



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS MONTEIRO
DIRETORIA DE DESENVOLVIMENTO E ENSINO
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

EZEQUIAS SOARES DE OLIVEIRA

PROCESSO DE DESENVOLVIMENTO DE *SOFTWARE* NA EMPRESA *COMPASS*
***UOL*: UM RELATO DE EXPERIÊNCIA PROFISSIONAL**

MONTEIRO

2024

EZEQUIAS SOARES DE OLIVEIRA

**PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NA EMPRESA COMPASS
UOL: UM RELATO DE EXPERIÊNCIA PROFISSIONAL**

Relatório de Estágio apresentado à Coordenação de Estágio do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *Campus Monteiro*, como requisito parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS).

Orientadora: Profa. Dra. Larissa Lucena Vasconcelos.

MONTEIRO

2024

Dados Internacionais de Catalogação na Publicação – CIP
Bibliotecária responsável Porcina Formiga dos Santos Salgado CRB15/204
IFPB Campus Monteiro.

O48p Oliveira, Ezequias Soares de.

Processo de desenvolvimento de software na Empresa COMPASS
UOL: um relato de experiência profissional / Ezequias Soares de
Oliveira – Monteiro-PB. 2024.
39fls. : il.

Relatório de estágio (Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas) - Instituto Federal de Educação,
Ciência e Tecnologia da Paraíba - IFPB campus, Monteiro.

Orientadora: Prof^ª. Dra. Larissa Lucena Vasconcelos.

1. Software - desenvolvimento
2. Programas – JAVA - Sprint Book
3. Banco Dados I. Título .

CDU 004.42

EZEQUIAS SOARES DE OLIVEIRA

RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE NA EMPRESA COMPASS UOL

Relatório de Estágio apresentado à Coordenação de Estágio do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *Campus Monteiro*, como requisito parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS).

Aprovado em 24 de Setembro de 2024

BANCA EXAMINADORA

Documento assinado digitalmente
 LARISSA LUCENA VASCONCELOS
Data: 06/11/2024 08:12:27-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dra. Larissa Lucena Vasconcelos (Orientadora - IFPB)

Documento assinado digitalmente
 GILDO FERRUCIO SANTOS MAIA DANTAS
Data: 08/11/2024 17:08:59-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Gildo Ferrucio Santos Maia Dantas (Examinador - IFPB)

Documento assinado digitalmente
 DAMIAO RIBEIRO DE ALMEIDA
Data: 09/11/2024 19:41:18-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Damião Ribeiro de Almeida (Examinador - IFPB)

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por tudo que Ele me possibilitou viver e aprender durante o curso e o estágio. Expresso, também, minha gratidão a toda minha família, em especial aos meus pais (Maria José da Conceição Soares e Inácio Eudes Soares de Oliveira), que me incentivaram a não desistir dos meus sonhos.

Em segundo lugar, a todo o corpo docente do curso de ADS (IFPB/Monteiro), pela partilha de conhecimento técnico e orientação ao longo de minha formação acadêmica e profissional.

Em especial, agradeço à minha orientadora, a Professora Dra. Larissa Lucena Vasconcelos, junto a banca examinadora composta pelos professores Me. Gildo Ferrucio Santos Maia Dantas e Dr. Damião Ribeiro de Almeida, por todas as contribuições, empenho e paciência na leitura deste trabalho.

Grato, ainda, a todos os meus amigos e colegas que me acompanharam durante toda a trajetória da graduação.

Por último - e não menos importante, registro meus agradecimentos a todo corporativo da Compass UOL pelas portas abertas para vivências do estágio, cujos frutos são apresentados neste relatório.

RESUMO

Este relatório apresenta as atividades vivenciadas no estágio supervisionado na Compass UOL. Ao longo do estágio, o discente aplicou os conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas no ambiente corporativo da Compass UOL, atuando em projetos de *Back-end* com *Java*, *Spring Boot* e integração com bancos de dados como *PostgreSQL*, *MySQL* e *MongoDB*. Durante esse período, o discente desenvolveu e aprimorou suas habilidades técnicas, especialmente no desenvolvimento de *APIs RESTful*, testes automatizados com *JUnit* e *Mockito*, e uso de metodologias ágeis como *Scrum* e *Kanban*. O estágio também proporcionou o desenvolvimento de competências interpessoais, como trabalho em equipe, resolução de problemas e comunicação eficaz em ambientes corporativos dinâmicos. As experiências adquiridas ao longo deste período contribuíram significativamente para o crescimento profissional do discente e sua preparação para desafios futuros no mercado.

Palavras-chave: Desenvolvimento de *software*; *Java*; *Spring Boot*; Bancos de dados; Metodologias ágeis.

ABSTRACT

This report presents the activities experienced during the supervised internship at Compass UOL. Throughout the internship, the student applied the knowledge acquired in the Systems Analysis and Development course in the corporate environment of Compass UOL, working on Back-end projects with Java, Spring Boot and integration with databases such as PostgreSQL, MySQL and MongoDB. During this period, the student developed and improved his technical skills, especially in the development of RESTful APIs, automated tests with JUnit and Mockito, and the use of agile methodologies such as Scrum and Kanban. The internship also provided the development of interpersonal skills, such as teamwork, problem solving and effective communication in dynamic corporate environments. The experiences acquired throughout this period contributed significantly to the student's professional growth and their preparation for future challenges in the market.

Keywords: Software development; Java; Spring Boot; Databases; Agile methodologies.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Logomarca da COMPASS UOL | 12 |
| Figura 2 - Logomarca Microsoft Teams | 14 |
| Figura 3 - Exemplo de teste e unidade | 22 |
| Figura 4 - Exemplo de teste de integração | 23 |
| Figura 5 - Diagrama de entidade e relacionamento | 27 |
| Figura 6 - Diagrama de estados | 29 |

LISTA DE ABREVIATURAS

| | |
|-------|--|
| ADS | <i>Análise e Desenvolvimento de Sistemas</i> |
| API | <i>Interface de Programação de aplicações</i> |
| AMQP | <i>Advanced Message Queuing Protocol</i> |
| AWS | <i>Amazon Web Service</i> |
| COVID | <i>Corona Virus Disease</i> |
| CRUD | <i>Create, Read, Update, Delete</i> |
| IDE | <i>Interface de Desenvolvimento Integrado</i> |
| JPA | <i>Java Persistence API</i> |
| MVC | <i>Model (Modelo) View (Visão) Controller (Controle)</i> |
| ORM | <i>Mapeamento Objeto-Relacional</i> |
| PO | <i>Product Owner</i> |
| REST | <i>Representational State Transfer</i> |
| SGBD | <i>Sistema de Gerenciamento de Banco de Dados</i> |
| SQL | <i>Structured Query Language (Linguagem de Consulta Estruturada)</i> |
| TI | <i>Tecnologia da Informação</i> |
| UOL | <i>Universo Online</i> |
| WEB | <i>World Wide Web</i> |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO..... | 9 |
| 2 OBJETIVOS..... | 11 |
| 2.1 Objetivo geral..... | 11 |
| 2.2 Objetivos específicos..... | 11 |
| 3 RELATO DE ESTÁGIO..... | 12 |
| 3.1 Identificação da empresa..... | 12 |
| 3.1.1 Local do estágio e aspectos físicos da empresa..... | 12 |
| 3.1.2 Dados referentes ao estágio..... | 12 |
| 3.1.3 Perfil da empresa..... | 13 |
| 3.2 Atividades desenvolvidas..... | 13 |
| 3.2.1 Participação nas cerimônias do scrum..... | 14 |
| 3.2.2 Jornada Spring Boot..... | 15 |
| 3.2.3 Desenvolvimento de código..... | 25 |
| 4 CONSIDERAÇÕES FINAIS..... | 31 |
| REFERÊNCIAS..... | 33 |

1 INTRODUÇÃO

Com o avanço das novas tecnologias, muitas atividades que antes eram realizadas manualmente passaram por transformações significativas. A pandemia de COVID-19 acelerou ainda mais esse processo, levando muitas atividades a migrarem para o meio digital. Esse cenário impulsionou a criação de novos setores tecnológicos para atender às demandas do mercado atual, aumentando a busca por profissionais qualificados na área de Tecnologia da Informação (TI). Estima-se que até 2025, teremos um déficit de 530 mil vagas não preenchidas nessa área, segundo levantamento da Endeavor (2023) com dados da Brasscom (Associação das Empresas de Tecnologia da Informação e Comunicação) (2023).

Nesse contexto, é perceptível o aumento de pessoas que buscam especialização para suprir essa demanda, direcionando suas escolhas para universidades que oferecem cursos na área de tecnologia. Com intuito de aprimorar tanto o conhecimento teórico quanto o prático, visando estar preparado para atender às exigências do mercado de trabalho contemporâneo, cada vez mais exigente em relação a profissionais.

Diante disso, o estágio na área de Tecnologia da Informação (TI) tem se tornado uma oportunidade tanto para o desenvolvimento profissional do aluno quanto para sua inserção no mercado de trabalho. Além de aplicar na prática toda a base teórica adquirida na universidade, o estágio proporciona o desenvolvimento de conhecimento prático e experiências do ambiente corporativo. A vivência de trabalhos em equipe durante o estágio também contribui para o aprimoramento das habilidades de comunicação, liderança, entre outras. Portanto, o estágio é visto como um meio essencial de adquirir a experiência que o mercado exige dos profissionais, facilitando assim a entrada em novas oportunidades de trabalho. Com isso, o presente trabalho tem como intuito relatar as atividades realizadas e experiências adquiridas pelo discente durante o estágio supervisionado.

O estágio foi realizado na Compass UOL, a divisão de tecnologia do Grupo Universo Online (UOL). Seu foco principal está em projetar e construir plataformas digitais nativas, utilizando tecnologias de ponta para auxiliar empresas. Com mais de 27 anos de atuação no mercado, a empresa já possui uma extensa lista de clientes,

incluindo empresas renomadas como Itaú Unibanco, Bradesco e Ambev. Seus principais serviços prestados incluem desenvolvimento de *softwares*, consultoria em TI, experiência do usuário (*User Experience - UX*) inteligência artificial (IA), utilizando diversas tecnologias consolidadas no mercado, bem como algumas emergentes.

O discente atuou como estagiário *Back-end* no programa de bolsas no *Studios Modern Application & Mobile Apps* da Compass UOL. Esse estúdio é dedicado à criação e modernização de aplicações *web* e *mobile*. Durante o estágio, foram oferecidos treinamentos em diversas áreas, e o discente foi selecionado para participar do *Back-end Journey (Spring Boot) - AWS Cloud Context* (IBM Brasil, 2023; Amazon Web Services, 2023). Seu principal objetivo era especializar-se nas tecnologias *Java* com *Spring Boot* e alguns *frameworks*, além de adquirir conhecimento em computação em nuvem, utilizando treinamentos para obter a certificação da *Amazon Web Services (AWS)*.

Este relatório está dividido em quatro seções distintas. A primeira seção corresponde à introdução, onde é apresentado o contexto geral do relatório. A segunda seção aborda os objetivos gerais e específicos do documento. Na terceira seção, são descritas as atividades realizadas pelo discente durante o período de estágio, juntamente com uma pequena apresentação da empresa. Por fim, a quarta seção contém a conclusão e considerações finais sobre o estágio.

2 OBJETIVOS

Nesta seção são apresentados os objetivos gerais e específicos.

2.1 Objetivo geral

Desenvolver e aprimorar habilidades e experiências no desenvolvimento de *software*, focando na aplicação prática de tecnologias *Java* e outros *frameworks* em um ambiente corporativo da Compass UOL. Ambiente este que visa capacitar equipes de desenvolvimento, utilizando boas práticas e abordagens modernas auxiliando na criação de soluções de *software* escaláveis.

2.2 Objetivos específicos

- Participação nos Projetos em Equipe: Colaborar em projetos de *software* dentro de um ambiente corporativo, desenvolvendo a capacidade de trabalhar em equipe, comunicando-se de forma clara e contribuindo para o sucesso coletivo.
- Compreensão de Processos: Entender e aplicar processos de desenvolvimento de *software*, como metodologias ágeis e práticas de gestão de projetos, para garantir a entrega de soluções de qualidade dentro dos prazos estabelecidos.
- Desenvolvimento de Soft Skills: Fortalecer habilidades interpessoais, como comunicação, resolução de problemas e trabalho em equipe, essenciais para uma carreira de sucesso na área de desenvolvimento de *software*.

3 RELATO DE ESTÁGIO

Nesta seção serão abordadas informações correspondentes à empresa Compass UOL, além da descrição das atividades desenvolvidas durante o estágio supervisionado.

3.1 Identificação da empresa

3.1.1 Local do estágio e aspectos físicos da empresa

O estágio foi realizado de forma remota na empresa Compass UOL, que se destaca na oferta de projetar e construir plataformas digitais nativas. Como uma empresa de grande porte, possui vários escritórios no Brasil e internacionalmente, incluindo Brasília, São Paulo - SP, Estados Unidos (São Francisco) e Europa (Lisboa) (Compasso 2023). A sede principal no Brasil está localizada em Passo Fundo - RS (Rio Grande do Sul) na Rua Coronel Chicuta, 575, Andar 4 - Centro (Econodata, 2023). Na Figura 1, é possível visualizar a logomarca da empresa Compass UOL.

Figura 1- Logomarca da Compass UOL.



Fonte: (Aws.amazon, 2024)

3.1.2 Dados referentes ao estágio

O estágio supervisionado teve início no dia 05 de junho de 2023 e finalizou no dia 04 de novembro de 2023, totalizando aproximadamente 6 meses de duração. O estágio foi desenvolvido totalmente em formato *home office*, seguindo tendência pós-pandemia. A carga horária de atividades na empresa foi de vinte horas

semanais, sendo 4 horas diárias, das 13h30 às 17h30, perfazendo um total de 480 horas ao final do estágio.

O estágio foi realizado sob a mentoria e supervisão dos desenvolvedores Victor Renan Ferreira e Giovanni Marchi. O discente ingressou na empresa na vaga de Estágio/Bolsista em Desenvolvimento *Back-end, Journey (Spring Boot) - AWS Cloud Context*, no time da *Studios Modern Application & Mobile Apps*.

3.1.3 Perfil da empresa

A Compass UOL foi fundada em 2013, através da união da Compasso, empresa especializada em soluções de TI fundada em 1995 (Upf, 2017), com a UOL Diveo, divisão de tecnologia do UOL (Uol diveo, 2023). A empresa oferece soluções em quatro áreas principais de conhecimento: Plataformas digitais de Última geração, Engenharia de Análise de Marketing, Transformação Ágil em Escala e Experiências 3D imersivas em tempo real, as quais são subdivididas em 14 *Innovation Studios*, centros de excelência e inovação focados nas tecnologias mais atuais (Compass UOL, 2024).

3.2 Atividades desenvolvidas

Durante o período de estágio na empresa, as atividades do discente tiveram início com tarefas administrativas, incluindo a alocação em uma turma de mais 20 (vinte) outros estagiários e a divisão da equipe (*squad*) em cinco pessoas. Em seguida, foram apresentados os mentores e supervisores da trilha de desenvolvimento, além de uma introdução às atividades que seriam realizadas ao longo do estágio. Foi também apresentado todo o modelo de metodologia ágil que seria utilizado, que consistia no uso do *Scrum* em conjunto com o *Kamba*, explicando o funcionamento do processo do estágio nos meses seguintes. O *Microsoft Teams* foi utilizado para a realização da comunicação entre o time da Compass UOL e o discente. Essa plataforma é utilizada para a comunicação e colaboração no meio corporativo de muitas empresas, incluindo a Compass UOL. Ela oferece funcionalidades como chats, videoconferências e compartilhamento de arquivos, tornando-se uma ferramenta útil para o trabalho em equipe (Microsoft, 2024). A Figura 2 apresenta a logomarca do Microsoft Teams.

Figura 2 - Logomarca Microsoft Teams.



Fonte: Inforrede, 2024.

3.2.1 Participação nas cerimônias do scrum

A organização do fluxo de trabalho dentro da empresa era baseada e definida utilizando a metodologia *Scrum*, reconhecida por sua abordagem ágil e iterativa no gerenciamento de projetos. O *Scrum* é uma metodologia ágil cujo objetivo é auxiliar na gestão de projetos e concluir atividades com prazos definidos em um curto e fixo período de tempo. Funciona em ciclos chamados de *sprints*, que representam o período no qual o trabalho é realizado. Durante cada *sprint*, a equipe trabalha para completar um conjunto específico de atividades ou itens do *backlog* do produto. Ao final de cada ciclo, é realizada uma revisão, avaliação e o planejamento para a próxima *sprint* (Da Silva, 2015; Drumond, 2023).

Quanto à equipe do *Scrum*, é importante destacar que os times geralmente são pequenos, variando de cinco a seis pessoas, para um melhor andamento do trabalho (Drumond, 2023). Entretanto, no *Studios Modern Application & Mobile Apps*, onde o discente estagiou, o time era composto por um total de 22 estagiários.

Para gerenciar esse número elevado e otimizar o tempo e a organização dos membros, o time foi dividido em 4 *squads* de cinco pessoas. A cada semana, aleatoriamente, um dos membros atuava como *Scrum Master*, sendo responsável por relatar o progresso das atividades de cada membro para o *Product Owner* (PO) (Da Silva, 2015; Treasy, 2020).

As *sprints* tinham duração de duas semanas. Antes de iniciar cada *sprint*, era realizada uma *planning*, na qual todas as informações sobre as tarefas a serem realizadas eram discutidas e definidas. Geralmente, essas tarefas incluíam a divisão

de conteúdos a serem estudados e a implementação ou resolução de desafios. Ao final de duas *sprints*, o *Product Owner* (PO) propunha um desafio para avaliar o desempenho dos membros e o conhecimento adquirido durante o período tanto individual como em equipe.

Durante essas semanas, eram realizadas *daily*s (reuniões diárias) com a finalidade de acompanhar o progresso das atividades e identificar possíveis obstáculos na execução do trabalho. Geralmente, essas reuniões tinham duração de 15 minutos, conforme padrão do Scrum (Treasy, 2020).

Para a organização e visualização das atividades, era utilizado o quadro do Trello que é uma ferramenta visual utilizada na gestão de projetos e organização de tarefas (Ribeiro, 2022), ferramenta esta com que os integrantes já apresentavam mais familiaridade com o uso. Essa ferramenta proporciona uma maneira eficiente de acompanhar o fluxo de trabalho e garantir a transparência das atividades para todos os membros da equipe.

3.2.2 Jornada Spring Boot

A Jornada *Spring Boot*, é uma trilha de aprendizado disponibilizada pela Compass UOL para capacitar estagiários a trabalharem com a tecnologia *Java* aliada ao *framework Spring Boot* e outras tecnologias. Esta jornada é composta por módulos abrangentes e práticos, cobrindo desde os fundamentos teóricos até aspectos avançados do desenvolvimento de aplicações com *Spring Boot*, em conjunto com outras tecnologias e *frameworks* que contribuem para o desenvolvimento de aplicações *Back-end*. Essa trilha de estudos foi realizada durante todo o período de estágio e, de acordo com o tema abordado em cada módulo, foram desenvolvidas aplicações para testar o nível de conhecimento teórico e prático adquirido através do curso.

Na primeira semana do estágio, foi concedido acesso à plataforma Udemy, onde todo o treinamento ocorreria por meio de cursos preparatórios nas principais tecnologias utilizadas pelo estúdio. Todo o conteúdo foi organizado em formato de trilha e estava disponível apenas em inglês. Os temas a serem estudados foram divididos em períodos de duas semanas e na semana seguinte havia a realização dos desafios. Durante esse tempo, os estagiários tinham a tarefa de assistir às aulas

e colocar em prática os conhecimentos adquiridos por meio de exercícios e projetos práticos. Essa abordagem proporcionou uma imersão gradual e estruturada nas tecnologias utilizadas pela empresa, preparando os estagiários de forma eficaz para as atividades futuras.

- Git

O *Git* é um sistema de controle de versão distribuído amplamente utilizado no desenvolvimento de *software*. Ele permite que os desenvolvedores acompanhem as alterações feitas em seus projetos, colaborem com outras pessoas e revertam para versões anteriores, caso necessário (Silverman, 2013; Torvalds, 2024).

O curso oferecido, intitulado "*Git Complete: The definitive, step-by-step*" (Khan, 2022), ministrado pelos professores Jason Taylor e John Myers, proporcionou uma compreensão abrangente dos fundamentos do *Git*, desde a criação de repositórios locais e remotos até a aplicação dos principais comandos, tais como: *git add*, *git commit* e *git push*, entre outros. Além disso, explorou-se conceitos mais avançados, como a criação de *tags*, *releases*, o uso de *stash*, *reset* e *reflog* (Silverman, 2013). As atividades práticas incluídas no curso foram essenciais para consolidar o conhecimento adquirido e preparar-me para aplicá-lo em projetos futuros.

Durante o estágio, o *Git* desempenhou um papel fundamental como ferramenta central para o controle, gerenciamento e compartilhamento de nossas atividades com os colegas de equipe, além de facilitar a entrega do código ao supervisor. Foram propostas diversas situações, como a resolução de conflitos entre *commits* e *branches*, o gerenciamento de *branches* e a revisão de código, que permitiram aplicar os conceitos aprendidos no curso, demonstrando a utilidade e a relevância do conhecimento adquirido.

- Cibersegurança

Cibersegurança trata-se de um conjunto de procedimentos usado para proteger sistemas, redes, programas, equipamentos e dados de invasões que podem causar algum tipo de dano a pessoas físicas e jurídicas (Fencom, 2024).

O curso "*The Absolute Beginners Guide to Cyber Security 2024 - Part 1*" (ONI, 2024), ministrado pelo professor Alexander Oni, oferece uma introdução abrangente aos fundamentos da cibersegurança e segurança da informação. Inicialmente, o estagiário foi instruído sobre conceitos básicos, como protocolos de computadores e *cookies*, estabelecendo uma base sólida para o entendimento desses temas cruciais.

Em seguida, o curso explora os diferentes tipos de *hackers* e suas estratégias, bem como os métodos de ataque que empregam. Isso inclui uma análise detalhada de ataques de força bruta, *bots* e *botnets*, ataques de negação de serviço (*DoS*) e distribuídos (*DDoS*), além de técnicas como injeção *SQL*. Essa abordagem permite uma compreensão aprofundada dos variados riscos e ameaças presentes no cenário da cibersegurança.

Além disso, o curso oferece um breve conhecimento sobre os tipos de *malwares* mais comuns, como *trojans*, *spyware* e *ransomware*, incluindo seu funcionamento, detecção e prevenção. Também são apresentadas as defesas mais conhecidas e utilizadas no mercado, como *firewalls*, criptografia, biometria, antivírus, autenticação de dois fatores e gerenciamento de senhas.

Compreender esses princípios não só garantiu ao aluno confiança para lidar com a segurança desde o começo do estágio, mas também o permitiu aplicar práticas preventivas em relação à proteção de seus próprios dados e informações pessoais. Além disso, garantir a segurança das informações da empresa contribui para um ambiente mais protegido e confiável, promovendo a integridade e a confidencialidade dos dados.

- SQL

SQL ou Linguagem de Consulta Estruturada, é uma linguagem de comunicação utilizada para manipular dados em sistemas de gerenciamento de banco de dados relacionais. Em um banco de dados relacional, os dados são organizados em tabelas, com cada linha representando uma entrada específica e as colunas representando atributos ou características dessas entradas. Com o *SQL*, é possível executar uma variedade de operações, como inserir, atualizar, excluir e

pesquisar dados, além de otimizar o desempenho do banco de dados (Beaulieu, 2009; Tecmundo, 2024).

Durante o curso "*The Complete SQL Bootcamp: Go from Zero to Hero*" (Portilla, 2022), ministrado pelo professor José Portilla, foram apresentados conceitos fundamentais da linguagem SQL, bem como os primeiros comandos, como *SELECT*, *SELECT DISTINCT*, *COUNT*, *WHERE*, *ORDER BY*, *LIMIT*, *BETWEEN*, *IN* e *LIKE*. Além disso, o professor propôs um desafio utilizando o banco de dados *PostgreSQL*, oferecendo uma oportunidade prática de aplicar os conhecimentos adquiridos durante o curso (Postgresql, 2024).

Após essa etapa inicial, o curso se concentra em conceitos mais avançados do SQL, explorando a combinação de colunas por meio dos *JOINS*. Foi possível aprender a definir e aplicar diferentes tipos de *JOINS*, incluindo *INNER JOIN*, *FULL JOIN*, *LEFT JOIN*, *RIGHT JOIN* e *UNION*, permitindo unir dados de múltiplas tabelas de forma eficaz. Também foi abordado a manipulação de datas com funções de timestamps, a extração de informações específicas de datas e horas.

Além disso, foram exploradas as funcionalidades das funções matemáticas e de string, ampliando nossas capacidades para realizar cálculos e manipulações de texto dentro do SQL. Ao aprofundar o entendimento sobre operadores lógicos e aritméticos, foi possível realizar consultas mais complexas e elaborar critérios de seleção mais precisos. Outros tópicos abordados incluíram subconsultas, que nos permitiram realizar consultas dentro de outras consultas para obter resultados mais detalhados e específicos. Também foi explanado o conceito de auto-junção (*self-join*).

Ademais, foram abordadas as expressões condicionais, como *IF-THEN-ELSE* e *CASE*, que permitem realizar operações condicionais em nossas consultas. E, por fim, o curso versa sobre procedimentos, que são blocos de código SQL nomeados e armazenados no banco de dados para execução repetida, proporcionando maior modularidade e reutilização de código.

- Java

Java é uma linguagem de programação amplamente adotada para o desenvolvimento de aplicações *Web*. Utilizada há mais de duas décadas, tornou-se

uma escolha popular entre os desenvolvedores, com milhões de aplicações *Java* em uso atualmente. Conhecida por sua capacidade multiplataforma, orientação a objetos e foco em redes, essa linguagem oferece uma plataforma sólida para o desenvolvimento de uma ampla variedade de aplicações. Sua segurança e confiabilidade a tornam ideal para projetar desde aplicações móveis e *softwares* empresariais até soluções de *big data* e tecnologias de servidor (Horstmann, 2024; Aws, 2024).

Ao longo do curso "*Master Practical Java Development*", ministrado pelo instrutor Deovrat S (Deovrat S, 2024), foram apresentados os conceitos básicos da linguagem, incluindo tópicos fundamentais de orientação a objetos, como herança e polimorfismo. Esses assuntos são importantes no aprendizado da linguagem *Java*. Muitos desses conceitos básicos já haviam sido abordados durante o curso de ADS, proporcionando uma oportunidade de revisão e reforço do conhecimento.

Além disso, o curso aborda diferentes estruturas de dados como *HashSet*, *TreeMap*, *LinkedList* e *HashMap* que são utilizadas para organizar e manipular conjuntos de dados de forma eficiente. O *Java Shell*, também conhecido como *JShell*, foi outra ferramenta interessante apresentada no curso. Ele permite executar comandos *Java* diretamente no terminal.

No entanto, dois dos novos conceitos apresentados pelo curso são o uso de *lambdas* e *Streams API*. *Lambdas* são expressões de função anônimas que permitem tratar funções como objetos, tornando o código mais conciso e legível (Horstmann, 2024; Oracle, 2024 a). Já a *Streams API* oferece um conjunto de operações para processamento de dados de forma declarativa, permitindo realizar operações como filtragem, mapeamento e redução de forma eficiente e elegante (Oracle, 2024 b).

- Spring Boot

Após dominar os conceitos fundamentais de *Java*, o discente iniciou a aprendizagem do *framework Spring*, uma plataforma *open-source* que facilita o desenvolvimento de aplicações *Java* robustas e escaláveis. O *Spring Framework* oferece recursos como injeção de dependências, gerenciamento de transações e suporte a testes, mas geralmente exige configurações detalhadas (Spring

Framework, 2024). Nesta etapa, o foco do curso estava no *Spring Boot*, bem como em outras ferramentas como *MongoDB*, *RabbitMQ*, e outras tecnologias relevantes. O *Spring Framework* foi abordado apenas de forma introdutória, sem se aprofundar nas complexidades de configuração.

O *Spring Boot* foi desenvolvido como uma extensão do *Spring Framework*, fornecendo uma série de pré-configurações que permitem criar aplicações rapidamente. Com recursos como *auto-configuration*, que configura automaticamente componentes necessários com base nas bibliotecas presentes no projeto, e *starters*, que agrupam as dependências essenciais para funcionalidades específicas, o *Spring Boot* reduz significativamente o trabalho manual (Heckler, 2021).

Além disso, o *Spring Boot* inclui servidores *web* embutidos, como *Tomcat*, facilitando a execução direta da aplicação. O *Spring* também é projetado para ser pronto para produção, com suporte integrado a monitoramento e métricas, através do *Actuator*, tornando-o uma escolha prática para desenvolvedores que buscam combinar eficiência e simplicidade com a robustez do *Spring Framework* (Heckler, 2021; IBM, 2024).

Muitos dos conceitos abordados, tanto neste quanto nos tópicos seguintes, já haviam sido apresentados e desenvolvidos no curso de ADS, especialmente nas disciplinas de Desenvolvimento de Aplicações Corporativas (DAC) e Técnicas de Testes, proporcionando uma revisão e reforço do conhecimento teórico e prático.

- API Rest - SpringBoot

Ao fim dos conceitos iniciais do *Spring Boot*, o discente passou a explorar a criação de *APIs REST* usando o *framework*. O *Spring Boot* simplifica o desenvolvimento de *APIs* ao fornecer um conjunto de ferramentas e práticas recomendadas que facilitam a construção e a manutenção de serviços *WEB*. Além do *Spring WEB*, que oferece suporte para criar e gerenciar *endpoints HTTP*, o *Spring Boot* integra diversos componentes essenciais para o desenvolvimento de *APIs*.

No curso "*Building Real-Time REST APIs with Spring Boot* " (FADATARE, 2024) ministrado pelo professor Ramesh Fadatara, foram apresentadas as

ferramentas essenciais para trabalhar com *APIs* que auxiliam na construção de um sistema *WEB*, como: o *Spring Data JPA*, que simplifica o acesso a dados permitindo a criação de repositórios e a realização de operações de *CRUD* com menos código; o *Spring Security*, que fornece funcionalidades para autenticação e autorização, garantindo que a *API* esteja protegida contra acessos não autorizados; o *Spring Boot Actuator*, que oferece suporte para monitoramento e gerenciamento da aplicação em produção, permitindo a exposição de métricas e informações de saúde da aplicação (Heckler, 2021; Microsoft Azure, 2024) .

A partir dessa etapa, foram abordados conceitos cruciais para a construção de *APIs*, como o uso de controladores *@RestController* para definir *endpoints*, a criação de serviços para encapsular a lógica de negócios e o uso de repositórios para a manipulação de dados, utilizando o *Spring MVC*. O *Spring MVC* fornece a estrutura necessária para gerenciar a camada de apresentação e controlar o fluxo das requisições *HTTP*, permitindo uma integração eficiente com o *Spring Boot* para a construção de *APIs REST*. Esse *framework MVC* é fundamental para separar as responsabilidades entre a apresentação, a lógica de negócios e o acesso a dados, promovendo uma arquitetura mais organizada e modular. Além disso, foram discutidas boas práticas para a construção de uma arquitetura de sistemas escalável, destacando suas vantagens e desvantagens. Entre as arquiteturas exploradas, destacam-se:

- Arquitetura em camadas, que favorece a organização modular do código ao separar responsabilidades em diferentes camadas, como apresentação, lógica de negócios e acesso a dados (Fowler, 2012; Imasters,2022).
- Microsserviços, que permitem a criação de serviços independentes e especializados, escaláveis de forma autônoma, porém, demandam maior esforço no gerenciamento de comunicação entre serviços e controle da complexidade em sistemas distribuídos (Fowler, 2019; Marcelo, 2020).

Essas práticas e arquiteturas são importantes para garantir que a aplicação seja escalável e fácil de manter e expandir ao longo do tempo (Microsoft Azure,

2024).

- Testes - Spring Boot

Nessa etapa do treinamento, o entendimento da tecnologia *Spring* já estava consolidado. Assim, foi introduzido o desenvolvimento de testes por meio do curso "*Testing Spring Boot: Beginner to Guru*" (Gupta, 2024), ministrado pelo professor John Thompson. O curso apresentou brevemente os conceitos de testes de unidade, testes de integração e são técnicas de desenvolvimento de *software*. *TDD* (*Test-Driven Development*) — uma abordagem de desenvolvimento de *software* na qual os testes são escritos antes da implementação do código (Treinaweb, 2020) — e *BDD* (*Behavior-Driven Development*), que é uma evolução do *TDD* focada no comportamento do sistema sob a perspectiva do usuário ou do negócio (Astels, 2003; Coodesh, 2024).

Além dos conceitos de *TDD* e *BDD*, o curso destacou a importância do *JUnit* no desenvolvimento de testes automatizados. O *JUnit* é amplamente utilizado para testes unitários e de integração, permitindo validar o comportamento de classes e métodos de forma isolada ou em conjunto com outras partes do sistema. As anotações, conforme ilustra a Figura 3, tais como: `@Test`, `@BeforeEach` e `@AfterEach`, que ajudam a configurar o ambiente de testes antes e depois de cada execução, facilitando a criação de cenários de teste consistentes (Azevedo, 2018).

Figura 3 - Exemplo de teste de unidade *JUnit*.

```
@Test  ─ Ezequias Soares
void createPixKey() {
    PixKeyRequestDTO request = new PixKeyRequestDTO(PixKeyType.TELEPHONE, keyValue: "123456789");
    PixKeyResponseDTO response = pixKeyService.createPixKey(account.getId(), request);

    assertNotNull(response);
    assertEquals(request.keyType(), response.keyType());
    assertEquals(request.keyValue(), response.keyValue());
}
```

Fonte: Arquivo pessoal do autor.

A partir desse ponto, além do *JUnit*, foi apresentado também a utilização do *MockMvc* para realizar testes de integração com os controladores que atuam como

um intermediário entre a interface do usuário e a lógica de negócios em aplicações *Spring Boot*. O *MockMvc* permite testar endpoints *REST* de maneira simulada, sem a necessidade de iniciar todo o contexto da aplicação, tornando os testes mais rápidos e eficientes (Sinha, 2023).

Com o *MockMvc*, foi possível enviar requisições *HTTP* e verificar as respostas sem precisar rodar um servidor real. Essa abordagem facilita a validação de comportamentos de controladores, como as respostas de *status HTTP*, o corpo da resposta e as interações com os serviços da camada de negócios. O uso de métodos como *perform()* que é usado para simular a execução de uma requisição; o *andExpect()* utilizado para verificar se a resposta da requisição atende a determinadas expectativas ou condições e *content()* utilizado dentro de *andExpect()* para verificar o conteúdo da resposta *HTTP*, conforme exemplificado na Figura 4, com esses métodos é possível criar testes detalhados que asseguram o correto funcionamento dos *endpoints*.

Durante o curso, foi possível entender e combinar o *MockMvc* com o *JUnit* para criar cenários de testes mais robustos, como simular requisições *GET*, *POST*, *PUT* e *DELETE*, e verificar se as respostas atendiam aos requisitos esperados. Além disso, foi possível testar cenários de exceção e garantir que as respostas adequadas, como *404 Not Found* ou *400 Bad Request*, fossem retornadas nos casos de erro.

Figura 4 - Exemplo de teste de integração.

```
@Test
void testDeposit() throws Exception {
    mockMvc.perform(MockMvcRequestBuilders.post( uriTemplate: "/transaction/deposit")
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"accountType\": \"SAVINGS\", \"value\": 100 }"))
        .andExpect(status().isOk())
        .andExpect(jsonPath( expression: "$.valueTransaction", is( value: 100)))
        .andExpect(jsonPath( expression: "$.accountType", is( value: "SAVINGS")))
        .andExpect(jsonPath( expression: "$.transactionType", is( value: "DEPOSIT")));
}
```

Fonte: (Arquivo Pessoal).

- Mensageria - RabbitMQ

Com bons conhecimentos no *framework*, foi possível conhecer uma ferramenta importante para o desenvolvimento de aplicações *backend*: o conceito de mensageria. Esse tema foi abordado no curso “*RabbitMQ: Messaging with Java, Spring Boot And Spring MVC*”. O *RabbitMQ* é uma das principais ferramentas de mensageria utilizadas em sistemas distribuídos, permitindo a troca de mensagens de maneira assíncrona entre diferentes serviços ou aplicações agindo como um intermediário entre eles (Full Cycle, 2022).

Durante o curso, foi apresentada a integração do *RabbitMQ* com o *Spring Boot*, utilizando a biblioteca *Spring AMQP*, que facilita o envio e recebimento de mensagens, além de possibilitar o gerenciamento de filas (*queues*) e trocas (*exchanges*) de forma eficiente. O *Spring AMQP* abstrai grande parte da complexidade envolvida na comunicação entre serviços, tornando a implementação de mensageria muito mais simples.

A configuração envolve o uso de classes anotadas, como *@Bean* para definir filas e *exchanges*, e o *RabbitTemplate* para o envio de mensagens. Além disso, o consumo de mensagens foi implementado utilizando a anotação *@RabbitListener*, que automatiza a recepção de mensagens em filas específicas, permitindo a construção de sistemas altamente escaláveis e desacoplados.

Essa etapa proporcionou uma compreensão sólida sobre como utilizar o *RabbitMQ* para implementar comunicação assíncrona em aplicações *Java*, integrando os conceitos aprendidos no *Spring Boot* com a prática de mensageria, essencial para o desenvolvimento de sistemas robustos e escaláveis.

- Certificação AWS

Foi incluído, junto com os desafios, o estudo para a certificação *AWS Certified Cloud Practitioner*, um exame que avalia os conhecimentos básicos sobre a *AWS* e computação em nuvem. Essa certificação é amplamente reconhecida na indústria e pode aumentar significativamente as chances de conseguir um emprego, devido à alta demanda por profissionais com conhecimento em computação na nuvem, um setor que vem crescendo bastante nos últimos anos (Tiinside, 2024).

Foram sugeridos cursos disponíveis na Udemy à qual o discente utilizou como material para seu estudo, a saber: "*Ultimate AWS Certified Cloud Practitioner CLF-C02*" (Maarke, 2024), de Stephane Maarek, e "*Certificação Amazon AWS Certified Cloud Practitioner CLF-C02*" (Iacono,2024), de André Iacono, para que o discente pudesse estudar e realizar simulados como preparação para o exame. Todas as despesas relacionadas ao custo do exame foram custeadas pela Compass UOL.

Durante a preparação, o discente se concentrou em tópicos fundamentais, como os conceitos da nuvem, segurança e conformidade, serviços básicos e os aspectos econômicos da nuvem AWS. Em termos de serviços, foram estudados recursos essenciais como o *Amazon EC2 (Elastic Compute Cloud)*, que oferece capacidade de computação escalável, permitindo a execução de aplicações em servidores virtuais; o *Amazon S3 (Simple Storage Service)*, utilizado para armazenamento seguro e escalável de dados; e o *Amazon RDS (Relational Database Service)*, que facilita a configuração, operação e escalabilidade de bancos de dados relacionais na nuvem. O *RDS* suporta várias instâncias de bancos de dados, como *MySQL*, *PostgreSQL*, e também opções exclusivas da *AWS*, como o *Amazon Aurora*, projetado para oferecer alta performance e disponibilidade.

Além disso, o estudo incluiu práticas de segurança e conformidade, essenciais para garantir a proteção dos dados na nuvem e o cumprimento de normas regulatórias. Também foi abordada a compreensão dos preços e modelos de faturamento da *AWS*, proporcionando uma visão detalhada sobre os custos envolvidos e como otimizar o uso dos serviços.

Por fim, estratégias de suporte e melhores práticas foram analisadas, visando a utilização eficiente dos recursos oferecidos pela plataforma. Como resultado de todo esse preparo, o discente obteve sucesso na obtenção da certificação *AWS Certified Cloud Practitioner*, conforme comprovação no **Anexo A**.

3.2.3 Desenvolvimento de código

Durante todo o processo de estudo das tecnologias, foram realizados desafios práticos alinhados ao conteúdo apresentado em cada *sprint*. Esses desafios permitiram a aplicação dos conceitos aprendidos, proporcionando uma

experiência prática no desenvolvimento de código e na solução de problemas reais dentro das tecnologias abordadas.

Os desafios eram geralmente alternados entre atividades individuais e em equipe, o que permitiu ao discente desenvolver habilidades interpessoais (*Soft skills*) e adquirir mais experiência fora do ambiente acadêmico, como trabalho em equipe, boa comunicação e a aplicação das metodologias ágeis estudadas em sala de aula. Todos os desafios passaram por uma avaliação técnica conduzida por um dos responsáveis, geralmente o *PO*, que fornecia *feedback* sobre melhorias, tanto em habilidades técnicas. Ao todo foram passados 3 desafios técnicos de acordo com o nível de avanço no conhecimento teórico que o discente havia conseguido absorver durante as semanas de estudos.

A comunicação entre o discente e os demais membros da equipe foi realizada por meio da plataforma *Discord*, que é uma ferramenta de comunicação de voz e *chat* amplamente utilizada, especialmente em jogos (Agência Brasil, 2024). Como o discente e os demais membros já estavam familiarizados com essa plataforma, ela se tornou a principal fonte de comunicação entre eles.

Para o compartilhamento do código entre o discente, equipe e entrega de atividades para os líderes, foi utilizado o *GitHub*, que é uma plataforma de hospedagem de código e arquivos que implementa controle de versão por meio do *Git*. Essa ferramenta é amplamente adotada por desenvolvedores para facilitar a colaboração em projetos de *software*, permitindo o rastreamento de alterações e a gestão eficiente de versões (Hostinger, 2024).

Para o desenvolvimento dos projetos e desafios, foi utilizada a *IDE IntelliJ IDEA*, devido ao fato de o discente e a equipe já estarem habituados a sua utilização. É uma das mais populares ferramentas para desenvolvimento em *Java* e outras linguagens, oferecendo uma série de recursos avançados, como integração com *frameworks*, suporte a testes e depuração de código, facilitando o trabalho dos desenvolvedores (Jetbrains, 2024).

Após o aprendizado proporcionado pelo curso de *Java*, foi proposto, na semana seguinte, um primeiro desafio individual com requisitos específicos e um prazo de entrega de uma semana. A proposta envolveu a criação de uma aplicação simples, utilizando conceitos de orientação a objetos, como polimorfismo e aplicação

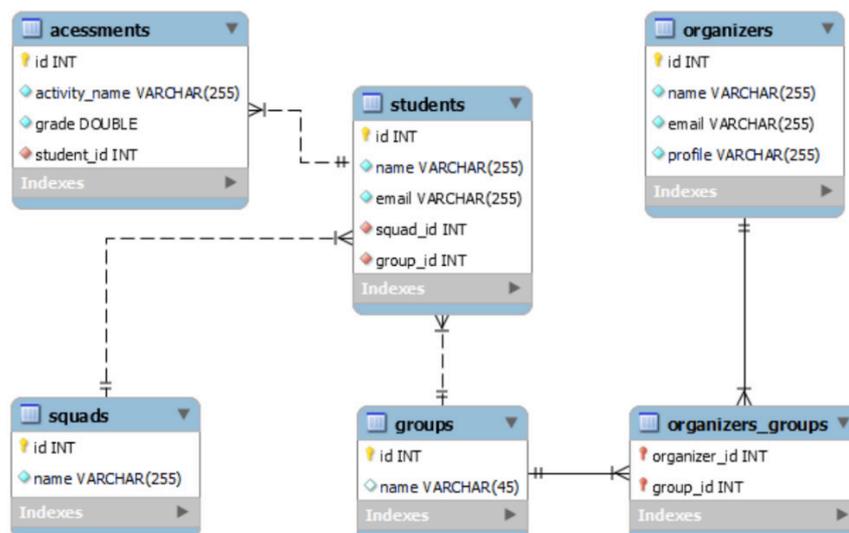
de interfaces, com o objetivo de avaliar o nível de conhecimento adquirido, as técnicas aplicadas e a qualidade do código. A experiência prática adquirida no ambiente acadêmico foi fundamental para o desenvolvimento desse desafio, no qual o discente não encontrou dificuldades.

Além disso, o discente já possuía familiaridade com o desenvolvimento de aplicações usando *Spring Boot*, incluindo a estruturação de projetos Web, o uso de *ORM* com *Hibernate* e a configuração básica do *framework*. Essa base sólida permitiu enfrentar os próximos desafios com maior segurança e agilidade, consolidando ainda mais o conhecimento nas tecnologias envolvidas.

No segundo desafio, foram introduzidos conceitos mais avançados, como boas práticas de desenvolvimento e arquitetura. Esse desafio envolveu o desenvolvimento da aplicação “*Scholarship Registration*”, que permitia registrar os alunos do estágio, a área em que estavam alocados, além dos relacionamentos com outras entidades, como a equipe em que participavam, as atividades realizadas e os organizadores responsáveis pela equipe. Na Figura 05 observa-se um exemplo do sistema representado através de um diagrama:

Figura 05 - Diagrama entidade e relacionamento.

Register of Compass Scholarship Program classes



Fonte: (Arquivo Pessoal).

Este desafio foi realizado em equipe, e o discente foi responsável pela criação de toda a arquitetura dos testes, além da implementação de alguns casos de teste e

do desenvolvimento de endpoints. Durante a avaliação, foram observados a qualidade e a organização do código, a arquitetura empregada, o desenvolvimento da solução, bem como o gerenciamento da equipe, incluindo a divisão das tarefas e a comunicação entre os membros. O desafio envolveu a criação de uma aplicação Spring completa, cobrindo os requisitos de uma aplicação *RESTful*, com os métodos *GET*, *POST*, *PUT* e *DELETE*, conforme os requisitos propostos no desafio. O projeto atingiu uma cobertura de testes de 95%, superando o requisito mínimo de 90%. As estatísticas de cobertura de testes foram obtidas através do *plugin Coverage* do *IntelliJ*, que ajuda a determinar a parcela de código coberta pelos testes e a identificar áreas com cobertura insuficiente (Jetbrains, 2024). Foram construídos testes de unidade com *JUnit* e integração com o *MockMvc*.

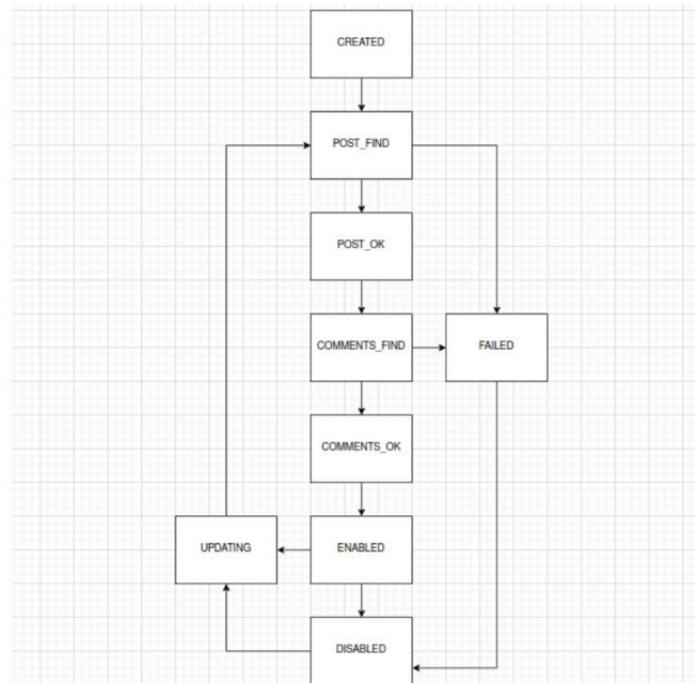
Além disso, a documentação foi realizada por meio do *Swagger*, uma ferramenta que facilita a criação, visualização e interatividade com *APIs RESTful*. O *Swagger* permite gerar documentação automática baseada no código, fornecendo uma interface amigável para testar e explorar os endpoints disponíveis da *API* (Gr1d, 2022).

Outra tecnologia utilizada foi o *HATEOAS (Hypermedia As The Engine Of Application State - Hipermídia Como Motor do Estado da Aplicação)*. Ele permite que a *API* forneça links contextuais junto aos dados de resposta, possibilitando a navegação entre os recursos de forma dinâmica. Com isso, os clientes da *API* podem descobrir facilmente quais ações são permitidas em cada recurso sem precisar consultar a documentação externa, promovendo um maior grau de desacoplamento entre o cliente e o servidor (Maciratti, 2024).

No terceiro desafio, realizado de forma individual, foram apresentados conceitos avançados de processamento assíncrono e integração com *APIs* externas utilizando a ferramenta *Feign Client*, que facilita a comunicação entre serviços em arquiteturas de microsserviços (ALVARO, 2019). Seguindo os mesmos critérios de avaliação dos desafios anteriores, o objetivo foi desenvolver uma aplicação que buscava posts de uma *API* chamada *JSONPlaceholder* que é uma plataforma para testes de *API*, que simula publicações de usuários em redes sociais que podem ser consumidas por outros clientes. De forma assíncrona, a aplicação enriquecia esses posts com dados de comentários e mantinha um *log* de atualizações do

processamento. A aplicação também permitia que o cliente buscasse os posts e o histórico de estados por meio da *API* em tempo real. Na Figura 06 é ilustrado um sistema representado através de um diagrama de estados:

Figura 6 - Diagrama de estados.



Fonte: (Arquivo Pessoal).

Esse desafio foi realizado de forma individual. O desafio envolveu a criação de uma aplicação *Spring Boot*, cobrindo os requisitos de uma *API RESTful* com métodos *GET*, *POST*, *PUT* e *DELETE*, conforme o proposto. Além disso, foi utilizado o banco de dados *H2* para simular o ambiente de desenvolvimento.

Durante o desenvolvimento, foi implementado um mecanismo de controle de estados para os posts, que transitavam entre estados como "*CREATED*", "*POST_OK*", "*COMMENTS_OK*" e "*ENABLED*", conforme as diferentes etapas do processamento. Essa funcionalidade garantiu que a aplicação pudesse lidar com um grande volume de requisições simultâneas de forma eficiente, assegurando a consistência dos dados processados.

Os métodos da *API* e suas funções no contexto do sistema incluem:

- Processar Post (*POST* /posts/{postId}): Esse método é responsável por processar e registrar um novo post na aplicação. O postId é validado e, se ainda não estiver presente no sistema, a aplicação busca os dados do post de uma *API* externa, enriquece com comentários e mantém um registro do estado de processamento.
- Desativar Post (*DELETE* /posts/{postId}): Esse método permite desativar um post que está no estado "*ENABLED*". Ele muda o estado do post para "*DISABLED*", garantindo que ele não será mais exibido nas consultas normais até ser reprocessado.
- Reprocessar Post (*PUT* /posts/{postId}): Esse método é usado para reprocessar um post que já foi ativado ou desativado. Ele atualiza os dados do post e seus comentários, permitindo que ele volte ao estado "*ENABLED*" ou seja novamente processado em caso de erro ou necessidade de atualização.
- Consultar Posts (*GET* /posts): Esse método retorna uma lista de posts disponíveis, fornecendo detalhes sobre o estado atual de cada post, como título, corpo, comentários e o histórico completo de estados pelos quais o post passou durante o processamento.

Ao longo do desenvolvimento das aplicações, foi facultado ao discente e à equipe a escolha do banco de dados a ser utilizado para o gerenciamento das tabelas e dados das aplicações, de acordo com a necessidade e o padrão em que os dados seriam armazenados. Para o armazenamento desses dados, o banco mais utilizado foi o *PostgreSQL*, reconhecido por sua robustez e eficiência em sistemas relacionais, além de serem amplamente adotados por grandes corporações (4Linux, 2024).

Para dados não estruturados, a escolha predominante foi o *MongoDB*, que proporcionou maior flexibilidade no armazenamento e já havia sido utilizado em demonstrações nos cursos estudados, porém não foi utilizado nos desafios. Um dos fatores determinantes na escolha dos bancos de dados foi o conhecimento e a experiência da equipe no uso dessas tecnologias. Na estrutura de testes, foi

utilizado o banco de dados *H2*, que opera em memória e, por isso, não exige uma estrutura complexa, sendo ideal para a fase de testes (Barancelli, 2022).

4 CONSIDERAÇÕES FINAIS

Ao longo do estágio, o discente teve a oportunidade de aplicar diversos conhecimentos teóricos na prática, aprimorando suas habilidades técnicas na área de *Back-end* com *Java* e *Spring Boot*. O estágio permitiu um aprofundamento do conhecimento sobre a construção e integração de *APIs* através dos conhecimentos e desafios passados ao longo do estágio, além da implementação de boas práticas de arquitetura e segurança. A experiência também proporcionou um contato valioso com metodologias ágeis, que contribuíram para uma melhor organização e execução das tarefas diárias, aspectos essenciais para a eficiência no ambiente profissional.

O desempenho durante o estágio foi reconhecido pela empresa, o que resultou em sua efetivação como Programador *Trainee*. Essa promoção reflete o comprometimento, a evolução técnica e a capacidade de trabalhar em equipe demonstrados ao longo do período de estágio. A efetivação permitiu ao discente assumir maiores responsabilidades e continuar a aprimorar suas habilidades, além de ingressar na área profissional antes mesmo da conclusão e defesa do Trabalho de Conclusão de Curso para obtenção de grau.

Assim sendo, a experiência ajudou a desenvolver a comunicação em equipe, uma habilidade essencial para evitar mal-entendidos e garantir que todos estejam alinhados com os objetivos do projeto. Os desafios enfrentados, desde a utilização de novas tecnologias até a preparação para a certificação *AWS Certified Cloud Practitioner*, foram cruciais para o crescimento pessoal e profissional do discente. O estágio reforçou a importância de aspectos como testes de *software*, segurança e otimização de recursos na construção de soluções robustas e escaláveis.

Essas experiências não apenas enriqueceram o conhecimento técnico, mas também fortaleceram a capacidade de trabalhar em equipe e solucionar problemas de forma eficiente. O conhecimento obtido no ambiente acadêmico, aliado à

experiência prática adquirida no estágio e como Programador *Trainee*, tornou-se um diferencial significativo e facilitou a entrada definitiva do discente no mercado de trabalho.

Em suma, a experiência no transcurso do estágio foi fundamental para a evolução profissional do discente, preparando-o para enfrentar novos desafios no mercado com mais confiança e habilidades aprimoradas. A vivência prática, somada ao suporte da empresa e ao aprendizado acadêmico, representa um marco importante na trajetória do discente.

REFERÊNCIAS

4LINUX. **Os principais bancos de dados open source do mercado.** 2023.

Disponível em: <https://4linux.com.br/principais-banco-dados-open-source/>. Acesso em: 17 set. 2024.

AGÊNCIA BRASIL. **Saiba o que é o Discord, aplicativo popular entre gamers.**

Disponível em:

<https://agenciabrasil.ebc.com.br/geral/noticia/2023-06/saiba-o-que-e-o-discord-aplicativo-popular-entre-gamers>. Acesso em: 20 set. 2024.

ALVARO, Fábio. **Simplificando chamadas HTTP com Feign Client no Spring Boot.** Medium, 2019. Disponível em:

<https://medium.com/@fabio.alvaro/simplificando-chamadas-http-com-feign-client-no-spring-boot-7926bdfad6d3>. Acesso em: 12 set. 2024.

AMAZON WEB SERVICES. **O que é a AWS?**. [S.l.], 2023. Disponível em:

https://aws.amazon.com/pt/what-is-aws/?nc1=f_cc. Acesso em: 14 mar. 2024.

AWS. **O que é Java?** [S.l.], 2024. Disponível em:

<https://aws.amazon.com/pt/what-is/java/>. Acesso em: 30 abr. 2024.

AWS.AMAZON. **Logomarca da Compass UOL.** [S.l.], 2016. Disponível em:

<https://aws.amazon.com/marketplace/seller-profile?id=c859e2b3-99ed-4dad-8ca5-5bbed6556ff0>. Acesso em: 14 mar. 2024.

Azevedo, M. **Qualidade e JUnit: introduzindo automatização de testes unitários do seu software Java no dia a dia.** Disponível em:

<https://mari-azevedo.medium.com/qualidade-e-junit-introduzindo-automatiza%C3%A7%C3%A3o-de-testes-unit%C3%A1rios-do-seu-software-java-no-dia-a-dia-849611de5574>. Acesso em: 12 set. 2024.

ASTELS, Dave. **Test driven development: A practical guide.** Prentice Hall Professional Technical Reference, 2003.

BARANCELLI, Gaspar. **Banco de dados H2 com Spring Boot.** 2022. Disponível

em: <https://gasparbarancelli.com/post/banco-de-dados-h2-com-spring-boot>. Acesso em: 17 set. 2024.

BEAULIEU, Alan. **Learning SQL: master SQL fundamentals.** " O'Reilly Media, Inc.", 2009.

BLOCH, Joshua. **Effective java.** Addison-Wesley Professional, 2017.

BRASSCOM. **Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação**. Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025. Disponível em: <https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>. Acesso em: 12 mar. 2024.

BRAGA, Marcelo. **Como o estágio prepara você para a vida profissional?** Blog de recrutamento e seleção da Reachr. 14 set. 2021. Disponível em: <https://www.blogger.com/>. Acesso em: 12 mar. 2024.

COMPASS UOL. **Home**. 2023. Disponível em: <https://compass.uol.pt/home/>. Acesso em: 13 mar. 2024.

COMPASS UOL. **Innovation Studios**. [S.l.], 2024. Disponível em: <https://compass.uol.pt/innovation-studios/>. Acesso em: 14 mar. 2024.

COMPASS UOL. **Sobre nós**. 2023. Disponível em: <https://compass.uol.pt/sobre-nos/>. Acesso em: 14 mar. 2024.

COODESH. **O que é BDD?** Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-bdd/>. Acesso em: 10 set. 2024.

DEV.TO. **Como retornar dados paginados no Spring Boot**. Disponível em: <https://dev.to/brunbs/como-retornar-dados-paginados-no-spring-boot-mhl>. Acesso em: 2 set. 2024.

DAVE ASTELS. 2003. **Test Driven development: A Practical Guide**. Prentice Hall Professional Technical Reference.

DEVMEDIA. **Como Começar com Spring: Um Guia para Iniciantes**. Disponível em: <https://www.devmedia.com.br/exemplo/como-comecar-com-spring/73>. Acesso em: 13 mai. 2024.

DA SILVA, Salmo Roberto; PIRES, Daniel Facciolo; NETO, **Silvio Carvalho**. **Scrum: um guia prático no gerenciamento de projetos**. Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica, v. 5, n. 1, 2015.

DRUMOND, Claire. **O que é scrum e como começar**. Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em: 01 abr. 2024.

ECONODATA. **Consulta de empresas**. Disponível em: <https://www.econodata.com.br/consulta-empresa/07654824000124-COMPASSUOL-TECNOLOGIA-LTDA>. Acesso em: 14 mar. 2024.

ENDEAVOR. **Profissionais de Tecnologia**. Disponível em: <https://endeavor.org.br/tecnologia/profissionais-de-tecnologia/>. Acesso em: 10 mar. 2023.

ENGADGET. **Discord's new Stage Discovery portal lets you find live audio events**. Disponível em: <https://www.engadget.com/discord-stage-discovery-portal-170039853.html>. Acesso em: 20 set. 2024.

FADATARE, Ramesh. **Building Real-Time REST APIs with Spring Boot**. Disponível em: <https://www.udemy.com/course/building-real-time-rest-apis-with-spring-boot/?couponCode=OF83024E>. Acesso em: 21 set. 2024.

FOWLER, Susan J. **Microserviços prontos para a produção: Construindo sistemas padronizados em uma organização de engenharia de software**. Novatec Editora, 2019.

FOWLER, Martin. **Patterns of enterprise application architecture**. Addison-Wesley, 2012.

FENCOM - Federação Nacional das Cooperativas Médicas. **O que é cibersegurança?**. [S.l.], Disponível em: <https://fencom.coop.br/o-que-e-ciberseguranca/>. Acesso em: 29 abr. 2024.

FLANE. **Como as certificações AWS se mostram um investimento de retorno garantido para sua carreira**. Disponível em: <https://www.flane.com.pa/blog/es/como-as-certificacoes-aws-se-mostram-um-investimento-de-retorno-garantido-para-sua-carreira/>. Acesso em: 17 set. 2024.

FULL CYCLE. **Como funciona o RabbitMQ?** Disponível em: <https://fullcycle.com.br/como-funciona-o-rabbitmq/>. Acesso em: 12 set. 2024.

GR1D. **O que é Swagger e como ele facilita a criação de APIs?** 2022. Disponível em: <https://gr1d.io/2022/04/15/swagger/>. Acesso em: 16 set. 2024.

GUPTA, Ranga Rao. **Testing Spring Boot: Beginner to Guru**. Disponível em: <https://www.udemy.com/course/testing-spring-boot-beginner-to-guru/?couponCode=OF83024E>. Acesso em: 21 set. 2024.

HOSTINGER. **O que é GitHub?** Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-github>. Acesso em: 20 set. 2024.

HORSTMANN, Cay S. **Core java, volume I: fundamentals**. Pearson Education, 2024.

HECKLER, Mark. **Spring Boot: Up and Running**. O'Reilly Media, 2021.

IACONO, Andre. **Certificação Amazon AWS Cloud Practitioner (CLF-C02)**.

Disponível em:

<https://www.udemy.com/course/certificacao-amazon-aws-cloud-practitioner-clf-c02/>.

Acesso em: 21 set. 2024.

IBM BRASIL. **O que é Java Spring Boot?**. Disponível

em:<https://www.ibm.com/br-pt/topics/java-spring-boot>. Acesso em: 14 mar. 2024.

IBM. (2024). **Java Spring Boot**. Disponível em:

<https://www.ibm.com/br-pt/topics/java-spring-boot>. Acesso em: 03 de setembro de 2024.

IMASTERS. **Arquitetura em camadas**. Disponível em:

<https://imasters.com.br/arquitetura-da-informacao/arquitetura-em-camadas>. Acesso em: 03 set . 2024.

INFORREDE. **Microsoft Teams**. Disponível em:

<https://www.inforrede.com.br/solucoes-microsoft/microsoft-teams/>. Acesso em: 19 set. 2024.

JETBRAINS. **Code coverage**. Disponível em:

https://www.jetbrains.com/help/idea/code-coverage.html#run_with_coverage. Acesso em: 17 set. 2024.

JETBRAINS. **IntelliJ IDEA: The Leading Java and Kotlin IDE**. Disponível em:

<https://www.jetbrains.com/idea/>. Acesso em: 20 set. 2024.

KHAN, Jason Taylor. **Git Complete: The definitive, step-by-step guide to Git**.

Udemy, 2024. Disponível em: <https://www.udemy.com/course/git-complete>. Acesso em: 20 set. 2024.

MAAREK, Stephane. **AWS Certified Cloud Practitioner**. Disponível em:

<https://www.udemy.com/course/aws-certified-cloud-practitioner-new/>. Acesso em: 21 set. 2024.

MACORATTI, J. **HATEOAS: Modelo para desenvolvimento de APIs RESTful**.

Disponível em: https://www.macoratti.net/19/05/net_hateoas1.htm. Acesso em: 16 set. 2024.

MARCELO, M. **Arquitetura de Microsserviços**. Medium. Disponível em: <https://medium.com/@marcelomg21/arquitetura-de-microsservi%C3%A7os-bc38d03fbf64>. Acesso em: 03 set. 2024.

MICROSOFT AZURE. **What is Java Spring Boot**. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-spring-boot>. Acesso em: 03 set. de 2024.

MICROSOFT. **Introdução ao Microsoft Teams**. Disponível em: <https://support.microsoft.com/pt-br/office/introdu%C3%A7%C3%A3o-ao-microsoft-teams-b98d533f-118e-4bae-bf44-3df2470c2b12>. Acesso em: 19 set. 2024.

ONI, Alexander. **The Absolute Beginners Guide to Cyber Security 2024 - Part 1**. Udemy, 2024. Disponível em: <https://www.udemy.com/course/the-absolute-beginners-guide-to-information-cyber-security/>. Acesso em: 20 set. 2024.

PORTILLA, José. **The Complete SQL Bootcamp: Go from Zero to Hero**. Udemy, 2024. Disponível em: <https://www.udemy.com/course/the-complete-sql-bootcamp/>. Acesso em: 20 set. 2024.

ORACLE a. **Java SE 8 Tutorial: Lambda Expressions**. Disponível em: <https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/lambda-quickstart/index.html>. Acesso em: 13 mai. 2024

ORACLE b. **Java SE 8: Streams API**. Disponível em: <https://www.oracle.com/technical-resources/articles/java/ma14-java-se-8-streams.html>. Acesso em: 13 mai. 2024.

POSTGRESQL. **O portal oficial do PostgreSQL: o poderoso sistema de gerenciamento de banco de dados relacional de código aberto**. [S.l.], 2024. Disponível em: <https://www.postgresql.org/>. Acesso em: 29 abr. 2024.

RIBEIRO, A. L. S. **Trello**. Alura, [s.l.], 16 dez. 2022. Disponível em: <https://www.alura.com.br/artigos/trello>. Acesso em: 04 ago. 2024

S., Deovrat. **Master Practical Java Development**. Udemy, 2024. Disponível em: <https://www.udemy.com/course/master-practical-java-development/>. Acesso em: 20 set. 2024.

SINHA, A. **Spring Boot JUnit**. Disponível em: <https://sinhaanwasha16.medium.com/spring-boot-junit-1b4544cb8bac>. Acesso em: 12 set. 2024.

SILVERMAN, Richard E. **Git Pocket Guide: A Working Introduction.** " O'Reilly Media, Inc.", 2013.

SPRING FRAMEWORK. **Spring Framework.** Disponível em: <https://spring.io/projects/spring-framework>. Acesso em: 16 mar. 2024.

Studios: Modern Application & Mobile Apps. 2023. Disponível em: <https://compass.uol/pt/studios/modern-application-mobile-apps/>. Acesso em: 14 mar. 2024

TECMUNDO. **SQL: o que é e para que serve.** [S.l.], 2024. Disponível em: <https://www.tecmundo.com.br/software/146482-sql-que-ele-serve.htm>. Acesso em: 29 abr. 2024

TORVALDS, Linus. **Git: um sistema de controle de versão distribuído.** Disponível em: <https://git-scm.com/>. Acesso em: 23 de abril de 2024.

TREASY. **Tudo sobre a Metodologia Scrum:** o que é, como usar, exemplos. Passo Fundo: Treasy, 2023. Disponível em: <https://www.treasy.com.br/blog/scrum/>. Acesso em: 14 mar. 2024.

TREINAWEB. **Afinal, o que é TDD?** Disponível em: <https://www.treinaweb.com.br/blog/afinal-o-que-e-tdd>. Acesso em: 10 set. 2024.

TRELLO. **Gerencie os projetos do time em qualquer lugar.** Disponível em: <https://trello.com/>. acesso em: 14 mar. 2024.

UOL DIVEO. **Em reestruturação, UOL Diveo separa áreas de negócio e muda nome.** [S.l.], 13 abr. 2023. Blog da Compass UOL. Disponível em: <https://blog.compass.uol/noticias/em-reestruturacao-uol-diveo-separa-areas-de-negocio-e-muda-nome/>. Acesso em: 22 mar. 2024.

UPF. **Compasso: referência na área de tecnologia da informação.** Disponível em: <https://www.upf.br/noticia/compasso--referencia-na-area-de-tecnologia-da-informacao>. Acesso em: 22 mar. 2024.

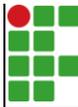
WIKIPEDIA. **Compasso UOL.** [S.l.], [s.d.]. Disponível em: https://pt.wikipedia.org/wiki/Compasso_UOL. Acesso em: 14 mar. 2024

WEISSMANN, Henrique Lobo. **Vire o jogo com Spring Framework.** Editora Casa do Código, 2014.

ANEXO A - Certificação AWS Cloud PractitionerThe logo consists of the text "aws" in a lowercase, sans-serif font, followed by a small orange square icon containing a white checkmark, and then the word "certified" in a lowercase, sans-serif font.**EZEQUIAS SOARES OLIVEIRA**

AWS Certified Cloud Practitioner

VALIDATION NUMBER: 1EFXX8L2XJE1QNKK**VALIDATE AT:** <https://aws.amazon.com/verification>**Issue Date:** October 23, 2023**Expiration Date:** October 23, 2026

| | |
|---|---|
|  | INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA |
| | Campus Monteiro - Código INEP: 25284940 |
| | Pb-264, S/N, Serrote, CEP 58500-000, Monteiro (PB) |
| | CNPJ: 10.783.898/0008-41 - Telefone: (83) 3351-3700 |

Documento Digitalizado Ostensivo (Público)

Relatorio de estagio

| | |
|-----------------------------|----------------------|
| Assunto: | Relatorio de estagio |
| Assinado por: | Ezequias Soares |
| Tipo do Documento: | Anexo |
| Situação: | Finalizado |
| Nível de Acesso: | Ostensivo (Público) |
| Tipo do Conferência: | Cópia Simples |

Documento assinado eletronicamente por:

- **Ezequias Soares de Oliveira, ALUNO (202025020025) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO**, em 08/01/2025 21:01:41.

Este documento foi armazenado no SUAP em 08/01/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1356338

Código de Autenticação: e168813d4a

