azure (30.77%)
  - Heroku (24%)
  - Digital Ocean (17.67%)

- Outras ferramentas:
  - Git (93.43%)
  - Docker (48.85%)
  - Yarn (17.73%)
  - Kubernetes (16.6%)
  - Ansible (7.68%)
  - Terraform (7.68%)
  - Puppet (1.8%)
  - Chef (1.35%)

- Sistemas Operacionais:
  - Windows (45.33%)
  - Linux-based (25.32%)
  - MacOs (25.19%)
  - Windows Subsystem for Linux (WSL) (3.29%)

Já na análise de tecnologias mais bem pagas, é possível perceber as tecnologias relacionadas a DevOps pagam melhor em relação a outras tecnologias. Por exemplo, temos Clojure sendo a linguagem de programação mais bem paga com uma média de $ 95.000 dólares anuais, enquanto temos o Pulumi como a tecnologia mais bem paga entre todas as outras (banco de dados, plataformas, frameworks etc), com uma média salarial anual de $ 109.824.

Na análise salarial os profissionais DevOps e SRE se encontram no top 5 de melhores remunerações como é possível se observar na figura 11.

> "Embora os gerentes de engenharia, SREs e especialistas em DevOps paguem mais, vemos que eles também têm em média, mais de dez anos de experiência profissional, enquanto os cientistas de dados ou especialistas em aprendizado de máquina estão em 8º da lista dos mais bem pagos, mas, em média, têm menos anos de experiência. Os designers são os mais mal pagos, embora tenham,

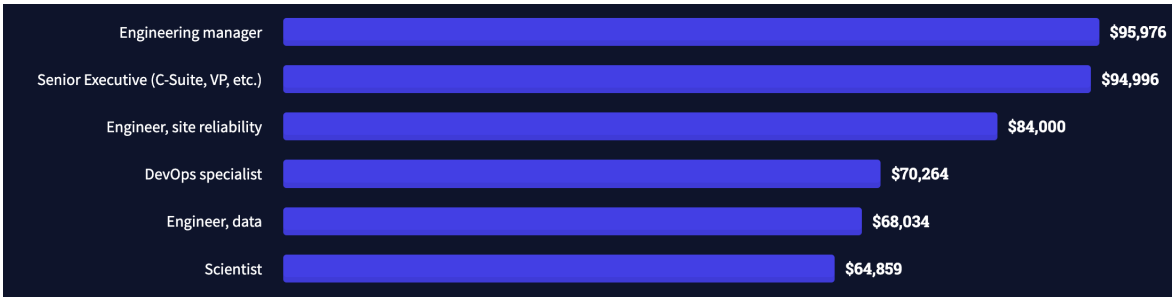

Figura 11 – Salário pelo tipo de desenvolvedor

Fonte: (STACKOVERFLOW, 2021).

em média, mais de dez anos de experiência"(STACKOVERFLOW, 2021). Tradução Livre: Autora

3.4 Entrevista não estruturada

A conversa desta entrevista se deu através da seguinte questão: "Quais os conhecimentos necessários para quem deseja ingressar na carreira de DevOps?". A partir disso, um documento no Google Drive foi criado, e o entrevistado abordou todos os conceitos/tecnologias com as quais ele trabalha, ou acredita serem fundamentais para quem deseja iniciar na área.

No documento em questão os seguintes aspectos foram abordados e tidos como essenciais para um profissional DevOps :

Operação - Revisão Geral

- Comandos básicos de linux/shell;
- Ferrametas de redes (nmap, ping, iptables, traceroute, netstat);
- Conceitos básicos de administração de servidores;
- Linux File System;
- Chave SSH;
- Gerenciamento de pacotes;

Redes em Docker

- Webserver, arquitetura (arquitetura simples back, front, bancos, load balancer, DNS, TCP/IP)
- Redes públicas e privadas

- Aplicação: como configurar redes em dockerfiles
- Firewall
- Proxy Server
- HTTP/HTTPS

Cloud Provider

- Amazon VPC (subnets, security groups) - virtual private cloud
- AWS EC2 (load balancer)
- Amazon ECS
- AWS SDK

Infra como código

- Versionamento de Infra
- Terraform e suas configurações
- Instância da infra em container
- Terraform e pipeline

Monitoramento

- Cloud Watch

3.5 Perfil Profissional DevOps Júnior

Através da análise dos dados colhidos nas seções anteriores do presente trabalho, é possível definir o perfil de um profissional DevOps Júnior e quais os conhecimentos necessários para o seu ingresso no mercado de trabalho. Vale ressaltar que aqui teremos foco em formar profissionais com conhecimento prévio em desenvolvimento, já que os conhecimentos que abordaremos nesta seção já assumem que o profissional tenha alguma prática em linguagens de programação e lógica de programação.

A tabela 1 nos mostra os resultados das vagas encontradas no Linkedin utilizando os filtros de busca definidos, através dos dados colhidos nesta busca foi possível observar o quão difícil é encontrar vagas para profissionais DevOps Júnior já que mesmo utilizando os

Conhecimentos Essenciais	Conhecimentos Desejáveis
- Controle de Versão (Git) - Containers (Docker, Docker Compose) - CI (Jenkins, GitLab CI etc) - SO (Linux-based) - Python ou alguma outra linguagem de script - Bash/shell - Cloud Computing (AWS) - Redes (TCP/IP, HTTP/HTTPS, Firewall, DNS)	- Infraestrutura como código e provisionamento (Terraform, Ansible) - Orquestração de Containers (Kubernetes, Docker Swarm) - Monitoramento (Grafana, Amazon CloudWatch etc) - Banco de Dados (MongoDB, PostgresSQL etc)

Tabela 3 – Conhecimentos Essenciais x Desejáveis para um DevOps Júnior

filtros de busca e as ferramentas disponibilizadas pelo Linkedin, encontrados pouquíssimas vagas para DevOps Júnior.

A tabela 3 mostra os conhecimentos específicos que foram julgados como essenciais e/ou desejáveis através dos inputs de dados do capítulo 2.

3.5.1 Processo seletivo de DevOps/SRE

Além dos conhecimentos específicos citados na 3, os profissionais que desejam ingressar no mercado devops, precisam ter algumas habilidades mais gerais da área de TI, já que o processo seletivo de DevOps se parece muito com um processo seletivo típico de TI, e tem basicamente as mesmas etapas.

O que difere um processo seletivo do outro são os tipos de problema que podem ser abordados em desafios técnicos mais específicos de cada área, por exemplo, em uma vaga de Engenheiro de Software uma questão técnica que pode cair é para que o candidato desenvolva uma aplicação que contenha uma API com determinados requisitos, e uma interface com uma determinada folha de estilo. Já num processo seletivo de DevOps a mesma etapa contaria com que o candidato escalasse a infra de um determinado sistema, ou que fizesse o CI da aplicação e depois fizesse o deploy em nuvem.

Dito isto, uma pessoa que deseja ingressar no mercado de trabalho como DevOps deve ter experiência não só nos conhecimentos citados na 3, mas também precisa ter conhecimento em estruturas de dados e algoritmos para a etapa de desafio técnico com problemas de maratona, noção de system design, conhecimentos em clean code/solid e padrões de projeto, complexidade do código, análise assintótica entre outros.

3.6 Roadmap de aprendizado DevOps Júnior

Baseado no RoadMap DevOps, criamos uma versão de RoadMap DevOps Júnior que contam com os aprendizados que julgamos essenciais para o perfil DevOps Júnior, todas as escolhas aqui feitas, têm como base a análise dos dados coletados anteriormente

tanto nas pesquisas quanto nos resultados, e feedbacks de profissionais que já atuam como DevOps no mercado de trabalho, abaixo podemos ver a primeira versão deste roadmap realizada pela autora:

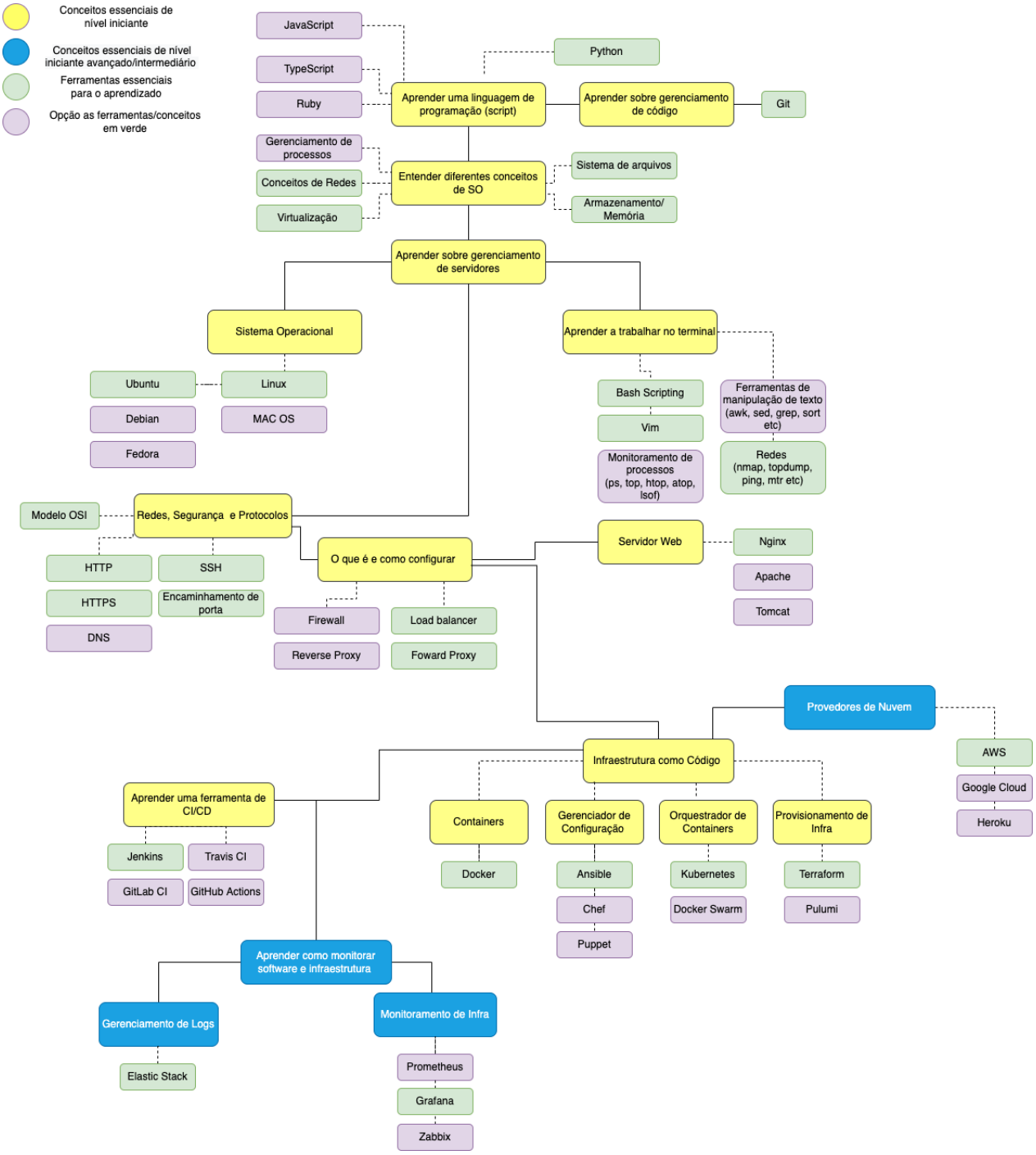

Figura 12 – Roadmap de Conhecimentos para um Devops Júnior - Versão 1

Fonte: Autora.

Após realizarmos as pesquisas para coleta de dados e mapearmos os conhecimentos principais, chegamos a primeira versão do RoadMap DevOps Júnior. Após a finalização da versão 1 (fig 12), o roadmap foi enviado para o profissional DevOps com quem realizamos

a entrevista desestruturada, alguma sugestões de mudanças foram feitas pelo mesmos, e assim surgiu a versão 2 (fig 13) do roadmap DevOps.

Nesta versão nós mudamos o título de algumas caixas, passamos a considerar gerenciamento de processos, firewall e provedores de nuvem conhecimentos obrigatórios, pois de acordo com o entrevistado esses são conhecimentos essenciais e muito usados no dia-a-dia de um profissional DevOps.

Ainda de acordo com o nosso entrevistado seria muito difícil um profissional que não tenha nenhum conhecimento com provedores de nuvem, consiga um emprego full-time, além disso também foram adicionados alguns serviços AWS, e o tema de DevSecOps, que é um tema em ascensão. Também substituimos o Heroku pelo Microsoft Azure que é mais utilizado no mercado, como podemos ver na pesquisa do StackOverflow e conforme feedback do entrevistado.

3.6.1 Explicando as escolhas das ferramentas e recomendações feitas no RoadMap

O RoadMap DevOps Júnior foi elaborado com quatro pilares que estão traduzidos nas legendas no canto superior esquerdo, são eles: Conceitos essenciais para iniciantes, bom ter, recomendação de aprendizado, opções as recomendações em verde.

O pilar de conceitos essenciais para iniciantes está sendo representado pela cor amarela, este pilar representa os principais conceitos a serem aprendidos, são conceitos "obrigatórios"para quem deseja ingressar no mercado DevOps em uma vaga CLT.

Os conceitos "Bom ter", estão representados pela cor azul e apesar de não serem obrigatórios, são conceitos muito presentes na vida de um profissional DevOps e que estão ganhando cada vez mais relevância no mercado.

Os conceitos representados pela cor verde são ferramentas e conceitos que fazem dos temas representados pela cor amarela, já a cor roxa são alternativas as recomendações em verde.

Para nos apronfundarmos no RoadMap iremos explicar o porque escolhemos cada uma das recomendações feitas, começando pela linguagem de programação. Python é a linguagem que mais apareceu como pré-requisito nas vagas de DevOps analisadas na tabela 2 tendo aparecido em 7 vagas (50% das vagas analisadas), sendo que as outras 7 vagas analisadas não especificam uma linguagem de programação, ao todo foram analisadas 14 vagas.

Em DevOps como utilizamos muitas ferramentas e scripts para realizarmos a automação do sistema,‘ python se torna uma excelente opção pois é uma linguagem com muitos recursos e diversas, bibliotecas e módulos a serem utilizados. Python também tem

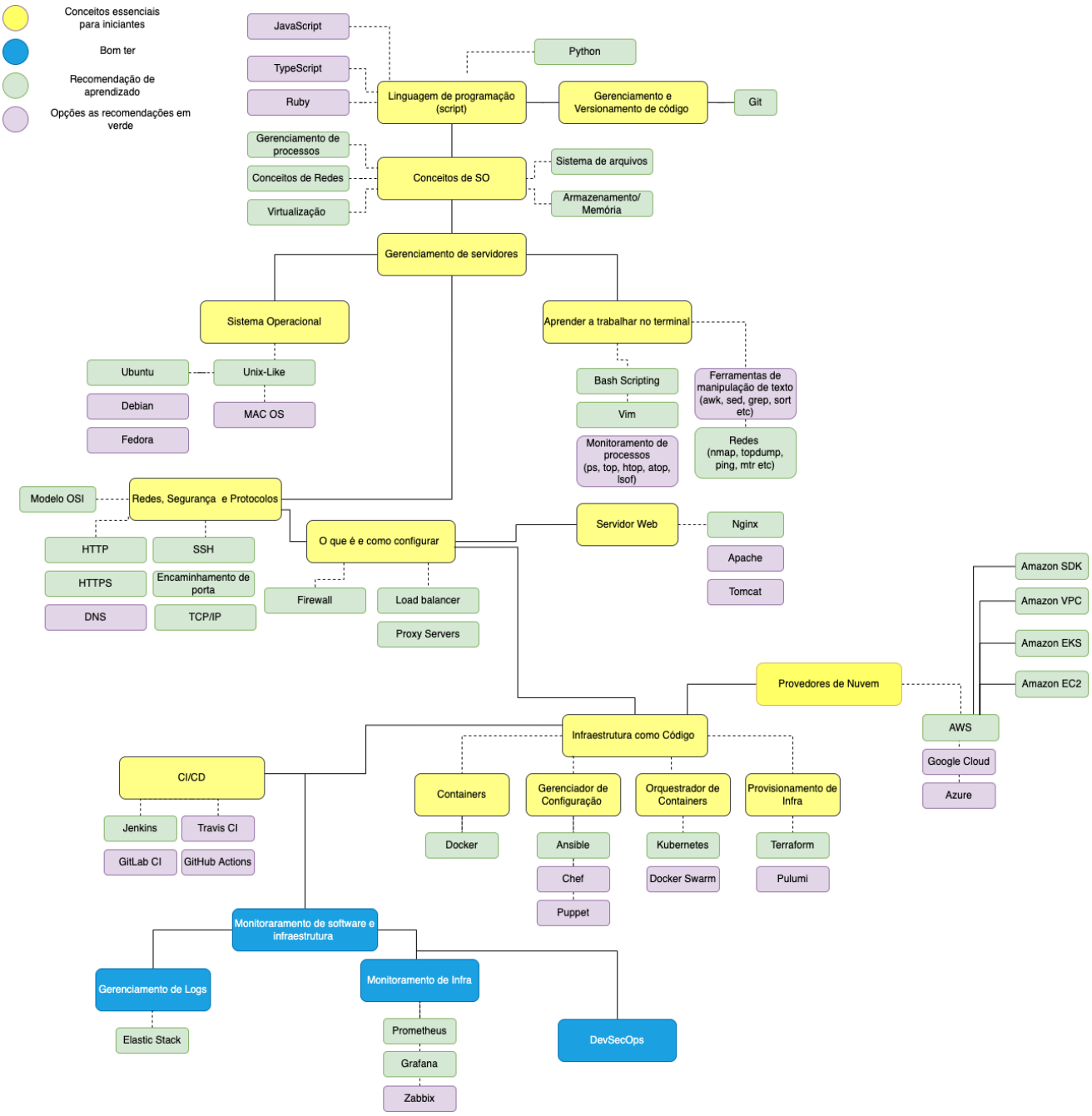

Figura 13 – Roadmap de Conhecimentos para um Devops Júnior - Versão 2

Fonte: Autora.

uma sintaxe simples e integração com muitas ferramentas, além disso, na pesquisa do (STACKOVERFLOW, 2021), a linguagem aparece como a segunda mais utilizada pelos desenvolvedores, com cerca de 48.24% dos usuários da pesquisa.

Saber gerenciamento e versionamento de código é um pré-requisito não só para quem deseja se tornar um profissional DevOps , mas para qualquer pessoa desenvolvedora. Atualmente existem diversas ferramentas que permitem com que façamos este versionamento, e o Git é sem sombra de dúvidas a mais utilizada, de acordo com a pesquisa do (STACKOVERFLOW, 2021), cerca de 93.43% dos desenvolvedores utilizam o git, e ele

aparece como pré-requisito de 9, das 14 vagas analisadas.

Para entendermos sobre gerenciamento de servidores precisamos entender mais a fundo alguns conceitos como sistemas operacionais(principalmente SO's Unix-like), e aprendermos a trabalhar no terminal, isso acontece pois a maioria dos servidores utilizam SO's Unix e estes são manipulados via linha de comando no terminal. Como sugestão escolhi o Ubuntu por ser uma distro simples de ser aprendida, porém bem completa e baseada no Debian, que é uma distro com muitos pacotes e com uma certa facilidade de instalação desses pacotes, diferente de distros baseadas no ArchLinux por exemplo.

Aprender a sobreviver no terminal exige conhecimentos de Bash Scripting, pois como já foi dito, muitos servidores são manipulados via linha de comando, e é necessário conhecermos scripts de manipulação de arquivos, manipulação de processos e principalmente de manipulação de redes.

Partindo para os conceitos de redes, segurança e protocolos, temos como principais recomendações o entendimento do modelo OSI pois assim poderemos entender as principais camadas de um sistema de software, como elas se comunicam entre si e os seus protocolos de comunicação. Entrando no modelo OSI temos a camada de aplicação, que é a camada mais próxima do usuário e é responsável pela comunicação end-to-end entre aplicações, os protocolos mais conhecidos dessa camada são os protocolos HTTP, HTTPS e SSH, por serem os mais comuns são os recomendados no roadmap.

Como a camada de aplicação trabalha em conjunto com a camada de transporte recomendamos o protocolo TCP/IP, este protocolo é uma junção do protocolo de transporte TCP com o protocolo de rede IP, este sem dúvidas é o protocolo mais utilizado da camada de transporte pois é seguro e garante que a mensagem será entregue ao remetente.

Ainda falando sobre redes, protocolos e servidores, temos alguns conceitos muito importantes como o balanceamento de carga, que é o responsável por manter os servidores estáveis conforme a carga de tráfego aumenta. Temos também os conceitos de firewall para segurança da rede e servidores proxy, além disso outro conceito importante de se entender são os servidores web, como por exemplo o Nginx que oferece alguns dos recursos citados, como o balanceamento de carga, proxy reverso entre outros. o Nginx é open source e é uma ferramenta muito robusta que suporta uma carga bem alta de conexões simultâneas, além de ser uma tecnologia mais moderna, por isso foi sugerido no roadmap.

O Apache também seria uma excelente opção de servidor web a ser aprendida, já que também é muito robusto, open source, e por ser mais antigo possui muita documentação etc. A nível de aprendizagem acredito que ambos são válidos pois o objetivo é aprender a utilizar servidores web e seus recursos, entretanto a nível de negócio eles possuem algumas diferenças que devem ser levadas em conta na hora da escolha do servidor web, mas isto não se aplica a este trabalho.

Na parte de CI/CD damos algumas sugestões de ferramentas, entretanto a recomendação principal é o Jenkis, esta ferramenta de automação open source permite que equipes monitorem tarefas repetidas e integrem mudanças de código, de forma a identificar rapidamente algum problema caso este exista. O Jenkins também possui diversos plugins para integrar diversas ferramentas como o Git, Amazon EC2 etc, por isso o recomendamos.

Em Iac temos diversos conceitos e ferramentas que são "sub-tópicos"de Iac, pois apenas com eles é possível realizarmos o provisionamento e o gerenciamento de uma infraestrutura em nuvem. Sendo assim na parte de containers recomendamos o Docker por ser uma ferramenta muito popular com cerca de 48.85% de usuários na pesquisa do (STACKOVERFLOW, 2021), além de aparecer como pré-requisito em 13 vagas de 14 mostradas na nossa pesquisa, ou seja mais de 92% das vagas encontradas exigem conhecimento de containers/Docker. Como gerenciador de configurações a recomendamos o Ansible (ANSIBLE, 2022) que é open source, escrita em Python, Shell e Ruby, permite conexões via SSH, e é uma alternativa mais leve que o Puppet por exemplo, além disso é a ferramenta de gerenciamento mais utilizada estando a frente do Puppet 1.8% , e Chef 1.35% com os seus 7.68% de usuários (STACKOVERFLOW, 2021).

Como orquestrador de containers temos o Kubernetes que é uma ferramenta open source que nos permite automatizar a implantação, gerenciamento e escalonamento de software baseado em containers, por ser uma ferramenta muito utilizada e com uma comunidade de contribuidores bem ativa, o Kubernetes pode ser integrado a diversas ferramentas além da capacidade enorme de gerenciamento de recursos, criação de containers conforme as demandas do servidor e diversas outras funcionalidades muito complexas e ricas, ele aparece na pesquisa do (STACKOVERFLOW, 2021) com 16.6% de usuários além de aparecer como pré-requisito em 13 das 14 vagas analisadas, ou seja, é um conhecimento indispensável.

Para fechar os sub-tópicos de Iac temos as ferramentas de provisionamento, a recomendação aqui foi o Terraform que também é open source, e oferece um fluxo de trabalho consistente de Interface de Linha de Comando (CLI), para ajudar no gerenciamento de serviços na nuvem. Com esta ferramenta nós podemos automatizar o provisionamento de recursos de Iac de forma a reduzir erros humanos, além disso esta ferramenta também é uma das mais utilizadas no mercado com cerca de 7.68% de usuário e por isto está aqui.

Na parte de provedores de nuvem recomendamos o AWS pois ele sem dúvidas é a plataforma mais utilizada com cerca de 54.22% de usuários, além de aparecer em 10 de 14 vagas analisadas. O provedor AWS é muito completo e tem diversos serviços disponíveis como o Amazon SDK que são conjuntos de bibliotecas para diversas linguagens de programação com o intuito de facilitar o uso do AWS Services para a linguagem do sdk, temos também o Amazon VPC que é um serviço de rede privada, temos o Amazon EKS que é

um serviço para executar e escalar aplicações do Kubernetes, temos o Amazon EC2 que é um serviço web que disponibiliza capacidade computacional segura e redimensionável na nuvem entre outros.

Como podemos ver temos uma gama bem diversa de serviços disponíveis, e recomendamos aqui os que acreditamos serem os mais comuns e os que mais serão vistos no mercado de trabalho, entretanto mais importante do que aprender minuciosamente todos os serviços listados, é entender para o que serve cada um deles e saber navegar pela sua documentação, para que através da documentação o desenvolvedor consiga resolver os problemas que surgirem.

Antes de finalizarmos a análise das ferramentas e conceitos do roadmap, vale lembrar que a pesquisa do stackoverflow é feita com desenvolvedores de todas as áreas, e que os valores das ferramentas de DevOps correspondem ao número total de usuários, e não apenas aos usuários do meio de SRE, que são os que utilizam as ferramentas citadas. Ou seja, os desenvolvedores front-end, por exemplo, não utilizam estas ferramentas no seu dia-a-dia de trabalho, mas fazem parte do valor total de usuários participantes da pesquisa.

Por fim chegamos aos conhecimentos "Bom ter", estes não são obrigatórios para um iniciante mas estão muito presentes no mercado de trabalho e são conhecimentos relevantes para quem deseja seguir na área.

A parte de monitoramento e gerenciamento de logs vai estar presente na maioria (se não em todas) as empresas que tem Iac, sendo assim é um conhecimento "Nice to have", aqui para o gerenciamento de logs recomendamos o Elastic Stack, que é composto por alguns serviços opens source como o Elasticsearch, Logstash e Kibana, cada uma destas ferramentas possuem funcionalidades específicas, e o conjunto destas nos permitem agregar logs de todas as aplicações, análises de segurança e outros.

Já as ferramentas de monitoramento aqui sugeridas são, o Prometheus, que é uma ferramenta de monitoramento que coleta métricas do sistema em determinados intervalos de tempo e as grava num banco de dados. E recomendamos também o Grafana, que é uma plataforma de visualização e análise de métricas, ela suporta integração com várias ferramentas de coleta de métricas como por exemplo o Prometheus, para isso o grafana se conecta ao banco de dados do prometheus e nos mostra as métricas de forma amigável em forma de gráficos.

Por último mas não menos importante, temos o conceito de DevSecOps, que basicamente é a abreviação de Desenvolvimento, Segurança e Operações, essa abordagem é muito parecida a abordagem DevOps tradicional, porém com foco em segurança em todas as etapas do desenvolvimento de software. Aqui a responsabilidade de gerar um produto de software seguro, é compartilhada entre as equipes de Desenvolvimento, Segurança e

Operações, este conceito foi acrescentado ao roadmap pois, empresas que se preocupam com qualidade de software estão cada vez mais adotando esta prática, então vale a pena conhecer mais sobre (EDUCATION, 2020).

3.6.2 Organizaço Guia DevOps Júnior no GitHub

Com o intuito de facilitar o acesso aos conhecimentos listados no RoadMap DevOps Júnior, foi criada uma organização na plataforma GitHub, onde reunimos diversos repositórios voltados para temáticas do roadmap, esses temas vão desde a conceitualização do que é DevOps , até como se preparar para entrevistas de emprego em big techs, além disso, um documento readme foi criado pela autora, e nele passamos por todos os tópicos do roadmap explicando cada conceito/ferramenta, e indicando materiais para o aprendizado destes.

Para que pudéssemos diversificar a forma de aprendizado e atender um público maior, foram trazidos diversos formatos de aprendizado, como vídeos, vídeo-aulas, documentações oficiais, tutoriais práticos, publicações em formato de blogs, repositórios no github, livros, jogos interativos etc, sendo alguns destes de desenvolvimento autoral. Além disso, também foram sugeridos conteúdos publicados tanto em inglês, quanto em português, para que não haja uma barreira linguística na hora do aprendizado.

Contudo para garantir a qualidade dos materiais que foram recomendados na organização, definimos alguns critérios de qualidade objetivos e subjetivos, como:

- **Repositórios no GitHub:** Para definir se um repositório é confiável, utilizamos alguas métricas do próprio github como número de forks, e número de estrelas, isto porque, como o nosso objetivo é facilitar o aprendizado repositórios utilizados e bem avaliados por muitas pessoas se tornam mais confiáveis do que os que não são.

- **Publicações do Medium:** O Medium é uma plataforma online, que tem um formato parecido com um blog, onde qualquer pessoas pode se cadastrar e publicar sobre temas diversos, esta plataforma se tornou muito popular entre desenvolvedores pois nela podemos encontrar diversos tutoriais e publicações muito didáticas, feitas por pessoa da área de tecnologia e que são uma alternativa de aprendizado para quem tem muita dificuldade em ler documentação oficial, ou quer saber como resolver um problema de forma mais objetiva. Nessa plataforma, caso você esteja logado, é possível "aplaudir"uma publicação, como se fosse uma curtida do Facebook, optamos por escolher publicações com mais de 50 aplausos, pois quanto mais pessoas aprovarem aquele conteúdo melhor, este conteúdo é um tipo de literatura cinzenta.

- **Publicações de grandes organizações tecnológicas:** Outro conteúdo que foi muito sugerido foram leituras de publicações feitas por grandes organizações tecnológicas como a Amazon, Digital Ocean, Red Hat etc. Essas organizações costumam fazer diversas publicações sobre assuntos de tecnologia e são uma excelente fonte de informação, pois além de serem confiáveis muitas vezes elas nos explicam conceitos mais gerais como por exemplo o que é devops, o que é gerenciamento de configurações, o que é Iac, e diversos outro, para depois nos apresentarem suas soluções para os temas abordados. A organização RedHat por exemplo tem uma publicação falando sobre gerenciamento de configurações, e nos recomendando sua ferramenta gerenciadora de configurações que é o Ansible.

- **Vídeo Aula:** Alguns cursos de vídeo aulas foram recomendados ao longo da documentação do roadmap, todos eles são gratuitos e são de plataformas de aprendizado online que já são bem consolidadas e conhecidas no meio tecnológico como a Udemy, que é uma plataforma de Ead com cursos de diversas áreas, cursos com vídeo aulas são uma excelente opção para quem tem dificuldade de aprender com leitura.

- **Vídeos abordando temas correlatos:** Outra recomendação feita foram vídeos no youtube sobre os assuntos abordados no roadmap, esses vídeos são produzidos por criadores de conteúdo de tecnologia que também atuam na área, como por exemplo o canal do DioLinux, o canal do Fillipe Deschamps, Gustavo Guanabara entre outros. Os canais citados são grandes e consolidados e ensinam tecnologia de um jeito descontraído e amigável, além disso, temos também alguns vídeos do projeto Big Open Source Sister, que é um projeto criado pela professora Carla Rocha da Universidade de Brasília e tem o intuito de ensinar programação para mulheres.

- **Documentação oficial:** A documentação oficial das ferramentas é o melhor lugar para se aprender, quem deseja se tornar um desenvolvedor precisa aprender a ler e a trabalhar com essas documentações. A maioria delas tem tutorias de "Get started"para introduzir as pessoas a ferramenta, entretanto no mercado de trabalho nem sempre você irá conseguir tutoriais te ensinando a usar uma ferramenta e o seu principal recurso será a documentação oficial, outro detalhe é que na maior parte das vezes essas documentações estão em inglês, então é fundamental que você aprenda a ler nessa língua.

- **Tutoriais práticos:** Outro método de aprendizado que costuma ser muito efetivo são os tutoriais práticos, existem diversos deles pela internet e eles são bons pois são muito focados na parte prática e ensinam através dela. Recomendamos tutoriais em diversos tópicos da documentação do roadmap, e estes foram escolhidos por mim após a execução prática dos mesmos, aqui levei em conta o sucesso que eu obtive na realização e entendimento do que estava sendo ensinado, porém levei em conta

a minha experiência e escolhi materiais que acredito que também serão de fácil entendimento para quem tem pouca ou nenhuma experiência. Vale ressaltar que os tutoriais são muito úteis, porém não focam muito nos conceitos teóricos, portanto na hora de executá-los o leitor deve certifica-se de entender como, e por que as coisas estão sendo feitas, e não apenas sair executando o passo-a-passo. Apenas um tutorial é de autoria própria, neste caso o do Docker, este foi feito com base nas documentações oficiais do Docker.

Por fim, vale ressaltar que todo o material recomendado foi validado pela autora tanto com base em critérios pessoais, quanto com base em critérios mais objetivos e quantitativos.

A organização se encontra no seguinte link Guia DevOps Iniciante: https://github.com/Guia-Devops-Iniciante.

4 Considerações Finais

O ingresso de profissionais DevOps Júnior no mercado de trabalho, envolve vários desafios que vão desde a dificuldade em se encontrar materiais que condicionem o caminho que esses profissionais devem seguir para se prepararem para o mercado de trabalho, até a escassez de vagas para profissionais desta senioridade, já que, analisando todos os dados e informações coletadas, foi possível observar que, mesmo com a alta demanda do mercado por profissionais DevOps , a maior parte das vagas encontradas são para profissionais com maior senioridade e que já tenham um background forte sobre os conhecimento abordados neste trabalho. Acredito que parte disto se dá pela complexidade das ferramentas a serem aprendidas, e pela grande quantidade de conhecimentos a serem absolvidos, então acaba sendo mais fácil que um profissional com mais tempo de mercado e com mais experiência em desenvolvimento tenha um pouco mais de facilidade de aprender os requisitos de vagas DevOps , do que pessoas que acabaram de começar na programação e ainda irão precisar desenvolver diversas skills que são fundamentais para qualquer desenvolvedor.

Além disso, através da nossa análise qualitativa pudemos observar que, ao contrário dos processos seletivos de desenvolvimento tradicionais, também não existem muitos materiais preparatórios para o processo seletivo específico de DevOps , e isso acaba sendo um fator que dificulta o ingresso de novos profissionais nesta área.

Com isto chegamos a conclusão de que existe uma demanda por conteúdos técnicos e teóricos para preparar esses profissionais tecnicamente, mas também por conteúdos que ajudem a orientar esses profissionais por qual caminho eles devem seguir, para ingressarem no mercado como profissionais DevOps .

Sendo assim o objetivo deste trabalho foi criar um conteúdo informativo sobre como formar um profissional DevOps , tentando ser o mais didático e claro o possível sobre quais os conhecimentos necessários, e qual estratégia deve ser traçada para chegarmos a esses conhecimentos. Com isto temos o RoadMap Devops Júnior que mostra claramente quais os conceitos e ferramentas devem ser aprendidos, e uma ordem que otimize esse aprendizado, além disso criamos ainda uma organização no GitHub com dicas e tutoriais de todos os conhecimentos citados no roadmap, com o intuito de facilitar ainda mais o acesso a esses conhecimentos.

4.1 Trabalhos Futuros

Para dar continuidade a este trabalho deixaremos o código da organização no GitHub aberto, para que qualquer pessoa que queira contribuir possa fazê-lo, e ajudar

ainda mais a enriquecer o material que lá se encontra. Além disso o nosso roadmap e todo o tema deste trabalho poderia se tornar uma publicação no Medium, que é uma plataforma de publicações online muito popular, e onde muitos desenvolvedores buscam novas informações sobre tecnologias, ferramentas etc, isto seria uma forma de espalhar o conhecimento aqui centralizadom já que diversas pessoas poderiam acessá-lo com mais facilidade do que buscando no GitHub.

Referências

Scale-up e scale-out, qual a diferença? Quando e como utilizar cada um deles? Disponível em: <https://somosagility.com.br/scale-up-e-scale-out-qual-a-diferenca-quando-e-como-utilizar-cada-um-deles/>. Citado na página 32.

Entrevistas na Amazon. Disponível em: <https://www.amazon.jobs/pt/landing_pages/interviewing-at-amazon>. Citado na página 19.

The Roadmap to Becoming a DevOps Professional — From Server to Serverless. Disponível em: <https://faun.pub/the-roadmap-to-become-a-devops-dude-from-server-to-serverless-dd97420f640e>. Citado na página 35.

Ansible. Disponível em: <https://www.ansible.com/>. Citado na página 52.

Docker Architecture. Disponível em: <https://www.aquasec.com/cloud-native-academy/docker-container/docker-architecture/>. Citado na página 27.

BEYER CHRIS JONES, J. P. . N. M. B. *Site Reliability Engineering: How Google Runs Production Systems*. [S.l.: s.n.], 2016. Citado 3 vezes nas páginas 21, 22 e 34.

RFC 1122 - Requirements for Internet Hosts – Communication Layers. Disponível em: <https://datatracker.ietf.org/doc/html/rfc1122#page-6>. Citado na página 28.

A Look Ahead: The DevOps Hiring Demand Increase. Disponível em: <https://www.harrisonclarke.com/devops-sre-recruiting-blog/a-look-ahead-the-devops-hiring-increase>. Citado na página 19.

Continuous Integration: A "Typical"Process. Disponível em: <https://developers.redhat.com/blog/2017/09/06/continuous-integration-a-typical-process>. Citado na página 24.

Docker Overview. Disponível em: <https://docs.docker.com/get-started/overview/>. Citado 2 vezes nas páginas 27 e 29.

Use bridge networks. Disponível em: <https://docs.docker.com/network/bridge/>. Citado na página 29.

Use host networks. Disponível em: <https://docs.docker.com/network/host/>. Citado na página 29.

DevOps. Disponível em: <https://github.com/raycad/devops-roadmap>. Citado na página 23.

DYCK, A.; PENNERS, R.; LICHTER, H. Towards definitions for release engineering and devops. In: *2015 IEEE/ACM 3rd International Workshop on Release Engineering*. [S.l.: s.n.], 2015. p. 3–3. Citado na página 21.

DevSecOps. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/devsecops>. Citado na página 54.

What is Infrastructure as Code? Disponível em: <https://blog.stackpath.com/infrastructure-as-code-explainer/>. Citado na página 33.

Continuous Integration. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>. Citado na página 24.

Fundamentos de Roteamento para Provedores. Disponível em: <https://wiki.brasilpeeringforum.org/w/Fundamentos_de_Roteamento_para_Provedores>. Citado na página 29.

2021 Upskilling Enterprise DevOps Skills Report. Disponível em: <https://page.gitlab.com/2021-DevOps-Skills-Report.html?utm_medium=cpc&utm_source=google&utm_campaign=devopsgtm_emea_pr_rsa_nb_phrase&utm_content=2021-doi-devops-upskilling-report_digital_x-pr_english_&&utm_term=%2Bdevops&_bt=525693523601&_bk=%2Bdevops&_bm=b&_bn=g&_bg=82528767276&gclid=CjwKCAjwgviIBhBkEiwA10D2jyVyfhPKRkk5ib-G0Vwp5_GLXcvC9dnEGD36o2li1nn5nf1RO59wlBoCw_YQAvD_BwE>. Citado na página 19.

KUROSE, J.; ROSS, K. *Redes de computadores e a Internet - Uma abordagem top-down*. [S.l.: s.n.], 2014. Citado na página 28.

LEITE CARLA ROCHA, F. K. D. M. P. M. L. A survey of devops concepts and challenges. ACM Computing Surveys, Vol. 52, No. 6, Article 127., 2019. Citado 2 vezes nas páginas 22 e 35.

O que é container? Como surgiu? Disponível em: <https://www.linuxtips.io/blogs/novidades/o-que-e-container-como-surgiu-e-como-isso-ira-te-ajudar>. Citado na página 26.

LIU, C.-Y. et al. Vertical/horizontal resource scaling mechanism for federated clouds. In: *2014 International Conference on Information Science Applications (ICISA)*. [S.l.: s.n.], 2014. p. 1–4. Citado na página 31.

MELNIK, G.; KRUCHTEN, P.; POPPENDIECK, M. (Ed.). *Agile Development Conference, AGILE 2008, Toronto, Canada, 4-8 August 2008*. IEEE Computer Society, 2008. ISBN 978-0-7695-3321-6. Disponível em: <https://ieeexplore.ieee.org/xpl/conhome/4599439/proceeding>. Citado 2 vezes nas páginas 19 e 21.

Guia de Infraestrutura de TI. Disponível em: <https://medium.com/sysadminas/guia-de-infraestrutura-de-ti-30543bfe9922>. Citado na página 35.

Google. Disponível em: <https://careers.google.com/how-we-hire/#step-self-reflection>. Citado na página 39.

CI/CD: integração e entrega contínuas. Disponível em: <https://www.redhat.com/pt-br/topics/devops/what-is-ci-cd>. Citado na página 24.

Cloud Computing. Disponível em: <https://www.redhat.com/pt-br/topics/cloud>. Citado na página 30.

O que são provedores de serviços de nuvem? Disponível em: <redhat.com/pt-br/topics/cloud-computing/what-are-cloud-providers>. Citado na página 31.

O que é devops? Disponível em: <https://www.redhat.com/pt-br/topics/devops>. Citado na página 22.

O que é um container Linux? Disponível em: <https://www.redhat.com/pt-br/topics/containers/whats-a-linux-container>. Citado 2 vezes nas páginas 26 e 27.

Gerenciamento de configuração. Disponível em: <https://www.redhat.com/pt-br/topics/automation/what-is-configuration-management>. Citado na página 33.

O que é Platform-as-a-service (PaaS)? Disponível em: <https://www.redhat.com/pt-br/topics/cloud-computing/what-is-paas>. Citado na página 30.

The state of Linux in the public cloud for enterprises. Disponível em: <https://www.redhat.com/en/resources/state-of-linux-in-public-cloud-for-enterprises>. Citado na página 26.

Design de Sistemas Distribuídos — Escalonamento Vertical e Horizontal. Disponível em: <https://medium.com/xp-inc/design-de-sistemas-distribu%C3%ADdos-escalonamento-vertical-e-escalonamento-horizontal-a162a2c66cbe#:~:text=Entende%2Dse%20por%20escalonamento%20horizontal,no%20design%20de%20sistemas%20distribu%C3%ADdos.> Citado na página 31.

SHAH, J.; DUBARIA, D.; WIDHALM, J. A survey of devops tools for networking. In: *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*. [S.l.: s.n.], 2018. p. 185–188. Citado na página 22.

SHAHIN, M.; BABAR, M. A.; ZHU, L. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. *IEEE Access*, IEEE, v. 5, p. 3909–3943, 2017. Citado na página 23.

StackOverflow Developer Survey 2021. Disponível em: <https://insights.stackoverflow.com/survey/2021#developer-profile-key-territories>. Citado 6 vezes nas páginas 25, 42, 43, 45, 50 e 52.

TANENBAUM, A. S. W. A. S. *Sistemas Operacionais: Projeto e Implementação*. [S.l.: s.n.], 2008. Citado na página 25.

What is Terraform? Disponível em: <https://www.terraform.io/intro>. Citado na página 32.

Iptables. Disponível em: <https://pt.wikipedia.org/wiki/Iptables>. Citado na página 28.

Linux. Disponível em: <https://pt.wikipedia.org/wiki/Linux#Distribui%C3%A7%C3%B5es>. Citado na página 25.

Modelo OSI. Disponível em: <https://pt.wikipedia.org/wiki/Modelo_OSI>. Citado na página 28.