

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NO ESTÁGIO

**RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE
TECNOLOGIAS DEVOPS EM UMA EMPRESA DE TECNOLOGIA DA
INFORMAÇÃO**

THIAGO CÉSAR RODRIGUES DA MATA

IPORÁ, GO

2024

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL GOIANO – CAMPUS IPORÁ
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NO ESTÁGIO:
ESTUDO DE TECNOLOGIAS DEVOPS EM UMA EMPRESA DE
TECNOLOGIA DA INFORMAÇÃO

THIAGO CÉSAR RODRIGUES DA MATA

LUCIANA RECart CARDOSO

Orientadora

Relatório de Atividades Desenvolvidas apresentado ao Instituto Federal Goiano – *Campus* Iporá, como requisito parcial para conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

IPORÁ, GO

Fevereiro/2024

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

MM425r Mata, Thiago Cesar Rodrigues
RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE
TECNOLOGIAS DEVOPS EM UMA EMPRESA DE TECNOLOGIA DA
INFORMAÇÃO / Thiago Cesar Rodrigues Mata;
orientadora Luciana Recart Cardoso. -- Iporá, 2024.
37 p.

TCC (Graduação em Tecnologia em Análise e
Desenvolvimento de Sistemas) -- Instituto Federal
Goiano, Campus Iporá, 2024.

1. DevSecOps. I. Recart Cardoso, Luciana ,
orient. II. Título.



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

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica (assinale com X)

- Tese
- Dissertação
- Monografia – Especialização
- Artigo - Especialização
- TCC - Graduação (Estágio)
- Artigo Científico
- Capítulo de Livro
- Livro
- Trabalho Apresentado em Evento
- Produção técnica. Qual: _____

Nome Completo do Autor: Thiago César Rodrigues da Mata

Curso: Tecnologia em Análise e Desenvolvimento de Sistemas

Título do Trabalho:

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE TECNOLOGIAS DEVOPS EM UMA EMPRESA DE TECNOLOGIA DA INFORMAÇÃO

Restrições de Acesso ao Documento [Preenchimento obrigatório]

Documento confidencial: Não Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: 04/03/2024

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. O documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;

2. Obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

3. Cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Iporá, 1 de março de 2024

Thiago César Rodrigues da Mata

Assinado eletronicamente pelo o Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Luciana Recart Cardoso

Assinatura eletrônica do(a) orientador(a)

Documento assinado eletronicamente por:

- Thiago César Rodrigues da Mata, 2018105210430177 - Discente, em 01/03/2024 15:36:00.
- Luciana Recart Cardoso, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/03/2024 15:33:28.

Este documento foi emitido pelo SUAP em 01/03/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 578785

Código de Autenticação: 150c31b97b



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000

(64) 3674-0400



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

Ata nº 35/2024 - GE-IP/CMPIPR/IFGOIANO

**ATA DA SESSÃO DE JULGAMENTO DO TRABALHO DE CURSO
DE THIAGO CÉSAR RODRIGUES DA MATA**

Aos vinte e nove dias, do mês de fevereiro de dois mil e vinte e quatro, às vinte e uma horas e trinta minutos, em sessão pública, a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar o trabalho de curso intitulado “**RELATÓRIO DE ATIVIDADES DESENVOLVIDAS: ESTUDO DE TECNOLOGIAS DEVOPS EM UMA EMPRESA DE TECNOLOGIA DA INFORMAÇÃO**”, apresentada pelo acadêmico Thiago César Rodrigues da Mata, como parte dos requisitos necessários à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas. A banca examinadora foi presidida pela orientadora do trabalho de curso, professora **Ma. Luciana Recart Cardoso**, tendo como membros avaliadores: o professor **Dr. Newarney Torrezão da Costa** e o professor **Me. Wesley Flávio de Miranda**. Aberta a sessão, o acadêmico expôs seu trabalho. Em seguida, foi arguida pelos membros da banca e:

() tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, sem restrições.

(X) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **aprovação** do acadêmico, **condicionada a satisfazer as exigências** listadas na Folha de Modificação de Trabalho de Curso anexa à presente ata, no prazo máximo de 80 (oitenta) dias, a contar da presente data, ficando o professor orientador responsável por atestar o cumprimento dessas exigências.

() não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de seu trabalho de curso, a banca conclui pela **reprovação** do acadêmico.

Conforme avaliação individual de cada membro da banca, será atribuída a nota nove (9,0) para fins de registro em histórico acadêmico.

Os trabalhos foram encerrados às 22 horas e 13 minutos. Nos termos do Regulamento do Trabalho de Curso do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal Goiano – Campus Iporá, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Assinado eletronicamente

Luciana Recart Cardoso, Ma. Orientadora

Newarney Torrezão da Costa, Dr.

Wesley Flávio de Miranda, Me.

Documento assinado eletronicamente por:

- Newarney Torreza da Costa, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/03/2024 12:27:33.
- Wesley Flavio de Miranda, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/03/2024 11:00:27.
- Luciana Recart Cardoso, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 29/02/2024 22:28:27.

Este documento foi emitido pelo SUAP em 29/02/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 578339

Código de Autenticação: 03316a6d04



INSTITUTO FEDERAL GOIANO

Campus Iporá

Av. Oeste, Parque União, 350, Parque União, IPORA / GO, CEP 76.200-000

(64) 3674-0400

SUMÁRIO

1 - INTRODUÇÃO.....	1
2 – CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO.....	3
2.1-Empresa.....	3
2.2-Dados do Estágio.....	3
3 – ATIVIDADES DESENVOLVIDAS.....	6
3.1 – Entendendo o Linux.....	7
3.2 – Certificação LPI Linux Essentials.....	7
3.3 – Exercício Linux.....	9
3.4 – Exercício Kubernetes.....	21
4 – CONSIDERAÇÕES FINAIS.....	36
5 – REFERÊNCIAS BIBLIOGRÁFICAS.....	37

1 - INTRODUÇÃO

O estágio pode ser elemento crucial na formação acadêmica e profissional de um estudante, oferecendo uma oportunidade única de adquirir experiência prática e habilidades fundamentais para o mercado de trabalho. A Lei do Estágio (Lei 11.788, 2008) destaca a importância de ter a orientação de um supervisor experiente por parte da instituição de ensino e por parte da empresa, que ajude no desenvolvimento de competências comportamentais importantes, como relacionamento interpessoal e gestão do tempo. Essa lei, estabelece os direitos e deveres tanto dos estagiários quanto das instituições de ensino e empresas concedentes de estágio. (Brasil, 2008).

Entre seus principais pontos, a lei define as condições para a realização do estágio, como carga horária, duração, obrigações das partes envolvidas, concessão de bolsa-auxílio e benefícios, além de estabelecer diretrizes para a elaboração do termo de compromisso de estágio. A legislação visa garantir uma experiência de estágio adequada e compatível com os objetivos de formação profissional do estudante, protegendo seus direitos e assegurando uma relação de aprendizado e desenvolvimento mútuo entre o estagiário e a empresa. (Brasil, 2008).

Durante o período em que estive envolvido na organização *Compass UOL*, tive a oportunidade de adquirir experiência prática, consolidar conhecimentos teóricos e desenvolver habilidades profissionais essenciais. Este relatório é fundamental para atender aos critérios de conclusão do Curso Tecnologia em Análise e Desenvolvimento de Sistemas. Neste documento apresento em detalhes as atividades realizadas e minha jornada de aprendizado e conhecimento durante o período de agosto a novembro do ano de 2021 na empresa *Compass UOL*.

Ao longo deste relatório, consta uma descrição detalhada das atividades e responsabilidades que desempenhei, bem como uma análise crítica das experiências vivenciadas. Abordo as contribuições que pude oferecer à *Compass UOL* e como essas contribuições estão alinhadas com os objetivos da organização.

Este estágio foi uma oportunidade enriquecedora que me permitiu explorar de forma prática a área de *DevSecOps* e compreender a dinâmica do mercado de trabalho. A estrutura deste relatório de estágio inclui uma análise do contexto da *Compass UOL*, uma descrição das minhas atividades e responsabilidades, uma revisão crítica das experiências vividas e um resumo das principais lições aprendidas.

Concluo com uma análise sobre como essa vivência teve um impacto no meu futuro profissional e acadêmico. Espero que este relatório forneça uma visão abrangente do meu estágio na *Compass* UOL e demonstre o meu comprometimento com a busca contínua de excelência e crescimento na minha área de atuação.

2 – CARACTERIZAÇÃO DA EMPRESA E DADOS DO ESTÁGIO

Neste capítulo abordo informações sobre o estágio, apresento uma descrição da organização, introduzindo o conceito de *DevSecOps* e destacando as técnicas e metodologias empregadas ao longo do período de estágio.

2.1-Empresa

A *Compass* UOL teve início em 1995 em Passo Fundo, Rio Grande do Sul, em 2020 a administração mudou para a empresa UOLDIVEO, que faz parte do Grupo Folha, quando passou a se chamar Compasso UOL. Em 2021, houve outra mudança de nome com o propósito de se tornar uma empresa internacional e competir no mercado global, passando a se chamar *Compass* UOL.

A empresa presta serviços de tecnologia a diversas empresas como, Natura, Esfera, Gol, Globo, Livello, RaiaDrogasil, PagSeguro, UOL, Renner, PagBank, dentre várias outras. Oferecendo serviços terceirizados nas áreas de infraestrutura Cloud Computing, operações, pagamentos, telecomunicação, Digital Commerce, *blockchain* e outros. A *Compass* conta com mais de 6000 funcionários, com presença global e escritórios no Brasil, nos EUA e na Europa.

2.2-Dados do Estágio

Durante o período do estágio, que transcorreu entre 6 de agosto de 2021 e 12 de novembro de 2021, tive a oportunidade de realizar minhas atividades remotamente na empresa *Compass* UOL. A supervisão acadêmica do estágio foi conduzida pela professora Luciana Recart Cardoso, enquanto, no ambiente da empresa, as instrutoras Adriana Rodrigues e Andreza Santarém desempenharam papéis essenciais ao acompanhar e monitorar minhas atividades.

O estágio seguiu uma carga horária de 240 horas totais, divididas de 4 horas de estudos de segunda a sexta-feira, representando 20 horas semanais, período que proporcionou um ambiente de aprendizado valioso e um profundo envolvimento com as operações da empresa *Compass* UOL.

2.3-Metodologias

No início do estágio foi apresentado o funcionamento das dinâmicas de estudos e atividades a serem executadas. A realização do estágio seguiu uma metodologia ágil chamada *Scrum*, a qual foi o primeiro tema que estudamos no estágio.

O *Scrum* é um framework ágil de desenvolvimento de software, introduzido por Hirotaka Takeuchi e Ikujiro Nonaka em 1986, como uma estratégia flexível e holística onde equipes trabalham em unidade para alcançar objetivos comuns. Emprega processos de tomada de decisão em tempo real, exigindo equipes especializadas em auto-gerenciamento, comunicação e decisão. Essa metodologia pode ser aplicada a vários projetos, mas é amplamente utilizada no desenvolvimento de software. (SACHDEVA, 2016).

Uma de suas principais características é fornecer *feedback* contínuo para melhorar resultados, destacando a melhoria contínua do processo. As práticas do *Scrum* incluem *Backlogs* de Produto e *Sprint*, Reuniões de Planejamento de *Sprint* e a construção do produto em iterações fixas chamadas *Sprints*, enfatizando a importância das iterações curtas e da auto-organização da equipe. (SACHDEVA, 2016).

De acordo com essa metodologia apresentada, as instrutoras do estágio organizavam reuniões diárias de 15 minutos conhecidas também como *daily*, onde eram feitas as perguntas do *Scrum*. Com base nas respostas, eram alinhadas as dúvidas e por vezes era solicitada ajuda ao time para solucionar os problemas relatados, não havendo impedimentos era dada continuidade no estudo e nas atividades.

As atividades do estágio eram realizadas por meio de máquinas virtuais instaladas em um computador, e um dos requisitos impostos era a execução das atividades em um ambiente Linux com a distribuição *Ubuntu*, assim fixando os conteúdos de estudo a serem apresentados.

O Linux, um sistema operacional criado por Linus Torvalds em 1991, é distribuído gratuitamente pela Internet como software livre, sob a licença GPL (*General Public License*). Sendo de código aberto, seu desenvolvimento é realizado por milhares de pessoas ao redor do mundo, cada uma contribuindo para partes específicas do *kernel*, Linus Torvalds ainda coordena esse processo. O sistema oferece suporte eficiente e rápido, superando muitos programas comerciais. Há uma ampla gama de consultores e empresas especializadas em suporte e treinamento para o Linux. A comunidade Linux também desempenha um papel crucial no suporte, onde usuários podem se inscrever em listas de discussão para obter ajuda

ou relatar problemas, contando com a colaboração de centenas de usuários para solucionar questões ou reportá-las para correção. (EVANGÉLICA, 2018).

DevSecOps é uma abordagem que visa integrar avaliações de segurança em todas as fases do ciclo de vida do desenvolvimento de *software*. Essa metodologia promove a colaboração entre programadores, especialistas em segurança e equipes operacionais, com o objetivo de criar software robusto e protegido. Uma característica fundamental do *DevSecOps* é a mudança cultural, onde a segurança se torna uma responsabilidade compartilhada por todos os envolvidos na criação do software. Antigamente, a segurança era delegada a uma equipe isolada que atuava no final do ciclo de desenvolvimento, o que não se adequava aos ciclos rápidos e frequentes do *DevOps*. (REDHAT, 2023).

No entanto, no contexto colaborativo do *DevOps*, a segurança é integrada desde o início, resultando na necessidade do termo "*DevSecOps*" para enfatizar essa integração. *DevSecOps* envolve pensar na segurança desde o início do desenvolvimento, automatizando algumas medidas de segurança para não impactar o fluxo de trabalho do *DevOps*. A seleção de ferramentas adequadas e a integração do trabalho das equipes de segurança desde o início são essenciais para uma segurança eficaz no contexto do *DevOps*. (REDHAT, 2023).

3 – ATIVIDADES DESENVOLVIDAS

Aqui apresento as atividades realizadas no período de estágio na *Compass UOL*, assim como assuntos da cultura *DevSecOps*. Durante o estágio, destaco meu envolvimento em projetos multidisciplinares, onde participei de equipes usando a metodologia *Scrum*. Participei ativamente na área de *DevSecOps*, completando cursos online na plataforma Alura. As atividades estão detalhadas nos quadros a serem apresentados no decorrer deste relatório, ressaltando a aplicação prática dos conhecimentos adquiridos em situações reais. No Quadro 1 apresento os cursos necessários para completar a sprint sobre Linux.

Quadro 1 - Primeira lista de estudos da Alura.

CURSO	TEMA	CARGA HORÁRIA
Entendendo o <i>Linux</i>	<i>Linux I - Conhecendo e utilizando o terminal</i>	4h
Entendendo o <i>Linux</i>	<i>Linux II - Programas, processos e pacotes</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Evolution and Distributions</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Open-Source Software and Licensing</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Command Line Basics</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Using the command line to get help</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Directories and Listing Files and managing files</i>	8h
Certificação <i>LPI Linux Essentials</i>	<i>Archiving Files on the Command Line</i>	5h
Certificação <i>LPI Linux Essentials</i>	<i>Searching and Extracting Data from Files</i>	4h
Certificação <i>LPI Linux Essentials</i>	<i>Turning Commands into a Script</i>	5h
Certificação <i>LPI Linux Essentials</i>	<i>Understanding Computer Hardware</i>	4h
Certificação <i>LPI Linux Essentials</i>	<i>Where Data is Stored</i>	4h
Certificação <i>LPI Linux Essentials</i>	<i>Your Computer on the Network</i>	4h
Certificação <i>LPI Linux Essentials</i>	<i>Security and File Permissions</i>	4h

Fonte: Autor.

3.1 – Entendendo o Linux

Este módulo foi uma introdução abrangente ao sistema operacional Linux, focando em dois tópicos principais “Conhecendo e utilizando o terminal” e “Programas, processos e pacotes”, visando capacitar e desenvolver uma compreensão sólida sobre o uso de terminal e conhecimento detalhado sobre programas, processos e pacotes no sistema operacional.

- **Conhecendo e Utilizando o Terminal:** aprendemos neste curso a exploração das funcionalidades e comandos essenciais do terminal no ambiente Linux. Vimos também instruções práticas sobre navegação de diretórios, manipulação de arquivos e administração de permissões e a demonstração de técnicas avançadas para aumentar a eficiência no uso do terminal.
- **Programas, Processos e Pacotes:** este curso teve como objetivo a compreensão dos conceitos fundamentais relacionados a programas, processos e pacotes no contexto do Linux. Foi visto como realizar uma análise detalhada dos processos em execução no sistema operacional, discussão sobre a gestão e instalação de pacotes, abordando ferramentas como apt e yum.

3.2 – Certificação LPI Linux Essentials

O curso de certificação *Linux Professional Institute* (LPI) visava instruir e preparar para a certificação LPI, que se trata de uma certificação com reconhecimento internacional para profissionais de TI que demonstram habilidades avançadas em ambientes Linux. A certificação LPI é dividida em vários níveis, cobrindo tópicos como administração do sistema, segurança, rede e desenvolvimento. Neste curso pude ver o abrangente mundo de sistemas operacionais baseados no *Kernel* Linux como o Linux Mint, Centos, Android, Debian, dentre outros. O curso foi dividido em vários tópicos.

- ***Evolution and Distributions:*** neste tópico foi vista a evolução histórica do sistema operacional Linux e a diversidade de distribuições disponíveis, e com essa variedade de distribuições vimos as particularidades de algumas delas, proporcionando uma base sólida para compreender o ecossistema Linux.
- ***Open Source Software and Licensing:*** aqui vimos os princípios fundamentais do software de código aberto e as questões relacionadas a licenciamento, compreendendo a importância da transparência, colaboração e distribuição livre de código-fonte.

Foram apresentados os diferentes tipos de licenças de software de código aberto, examinando como essas licenças afetam a utilização, distribuição e modificação do software. (REDHAT, 2019).

- ***Command Line Basics:*** praticamos neste tópico as operações fundamentais da linha de comando, incluindo a navegação no sistema de arquivos, a manipulação de diretórios e arquivos e a execução de comandos básicos. Essas habilidades são cruciais para qualquer profissional que busca uma compreensão sólida e eficaz do ambiente Linux.
- ***Using the command line to get help:*** neste tópico aprendemos a acessar manuais de comandos, páginas de ajuda e outras fontes de documentação diretamente pela linha de comando. Essa habilidade permite buscar informações detalhadas sobre a sintaxe, opções e funcionalidades de comandos específicos.
- ***Directories and Listing Files and managing files:*** aprendemos nesta parte do curso a navegar pelos diretórios, criar, copiar, mover e excluir arquivos, bem como gerenciar a estrutura de diretórios. Dando ênfase na utilização de comandos da linha de comando para listar o conteúdo de diretórios e modificar permissões.
- ***Archiving Files on the Command Line:*** vimos aqui como criar e manipular arquivos compactados, explorar técnicas para comprimir e descomprimir arquivos, bem como estratégias para criar backups eficientes usando comandos específicos.
- ***Turning Commands into a Script:*** este estudo foi voltado a criação de *scripts*, simples usando a linguagem de *script shell*, geralmente conhecida como *Bash*. Praticamos conceitos como variáveis, estruturas de controle de fluxo e parâmetros de *script*, permitindo compilar sequências de comandos para realizar tarefas específicas de maneira automatizada.
- ***Understanding Computer Hardware:*** com este tópico exploramos o funcionamento de componentes como processadores, memória, dispositivos de armazenamento, placas-mãe e periféricos. Vimos como identificar e compreender a função de cada componente de hardware, destacando a importância de entender a arquitetura do computador para a eficaz administração do sistema operacional Linux.
- ***Where Data is Stored:*** este estudo foi sobre a hierarquia do sistema de arquivos, destacando diretórios-chave como `/bin`, `/etc`, `/home` e `/var`. Aprendemos a reconhecer a organização do sistema de arquivos, entendendo a finalidade de cada diretório e como os dados são estruturados para suportar as operações do sistema.

- ***Your Computer on the Network:*** estudamos sobre a configuração de interfaces de rede, entendendo como atribuir endereços IP e configurar parâmetros de rede no ambiente Linux. Além disso, foram explorados conceitos de resolução de nomes e o papel do sistema operacional na comunicação eficiente em redes.
- ***Security and File Permissions:*** aqui pudemos ver conceitos como usuários, grupos e níveis de acesso e aprendemos a controlar o acesso aos arquivos e diretórios. Além disso, foram abordadas práticas recomendadas para fortalecer a segurança do sistema, como a concessão adequada de permissões e a implementação de políticas de acesso.

3.3 – Exercício Linux

Depois de concluídos os cursos passados na *Sprint*, foi proposto um exercício para pôr em prática e consolidar os temas abordados, a Figura 1 representa um diagrama de como a estrutura deve ser, o exercício consistiu em:

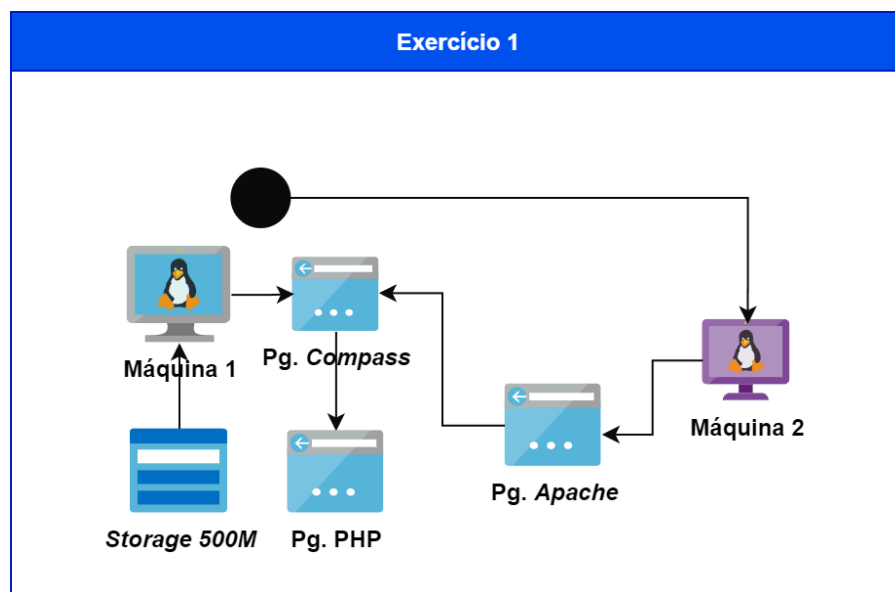


Figura 1 - Exercício Linux.

Fonte: Autor.

- **Máquina 1:**
 1. Criar uma máquina virtual com sistema operacional Linux (escolham a distribuição que mais agrada).
 2. Instalar e configurar o servidor *apache* para rodar na porta 8070.
 3. Instalar e configurar o *php*.

4. Adicionar um disco (500mb) a essa máquina e montar a partição no diretório /u01 – partição em *ext4*.
5. Quando a máquina reiniciar, esse diretório precisa continuar aparecendo como montado.
6. Criar os diretórios /u01/apache/www/imagens.
7. Criar uma página html (*index.html*) que tenha um *link* para outra página chamada *infos_php.php*.
8. Colocar a imagem do logo da *Compass* UOL como conteúdo da página *index.html* junto com o *link*. A imagem precisa estar localizada no diretório criado anteriormente. Exemplo: /u01/apache/www/imagens/logo_compasso.png.
9. Criar a página *infos_php.php* e dentro dela trazer o resultado da função *phpinfo()*.

● **Máquina 2:**

1. Criar uma máquina virtual com sistema operacional Linux (escolham a distribuição que mais agrada).
2. Instalar e configurar o servidor *nginx* para rodar na porta 8090.
3. Criar uma página html (*index.html*) que tenha um *link* para a página criada no exercício anterior (Máquina 1).

Para a realização do exercício foi instalado o sistema operacional *Ubuntu Server*, na máquina 1, onde foi necessário realizar a instalação do servidor *apache* como descrito na documentação descrita por MARCELO (2005) e as demais configurações para atender os requisitos da atividade. Abaixo demonstro os comandos e configurações necessárias.

- ***sudo apt-get update***: Comando para atualização da lista de pacote do sistema operacional.
- ***sudo apt install apache2 -y***: Realiza a instalação do servidor *apache*

Com o intuito de configurar o servidor *apache* e instalar o PHP para rodar na porta 8070, fizemos a alteração no arquivo *ports.conf* alterando a linha do arquivo que corresponde a “Listen 80” para “Listen 8070” conforme na Figura 2. O arquivo pode ser encontrado no diretório */etc/apache2/ports.conf*.

- ***sudo nano /etc/apache2/ports.conf***: este comando abre o editor de texto *nano* para editar o arquivo *ports.conf*.
- ***sudo apt install php libapache2-mod-php -y***: instala o PHP e as dependências necessárias para o funcionamento.
- ***sudo restart apache2***: realiza o restart do *apache* para aplicar as alterações.

- ***sudo status apache2***: verifica o *status* do *apache* se está ativo ou não.

```
1 # If you just change the port or add more ports here, you will likely also
2 # have to change the VirtualHost statement in
3 # /etc/apache2/sites-enabled/000-default.conf
4
5 Listen 8070
6
7 <IfModule ssl_module>
8     Listen 443
9 </IfModule>
10
11 <IfModule mod_gnutls.c>
12     Listen 443
13 </IfModule>
14
```

Figura 2 – arquivo *ports.conf*.

Fonte: Autor.

Com o objetivo de adicionar os 500M de disco foi necessário configurar manualmente na *VM* (Virtual Machine - do Inglês Máquina Virtual). Para esta atividade foi utilizado a *Oracle VM VirtualBox*, as configurações foram feitas de acordo com o passo a passo das Figuras 3 e 4.

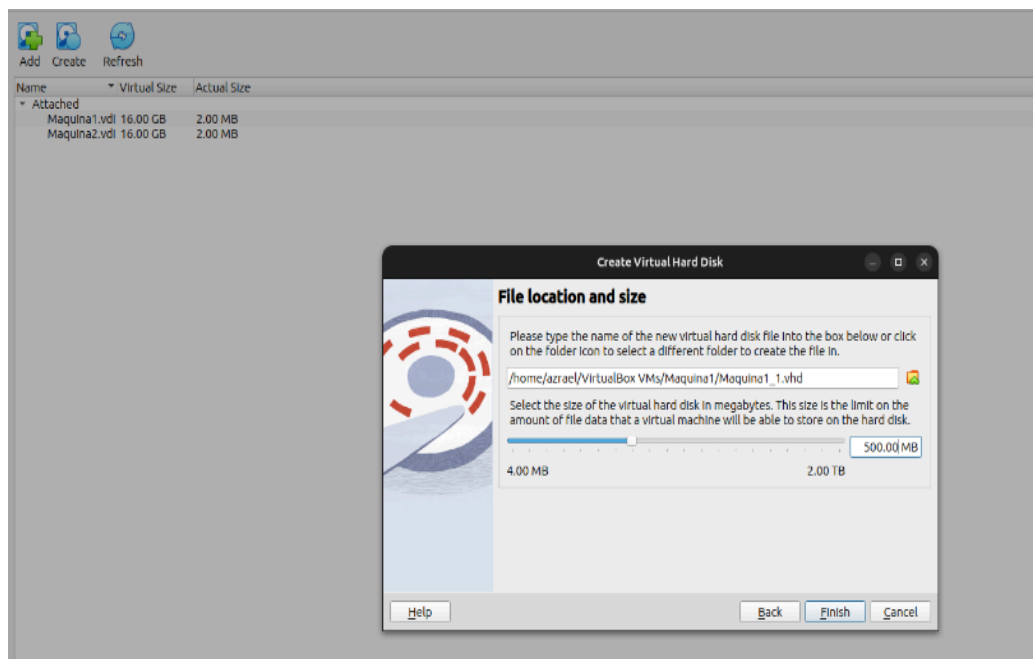


Figura 3 – Definindo o tamanho do disco.

Fonte: Autor.

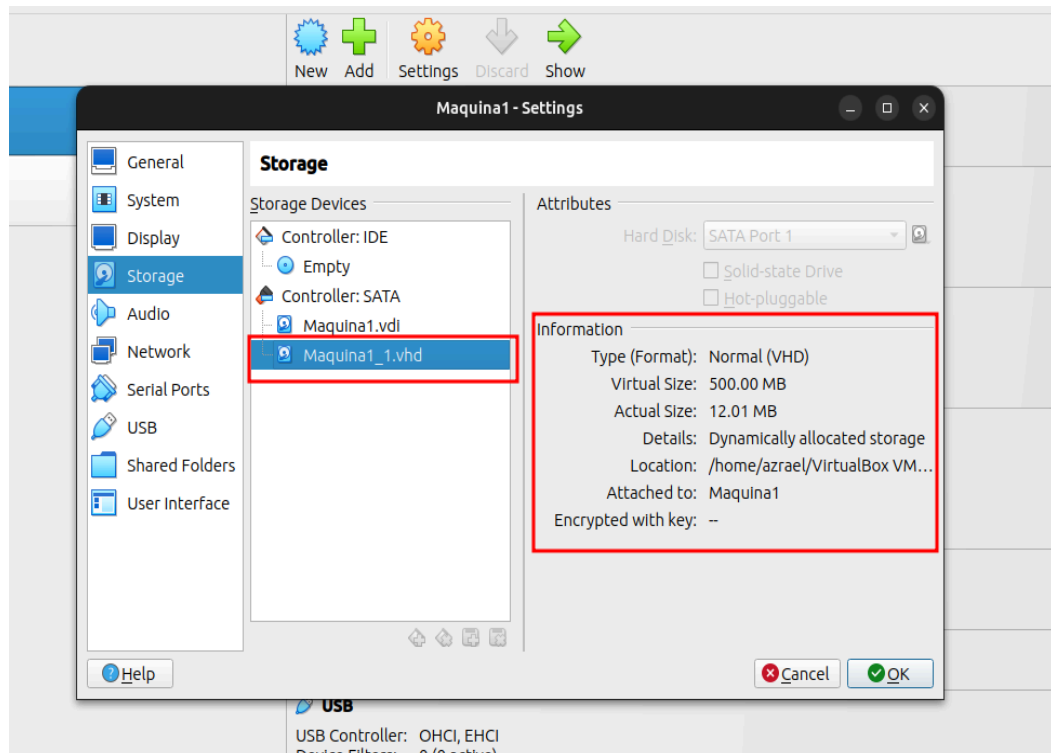


Figura 4 – attach do disco.

Fonte: Autor.

Uma vez realizada a adesão do disco a VM é preciso fazer a montagem do volume ao sistema operacional. Para realizar a montagem deste volume foi seguido um passo a passo, digitando os seguintes comandos.

- ***sudo fdisk -l***: este comando lista os discos disponíveis no sistema. Na Figura 5 podemos ver a lista de discos disponíveis, na figura podemos observar o disco criado e anexado.

```

Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 63.46 MiB, 66547712 bytes, 129976 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: VBox HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 84D9B02F-10CF-4B6C-BCD6-87346446DF83

Device      Start      End  Sectors  Size Type
/dev/sda1    2048      4095    2048    1M BIOS boot
/dev/sda2    4096  3674111  3670016  1.8G Linux filesystem
/dev/sda3   3674112 33552383 29878272 14.2G Linux filesystem

Disk /dev/sdb: 500 MiB, 524288000 bytes, 1024000 sectors
Disk model: VBox HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

```

Figura 5 – terminal *Linux*.

Fonte: Autor.

Para realizar a montagem da partição com o disco adicionado, é necessário executar o comando ***sudo fdisk /dev/sdb***, este comando abre um menu de configurações para realizar a montagem do novo volume. Podemos ver as opções de de criação na Figura 6:

- **n**: para opção de criar nova partição.
- **p**: para informar que é uma partição primária.
- **1**: para o número da partição.
- **+500M**: para adicionar os 500MB da partição.

```

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1023999, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1023999, default 1023999): +500M

```

Figura 6 – menu de opções (*sudo fdisk /dev/sdb*).

Fonte: Autor.

Uma vez criada a nova partição seguimos para o próximo tópico do exercício que consiste em formatar e montar o diretório como ilustrados nas Figuras 7, 8 e 9, para manter essas configurações ao reiniciar a máquina é necessário adicionar o diretório montado na tabela de inicialização do sistema. Seguem abaixo os passos necessários para realizar essas configurações.

- **`sudo fdisk /dev/sdb`**: para ver o caminho atribuído a partição montada.

```
Command (m for help): p
Disk /dev/sdb: 500 MiB, 524288000 bytes, 1024000 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xfda69c41

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1           2048 1023999 1021952  499M 83 Linux
```

Figura 7 – partição criada.

Fonte: Autor.

- **`sudo mkfs.ext4 /dev/sdb1`**: realiza a formatação da partição no formato *ext4*.
- **`sudo mkdir /u01`**: cria um subdiretório no diretório principal (*/*).
- **`sudo mount -t ext4 /dev/sdb1 /u01`**: faz a montagem a partição “/dev/sdb1” e atribui o subdiretor “/u01” a esta partição.

```
> sudo mkdir /u01
~                                     at 00:49:36
> sudo mount -t ext4 /dev/sdb1 /u01
~                                     at 00:49:42
>
```

Figura 8 – montagem partição.

Fonte: Autor.

- **`sudo blkid`**: mostra os UUID (*Universally Unique Identifier*) das partições existentes no sistema. Este UUID é importante para adicioná-lo na tabela de discos de inicialização do sistema.
- **`sudo nano /etc/fstab`**: abre o arquivo que contém as configurações dos discos. Neste arquivo adicionamos o UUID referente a partição montada, assim resolvendo o problema de perda de configuração ao realizar o *restart* da máquina.

```

# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/ubuntuv-g/ubuntuv-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-9i8cMd9d1vd4yYihvEA5y0oQnDV2Dfz2gpaYyh5J1xLTSs1DT00q4gzV96141Tz5 / ext4
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/4d2c00b9-1fe8-4261-93cd-c3040e512890 /boot ext4 defaults 0 1
#New Partition
UUID="21663029-3e48-43bd-a03a-0b1cfc2fafdd" /u01 ext4 defaults 0 0

```

Figura 9 – montagem partição.

Fonte: Autor.

Para criar a página *web* foi preciso criar o diretório `/u01/apache/www/imagens` dentro da partição montada anteriormente. Executando o comando `sudo mkdir -p /u01/apache/www/imagens`. Após isso, para atender os requisitos do exercício o próximo passo foi editar o `index.html` encontrado no caminho `/var/www/html`. O arquivo `index.html` depois de configurado para atender os requisitos ficou conforme a Figura 10. Para a exibição da logo *Compass* foi feito o download no diretório `/u01/apache/www/imagens`.

```

1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Compass UOL</title>
7   <style>
8     body {
9       background-color: #9b9a9a;
10      margin: 0;
11      display: flex;
12      align-items: center;
13      justify-content: center;
14      height: 100vh;
15    }
16
17    #conteudo {
18      text-align: center;
19    }
20
21    #conteudo img {
22      max-width: 100%;
23      height: auto;
24    }
25  </style>
26 </head>
27 <body>
28   <div id="conteudo">
29     
30     <a href="./infos_php.php" ><h1>PAGINA PHP</h1></a>
31   </div>
32 </body>
33 </html>

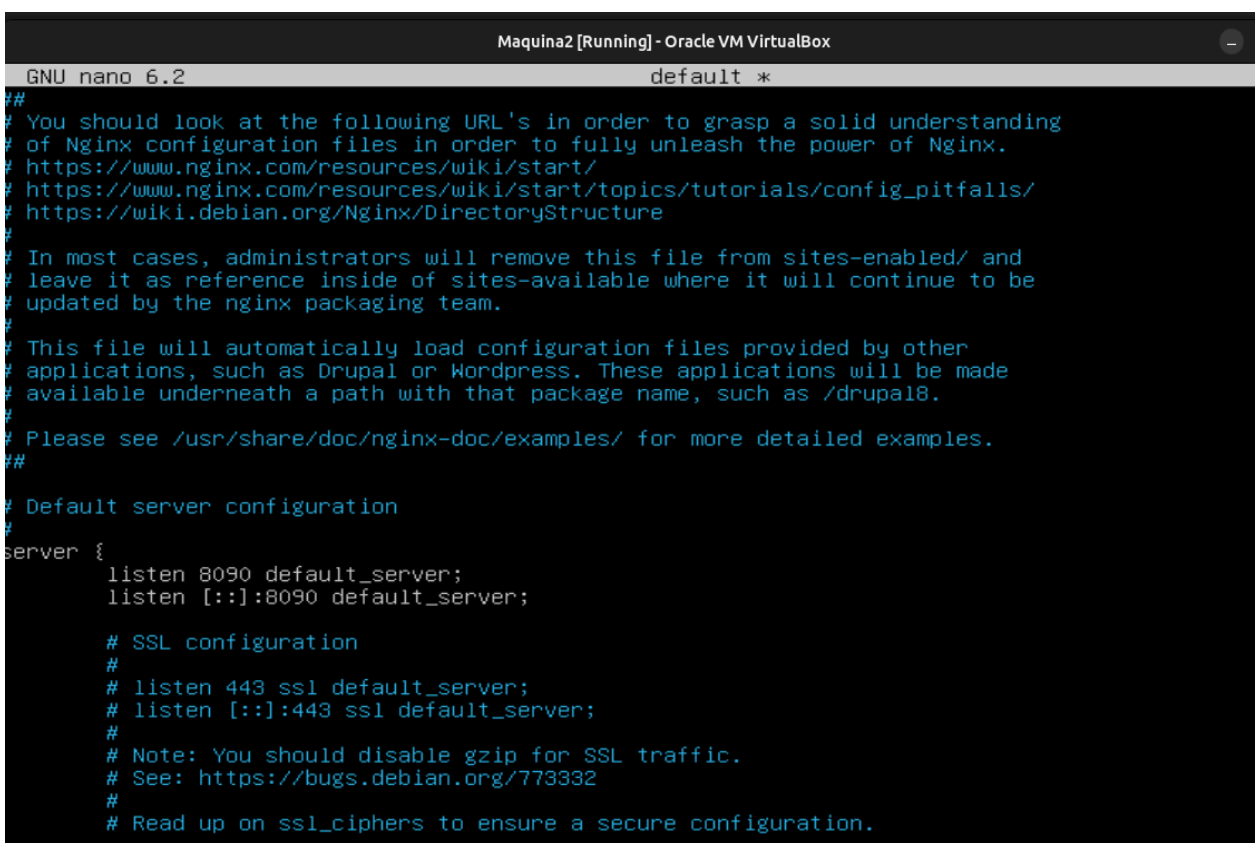
```

Figura 10 – `index.html`.

Fonte: Autor.

A página `infos_php.php` foi criada no mesmo caminho do `index.html`, para criar e adicionar o comando necessário para exibição das informações do PHP foi executado o comando `sudo nano infos_php.php` e adicionado à função `phpinfo()`.

Na máquina 2 também foi utilizado Ubuntu Server e foi instalado um servidor `nginx`. A instalação foi feita com o comando `sudo apt install nginx`, para alteração da porta padrão do servidor foi editada a linha que corresponde a porta do servidor “`listen 80 default_server`” no arquivo `default.conf` encontrado no diretório `/etc/nginx/sites-enabled/default.conf` (Figura 11). Depois de editado foi feita a reinicialização do servidor para aplicar a alteração executando o comando `sudo systemctl restart nginx`.



```
Maquina2 [Running] - Oracle VM VirtualBox
GNU nano 6.2                                default *
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8090 default_server;
    listen [::]:8090 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
```

Figura 11 – `default.conf`.

Fonte: Autor.

Para fazer o redirecionamento para a máquina 1 foi criado um arquivo `index.html` no caminho `/var/www/html` contendo o endereço e a porta correspondente ao servidor da máquina 1 igual na Figura 12.


```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>NGINX</title>
7   <style>
8     body {
9       background-color: #9b9a9a;
10      margin: 0;
11      display: flex;
12      align-items: center;
13      justify-content: center;
14      height: 100vh;
15    }
16
17    #conteudo {
18      text-align: center;
19    }
20  </style>
21 </head>
22 <body>
23   <div id="conteudo">
24     <a href="http:10.0.2.15:8070" ><h1>SRV01</h1></a>
25   </div>
26 </body>
27 </html>
```

Figura 12 – *index.html*.

Fonte: Autor.

O resultado final da atividade foi acessar a máquina 1 através do endereço IP da máquina 2 e clicando no link “SRV01” sendo direcionado para página com a logo da “Compass” e com o link para página *infos.php*. O fluxo da atividade está representado na figura 1. Segue abaixo as Figuras 13, 14, 15 das telas apresentando o resultado final.

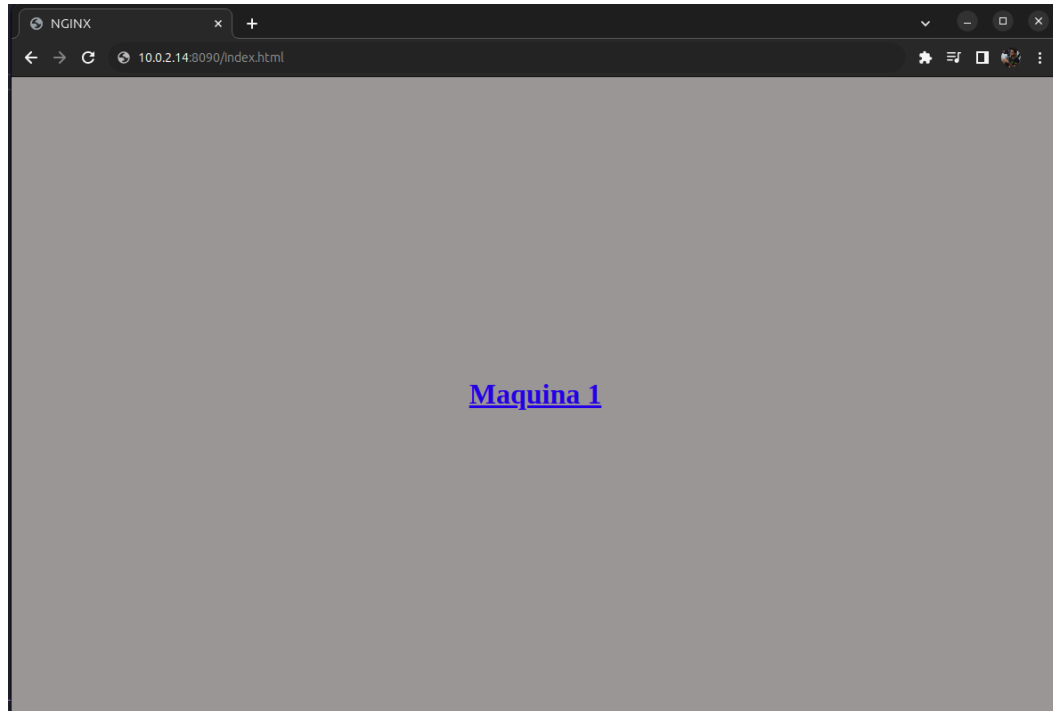


Figura 13 – página *nginx*.

Fonte: Autor.

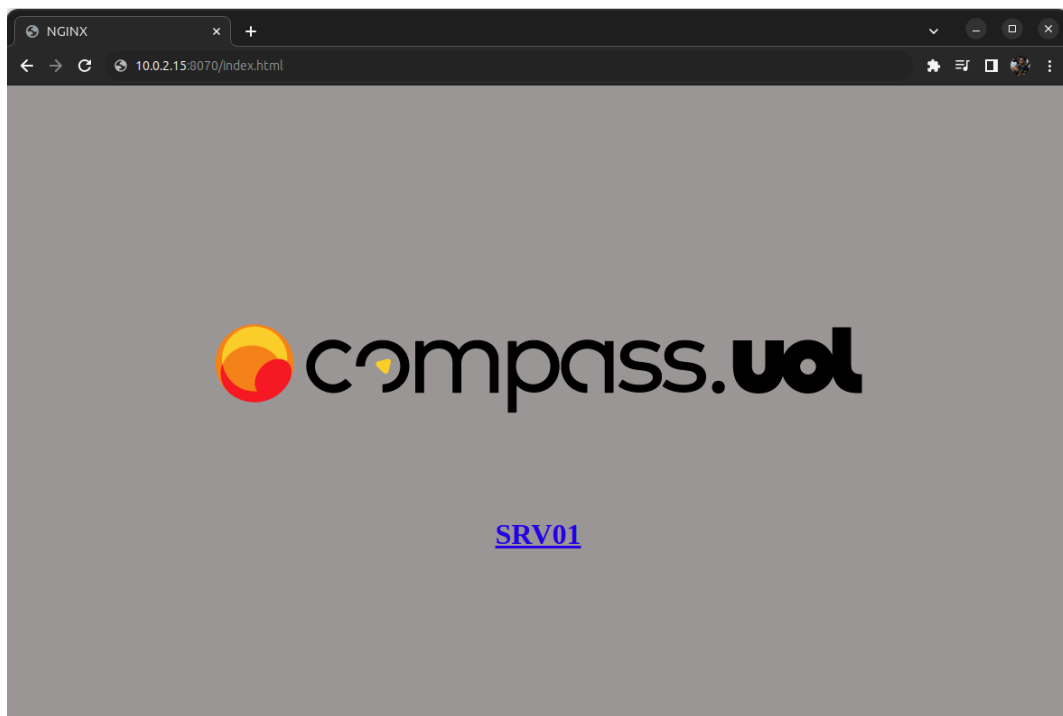


Figura 14 – página com a logo *Compass*.

Fonte: Autor.

The screenshot shows the PHP info page for version 8.1.12-1ubuntu4.3. The page is titled 'PHP Version 8.1.12-1ubuntu4.3' and features the PHP logo. It contains a table with the following information:

System	Linux azra6l-pc 6.2.0-36-generic #37-Ubuntu SMP PREEMPT_DYNAMIC Wed Oct 4 10:14:28 UTC 2023 x86_64
Build Date	Aug 17 2023 17:37:48
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/apache2
Loaded Configuration File	/etc/php/8.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/apache2/conf.d
Additional .ini files parsed	/etc/php/8.1/apache2/conf.d/10-opcache.ini, /etc/php/8.1/apache2/conf.d/10-pdo.ini, /etc/php/8.1/apache2/conf.d/20-calendar.ini, /etc/php/8.1/apache2/conf.d/20-ctype.ini, /etc/php/8.1/apache2/conf.d/20-curl.ini, /etc/php/8.1/apache2/conf.d/20-dom.ini, /etc/php/8.1/apache2/conf.d/20-ffi.ini, /etc/php/8.1/apache2/conf.d/20-fileinfo.ini, /etc/php/8.1/apache2/conf.d/20-ftp.ini, /etc/php/8.1/apache2/conf.d/20-gettext.ini, /etc/php/8.1/apache2/conf.d/20-gd.ini, /etc/php/8.1/apache2/conf.d/20-gmp.ini, /etc/php/8.1/apache2/conf.d/20-gnupg.ini, /etc/php/8.1/apache2/conf.d/20-http.ini, /etc/php/8.1/apache2/conf.d/20-igmp.ini, /etc/php/8.1/apache2/conf.d/20-intl.ini, /etc/php/8.1/apache2/conf.d/20-ldap.ini, /etc/php/8.1/apache2/conf.d/20-libxml.ini, /etc/php/8.1/apache2/conf.d/20-mcrypt.ini, /etc/php/8.1/apache2/conf.d/20-memcached.ini, /etc/php/8.1/apache2/conf.d/20-mysql.ini, /etc/php/8.1/apache2/conf.d/20-mysqli.ini, /etc/php/8.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.1/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/8.1/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/8.1/apache2/conf.d/20-pgsql.ini, /etc/php/8.1/apache2/conf.d/20-readline.ini, /etc/php/8.1/apache2/conf.d/20-redis.ini, /etc/php/8.1/apache2/conf.d/20-session.ini, /etc/php/8.1/apache2/conf.d/20-shmop.ini, /etc/php/8.1/apache2/conf.d/20-sockets.ini, /etc/php/8.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.1/apache2/conf.d/20-sysvsem.ini, /etc/php/8.1/apache2/conf.d/20-sysvshm.ini, /etc/php/8.1/apache2/conf.d/20-tk.ini, /etc/php/8.1/apache2/conf.d/20-xmlrpc.ini, /etc/php/8.1/apache2/conf.d/20-xsl.ini, /etc/php/8.1/apache2/conf.d/20-zip.ini, /etc/php/8.1/apache2/conf.d/20-zlib.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS
PHP Extension Build	API20210902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available disabled
Registered PHP Streams	ftp, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv*

At the bottom of the page, it says 'Configuration apache2handler' and includes the Zend Engine logo.

Figura 15 – página *infos_php.php*.

Fonte: Autor.

A segunda *Sprint* do estágio foi com foco no estudo de orquestração de container e comunicação entre eles, com isso foi apresentado cursos focados em redes de computadores, containers e Kubernetes, como apresentado no Quadro 2.

Quadro 2 - Segunda lista de estudos da Alura.

CURSO	TEMA	CARGA HORÁRIA
Redes	Redes parte 1: Introdução, Conceitos e Prática	10h
Redes	DNS: Entenda a resolução de nomes na internet	12h
<i>Docker</i>	<i>Docker</i> : Criando containers sem dor de cabeça	10h
<i>Kubernetes</i>	<i>Kubernetes</i> : <i>Pods</i> , <i>Services</i> e <i>ConfigMaps</i>	8h
<i>Kubernetes</i>	<i>Kubernetes</i> : <i>Deployments</i> , <i>Volumes</i> e Escalabilidade	8h
Exercícios de Avaliação	Exercício <i>Kubernetes</i>	

Fonte: Autor.

- **Redes Parte 1: Introdução, Conceitos e Prática:** neste curso foi possível ter uma visão abrangente sobre os principais componentes de redes, incluindo *hardware*, *software*, protocolos DHCP, TCP, DNS, IP, foi visto o funcionamento de *switches* e roteadores, através do programa *Packet Tracer* que simula cenários e equipamentos de redes.
- **DNS: Entenda a resolução de nomes na internet:** no curso foi aprendido o funcionamento do Sistema de Nomes de Domínio (DNS) e a tradução de nomes de domínio em endereços IP foi visto também conceitos fundamentais, como registros DNS, servidores de nomes (DNS servers) e os diferentes tipos de consultas DNS.
- **Docker: Criando containers sem dor de cabeça:** neste curso vimos como a tecnologia de containers Docker e seu processo de criação de ambientes isolados funcionam, vimos também conceitos fundamentais do Docker, incluindo imagens, containers, e o Docker *Hub*. Foi estudado a criação, execução e gerenciamento de containers de maneira eficiente, explorando a portabilidade e escalabilidade proporcionadas por essa tecnologia, abordando práticas como *Dockerfile*, que permite a definição da configuração do container por meio de código. (DOCKER, 2023).
- **Kubernetes: Pods, Services e ConfigMaps:** aqui vimos sobre a orquestração de containers utilizando os recursos disponíveis no kubernetes, como o conceito de *Pods* que representa a menor unidade existente no ecossistema do *kubernetes*, *services* que empenham o papel de expor os *pods* para que possam ser acessíveis e também *Configmaps* que oferece uma maneira de separar a configuração do contêiner do próprio código da aplicação.
- **Kubernetes: Deployments, Volumes e Escalabilidade:** neste tópico foi aprendido como funcionam alguns gerenciadores de estado da aplicação como *ReplicaSets* e *Deployment*, foi visto também como configurar a elasticidade dos containers usando *Horizontal Pod Autoscaler* podendo aumentar ou diminuir o número de *Pods* de acordo com a quantidade de carga. Foi apresentado a criação de volumes estáticos e dinâmicos a partir de *Storage Class*, *Persistent volume* e *Persistent volume claim*.

3.4 – Exercício Kubernetes

Para consolidar os conceitos e práticas apresentadas nos cursos de redes, contêineres e orquestração de contêiner foi proposto duas atividades, para serem implementadas utilizando *Kubernetes* este exercício teve como base as boas práticas recomendadas por KUBERNETES, (2023), falando da importância da orquestração de containers.

- **Atividade 1:**

1. Criar um *namespace* com o nome de “labnginx”.
2. Fazer o *apply* de dois *pods nginx* neste *namespace*, dentro de um único *Ingress*, onde o *Load Balancer* distribuirá a carga entre estes dois *pods*.
3. Abrir o *browser* e fazer o teste no navegador se a tela padrão do *nginx*.
4. Fazer um *describe* dos *pods nginx* criado. Qual o IP que foi estabelecido para cada *pod*?
5. Verificar os *logs* dos dois *pods nginx* que foram criados, que informações são mostradas?

Como solicitado no exercício, foi primeiramente criado o *namespace*, onde toda infraestrutura será implementada. Podemos ver as configurações necessárias para criar um *namespace* em um cluster kubernetes na Figura 16. Para atender o requisito de balanceamento de carga entre os *pods* foi criado um *Ingress Nginx* com um *Service* para o *Deployment* possibilitando fazer a conexão dos *pods* a redes externas. É possível ver os códigos de configuração desses recursos nas figuras a seguir, Figura 17 código de criação *Ingress Nginx*, Figura 18 código de criação *Service* e Figura 19 contendo a implementação do código do *Deployment* com as especificações necessárias para subir os dois *pods nginx*.

```
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: labnginx
```

Figura 16 – código de criação *Namespace*.

Fonte: Autor.

```
1  apiVersion: networking.k8s.io/v1
2  kind: Ingress
3  metadata:
4    name: nginx-ingress
5    namespace: labnginx
6
7  spec:
8    rules:
9    - host: labnginx.test
10     http:
11       paths:
12       - path: /
13         pathType: Prefix
14         backend:
15           service:
16             name: nginx-svc
17             port:
18               number: 80
```

Figura 17 – código de criação *Ingress Nginx*.

Fonte: Autor.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-svc
5    namespace: labnginx
6  spec:
7    selector:
8      app: nginx
9    type: ClusterIP
10   ports:
11   - name: nginx
12     protocol: TCP
13     port: 80
```

Figura 18 – código de criação *Service*.

Fonte: Autor.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: nginx
5    namespace: labnginx
6    labels:
7      app: nginx
8  spec:
9    selector:
10     matchLabels:
11       app: nginx
12   replicas: 2
13   template:
14     metadata:
15       labels:
16         app: nginx
17     spec:
18       containers:
19         - name: nginx
20           image: nginx:latest
21           ports:
22             - containerPort: 80
23               name: nginx
24

```

Figura 19 – código de criação *Deployment*.

Fonte: Autor.

- **Atividade 2.**

1. Criar um namespace chamado “*labwordpress*”, tudo o que for feito deverá estar dentro deste namespace;
2. Fazer o apply do arquivo de *service* do *MySQL*, e mudar a porta padrão do banco *MySQL* para 3308;
3. Criar o arquivo *secret* que deverá conter o *password* do banco *MySQL*;
4. Fazer o *apply* do arquivo de *Persistent Volume Claim* do *MySQL* para um *capacity* de 3GB;
5. Fazer o apply do arquivo de *deployment* do *MySQL*, criar um *volume mount* no *deployment* do *MySQL* chamado “*mysql-persistent-storage-lab*”, apontando para */var/lib/mysql*. Lembre-se de criar o *volume* em si com o mesmo nome do *volume mount*;
6. Fazer o apply do arquivo de *service* do *wordpress* alterar para a TCP Porta 80;

7. Fazer o *apply* do arquivo de *Persistent Volume Claim* (PVC) do *wordpress*, para um *capacity* de 3GB;
8. No arquivo de *deployment* do *wordpress*, criar um *volume mount* no *deployment* do *Wordpress* chamado “*wordpress-persistent-storage-lab*”, apontando para */var/www/html*. Lembre-se de criar o *volume* em si com o mesmo nome do *volume mount*;
9. No arquivo de *deployment* do *wordpress*, inserir o *secret* contendo o *password* do *MySQL*, criado no começo do exercício.
10. Fazer o *apply* do arquivo de *deployment* do *wordpress*;
11. Verificar se os *Pods*, *services* e os *PVC* foram criados da forma correta dentro *namespace* criado no início deste exercício;
12. Verificar qual foi a URI (*Uniform Resource Name*) gerada através do *ingress* do *kubernetes*;
13. Copiar essa URI do *ingress* e colar no browser para abrir a tela inicial do *wordpress*.

Criação do namespace *labwordpress* para realização da segunda atividade sobre *kubernetes* (Figura 20).

```
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: labwordpress
```

Figura 20 – código de criação *Namespace*.

Fonte: Autor.

Para realizar as etapas da segunda atividade, primeiro foi configurado a base de dados *MySQL*, seguindo a mesma configuração padrão de uma aplicação usando *Kubernetes*, criando e configurando, *service*, *deployment*, *secret* e *volume*. Nas Figuras 21, 22, 23 e 24 estão ilustrados os códigos necessários para a infra estrutura de uma base de dados *MySQL*.

Como requisito da atividade foram alterados a porta padrão do MySQL para a porta 3308 e também foi criado *Secret* contendo a senha de acesso ao banco de dados. Essas configurações estão nos arquivos *Service* e *Secret*.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql-svc
5    namespace: labwordpress
6  spec:
7    selector:
8      app: mysql
9    ports:
10   - name: mysql
11     protocol: TCP
12     port: 3308
13     targetPort: 3306
14   type: ClusterIP
15
```

Figura 21– código de criação *Service*.

Fonte: Autor.

```
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: mysql-secret
5    namespace: labwordpress
6  type: Opaque
7  data:
8    PASSWORD: c2VuaGFkaWZpY2lsMTIzQCMh
9
```

Figura 22 – código de criação *Secret*.

Fonte: Autor.

As configurações de armazenamento utilizando *Volumes* e mapeamento das configurações de *Secret* e *Volume*, estão nos arquivos de PVC e *Deployment*.

```
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: mysql-pvc
5    namespace: labwordpress
6    labels:
7      app: mysql-pvc
8  spec:
9    accessModes:
10   - ReadWriteOnce
11   resources:
12     requests:
13       storage: 3Gi
```

Figura 23 – código de criação PVC.

Fonte: Autor.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mysql
5    namespace: labwordpress
6    labels:
7      app: mysql
8  spec:
9    selector:
10     matchLabels:
11       app: mysql
12    replicas: 1
13    template:
14     metadata:
15     labels:
16       app: mysql
17     spec:
18     containers:
19     - name: mysql
20       image: mysql:8.0
21       env:
22       - name: MYSQL_ROOT_PASSWORD
23         valueFrom:
24           secretKeyRef:
25             name: mysql-secret
26             key: PASSWORD
27       - name: MYSQL_DATABASE
28         value: wordpress
29       - name: MYSQL_USER
30         value: wordpress
31       - name: MYSQL_PASSWORD
32         valueFrom:
33           secretKeyRef:
34             name: mysql-secret
35             key: PASSWORD
36     ports:
37     - containerPort: 3306
38       name: mysql
39     volumeMounts:
40     - name: mysql-persistent-storage-lab
41       mountPath: /var/lib/mysql
42     volumes:
43     - name: mysql-persistent-storage-lab
44       persistentVolumeClaim:
45         claimName: mysql-pvc

```

Figura 24 – código de criação *Deployment*.

Fonte: Autor.

As configurações necessárias para aplicação Wordpress foram criadas seguindo o mesmo padrão da base de dados MySQL, com um arquivo de configuração a mais chamado *ConfigMap* Figura 25, este arquivo contém as variáveis de ambiente necessária para que a aplicação faça a comunicação com a base de dados criada anteriormente. Figuras 26, 27 e 28 correspondem às configurações necessárias para a comunicação da aplicação.

```
1 kind: ConfigMap
2 apiVersion: v1
3 metadata:
4   name: wordpress-configmap
5   namespace: labwordpress
6 data:
7   WORDPRESS_DB_HOST: mysql-svc.labwordpress.svc.cluster.local:3308
8   WORDPRESS_DB_USER: wordpress
9   WORDPRESS_DB_NAME: wordpress
10
```

Figura 25 – código de criação *Deployment*.

Fonte: Autor.

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: wordpress-svc
5   namespace: labwordpress
6 spec:
7   selector:
8     app: wordpress
9   type: ClusterIP
10 ports:
11 - name: wordpress
12   protocol: TCP
13   port: 80
14
```

Figura 26 – código de criação *Service*.

Fonte: Autor.

```
1 apiVersion: v1
2 kind: PersistentVolumeClaim
3 metadata:
4   name: wordpress-pvc
5   namespace: labwordpress
6 labels:
7   app: wordpress-pvc
8 spec:
9   accessModes:
10  - ReadWriteOnce
11 resources:
12   requests:
13     storage: 3Gi
```

Figura 27 – código de criação *PVC*.

Fonte: Autor.

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: wordpress
5    namespace: labwordpress
6    labels:
7      app: wordpress
8  spec:
9    selector:
10     matchLabels:
11       app: wordpress
12   replicas: 1
13   template:
14     metadata:
15       labels:
16         app: wordpress
17     spec:
18       containers:
19         - name: wordpress
20           image: wordpress:6.2.1-apache
21           envFrom:
22             - configMapRef:
23                 name: wordpress-configmap
24           env:
25             - name: WORDPRESS_DB_PASSWORD
26               valueFrom:
27                 secretKeyRef:
28                   name: mysql-secret
29                   key: PASSWORD
30           ports:
31             - containerPort: 80
32               name: wordpress
33           volumeMounts:
34             - name: wordpress-persistent-storage-lab
35               mountPath: /var/www/html
36       volumes:
37         - name: wordpress-persistent-storage-lab
38           persistentVolumeClaim:
39             claimName: wordpress-pvc

```

Figura 28 – código de criação *Deployment*.

Fonte: Autor.

Tendo como resultado final dessa atividade a tela inicial do Wordpress (Figura 29), uma vez que todas as configurações estejam corretas é possível visualizar a tela inicial para que as configurações de primeiro acesso possam ser realizadas.

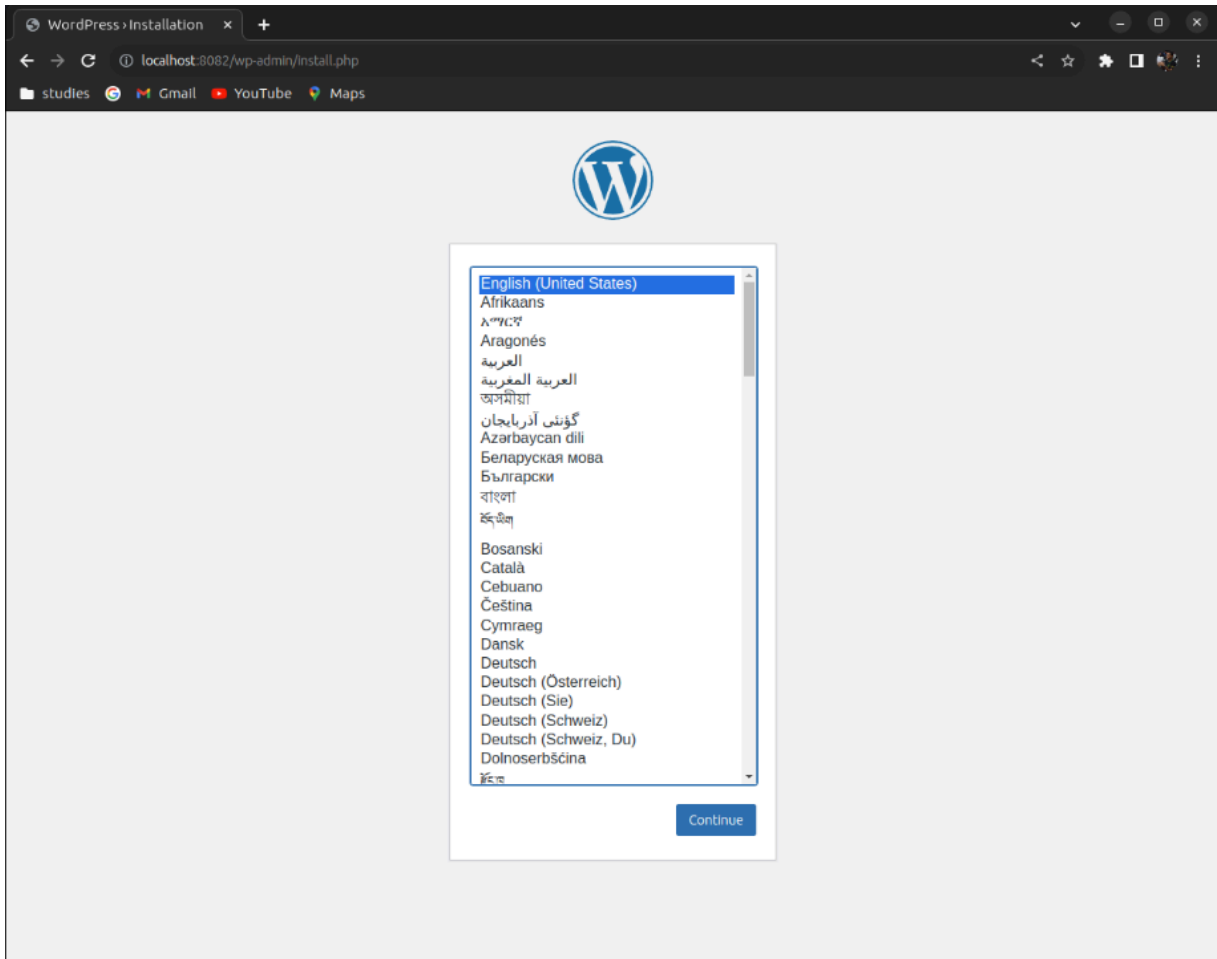


Figura 29 – tela inicial Wordpress.

Fonte: Autor.

No Quadro 3, apresento os temas vistos no curso sobre *Cloud*, cada um dos cursos estudados trouxeram impactos positivos sobre o entendimento da computação em nuvem.

Quadro 3 - Terceira lista de estudos da Alura.

CURSO	TEMA	CARGA HORÁRIA
AWS: Conhecimentos Gerais	<i>Deploy no Amazon EC2: Alta disponibilidade e escalabilidade de uma aplicação</i>	10h
AWS: Conhecimentos Gerais	<i>Amazon Lightsail: Descomplicando a nuvem</i>	8h
AWS: Conhecimentos Gerais	<i>Amazon CloudWatch: Visibilidade completa das aplicações e serviços na nuvem</i>	6h
AWS: Conhecimentos Gerais	<i>Amazon S3: Manipule e armazene objetos na nuvem</i>	12h
AWS: Conhecimentos Gerais	<i>Amazon VPC: Provisione uma nuvem privada</i>	6h
AWS: Conhecimentos Gerais	<i>Amazon CloudFront e Route53: Distribua os seus arquivos e crie seu próprio domínio</i>	8h
AWS: Conhecimentos Gerais	<i>Amazon ECS: Gerencie Docker na nuvem da AWS</i>	8h
AWS: Conhecimentos Gerais	<i>Amazon Elastic Beanstalk Parte 1: Container Docker</i>	6h
AWS: Conhecimentos Gerais	<i>Amazon Elastic Beanstalk Parte 2: Múltiplos containers e nginx</i>	8h
<i>AWS Certified Cloud Practitioner</i>	<i>AWS Certified Cloud Practitioner: Domain 1 e 2</i>	8h
<i>AWS Certified Cloud Practitioner</i>	<i>AWS Certified Cloud Practitioner: Domain 3 e 4</i>	10h

Fonte: Autor.

- ***Deploy no Amazon Elastic Compute Cloud (EC2)***: este curso abordou a implementação e gestão eficaz de aplicações em ambientes de nuvem utilizando os recursos oferecidos pelo Amazon *EC2*. Foi visto também como configurar e implantar aplicações de forma a garantir alta disponibilidade, permitindo que estas sejam acessíveis de forma contínua e confiável. Além disso, exploramos estratégias de escalabilidade, para dimensionar aplicações conforme a demanda, garantindo um desempenho consistente mesmo em períodos de pico de tráfego.
- ***Amazon Lightsail***: aqui tivemos uma introdução abrangente e prática ao serviço de hospedagem simplificada da Amazon. Vimos como criar, configurar e gerenciar instâncias de servidores virtuais pré-configurados com facilidade, sem a necessidade de conhecimentos avançados em infraestrutura de nuvem. Ao longo do curso, foram

abordados tópicos como provisionamento de servidores, configuração de redes, implementação de aplicativos, monitoramento de recursos e segurança. Com ênfase na simplicidade e na acessibilidade, para hospedar uma variedade de aplicações web, blogs, sites estáticos e muito mais, permitindo-lhes aproveitar os benefícios da nuvem sem complicações.

- **Amazon CloudWatch:** este curso foi um guia abrangente sobre o serviço de monitoramento e observabilidade da *Amazon Web Services* (AWS). Exploramos os recursos essenciais do *CloudWatch*, aprendendo a coletar e analisar métricas, logs e eventos de forma eficiente em toda a infraestrutura da AWS. Os tópicos abordados incluem configuração de gatilhos para alertas proativos, criação de painéis personalizados para visualização de dados, automação de ações com base em eventos.
- integração com outros serviços da AWS. Com uma abordagem prática e orientada para casos reais, este curso capacitou a utilização do *CloudWatch* para garantir a integridade, o desempenho e a eficiência de aplicações e recursos na nuvem.
- **Amazon Simple Storage Service (S3):** vimos neste curso de forma abrangente e prática um dos serviços de armazenamento em nuvem mais populares e escaláveis da AWS. Durante o curso, aprendemos a criar, configurar e gerenciar *buckets* de armazenamento, armazenar e recuperar dados de forma segura e eficiente, além de explorar recursos avançados, como controle de acesso, criptografia, versionamento de objetos e políticas de ciclo de vida. Com uma abordagem hands-on na integração do *Amazon S3* com outros serviços da AWS, como *EC2*, *Lambda* e *CloudFront*, para construir soluções escaláveis e altamente disponíveis.
- **Amazon Virtual Private Cloud (VPC):** aprendemos neste curso a projetar e configurar uma VPC personalizada, incluindo a definição de sub-redes, rotas, tabelas de roteamento e gateways de Internet e VPN. Tópicos avançados como segurança de rede, grupos de segurança, ACLs e conexões *peering* também foram abordados em detalhes. Com uma abordagem prática e orientada a casos reais, fomos capacitados a construir arquiteturas de rede seguras e altamente escaláveis na AWS, essenciais para hospedar aplicações críticas na nuvem. Este curso foi um guia sobre como criar e gerenciar redes virtuais isoladas na AWS.
- **Amazon CloudFront e Route53:** este curso ensinou a configurar e otimizar o *Amazon CloudFront*, um serviço de *CDN* (*Content Delivery Network*), que serve para distribuir conteúdo de forma rápida e segura a usuários finais em todo o mundo. Além disso, fomos capacitados a utilizar o *Amazon Route 53*, um serviço de DNS (*Domain Name*

System), para gerenciar a resolução de nomes de domínio e roteamento de tráfego de forma inteligente e escalável.

- **Amazon *Elastic Container Service (ECS)***: foi abordado como utilizar este serviço de gerenciado de contêineres para implantar, gerenciar e escalar containers Docker de forma eficiente na nuvem. Foi visto também como configurar e executar clusters de contêineres *ECS*, criar definições de tarefas e serviços, implantar aplicações contidas em contêineres, e integrar o *ECS* com outros serviços da AWS, como o *Elastic Load Balancing* e *Amazon CloudWatch*.
- **Amazon *Elastic Beanstalk***: com o curso de *Elastic Beanstalk* aprendemos um pouco sobre serviços PaaS (*Platform as a Service*), execução e configuração de recursos como balanceamento de carga e auto escalonamento, para monitorar o desempenho de aplicações e integrar com outros serviços da AWS, como RDS e S3. Aprendemos também a partir de um teste local utilizando o Docker a integrar o *Elastic Beanstalk* com o *Amazon Elastic Block Store (EBS)*, que fornece armazenamento em nível de bloco para instâncias do EC2.
- ***AWS Certified Cloud Practitioner***: o curso de certificação *AWS Certified Cloud Practitioner* foi uma introdução abrangente aos conceitos fundamentais da AWS e serviços em nuvem. O exame abrange uma variedade de tópicos essenciais, incluindo os benefícios da nuvem, modelos de entrega de serviços em nuvem, arquitetura de nuvem, segurança e conformidade, e práticas recomendadas de faturamento e preços. No curso foi apresentado os serviços e suas respectivas categorias buscando mostrar os principais serviços da AWS, como *EC2*, *Relational Database Service (RDS)*, *S3* e *Identity and Access Management (IAM)*.

Os estudos designados para a última *Sprint* (Quadro 4) foram a respeito do tema central do estágio, que é sobre a área de *DevSecOps*, colocando em prática os conceitos aprendidos durante o estágio.

Quadro 4 - Primeira lista de estudos da Alura.

CURSO	TEMA	CARGA HORÁRIA
Conceitos <i>DevSecOps</i>	<i>The Twelve-Factor App</i> : Metodologia para construção de aplicações robustas	10h
Conceitos <i>DevSecOps</i>	<i>Git e Github</i> : Controle e compartilhe seu código	6h
Conceitos <i>DevSecOps</i>	<i>Git e Github</i> : Estratégias de ramificação, Conflitos e <i>Pull Requests</i>	8h
Conceitos <i>DevSecOps</i>	Integração Contínua: Mais qualidade e menos risco no desenvolvimento	6h
Conceitos <i>DevSecOps</i>	Entrega Contínua: Confiabilidade e qualidade na implantação de software	8h
Conceitos <i>DevSecOps</i>	<i>Jenkins e Docker</i> : <i>Pipeline</i> de entrega contínua	12h
Conceitos <i>DevSecOps</i>	<i>Jenkins</i> : Integração contínua	8h
Conceitos <i>DevSecOps</i>	<i>Gitlab CI e Docker</i> : <i>Pipeline</i> de entrega contínua	8h
Conceitos <i>DevSecOps</i>	<i>Maven</i> : Gerenciamento de dependências e build de aplicações Java	8h
Conceitos <i>DevSecOps</i>	<i>Build</i> de uma aplicação .NET: construindo seus projetos com <i>MSBuild</i>	10h
Conceitos <i>DevSecOps</i>	<i>Ansible</i> : Sua infraestrutura como código	12h
Conceitos <i>DevSecOps</i>	<i>Terraform</i> : Automatize a infraestrutura na nuvem	8h
Conceitos <i>DevSecOps</i>	<i>Grafana e Telegraf</i> : Monitoramento em tempo real	10h

Fonte: Autor.

- ***The Twelve-Factor App***: neste tópico aprendemos os princípios fundamentais para desenvolver aplicações que são escaláveis, flexíveis e fáceis de manter em ambientes *cloud*. A metodologia dos “Dozes fatores” aborda o conceito de construir aplicações modernas seguindo doze princípios, os quais são, *codebase*, dependências, configurações, *Backing Services*, builds e releases, processos, *Port binding*, concorrência, *Disposability*, *Dev/prod*, logs e processos de administração. O curso trouxe esses temas de forma simplificada buscando entender as vantagens e desvantagens, fluxos de eventos, monitorias, sistemas operacionais e conceitos de programação e infraestrutura.
- ***Git e Github***: o curso ensinou abrangentemente os fundamentos da ferramenta *Git*, *repositories*, *commits*, *branches*, merges e conflitos. Mostrando comandos básicos do *Git* para iniciar um repositório, adicionar e confirmar mudanças, criar e mesclar

branches, lidar com conflitos e revisar históricos de *commit*. No curso foi apresentado estratégias de Workflow como *git flow*, *cherry-pick*, *bisect*, dentre outros.

- **Integração Contínua:** apresentou-se o funcionamento de uma integração contínua na qual os desenvolvedores consolidam regularmente suas alterações de código em um repositório centralizado. Neste estudo vimos as vantagens e desvantagens que esse método pode trazer para o desenvolvimento de software. Reconhecendo sua importância na garantia da consistência e qualidade do código. Além disso, compreendem a interação entre testes e integração contínua, reconhecendo como os testes automatizados são fundamentais para o sucesso desse processo.
- **Entrega Contínua:** ele destaca a importância da automação no processo de integração de código, apresenta as principais ferramentas de CI, como Jenkins e GitLab CI/CD, e enfatiza a automação de testes e a análise de segurança como componentes essenciais da prática de CI. Além disso, discute como incorporar práticas de DevSecOps, como revisões de código focadas em segurança e integração de ferramentas de análise de vulnerabilidades, para garantir a entrega contínua de software de alta qualidade e segurança.
- **Jenkins e Docker:** o curso explorou a integração contínua utilizando as ferramentas de Docker e Jenkins e Git com Github. Aprendemos a configurar o Jenkins para automatizar builds contínuas, integrar o controle de versão com Git/GitHub, analisar a qualidade do código com o SonarQube e definir o pipeline como código. Com objetivo de capacitar a implementação e gerenciamento eficiente das práticas de integração contínua, enfatizando segurança e qualidade do código ao longo do ciclo de desenvolvimento de *software*.
- **Jenkins:** o curso sobre Jenkins ensina como configurar e gerenciar esta ferramenta de integração contínua. É apresentado como criar pipelines para automatizar tarefas como compilação, teste e implantação de software, integrar o Jenkins com outras ferramentas, gerenciar plugins, configurar segurança e acesso, melhorando assim a qualidade e eficiência do desenvolvimento de software.
- **Gitlab CI e Docker:** este curso ensina automatizar pipelines de integração e implantação contínua de software, integrado com a tecnologia de containers Docker. É visto como configurar pipelines para compilação, testes e distribuição de aplicações, utilizando Docker para garantir a consistência dos ambientes de desenvolvimento. Além disso, exploram práticas avançadas de automação, como testes de integração,

análise de código e implantação automatizada, melhorando o ciclo de vida de uma aplicação. (UNITY, S.D)

- **Maven:** com este curso vimos como fazer o gerenciamento de dependências em uma aplicação *Java*. Foi feito o gerenciamento das dependências utilizando *Maven*, uma ferramenta de build e gerenciamento de dependências para projetos *Java*. Aprendemos a configurar projetos *Maven*, gerenciar dependências de bibliotecas externas, compilar código-fonte, executar testes automatizados, empacotar e implantar aplicativos *Java*.
- **Build de uma aplicação .NET:** destaca a abordagem sobre o processo de construção de uma aplicação *.NET*, desde o entendimento da estrutura do arquivo de projeto *.CSPROJ* até a criação de uma imagem *Docker* para implantação. Os participantes aprenderão a personalizar o *script* de *build* usando o *MSBuild*, reutilizar eficientemente *scripts* com o *SDK .NET* e compreender as configurações de *debug* e *release* para otimizar o desempenho da aplicação.
- **Ansible:** vimos os fundamentos do Ansible, incluindo a instalação e configuração do ambiente, a criação de *playbooks* (templates) e *roles*, e a execução de tarefas de automação em larga escala em ambientes heterogêneos. Com este curso aprendemos a dominar conceitos avançados, como gestão de inventário dinâmico, integração com ferramentas de controle de versão e orquestração de infraestrutura em nuvem.
- **Terraform:** este tópico ofereceu uma introdução à infraestrutura como código (*IaC*) usando Terraform. Foram mostrados conceitos do Terraform, incluindo a sintaxe HCL (*HashiCorp Configuration Language*), e gestão de recursos em provedores de nuvem como *AWS*, *Azure* e *Google Cloud*, e as práticas recomendadas para criar, modificar e destruir infraestrutura de maneira eficiente e escalável, podendo ser automatizados a partir de *pipelines* de *CI/CD*. (TERRAFORM, 2023).
- **Grafana e Telegraf:** este curso foi uma introdução ao mundo do monitoramento de sistemas e aplicações. Configuramos e utilizamos o Telegraf, uma ferramenta de coleta de dados, para capturar métricas importantes de uma variedade de fontes, como servidores, bancos de dados e serviços em nuvem. Em seguida, exploramos o Grafana, uma plataforma de visualização de dados, para criar painéis dinâmicos e personalizados que apresentam essas métricas de forma clara e intuitiva. Introduzimos conceitos avançados, como alertas, funções de consulta e integração com outras ferramentas de monitoramento.

4 – CONSIDERAÇÕES FINAIS

Durante o período de estágio, pude vivenciar os impactos positivos que a aplicação de tecnologias inovadoras pode ter em processos, eficiência e resultados organizacionais. Além disso, ficou claro que a adaptação contínua a novas ferramentas e a busca por soluções tecnológicas alinhadas aos objetivos da empresa, são essenciais para manter a competitividade no mercado. Ao longo do período de estágio na *Compass* UOL, pude perceber o quão fundamental foi essa experiência para minha formação acadêmica e profissional.

Os treinamentos recebidos foram cruciais para compreender não apenas as atividades da empresa, mas também tive a oportunidade de aplicar os conhecimentos adquiridos durante minha graduação no curso de Tecnologia em Análise e Desenvolvimento de Sistemas no Instituto Federal Goiano - Campus Iporá.

Os quais muitas vezes pareciam nebulosos, e assim ganhei uma compreensão clara de como e quando aplicá-los em situações práticas. Essa conciliação entre o conhecimento acadêmico e a oportunidade de colocá-los em prática em um estágio contribuíram para suprir as necessidades fundamentais para o exercício da profissão no contexto tecnológico.

Uma consequência significativa dessa experiência foi a minha contratação pela *Compass* UOL no fim do estágio e com a contratação pude participar de projetos reais e aplicar os conceitos *DevSecOps*. Este estágio foi de suma importância, complementando meu currículo profissional que me proporcionou uma valiosa perspectiva sobre as adversidades diárias enfrentadas no campo da tecnologia.

Esta experiência contribuiu significativamente para minha trajetória profissional, destacando a importância da integração entre teoria e prática no contexto desafiador da tecnologia. Estou grato pela oportunidade e confiante no impacto positivo que essa vivência teve e terá em minha carreira futura.

5 – REFERÊNCIAS BIBLIOGRÁFICAS

AWS, 2023. **AMAZON WEB SERVICES**. Disponível em: <https://docs.aws.amazon.com>. Acesso em: 05/12/2023.

DOCKER, 2023. **Docker**. Disponível em: <https://docs.docker.com/reference>. Acesso em: 20/11/2023.

Lei nº 11.788, de 25 de setembro de 2008. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/111788.htm. Acesso em: 27/02/2024.

KUBERNETES, 2023. **KUBERNETES**. Disponível em: <https://kubernetes.io/pt-br/docs/home/>. Acesso em: 15/11/2023.

MARCELO, Antonio. **APACHE: Configurando o servidor WEB para Linux**. Brasport, 2005.

REDHAT, 2019. **What is open source?**. Disponível em: <https://www.redhat.com/en/topics/open-source/what-is-open-source>. Acesso em: 23/11/2023.

REDHAT, 2023. **O que é DevSecOps?**. Disponível em: <https://www.redhat.com/pt-br/topics/devops/what-is-devsecops>. Acesso em: 01/03/2024.

SACHDEVA, S. 2016, 6 de junho. **Scrum Methodology. International Journal of Engineering and Computer Science**. Disponível em: <http://www.ijecs.in/index.php/ijecs/article/view/1989/1839>

TERRAFORM, 2023. **TERRAFORM**. Disponível em: <https://developer.hashicorp.com/terraform/docs>. Acesso em: 20/11/2023.

UNIVERSIDADE EVANGÉLICA DE ANÁPOLIS. **Linux**. Goiás, 2018. 59 p. Disponível em: <http://45.4.96.19/bitstream/aee/1138/1/Linux.pdf>

UNITY, (s.d). **O que é CI/CD?**. Disponível em: [\[Link\]](#). Acesso em: 20/11/2023.