



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

**RELATO DE EXPERIÊNCIA PROFISSIONAL COMO TRAINEE NO
DESENVOLVIMENTO DE SOLUÇÕES DE SOFTWARE**

BRUNO LACERDA NOGUEIRA

Cajazeiras - PB

2026

BRUNO LACERDA NOGUEIRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL COMO TRAINEE NO
DESENVOLVIMENTO DE SOLUÇÕES DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Diogo Dantas Moreira.

IFSertãoPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

N778r Nogueira, Bruno Lacerda.
Relato de experiência profissional como trainee no desenvolvimento de soluções de software / Bruno Lacerda Nogueira.– 2026.

33f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Sertão Paraibano, Cajazeiras, 2026.

Orientador(a): Prof. Me. Diogo Dantas Moreira.

1. Desenvolvimento de sistemas. 2. Engenharia de software. 3. Trainee. 4. Formação profissional. I. Instituto Federal de Educação, Ciência e Tecnologia do Sertão Paraibano. II. Título.

IFSertão
PB/CZ

CDU: 004.4(043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

BRUNO LACERDA NOGUEIRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL COMO TRAINEE NO DESENVOLVIMENTO DE
SOLUÇÕES DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Diogo Dantas Moreira

Aprovada em: **14 de agosto de 2026.**

Prof. Me. Diogo Dantas Moreira - Orientador

IFPB - Campus Cajazeiras

Prof. Me. Fábio Abrantes Diniz - Avaliador

IFPB - Campus Cajazeiras

Prof. Dr. Fábio Gomes de Andrade - Avaliador

Documento assinado eletronicamente por:

- **Fabio Abrantes Diniz**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 18/08/2026 13:12:07.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 18/08/2026 13:13:30.
- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 18/08/2026 14:14:42.

Este documento foi emitido pelo SUAP em 18/08/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 925098
Verificador: 878d4b42b8
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

RESUMO

Este Trabalho de Conclusão de Curso apresenta um relato de experiência sobre minha atuação como trainee de Engenharia de Software na Empresa X, durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB) – Campus Cajazeiras. O trabalho tem como objetivo descrever as atividades desenvolvidas, as tecnologias utilizadas e os principais conhecimentos adquiridos ao longo da experiência profissional, relacionando a teoria estudada na graduação com sua aplicação prática no ambiente corporativo. Durante minha atuação como trainee, participei do desenvolvimento e da manutenção de soluções de software utilizando Java e Spring Boot, atuando na implementação de APIs REST, manutenção de microsserviços e correção de funcionalidades. Também utilizei tecnologias e ferramentas como MongoDB, Insomnia, JUnit, Docker, Amazon Web Services (AWS), Amazon CloudWatch, Git, GitHub, Jenkins e IntelliJ IDEA, além de metodologias ágeis, especialmente o Scrum. A experiência contribuiu para o aprimoramento dos conhecimentos técnicos em Engenharia de Software e para o desenvolvimento de competências profissionais, como trabalho em equipe, comunicação, autonomia e resolução de problemas. Conclui-se que a atuação como trainee foi fundamental para minha formação acadêmica e profissional, proporcionando uma compreensão mais ampla do desenvolvimento de software em ambientes corporativos.

Palavras-chave: Trainee. Engenharia de Software. Java. Spring Boot. Microsserviços.

ABSTRACT

This Final Course Project presents an experience report on my work as a Software Engineering trainee at Company X, during the Technology in Systems Analysis and Development undergraduate program at the Federal Institute of Education, Science and Technology of Paraíba (IFPB) – Cajazeiras Campus. The objective of this work is to describe the activities performed, the technologies used, and the main knowledge acquired throughout this professional experience, relating the concepts learned during undergraduate studies to their practical application in a corporate environment. During my work as a trainee, I participated in the development and maintenance of software solutions using Java and Spring Boot, working on REST API implementation, microservices maintenance, and software improvements. I also used technologies and tools such as MongoDB, Insomnia, JUnit, Docker, Amazon Web Services (AWS), Amazon CloudWatch, Git, GitHub, Jenkins, and IntelliJ IDEA, as well as Agile methodologies, especially Scrum. The experience strengthened my technical knowledge in Software Engineering and contributed to the development of professional skills, including teamwork, communication, autonomy, and problem-solving. It is concluded that my experience as a trainee played a fundamental role in my academic and professional development, providing a broader understanding of software development in corporate environments.

Keywords: Trainee. Software Engineering. Java. Spring Boot. Microservices.

LISTA DE ABREVIATURA E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ADS	Análise e Desenvolvimento de Sistemas
API	Application Programming Interface
AWS	Amazon Web Services
CI	Continuous Integration
CRUD	Create, Read, Update, Delete
CPU	Central Processing Unit
DevOps	Development and Operations
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
JDK	Java Development Kit
JPA	Java Persistence API
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
PO	Product Owner
POO	Programação Orientada a Objetos
QA	Quality Assurance
REST	Representational State Transfer
SDK	Software Development Kit
SQL	Structured Query Language
SSH	Secure Shell
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO	7
1.1 OBJETIVOS	8
1.1.1 Objetivo Geral	8
1.1.2 Objetivos Específicos	8
1.2 ORGANIZAÇÃO DO DOCUMENTO	8
2 APRESENTAÇÃO DA EMPRESA	10
3 FUNDAMENTAÇÃO TEÓRICA	12
3.1 Engenharia de Software	12
3.2 Linguagem Java	13
3.3 Spring Boot	14
3.4 Arquitetura de Microsserviços	15
3.5 APIs REST	15
3.6 Docker	16
3.7 Computação em Nuvem e Amazon Web Services (AWS)	17
3.8 Metodologias Ágeis	17
3.9 Considerações da Fundamentação Teórica	18
4 RELATO DE EXPERIÊNCIA	20
4.1 Ingresso na empresa	20
4.2 Integração à equipe de desenvolvimento	21
4.3 Atividades Desenvolvidas	22
4.4 Tecnologias e Ferramentas Utilizadas	24
4.5 Desafios Enfrentados	26
4.6 Aprendizados Adquiridos	27
4.7 Relação entre a Graduação e a Experiência Profissional	28
4.8 Considerações sobre a Experiência	28
5 CONCLUSÃO	30
REFERÊNCIAS	32

1 INTRODUÇÃO

Nas últimas décadas, a transformação digital tem alterado de maneira significativa a forma como as organizações desenvolvem, disponibilizam e mantêm soluções tecnológicas. Para se manterem competitivas, empresas de diversos setores foram levadas a adotar arquiteturas de software modernas, capazes de oferecer maior escalabilidade, disponibilidade, segurança e facilidade de manutenção. Nesse cenário, tecnologias como microsserviços, computação em nuvem, containerização e integração por meio de *Application Programming Interfaces* (APIs) passaram a ser componentes importantes para a construção de sistemas corporativos (PRESSMAN; MAXIM, 2020).

Simultaneamente ao avanço tecnológico, a Engenharia de Software tornou-se uma das principais áreas da Computação, englobando métodos, processos e boas práticas que possibilitam desenvolver aplicações com maior qualidade, confiabilidade e capacidade de evolução. Além dos aspectos técnicos, essa área também contempla metodologias de desenvolvimento, gestão de projetos, testes de software, integração contínua e trabalho em equipe com profissionais de diferentes áreas, aspectos fundamentais para atender às exigências do mercado de tecnologia contemporâneo (SOMMERVILLE, 2019).

Nesse sentido, a formação acadêmica é fundamental para proporcionar ao profissional a base teórica necessária para sua atuação. No entanto, embora os conhecimentos adquiridos na graduação sejam essenciais, a experiência no ambiente corporativo permite compreender como esses conceitos são aplicados na criação e manutenção de soluções reais. Dessa forma, a participação em programas de desenvolvimento profissional, como programas de trainee, possibilita aproximar teoria e prática e contribui para o aprimoramento de habilidades técnicas e comportamentais que dificilmente seriam desenvolvidas apenas em sala de aula.

Foi nesse contexto que, em 2024, tive a oportunidade de atuar como trainee de Engenharia de Software na Empresa X. A organização atua no setor de Tecnologia da Informação e desenvolve soluções digitais, incluindo sistemas voltados ao setor financeiro. Durante minha atuação, participei da manutenção e evolução de microsserviços relacionados a uma plataforma financeira, utilizando tecnologias e ferramentas adotadas no mercado, como Java, Spring Boot, Docker e Amazon Web Services (AWS). Essa experiência proporcionou contato direto com práticas modernas de desenvolvimento de software e possibilitou compreender, na prática, conceitos que até então haviam sido estudados predominantemente de forma teórica.

Ao longo do programa de trainee, participei da implementação de novas funcionalidades, da manutenção corretiva e evolutiva de APIs REST, da integração entre serviços distribuídos, da criação de testes automatizados e das cerimônias previstas pela metodologia Scrum. Além das atividades técnicas, vivenciei situações que demandaram

análise, comunicação, trabalho em equipe e aprendizado constante, competências cada vez mais valorizadas no campo da Tecnologia da Informação.

Um dos principais desafios dessa experiência foi compreender a arquitetura de microsserviços utilizada pela empresa, adaptar-me aos padrões de desenvolvimento adotados pela equipe, analisar códigos desenvolvidos por outros profissionais e encontrar soluções para problemas relacionados à integração entre serviços e à manutenção de aplicações em um ambiente corporativo. Esses desafios foram importantes para meu crescimento profissional, contribuindo para ampliar minha capacidade de resolução de problemas, minha autonomia e minha compreensão sobre sistemas distribuídos de maior complexidade.

Dessa forma, este Trabalho de Conclusão de Curso apresenta um relato de experiência cujo propósito é descrever as atividades que desenvolvi durante minha atuação como trainee, evidenciando as tecnologias utilizadas, os desafios enfrentados, os conhecimentos adquiridos e a contribuição dessa vivência para minha formação acadêmica e profissional. Além de relatar minha experiência, este trabalho busca evidenciar como a aproximação entre a universidade e o mercado de trabalho é importante para a formação de profissionais mais preparados para atuar no desenvolvimento de software.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Relatar minha experiência como trainee de Engenharia de Software no desenvolvimento e na manutenção de microsserviços em uma empresa do setor tecnológico, destacando as atividades que realizei, as tecnologias que utilizei, os desafios que enfrentei e os conhecimentos adquiridos ao longo dessa experiência profissional.

1.1.2 Objetivos Específicos

- Contextualizar o papel da Engenharia de Software no desenvolvimento de sistemas corporativos baseados em microsserviços;
- Discutir as principais tecnologias, ferramentas e metodologias empregadas durante as atividades desenvolvidas no programa de trainee;
- Relatar as tarefas executadas ao longo da experiência profissional, destacando minha contribuição nas demandas da equipe de desenvolvimento;
- Descrever os principais desafios técnicos enfrentados durante a implementação e manutenção de funcionalidades em aplicações distribuídas;
- Demonstrar de que forma a experiência prática contribuiu para consolidar os conhecimentos adquiridos durante a graduação;
- Analisar as habilidades técnicas e comportamentais que desenvolvi durante minha atuação como trainee e suas contribuições para minha formação profissional.

1.2 ORGANIZAÇÃO DO DOCUMENTO

Este TCC é dividido em cinco capítulos, organizados de modo que, inicialmente, o leitor conheça o contexto da experiência relatada e, posteriormente, a fundamentação teórica que sustenta as atividades realizadas. Em seguida, é apresentado o relato da experiência profissional e, por fim, as considerações finais.

O Capítulo 2 apresenta a Empresa X, organização na qual atuei como trainee, abordando seu ramo de atuação, sua estrutura organizacional, seu modelo de desenvolvimento de software e o ambiente em que as atividades profissionais foram realizadas.

No Capítulo 3 é apresentada a fundamentação teórica necessária para compreender os conceitos, tecnologias e metodologias utilizados ao longo da experiência profissional, incluindo tópicos como Engenharia de Software, linguagem Java, Spring Boot, arquitetura de microsserviços, APIs REST, Docker, computação em nuvem, Amazon Web Services (AWS) e metodologias ágeis.

O Capítulo 4 constitui o núcleo deste trabalho e apresenta o relato da minha experiência profissional. Nesse capítulo, detalho minha entrada na empresa e no programa de trainee, minha integração à equipe de desenvolvimento, as atividades que realizei, as tecnologias e ferramentas utilizadas, os principais desafios que enfrentei e os aprendizados técnicos e profissionais adquiridos durante esse período.

Por fim, o Capítulo 5 apresenta as considerações finais deste relato de experiência, destacando as contribuições da minha atuação como trainee para minha formação acadêmica e profissional, além de reflexões sobre a importância da integração entre teoria e prática na formação de profissionais de Tecnologia da Informação.

2 APRESENTAÇÃO DA EMPRESA

As informações apresentadas neste capítulo descrevem o contexto organizacional e as atividades desenvolvidas durante minha atuação como trainee, sem expor dados estratégicos ou sensíveis da empresa.

A Empresa X atua na área de Tecnologia da Informação, oferecendo desenvolvimento de soluções digitais para o setor financeiro, varejista e para grandes corporações. Desde sua fundação, a empresa tem direcionado suas atividades para o desenvolvimento de sistemas corporativos baseados em tecnologias modernas, priorizando aspectos como escalabilidade, segurança, disponibilidade e desempenho. Para isso, utiliza arquiteturas orientadas a microsserviços, computação em nuvem e integração entre aplicações por meio de APIs REST, alinhando-se às tendências da Engenharia de Software moderna.

A organização, que possui sede no estado de São Paulo (SP), conta com equipes distribuídas em diferentes estados do Brasil, operando em modelos de trabalho híbrido e remoto. Sua estrutura envolve profissionais de diferentes áreas, incluindo Engenharia de Software, Arquitetura de Sistemas, Qualidade de Software (QA), DevOps, Segurança da Informação, Ciência de Dados, Gestão de Produtos (*Product Management*) e Experiência do Usuário (UX). Essa estrutura multidisciplinar favorece a colaboração entre diferentes especialidades e contribui para o desenvolvimento de soluções alinhadas às necessidades dos clientes.

A empresa desenvolve seus projetos com base em princípios da Engenharia de Software, dando ênfase à modularização dos sistemas, à reutilização de componentes e à facilidade de manutenção das aplicações. A adoção de uma arquitetura de microsserviços possibilita que diferentes serviços sejam desenvolvidos e disponibilizados de maneira independente, reduzindo o impacto de alterações específicas e favorecendo a escalabilidade dos sistemas.

No ambiente de desenvolvimento, são empregadas tecnologias amplamente utilizadas no setor de desenvolvimento de software, incluindo Java, Spring Boot, bancos de dados relacionais e não relacionais, Docker para a containerização de aplicações e serviços oferecidos pela Amazon Web Services (AWS). Além disso, a empresa adota práticas modernas de desenvolvimento, como versionamento de código, integração contínua, testes automatizados, monitoramento de aplicações e revisão colaborativa de código (*Code Review*), contribuindo para a manutenção da qualidade das soluções desenvolvidas.

A gestão dos projetos ocorre por meio de metodologias ágeis, tendo o Scrum como principal *framework* utilizado para a organização e o acompanhamento das atividades. As equipes são organizadas em *squads* multidisciplinares responsáveis por produtos ou partes específicas dos sistemas, realizando cerimônias como *Sprint Planning*, *Daily Meeting*, *Sprint Review* e *Sprint Retrospective*. Esse modelo de trabalho favorece a comunicação entre os

integrantes da equipe, permite um acompanhamento mais transparente das atividades e contribui para a realização contínua das entregas.

O programa de trainee proporcionava aos participantes a oportunidade de atuar em projetos reais de desenvolvimento de software e ampliar seus conhecimentos técnicos e profissionais. Durante esse período, os trainees tinham contato com profissionais mais experientes e participavam das atividades relacionadas ao fluxo de desenvolvimento adotado pela equipe, incluindo reuniões de planejamento, desenvolvimento de funcionalidades, revisões de código e discussões sobre boas práticas de Engenharia de Software. Essa interação com profissionais de diferentes níveis de experiência contribuiu para o desenvolvimento técnico e para o aprimoramento de competências interpessoais importantes para a atuação profissional.

Foi nesse contexto que ingressei na Empresa X como trainee de Engenharia de Software, integrando uma equipe de desenvolvimento *back-end* responsável pela manutenção e evolução de microsserviços utilizados por uma plataforma financeira de grande porte. Durante minha atuação, participei do desenvolvimento e da manutenção de APIs REST, da implementação de novas funcionalidades, da correção de *bugs*, da criação de testes automatizados e de atividades relacionadas à evolução das aplicações.

Desde o início da minha atuação como trainee, percebi que o ambiente de trabalho era caracterizado pela colaboração constante entre os integrantes da equipe. Quando surgiam dúvidas durante a execução das tarefas, contava com o suporte de desenvolvedores mais experientes, que auxiliavam na resolução das questões e compartilhavam orientações relacionadas às boas práticas de programação e arquitetura de software. Essa troca de experiências foi importante para meu processo de aprendizagem e para o desenvolvimento de maior autonomia na realização das atividades.

Outro aspecto que considerei importante foi a cultura de aprendizado contínuo presente no ambiente de trabalho. Ao longo do programa de trainee, fui incentivado a consultar documentações oficiais, explorar tecnologias utilizadas no projeto, compreender os padrões arquiteturais adotados pela equipe e buscar soluções fundamentadas para os problemas encontrados. Essa prática contribuiu para meu desenvolvimento profissional e reforçou a importância da atualização constante no campo da Tecnologia da Informação.

A experiência adquirida nesse ambiente possibilitou compreender, na prática, diversos conceitos estudados ao longo da graduação, como Programação Orientada a Objetos, Engenharia de Software, Banco de Dados, Redes de Computadores e Desenvolvimento Web. Também contribuiu para o desenvolvimento de competências comportamentais importantes para minha formação profissional, como comunicação, organização, trabalho em equipe, responsabilidade, capacidade de análise e resolução de problemas. Dessa forma, minha atuação como trainee representou uma oportunidade de relacionar os conhecimentos adquiridos durante a graduação às demandas encontradas no mercado de desenvolvimento de software, contribuindo para minha preparação profissional.

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os fundamentos teóricos que sustentam as práticas realizadas durante a experiência profissional relatada neste trabalho. A compreensão desses conceitos é importante para contextualizar as tecnologias, arquiteturas e metodologias com as quais tive contato no ambiente empresarial e para demonstrar como os conhecimentos adquiridos durante a graduação se relacionam com a prática do desenvolvimento de software.

Ao longo da minha atuação como trainee, tive contato com diferentes tecnologias e processos amplamente utilizados na Engenharia de Software, como Java, o *framework* Spring Boot, arquitetura de microsserviços, APIs REST, Docker, computação em nuvem e metodologias ágeis. Dessa forma, esta fundamentação busca apresentar uma visão geral desses temas, destacando sua importância para o desenvolvimento de aplicações contemporâneas e sua contribuição para a construção de sistemas robustos, escaláveis e de qualidade.

3.1 Engenharia de Software

A Engenharia de Software é uma área da Computação que se ocupa do desenvolvimento, manutenção e evolução de sistemas computacionais, utilizando métodos, técnicas e boas práticas que buscam assegurar maior qualidade, confiabilidade, segurança e facilidade de manutenção do software. Sommerville (2019) afirma que essa área abrange todos os aspectos da produção de software, desde os estágios iniciais de especificação até a manutenção do sistema após sua entrada em operação.

O desenvolvimento de software é uma atividade que vai além da codificação. Segundo Pressman e Maxim (2021), é necessário compreender os requisitos do sistema, planejar sua construção, implementar as funcionalidades de maneira sistemática, realizar testes e acompanhar a evolução do produto ao longo de seu ciclo de vida.

À medida que a transformação digital avança, as organizações passaram a demandar sistemas cada vez mais escaláveis, seguros e disponíveis. Nesse contexto, a Engenharia de Software desempenha um papel importante para organizações que utilizam a tecnologia como suporte aos seus processos de negócio, possibilitando o desenvolvimento de aplicações capazes de evoluir diante das mudanças do mercado e das necessidades dos usuários.

Durante minha atuação como trainee, tive a oportunidade de observar como esses princípios eram aplicados na prática. Antes do desenvolvimento de uma funcionalidade, era necessário compreender os requisitos da demanda, analisar os possíveis impactos das alterações no sistema e definir abordagens adequadas para sua implementação. Essa experiência permitiu compreender que o desenvolvimento de software não se limita à implementação do código, mas envolve planejamento, organização, comunicação entre os integrantes da equipe e preocupação com a manutenção e evolução das aplicações.

Também observei que práticas como revisão de código (*Code Review*), testes automatizados, integração contínua e padronização da arquitetura estavam presentes no cotidiano da equipe de desenvolvimento e contribuíam para a qualidade das soluções desenvolvidas. Essas práticas reforçaram minha compreensão sobre a importância da Engenharia de Software como base para a construção e manutenção de sistemas confiáveis e sustentáveis.

3.2 Linguagem Java

Java é uma linguagem de programação orientada a objetos criada pela Sun Microsystems em 1995 e atualmente administrada pela Oracle. Uma de suas principais características é a portabilidade, possibilitando que aplicações sejam executadas em diferentes sistemas operacionais por meio da Java Virtual Machine (JVM), característica resumida pelo princípio *Write Once, Run Anywhere* (ORACLE, 2026).

Ao longo dos anos, Java consolidou-se como uma das linguagens mais utilizadas no desenvolvimento de aplicações corporativas graças à sua estabilidade, desempenho, segurança e a vasta gama de bibliotecas e frameworks disponíveis. Essas características fazem com que seja amplamente empregada em sistemas financeiros, aplicações web, serviços em nuvem e arquiteturas distribuídas.

A linguagem segue o modelo da Programação Orientada a Objetos (POO), baseado em conceitos como encapsulamento, herança, polimorfismo e abstração. Esses princípios ajudam a manter o código bem organizado, a reutilização de componentes e a manutenção de sistemas de grande porte, tornando o desenvolvimento mais estruturado e facilitando sua evolução ao longo do tempo.

No estágio, utilizei Java como a linguagem principal para a manutenção e desenvolvimento de microsserviços da aplicação. As atividades consistiram na implementação de novas funcionalidades, resolução de defeitos, atualização de regras de negócio e revisão de códigos que já haviam sido desenvolvidos pela equipe.

No início do estágio, um dos maiores desafios que enfrentei no início do estágio foi entender a estrutura do projeto e os padrões de desenvolvimento utilizados pela empresa. Como grande parte do código já havia sido escrita por outros desenvolvedores, foi necessário que eu estudasse detalhadamente a estrutura das classes, como os componentes interagem e o fluxo de execução das funcionalidades antes de fazer qualquer modificação.

Com o tempo, à medida que fui me acostumando mais com a aplicação, comecei a entender melhor alguns conceitos que estudei na graduação, como Programação Orientada a Objetos, tratamento de exceções, coleções, interfaces e a organização em camadas (layers). O conhecimento teórico foi colocado à prova na prática, o que reforçou o quanto é fundamental ter uma formação acadêmica para trabalhar em projetos dentro de uma empresa.

Outro ponto importante que pude observar durante o estágio foi a adoção de boas práticas de programação, como padronização do código, utilização de nomes significativos para classes e métodos, reutilização de componentes e preocupação constante com a legibilidade do software. Essas práticas tornaram tanto o desenvolvimento quanto a manutenção das funcionalidades que implementei muito mais simples, além de terem sido cruciais para o meu crescimento como desenvolvedor.

3.3 Spring Boot

O Spring Boot é um *framework* baseado no ecossistema Spring, desenvolvido com o objetivo de simplificar a criação de aplicações Java por meio da redução de configurações manuais e da disponibilização de recursos que aceleram o desenvolvimento de sistemas corporativos. De acordo com a documentação oficial do Spring (2026), o *framework* possibilita a criação de aplicações independentes, prontas para produção e com pouca necessidade de configuração.

Dentre suas principais características, destacam-se a configuração automática (*Auto Configuration*), o servidor web embarcado, o gerenciamento de dependências e a integração simplificada com bancos de dados, sistemas de autenticação e outros componentes utilizados em aplicações empresariais. Esses recursos permitem que o desenvolvedor concentre maior atenção na implementação das regras de negócio, reduzindo a necessidade de configurações manuais relacionadas à infraestrutura da aplicação.

Outra característica relevante do Spring Boot é sua ampla utilização no desenvolvimento de APIs REST e de aplicações baseadas em arquiteturas de microsserviços. O *framework* disponibiliza recursos para validação de dados, tratamento de exceções, gerenciamento de dependências e configurações, além de integração com diferentes componentes do ecossistema Spring, contribuindo para a construção de aplicações organizadas e de fácil manutenção.

Durante minha atuação como trainee, utilizei o Spring Boot como principal *framework* para o desenvolvimento e a manutenção dos microsserviços da aplicação. Grande parte das atividades envolveu a criação de novos *endpoints*, a atualização de regras de negócio, a correção de *bugs* identificados durante os testes e atividades relacionadas à integração entre os diferentes serviços da aplicação.

Além da implementação das funcionalidades, compreendi a importância da organização da aplicação em camadas, como *Controller*, *Service* e *Repository*, padrão adotado pela equipe para facilitar a manutenção, favorecer a reutilização de código e estabelecer uma separação mais clara das responsabilidades entre os diferentes componentes do sistema.

Essa experiência permitiu compreender, na prática, como a adoção de um *framework* consolidado, como o Spring Boot, pode contribuir para a produtividade da equipe, para a padronização do desenvolvimento e para a evolução contínua das aplicações.

3.4 Arquitetura de Microsserviços

A arquitetura de microsserviços consiste em um modelo de desenvolvimento no qual uma aplicação é dividida em diferentes serviços independentes, sendo cada um responsável por um conjunto específico de funcionalidades. De acordo com Fowler (2014), essa abordagem possibilita que diferentes componentes sejam desenvolvidos, implantados e escalados de maneira independente, favorecendo a flexibilidade e a evolução das aplicações.

Ao contrário da arquitetura monolítica, na qual diferentes funcionalidades estão integradas em uma única aplicação, a abordagem de microsserviços distribui as responsabilidades entre serviços menores e especializados. Essa estrutura pode facilitar a manutenção e reduzir o impacto de determinadas alterações, além de possibilitar que diferentes equipes trabalhem simultaneamente em partes distintas do sistema.

Entretanto, essa arquitetura também apresenta desafios significativos. A comunicação entre serviços, a consistência dos dados, o monitoramento das aplicações, a autenticação e o tratamento de falhas exigem planejamento e adoção de boas práticas para garantir o funcionamento adequado da solução.

Durante minha atuação como trainee, trabalhei com uma arquitetura composta por múltiplos microsserviços relacionados ao processamento das operações da plataforma financeira. Em diferentes atividades, foi necessário compreender como esses serviços se relacionavam antes de realizar alterações, uma vez que modificações em determinado componente poderiam afetar o comportamento de outros serviços da aplicação.

Essa experiência permitiu compreender que o funcionamento adequado de uma arquitetura distribuída depende não apenas da qualidade do código, mas também de fatores como documentação, comunicação entre os integrantes das equipes, monitoramento e padronização das interfaces utilizadas na comunicação entre os serviços.

3.5 APIs REST

Application Programming Interfaces (APIs) são interfaces que possibilitam a comunicação e a troca de informações entre diferentes sistemas ou componentes de software. Entre os estilos arquiteturais utilizados no desenvolvimento de aplicações distribuídas está o REST (*Representational State Transfer*), apresentado por Roy Fielding em sua tese de doutorado (2000) e amplamente adotado no desenvolvimento de sistemas para a Web.

As APIs REST utilizam recursos do protocolo HTTP para possibilitar a comunicação entre aplicações, sendo comum a utilização de métodos como GET, POST, PUT e DELETE em operações de consulta, criação, atualização e exclusão de recursos. Também são utilizados códigos de status HTTP para indicar o resultado das requisições, permitindo que clientes e servidores interpretem adequadamente as respostas geradas durante a comunicação.

Esse modelo tornou-se amplamente utilizado devido a características como simplicidade, escalabilidade e interoperabilidade entre diferentes linguagens de programação e plataformas, permitindo que sistemas distintos troquem informações por meio de interfaces bem definidas.

Durante minha atuação como trainee, trabalhei frequentemente com APIs REST desenvolvidas em Java e Spring Boot. Minhas atividades incluíram a criação de novos *endpoints*, a manutenção de funcionalidades existentes, a implementação e atualização de regras de negócio e a correção de *bugs* identificados durante o processo de desenvolvimento e validação.

Também realizei a validação das APIs utilizando o Insomnia, por meio do qual enviava requisições HTTP aos *endpoints* e analisava as respostas retornadas pelos serviços. Essas verificações permitiam avaliar o funcionamento das funcionalidades implementadas, os códigos de status HTTP, os dados retornados e o tratamento de possíveis erros. Essa experiência contribuiu para ampliar minha compreensão sobre a comunicação entre aplicações distribuídas e sobre a importância da padronização das interfaces de integração.

O contato contínuo com APIs REST foi importante para relacionar os conhecimentos adquiridos durante a graduação sobre desenvolvimento web, arquitetura cliente-servidor e protocolos de comunicação com sua aplicação prática no desenvolvimento de sistemas corporativos.

3.6 Docker

O Docker é uma plataforma de containerização que possibilita o empacotamento de aplicações com suas dependências, bibliotecas e configurações necessárias para sua execução. Conforme a documentação oficial do Docker (2026), os contêineres permitem criar ambientes padronizados para execução das aplicações, contribuindo para reduzir diferenças entre os ambientes de desenvolvimento, testes e produção.

Ao contrário das máquinas virtuais tradicionais, os contêineres compartilham recursos do sistema operacional hospedeiro, o que permite uma utilização mais eficiente dos recursos computacionais. Essa característica contribui para facilitar processos relacionados ao desenvolvimento, aos testes e à implantação das aplicações.

O Docker é amplamente utilizado em arquiteturas de microsserviços, possibilitando que diferentes serviços sejam executados de maneira isolada, cada um com as dependências necessárias para seu funcionamento. Essa abordagem contribui para a padronização dos ambientes e facilita a execução e a manutenção das aplicações.

Durante minha atuação como trainee, trabalhei com aplicações executadas em containers Docker. Embora minha atuação estivesse concentrada principalmente no desenvolvimento e na manutenção das funcionalidades dos microsserviços, compreender o

funcionamento dessa tecnologia foi importante para entender como as aplicações eram executadas nos diferentes ambientes utilizados pela equipe.

Essa experiência permitiu compreender como a padronização proporcionada pela containerização contribui para reduzir problemas relacionados às diferenças entre ambientes e para tornar o processo de desenvolvimento e disponibilização das aplicações mais previsível.

3.7 Computação em Nuvem e Amazon Web Services (AWS)

A computação em nuvem (*Cloud Computing*) consiste na disponibilização de recursos computacionais por meio da internet, possibilitando o acesso a recursos como processamento, armazenamento, bancos de dados e outros serviços sem a necessidade de manter toda a infraestrutura computacional localmente. Segundo a documentação oficial da Amazon Web Services (2026), esse modelo possibilita maior flexibilidade e escalabilidade na utilização dos recursos computacionais.

Entre os benefícios da computação em nuvem estão a possibilidade de ajustar os recursos computacionais de acordo com a demanda, a disponibilidade de diferentes serviços e a redução da necessidade de manutenção de infraestrutura física própria. Por essas características, a computação em nuvem é amplamente utilizada no desenvolvimento e na disponibilização de aplicações corporativas.

A Amazon Web Services (AWS) é uma plataforma de computação em nuvem que disponibiliza diferentes serviços relacionados a processamento, armazenamento, bancos de dados, redes, segurança, monitoramento e implantação de aplicações.

Durante minha atuação como trainee, os microsserviços utilizados pela equipe eram executados em uma infraestrutura baseada na AWS. Embora eu não atuasse diretamente na administração dessa infraestrutura, acompanhei algumas implantações realizadas pela equipe e tive contato com o Amazon CloudWatch, serviço utilizado para monitoramento e observabilidade das aplicações. Por meio dele, consultava logs de execução dos microsserviços para auxiliar na identificação e análise de falhas encontradas durante o desenvolvimento e a validação das funcionalidades.

Essa experiência permitiu compreender que a computação em nuvem não se limita à hospedagem das aplicações, mas envolve um conjunto de serviços que contribuem para aspectos como disponibilidade, escalabilidade, monitoramento e segurança dos sistemas corporativos. O contato com o Amazon CloudWatch também demonstrou, na prática, a importância da observabilidade para a identificação de problemas em aplicações distribuídas.

3.8 Metodologias Ágeis

As metodologias ágeis apareceram como uma resposta aos modelos tradicionais de desenvolvimento de software, sugerindo períodos de desenvolvimento mais curtos, mais colaboração entre as equipes e uma adaptação constante às mudanças nas necessidades dos

projetos. Dentre os diversos frameworks ágeis, o Scrum se destaca como um dos mais utilizados, sendo definido por Ken Schwaber e Jeff Sutherland (2020) como uma estrutura para o desenvolvimento iterativo e incremental de produtos complexos.

No Scrum, as atividades são organizadas em Sprints, que são períodos que geralmente variam de uma a quatro semanas. No final de cada ciclo, uma nova versão do produto é lançada, o que possibilita que as melhorias sejam feitas de forma contínua, com base nas necessidades que a equipe e os clientes identificam.

O framework também define cerimônias específicas, incluindo Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective. O objetivo dessas reuniões é planejar as atividades, monitorar o progresso do desenvolvimento, reportar os resultados obtidos e implementar melhorias contínuas nos processos de trabalho.

No meu estágio, estive presente nessas cerimônias com os outros membros da equipe. As reuniões diárias eram o momento de acompanhar o progresso das tarefas, tirar dúvidas, identificar bloqueios e alinhar prioridades do projeto.

Eu também estive presente nas reuniões de planejamento das Sprints, onde eram definidas as funcionalidades prioritárias, estimado o esforço necessário para cada tarefa e as atividades eram distribuídas entre os integrantes da equipe. Isso tudo mostrou o quanto o planejamento em conjunto ajuda a manter os prazos e a qualidade das entregas.

A experiência em um ambiente que implementava metodologias ágeis evidenciou que o êxito de um projeto não se resume à aplicação de tecnologias de ponta, mas é também resultado de processos claramente definidos, comunicação eficaz e colaboração contínua entre os profissionais envolvidos.

3.9 Considerações da Fundamentação Teórica

Os conceitos abordados neste capítulo são a fundamentação teórica que possibilita entender as ações realizadas no estágio descrito neste trabalho. O conhecimento adquirido em Engenharia de Software, na linguagem Java, no framework Spring Boot, na arquitetura de microsserviços, nas APIs REST, no Docker, na computação em nuvem e nas metodologias ágeis permitiu uma compreensão mais aprofundada das tecnologias e práticas utilizadas pela empresa.

Na prática profissional, percebi que esses conceitos não ficam limitados ao ambiente escolar, sendo utilizados com frequência no desenvolvimento de sistemas empresariais. A adoção conjunta dessas tecnologias permite a criação de aplicações que são mais organizadas, escaláveis, seguras e fáceis de manter, qualidades que são cruciais para atender às exigências do setor de Tecnologia da Informação.

Portanto, a teoria apresentada neste capítulo embasa o relato de experiência que será exposto no próximo, mostrando a conexão entre o que foi aprendido na graduação e a prática nas ações desenvolvidas durante o estágio.

4 RELATO DE EXPERIÊNCIA

Este capítulo apresenta o relato das experiências que vivenciei durante minha atuação como trainee de Engenharia de Software na Empresa X. Nele, descrevo meu ingresso no ambiente corporativo, minha integração à equipe de desenvolvimento, as atividades que realizei, os desafios que enfrentei e os conhecimentos que adquiri ao longo dessa experiência profissional.

Essa experiência representou meu primeiro contato com um ambiente profissional dedicado ao desenvolvimento de software em larga escala. Além de aplicar conhecimentos adquiridos durante a graduação, pude compreender como tecnologias, metodologias e boas práticas estudadas no ambiente acadêmico são utilizadas na construção e manutenção de sistemas corporativos. Durante esse período, também desenvolvi competências relacionadas à comunicação, organização, trabalho em equipe, autonomia e resolução de problemas, importantes para a atuação profissional na área de Tecnologia da Informação.

4.1 Ingresso na empresa

Minha oportunidade de ingressar na Empresa X como trainee de Engenharia de Software surgiu por meio de um processo de recrutamento promovido pela organização. O processo seletivo foi composto por diferentes etapas, incluindo análise curricular, avaliação técnica e entrevistas com profissionais da área de Engenharia de Software. Durante esse processo, tive a oportunidade de demonstrar os conhecimentos que havia adquirido ao longo da graduação e meu interesse pela área de desenvolvimento de software.

Após minha aprovação no processo seletivo, iniciei o período de integração (*onboarding*), destinado à adaptação ao ambiente organizacional e à compreensão dos processos internos da empresa. Nessa etapa, conheci a estrutura da organização, as equipes, os principais projetos em andamento e as ferramentas utilizadas no cotidiano de trabalho. Também participei de treinamentos iniciais relacionados aos processos internos, às metodologias de trabalho e às boas práticas adotadas pela empresa.

Nas primeiras semanas, dediquei grande parte do tempo à compreensão da arquitetura dos sistemas, da estrutura dos projetos e do fluxo de desenvolvimento adotado pela equipe. Como a aplicação era composta por múltiplos microsserviços interconectados, compreender a organização e o funcionamento do sistema foi uma etapa importante antes de realizar alterações no código e atuar diretamente nas demandas de desenvolvimento.

Além do aprendizado técnico, esse período foi importante para compreender a cultura e a dinâmica de trabalho da empresa. Observei que a colaboração entre os profissionais era constantemente incentivada, contribuindo para minha adaptação à equipe. Quando surgiam dúvidas, contava com o auxílio de desenvolvedores mais experientes, que compartilhavam conhecimentos e explicavam as decisões técnicas e arquiteturais adotadas nos projetos.

Outro aspecto que chamou minha atenção durante esse período foi a preocupação da equipe com a qualidade do software desenvolvido. Antes de começar a implementar funcionalidades, recebi orientações relacionadas aos padrões de codificação, à organização dos projetos, ao uso do sistema de controle de versão, ao processo de revisão de código (*Code Review*) e à importância da realização de testes antes da conclusão das demandas. Esses processos contribuíram para compreender que o desenvolvimento de software não se resume à implementação do código, mas também envolve planejamento, organização, validação e preocupação contínua com a qualidade e a manutenção das soluções desenvolvidas.

Após o período inicial de integração, passei a compreender melhor o funcionamento da equipe e o ambiente de desenvolvimento. Inicialmente, comecei a atuar em demandas de menor complexidade e, à medida que adquiria maior conhecimento sobre a aplicação, passei gradualmente a assumir atividades que exigiam maior autonomia e compreensão das regras de negócio e da arquitetura do sistema.

4.2 Integração à equipe de desenvolvimento

Após o período inicial de integração, fui designado para uma *squad* responsável pela manutenção e evolução de uma plataforma financeira desenvolvida pela empresa. A equipe era composta por desenvolvedores *back-end* e *front-end*, profissionais de Qualidade de Software (QA), DevOps, *Product Owner* (PO) e *Scrum Master*, que atuavam de maneira colaborativa ao longo do processo de desenvolvimento.

Desde o início, observei que a execução das atividades dependia da colaboração entre os integrantes da equipe. As decisões técnicas eram discutidas em conjunto e, quando necessário, os profissionais mais experientes auxiliavam os demais integrantes na resolução de problemas e na compreensão das regras de negócio relacionadas às funcionalidades.

As atividades eram organizadas em *Sprints* de duas semanas, seguindo as práticas do *framework* Scrum. No início de cada *Sprint*, eram discutidas as funcionalidades e demandas prioritárias, estimado o esforço necessário para a realização das atividades e definido o trabalho que seria desenvolvido durante o ciclo.

Participei das *Daily Meetings*, reuniões curtas nas quais os integrantes compartilhavam o andamento de suas atividades, comunicavam possíveis impedimentos e alinhavam os próximos passos relacionados ao desenvolvimento. Essas reuniões foram importantes para acompanhar o progresso das demandas e compreender como as atividades realizadas individualmente estavam relacionadas aos objetivos da equipe.

Além das reuniões diárias, participei das *Sprint Reviews*, nas quais eram apresentadas e discutidas as funcionalidades concluídas ao longo do ciclo de desenvolvimento, e das *Sprint Retrospectives*, voltadas à análise do processo de trabalho e à identificação de oportunidades de melhoria. A participação nessas cerimônias demonstrou, na prática, a importância da comunicação constante e da melhoria contínua dos processos de desenvolvimento.

Durante meu período inicial na equipe, busquei compreender como os desenvolvedores estruturavam suas soluções, organizavam o código e discutiam diferentes alternativas para a resolução de problemas técnicos. A convivência com profissionais mais experientes contribuiu para ampliar meus conhecimentos em desenvolvimento de software e proporcionou maior segurança e autonomia para lidar com demandas progressivamente mais complexas.

À medida que adquiri maior conhecimento sobre a arquitetura da aplicação e os processos adotados pela equipe, passei a participar de maneira mais ativa das discussões técnicas, esclarecendo dúvidas, apresentando sugestões para determinadas demandas e implementando as funcionalidades que estavam sob minha responsabilidade. Essa evolução ocorreu de forma gradual e foi importante para meu desenvolvimento técnico e profissional durante minha atuação como trainee.

4.3 Atividades Desenvolvidas

Durante minha atuação como trainee, participei de diferentes atividades relacionadas ao desenvolvimento, à manutenção e à evolução dos microsserviços que compunham a plataforma financeira. As demandas eram definidas e distribuídas de acordo com as necessidades do projeto e organizadas dentro das *Sprints*, permitindo que eu acompanhasse diferentes etapas do ciclo de desenvolvimento de software.

Inicialmente, minhas atividades estavam concentradas em demandas de menor complexidade, como correções de erros, pequenos ajustes em funcionalidades existentes e alterações pontuais no código. Essa etapa foi importante para que eu pudesse compreender gradualmente a estrutura dos projetos, os padrões adotados pela equipe e as principais regras de negócio da plataforma.

Com o avanço da minha atuação como trainee e a aquisição de maior familiaridade com o ambiente de desenvolvimento, passei a atuar em demandas mais complexas. Entre as principais atividades que realizei durante esse período, destacam-se:

- implementação de novos *endpoints* em APIs REST;
- manutenção corretiva e evolutiva de microsserviços;
- implementação e atualização de regras de negócio;
- correção de *bugs* identificados durante o desenvolvimento e a homologação;
- consulta e análise de dados armazenados no MongoDB;
- validação de APIs REST utilizando o Insomnia;
- criação e execução de testes unitários em Java utilizando o JUnit;
- análise de logs dos microsserviços por meio do Amazon CloudWatch;
- utilização do Git e do GitHub no controle de versões e gerenciamento do código;
- participação em revisões de código (*Code Reviews*);

- acompanhamento de processos automatizados de integração e entrega por meio do Jenkins;
- atualização de documentação técnica; e
- acompanhamento de implantações realizadas pela equipe.

No desenvolvimento das funcionalidades, utilizava principalmente Java e Spring Boot, trabalhando no IntelliJ IDEA como ambiente de desenvolvimento. As demandas geralmente envolviam a análise de uma funcionalidade ou problema existente, a identificação das partes do sistema relacionadas à solicitação e, posteriormente, a implementação das alterações necessárias. Dependendo da demanda, também era necessário consultar informações armazenadas no MongoDB para compreender o comportamento da aplicação e verificar dados relacionados às regras de negócio.

Após a implementação, realizava validações das APIs por meio do Insomnia, enviando requisições HTTP aos *endpoints* modificados e verificando os dados e os códigos de resposta retornados pelos serviços. Além dessas verificações, utilizava o JUnit para criar e executar testes unitários, buscando verificar se as regras de negócio apresentavam o comportamento esperado e reduzir a possibilidade de regressões decorrentes das alterações realizadas.

Quando surgiam problemas cuja causa não era imediatamente identificável no ambiente de desenvolvimento, a investigação podia envolver a análise de dados, o comportamento de outros microsserviços e a consulta de logs por meio do Amazon CloudWatch. Essa atividade foi importante para que eu desenvolvesse uma visão mais ampla sobre sistemas distribuídos, pois determinadas falhas podiam estar relacionadas à comunicação entre diferentes serviços e não necessariamente ao microsserviço em que a alteração havia sido realizada.

Após a implementação e as validações iniciais, as alterações eram registradas utilizando o Git e disponibilizadas no GitHub, seguindo o fluxo de versionamento adotado pela equipe. O código desenvolvido passava por *Code Review*, momento em que outros desenvolvedores analisavam a solução implementada e apresentavam sugestões relacionadas à legibilidade, organização, padronização e qualidade do código. Quando necessário, realizava os ajustes indicados antes que a alteração avançasse para as etapas seguintes.

Também acompanhei processos automatizados executados por meio do Jenkins, ferramenta presente no fluxo de integração e entrega das aplicações. Embora minha atuação estivesse concentrada principalmente no desenvolvimento, esse contato permitiu compreender melhor o caminho percorrido por uma alteração desde sua implementação e versionamento até sua validação e disponibilização nos diferentes ambientes utilizados pela equipe.

A participação nessas atividades proporcionou uma compreensão mais ampla do ciclo de desenvolvimento de software em um ambiente corporativo. Passei a perceber que a implementação de uma funcionalidade não se limita à escrita do código, mas envolve também

análise do problema, consulta de dados, testes, validação de APIs, controle de versão, revisão por outros desenvolvedores, monitoramento e acompanhamento das etapas necessárias para que uma alteração seja disponibilizada com segurança.

4.4 Tecnologias e Ferramentas Utilizadas

Durante minha atuação como trainee, utilizei diferentes tecnologias e ferramentas que faziam parte do ambiente de desenvolvimento da empresa e contribuíam para a construção, manutenção, validação e acompanhamento das aplicações. O contato com essas tecnologias ocorreu de forma integrada às atividades desenvolvidas no projeto, permitindo ampliar meus conhecimentos técnicos e compreender melhor o funcionamento de um ambiente profissional de desenvolvimento de software.

A principal linguagem de programação utilizada foi o Java, aplicado no desenvolvimento e na manutenção dos microsserviços que compunham a plataforma. Por meio dele, implementei regras de negócio, realizei correções e desenvolvi novas funcionalidades de acordo com as demandas atribuídas à equipe. A utilização contínua da linguagem também contribuiu para aprofundar meus conhecimentos sobre Programação Orientada a Objetos, tratamento de exceções, organização do código e aplicação de boas práticas de desenvolvimento.

Em conjunto com o Java, utilizei o Spring Boot como principal *framework* para o desenvolvimento das aplicações. Ele era empregado na estruturação dos microsserviços e na implementação das APIs REST, permitindo organizar as diferentes camadas da aplicação e facilitar o desenvolvimento das funcionalidades. Durante as atividades, tive contato com recursos relacionados à criação de *endpoints*, validação de dados, implementação de regras de negócio e integração entre diferentes componentes do sistema.

Para o armazenamento e gerenciamento de dados, tive contato com o MongoDB, banco de dados NoSQL utilizado pelas aplicações. Minha interação com o banco ocorreu principalmente durante o desenvolvimento e a manutenção das funcionalidades, por meio da consulta e análise dos dados necessários para compreender determinadas regras de negócio, validar o comportamento da aplicação e auxiliar na investigação de possíveis falhas. Essa experiência também permitiu relacionar conhecimentos adquiridos nas disciplinas de Banco de Dados com situações encontradas no ambiente profissional.

Como ambiente de desenvolvimento integrado, utilizei o IntelliJ IDEA, principalmente para escrever, analisar, executar e depurar os projetos desenvolvidos em Java e Spring Boot. A ferramenta esteve presente de forma constante nas atividades realizadas durante minha atuação como trainee, oferecendo recursos que facilitavam a navegação pelo código, a identificação de problemas e a execução local das aplicações.

Para o controle de versões, a equipe utilizava o Git, em conjunto com o GitHub, para hospedagem e gerenciamento dos repositórios de código. Essas ferramentas possibilitavam registrar as alterações realizadas no código-fonte, trabalhar com *branches* e colaborar com os

demais desenvolvedores. Durante minha atuação como trainee, utilizei recursos relacionados à criação e atualização de *branches*, sincronização dos repositórios, resolução de conflitos e envio das alterações para revisão de código (*Code Review*).

Para realizar a validação das APIs REST, utilizei o Insomnia, ferramenta que possibilitava enviar requisições HTTP aos *endpoints* da aplicação e analisar as respostas retornadas pelos serviços. Durante o desenvolvimento, essa ferramenta era utilizada para verificar o funcionamento das funcionalidades implementadas, analisar os dados retornados, conferir os códigos de resposta HTTP e identificar possíveis comportamentos inesperados antes que as alterações avançassem para outras etapas do processo.

Além das validações realizadas por meio do Insomnia, utilizei o JUnit para a criação e execução de testes unitários em Java. Os testes permitiam validar partes específicas das funcionalidades e das regras de negócio, contribuindo para identificar possíveis falhas e verificar se as alterações realizadas apresentavam o comportamento esperado. O contato com testes automatizados reforçou a importância da validação contínua do código e da prevenção de regressões durante a evolução das aplicações.

Os microsserviços eram executados utilizando Docker, que possibilitava o uso de contêineres para padronizar os ambientes e suas dependências. Durante minha atuação como trainee, utilizei os ambientes disponibilizados pela equipe e acompanhei o funcionamento das aplicações nesse contexto, o que contribuiu para compreender melhor a utilização de contêineres em sistemas distribuídos e a importância da padronização dos ambientes de desenvolvimento e execução.

A infraestrutura do projeto estava hospedada na Amazon Web Services (AWS). Durante minha atuação como trainee, acompanhei algumas implantações realizadas pela equipe e tive contato com o Amazon CloudWatch, serviço utilizado para monitoramento e observabilidade das aplicações. Por meio dele, consultava logs de execução dos microsserviços para auxiliar na identificação e análise de falhas durante o desenvolvimento e a validação das funcionalidades. Embora eu não atuasse diretamente na administração da infraestrutura em nuvem, esse contato permitiu compreender melhor o comportamento das aplicações nos diferentes ambientes e a importância do monitoramento em sistemas distribuídos.

Também tive contato com o Jenkins, utilizado pela equipe na automação de etapas relacionadas à integração e à entrega das aplicações. A ferramenta fazia parte do fluxo de desenvolvimento adotado no projeto e auxiliava na execução automatizada de processos relacionados à construção, validação e disponibilização das alterações realizadas no código. Esse contato contribuiu para minha compreensão sobre a importância da automação no ciclo de desenvolvimento de software e sobre como as alterações realizadas pelos desenvolvedores percorrem diferentes etapas antes de serem disponibilizadas nos ambientes da aplicação.

Além dessas tecnologias, utilizei ferramentas de comunicação e gerenciamento de atividades para acompanhar as demandas atribuídas à equipe, registrar o andamento das

tarefas e participar da organização das *Sprints*. A combinação dessas ferramentas com as práticas do Scrum possibilitava acompanhar o fluxo de desenvolvimento desde a definição da demanda até sua implementação, revisão, validação e posterior disponibilização.

O contato com Java, Spring Boot, MongoDB, IntelliJ IDEA, Git, GitHub, Insomnia, JUnit, Docker, AWS, Amazon CloudWatch e Jenkins ampliou minha compreensão sobre o ecossistema envolvido no desenvolvimento profissional de software. Mais do que aprender ferramentas isoladamente, essa experiência permitiu compreender como diferentes tecnologias são utilizadas de maneira integrada durante o ciclo de desenvolvimento, desde a implementação e o armazenamento de dados até os testes, controle de versão, monitoramento e entrega das aplicações.

4.5 Desafios Enfrentados

Durante minha atuação como trainee, enfrentei diferentes desafios que foram importantes para meu desenvolvimento profissional. Um dos primeiros foi compreender como os sistemas desenvolvidos pela empresa estavam estruturados. Como a aplicação era composta por múltiplos microsserviços interconectados, uma alteração em determinado componente poderia afetar o comportamento de outros serviços, tornando necessária uma análise cuidadosa antes da realização de modificações.

Outro desafio significativo foi compreender as regras de negócio que já haviam sido implementadas por outros desenvolvedores. Em diferentes situações, precisei analisar códigos existentes, compreender a lógica utilizada nas implementações e avaliar como determinadas alterações poderiam afetar outras partes da aplicação. Essa atividade exigiu atenção, capacidade de análise e preocupação constante com a estabilidade e o funcionamento adequado do sistema.

Durante o desenvolvimento de algumas funcionalidades, também me deparei com problemas cuja origem não era imediatamente identificável. Em determinadas situações, um erro apresentado pela aplicação não estava diretamente relacionado ao código que eu havia desenvolvido, mas à comunicação entre microsserviços, à inconsistência de dados ou a alguma regra de negócio já existente. Nesses casos, aprendi a utilizar recursos de depuração, testes e análise de logs por meio do Amazon CloudWatch para investigar as possíveis causas das falhas antes de propor uma solução.

Outro aspecto importante para meu aprendizado foi a necessidade de trabalhar de forma colaborativa em um projeto de maior porte. Diferentemente de muitos projetos acadêmicos, nos quais o desenvolvimento geralmente ocorre com equipes menores, durante minha atuação como trainee passei a trabalhar em uma aplicação desenvolvida e mantida por diferentes profissionais. Esse contexto exigiu maior atenção aos padrões definidos pela equipe, à organização do código, ao controle de versões e à comunicação constante com os demais profissionais envolvidos.

Embora esses desafios tenham exigido dedicação e aprendizado contínuo, foram importantes para meu desenvolvimento profissional. Cada situação enfrentada proporcionou novos conhecimentos e contribuiu para ampliar minha capacidade de análise e resolução de problemas de maneira mais estruturada e autônoma.

4.6 Aprendizados Adquiridos

Durante minha atuação como trainee, enfrentei diferentes desafios que foram importantes para meu desenvolvimento profissional. Um dos primeiros foi compreender como os sistemas desenvolvidos pela empresa estavam estruturados. Como a aplicação era composta por múltiplos microsserviços interconectados, uma alteração em determinado componente poderia afetar o comportamento de outros serviços, tornando necessária uma análise cuidadosa antes da realização de modificações.

Outro desafio significativo foi compreender as regras de negócio que já haviam sido implementadas por outros desenvolvedores. Em diferentes situações, precisei analisar códigos existentes, compreender a lógica utilizada nas implementações e avaliar como determinadas alterações poderiam afetar outras partes da aplicação. Essa atividade exigiu atenção, capacidade de análise e preocupação constante com a estabilidade e o funcionamento adequado do sistema.

Durante o desenvolvimento de algumas funcionalidades, também me deparei com problemas cuja origem não era imediatamente identificável. Em determinadas situações, um erro apresentado pela aplicação não estava diretamente relacionado ao código que eu havia desenvolvido, mas à comunicação entre microsserviços, à inconsistência de dados ou a alguma regra de negócio já existente. Nesses casos, aprendi a utilizar recursos de depuração, testes e análise de logs por meio do Amazon CloudWatch para investigar as possíveis causas das falhas antes de propor uma solução.

Outro aspecto importante para meu aprendizado foi a necessidade de trabalhar de forma colaborativa em um projeto de maior porte. Diferentemente de muitos projetos acadêmicos, nos quais o desenvolvimento geralmente ocorre com equipes menores, durante minha atuação como trainee passei a trabalhar em uma aplicação desenvolvida e mantida por diferentes profissionais. Esse contexto exigiu maior atenção aos padrões definidos pela equipe, à organização do código, ao controle de versões e à comunicação constante com os demais profissionais envolvidos.

Embora esses desafios tenham exigido dedicação e aprendizado contínuo, foram importantes para meu desenvolvimento profissional. Cada situação enfrentada proporcionou novos conhecimentos e contribuiu para ampliar minha capacidade de análise e resolução de problemas de maneira mais estruturada e autônoma.

4.7 Relação entre a Graduação e a Experiência Profissional

A experiência adquirida durante minha atuação como trainee evidenciou de maneira clara como os conhecimentos abordados ao longo do curso de Tecnologia em Análise e Desenvolvimento de Sistemas se relacionam com as atividades realizadas no ambiente profissional.

Os conhecimentos adquiridos em Programação Orientada a Objetos contribuíram para que eu compreendesse a estrutura dos projetos desenvolvidos em Java, os padrões adotados pela equipe e a organização das classes e dos diferentes componentes da aplicação. Conceitos como encapsulamento, abstração, herança e polimorfismo, estudados durante a graduação, passaram a ser observados de maneira prática no desenvolvimento e na manutenção das aplicações.

Os conhecimentos adquiridos nas disciplinas relacionadas a Banco de Dados também foram importantes para compreender como as informações eram armazenadas, consultadas e utilizadas pelas aplicações. Durante minha atuação, pude relacionar esses conhecimentos às atividades de consulta e análise de dados armazenados no MongoDB. Da mesma forma, os conteúdos estudados em Desenvolvimento Web contribuíram para compreender a comunicação entre diferentes componentes dos sistemas por meio de APIs REST e do protocolo HTTP.

Os conhecimentos adquiridos nas disciplinas relacionadas à Engenharia de Software foram especialmente relevantes durante minha atuação como trainee. Conceitos como levantamento e análise de requisitos, arquitetura de software, controle de versão, testes de software e metodologias ágeis deixaram de ser apenas conteúdos estudados no ambiente acadêmico e passaram a fazer parte das atividades e dos processos que acompanhei no cotidiano da equipe de desenvolvimento.

A relação entre teoria e prática fortaleceu minha compreensão sobre a importância da formação acadêmica para a atuação profissional. Ao mesmo tempo, essa experiência demonstrou que o aprendizado contínuo é fundamental na área de Tecnologia da Informação, considerando a constante evolução das tecnologias, ferramentas, práticas e necessidades encontradas no mercado de desenvolvimento de software.

4.8 Considerações sobre a Experiência

Ao final do programa de trainee, pude perceber o quanto essa experiência foi significativa para minha formação acadêmica e profissional. Além de aplicar conhecimentos adquiridos durante a graduação, tive a oportunidade de compreender como equipes de desenvolvimento organizam seus processos, utilizam diferentes tecnologias e trabalham de maneira colaborativa para desenvolver e manter soluções de software.

A interação com profissionais mais experientes, aliada à participação em atividades relacionadas a projetos reais, proporcionou aprendizados que foram além dos conhecimentos

técnicos. Passei a compreender de maneira mais ampla a importância do trabalho em equipe, da responsabilidade na execução das atividades, da comunicação clara e do aprendizado contínuo no ambiente profissional.

Outro aspecto importante foi a mudança na minha percepção sobre o desenvolvimento de software. Antes dessa experiência profissional, minha atenção estava concentrada principalmente na implementação das funcionalidades. A partir da vivência em uma equipe de desenvolvimento, compreendi que a qualidade de um sistema depende de diferentes fatores, incluindo planejamento, arquitetura, testes, documentação, revisão de código, monitoramento e colaboração entre profissionais de diferentes áreas.

Por fim, considero que a experiência relatada neste trabalho foi de grande importância para minha formação profissional. Os conhecimentos técnicos adquiridos, aliados ao desenvolvimento de competências comportamentais, contribuíram para ampliar minha preparação para atuar na área de Tecnologia da Informação e minha compreensão sobre os desafios, processos e responsabilidades envolvidos no desenvolvimento de software em ambientes corporativos.

5 CONCLUSÃO

O presente relato de experiência teve como objetivo apresentar minha atuação como trainee de Engenharia de Software na Empresa X, cujo nome foi ocultado devido a questões jurídicas, destacando as atividades desenvolvidas, as tecnologias utilizadas, os desafios enfrentados e os principais conhecimentos adquiridos ao longo dessa experiência profissional. A atuação proporcionou uma oportunidade significativa de relacionar os conhecimentos adquiridos durante a graduação com sua aplicação prática em um ambiente profissional de desenvolvimento de software.

Durante minha atuação como trainee, pude aplicar conhecimentos relacionados à Programação Orientada a Objetos, Engenharia de Software, Desenvolvimento Web, Banco de Dados, arquitetura de microsserviços e computação em nuvem. O contato com essas áreas em um contexto profissional permitiu compreender que o desenvolvimento de software não se limita à implementação de funcionalidades, mas envolve também planejamento, organização, comunicação entre os integrantes da equipe, testes, controle de qualidade, monitoramento e melhoria contínua dos processos.

O convívio com profissionais mais experientes foi importante para meu desenvolvimento técnico e profissional. Ao longo dessa experiência, participei de revisões de código (*Code Reviews*), reuniões de planejamento, cerimônias relacionadas ao Scrum e discussões técnicas. Essas atividades contribuíram para ampliar minha compreensão sobre o ciclo de desenvolvimento de software e sobre a importância da colaboração entre profissionais de diferentes áreas para a construção e manutenção de soluções de qualidade.

Os desafios enfrentados durante minha atuação como trainee também contribuíram significativamente para meu desenvolvimento. Compreender sistemas já existentes, interpretar regras de negócio, analisar códigos desenvolvidos por outros profissionais e investigar a origem de falhas técnicas foram atividades que exigiram dedicação, estudo contínuo e capacidade de análise. Cada desafio proporcionou novos aprendizados e contribuiu para ampliar minha autonomia e minha confiança na resolução de problemas progressivamente mais complexos.

Além dos conhecimentos técnicos, essa experiência contribuiu para o desenvolvimento de competências comportamentais importantes para minha atuação profissional. Ao longo do período, desenvolvi habilidades relacionadas à comunicação, organização, trabalho em equipe, gerenciamento de atividades, responsabilidade e adaptação a novas situações. Essas competências complementaram os conhecimentos adquiridos durante a graduação e demonstraram a importância de aspectos técnicos e interpessoais para a atuação na área de Tecnologia da Informação.

Outro aprendizado proporcionado por essa experiência foi compreender que a formação de um profissional de tecnologia exige aprendizado contínuo. A constante evolução de ferramentas, linguagens, metodologias e arquiteturas demanda atualização e disposição para adquirir novos conhecimentos. Minha atuação como trainee permitiu vivenciar essa realidade e compreender a importância de acompanhar as transformações tecnológicas e buscar continuamente o aprimoramento profissional.

Essa experiência também evidenciou a importância da aproximação entre a formação acadêmica e a prática profissional. A oportunidade de participar de atividades relacionadas a projetos reais possibilitou aplicar conhecimentos adquiridos durante a graduação e compreender como conceitos teóricos são utilizados na resolução de problemas encontrados no ambiente corporativo. Essa relação entre universidade e mercado de trabalho ampliou minha compreensão sobre a profissão e contribuiu para minha preparação para os desafios relacionados à atuação na área de desenvolvimento de software.

Por fim, concluo que minha atuação como trainee de Engenharia de Software representou uma etapa importante para minha formação acadêmica e profissional durante o curso de Tecnologia em Análise e Desenvolvimento de Sistemas. Mais do que proporcionar experiência prática, essa vivência contribuiu para meu amadurecimento profissional, para o desenvolvimento de maior autonomia e capacidade de colaboração e para o aprofundamento dos meus conhecimentos técnicos. Os aprendizados adquiridos nesse período constituem uma base importante para a continuidade da minha trajetória profissional e reforçam meu interesse em continuar aprimorando meus conhecimentos, acompanhando a constante evolução da Tecnologia da Informação e contribuindo, de maneira ética e responsável, para o desenvolvimento de soluções de software.

REFERÊNCIAS

AWS. **What is cloud computing?** Seattle: Amazon Web Services, 2026. Disponível em: <https://aws.amazon.com/what-is-cloud-computing/>. Acesso em: 3 ago. 2026.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 10. ed. São Paulo: Pearson Education do Brasil, 2017.

DOCKER INC. **What is Docker?** Palo Alto: Docker Inc., 2026. Disponível em: <https://www.docker.com/resources/what-container/>. Acesso em: 3 ago. 2026.

FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado em Ciência da Computação) – University of California, Irvine, 2000. Disponível em: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>. Acesso em: 3 ago. 2026.

FOWLER, Martin. **Microservices: a definition of this new architectural term**. [S. l.]: Martin Fowler, 2014. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 3 ago. 2026.

ORACLE. **Java documentation**. Austin: Oracle Corporation, 2026. Disponível em: <https://docs.oracle.com/en/java/>. Acesso em: 3 ago. 2026.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software: uma abordagem profissional**. 9. ed. Porto Alegre: AMGH, 2021.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**. [S. l.]: Scrum.org, 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>. Acesso em: 3 ago. 2026.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.

SPRING. **Spring Boot Reference Documentation**. [S. l.]: VMware, 2026. Disponível em: <https://docs.spring.io/spring-boot/docs/current/reference/html/>. Acesso em: 3 ago. 2026.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

TCC 2

Assunto:	TCC 2
Assinado por:	Bruno Nogueira
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Documento Original

Documento assinado eletronicamente por:

- **Bruno Lacerda Nogueira, ALUNO (201922010014) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 20/08/2026 13:07:44.

Este documento foi armazenado no SUAP em 20/08/2026. 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: 1959328

Código de Autenticação: 42630175a1

