

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

**RELATO DE EXPERIÊNCIA COMO ENGENHEIRO DE SOFTWARE NA
EMPRESA ADIRETO BRASIL NO SETOR DE MARKETING DIRETO**

AGUIRRE SABINO FONSECA

**CAJAZEIRAS - PB
2024**

AGUIRRE SABINO FONSECA

**RELATO DE EXPERIÊNCIA COMO ENGENHEIRO DE SOFTWARE NA EMPRESA
ADIRETO BRASIL NO SETOR DE MARKETING DIRETO**

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. Esp. João Igor Barros Rocha.

**CAJAZEIRAS - PB
2024**

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

F676r Fonseca, Aguirre Sabino.
Relato de experiência como engenheiro de software na empresa Adireto Brasil no setor de marketing direto / Aguirre Sabino Fonseca. – 2024.

44f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.

Orientador(a): Prof. Esp. João Igor Barros Rocha.

1. Desenvolvimento de sistemas. 2. Mercado de Tecnologia. 3. Relato de experiência. 4. Prática profissional. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.



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

AGUIRRE SABINO FONSECA

RELATO DE EXPERIÊNCIA PROFISSIONAL NO MERCADO DE TI

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. Esp. João Igor Barros Rocha

Aprovada em: **18 de Outubro de 2024.**

Prof. Esp. João Igor Barros Rocha - Orientador

Prof. Me. Afonso Serafim Jacinto - Avaliador
IFPB - Campus Cajazeiras

Prof. Me. Wysterlanya Kyury Pereira Barros
IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Joao Igor Barros Rocha, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 19/10/2024 14:17:06.
- **Wysterlanya Kyury Pereira Barros, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 19/10/2024 14:37:50.
- **Afonso Serafim Jacinto, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 20/10/2024 08:22:06.

Este documento foi emitido pelo SUAP em 19/10/2024. 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 622361

Verificador: 1f8247d4ef

Código de Autenticação:



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



ATA 41/2024 - CADS/UNINFO/DDE/DG/CZ/REITORIA/IFPB

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO (TCC) CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (ADS)

Às 15h00 do dia 18 do mês de OUTUBRO do ano de 2024, o(a) **aluno(a) AGUIRRE SABINO FONSECA**, matrícula **201612010164**, apresentou, como parte dos requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, seu trabalho de conclusão de curso, tendo como título "**RELATO DE EXPERIÊNCIA PROFISSIONAL NO MERCADO DE TI**". Constituíram a banca examinadora os professores **João Igor Barros Rocha** (orientador), **Afonso Serafim Jacinto** (examinador) e **Wysterlanya Kyury Pereira Barros** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **95**, com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho com as correções sugeridas pelos membros da banca examinadora. Eu, Francisco Paulo de Freitas Neto, Coordenador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, lavrei a presente ata, que segue assinada digitalmente por mim e pelos membros da banca examinadora.

Cajazeiras, 19 de outubro de 2024.

Documento assinado eletronicamente por:

- Francisco Paulo de Freitas Neto, COORDENADOR(A) DE CURSOS - FUC1 - CADS-CZ, em 19/10/2024 08:57:20.
- Wysterlanya Kyury Pereira Barros, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/10/2024 10:48:51.
- Afonso Serafim Jacinto, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/10/2024 10:58:23.
- Joao Igor Barros Rocha, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 19/10/2024 14:17:38.
- Aguirre Sabino Fonseca, ALUNO (201612010164) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS em 21/10/2024 09:29:45.

Este documento foi emitido pelo SUAP em 19/10/2024. 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 622360
Verificador: 2ea6f1e4ef
Código de Autenticação:



AGRADECIMENTOS

Expresso minha profunda gratidão aos meus pais, cuja dedicação, amor incondicional e apoio constante foram a base que sustentou minha trajetória acadêmica e profissional. Sem o incentivo deles, este momento não seria possível.

À minha esposa, minha companheira de vida, agradeço imensamente pela paciência, compreensão e pelo apoio inabalável, mesmo nos momentos de ausência. Sua confiança e motivação foram peças-chave para que eu pudesse alcançar este objetivo.

Ao Instituto Federal da Paraíba - Campus Cajazeiras, sou grato pela excelente formação acadêmica e pelas oportunidades de crescimento pessoal e profissional. Aos professores e colegas de curso, obrigado por compartilharem conhecimentos e experiências que foram além da sala de aula.

À Adireto Brasil Tecnologia Holding Ltda., meu sincero agradecimento por me proporcionar um ambiente de trabalho desafiador, onde pude crescer profissionalmente e aprender diariamente. Aos meus colegas de trabalho, agradeço pela colaboração, pelo espírito de equipe e pela troca constante de conhecimento.

A todos que, de alguma forma, contribuíram direta ou indiretamente para a minha formação e para a conclusão deste trabalho, deixo meu mais profundo agradecimento.

RESUMO

Este trabalho apresenta um relato de experiência profissional no mercado de tecnologia da informação, focando nas atividades desenvolvidas pelo autor como engenheiro de software na empresa Adireto Brasil Tecnologia Holding Ltda. São descritas as principais responsabilidades assumidas, incluindo a integração de serviços, migração de plataforma, otimização de performance em integrações de API, implementação de um serviço de agendamento centralizado e migração de servidores. O documento aborda o processo de desenvolvimento adotado pela empresa, destacando o uso do método Kanban, e detalha as tecnologias utilizadas, como TypeScript, NestJS, MongoDB, Docker e ferramentas de infraestrutura como código. São também relatadas as dificuldades encontradas durante o período de adaptação. O trabalho contextualiza a aplicação prática dos conhecimentos adquiridos durante a formação acadêmica no Instituto Federal da Paraíba, demonstrando como estes contribuíram para o desempenho profissional do autor.

Palavras-chave: Mercado de Tecnologia; Relato de Experiência; Desenvolvimento de Software; Adireto.

ABSTRACT

This work presents a professional experience report in the Information Technology market, focusing on the activities carried out by the author as a software engineer at Adireto Brasil Tecnologia Holding Ltda. The main responsibilities undertaken are described, including service integration, platform migration, performance optimization in API integrations, implementation of a centralized scheduling service, and server migration. The document discusses the development process adopted by the company, highlighting the use of the Kanban method, and details the technologies employed, such as TypeScript, NestJS, MongoDB, Docker, and infrastructure as code tools. Challenges faced during the adaptation period are also reported. The work contextualizes the practical application of the knowledge acquired during the academic education at the Federal Institute of Paraíba, demonstrating how this contributed to the author's professional performance.

Keywords: Technology Market; Experience Report; Software Development; Adireto.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação dos projetos de integração	18
---	----

LISTA DE ABREVIATURAS E SIGLAS

AMQP	<i>Advanced Message Queuing Protocol</i>
API	<i>Application Programming Interface</i>
AWS	Amazon Web Services
CD	<i>Continuous Deployment</i>
CI	<i>Continuous Integration</i>
CLT	Consolidação das Leis do Trabalho
DBaaS	<i>Database as a Service</i>
E/S	Entrada/Saída
GA4	Google Analytics 4
GCP	Google Cloud Platform
HCL	<i>HashiCorp Configuration Language</i>
IaC	<i>Infrastructure as Code</i>
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
PJ	Pessoa Jurídica
UA	Universal Analytics
VPN	<i>Virtual Private Network</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Importância da formação no Instituto Federal da Paraíba	13
1.2	Objetivos	14
1.2.1	Objetivo geral	14
1.2.2	Objetivos específicos	14
1.3	Organização do documento	14
2	IDENTIFICAÇÃO DA EMPRESA	15
3	DESCRIÇÃO DAS ATIVIDADES REALIZADAS	17
3.1	Integração de serviços	18
3.1.1	Migração para o GA4	20
3.1.2	Melhoria da performance nas integrações com a Analytics Data API	20
3.2	Implementação de um serviço para centralizar agendamentos de processos	22
3.3	Migração dos servidores para a DigitalOcean	23
4	PROCESSO DE DESENVOLVIMENTO	25
4.1	Kanban	25
4.1.1	<i>Backlog</i>	26
4.1.2	<i>To Do</i>	26
4.1.3	<i>In Progress</i>	26
4.1.4	<i>In Review</i>	26
4.1.5	<i>Staging</i>	27
4.1.6	<i>Done</i>	27
4.2	Ferramentas	27
5	TECNOLOGIAS UTILIZADAS	28
5.1	Typescript	28
5.2	NestJS	29

5.3	Node.js	29
5.4	MongoDB	30
5.5	Redis	31
5.6	RabbitMQ	32
5.7	Nginx	32
5.8	Docker	33
5.9	Terraform	34
5.10	Ansible	34
5.11	DigitalOcean	35
5.12	Codeship	36
6	DIFICULDADES ENCONTRADAS	37
6.1	Adaptação ao regime de trabalho remoto	37
6.2	Adaptação ao modelo de contratação pessoa jurídica	38
6.3	Novas tecnologias	39
7	CONCLUSÃO	40
	REFERÊNCIAS	41

1 INTRODUÇÃO

A crescente digitalização dos meios de comunicação e o avanço das tecnologias redefiniram as interações entre marcas e consumidores, exigindo das empresas uma profunda adaptação estratégica para prosperar no novo cenário. Conforme destacado por Strauss e Frost (2013), o marketing digital se tornou um pilar essencial para o sucesso empresarial, moldando a maneira como as empresas constroem suas marcas e se relacionam com o público. Esta mudança exige das empresas uma compreensão mais profunda do comportamento do cliente, adaptando suas estratégias para aproveitar as tecnologias mais recentes e enfrentar os desafios do meio online, como a segmentação das audiências e a alta concorrência.

Diante dessas mudanças no cenário digital, o marketing direto também passou por uma significativa evolução. Tradicionalmente definido por Kotler e Keller (2019) como "o uso de canais diretos ao consumidor para se comunicar e entregar bens e serviços aos clientes sem o uso de intermediários", o marketing direto encontrou no ambiente online novas possibilidades de atuação. A digitalização permitiu que essa abordagem se tornasse ainda mais eficaz, possibilitando uma comunicação mais pertinente e individualizada com os consumidores. Além disso, os avanços em coleta e processamento de dados aprimoraram substancialmente a capacidade de mensuração e análise dos resultados das campanhas, permitindo que as empresas refinem continuamente suas estratégias com base em dados concretos sobre o comportamento e as preferências do público-alvo.

Nesse contexto de rápida evolução tecnológica e transformação nas práticas de marketing, os profissionais de tecnologia da informação possuem um importante papel. Eles são responsáveis por desenvolver e implementar as soluções que viabilizam essas estratégias de marketing, como sistemas de gerenciamento do relacionamento com o cliente, que ajudam as empresas a centralizar informações e melhorar o atendimento; plataformas de automação de marketing, que permitem a criação e execução de campanhas automatizadas e segmentadas; e ferramentas avançadas de análise de dados, que oferecem informações detalhadas para otimizar o desempenho de campanhas. A capacidade de criar essas soluções tecnológicas, que facilitam a segmentação precisa do público-alvo, a personalização em larga escala e a mensuração detalhada dos resultados, é fundamental para o sucesso do marketing direto digital.

Este trabalho relata a experiência profissional do autor como engenheiro de software na Adireto Brasil Tecnologia Holding Ltda., uma empresa especializada em

publicidade e propaganda digital.

1.1 IMPORTÂNCIA DA FORMAÇÃO NO INSTITUTO FEDERAL DA PARAÍBA

A formação acadêmica no curso de Análise e Desenvolvimento de Sistemas do Instituto Federal da Paraíba (IFPB) desempenhou um papel fundamental na preparação do autor para os desafios enfrentados no mercado de trabalho. O curso proporcionou não apenas conhecimentos teóricos, mas também experiências práticas que contribuíram diretamente para o desenvolvimento profissional.

Desde as disciplinas iniciais até os módulos mais avançados, o curso ofereceu oportunidades que transcenderam a sala de aula. Por exemplo, no primeiro semestre de 2017, a disciplina de Sistemas Operacionais possibilitou a obtenção de uma certificação da Cisco sobre Linux, fornecendo conhecimentos e habilidades que são frequentemente aplicados na rotina de trabalho do autor.

A abordagem prática e orientada à inovação foi outro aspecto marcante do curso. No segundo semestre de 2017, durante a disciplina de Bancos de Dados II, o incentivo do professor para explorar tecnologias além do currículo formal levou o autor a aplicar conhecimentos iniciais em NodeJS no desenvolvimento de um projeto. Essa experiência não apenas ampliou sua familiaridade com novas tecnologias, mas também reforçou a importância da adaptação e do aprendizado contínuo.

Além das atividades acadêmicas, a participação na empresa júnior do instituto, inicialmente chamada Recursive e posteriormente conhecida como Loopis, foi outro ponto significativo na formação. Essa experiência proporcionou um primeiro contato com clientes reais e o desenvolvimento de habilidades essenciais para a gestão de projetos, comunicação eficaz e resolução de problemas práticos.

A combinação de teoria, prática e oportunidades extracurriculares oferecidas pelo IFPB forneceu a confiança e as competências necessárias para enfrentar os desafios do mercado de trabalho. Essa base sólida construída durante o curso foi fundamental para que o autor pudesse lidar com segurança com as dificuldades encontradas em sua carreira profissional, demonstrando a eficácia do curso em preparar seus alunos para as exigências reais de um ambiente corporativo.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Descrever as atividades realizadas pelo autor como engenheiro de software na Adireto Brasil Tecnologia Holding Ltda., com foco nos processos de desenvolvimento de software, destacando as tecnologias e metodologias utilizadas no contexto de uma empresa de marketing.

1.2.2 Objetivos específicos

- Descrever o processo de socialização organizacional e adaptação inicial na empresa.
- Apresentar as práticas de engenharia de software adotadas no dia a dia que auxiliam no desenvolvimento de projetos.
- Identificar e analisar as principais tecnologias e ferramentas empregadas pela equipe de desenvolvimento.
- Relatar as dificuldades encontradas pelo autor.

1.3 ORGANIZAÇÃO DO DOCUMENTO

O documento inicia com uma apresentação da empresa, sua história e atuação no mercado de marketing digital, como detalhado no Capítulo 2. Em seguida, no Capítulo 3, são detalhadas as atividades técnicas realizadas pelo autor na empresa.

No Capítulo 4, o trabalho aborda a análise do processo de desenvolvimento de software da empresa, focando no método Kanban e no uso de ferramentas como Asana. No Capítulo 5, é apresentado um panorama das tecnologias e ferramentas utilizadas, contextualizando suas aplicações práticas.

No Capítulo 6, são apresentados os desafios enfrentados pelo autor em sua experiência profissional. O Capítulo 7 encerra o trabalho com uma síntese dos pontos principais discutidos ao longo do documento.

2 IDENTIFICAÇÃO DA EMPRESA

A empresa objeto deste estudo é a Adireto Brasil Tecnologia Holding Ltda., uma entidade privada que atua no setor de marketing direto com especialidade em diversas áreas, incluindo marketing online, criação e publicação de conteúdos e plataformas de publicidade.

As operações da Adireto no Brasil tiveram início no primeiro semestre de 2015, com o objetivo principal de se profissionalizar como afiliado no mercado de marketing digital. Este período foi marcado por muitos desafios e aprendizados, à medida que a empresa buscava estabelecer sua presença em um mercado competitivo e em constante evolução. Com a persistência e dedicação de seus fundadores, a empresa começou a alcançar resultados expressivos. Um marco significativo nessa trajetória foi atingir o valor de um milhão de reais em comissões na plataforma Hotmart, uma das principais plataformas de afiliados do Brasil.

Na fase inicial de suas operações como afiliados, os fundadores da Adireto concentraram-se na divulgação de produtos de terceiros, não possuindo produtos próprios. Esta abordagem permitiu que a empresa desenvolvesse um entendimento do mercado, das preferências dos consumidores e das melhores práticas de marketing digital sem os riscos associados à criação de produtos próprios. Esse período foi fundamental para o desenvolvimento de expertise em diversas plataformas de publicidade, como Facebook Ads, Google Ads, Taboola e Outbrain, que foram utilizadas de forma estratégica para escalar campanhas de marketing, visando atrair potenciais compradores para os produtos divulgados.

Com a experiência acumulada em marketing digital, a Adireto expandiu suas operações, passando a atuar também como co-produtora, além de afiliada. Essa estratégia envolveu, por exemplo, a aquisição de participação em outras empresas, oferecendo-lhes todo o conhecimento e expertise para otimizar seus resultados e ampliar seu alcance no mercado.

A sede da Adireto no Brasil está localizada em Brasília, no seguinte endereço: Setor de Mansões (SMT) conjunto 14 lote 12, Brasília, Distrito Federal, CEP 72023-470. Apesar de possuir um escritório físico, a empresa adota um modelo de trabalho distribuído globalmente, com uma parcela significativa de seus colaboradores atuando em regime remoto.

No que tange ao desenvolvimento de software, a Adireto está focada principalmente na criação de soluções tecnológicas que suportam suas operações de marketing digital. Estes softwares abrangem ferramentas de análise de dados, automação de campanhas publicitárias e otimização de performance em plataformas de anúncios online.

Abaixo, são apresentados os endereços eletrônicos para contato com a empresa:

- **Site:** <https://adireto.com>
- **LinkedIn:** <https://www.linkedin.com/company/adireto/>
- **E-mail de contato:** contact@adireto.com

É importante mencionar que a publicação deste documento foi informalmente autorizada pela liderança da empresa durante uma conversa por chat. Embora não tenha havido um processo formal de aprovação, o conteúdo apresentado aqui foi compartilhado com o conhecimento e consentimento dos responsáveis pela empresa. Ainda assim, foram tomados cuidados para respeitar as políticas de confidencialidade e as diretrizes éticas apropriadas para um trabalho acadêmico desta natureza.

3 DESCRIÇÃO DAS ATIVIDADES REALIZADAS

Ao ingressar na empresa, o desenvolvedor foi recebido com um processo estruturado de socialização organizacional. Este programa visava facilitar a familiarização com as metodologias, ferramentas e tecnologias empregadas no cotidiano da organização. Ele também se mostrou importante, ao possibilitar uma adaptação rápida do novo membro ao ambiente de trabalho.

Na primeira etapa do processo, foi introduzido o modelo de trabalho remoto e como a organização o gerencia, baseando-se essencialmente em feedback constante através de ferramentas de comunicação assíncrona (permite troca de mensagens sem interação simultânea) e síncrona (exige interação em tempo real). A comunicação ocorre predominantemente por meio de uma ferramenta de chat, onde os membros da equipe podem trocar mensagens e manter um histórico para referência futura. Chamadas de vídeo são situações excepcionais, em que a resolução rápida de problemas se faz necessária e pode ser mais eficaz por esse meio. As reuniões podem ser agendadas previamente ou combinadas de forma imediata durante uma conversa, desde que os envolvidos estejam disponíveis no momento.

Durante a integração, foram apresentadas as ferramentas e tecnologias essenciais para o trabalho diário. Entre essas ferramentas, destacam-se o Google Workspaces e GitHub. A primeira é uma plataforma que reúne diferentes serviços para comunicação e colaboração, como e-mail, armazenamento de documentos, calendário e videoconferência. Essa centralização facilita a comunicação entre os membros da equipe e permite a organização e o compartilhamento de informações de forma eficiente. O GitHub, por sua vez, é utilizado para versionamento de código e colaboração no desenvolvimento de software, permitindo que o trabalho seja organizado em repositórios e que as alterações sejam rastreadas e revisadas.

No que diz respeito às tecnologias adotadas, durante o período de socialização organizacional, foi dado um destaque significativo às tecnologias que seriam essenciais para o autor conseguir entregar suas primeiras atividades. Como resultado, o desenvolvedor integrou-se rapidamente em diversas atividades importantes nos projetos em andamento, cujas principais são apresentadas a seguir.

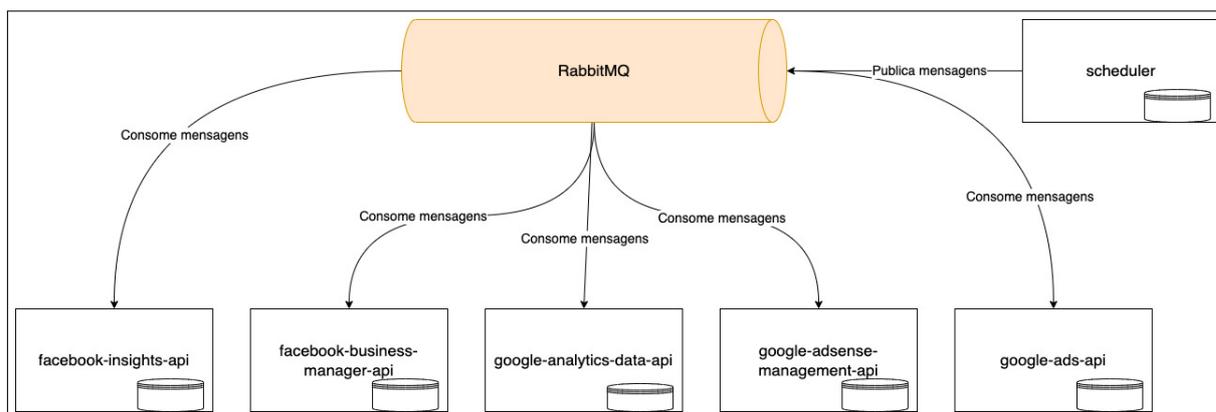
3.1 INTEGRAÇÃO DE SERVIÇOS

Os projetos de integração desempenham um importante papel na agregação e utilização de dados provenientes de diversas plataformas utilizadas pela empresa. Esses projetos, que variam em escopo e complexidade, utilizam uma estrutura típica de um *data warehouse* para consolidar informações sobre as campanhas de marketing em execução. Por meio desses dados, é possível extrair informações relevantes que são posteriormente consumidas pela operação da empresa, através de aplicações de análise próprias. Esses recursos são importantes para embasar as tomadas de decisões diárias, oferecendo uma visão abrangente e precisa do desempenho das campanhas. Além disso, os dados obtidos são explorados por outros serviços da organização, que por exemplo, se beneficiam para a implementação de automações estratégicas para as campanhas.

No dia a dia, o autor desenvolveu uma variedade de tarefas, incluindo a busca por informações das APIs de Marketing, Insights e Business Manager da Meta, bem como algumas APIs do Google, especialmente a Analytics Data API, AdSense Management API e a Ads API.

A Figura 1 apresenta a arquitetura dos principais serviços de integração, destacando o RabbitMQ como o componente central responsável pela comunicação entre os serviços por meio de filas. O RabbitMQ é uma plataforma de mensageria de código aberto que possibilita a troca confiável, escalável e assíncrona de mensagens entre serviços distribuídos.

Figura 1 – Representação dos projetos de integração



Fonte: Elaborado pelo autor (2024)

Por sua vez, o scheduler desempenha um papel fundamental na arquitetura, sendo um serviço desenvolvido internamente pela empresa. Ele é responsável por agendar a execução das integrações, determinando o momento exato em que cada uma deve iniciar o processo, de acordo com os parâmetros definidos durante o agendamento. Também é responsável por produzir as mensagens que serão enviadas para o RabbitMQ.

Por fim, os demais serviços apresentados assumem a tarefa de consumir as mensagens provenientes do RabbitMQ e, em seguida, processá-las, estabelecendo a conexão com uma das APIs mencionadas anteriormente.

Uma das primeiras atividades realizadas envolveu o desenvolvimento de novas integrações com a API de marketing da Meta, exigindo a sincronização de dados com as informações mais recentes fornecidas por essa API. Dentre essas informações, destaca-se o status de campanhas e contas de anúncios. Manter esses dados atualizados assegura a execução de processos apenas para itens ativos, evitando o desperdício de capacidade dos servidores com itens inativos. Por exemplo, quando uma conta de anúncio se torna inativa, as integrações correspondentes são desativadas, otimizando assim o uso dos recursos disponíveis.

Ainda no início de seu trabalho, o autor executou uma outra tarefa de maior complexidade, que consistiu na integração com o Google Ads, anteriormente conhecido como Google AdWords, utilizando a Ads API. Ao contrário da primeira atividade relacionada à API de marketing do Facebook, esta exigiu uma abordagem mais investigativa, uma vez que não existiam implementações prévias para servir de referência. Nesse sentido, foi necessário dedicar esforços à pesquisa de informações através de artigos, recursos de ajuda do Google Ads e vídeo aulas, visando adquirir uma compreensão abrangente da plataforma de publicidade do Google. Somente após essa fase preliminar, foi possível aprofundar o conhecimento na API e, finalmente, realizar as integrações de maneira eficaz e confiável.

Além de seu papel no desenvolvimento de novas integrações, o autor também se encarregou de atividades de manutenção, como a atualização das APIs e bibliotecas de terceiros e a melhoria dos serviços já existentes. Essas melhorias são direcionadas especialmente para a otimização de desempenho, visando a utilização eficiente dos recursos de hardware disponíveis. Entre essas atividades, serão destacadas a seguir a migração para o Google Analytics 4 (GA4) e também a otimização das integrações com a Analytics Data API.

3.1.1 Migração para o GA4

Em 1º de julho de 2023, o Google encerrou o processamento de novos dados para o Universal Analytics (UA), forçando a migração para o GA4, versão mais recente da plataforma. Antecipando-se a essa transição, o desenvolvedor assumiu a responsabilidade de conduzir as migrações para o novo ambiente do Google Analytics.

Inicialmente, ciente das disparidades entre os modelos de dados do UA e do GA4, o autor fez uma revisão das principais integrações já existentes, buscando mapear como essas implementações poderiam ser adaptadas ao GA4. A partir dessa revisão, foi definido um plano de ação detalhado, delineando os passos necessários para cada migração e estabelecendo as integrações que demandavam maior prioridade.

Com o intuito de viabilizar a recuperação de dados históricos do UA, optou-se pela elaboração de um novo módulo específico para o GA4, no qual as implementações prévias foram replicadas e adaptadas com base nos requisitos da nova versão. Apesar da opção de manter dois módulos para as integrações com o Google Analytics, assegurar a mesma suíte de testes para as novas implementações foi fundamental para preservar a consistência das regras de negócio estabelecidas anteriormente. Felizmente, todas as integrações com o Google Analytics seguem uma interface padronizada, o que facilitou a adaptação das suítes de testes, tornando o processo simples e rápido.

3.1.2 Melhoria da performance nas integrações com a Analytics Data API

Os relatórios do Analytics são elaborados para recuperar informações sobre as páginas de um site, proporcionando uma variedade de métricas que auxiliam na compreensão do comportamento do tráfego. Em 2022, identificou-se um problema de desempenho nas integrações com a Analytics Data API, que comprometia a análise desses dados. O processo de atualização apresentava atrasos consideráveis e, em alguns casos, não conseguia ser concluído.

Devido à demora no processamento dos dados, o sistema de monitoramento do RabbitMQ interpretava erroneamente a conexão como inativa, resultando em sua reinicialização. O RabbitMQ, seguindo seu protocolo de garantia de entrega, mantém as mensagens não processadas em uma fila persistente até que sejam confirmadas como concluídas pelo consumidor.

Quando a conexão era reiniciada, o RabbitMQ, não tendo recebido confirmação de processamento, reenviava essas mensagens para o consumidor. Isso agravava o problema existente, pois o consumidor, ainda lidando com o processamento da

mensagem original, recebia novamente a mesma tarefa.

Consequentemente, iniciava-se um processamento paralelo e redundante da mesma informação, consumindo recursos adicionais do servidor e levando a um estado de sobrecarga. Esta situação criava um ciclo vicioso em que o aumento no consumo de recursos prolongava ainda mais o tempo de processamento, perpetuando o problema.

Uma análise do código revelou falhas na arquitetura original que comprometiam a performance. O processo de integração adotava uma abordagem que pode ser descrita nas seguintes etapas:

- Recuperação de dados: Os dados da API eram obtidos e armazenados em memória.
- Transformação e padronização dos dados: Os dados da Analytics Data API passavam por um processo de adequação que incluía: a) Conversão para formatos compatíveis com os modelos da aplicação. b) Normalização de atributos para consistência. c) Adição de informações e cálculos derivados.
- Remoção de duplicatas: O processo identificava e consolidava URLs duplicadas. URLs com a mesma base, mas parâmetros diferentes (por exemplo, `example.com/pagina` com parâmetros `?id=1` e `?id=2`) eram consideradas iguais. As métricas dessas URLs eram somadas e mantidas em um único objeto, removendo os parâmetros. Este processo comparava cada URL com todas as outras na lista, incluindo ela mesma, resultando em uma complexidade $O(n^2)$. Para uma lista de 1000 objetos, seriam realizadas 1.000.000 de comparações ($1000 * 1000$), seguidas de iterações adicionais para consolidar os dados das URLs duplicadas. Esta abordagem, embora garantisse a identificação de todas as duplicatas, era computacionalmente intensiva e ineficiente para grandes volumes de dados.
- Armazenamento: Os dados consolidados eram inseridos em lote no MongoDB.

O processo enfrentava sérias ineficiências que comprometiam o desempenho e a confiabilidade do sistema. O consumo excessivo de memória, resultante do armazenamento integral dos dados recuperados da API, sobrecarregava os recursos disponíveis, especialmente durante a execução concorrente de múltiplos processos. Isso frequentemente levava a situações críticas de esgotamento de memória.

Além disso, o processamento sequencial adotado era ineficiente. Cada etapa operava sobre todos os dados, resultando em múltiplas iterações completas sobre o

conjunto. Essa abordagem não apenas era computacionalmente custosa, mas também aumentava o tempo de processamento com o aumento do volume de dados.

Para solucionar os problemas identificados, foi implementada uma abordagem baseada em *streaming* de dados, utilizando *generator functions* do JavaScript. Ao contrário da estratégia anterior, que armazenava todos os dados na memória, essa solução permite o processamento de cada item de forma individual e sequencial. Assim, à medida que um dado é processado, ele passa por todas as etapas antes de avançar para o próximo, eliminando a necessidade de manter o conjunto completo de dados em memória e otimizando o seu uso.

Na nova abordagem, não há mais necessidade de um algoritmo específico para tratar duplicatas. Agora, cada item passa pelo processo de transformação e normalização e, em seguida, é imediatamente salvo no banco de dados. A principal diferença é que, ao processar o mesmo relatório para uma mesma URL, as métricas são incrementadas diretamente no banco de dados. Isso garante que o relatório final contenha os dados corretos somados e apenas um item para cada URL por relatório.

Essa mudança na arquitetura do processo resultou em uma significativa melhoria na complexidade algorítmica. A implementação anterior tinha uma complexidade de $O(n^2)$ devido à necessidade de comparar cada URL com todas as outras na lista. A nova abordagem, por outro lado, apresenta uma complexidade linear $O(n)$, onde n é o número de itens no relatório. Isso ocorre porque cada item é processado apenas uma vez, sem a necessidade de comparações adicionais com outros itens.

A complexidade $O(n)$ da nova implementação melhora o desempenho, especialmente para grandes volumes de dados. À medida que o tamanho do relatório aumenta, o tempo de processamento cresce linearmente, em contraste com o crescimento quadrático da abordagem anterior. Além de melhorar o tempo de processamento, também reduziu o consumo de recursos do servidor.

3.2 IMPLEMENTAÇÃO DE UM SERVIÇO PARA CENTRALIZAR AGENDAMENTOS DE PROCESSOS

O desenvolvedor foi responsável pelo desenvolvimento da primeira versão de um sistema de agendamento, ao qual foi dado o nome *Scheduler*, utilizando o framework NestJS, banco de dados MongoDB e GraphQL para expôr uma API que permite o gerenciamento dos agendamentos. Esta seção destaca o processo de funcionamento, desde a preparação da mensagem até o seu disparo via RabbitMQ, além de apresentar os motivos que levaram à sua implementação.

O principal objetivo por trás da criação do *Scheduler* foi centralizar os agendamentos de todos os projetos citados na seção anterior, visando reduzir custos com servidores, e ao mesmo tempo oferecer maior controle e eficiência na gestão dos mesmos. Anteriormente, os agendamentos eram administrados de forma descentralizada, resultando em complexidade e desperdício de recursos. Além disso, também era um requisito importante simplificar o cadastro de novos agendamentos e isto se tornou possível através de uma API simples.

O funcionamento do *Scheduler* baseia-se na sua capacidade de consumir informações de diversas fontes de dados e preparar a carga de trabalho que será posteriormente publicada no RabbitMQ. Para isso, são realizadas verificações periódicas a cada minuto, identificando as mensagens que devem ser processadas e encaminhadas para uma *exchange* do RabbitMQ. Essa abordagem permite que as cargas de trabalho sejam executadas no momento adequado.

Uma estratégia adotada para otimizar o desempenho do *Scheduler* foi a implementação de uma camada de *cache* utilizando Redis. Essa camada de *cache* visa reduzir a necessidade de acessar o banco de dados a cada minuto, minimizando assim o tempo de processamento e aumentando a escalabilidade do sistema. É importante ressaltar que, apesar da complexidade das operações realizadas pelo *Scheduler*, ele foi projetado para operar de forma eficiente, exigindo apenas uma instância modesta com um gigabyte de memória e um núcleo de CPU.

3.3 MIGRAÇÃO DOS SERVIDORES PARA A DIGITALOCEAN

Anteriormente, todos os projetos estavam hospedados exclusivamente no serviço de nuvem do Google Cloud Platform (GCP), uma plataforma robusta e amplamente reconhecida no mercado de computação em nuvem. No entanto, embora o GCP ofereça uma ampla gama de recursos e serviços, os custos associados à sua utilização estavam se tornando uma preocupação, principalmente por conta dos modelos de pagamento, que geralmente se baseiam no uso e na demanda dos recursos, o que dificulta a previsão e o controle dos gastos.

Para resolver o problema da falta de previsibilidade dos custos relacionados aos servidores, decidiu-se realizar a migração para a DigitalOcean, uma plataforma de infraestrutura em nuvem conhecida por sua simplicidade, preços transparentes e que apresentou uma alternativa atraente ao modelo de pagamento do GCP. Com opções de preços simplificados, incluindo planos de pagamento fixos com recursos definidos, a DigitalOcean proporciona uma previsibilidade muito maior em termos de custos. Isso permitiu à equipe estimar de forma mais precisa os gastos e otimizar os recursos de

acordo com as necessidades dos projetos.

O autor deste trabalho foi responsável por executar as migrações dos projetos de *backend*. Diante da complexidade de migrar dezenas de serviços, o desenvolvedor optou por automatizar integralmente o processo. Isso incluía não apenas o provisionamento dos servidores na DigitalOcean, mas também sua configuração e a implementação de uma esteira de *Continuous Integration* (CI) e *Continuous Deployment* (CD).

Para alcançar essa automação, foram empregadas ferramentas amplamente utilizadas no campo da *Infrastructure as Code* (IaC). O Terraform, por exemplo, é uma ferramenta que permite a definição e o provisionamento de infraestrutura de forma declarativa. Com ele, é possível descrever a infraestrutura desejada em um arquivo de configuração e, em seguida, implantá-la de forma consistente e reproduzível em diferentes ambientes.

Já o Ansible é uma ferramenta de automação que simplifica a configuração e o gerenciamento de servidores e aplicativos. Utilizando linguagem YAML para descrever as tarefas, o Ansible permite automatizar processos como instalação de pacotes, configuração de serviços e aplicação de políticas de segurança.

Além disso, a implementação de uma esteira de CI/CD específica para a entrega de projetos de IaC foi fundamental para garantir a qualidade e a consistência do processo de desenvolvimento e implantação de infraestrutura. A esteira de CI/CD, gerenciada pelo Codeship Pro, permite automatizar a execução de testes, garantir a conformidade com padrões de formatação de código e realizar verificações de segurança utilizando o Trivy, uma ferramenta de código aberto que identifica vulnerabilidades no código fonte ou em imagens Docker.

4 PROCESSO DE DESENVOLVIMENTO

De acordo com Sommerville (2019), engenharia de software é uma disciplina de engenharia que se preocupa com os aspectos da produção de software, desde sua concepção inicial até sua operação e manutenção. Essa área abrange um conjunto de princípios, métodos e ferramentas para a análise, design, implementação, teste e manutenção de sistemas de software.

As metodologias de desenvolvimento variam desde modelos tradicionais, como o modelo cascata, até abordagens mais iterativas e incrementais, como as metodologias ágeis. Em um cenário de constante avanço tecnológico, é essencial possuir conhecimento teórico e prático sobre essas diferentes metodologias e conceitos da engenharia de software, pois com esse conhecimento será possível desenvolver projetos de alta qualidade que atendam às necessidades dos usuários.

Nesse contexto, as metodologias ágeis se destacam por permitir uma rápida adaptação às novas direções e demandas do mercado, pois elas seguem uma abordagem iterativa e incremental, que possui uma maior flexibilidade e facilita mudanças durante o ciclo de desenvolvimento. Enquanto que os modelos tradicionais são mais engessados, com etapas rígidas e sequenciais, dificultando a adaptação a novas necessidades e mudanças que surgem ao longo do projeto.

4.1 KANBAN

Entre as metodologias ágeis, o Kanban é uma das mais utilizadas na empresa pelo time de desenvolvimento. Criado no Japão e inicialmente aplicado na indústria automotiva pela Toyota, o Kanban também pode ser usado no desenvolvimento de software com o objetivo de dar maior visibilidade ao progresso das tarefas e aumentar a produtividade.

O Kanban é um sistema de gerenciamento visual que utiliza um quadro dividido em colunas que representam as diferentes etapas do processo de desenvolvimento, tais como "A Fazer", "Em Progresso" e "Concluído". As tarefas são representadas por cartões que se movem através dessas colunas, permitindo que a equipe tenha uma compreensão precisa do progresso e facilitando a identificação de gargalos no processo. Além disso, essa visibilidade ajuda na priorização e em uma melhor alocação de recursos, promovendo maior colaboração e transparência entre os membros da equipe.

No quadro Kanban utilizado pela equipe de desenvolvimento, segue-se uma sequência específica de colunas que refletem as etapas do processo de entrega de código. Essas colunas são: *Backlog*, *To Do*, *In Progress*, *In Review*, *Staging* e *Done*. A seguir, detalha-se cada uma dessas etapas.

4.1.1 *Backlog*

Esta coluna contém todas as tarefas e funcionalidades que foram identificadas e que precisam ser desenvolvidas. O *backlog* serve como um repositório centralizado de ideias, requisitos e melhorias que a equipe pretende implementar. Ele é continuamente atualizado e priorizado pelo time de desenvolvimento, garantindo que as tarefas de maior valor sejam tratadas primeiro.

4.1.2 *To Do*

Nesta coluna ficam as tarefas que foram priorizadas e estão prontas para serem iniciadas pelo time. Semanalmente, na segunda-feira, é realizada uma reunião para alinhar os objetivos da semana, com base no que está definido no *backlog*. Durante essa reunião, as tarefas selecionadas e priorizadas para a semana são movidas para a coluna *To Do*, com a proposta de serem concluídas ao longo dos próximos dias.

No entanto, demandas emergenciais provenientes principalmente da operação, como correções de problemas críticos, podem surgir e exigir atenção imediata. Essas demandas são inseridas diretamente na coluna *To Do*, podendo sobrepor tarefas previamente planejadas. A flexibilidade do Kanban permite que essas urgências sejam rapidamente priorizadas e iniciadas, ajustando o planejamento e alocando recursos conforme necessário para garantir a resolução ágil, mantendo a eficiência e a adaptabilidade do processo de desenvolvimento.

4.1.3 *In Progress*

Esta coluna mostra todas as tarefas que estão atualmente em desenvolvimento. Nesta fase, cada integrante trabalha ativamente em suas respectivas tarefas. Para manter um fluxo de trabalho eficiente e evitar sobrecarga, existe uma regra que define que cada membro da equipe tenha apenas um cartão atribuído nesta coluna.

4.1.4 *In Review*

Após a conclusão do desenvolvimento, a tarefa é movida para a coluna *In Review*. Nesta etapa, o código implementado passa por um processo de revisão detalhado, onde outros membros da equipe avaliam a qualidade e a conformidade do código com os padrões estabelecidos.

4.1.5 *Staging*

Após a aprovação do novo código na etapa de revisão, ele é movido para a coluna *Staging*. Nesta fase, a tarefa é implantada em um ambiente de pré-produção conhecido como *staging*. Este ambiente simula o ambiente de produção e permite a validação final das funcionalidades e integração do código antes do lançamento para os usuários finais.

Esta coluna tem um limite de apenas um cartão por projeto, essa restrição garante que cada tarefa recebendo atenção dedicada no ambiente de *staging* possa ser testada de forma independente, sem a influência de outras tarefas em desenvolvimento. Essa validação no ambiente de *staging* é importante para detectar problemas que não foram identificados nas etapas anteriores e para assegurar que o código está pronto para ser lançado em produção, mantendo a qualidade e a estabilidade do sistema.

4.1.6 *Done*

Esta coluna representa o trabalho concluído, indicando que a tarefa foi completamente finalizada e está agora disponível no ambiente de produção.

4.2 FERRAMENTAS

Para apoiar a implementação do Kanban e a gestão das tarefas, a empresa pode utilizar diferentes ferramentas, sendo as mais utilizadas pelo time o Asana, o Google Tasks e o GitHub Projects. O Asana é uma ferramenta de gestão de projetos que permite a criação de quadros Kanban, facilitando a visualização das tarefas e o acompanhamento do progresso da equipe. Nessa ferramenta são gerenciados projetos que são parte central do negócio da empresa, proporcionando uma visão ampla e integrada das atividades do time.

O Google Tasks é utilizado para gerenciar tarefas mais simples e individuais. Esta ferramenta proporciona uma maneira rápida de manter o controle das atividades diárias com uma abordagem mais direta e acessível.

O GitHub Projects é utilizado com menos frequência, principalmente em projetos específicos que não requerem uma visão geral de todo o time. Sua principal vantagem é a integração direta com o repositório de código, permitindo que *issues* e *pull requests* sejam facilmente vinculadas aos cartões do quadro.

5 TECNOLOGIAS UTILIZADAS

Diariamente, diversas tecnologias são empregadas para auxiliar a equipe de desenvolvimento na implementação, teste e manutenção de aplicações. A escolha das ferramentas apresentadas nesta seção baseia-se, primeiramente, no conhecimento prévio da equipe sobre elas, uma vez que maior familiaridade facilita a adaptação e permite um foco mais rápido na geração de valor para os serviços. Além disso, outros critérios são considerados, como a preferência por ferramentas de código aberto, a existência de uma comunidade ativa de suporte e o custo financeiro, buscando sempre a melhor ferramenta que ofereça o preço mais atraente.

5.1 TYPESCRIPT

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft e trata-se de um superconjunto do JavaScript que adiciona tipagem estática e outros recursos avançados à linguagem base. Isso significa que todo código JavaScript é um código TypeScript válido, mas o TypeScript oferece funcionalidades adicionais que facilitam o desenvolvimento e a manutenção de sistemas complexos. Entre as principais funcionalidades destacam-se:

- Detecção de erros em tempo de compilação: Muitos erros são identificados durante o processo de compilação para JavaScript, antes mesmo de o código ser executado, o que torna o desenvolvimento mais seguro.
- Tipagem estática: A tipagem estática permite ao desenvolvedor definir tipos para variáveis, parâmetros de funções e retornos de métodos. Principalmente em projetos maiores, isso ajuda a evitar alguns erros que só seriam detectados em tempo de execução no JavaScript.
- Suporte a versões futuras do JavaScript: O TypeScript é compatível com as versões mais recentes do ECMAScript, a especificação que serve de base para o JavaScript, o que permite que novos recursos planejados para futuras versões do JavaScript possam ser utilizados antes mesmo do seu lançamento. Isso só é possível, pois durante a compilação, o código TypeScript é transformado em JavaScript compatível com versões mais antigas do ECMAScript, garantindo que o código final funcione em uma ampla variedade de ambientes, ao mesmo tempo que aproveita as funcionalidades mais modernas.

Na empresa, o TypeScript é a principal linguagem utilizada em todos os projetos. No *backend*, é amplamente empregado em conjunto com o *framework* NestJS, enquanto que no *frontend* é utilizado com a biblioteca React. O uso do TypeScript nos projetos torna o desenvolvimento eficiente, garantindo uma melhor organização do código, facilitando a manutenção e escalabilidade das aplicações. Além disso, entrega mais confiança e segurança nos sistemas implementados, uma vez que erros comuns de programação são identificados mais facilmente, ainda no ambiente de desenvolvimento.

5.2 NESTJS

O NestJS é um *framework* para o desenvolvimento de aplicações *backend* em Node.js que suporta tanto TypeScript quanto JavaScript. Com uma abordagem opinativa, o *framework* orienta os desenvolvedores sobre as melhores práticas e organização de código, ao mesmo tempo em que oferece flexibilidade para a personalização.

Sua arquitetura foi fortemente inspirada no Angular, utilizado no desenvolvimento de aplicações de *frontend*, o que torna a curva de aprendizado do NestJS suave para quem já utiliza o Angular.

O *framework* oferece uma série de recursos avançados, incluindo injeção de dependências, controle de rotas e *middlewares*, validação de dados e suporte para testes unitários e de integração. Além disso, oferece suporte nativo que simplifica a integração de diversas tecnologias, como GraphQL, RabbitMQ, Kafka e MongoDB, acelerando o desenvolvimento das aplicações. Devido a essas características, o NestJS foi adotado como padrão para o desenvolvimento de projetos de *backend* na empresa.

5.3 NODE.JS

O Node.js é uma plataforma de desenvolvimento de código aberto que permite a execução de JavaScript fora do navegador, utilizando o motor V8 do Google Chrome. Essa característica possibilita o desenvolvimento de aplicações do lado do servidor utilizando uma única linguagem de programação, tanto no *frontend* quanto no *backend*, promovendo uma maior uniformidade no desenvolvimento de software.

Com sua arquitetura orientada a eventos e modelo de Entrada/Saída (E/S) não bloqueante, o Node.js é uma alternativa eficiente em cenários que exigem alta concorrência e processamento de dados em tempo real. Este modelo elimina a sobrecarga de gerenciamento de múltiplas *threads*, comum em outras linguagens de programação. Ao evitar o bloqueio da *thread* principal durante operações de E/S, o

Node.js maximiza a utilização de recursos e garante a escalabilidade eficiente das aplicações, permitindo atender a um grande volume de requisições simultâneas com baixo consumo de memória.

O Node.js é utilizado em todos os projetos de *backend*, em combinação com o TypeScript e o NestJS. O uso conjunto dessas tecnologias, além de permitir o desenvolvimento de aplicações escaláveis e de fácil manutenção, também acelera o processo de construção do software.

5.4 MONGODB

O MongoDB é um sistema de gerenciamento de banco de dados não relacional de código aberto conhecido por sua facilidade de uso e capacidade de oferecer alta performance e escalabilidade, mesmo em cenários com grandes volumes de dados e consultas complexas. Ao contrário dos bancos de dados relacionais tradicionais, o MongoDB adota um modelo de armazenamento baseado em documentos, utilizando o formato *Binary JSON (BSON)*, o que possibilita uma modelagem flexível e dinâmica dos dados.

O MongoDB oferece diversas opções de provisionamento para atender às necessidades específicas das aplicações:

- *Standalone*: Ideal para desenvolvimento e pequenas aplicações. Ele oferece uma configuração simples e atende adequadamente a cenários com baixa demanda.
- *Replica Set*: Recomendado para ambientes de produção que requerem alta disponibilidade e tolerância a falhas, formando um conjunto de réplicas que garantem a redundância dos dados.
- *Sharding*: Utilizado para distribuir dados entre múltiplos servidores, proporcionando a escalabilidade horizontal necessária para grandes volumes de dados. Também é possível combinar com o modelo *replicaset*, assegurando que cada *shard* possua alta disponibilidade e tolerância a falhas.

Embora um banco de dados como o PostgreSQL pudesse atender às necessidades da empresa, o MongoDB é preferido por oferecer uma característica extremamente importante, que é a flexibilidade para modificar a estrutura dos dados. Essa flexibilidade permite realizar mudanças rápidas nas aplicações, algo essencial para adaptá-las às necessidades do negócio, sem enfrentar as complexidades típicas de reestruturações em bancos de dados relacionais. Além disso, a equipe possui uma

vasta experiência com o MongoDB, o que resulta em entregas mais rápidas, sistemas confiáveis e com excelente performance.

A empresa utiliza o MongoDB Atlas, uma plataforma de *Database as a Service* (DBaaS) que gerencia todos os bancos de dados de produção. Esta solução libera a equipe de desenvolvimento das preocupações com infraestrutura, como provisionamento, configuração, manutenção e segurança dos servidores. Com o MongoDB Atlas cuidando dessas tarefas, a equipe pode concentrar-se totalmente no desenvolvimento de funcionalidades e na melhoria dos sistemas.

5.5 REDIS

O Redis é um sistema de armazenamento de estrutura de dados em memória de código aberto, altamente escalável e com excelente desempenho. É comumente usado em sistemas para fornecer um *cache* de dados rápido e evitar acessos aos bancos de dados principais, que persistem no disco e conseqüentemente possuem um tempo de leitura maior.

Devido à sua estrutura de chave-valor, o Redis é extremamente versátil e pode ser aplicado em diversos cenários que requerem acesso rápido aos dados. Além de servir como armazenamento de *cache*, o Redis é utilizado para gerenciar filas de tarefas, sistemas de notificação ou chat por meio do padrão de *Publisher/Subscriber* (Pub/Sub), e para a contagem de eventos em tempo real. Este último caso de uso é particularmente útil para implementar mecanismos de limitação de acesso, onde, por exemplo, um usuário pode ser restringido a executar uma determinada ação no sistema apenas dez vezes por hora.

Na empresa, o Redis é principalmente utilizado para melhorar o desempenho dos sistemas, com o objetivo de proporcionar uma boa experiência ao usuário. Informações que são acessadas com frequência e raramente atualizadas são armazenadas no Redis, aliviando a carga dos bancos de dados principais ao reduzir o número de consultas repetitivas. Além disso, relatórios que agregam dados de diversas fontes e exigem processamento intensivo são pré-computados e armazenados no Redis. Esses relatórios têm um tempo de expiração definido, garantindo que os usuários tenham acesso a versões atualizadas periodicamente. Durante esse período, o sistema entrega os relatórios de forma rápida, sem a necessidade de consultar novamente os bancos de dados.

5.6 RABBITMQ

O RabbitMQ é um sistema de mensagens de código aberto que atua como um intermediário entre os diferentes componentes de um sistema distribuído, facilitando a comunicação assíncrona e a troca de mensagens entre eles. Ele implementa o protocolo de mensagens *Advanced Message Queuing Protocol (AMQP)*, oferecendo uma solução escalável para a comunicação entre sistemas distribuídos.

O RabbitMQ opera com base em um modelo de troca de mensagens assíncrono, onde os produtores de mensagens enviam dados para as filas de mensagens, e os consumidores as recuperam conforme necessário. Ele utiliza um sistema de filas para armazenar temporariamente as mensagens enviadas pelos produtores, garantindo que elas sejam processadas de forma confiável e ordenada pelos consumidores.

Os principais componentes do RabbitMQ incluem as filas de mensagens (*message queues*), as trocas de mensagens (*message exchanges*) e as ligações (*bindings*) entre elas. As filas são responsáveis por armazenar temporariamente as mensagens enviadas pelos produtores até que os consumidores estejam prontos para processá-las. As trocas são responsáveis por receber as mensagens dos produtores e decidir para qual fila cada mensagem deve ser encaminhada com base em regras de roteamento predefinidas. As ligações definem as relações entre as trocas e as filas, especificando como as mensagens devem ser encaminhadas de uma para a outra. Esses componentes trabalham em conjunto para facilitar a comunicação assíncrona e a troca de mensagens entre os diferentes componentes de um sistema distribuído.

5.7 NGINX

O Nginx é um servidor web de código aberto conhecido por sua eficiência e alto desempenho em servir conteúdo estático, bem como por sua capacidade de atuar como proxy reverso para servidores de aplicativos. Com sua arquitetura leve e modular, o Nginx é amplamente utilizado na indústria de desenvolvimento de software. Segundo a Netcraft, o Nginx foi usado em 20.42% dos sites mais movimentados.

Ele é aplicado na empresa de diversas formas para melhorar a infraestrutura e o desempenho das aplicações. Primeiramente, ele é usado como um proxy para contêineres, facilitando a gestão e o roteamento de tráfego para as aplicações que ficam em ambientes isolados. Além disso, em algumas aplicações, o Nginx também desempenha o papel de resolução de certificados, aumentando a segurança nas comunicações entre o usuário final e aplicação.

O Nginx é empregado como balanceador de carga, essencial para distribuir o tráfego entre múltiplas instâncias de um mesmo serviço hospedadas em servidores distintos. E ele também é utilizado para servir arquivos estáticos, que podem estar armazenados localmente nos servidores ou em serviços de armazenamento remoto.

5.8 DOCKER

O Docker é uma plataforma de código aberto que automatiza a implantação de aplicações em contêineres, permitindo que desenvolvedores empacotem uma aplicação e suas dependências em uma imagem, que poderá ser executada em um contêiner. Este contêiner pode ser executado de forma consistente, eliminando problemas de compatibilidade, em qualquer ambiente, seja de desenvolvimento ou produção.

Na empresa, o Docker é utilizado para rodar todos os serviços no ambiente de produção. Para cada nova modificação enviada ao repositório de um projeto no GitHub, é executada uma automação que gera uma imagem Docker atualizada na etapa de CI. Durante o CD, esta imagem é carregada no servidor de produção e executada, tornando a aplicação disponível para uso imediatamente após essa etapa final.

Ainda durante a CI, que é iniciada após a entrada de novo código no repositório do projeto, são executadas suítes de testes de unidade e de integração. Para os testes de integração, o Docker auxilia na criação de serviços de terceiros, como bancos de dados, necessários para as suítes de testes. Isso permite que os testes de integração sejam mais precisos e confiáveis, garantindo que a aplicação funcione corretamente em produção.

O Docker também é extremamente recomendado para uso pelos desenvolvedores da equipe em seus ambientes de desenvolvimento. Ao padronizar o ambiente através de contêineres, são eliminados possíveis problemas de compatibilidade que podem surgir devido ao uso de diferentes sistemas operacionais pelos desenvolvedores. Isso assegura que todos os membros da equipe trabalhem em um ambiente consistente e o mais próximo possível do que existe em produção.

Para simplificar o gerenciamento de aplicações que necessitam de múltiplos contêineres, utiliza-se o Docker Compose. O Docker Compose é uma ferramenta que permite definir e executar aplicações multi-contêiner. Através de um arquivo de configuração YAML, é possível especificar todos os serviços que compõem a aplicação, bem como suas dependências, redes e volumes. Com um único comando, o Docker Compose cria e inicia todos os serviços definidos, facilitando a orquestração e automação do ambiente. Isso é particularmente útil para ambientes de desenvolvimento e testes.

5.9 TERRAFORM

O Terraform é uma ferramenta de IaC desenvolvida pela HashiCorp. Ele permite que toda a infraestrutura necessária para uma empresa ou projeto específico seja provisionada de forma declarativa e automatizada, através de arquivos de configuração, utilizando a linguagem HashiCorp Configuration Language (HCL).

Uma característica importante do Terraform é a idempotência, que garante que a aplicação repetida de um mesmo código de configuração resulte no mesmo estado da infraestrutura, sem causar efeitos colaterais indesejados. Essa propriedade é fundamental para manter a consistência e a previsibilidade da infraestrutura, independentemente do número de vezes que o código seja aplicado.

O Terraform é compatível com múltiplos provedores de nuvem, como AWS, Azure e Google Cloud, evitando o *lock-in* com um único fornecedor. Essa compatibilidade é possível por conta dos *providers*, que são plugins específicos que permitem ao Terraform interagir com APIs de cada serviço de nuvem. Logo, apesar do Terraform permitir que uma infraestrutura utilize serviços de nuvem diferentes, um código utilizado para provisionar servidores na AWS precisará ser adaptado para provisionar servidores na Google Cloud, por exemplo.

O Terraform foi adotado na empresa durante a migração do GCP para a DigitalOcean, conforme apresentado neste trabalho. Durante esse processo, o Terraform facilitou o provisionamento dos servidores, trazendo agilidade na criação dos ambientes de produção e testes. Além disso, o uso do Terraform contribuiu para a padronização e documentação da infraestrutura.

Essa ferramenta continua a ser utilizada na empresa para gerenciar todos os recursos na nuvem da DigitalOcean. Foi desenvolvido um projeto template que serve como base para a implementação de novos projetos de infraestrutura, dando maior eficiência a equipe e facilitando a manutenção do código, uma vez que todos os projetos adotam um mesmo padrão.

5.10 ANSIBLE

O Ansible é uma ferramenta de código aberto que permite automatizar a configuração de sistemas. Desenvolvido pela Red Hat, o Ansible utiliza uma abordagem declarativa, onde o estado desejado dos sistemas é definido através de *playbooks* escritos em YAML. Ele é compatível com uma variedade de sistemas operacionais, como Linux, Windows e macOS, e pode ser utilizado para gerenciar tanto servidores

físicos quanto máquinas virtuais e contêineres, uma vez que a comunicação com os servidores é por meio do protocolo SSH, que oferece uma forma segura de acesso remoto a sistemas e permite a execução de comandos através de uma conexão criptografada.

Como o Terraform, o Ansible também possui a característica de idempotência, que garante que a aplicação dos *playbooks* sempre levará os sistemas ao estado desejado, independentemente do estado inicial.

Na empresa, o Ansible é empregado para automatizar a configuração de todos os servidores. A ferramenta é responsável por diversas tarefas, incluindo a instalação e atualização de softwares, a implementação de agentes de monitoramento dos servidores e a execução de configurações de segurança, como por exemplo, permitir apenas acesso utilizando chave pública/privada, remover o acesso superusuário (*root*) e criar novos usuários com responsabilidades específicas.

Assim como o Terraform, o Ansible foi adotado pela empresa durante a migração para a DigitalOcean. Ele desempenhou um importante papel acelerando o processo de transição e minimizando as chances de erro, uma vez que todas as operações são declaradas em código, reduzindo significativamente a intervenção humana.

5.11 DIGITALOCEAN

A DigitalOcean é uma plataforma de computação em nuvem fundada em 2012 nos Estados Unidos que oferece diversos serviços para desenvolvedores e empresas. Ela se destaca devido à sua simplicidade, preços acessíveis e facilidade de uso, sendo uma excelente opção para startups, pequenos e médios negócios.

A plataforma oferece uma variedade de recursos, incluindo servidores virtuais (também conhecidos como Droplets), armazenamento em blocos, bancos de dados gerenciados, balanceamento de carga e redes privadas virtuais (VPNs). Com uma interface intuitiva, os usuários podem facilmente provisionar, gerenciar e escalar recursos de computação em nuvem de forma eficiente.

A DigitalOcean é a escolha principal da empresa para hospedar seus serviços na nuvem. Essa preferência se fundamenta em diversos motivos, como custo acessível, facilidade de uso, recursos de automação e, especialmente, na confiança proporcionada pela escalabilidade necessária para atender às demandas atuais da empresa.

5.12 CODESHIP

O Codeship é uma plataforma de CI/CD, projetada para automatizar o ciclo de desenvolvimento e entrega de software. Ela oferece uma interface intuitiva que permite aos usuários configurar esteiras de CI/CD de forma simples, facilitando a automação de testes e a entrega de novas funcionalidades de forma rápida e segura.

Na empresa, o Codeship é utilizado em todos os projetos, sendo responsável por automatizar a entrega de novas atualizações que são submetidas aos repositórios do GitHub. Através dele, cada solicitação de mudança de código passa por testes automatizados e verificações para garantir que esteja em conformidade com os padrões de formatação e boas práticas estabelecidos pela equipe de desenvolvimento. Isso proporciona uma segurança de que os desenvolvedores estão sempre entregando código com um excelente nível de qualidade antes de ser integrado ao repositório principal.

Além disso, o Codeship permite que os desenvolvedores continuem focados no que realmente importa, como a entrega de novas funcionalidades ou correção de problemas. Ao automatizar tarefas repetitivas e garantir a qualidade do código através de testes automatizados, os desenvolvedores ganham um tempo que pode ser aplicado em atividades mais estratégicas e criativas no ciclo de desenvolvimento.

6 DIFICULDADES ENCONTRADAS

Ingressar no mercado de trabalho é um desafio em todas as profissões, e no campo da tecnologia não é diferente. Nesta seção, serão abordadas algumas das dificuldades encontradas pelo autor, desde a adaptação a novas tecnologias até a mudança para modelos de trabalho e contratação diferentes.

6.1 ADAPTAÇÃO AO REGIME DE TRABALHO REMOTO

A transição do modelo presencial para o remoto exigiu, por parte do autor deste trabalho, uma significativa adaptação ao novo regime. Inicialmente, a mudança trouxe desafios, especialmente em relação à criação de uma rotina que separasse atividades profissionais das pessoais.

No início, a dificuldade de dissociar-se mentalmente das responsabilidades profissionais tornou-se algo comum. Trabalhar em casa e sem uma rotina definida resultou em um envolvimento constante com as atividades da empresa. A presença contínua no computador, seja para adquirir novos conhecimentos aplicáveis ao trabalho ou para adiantar demandas fora do horário, tornou-se uma prática comum, dificultando ainda mais a desconexão do ambiente profissional.

A ausência de uma rotina saudável e equilibrada para separar o pessoal do profissional pode acarretar diversos problemas, como a falta de cuidado com a saúde, o isolamento social e o risco de esgotamento profissional. Essa mistura de responsabilidades tem o potencial de comprometer o bem-estar e portanto, é importante estabelecer hábitos saudáveis. O autor está sempre se policiando para manter uma rotina equilibrada, pois ele reconhece que dessa forma poderá maximizar sua performance e capacidade de gerar valor para a empresa, ao mesmo tempo em que preserva sua saúde física e mental.

Além disso, o autor considera importante a mudança ocasional do ambiente de trabalho para estimular a criatividade e produtividade. Ele acredita que trabalhar em um local diferente, como um café, uma biblioteca, um espaço de coworking, ou até mesmo em outro cômodo da casa, pode acabar com a monotonia de estar sempre no mesmo lugar e trazer novos estímulos visuais e sociais. O autor procura incorporar essa prática em sua rotina, reservando momentos para trabalhar em ambientes diversos, o que tem se mostrado eficaz para manter o entusiasmo e a inspiração em suas atividades profissionais.

6.2 ADAPTAÇÃO AO MODELO DE CONTRATAÇÃO PESSOA JURÍDICA

No Brasil, existem diferentes modalidades de contratação no mercado de trabalho, cada uma com suas particularidades, vantagens e desafios. A Consolidação das Leis do Trabalho (CLT) é o regime mais tradicional, oferecendo uma série de direitos e benefícios aos trabalhadores, como férias remuneradas, 13º salário, FGTS e seguro-desemprego. Este modelo proporciona maior estabilidade e segurança ao empregado, mas pode resultar em custos mais elevados para o empregador.

Além da CLT, o mercado de trabalho oferece outras modalidades de contratação para atender diferentes necessidades. O trabalho temporário permite contratar por períodos determinados para demandas transitórias, enquanto o contrato de estágio oferece oportunidades práticas para estudantes. Estas modalidades possuem regulamentações específicas, visando equilibrar as necessidades das empresas e dos trabalhadores em várias fases de suas carreiras.

Uma modalidade que tem ganhado destaque é a contratação de profissionais como Pessoa Jurídica (PJ). Neste modelo, o profissional se torna, efetivamente, um empresário, prestando serviços a outras empresas. Esta opção oferece potencialmente maior remuneração e flexibilidade, mas também traz consigo uma série de responsabilidades e desafios.

Ao optar pelo modelo PJ, o profissional precisa lidar com uma série de obrigações burocráticas e adquirir novos conhecimentos. Um dos principais desafios é a gestão das questões contábeis e fiscais. Embora seja recomendável contar com serviços especializados de contabilidade, é essencial que o profissional compreenda ao menos o básico desses procedimentos. Isso inclui desde a emissão de notas fiscais até o entendimento das obrigações tributárias associadas a este modelo de contratação.

O profissional PJ deve estar constantemente atento para garantir a conformidade com a legislação e evitar problemas fiscais. Além disso, é fundamental compreender as questões burocráticas junto à prefeitura local, pois cada município pode ter exigências específicas que precisam ser atendidas para assegurar que o negócio esteja em conformidade com as regulamentações locais.

A escolha pelo modelo PJ também traz implicações significativas em termos legais e contratuais. Enquanto um contrato CLT estabelece direitos e deveres claros entre empregado e empregador, a atuação como PJ requer a elaboração de contratos comerciais detalhados, que definem não apenas os serviços a serem prestados, mas também os termos de pagamento, prazos e responsabilidades das partes envolvidas.

Apesar dos desafios e burocracias associados, especialmente durante o processo de abertura da empresa, o modelo PJ também oferece vantagens consideráveis. Entre elas, destacam-se a maior flexibilidade para definir o horário de trabalho e a liberdade para gerir os recursos financeiros. Para o autor, essas vantagens se mostraram significativas, fazendo com que a escolha por este modelo de contratação fosse considerada acertada, alinhando-se com suas necessidades e objetivos profissionais.

6.3 NOVAS TECNOLOGIAS

Ao ingressar na empresa, o autor deste trabalho encontrou alguns desafios ao lidar com novas tecnologias. Uma dessas dificuldades foi o framework NestJS, amplamente utilizado para os serviços de *backend*, com o qual o desenvolvedor não tinha experiência prévia. No entanto, a familiaridade com o Angular facilitou a sua curva de aprendizado, permitindo uma adaptação mais rápida ao NestJS.

Outro desafio foi a falta de experiência com o RabbitMQ, uma tecnologia de mensageria utilizada para comunicação entre serviços, como já foi detalhado na seção "Tecnologias utilizadas". A complexidade dessa ferramenta exigiu um investimento considerável de tempo em estudo e prática. O autor dedicou-se a compreender os aspectos técnicos e também os padrões de arquitetura de sistemas distribuídos nos quais essa tecnologia é comumente aplicada. Este processo de aprendizagem foi gradual e envolveu a realização de projetos práticos e a leitura de documentação técnica. Com o tempo e a persistência, o autor conseguiu atingir um nível de proficiência que lhe permitiu utilizar o RabbitMQ com segurança e atender as demandas do trabalho.

Embora o autor possuísse conhecimento prévio e certa afinidade com sistemas operacionais baseados no *kernel* Linux, o ambiente de produção da empresa apresentou novos desafios e oportunidades de aprendizado. Foi necessário um aprofundamento nos conhecimentos de administração de sistemas Linux, incluindo tópicos como gerenciamento de processos, otimização de performance, segurança e configuração de serviços. Este aprendizado foi essencial para alcançar e manter um nível de eficiência que inspirasse confiança na liderança do setor de tecnologia da empresa.

O domínio crescente dessas tecnologias permitiu que o autor não apenas cumprisse suas responsabilidades básicas, mas também propusesse e implementasse inovações na infraestrutura da empresa. Estas melhorias, fruto do conhecimento adquirido e da experiência acumulada, continuam sendo utilizadas e aprimoradas constantemente, demonstrando o valor do investimento no desenvolvimento profissional contínuo.

7 CONCLUSÃO

Este trabalho apresentou um relato da experiência profissional do autor como engenheiro de software na empresa Adireto Brasil Tecnologia Holding Ltda. Ao longo do documento, foram descritas as principais atividades desenvolvidas, as tecnologias utilizadas e os desafios enfrentados durante o período de adaptação e atuação profissional.

O curso de Análise e Desenvolvimento de Sistemas do IFPB demonstrouse fundamental na preparação do autor para os desafios profissionais enfrentados. A estrutura curricular, que combina sólida base teórica com experiências práticas, proporcionou os conhecimentos e habilidades necessários para uma transição bem-sucedida para o ambiente corporativo.

Além da formação acadêmica formal, a participação em atividades extracurriculares, como a empresa júnior do instituto, desempenhou um papel crucial no desenvolvimento profissional do autor. Essa experiência proporcionou uma exposição a situações reais de mercado, permitindo o aprimoramento de habilidades como gestão de projetos, comunicação e resolução de problemas práticos.

Na Adireto Brasil, o autor pôde aplicar e expandir os conhecimentos adquiridos durante sua formação. O trabalho com tecnologias modernas aliado à utilização de metodologias ágeis, demonstrou a aplicabilidade prática dos conceitos aprendidos na academia. Essa experiência também evidenciou a importância da adaptabilidade e do aprendizado contínuo na área de tecnologia da informação, onde novas ferramentas e métodos surgem constantemente.

Na sua trajetória profissional, o autor enfrentou desafios significativos, como a adaptação ao trabalho remoto, a transição para o modelo de contratação como Pessoa Jurídica e a necessidade de dominar novas tecnologias. Para superar essas dificuldades, foi necessário desenvolver habilidades de autogestão e disciplina, além de manter-se em constante capacitação. A implementação de rotinas, aliada à busca proativa por conhecimento e à disposição para enfrentar novos cenários, foram importantes para a superação desses obstáculos.

REFERÊNCIAS

AMAZON. **O que é um data warehouse?** Acesso em: 24 ago. 2024. Disponível em: <<https://aws.amazon.com/pt/what-is/data-warehouse/>>.

BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding typescript. In: JONES, R. (Ed.). **ECOOOP 2014 – Object-Oriented Programming**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 257–281. ISBN 978-3-662-44202-9.

CANTELON, M.; HARTEK, M.; HOLOWAYCHUK, T.; RAJLICH, N. **Node. js in Action**. [S.l.]: Manning Greenwich, 2014.

CARLSON, J. **Redis in Action**. [S.l.]: Manning, 2013.

CLOUDBEES. **CloudBees CodeShip**. Acesso em: 24 ago. 2024. Disponível em: <<https://www.cloudbees.com/products/codeship>>.

DIGITALOCEAN. **Meet DigitalOcean**. Acesso em: 24 ago. 2024. Disponível em: <<https://www.digitalocean.com/about>>.

DOCKER. **What is Docker?** Acesso em: 24 set. 2024. Disponível em: <<https://docs.docker.com/get-started/docker-overview/>>.

GOOGLE. **Ads API**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.google.com/google-ads/api/docs/start>>.

_____. **AdSense Management API**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.google.com/adsense/management>>.

_____. **Google Ads Help**. Acesso em: 24 ago. 2024. Disponível em: <<https://support.google.com/google-ads/answer/6146252?hl=en>>.

_____. **Google Analytics Data API Overview**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.google.com/analytics/devguides/reporting/data/v1>>.

_____. **O Google Analytics 4 substituiu o Universal Analytics**. Acesso em: 24 ago. 2024. Disponível em: <<https://support.google.com/analytics/answer/11583528?hl=pt-br>>.

_____. **Comparação de métricas: Google Analytics 4 x Universal Analytics**. Acesso em: 24 ago. 2024. Disponível em: <<https://support.google.com/analytics/answer/11986666?hl=pt-BR>>.

_____. **Dados do Universal Analytics x Google Analytics 4**. Acesso em: 24 ago. 2024. Disponível em: <<https://support.google.com/analytics/answer/9964640?hl=pt-BR>>.

HASHICORP. **What is Terraform?** Acesso em: 24 set. 2024. Disponível em: <<https://developer.hashicorp.com/terraform>>.

INTERNATIONAL, E. **ECMAScript® 2025 Language Specification**. Acesso em: 25 ago. 2024. Disponível em: <<https://tc39.es/ecma262/>>.

KOTLER, P.; KELLER, K. L. **Administração de Marketing**. 15. ed. São Paulo: Pearson Universidades, 2019. ISBN 9788543024950.

META. **Business Manager API**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.facebook.com/docs/business-manager-api/>>.

_____. **Insights API**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.facebook.com/docs/marketing-api/insights>>.

_____. **Marketing API**. Acesso em: 24 ago. 2024. Disponível em: <<https://developers.facebook.com/docs/marketing-apis/>>.

MONGODB. **Deploy a Standalone MongoDB Instance**. Acesso em: 17 set. 2024. Disponível em: <<https://www.mongodb.com/docs/cloud-manager/tutorial/deploy-standalone/>>.

_____. **Introduction to MongoDB**. Acesso em: 17 set. 2024. Disponível em: <<https://www.mongodb.com/docs/manual/introduction/>>.

_____. **Replication**. Acesso em: 17 set. 2024. Disponível em: <<https://www.mongodb.com/docs/manual/replication/>>.

_____. **Sharding**. Acesso em: 17 set. 2024. Disponível em: <<https://www.mongodb.com/docs/manual/sharding/>>.

NETCRAFT. **May 2024 Web Server Survey**. 2024. Acesso em: 24 set. 2024. Disponível em: <<https://www.netcraft.com/blog/may-2024-web-server-survey/>>.

NGINX. **Beginner's Guide**. Acesso em: 24 set. 2024. Disponível em: <https://nginx.org/en/docs/beginners_guide.html>.

_____. **nginx documentation**. Acesso em: 24 set. 2024. Disponível em: <<https://nginx.org/en/docs/>>.

NODE.JS. **Introduction to Node.js**. Acesso em: 17 set. 2024. Disponível em: <<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>>.

OPENSOURCE.COM. **What is Ansible?** Acesso em: 24 ago. 2024. Disponível em: <<https://opensource.com/resources/what-ansible>>.

DAN CORTAZIO. **1 Milhão em 1 Ano como Afiliado: O Grande Segredo - Palestra 100% Online e Gratuita**. 2018. Acesso em: 20 out. 2024. Disponível em: <https://www.youtube.com/watch?v=UT_xyuSHWlg>.

VIDA DE FREESIDER. **Palestra como anunciar e escalar no Facebook - Rafa Cortazio - Freesider Meeting**. 2018. Acesso em: 20 out. 2024. Disponível em: <<https://www.youtube.com/watch?v=0AyzAalDEz4>>.

RABBITMQ. **Exchanges and Exchange Types**. Acesso em: 24 ago. 2024. Disponível em: <<https://www.rabbitmq.com/tutorials/amqp-concepts#exchanges>>.

_____. **RabbitMQ 3.13 Documentation**. Acesso em: 24 ago. 2024. Disponível em: <<https://www.rabbitmq.com/docs>>.

SILVA, J. B. d.; ANASTÁCIO, F. A. d. M. Método kanban como ferramenta de controle de gestão. **Id on Line Rev.Mult. Psic.**, v. 13, n. 43, p. 1018–1027, 2019. ISSN 1981-1179.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.

STRAUSS, J.; FROST, R. **E-Marketing**. 7. ed. Upper Saddle River, NJ: Prentice Hall, 2013. ISBN-10: 0132953447. ISBN 9780132953443.

TYPESCRIPT. **TypeScript for the New Programmer**. Acesso em: 24 ago. 2024. Disponível em: <<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>>.

	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

Assunto:	TCC
Assinado por:	Aguirre Fonseca
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Aguirre Sabino Fonseca, ALUNO (201612010164) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 21/10/2024 17:44:08.

Este documento foi armazenado no SUAP em 21/10/2024. 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: 1286133

Código de Autenticação: 23636b808e

