



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

KEILLA VITÓRIA FELIPE BEZERRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA DOCK**

MONTEIRO - PB

2023

KEILLA VITÓRIA FELIPE BEZERRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL NO PROCESSO DE
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA DOCK**

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

Orientador: Prof. Dr. Tiago Brasileiro Araújo.

MONTEIRO - PB

2023

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

B574r Bezerra, Keilla Vitória Felipe.

Relato de experiência profissional no processo de desenvolvimento de software na Empresa DOCK / Keilla Vitória Felipe Bezerra – Monteiro-PB. 2023.

46 fls. : 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.

Orientador: Prof. Dr. Tiago Brasileiro Araújo.

1. Software - Desenvolvimento 2. Tecnologia – Informação 3. Trabalho Remoto - Estágio 4. Empresa DOCK – São Paulo -SP I.
Título.

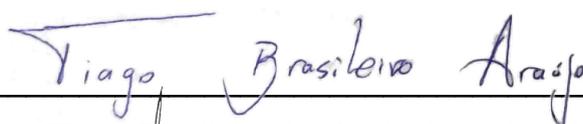
CDU 004.453

KEILLA VITÓRIA FELIPE BEZERRA

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

Aprovado em 20 de junho de 2023.

BANCA EXAMINADORA



Prof. Tiago Brasileiro Araújo (Orientador - IFPB)



Prof. Gilmar de Jesus Barros



Prof. Wagner de Oliveira Santos

RESUMO

Nos dias atuais, o mercado de TI cresce de forma exponencial em todo o mundo, firmando-se como uma das áreas com maior necessidade de mão-de-obra. Nesse contexto, os estágios supervisionados nas áreas de TI surgem como uma importante forma de inserir alunos em formação nas empresas de tecnologia. Dessa forma, neste relatório serão apresentadas as atividades realizadas pela discente Keilla Vitória Felipe Bezerra, no período de estágio supervisionado na empresa Dock, em virtude da conclusão do curso de Análise e Desenvolvimento de Sistemas. O estágio foi realizado no modelo de trabalho remoto. A discente realizou atividades como o monitoramento de *dashboards*, desenvolvimento de código, correção de *bugs*, testes de *software*, desenvolvimento de documentação, desenho de diagramas, *queries* em banco de dados, *code review*, estudo de cursos e outras atividades internas da empresa. A atuação no estágio permitiu à discente aplicar os conceitos aprendidos em sala de aula e obter novos conhecimentos e experiências, contribuindo consideravelmente para seu desenvolvimento profissional e capacitando-a para o mercado de trabalho.

Palavras-chave: estágio supervisionado; ti; empresa de tecnologia; desenvolvimento.

ABSTRACT

Nowadays, the IT market is growing exponentially worldwide, establishing itself as one of the areas with the greatest need for labor. In this context, the supervised internships in IT areas arise as an important way to insert students in training in technology companies. In this way, this report will present the activities carried out by the student Keilla Vitória Felipe Bezerra, during the supervised internship period at the company Dock, as a result of the conclusion of the Systems Analysis and Development course. The internship was carried out in the remote work model. The student performed activities such as dashboard monitoring, code development, bug fixing, software testing, documentation development, diagram design, database queries, code review, course studies, and other internal company activities. The internship allowed the student to apply the concepts learned in the classroom and to obtain new knowledge and experiences, contributing considerably to her professional development and qualifying her for the job market.

Key-words: supervised internship; ti; technology companies; development.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - Logomarca da empresa Dock

FIGURA 2 - Escritório e sala de reuniões da empresa Dock

FIGURA 3 - Representação da união Dock

FIGURA 4 - Interface da plataforma Jira

FIGURA 5 - Conteúdo do curso preparatório Java

FIGURA 6 - Cursos realizados na plataforma Alura

FIGURA 7 - Exemplo de *dashboard* da plataforma Splunk

FIGURA 8 - Exemplo de *query* de consulta de *logs* na plataforma Splunk

FIGURA 9 - Interface da ferramenta Postman Runner

FIGURA 10 - Interface da plataforma DBeaver

FIGURA 11 - Exemplo de consulta SQL

FIGURA 12 - Interface da plataforma Studio 3T

LISTA DE ABREVIATURAS

ADS	<i>Análise e Desenvolvimento de Sistemas</i>
AOP	<i>Aspect Oriented Programming</i> (Programação Orientada a Aspectos)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
DAC	<i>Desenvolvimento de Aplicações Corporativas</i>
PO	<i>Product Owner</i> (Proprietário do Produto)
REST	<i>Representational State Transfer</i> (Transferência de Estado Representacional)
SGBD	<i>Sistema de Gerenciamento de Banco de Dados</i>
SQL	<i>Structured Query Language</i> (Linguagem de Consulta Estruturada)
UML	<i>Unified Modeling Language</i> (Linguagem Unificada de Modelagem)
VPN	<i>Virtual Private Network</i> (Rede privada virtual)

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	13
3.1.3 Perfil da empresa	14
3.2 Atividades desenvolvidas	15
3.2.1 Cursos preparatórios	18
3.2.2 Monitoramento de dashs e queries do Splunk	21
3.2.3 Atendimento de chamados internos	24
3.2.4 Desenvolvimento de código	26
3.2.5 Correção de bug	29
3.2.6 Testes de software	30
3.2.7 Code review	32
3.2.8 Queries em bancos de dados	34
3.2.9 Desenvolvimento de diagramas	37
3.2.10 Desenvolvimento de documentação de processos	38
4 CONCLUSÃO	40
REFERÊNCIAS	42

1 INTRODUÇÃO

Nos dias atuais, o mercado de TI cresce de forma exponencial em todo o mundo, firmando-se como uma das áreas com maior necessidade de mão-de-obra. Entretanto, há uma escassez de profissionais de TI qualificados no mercado de trabalho. No Brasil, um estudo realizado pela Brasscom (2021) aponta que, até o final de 2021, o país possuía demanda média anual de cerca de 159 mil profissionais. Por outro lado, apenas 53 mil profissionais se formaram na área anualmente. O relatório estima ainda que, até 2025, as empresas de tecnologia demandem por volta de 797 mil vagas. Esses dados enfatizam a necessidade de profissionais formados e capacitados para suprir as vagas ofertadas.

Para os profissionais em processo de formação, o estágio permite um primeiro contato do estudante com o mercado de trabalho dentro de sua área, auxiliando no processo de aprendizagem do aluno. Dessa forma, o aluno pode colocar em prática os conteúdos trabalhados em sala de aula, além de trocar experiências com profissionais que já atuam na área (GRUPO INTEGRADO, 2021).

Nesse contexto, os estágios supervisionados nas áreas de TI surgem como uma importante forma de inserir alunos em formação nas empresas de tecnologia. Além disso, a combinação do conhecimento teórico obtido em sala de aula com a vivência em ambiente corporativo possibilita a formação de profissionais melhor capacitados para o mercado de trabalho (BRAGA, 2021).

Diante do que foi apresentado, o presente documento tem como propósito relatar as atividades desenvolvidas e experiências adquiridas pela discente durante um estágio supervisionado realizado em uma empresa de tecnologia em meios de pagamentos.

O estágio foi realizado na empresa Dock, que oferece soluções tecnológicas na área de meios de pagamentos, fornecendo uma plataforma global de pagamentos e *banking* para organizações de todos os tamanhos. A empresa surgiu a partir da unificação de três empresas: Conductor, Dock e Muxi, passando a integrar também as empresas BPP e Cacao, posteriormente. Trata-se de uma empresa multinacional que, além de estar localizada no Brasil, estado de São Paulo,

também possui escritórios nos países do México, Chile, Argentina, Colômbia e Peru (DOCK, 2023).

A atuação da empresa tem seu foco no mercado de meios de pagamento e oferece soluções que partem desde a interação com o cliente final, por meio de aplicativos, até a validação de transações e prevenção de fraudes (DOCK FLUID, 2021).

Dessa forma, a discente atuou como estagiária de engenharia de *software*, na área de *Risk & Compliance*, responsável por garantir a segurança e integridade nas operações realizadas pelos clientes. Durante o período de estágio, a discente fez parte do time de antifraude, responsável pelo monitoramento de transações fraudulentas, desenvolvimento de soluções flexíveis e atendimento de chamados para resolução de problemas.

A discente realizou atividades como o monitoramento de *dashboards*, desenvolvimento de código, correção de *bugs* do sistema, testes de *software*, desenvolvimento de documentação, desenho de diagramas, atendimento e abertura de chamados internos, *queries* em banco de dados, revisão de código com colaboração de membros da equipe, estudo de curso preparatório Java e outros cursos relacionados à tecnologias utilizadas no estágio.

O presente relatório está dividido em quatro seções, onde a primeira seção corresponde à introdução. A segunda seção contém os objetivos gerais e específicos do estágio realizado. Na terceira seção encontram-se as atividades realizadas pela discente durante o estágio, além da apresentação do perfil da empresa Dock. Por fim, na quarta seção está a conclusão, contendo as considerações finais acerca do estágio.

2 OBJETIVOS

2.1 Objetivo geral

O estágio teve como objetivo principal manter e desenvolver novas funcionalidades para um sistema responsável por garantir a segurança e integridade nas operações realizadas pelos clientes da empresa Dock.

2.2 Objetivos específicos

- Desenhar, codificar, testar e documentar um *software* desenvolvido em Java;
- Atuar na manutenção e desenvolvimento de funcionalidades para o sistema de antifraude;
- Fornecer suporte aos clientes para uso do *software*;

3 RELATO DE ESTÁGIO

Nessa seção, serão apresentadas informações referentes à empresa Dock, seguida 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 na empresa Dock, que oferece soluções tecnológicas na área de meios de pagamentos, fornecendo uma plataforma global de pagamentos e *banking* para organizações de todos os tamanhos. Trata-se de uma empresa multinacional, sendo localizada no Brasil, Avenida Tamboré, 267, Alphaville, Barueri, São Paulo, além de possuir escritórios nos países do México, Chile, Argentina, Colômbia e Peru (DOCK, 2023). A logomarca da empresa Dock pode ser observada na Figura 1.

Figura 1 - Logomarca da empresa Dock



Fonte: Labs

A empresa existe fisicamente no estado de São Paulo, dispondo de um escritório e sala de reuniões, como evidenciado na Figura 2, além de possuir uma área de lazer e descanso. Antes da pandemia causada pelo vírus COVID-19, a empresa funcionava de forma inteiramente presencial. Entretanto, com o início da pandemia, a empresa adotou o modelo *anywhere office*, possibilitando aos colaboradores a execução do trabalho em qualquer lugar desejado.

Figura 2 - Escritório e sala de reuniões da empresa Dock



Fonte: Glassdoor

As reuniões virtuais da empresa são realizadas por meio da ferramenta Zoom¹, que permite a efetuação de videoconferências entre usuários de diversos dispositivos (ZOOM, 2023?). O repasse de comunicados oficiais e documentações essenciais é efetuado através do Gmail, serviço gratuito de correio eletrônico desenvolvido pela empresa Google (TECHTUDO, 2023?), onde cada funcionário possui um *email* corporativo. A comunicação diária entre pessoas de diferentes setores da empresa é feita através de vários canais e grupos de mensagem, organizados na plataforma Slack, aplicativo de mensagens para comunicação interna de empresas (SLACK, 2023?).

3.1.2 Dados referentes ao estágio

O estágio supervisionado teve início no dia 8 de Novembro de 2021 e finalizou no dia 5 de maio de 2023, tendo duração de aproximadamente 1 ano e 6 meses, possuindo carga horária de trinta horas semanais. A carga horária diária era de seis horas, iniciando a partir das 09:00 (nove) horas da manhã e finalização do expediente às 16:00 (dezesseis) horas, com pausa de uma hora para o almoço. A

¹ Link para ferramenta: <https://zoom.us/pt>

finalização do período de estágio se deu devido à efetivação da estagiária, no dia 8 de maio de 2023, que passou a atuar como Engenheira de *Software* Júnior na empresa, com jornada de trabalho de quarenta horas semanais. A atuação na empresa se estende até os dias atuais.

O estágio iniciou sob gerência de Rodolfo Freire e tutoria do desenvolvedor pleno Bruno Santos, entretanto, com o decorrer do estágio, ocorreram mudanças: por cerca de 10 meses, a mentoria e gerência foram assumidas inteiramente por Rodolfo Freire. Após esse período, a gerência e mentoria passaram para Luan Brito, tendo duração de 3 meses. Por fim, a gerência do estágio foi assumida por Marcos Dias e a mentoria, pelo desenvolvedor especialista Carlos Maradeia. Além disso, todos os desenvolvedores que integraram a equipe contribuíram direta ou indiretamente para o desenvolvimento pessoal e profissional da estagiária.

3.1.3 Perfil da empresa

A empresa Dock surgiu a partir da união de três empresas: Conductor, Dock e Muxi, representada na Figura 3, e posteriormente passou a integrar também as empresas BPP e Cacao. A união das empresas ocorreu recentemente, no ano de 2021, no entanto, elas já eram conhecidas no mercado, contando com anos de experiência. Com a unificação, as empresas consolidaram-se sob uma única identidade, a Dock.

A atuação da empresa tem seu foco no mercado de meios de pagamento, com a finalidade de entregar soluções inovadoras e flexíveis de forma a potencializar processos essenciais que aceleram a capacidade das empresas e bancos, resultando em um maior acesso dos consumidores aos serviços financeiros e uma melhor jornada do cliente. As soluções oferecidas partem desde a interação com o cliente final, por meio de aplicativos, até a validação de transações e prevenção de fraudes (DOCK FLUID, 2021).

Figura 3 - Representação da união Dock



Fonte: Dock Fluid

Atualmente, a empresa viabiliza mais de 70 milhões de contas ativas, com cerca de 70 bilhões de transações e 180 bilhões de dólares processados anualmente. Desde o ano de 2014, a empresa apresenta crescimento médio anual de 45%. As soluções são divididas em quatro frentes: fornecimento de conta digital completa, cartões e crédito das principais bandeiras, solução de aquisição *white label* e redução de riscos das operações, de ponta a ponta (DOCK, 2023).

A empresa conta com mais de 2 mil funcionários, apelidados *dockers*, que são divididos em equipes (*squads*), atuando em áreas específicas da empresa. Especificamente na área de Tech, relacionada às equipes de desenvolvimento, as *squads* são formadas por um gerente de tecnologia, um PO, desenvolvedores e podem contar com a presença de estagiários.

Na organização das atividades, os times adotam o modelo ágil Scrum, descrito no tópico seguinte, e fazem uso de ferramentas para apoiar no gerenciamento das tarefas seguindo o modelo utilizado. Com o modelo, as tarefas são desenvolvidas de forma mais eficiente e com maior rapidez nas entregas.

3.2 Atividades desenvolvidas

As atividades relatadas foram desenvolvidas pela discente durante o período de estágio supervisionado, que atuou como estagiária da área de engenharia de *software*, no time de *Risk & Compliance*, responsável por garantir a segurança e integridade nas operações realizadas pelos clientes. O monitoramento de *dashes* do Splunk, atendimento de chamados internos e outras atividades realizadas pela

estagiária eram extremamente importantes no dia a dia da empresa e, por esse motivo, deveriam ser executadas diariamente.

Por outro lado, as tarefas de desenvolvimento de código, revisão de código (*code review*), consultas no banco de dados e Splunk, correção de *bugs*, desenvolvimento de testes, diagramas e documentação, eram feitas dentro de *sprints*, visto que a equipe seguia o modelo ágil Scrum para o processo de desenvolvimento.

O Scrum é uma metodologia ágil que tem como principal foco auxiliar na gestão de projetos que possuam prazos curtos de entrega, existindo em razão da necessidade de disponibilizar agilidade em projetos de alta complexidade. O Scrum funciona a partir de ciclos, denominados de *sprints*, que são períodos de tempo para desenvolvimento de cada fase. Ao fim de cada fase, ocorre a revisão e validação do produto. Por fim, é realizado um novo planejamento para o início da próxima *sprint*. (EGOV, 2021).

Antes do início da *sprint*, era realizada a *planning*, que era uma reunião para planejamento das atividades a serem realizadas, de acordo com sua prioridade. Nessa reunião, participavam o gerente de tecnologia, o PO e toda a equipe de desenvolvedores. O PO era responsável por descrever cada item priorizado, com os desenvolvedores livres para discutir a parte técnica a ser desenvolvida e fazer questionamentos.

Após a descrição, era realizado o *planning poker*, estratégia para estimar o esforço necessário para desenvolvimento das atividades, onde cada integrante da equipe pontua a tarefa apresentada, utilizando valores da Sequência de Fibonacci (sequência em que, a partir dos números iniciais 0 e 1, os seguintes são obtidos a partir da soma dos dois antecessores: 0, 1, 2, 3, 5, 8, etc.). Os pontos atribuídos são chamados de *story points*, ou pontos de história, medida para o esforço, complexidade e risco atrelados ao item. (BRASILEIRO, 2017).

Finalizado o alinhamento e atribuição de *story points* para as atividades, a *sprint* era iniciada, durando aproximadamente 15 dias. Ao final da *sprint*, eram realizadas duas reuniões: a *review* e a *retrospective*. Na *review*, ocorria a apresentação de tudo o que foi desenvolvido durante a *sprint*. Por fim, na *retrospective*, a equipe levantava os pontos positivos, pontos a melhorar e lições

aprendidas durante a *sprint* realizada, finalizando a reunião com ações a serem tomadas a partir das próximas *sprints*, buscando sempre a melhoria contínua.

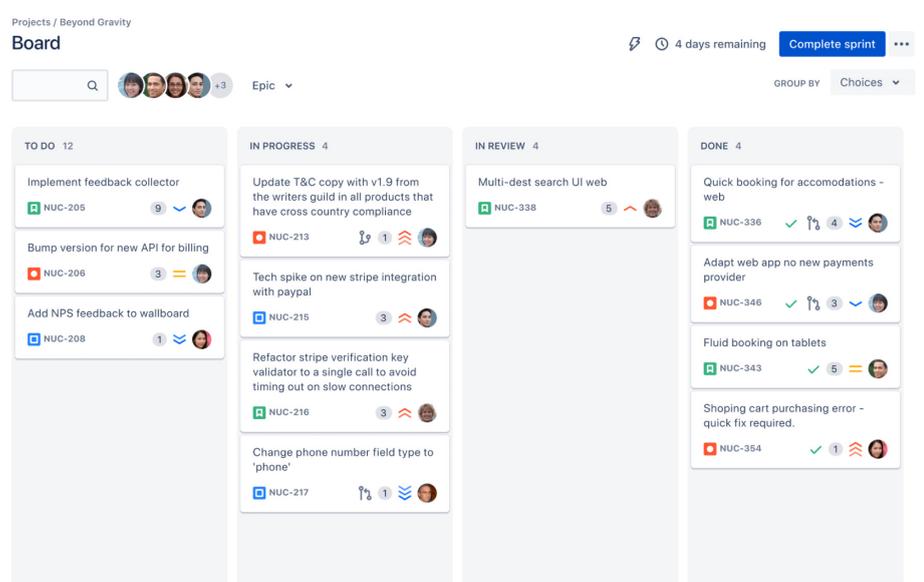
No decorrer das *sprints*, poderiam surgir demandas extras que necessitavam ser priorizadas, ou eram abertos chamados internos por outras equipes, referentes à problemas encontrados em ambiente de homologação ou produção, impactando outros times e até mesmo clientes da Dock. Nesses casos, era preciso dedicar tempo para apoiar a resolução dos chamados, o que poderia provocar atrasos no desenvolvimento das atividades propostas para a *sprint*. Como consequência, algumas tarefas se estendiam para a *sprint* seguinte, objetivando evitar a aglomeração de itens para um único membro da equipe.

Esses obstáculos diminuiriam consideravelmente com o surgimento da figura do “bombeiro”, membro do *squad* que ficava integralmente responsável por auxiliar com a fila de chamados, permitindo que os outros integrantes não fossem acionados constantemente enquanto atuavam nas suas atividades. A cada *sprint*, um membro diferente do time recebia o papel de bombeiro. A estagiária atuou como bombeira em algumas *sprints*, atuando principalmente na frente de *setup* de emissores, que será abordada nos próximos tópicos.

Para gerenciamento das *sprints*, era utilizada a plataforma Jira², desenvolvida pela Atlassian, que reúne as funcionalidades necessárias para organizar diversos tipos de projetos, oferecendo recursos essenciais para criação e controle do fluxo de trabalho, otimizando a eficiência, rastreabilidade das atividades desenvolvidas e comunicação entre o time. (HÖRLLE, 2022). Na Figura 4, é possível observar a interface da ferramenta, com a demonstração do quadro *scrum*, um painel de visualização de tarefas que separa os itens que estão em *backlog*, os que estão em andamento (*in progress*), em revisão (*code review*) ou que já foram concluídos (*done*), tornando claro todo o fluxo de trabalho.

² Link para ferramenta: <https://www.atlassian.com/br/software/jira>

Figura 4 - Interface da plataforma Jira



Fonte: Atlassian

3.2.1 Cursos preparatórios

No início do estágio, foi repassado pelo tutor um curso completo da linguagem de programação Java³, presente na plataforma Udemy⁴, para que a discente estudasse e consolidasse seus conhecimentos em relação à assuntos vistos em sala de aula. Uma vez que a principal linguagem de programação utilizada no decorrer do curso de ADS é o Java, a compreensão do conteúdo se tornou mais simples, entretanto, o curso preparatório também disponibilizava assuntos não priorizados, como o desenvolvimento de interface gráfica utilizando JavaFX⁵. Os tópicos de maior relevância de aprendizado foram destacados pelo tutor, com a orientação de que assuntos já vistos em disciplinas anteriores poderiam ser ignorados, com o objetivo de concluir o curso mais rapidamente.

O Java é uma linguagem de programação orientada a objetos de fácil entendimento que possui como principais características a dinamicidade, garantia de segurança e confiabilidade dos programas escritos, além de possuir uma vasta

³ Link para ferramenta: <https://www.java.com/pt-BR/>

⁴ Link para ferramenta: <https://www.udemy.com/pt/>

⁵ Link para ferramenta: <https://openjfx.io>

biblioteca de classes e interfaces, contendo recursos que permitem o desenvolvimento de sistemas extremamente robustos (FILGUEIRAS, 2015).

Por sua eficiência, a linguagem Java é amplamente utilizada pela Dock, sendo aplicada integralmente no sistema de antifraude, pelo qual a equipe da discente tinha responsabilidade de manter e aprimorar. Dessa forma, o estudo do curso preparatório era essencial para, além de contribuir com a aprendizagem da discente, também guiá-la na execução das atividades da equipe.

O curso contava com 398 aulas, divididas em 26 seções, contabilizando 54 horas e 14 minutos totais de duração. Na Figura 5, é possível observar os dados informados e algumas das seções disponíveis para estudo. Entre os assuntos abordados, destacam-se os conceitos básicos de programação (estruturas sequenciais, condicionais, de repetição, etc.) e orientação de objetos (herança, polimorfismo), indispensáveis no aprendizado da linguagem. O curso também propunha a criação de projetos, para colocar em prática os assuntos apresentados. A discente iniciou o curso prontamente, focando nas seções destacadas pelo tutor como importantes para o desenvolvimento dentro do *squad*.

Figura 5 - Conteúdo do curso preparatório Java

Java COMPLETO 2023 Programação Orientada a Objetos +Projetos	
Mais vendidos	4,8 ★ (41.672 classificações) 119.939 alunos
Conteúdo do curso	
26 seções • 398 aulas • Duração total: 54h 14m Expandir todas as seções	
▼ Introdução	1 aulas • 17m
▼ Conceitos de programação	7 aulas • 24m
▼ Introdução à linguagem Java	9 aulas • 41m
▼ Estrutura sequencial	12 aulas • 1h 15m
▼ Estrutura condicional	11 aulas • 54m
▼ Estruturas repetitivas	14 aulas • 1h 10m
▼ Outros tópicos básicos sobre Java	6 aulas • 49m
▼ Introdução à Programação Orientada a Objetos	12 aulas • 1h 29m
▼ Construtores, palavra this, sobrecarga, encapsulamento	11 aulas • 1h 16m
▼ Comportamento de memória, arrays, listas	20 aulas • 2h 40m
mais 16 seções	

Fonte: Udemy

Apesar da orientação inicial para a estagiária fosse focar no estudo do curso e no desenvolvimento dos projetos propostos, antes de começar a atuar diretamente, foram recebidas atividades da *squad* para serem desenvolvidas, em decorrência da falta de pessoas no time, que, com a saída do desenvolvedor pleno e tutor, passou a contar apenas com o gerente de tecnologia e os estagiários. Por isso, era necessária a colaboração dos estagiários com outras demandas, enquanto prosseguiram estudando o curso nos tempos disponíveis, o que ocasionou no atraso para finalização do mesmo. Entretanto, apesar da falta de tempo disponível para estudo e necessidade de atuação em outras atividades, foi possível finalizar os tópicos priorizados do curso.

Além do curso inicial preparatório, eram disponibilizadas pela empresa licenças mensais na plataforma de estudos Alura⁶, para que os colaboradores pudessem se aprofundar em assuntos relacionados às suas atividades dentro da equipe. A estagiária fez uso das licenças para estudar assuntos específicos de tecnologia, que posteriormente foram aplicados na resolução de tarefas da *squad* e no desenvolvimento de código. Entre os cursos realizados, destacam-se: gerenciamento de dependências com a ferramenta Maven⁷, consultas no banco de dados não-relacional MongoDB⁸, conceitos de Scrum e os princípios SOLID com a linguagem Java, como descrito na Figura 6. Além dos cursos informados, também foram estudados conceitos de mensageria com a tecnologia Apache Kafka⁹ e aplicações com o *framework* Spring¹⁰, com o objetivo de compreender melhor o funcionamento de alguns módulos do sistema de antifraude.

O aprendizado obtido com a realização dos cursos foi extremamente significativo para a discente, promovendo uma boa base de conhecimento para as práticas a serem realizadas nas demandas da *squad*. Vale ressaltar que o conhecimento adquirido no decorrer do curso de ADS foi essencial para o desempenho da discente nas atividades realizadas durante todo o período de estágio.

⁶ Link para ferramenta: <https://www.alura.com.br>

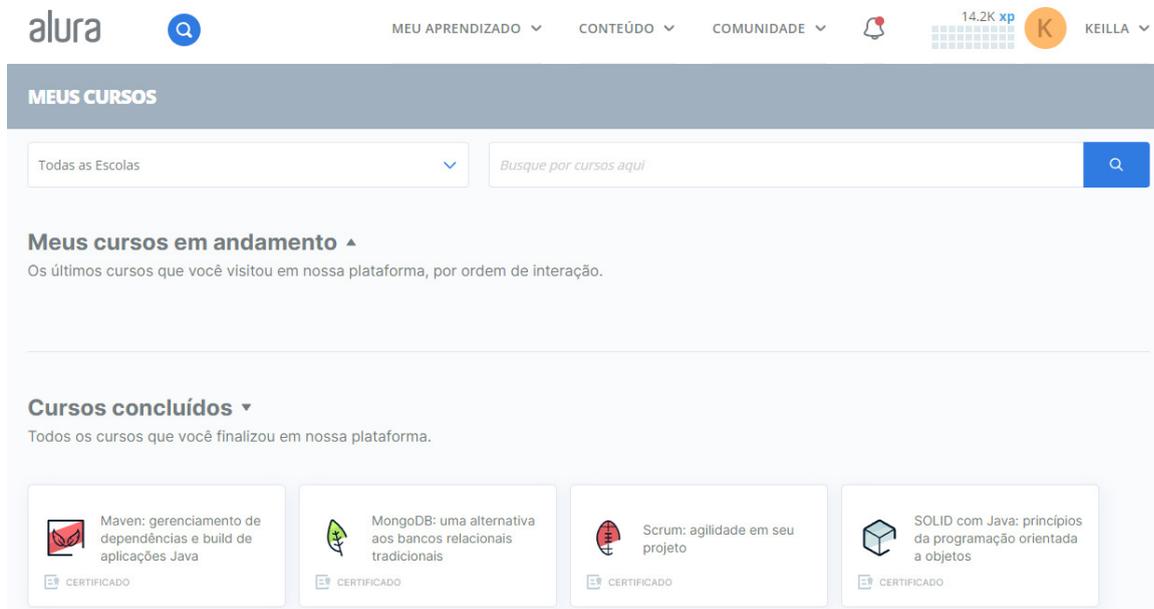
⁷ Link para ferramenta: <https://maven.apache.org>

⁸ Link para ferramenta: <https://www.mongodb.com>

⁹ Link para ferramenta: <https://kafka.apache.org>

¹⁰ Link para ferramenta: <https://spring.io>

Figura 6 - Cursos realizados na plataforma Alura



Fonte: Alura

3.2.2 Monitoramento de *dashes* e *queries* do Splunk

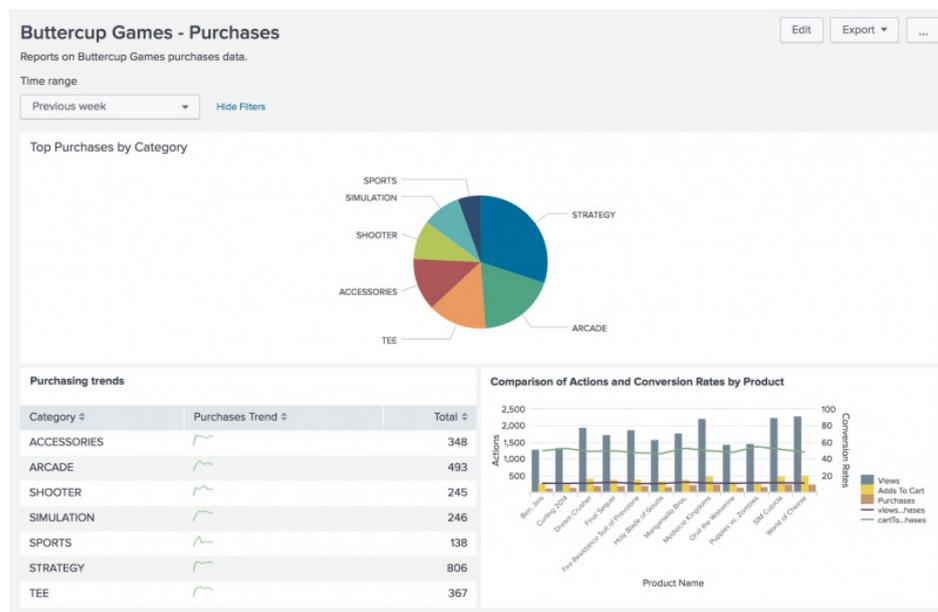
O monitoramento de *dashes* refere-se à verificação do funcionamento do fluxo das aplicações por meio da análise de painéis de informações e indicadores disponibilizados na plataforma Splunk Enterprise, que é uma solução de software desenvolvida pela empresa Splunk, que coleta, processa e distribui dados de diferentes sistemas em tempo real, oferecendo funcionalidades como: visualização dos dados através de *dashboards* interativos, monitoria de performance, mecanismos poderosos de pesquisa, entre outros (SPLUNK, 2023?).

Através da plataforma Splunk, era possível analisar todo o fluxo transacional, dados pessoais de pessoa física ou jurídica, ações de bloqueio ou desbloqueio de contas e cartões, erros ocasionados durante o envio de transações ou de dados cadastrais, ocorrência de *timeouts*, entre outras informações. Os dados observados eram coletados a partir do envio de *logs* originados na execução dos módulos que integravam a aplicação de antifraude.

Durante a realização de qualquer processo em um dos módulos do sistema de antifraude, o fluxo de execução da aplicação gerava um *log*. Segundo Oliveira (2019), o *log* é um registro contínuo com anotação de data e hora de eventos gerados automaticamente pelos sistemas de software. O formato dos *logs* varia consideravelmente de acordo com suas origens, porém, geralmente são registrados: a hora em que foi disparado o evento, detalhes do que aconteceu, informações do usuário que ocasionou o evento e detalhes sobre a reação do sistema (AIQON, 2020?). Finalizada a geração do *log*, o mesmo era enviado ao Splunk, que realizava a coleta e armazenamento.

Os *logs* das aplicações armazenados no Splunk eram verificados de duas formas: por meio de *dashboards* (painéis de informação) personalizados, como pode ser observado na Figura 7, ou através da coleta textual dos eventos, por meio de *queries* específicas, conforme a Figura 8.

Figura 7 - Exemplo de *dashboard* da plataforma Splunk



Fonte: Splunk

O monitoramento das *dashboards* era realizado diariamente pela discente, que tinha como responsabilidade disponibilizar *prints* das mesmas em um canal privado da equipe no Slack, repassando o estado atual das aplicações. Por outro

lado, a execução de *queries* de busca dos *logs* era realizada principalmente para resolução de chamados internos ou para testes de funcionamento dos módulos, tópicos que serão abordados mais adiante. Vale ressaltar que, para fazer a análise, era necessário conectar a máquina à VPN da empresa.

A VPN é um serviço que cria uma rede privada virtual para conexão entre dois pontos distintos da *internet*, permitindo a navegação de forma anônima, o que contribui para a melhora da segurança *online* (BRASIL, 2022).

Figura 8 - Exemplo de *query* de consulta de logs na plataforma Splunk

The screenshot displays the Splunk Enterprise interface. At the top, there's a navigation bar with 'Search & Reporting' selected. Below it, the 'New Search' window is open, showing a search query: `sourcetype=github_json \"alert.affected_package_name\"='org.apache.logging.log4j:log4j-api\"`. The search results show 2 events, with the first event selected. The event details are as follows:

Time	Event
12/10/21 6:17:59.000 PM	<pre>{ "action": "create", "alert": { "affected_package_name": "org.apache.logging.log4j:log4j-api", "affected_range": "< 2.15.0", "created_at": "2021-12-10T18:17:59Z", "external_identifier": "CVE-2021-44228", "external_reference": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228", "fixed_in": "2.15.0", "ghsa_id": "GHSA-jfh8-c2jp-5v3q", "id": "1538842835", "node_id": "RVA_kw00Gg84J85bu00T", "severity": "critical" }, "organization": { }, "repository": { }, "sender": { } }</pre>

At the bottom of the event details, there are filters: `host = 174.129.94.119:8088`, `source = githubwebhook`, and `sourcetype = github_json`.

Fonte: Splunk

Ao realizar a análise das *dashboards*, caso fosse identificado algum comportamento irregular do sistema, como uma quantidade de *timeouts* acima do esperado ou uma baixa quantidade de arquivos processados, por exemplo, era necessário informar imediatamente ao gestor da equipe, para que pudesse ser alinhado sobre quais ações deveriam ser tomadas para neutralizar o problema.

Em alguns casos, a ação tomada era a abertura de chamados para a equipe de infraestrutura, solicitando a re-inicialização do módulo específico ou verificação do sistema de antifraude como um todo, em outros casos, era necessário investigar

o código buscando corrigir algum *bug*, tópico que será abordado mais adiante, ou ainda, era realizado um alinhamento geral com o time, para casos de atualizações ou mudanças que ocasionassem na indisponibilidade momentânea de algum sistema ou plataforma utilizados pela aplicação de antifraude.

A atividade permitiu o aprimoramento das habilidades de análise e apuração de informações da discente, dando a oportunidade de verificar e monitorar o fluxo de uma aplicação em ambiente de produção. A discente também obteve aprendizado sobre o funcionamento do Splunk e conhecimentos sobre a criação de *queries* personalizadas na plataforma, gerando tabelas e planilhas que serviriam de evidência para resolução de vários chamados internos. Importante destacar que os conhecimentos em relação à consulta de dados adquiridos com as disciplinas de Banco de Dados I e II facilitaram o desenvolvimento da discente na atividade, uma vez que os conceitos de pesquisa eram semelhantes.

3.2.3 Atendimento de chamados internos

Os chamados são abertos para solicitar informações ou auxílio de outras equipes na resolução de problemas, sejam eles internos às equipes da empresa ou problemas que impactam diretamente os clientes. Para abertura e gerenciamento dos chamados, eram utilizadas duas ferramentas, o Jira e o Zendesk¹¹, com particularidades no uso de cada uma. A ferramenta Jira era usada essencialmente para comunicação entre equipes da empresa, o Zendesk, por outro lado, era utilizado na comunicação com o cliente, geralmente envolvendo informações sigilosas.

Durante a realização das suas atividades, a discente viu a necessidade de abrir chamados para outros times. Boa parte dos chamados abertos correspondiam a solicitações de verificação de algum módulo em ambiente de homologação ou produção, quando era percebida alguma irregularidade no seu funcionamento. Também era solicitada a verificação do ambiente Kafka, que impactava diretamente no processamento de arquivos cadastrais.

¹¹ Link para ferramenta: <https://www.zendesk.com.br>

Por outro lado, eram abertos chamados solicitando *logs* ou a verificação de dados de algum banco de dados da aplicação, a fim de reunir evidências e solucionar problemas. Importante ressaltar que a abertura de vários chamados para outros times era realizada com o intuito de auxiliar na resolução de chamados abertos para a equipe da estagiária.

Em relação aos chamados abertos para a equipe, inicialmente ocorria a divisão de tarefas entre os membros. A discente atuou na resolução de vários chamados, com o apoio de outros times e membros da equipe. Alguns chamados atendidos correspondiam à validação de dados cadastrais de determinados emissores, nesses casos, eram realizadas consultas SQL para coleta dos dados, ou era solicitada à equipe de banco de dados a realização das consultas, caso a discente não possuísse acesso à base em questão, por motivos de segurança.

Também eram recebidos chamados de solicitação de evidências de transações de algum emissor, de acionamento de bloqueio/desbloqueio de contas ou cartões, ou ainda evidências de envio de dados cadastrais de determinado cliente. Em boa parte desses casos, além das consultas SQL, era realizada a coleta de *logs* por meio de pesquisas na ferramenta Splunk.

Ainda na resolução de chamados, devido à instabilidades no módulo responsável pela geração de arquivos cadastrais, uma boa quantidade de arquivos foi gerada com erros em suas linhas e, por consequência, não foram enviados corretamente, ficando “represados” no sistema. Os arquivos represados foram enviados novamente para o time, e foi solicitada a correção dos mesmos, para que pudessem ser reenviados.

A discente, com apoio de um desenvolvedor sênior, foi responsável por atender boa parte dos chamados, realizando a correção manual dos arquivos e testes de reenvio para o sistema externo. Por motivos de segurança da empresa, não é possível repassar informações detalhadas sobre o conteúdo dos arquivos, nem sobre o processo de correção e testes com o sistema externo.

Em alguns casos, os chamados abertos indicavam alguma instabilidade ou mau funcionamento de um determinado módulo do sistema. Dependendo da criticidade do problema evidenciado, era necessário solicitar a verificação do ambiente ou revisar o código do projeto, em busca de corrigir erros no sistema. A

necessidade de correção de semáforos, descrita no tópico de correção de *bugs*, por exemplo, surgiu a partir da abertura de chamados indicando o problema.

Com o surgimento da figura do “bombeiro”, desenvolvedor responsável unicamente pela resolução de chamados durante o decorrer da *sprint*, a divisão de atividades foi melhor organizada entre os membros. A discente teve a oportunidade de atuar como bombeira por duas *sprints*, atuando principalmente no atendimento de chamados de *setup* de emissores no sistema de antifraude. O processo de *setup* era realizado de forma manual e necessitava do apoio de outros times para sua resolução. Por motivos de segurança da empresa, informações detalhadas sobre o passo-a-passo do *setup* não podem ser repassadas nesse documento.

Importante ressaltar que o processo de *setup* não era inteiramente conhecido pela equipe e, por isso, não se tinha conhecimento das dependências existentes com outras equipes, nem do tempo necessário para resolução dos chamados. Por esse motivo, foi solicitado à discente o desenvolvimento de uma documentação detalhada sobre o fluxo de configuração dos emissores. O desenvolvimento será abordado no tópico de documentação de processos.

Durante a resolução dos chamados, ocorreram inúmeros momentos de comunicação e entrosamento com membros de outros times, o que possibilitou à discente o desenvolvimento de relações interpessoais em sua vida profissional. Dessa forma, a atividade estimulou o trabalho em equipe e as dinâmicas de interação entre membros de equipes diferentes.

3.2.4 Desenvolvimento de código

Uma das principais demandas da equipe de antifraude era o desenvolvimento de código. O time era responsável por uma aplicação em Java denominada Jarvis, que realizava a integração dos serviços da empresa Dock com serviços externos de verificação de fraudes nas transações. O código da aplicação era separado em módulos, sendo cada um responsável por integrar um serviço diferente. No decorrer do estágio, foi solicitado à discente o desenvolvimento de novos módulos, em conjunto com outros desenvolvedores da equipe.

Após a realização do curso preparatório em Java, foram realizadas reuniões de *bootcamp* com participação de toda equipe, para explicações referentes ao funcionamento do fluxo transacional e informações a respeito da aplicação Jarvis. O conhecimento adquirido com as reuniões foi de suma importância para a realização das atividades de desenvolvimento de código. Entretanto, por motivos de políticas de segurança da empresa, as atividades citadas não possuem informações detalhadas.

A primeira atividade repassada para desenvolvimento pela estagiária foi um módulo responsável pelo envio de dados cadastrais de clientes, utilizando comunicação via *socket*. De acordo com Oliveira (2022), *sockets* são um processo de comunicação que permite a troca de informações entre dois processos diferentes, funcionando a partir da arquitetura cliente-servidor. O módulo recebia os dados de outro módulo do Jarvis e os enviava por meio do *socket* para um serviço externo. Por se tratar de um módulo complexo, a discente teve apoio do gerente de tecnologia da equipe durante o desenvolvimento.

A segunda atividade foi realizada em todos os módulos da aplicação Jarvis, e se trata da centralização das chamadas ao banco de dados em um único módulo. Anteriormente, cada módulo era responsável por se comunicar com o banco individualmente. Entretanto, após a criação do módulo centralizador do acesso ao banco por outro desenvolvedor da equipe, foi solicitado à estagiária, em conjunto com dois desenvolvedores, a refatoração de todos os módulos da aplicação, removendo as chamadas diretas ao banco e passando a fazer requisições para o novo módulo implementado.

O módulo de dados foi desenvolvido utilizando programação reativa, com o módulo Spring WebFlux do *framework* Spring, com o objetivo de melhorar a *performance* e capacidade de processamento do sistema. Programação reativa é um paradigma orientado a fluxo de dados assíncronos, ou seja, as operações realizadas não precisam seguir uma sequência previamente definida, sendo independentes umas das outras. (GUEDES, 2021).

A terceira atividade foi realizada também em todos os módulos da aplicação Jarvis, tratando-se da implementação de aspecto para captura de *logs* dos módulos. A discente, em conjunto com outro desenvolvedor estagiário, foi responsável pela

refatoração dos *logs* dos módulos existentes, buscando padronizar e remover duplicações nos mesmos. Guedes (2020) afirma que a programação orientada a aspectos é uma abordagem que possibilita a separação de componentes funcionais do sistema de forma concisa, usando mecanismos de abstração e composição.

Em resumo, a atividade consistiu em refatorar o código dos módulos, substituindo a lógica anterior de criação dos *logs* pela implementação com Spring AOP. Com isso, conforme a aplicação realizava seu fluxo padrão de funcionamento, o aspecto iniciava a criação do *log*, para que o mesmo fosse enviado para a plataforma do Splunk.

A quarta atividade se refere a um módulo separado do sistema principal, desenvolvido para testes de funcionamento do módulo de leitura, geração e envio de arquivos cadastrais do sistema Jarvis. Com o aumento de chamados evidenciando instabilidades do módulo de arquivos cadastrais, foi necessário investigar o fluxo de funcionamento do mesmo, a fim de realizar as devidas correções.

Para auxiliar no processo, foi desenvolvido um módulo responsável por receber arquivos com erros gerados pelo módulo do Jarvis, ler os campos existentes, organizar os dados e enfileira-los na plataforma Kafka, para que posteriormente fossem recebidos novamente pelo módulo principal do sistema. O objetivo era evidenciar os erros dos arquivos, a fim de corrigir os *bugs* encontrados. O Kafka é uma plataforma de *streaming* de eventos distribuídos, disponibilizada em código aberto (KAFKA, 2023). Com o desenvolvimento do módulo e realização de testes de envio, foi possível verificar os erros dos arquivos gerados pelo sistema e realizar as devidas correções e refatorações de código.

Outras atividades de desenvolvimento de código foram realizadas para correções pontuais de *bugs* encontrados, como evidenciado no tópico seguinte, ou melhorias de código, removendo duplicações existentes e tornando o código mais organizado e funcional.

A atividade proporcionou um amplo desenvolvimento da estagiária, visto que foram aplicados conceitos e novas tecnologias que eram anteriormente desconhecidos por ela. O conhecimento agregado por meio do estudo e aplicação das tecnologias foi de suma importância para sua experiência profissional. Vale ressaltar a contribuição das disciplinas de Programação II e DAC, que trabalharam o

desenvolvimento de código em linguagem Java e a utilização do *framework* Spring, tecnologias que foram extremamente necessárias para realização das atividades solicitadas.

3.2.5 Correção de *bug*

A atividade trata da investigação e correção de *bugs* encontrados em ambientes de homologação e produção da aplicação. *Bug* é um termo da informática que se refere a falhas inesperadas que ocorrem durante a execução de algum *software*, podendo colocar em risco o sistema de segurança da máquina (STIVANI, 2019). Vale ressaltar que a atividade era realizada pela discente em acompanhamento de outros desenvolvedores da equipe, visto que eram problemas que necessitavam de rápida correção.

Com o surgimento de chamados relatando problemas nos envios de arquivos cadastrais, foi levantada a necessidade de revisão do código de um dos módulos da aplicação, responsável pela leitura, geração e envio dos arquivos. Após a realização de testes de fluxo e revisão do código implementado, foi identificado que o problema estava na lógica de criação de semáforos, que eram utilizados pela aplicação para notificar o sistema externo da existência dos arquivos para leitura. Dessa forma, foi realizada a alteração do código e, após a realização de testes para confirmar o funcionamento correto do sistema, o código foi submetido para produção como *hotfix* (correção rápida).

Além disso, com o desenvolvimento de um novo módulo para aplicação, responsável por receber requisições via API REST e direcioná-las para um serviço externo, foi identificado nos testes que o módulo não estava funcionando como esperado, em ambiente de homologação. Segundo Souza (2020), API REST se refere a um conjunto de requisições que permite o envio de dados entre aplicações, com os padrões definidos pelo estilo de arquitetura Rest.

A estagiária, em conjunto com outro desenvolvedor estagiário, realizou a revisão do código implementado. Após a revisão, foi verificado que o problema estava na lógica de chamada do *endpoint* do sistema externo, pois estava sendo utilizada a variável de ambiente errada, resultando em erros durante o envio das

requisições. Corrigido o erro, o código foi submetido a um *code review* e, por fim, subiu para homologação, a fim de que fossem realizados novos testes.

A atividade foi extremamente importante para a vivência profissional da estagiária, uma vez que desenvolveu sua capacidade de análise de código e raciocínio lógico, além de alertar sobre a necessidade de revisar e testar corretamente as lógicas de código implementadas, antes de realizar qualquer subida para ambientes de homologação ou produção.

3.2.6 Testes de *software*

Ao realizar atividades de desenvolvimento de código ou para verificar o funcionamento do código implementado de algum módulo da aplicação, era imprescindível a realização de testes no sistema. A discente realizou e configurou testes unitários, testes de integração e testes de sistema.

No desenvolvimento do módulo responsável pelo envio de dados cadastrais via *socket*, foram criados testes unitários. Os testes unitários validam o comportamento de partes pequenas e isoladas do código, como funções e métodos, trata-se de testes de baixo custo e rápida execução (RIBEIRO, 2023).

Nas atividades que envolviam a refatoração dos módulos, foi realizada também a refatoração dos testes unitários e testes de integração existentes, para utilizarem a nova lógica implementada. Os testes de integração são realizados após os testes unitários, sendo realizados para validar o funcionamento das unidades de código em conjunto, a fim de garantir a funcionalidade total do sistema (KRIGER, 2021).

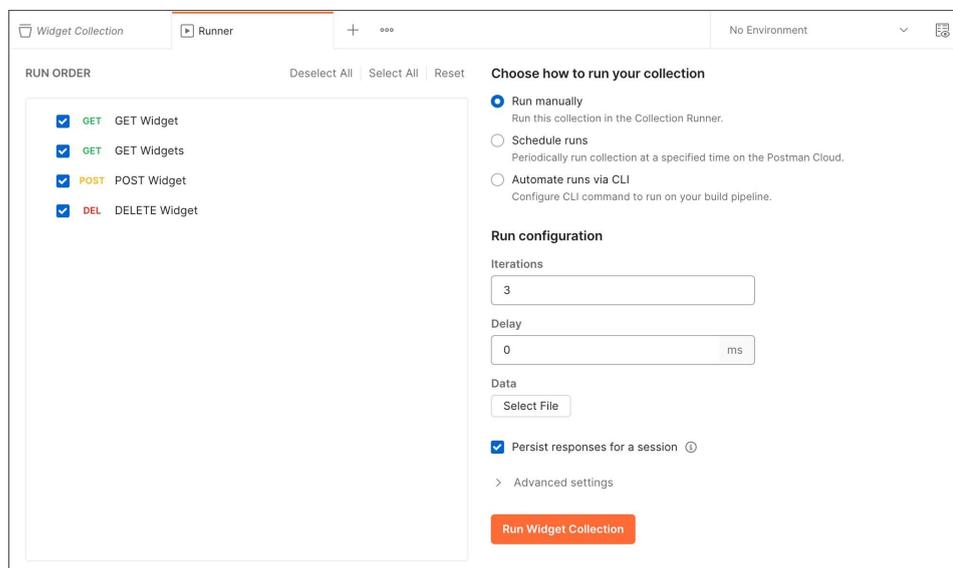
Os testes foram desenvolvidos e executados utilizando as tecnologias JUnit e Mockito. JUnit é um *framework open-source* para criação de testes automatizados na linguagem Java que funciona a partir de anotações (AZEVEDO, 2018). O Mockito, por sua vez, é um *framework* de testes cujo principal objetivo é controlar o comportamento dos métodos, simulando o funcionamento das classes (CANDIOLLI, 2020). Importante ressaltar que em todas as atividades que envolviam alterações no código do sistema, era necessário executar os testes existentes, a fim de validar se a aplicação estava funcionando corretamente.

Outra atividade realizada durante o estágio foram os testes de sistema de uma funcionalidade específica da aplicação Jarvis, com o objetivo de verificar se o sistema se comportava como esperado, na ausência de comunicação com um sistema externo, e se suportava uma determinada quantidade de requisições simultâneas. Por motivos de segurança da empresa, não serão expostos mais detalhes sobre a funcionalidade em questão.

Antunes (2021) afirma que os testes de sistema focam no comportamento e capacidade de todo um sistema, simulando a experiência do usuário ao utilizar a aplicação. Por esse motivo, os testes de sistema são extremamente importantes e agregam muito valor à qualidade do *software*.

O envio das requisições foi efetuado através da ferramenta Runner da plataforma Postman, que fornece um ambiente simplificado para construção e utilização de APIs. O Runner permite a execução de requisições em uma sequência específica, com quantidade de iterações e tempo de *delay* configuráveis (POSTMAN, 2023). A interface da ferramenta pode ser observada na Figura 9.

Figura 9 - Interface da ferramenta Postman Runner



Fonte: Postman

Os testes foram realizados em ambiente de homologação e foram divididos em duas etapas: na primeira, era efetuada uma determinada quantidade de requisições com o sistema rodando normalmente, em seguida, era simulada a

desconexão do sistema externo e era enviada a mesma quantidade de requisições. Em ambas as etapas, o comportamento do sistema era verificado por meio dos *logs* da aplicação e as informações dos documentos salvos no banco de dados, através da ferramenta Studio 3T¹².

Nas primeiras execuções dos testes, foi identificada a necessidade de pequenas correções na lógica existente e em variáveis de ambiente da aplicação. Após as correções, foram realizadas novas sessões de testes. Por fim, a funcionalidade foi validada corretamente e liberada para uso.

A execução das atividades descritas reforçou a importância dos testes no processo de desenvolvimento de *software*, especialmente em uma aplicação corporativa de grande porte. A estagiária pôde observar na prática como a construção de testes contribui para evidenciar a eficiência do sistema e garantir sua qualidade e segurança, além de facilitar na busca por possíveis *bugs* e erros na implementação. A vivência em sala de aula com os *frameworks* JUnit e Mockito e os conceitos aprendidos na disciplina de Testes contribuíram ricamente para o desempenho da estagiária na atividade.

3.2.7 Code review

Ao finalizar alguma atividade de desenvolvimento de código, era realizada a revisão do código implementado, a fim de verificar se a solução desenvolvida atendia a todos os requisitos propostos, ou se a correção de algum *bug* existente na aplicação resolvia o problema. Durante a maior parte do estágio, essa revisão era realizada exclusivamente pelo gerente de tecnologia, que analisava o código e apontava as correções necessárias ou sugestões de melhoria. As mudanças solicitadas poderiam ser desde simples alterações nas informações disponibilizadas nos *logs* do sistema, até refatorações mais complexas.

Terminada a implementação do código, a discente realizava a abertura de *merge requests* e os enviava para o gerente de tecnologia iniciar o *code review*. Um *merge* ou *pull request* é um mecanismo presente nas ferramentas de versionamento

¹² Link para ferramenta: <https://studio3t.com>

de código, como Github¹³ e Gitlab¹⁴, criado com o intuito de sinalizar a outras pessoas que um novo trecho de código foi desenvolvido ou modificado, e que o desenvolvedor responsável deseja incluir as alterações nas ramificações principais do repositório do projeto (SOUZA, 2021).

Em alguns casos, era realizada uma reunião para apresentação do código desenvolvido, enquanto o gestor discorria sugestões de melhorias ou solicitava mudanças mais extremas. A vídeo chamada poderia ocorrer de forma individual, contando apenas com a presença da discente e do gerente de tecnologia, ou poderia acontecer em grupo, com a presença de outros desenvolvedores, sejam eles envolvidos ou não no processo de desenvolvimento.

Após mudanças na equipe de desenvolvimento, o processo de *code review* se tornou mais dinâmico. Após a abertura de *merge requests* por algum integrante da equipe, toda a equipe era encorajada a revisar o código implementado, dando suas contribuições por meio de comentários. Para aprovação do *merge request* e subida do código, era necessária a aprovação de pelo menos dois desenvolvedores.

A discente teve a oportunidade tanto de revisar códigos de colegas de equipe, contribuindo com comentários, quanto de receber *feedbacks* de outros membros, com apontamentos importantes de melhoria e correção de código. A maior participação dos integrantes na revisão do código possibilitou que o processo se tornasse mais rápido, além de permitir uma melhor compreensão geral dos códigos implementados.

Essa atividade foi essencial para o aprendizado da discente, encorajando o trabalho em equipe, uma vez que a validação de código não era realizada individualmente, mas sim com a colaboração de outros desenvolvedores. A discente também pôde obter um melhor conhecimento técnico e aplicá-lo de forma a melhorar a qualidade do código desenvolvido, graças às contribuições de colegas de equipe mais experientes.

¹³ Link para ferramenta: <https://github.com>

¹⁴ Link para ferramenta: <https://about.gitlab.com>

3.2.8 Queries em bancos de dados

No desenvolvimento *backend* de um *software*, a forma como os dados devem ser manipulados e armazenados é extremamente importante, para garantir o bom desempenho da aplicação. Considerando um sistema corporativo complexo, que lida com grandes volumes de dados de diferentes usuários, uma boa comunicação com o banco de dados torna-se um ponto crítico para seu funcionamento.

Um banco de dados é um conjunto de dados relacionados entre si, gerenciados por um SGBD, que é um conjunto de programas responsáveis por acessar esses dados. Esses dados podem ser armazenados seguindo diferentes modelos, um dos mais utilizados é o modelo relacional, caracterizado pelo uso de coleção de tabelas para representar os dados e as relações existentes entre eles (SILBERSCHATZ; KORTH; SUDARSHAN, 2006). A execução de comandos em bancos de dados relacionais ocorre por meio de *queries* em SQL. De acordo com Kriger (2022), SQL é a linguagem padrão de comunicação com banco de dados.

O time possuía acesso a três bancos de dados relacionais, acessando-os através da ferramenta DBeaver, um sistema multi plataforma *open-source* que auxilia no gerenciamento de diversos bancos de dados populares (DBEAVER COMMUNITY, 2010?). A interface da plataforma está evidenciada na Figura 10.

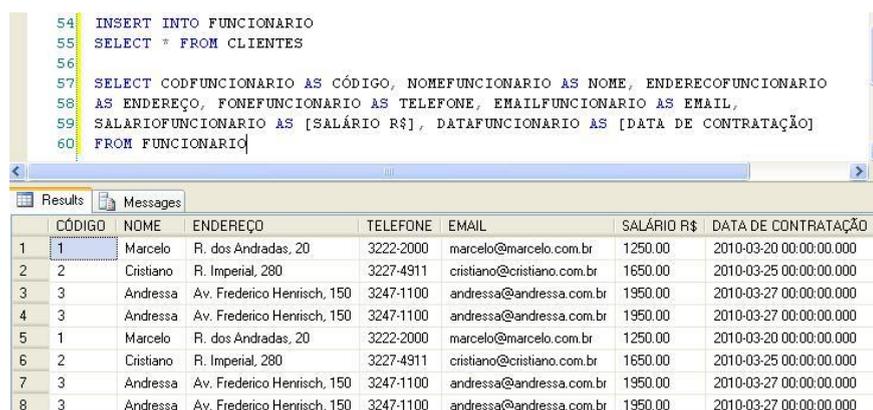
Figura 10 - Interface da plataforma DBeaver

#	Address	City	State	Country	Value
13	Qe 7 Bloco G	Brasilia	DF	Brazil	71
14	8210 111 ST NW	Edmonton	AB	Canada	T6
15	700 W Pender Street	Vancouver	BC	Canada	V6
16	1600 Amphitheatre Parkway	Mountain View	CA	USA	94
17	1 Microsoft Way	Redmond	WA	USA	98
18	627 Broadway	New York	NY	USA	10
19	1 Infinite Loop	Cupertino	CA	USA	98
20	541 Del Medio Avenue	Mountain View	CA	USA	94
21	801 W 4th Street	Reno	NV	USA	89
22	120 S Orange Ave	Orlando	FL	USA	32
23	Tauentzienstraße 6	Berlin		Germany	10
24	69 Salem Street	Boston	MA	USA	21
25	182 E Superior Street	Chicago	IL	USA	60
26	319 N. Frances Street	Madison	WI	USA	53
27	2211 W Berry Street	Fort Worth	TX	USA	76
28	1033 N Park Ave	Tucson	AZ	USA	85
29	302 S 700 E	Salt Lake City	UT	USA	84
30	796 Dundas Street West	Toronto	ON	Canada	M6
31	230 Elgin Street	Ottawa	ON	Canada	K2
32	194A Chain Lake Drive	Halifax	NS	Canada	B3
33	696 Osborne Street	Winnipeg	MB	Canada	R3
34	5112 48 Street	Yellowknife	NT	Canada	X1

Fonte: Dbeaver Community

A utilização de consultas SQL era solicitada essencialmente para resolução de chamados internos, buscando evidenciar dados específicos de contas de clientes, ou para validar o recebimento de transações de determinado emissor. Na Figura 11 está um exemplo de consulta SQL.

Figura 11 - Exemplo de consulta SQL



```

54 INSERT INTO FUNCIONARIO
55 SELECT * FROM CLIENTES
56
57 SELECT CODFUNCIONARIO AS CÓDIGO, NOMEFUNCIONARIO AS NOME, ENDERECOFUNCIONARIO
58 AS ENDEREÇO, FONEFUNCIONARIO AS TELEFONE, EMAILFUNCIONARIO AS EMAIL,
59 SALARIOFUNCIONARIO AS [SALÁRIO R$], DATAFUNCIONARIO AS [DATA DE CONTRATAÇÃO]
60 FROM FUNCIONARIO

```

	CÓDIGO	NOME	ENDEREÇO	TELEFONE	EMAIL	SALÁRIO R\$	DATA DE CONTRATAÇÃO
1	1	Marcelo	R. dos Andradas, 20	3222-2000	marcelo@marcelo.com.br	1250.00	2010-03-20 00:00:00.000
2	2	Cristiano	R. Imperial, 280	3227-4911	cristiano@cristiano.com.br	1650.00	2010-03-25 00:00:00.000
3	3	Andressa	Av. Frederico Henrich, 150	3247-1100	andressa@andressa.com.br	1950.00	2010-03-27 00:00:00.000
4	3	Andressa	Av. Frederico Henrich, 150	3247-1100	andressa@andressa.com.br	1950.00	2010-03-27 00:00:00.000
5	1	Marcelo	R. dos Andradas, 20	3222-2000	marcelo@marcelo.com.br	1250.00	2010-03-20 00:00:00.000
6	2	Cristiano	R. Imperial, 280	3227-4911	cristiano@cristiano.com.br	1650.00	2010-03-25 00:00:00.000
7	3	Andressa	Av. Frederico Henrich, 150	3247-1100	andressa@andressa.com.br	1950.00	2010-03-27 00:00:00.000
8	3	Andressa	Av. Frederico Henrich, 150	3247-1100	andressa@andressa.com.br	1950.00	2010-03-27 00:00:00.000

Fonte: DevMedia

Entretanto, devido à constantes acionamentos relacionados à ausência de dados cadastrais de clientes de diferentes emissores, foi solicitado à discente que fosse realizado um levantamento da quantidade de dados cadastrais presente em todas as bases dos clientes, a fim de comparar com a quantidade de dados salvos em outras bases de dados e verificar a discrepância dos valores.

Como se tratavam de muitas bases e grandes quantidades de informações a serem consultadas, foi desenvolvido pela discente, com auxílio do gestor, um *script* em Python, utilizando a biblioteca *pymssql*, para automatizar as consultas. O *script* recebia como parâmetro os nomes dos servidores dos bancos de dados, uma lista de nomes das bases e as consultas SQL a serem performadas. Com essas informações, o *script* acessava os servidores remotamente e executava as consultas, exibindo o resultado das *queries* no terminal. Finalizado o levantamento, a discente abriu um chamado para a equipe de banco de dados, solicitando que fossem realizadas as mesmas consultas em outras bases de dados.

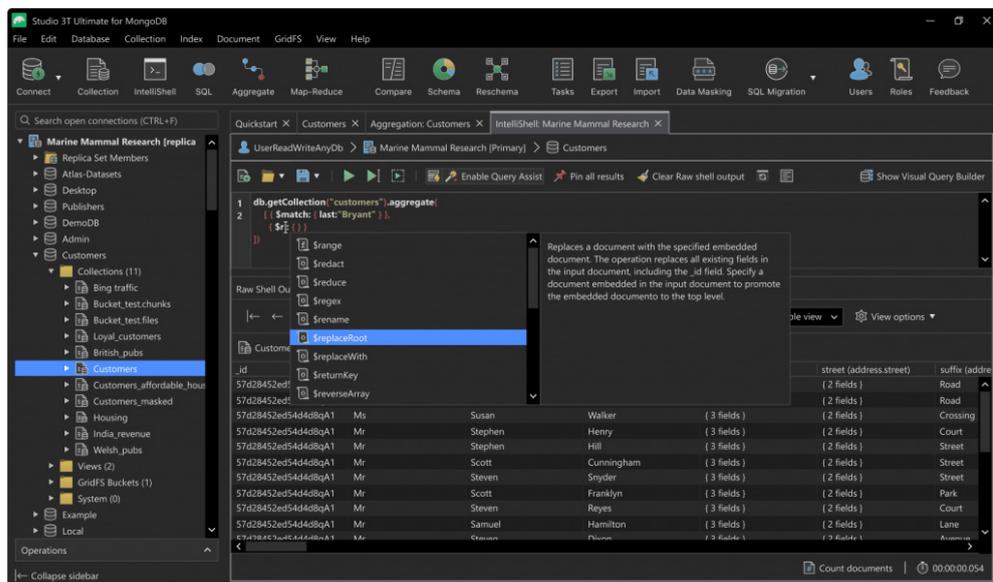
Além dos bancos relacionais, existem ainda os bancos NoSQL, ou não relacionais. Sadalage e Fowler (2013) apontam que as características que esses

bancos possuem em comum são a ausência de esquema, não utilizam o modelo relacional, são *open-source* e possuem uma boa execução em *clusters*.

Bancos de dados de documentos são um exemplo de banco NoSQL, os documentos armazenados são estruturas de dados em forma de árvores hierárquicas e autodescritivas, podendo ser no formato XML, JSON, entre outros. Outra característica importante é que, apesar dos documentos serem semelhantes entre si, eles não necessariamente precisam ter a mesma estrutura. Um banco de dados de documentos amplamente utilizado atualmente é o MongoDB (SADALAGE; FOWLER, 2013).

O time possuía acesso a um banco de dados não relacional da aplicação, acessando-o por meio da plataforma Studio 3T, uma interface gráfica exclusiva para o MongoDB, permitindo a visualização e manipulação dos documentos persistidos (STUDIO 3T, 2023). A interface da plataforma pode ser observada na Figura 12.

Figura 12 - Interface da plataforma Studio 3T



Fonte: Studio 3T

Em atividades de realização de testes e correção de *bugs*, foram realizadas consultas dos documentos no MongoDB, com o objetivo de validar o funcionamento do que foi desenvolvido. Na atividade de *setup* de emissores, foram realizadas *queries* de consulta, inserção, atualização, e deleção de documentos, a fim de fazer os ajustes necessários e coletar evidências para finalização dos chamados.

Com a demanda de migração das chamadas ao banco para um módulo específico, além da refatoração dos outros módulos do sistema, também foi solicitado à discente, junto com outro estagiário da equipe, o ajuste nos documentos salvos no banco, de forma a seguirem o novo modelo de documento mapeado.

Para otimizar o tempo de realização das *queries* de *update*, por se tratar de uma grande quantidade de documentos, foi utilizado o método `updateMany`, que permite a atualização de todos os documentos que atenderem a um critério específico (MONGODB, 2023).

A atividade enriqueceu os conhecimentos da discente vistos em sala de aula, nas disciplinas de Banco de Dados I e II, em relação às consultas SQL. Além de possibilitar a aplicação do aprendizado já obtido, foi possível trabalhar com entidades mais complexas e manipular grandes quantidades de dados. Ademais, a realização de consultas em bancos de dados de documentos foi essencial para consolidar os conhecimentos teóricos adquiridos, uma vez que a prática não foi muito trabalhada na disciplina.

3.2.9 Desenvolvimento de diagramas

No processo de desenvolvimento de *software*, é frequente a utilização de diagramas, para facilitar a definição e compreensão do escopo do projeto. Para criação desses diagramas, existe a linguagem UML. Segundo Guedes (2011), a UML é uma linguagem de modelagem de *software* que tem como objetivo auxiliar os engenheiros de *software* a estabelecer os aspectos do sistema, como seus requisitos, sua estrutura lógica, seu comportamento ou a dinâmica de seus processos, antes do sistema ser desenvolvido propriamente.

Em vista disso, com a criação e integração de um novo módulo ao sistema de antifraude, foi necessária a criação de novos diagramas de classe e de sequência. A atividade foi realizada pela discente, em conjunto com outros desenvolvedores da equipe.

O `draw.io`¹⁵ foi a ferramenta utilizada para modelagem dos diagramas solicitados. Trata-se de um editor gráfico *online* e gratuito, que fornece recursos para

¹⁵ Link para ferramenta: <https://www.drawio.com>

criação de diagramas e fluxogramas, possuindo uma interface simples e de fácil uso (TECHTUDO, 2023).

O diagrama de classes serve para determinar a estrutura das classes do sistema, assim como a definição de como se relacionam entre si e como é realizada a troca de informações. Esse diagrama serve de base para os demais diagramas UML. O diagrama de sequências, por sua vez, define a ordem temporal em que ocorre a comunicação entre objetos envolvidos em um determinado processo do sistema. (GUEDES, 2011). Por motivos de política de segurança da empresa, não é possível repassar informações detalhadas em relação à criação dos diagramas nesse documento

A atividade, mesmo não sendo frequentemente realizada durante o período de estágio, proporcionou à discente a possibilidade de colocar em prática os conhecimentos adquiridos em disciplinas vistas no curso, como Análise e Projeto e Projeto I. A criação dos diagramas também facilitou a compreensão dos desenvolvedores da equipe e de outros times em relação ao fluxo da nova aplicação.

3.2.10 Desenvolvimento de documentação de processos

Independente da área de atuação, boa parte do trabalho de uma empresa pode ser resumido e organizado em processos recorrentes. Por esse motivo, torna-se necessário documentar esses processos, buscando agilizar a execução do trabalho e facilitar o entendimento geral. A documentação interna é a prática de registrar e manter processos de forma clara e precisa, para que possam servir de referência para membros internos da equipe (ATLASSIAN, 2023).

A criação de documentações de processos traz diversos benefícios para a produtividade da organização, produzindo maior eficiência, melhoria nos processos e diminuição nas lacunas de conhecimento da equipe, além de permitir o treinamento mais rápido e eficaz de novos membros do time (ATLASSIAN, 2023).

Com o acionamento constante do time por meio de chamados relacionados ao *setup* de emissores na aplicação, e por se tratar de um processo do qual poucos membros da equipe possuíam conhecimento, surgiu a necessidade de organizar

uma documentação do processo de *setup*. A discente, por estar atuando diretamente com os chamados, ficou responsável pelo desenvolvimento. O acesso à documentação foi disponibilizado por meio da plataforma Confluence¹⁶. O Confluence é um espaço de trabalho para ambiente remoto que permite a criação e organização de informação de projetos, com o intuito de facilitar a colaboração entre os times de uma empresa (ATLASSIAN, 2023).

Após compreender o fluxo e conforme ia atendendo aos chamados abertos, a discente organizou a documentação detalhando as informações necessárias para realizar o *setup*, o passo-a-passo do processo e as equipes envolvidas, adicionando ilustrações da execução, para facilitar o entendimento e observações pontuais. Por fim, foram disponibilizadas duas versões da documentação: uma mais técnica e detalhada para a equipe de desenvolvimento, e outra mais resumida e focada para outras equipes. Vale ressaltar que, à medida que surgiam novos detalhes importantes ou novos tipos de *setup* de emissor, a documentação era revisada e atualizada de acordo pela discente.

Com a realização da atividade, a discente pode perceber como a documentação é essencial para agilizar os processos da empresa, além da importância do compartilhamento de informações entre a equipe, evitando centralizar o conhecimento em apenas alguns membros. Posteriormente, a documentação também serviu para que a equipe pudesse estudar e discutir maneiras de automatizar o fluxo de *setup*.

¹⁶ Link para ferramenta: <https://www.atlassian.com/br/software/confluence>

4 CONCLUSÃO

Com o período de estágio na empresa Dock, a discente obteve muita aprendizagem prática e teve a oportunidade de aplicar os conhecimentos obtidos em sala de aula, o que destaca a importância da realização de um estágio supervisionado em uma empresa focada em desenvolvimento de *software*. Como evidenciado no documento, os conceitos aprendidos no curso de ADS foram essenciais na execução das atividades do estágio, contribuindo grandemente para o desenvolvimento profissional e pessoal da discente.

As atividades executadas foram fundamentais para desenvolver a discente de forma técnica, com a aplicação e utilização de novos conceitos e tecnologias, e social, por meio da vivência em equipe e interações com outros times. Sendo este o principal ponto positivo do estágio.

Entretanto, durante o período de estágio, foram evidenciados pontos a melhorar. Uma das principais adversidades enfrentadas foi a falta de suporte e direcionamento de pessoas mais qualificadas da equipe em determinadas atividades, especialmente na compreensão de regras de negócio, necessárias para execução das tarefas, e na aplicação de novas tecnologias no desenvolvimento do sistema, que não eram do domínio da discente.

Outro ponto a melhorar foi a quantidade demasiada de atividades solicitadas de forma paralela, dificultando a conclusão organizada e estruturada das mesmas. Além disso, em razão da pressão no ambiente de trabalho, ocorreram alguns problemas gerenciais entre os membros da equipe. Por outro lado, vale destacar que, com as mudanças na equipe que ocorreram durante o período de estágio, boa parte das dificuldades encontradas foi amenizada.

Apesar das adversidades ocorridas, o estágio foi fundamental para o crescimento profissional da discente. Graças à experiência obtida no ambiente de estágio e aprendizagem adquirida no curso de ADS, a discente pôde ingressar no mercado de trabalho, através da efetivação na empresa Dock, onde segue trabalhando atualmente.

Portanto, nesse documento foram apresentadas as atividades realizadas pela discente, exibindo as características essenciais para a execução das mesmas. Entre

as atividades descritas, destacam-se o desenvolvimento de novas funcionalidades para o sistema de antifraude, correção de *bugs*, criação de testes, diagramas e documentações relacionados ao sistema, além do suporte aos clientes por meio do atendimento de chamados. Com isso, foram atingidos os objetivos do estágio pré-determinados neste relatório.

REFERÊNCIAS

- AIQON. **Princípios de Monitoramento de Eventos e de Logs de Auditoria**. [2020?]. Disponível em: <https://aiqon.com.br/blog/principios-de-monitoramento-de-eventos-e-de-logs-de-auditoria/>. Acesso em: 25 abr. 2023.
- ALURA. **Meus Cursos**. Abr. 2023. Disponível em: <https://cursos.alura.com.br/courses/mine>. Acesso em: 22 abr. 2023.
- ANTUNES, F. Diferença entre fases de teste, tipos de teste e formas de execução. **Zup**, 01 abr. 2021. Disponível em: <https://www.zup.com.br/blog/tipos-de-teste>. Acesso em: 13 jun. 2023.
- ATLASSIAN. **Como dominar a documentação interna**: Tire informações da cabeça da equipe e guarde em um local seguro. [2023?]. Disponível em: <https://www.atlassian.com/br/work-management/knowledge-sharing/documentation>. Acesso em: 05 maio 2023.
- _____. **Confluence**. [2023?]. Disponível em: <https://www.atlassian.com/br/software/confluence>. Acesso em: 06 maio 2023.
- _____. **O guia definitivo para documentação de processo**: Melhore a produtividade capturando seus fluxos de trabalho. [2023?]. Disponível em: <https://www.atlassian.com/br/work-management/knowledge-sharing/documentation/process-documentation>. Acesso em: 05 maio 2023.
- _____. **Painéis do jira scrum**. [2023?]. Disponível em: <https://www.atlassian.com/br/software/jira/features/scrum-boards>. Acesso em: 22 abr. 2023.
- AZEVEDO, M. Qualidade e JUnit: introduzindo automatização de testes unitários do seu software Java no dia-a-dia. **Medium**, 23 out. 2018. Disponível em: <https://mari-azevedo.medium.com/qualidade-e-junit-introduzindo-automatizacao-de-testes-unitarios-do-seu-software-java-no-dia-a-dia-849611de5574>. Acesso em: 13 jun. 2023.
- BRAGA, M. Como o estágio prepara você para a vida profissional? **Reachr**, 14 set. 2021. Disponível em: <https://blog.reachr.com.br/como-o-estagio-prepara-voce-para-a-vida-profissional/>. Acesso em: 14 jun. 2023.
- BRASIL, N. P. O que é VPN: entenda o que é e quais são as funcionalidades. **Olhar Digital**. Disponível em: <https://olhardigital.com.br/2022/07/05/tira-duvidas/o-que-e-vpn/>. Acesso em: 01 jun. 2023.

BRASSCOM. Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025. 01 dez. 2021. Disponível em: <https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>. Acesso em: 14 jun. 2023.

BRASILEIRO, R. Planning Poker: A melhor maneira de estimar qualquer atividade. **Métodoágil**, 17 maio 2017. Disponível em: <https://www.metodoagil.com/planning-poker/>. Acesso em: 20 abr. 2023.

CANDIOLLI, S. Testando seu código Java com o Mockito framework. **Medium**, 12 nov. 2020. Disponível em: <https://medium.com/cwi-software/testando-seu-código-java-com-o-mockito-framework-k-8bea7287460a>. Acesso em: 13 jun. 2023.

DBEAVER COMMUNITY. **Dbeaver community | free universal database tool.** [2010?]. Disponível em: <https://dbeaver.io>. Acesso em: 29 abr. 2023.

DEVMEDIA. **SQL Server - Comandos DML - Select.** Disponível em: <https://www.devmedia.com.br/sql-server-2005-comandos-dml-select/17054/>. Acesso em: 29 abr. 2023.

DOCK. **Home - Dock.** 2023. Disponível em: <https://dock.tech/>. Acesso em: 27 abr. 2023.

DOCK FLUID. **Conductor unifica todas as suas ofertas sob uma nova identidade para a marca Dock.** 27 ago. 2021. Disponível em: https://dock.tech/blog/nova_dock/. Acesso em: 27 abr. 2023.

EGOV. **Metodologias Ágeis/Scrum/Lean.** 13 set. 2021. Disponível em: <https://egov.df.gov.br/metodologias-ageis-scrum-lean/>. Acesso em: 20 abr. 2023.

FILGUEIRAS, F. JAVA - Principais características. **Tableless**, 05 maio 2015. Disponível em: <https://tableless.com.br/java-principais-caracteristicas/>. Acesso em: 27 abr. 2023.

GLASSDOOR. **Fotos do escritório da DOCK.** Disponível em: <https://www.glassdoor.com.br/Fotos/DOCK-Fotos-escritório-IMG9964085.htm>. Acesso em: 27 abr. 2023.

GRUPO INTEGRADO. **A importância do estágio para a carreira profissional.** 6 jul. 2021. Disponível em: <https://blog.grupointegrado.br/?p=525>. Acesso em: 14 jun. 2023.

GUEDES, G. T. A. **UML 2: Uma abordagem prática.** 2. ed. São Paulo: Novatec Editora, 2011.

GUEDES, M. O que é programação reativa? **TreinaWeb**, abr. 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-programacao-reativa#:~:text=Programa%20reativa%20%20%20um%20modelo,dados%20e%20propaga%20es%20de%20esta dos>. Acesso em: 12 jun. 2023.

_____. O que é AOP - Programação Orientada a Aspectos? **TreinaWeb**, set. 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-aop-programacao-orientada-a-aspectos>. Acesso em: 12 jun. 2023.

HÖRLLE, W. Jira software: tudo o que você precisa saber. **CSP Tech**, 12 jan. 2022. Disponível em: <https://blog.csptecnologia.com/jira-software/>. Acesso em: 22 abr. 2023.

KAFKA. **Apache Kafka**. Disponível em: <https://kafka.apache.org>. Acesso em: 12 jun. 2023.

KRIGER, D. Linguagem SQL: O que é e quais os seus principais comandos. **Kenzie**, 15 jun. 2022. Disponível em: <https://kenzie.com.br/blog/sql/>. Acesso em: 29 abr. 2023.

_____. O que é teste de integração e quais são os tipos de teste? **Kenzie**, 08 out. 2021. Disponível em: <https://kenzie.com.br/blog/teste-de-integracao/>. Acesso em: 12 jun. 2023.

LABS. **Investida da Riverwood Capital e da Visa, Dock compra fintech mexicana Cacao**. 2021. Disponível em: <https://labsnews.com/pt-br/noticias/negocios/dock-compra-cacao/>. Acesso em: 27 abr. 2023.

MONGODB. **Docs Menu | db.collection.updateMany()**. [2023?]. Disponível em: <https://www.mongodb.com/docs/manual/reference/method/db.collection.updateMany/>. Acesso em: 29 abr. 2023.

OLIVEIRA, P. H. Sistema de gerenciamento de logs: Como funciona? **Linux Solutions**, 18 mar. 2019. Disponível em: <https://www.linuxsolutions.com.br/sistema-de-gerenciamento-de-logs-como-funciona/>. Acesso em: 25 abr. 2023.

_____. Sockets: o que é e como eles funcionam? **Linux Solutions**, 31 maio 2022. Disponível em: <https://linuxsolutions.com.br/sockets-o-que-e-e-como-eles-funcionam/>. Acesso em: 12 jun. 2023.

POSTMAN. **Postman API Platform**. 2023. Disponível em: <https://www.postman.com>. Acesso em: 13 jun. 2023.

_____. **Using the Collection Runner**. 13 jun. 2023. Disponível em: <https://learning.postman.com/docs/collections/running-collections/intro-to-collection-runs/>. Acesso em: 13 jun. 2023.

RIBEIRO, M. Como escrever testes unitários que fazem a diferença? **Zup**, 25 abr. 2023. Disponível em: <https://www.zup.com.br/blog/testes-unitarios>. Acesso em: 12 jun. 2023.

SADALAGE, P. J.; FOWLER, M. **NoSQL Essencial**: Um guia conciso para o mundo emergente da persistência poliglota. 1. ed. São Paulo: Novatec, 2013.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2006.

SOUZA, I. Entenda o que é Rest API e a importância dele para o site da sua empresa. **RockContent**, 7 abr. 2020. Disponível em: <https://rockcontent.com/br/blog/rest-api/>. Acesso em: 12 jun. 2023.

SOUZA, I. F. Pull/Merge Requests. **guia.dev**, 8 abr. 2021. Disponível em: <https://guia.dev/pt/pillars/continuous-integration/pull-request.html>. Acesso em: 27 abr. 2023.

SLACK. **O que é o Slack?** [2023?]. Disponível em: <https://slack.com/intl/pt-br/help/articles/115004071768-O-que-é-o-Slack-#:~:text=O%20Slack%20é%20um%20app,como%20as%20organizações%20se%20comunicam>. Acesso em: 27 abr. 2023.

SPLUNK. **About dashboards**. 28 out. 2021. Disponível em: <https://docs.splunk.com/Documentation/Splunk/9.0.4/SearchTutorial/Aboutdashboards>. Acesso em: 25 abr. 2023.

_____. **Log4Shell - Detecting Log4j 2 RCE Using Splunk**. 9 dez. 2021. Disponível em: https://www.splunk.com/en_us/blog/security/log-jammin-log4j-2-rce.html. Acesso em: 25 abr. 2023.

_____. **Splunk Enterprise**. [2023?]. Disponível em: https://www.splunk.com/en_us/products/splunk-enterprise.html. Acesso em: 25 abr. 2023.

STUDIO 3T. **Studio 3T, the professional GUI for MongoDB**. [2023?]. Disponível em: <https://studio3t.com/>. Acesso em: 29 abr. 2023.

STIVANI, M. O que é bug? Entenda a origem da palavra e conheça exemplos. **TechTudo**, 21 jan. 2019. Disponível em: <https://www.techtudo.com.br/noticias/2019/01/o-que-e-bug-entenda-a-origem-da-palavra-e-conheca-exemplos.ghtml>. Acesso em: 12 jun. 2023.

TECHTUDO. **Draw.io online permite criar gráficos e desenhos grátis sem baixar nada.** [2023?]. Disponível em: <https://www.techtudo.com.br/tudo-sobre/drawio/>. Acesso em: 26 abr. 2023.

_____. **Download do Gmail:** saiba baixar e entrar no e-mail pelo PC ou celular [2023?]. Disponível em: <https://www.techtudo.com.br/tudo-sobre/gmail/>. Acesso em: 27 abr. 2023.

UDEMY. **Java COMPLETO 2023 Programação Orientada a Objetos +Projetos.** Abr. 2023. Disponível em: <https://www.udemy.com/course/java-curso-completo/>. Acesso em: 22 abr. 2023.

ZOOM. **Zoom meetings:** video conferencing with chat | zoom. [2023?]. Disponível em: <https://explore.zoom.us/pt/products/meetings/>. Acesso em: 27 abr. 2023.



Documento Digitalizado Ostensivo (Público)

Relatório de Estágio Supervisionado

Assunto: Relatório de Estágio Supervisionado
Assinado por: Keilla Bezerra
Tipo do Documento: Relatório
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- Keilla Vitória Felipe Bezerra, ALUNO (202015020012) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO, em 27/10/2023 01:04:34.

Este documento foi armazenado no SUAP em 27/10/2023. 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: 980545
Código de Autenticação: b4ff76a46a

