

INSTITUTO FEDERAL

Paraíba

Campus Monteiro

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA -
CAMPUS MONTEIRO**

DIRETORIA DE DESENVOLVIMENTO E ENSINO

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

JAN JORIS TOMÉ DE LIRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL EM
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA AKROSS**

MONTEIRO

2025

JAN JORIS TOMÉ DE LIRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL EM
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA AKROSS**

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

Orientador: Prof. Me. Elenilson Vieira da
Silva Filho

MONTEIRO

2025

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

L768r Lira, Jan Joris Tomé de.

Relato de experiência profissional em desenvolvimento de software
na Empresa AKROSS / Jan Joris Tomé de Lira – Monteiro-PB. 2025.
37fls. : il.

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

Orientador: Prof. Esp. Elenilson Vieira da Silva Filho.

1. Software - desenvolvimento 2. Projetos – Sistemas
3. Banco Dados 4. Empresa AKROSS – Rio de Janeiro -RJ I. Título .

CDU 004.42

JAN JORIS TOMÉ DE LIRA

**RELATO DE EXPERIÊNCIA PROFISSIONAL EM
DESENVOLVIMENTO DE SOFTWARE NA EMPRESA AKROSS**

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

Aprovado em 24 / 02 / 2025.

BANCA EXAMINADORA:



Documento assinado digitalmente
ELENILSON VIEIRA DA SILVA FILHO
Data: 11/04/2025 16:05:26-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Elenilson Vieira da Silva Filho (Orientador)

Instituto Federal da Paraíba (IFPB)



Documento assinado digitalmente
CLEYTON CAETANO DE SOUZA
Data: 14/04/2025 03:29:27-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Cleyton Caetano de Souza

Instituto Federal da Paraíba (IFPB)



Documento assinado digitalmente
TIAGO BRASILEIRO ARAUJO
Data: 11/04/2025 08:44:30-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Tiago Brasileiro Araújo

Instituto Federal da Paraíba (IFPB)

MONTEIRO

2025

RESUMO

Este documento tem como objetivo relatar a experiência profissional adquirida pelo discente no ramo de desenvolvimento de sistemas da informação durante seu estágio na equipe de aplicativos da empresa Akross. Durante o estágio, o discente teve a oportunidade de colocar em prática seus conhecimentos e habilidades adquiridos ao longo de sua formação em Análise e Desenvolvimento de Sistemas, abrangendo competências em desenvolvimento de software, projeto de sistemas, planejamento de tarefas e bancos de dados. O discente atuou tanto em sistemas *backend* quanto em projetos de *frontend*, utilizando tecnologias como Java, Spring Boot, JavaScript e React Native. Além disso, implementou baterias de testes automatizados com Jest e WireMock, participou de projetos utilizados por grandes empresas com inúmeros usuários em todo o Brasil e integrou reuniões com colaboradores estrangeiros que testaram seu conhecimento em Inglês. A experiência de estágio contribuiu para a aquisição de novas competências sociais, como trabalho em equipe, comunicação efetiva e compreensão das necessidades do cliente, ao mesmo tempo em que aprimorou sua proficiência como programador e ampliou sua exposição às tecnologias de desenvolvimento de software, reforçando seu desenvolvimento profissional na área.

Palavras-chave: Análise e Desenvolvimento de Sistemas; estágio; Desenvolvimento de Software.

ABSTRACT

The purpose of this document is to report on the professional experience gained by the student in the field of systems development during his internship in the Applications Team at Akross. During the internship, the student had the opportunity to apply the knowledge and skills acquired during his studies in Systems Analysis and Development, which included competencies in *software* development, systems design, task planning, and databases. The student's work involved both *backend* systems and *frontend* projects, using technologies such as Java, Spring Boot, JavaScript, and React Native. Moreover, he implemented automated test suites with Jest and WireMock, contributed to projects utilized by major companies with extensive user bases across Brazil, and participated in meetings with foreign collaborators which tested his English language skills. The internship experience contributed to the acquisition of new social skills, such as teamwork, effective communication, and understanding customer needs, while at the same time improving his proficiency as a programmer and broadening his exposure to software development technologies, reinforcing his professional development in the field.

Keywords: Systems Analysis and Development; internship; Software Development.

LISTA DE ILUSTRAÇÕES

Figura 01 – Logo Akross.....	11
Figura 02 – Resultado da Landing Page.....	15
Figura 03 – Quatro designs únicos de portais cativos de clientes diferentes.....	17
Figura 04 – Interface de customização e gerenciamento do portal cativo.....	17
Figura 05 – Timeline de Features e subtarefas no Jira.....	18
Figura 06 – Pseudocódigo do Portal Config multilíngue.....	20
Figura 07 – Tabela de mensagens internacionalizadas.....	22
Figura 08 – Exemplo de Exception padronizada.....	22
Figura 09 – Mensagens de erro internacionalizadas.....	23
Figura 10 – Resultado da refatoração do PortalService.....	24
Figura 11 – Exemplos de interfaces de métricas no Kibana.....	30
Figura 12 – Pseudocódigo do bloqueio financeiro.....	31
Figura 13 – Variável armazenada no Banco de Dados.....	32
Figura 14 – Pseudocódigo do envio do novo endereço.....	34

LISTA DE SIGLAS E ABREVIATURAS

ADS	<i>Análise e Desenvolvimento de Sistemas</i>
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
B2B	<i>Business-to-Business</i>
B2C	<i>Business-to-Customer</i>
DevOps	<i>Development Operations</i>
ELK	<i>ElasticSearch, Logstash e Kibana</i>
GMUD	<i>Gestão de Mudanças</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IFPB	<i>Instituto Federal da Paraíba</i>
JPA	<i>Java/Jakarta Persistence API</i>
LLM	<i>Large Language Model</i>
RH	<i>Recursos Humanos</i>
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
UX	<i>User Experience</i>

SUMÁRIO

1	INTRODUÇÃO	8
2	OBJETIVOS	10
2.1	Objetivo geral	10
2.2	Objetivos específicos	10
3	A EMPRESA	11
3.1	Origem	11
3.2	Perfil da empresa	11
3.3	Estrutura organizacional	12
4	RELATO DE ESTÁGIO	13
4.1	Primeiro contato	13
4.2	Processo seletivo	13
4.3	Onboarding e a primeira atividade	14
4.4	Vivência na SQUAD do projeto Nomad Business	16
4.4.1	Sobre o projeto	16
4.4.2	Atuações em backend	20
4.4.2.1	Configuração multilíngue dos portais	20
4.4.2.2	Mensagens de erro internacionalizadas	21
4.4.2.3	Refatoração do serviço de portais	23
4.4.3	Reunião de investigação técnica em Inglês	24
4.4.4	Descoberta de falha nas regras de negócio	25
4.5	Vivência na SQUAD do projeto Codinome Fibra	27
4.5.1	Sobre o projeto	27
4.5.2	Onboarding e insegurança inicial	28
4.5.3	Atuações em backend	29
4.5.3.1	Estudo dos logs	29
4.5.3.2	Bloqueio financeiro na mudança de endereço	30
4.5.3.3	Correção na migração de plano	31
4.5.3.4	Impedimentos na mudança de endereço	32
4.5.3.5	Mudança de endereço com migração de oferta	33
5	CONSIDERAÇÕES FINAIS	35
	REFERÊNCIAS	37

1 INTRODUÇÃO

A tecnologia vem desempenhando um papel fundamental na solução de problemas que são considerados inviáveis ou até mesmo impossíveis de serem resolvidos por meios tradicionais. Esse fato foi claramente evidenciado durante a Segunda Guerra Mundial, quando o matemático e criptoanalista britânico Alan Mathison Turing (1912 – 1954) aplicou seus conhecimentos em computação para desenvolver algoritmos e melhorias que aceleraram a quebra do código de criptografia da máquina Enigma, utilizada pela Alemanha nazista para cifrar suas comunicações (Copeland, 2012).

Dessa forma, Turing não somente contribuiu para a vitória dos Aliados sobre o Eixo em diversas batalhas cruciais, encurtando a guerra e salvando inúmeras vidas no processo (Copeland, 2012), mas também revolucionou o campo da criptoanálise e da ciência da computação para sempre. Seu trabalho serve como prova do poder da inovação da tecnologia na superação de desafios aparentemente intransponíveis e na pavimentação do caminho para futuros avanços.

Com a tecnologia evoluindo a uma velocidade exponencial, sobretudo com o avanço da Inteligência Artificial Generativa e dos LLMs, a crescente necessidade de profissionais qualificados aptos a desenvolver, implementar e manter Sistemas da Informação complexos torna-se cada vez mais evidente (Visionnaire, 2024).

O curso de Análise e Desenvolvimento de Sistemas oferece uma capacitação profissional em todos os aspectos da engenharia de *software*, desde a elicitación e análise de requisitos até o projeto, implementação, teste e manutenção do sistema. Os estudantes têm a oportunidade de adquirir experiência prática por meio de projetos de pesquisa ou extensão, o que os auxilia no desenvolvimento de habilidades avançadas de resolução de problemas e na capacidade de pensar de forma criativa sob pressão. Após a conclusão do curso, os formandos estarão contemplados com as habilidades e técnicas necessárias para atender às demandas cada vez mais complexas de empresas e organizações que dependem intensivamente da tecnologia para alcançar o sucesso. (Portal do Estudante IFPB, s.d.)

O estágio supervisionado desempenha um papel fundamental na aptidão profissional do discente, preenchendo o espaço entre o aprendizado teórico e o mundo real. Ele proporciona ao discente a oportunidade de trabalhar ao lado de profissionais experientes da área em um ambiente empresarial, fazendo com que eles utilizem seus conhecimentos e habilidades adquiridos ao longo de sua formação para enfrentar desafios em projetos reais. Durante o estágio, o discente é exposto a várias práticas da área de Desenvolvimento de *Software*, culturas organizacionais e diferentes ambientes de trabalho, aperfeiçoando assim sua adaptabilidade e habilidades interpessoais. Além disso, ele poderá ter acesso a tecnologias e recursos que talvez não lhe foram ensinados durante sua capacitação acadêmica, o que o permite expandir ainda mais seus conhecimentos e explorar novas áreas de interesse. De modo geral, o estágio supervisionado ajuda o discente a realizar uma transição suave do mundo acadêmico para uma carreira profissional, proporcionando experiências valiosas e oportunidades de *networking* fundamentais para traçar seu futuro.

Dito isto, o presente trabalho tem como objetivo denotar a experiência adquirida pelo discente Jan Joris Tomé de Lira como estagiário em Desenvolvimento *Backend* em uma empresa de tecnologia. Este documento é dividido em vários capítulos, nos quais serão abordados tópicos como os objetivos do estágio e os detalhes da empresa onde ele foi realizado. Além disso, constarão relatos detalhados sobre os dois projetos em que o discente participou durante o estágio, apresentando algumas tarefas realizadas, desafios enfrentados e as contribuições efetuadas. Por fim, serão incluídas as reflexões finais sobre as impressões gerais da experiência de estágio.

2 OBJETIVOS

2.1 Objetivo geral

Complementar as habilidades adquiridas pelo aluno em sua graduação por meio de uma experiência prática na área de desenvolvimento de *software*, participando de ciclos de desenvolvimento, colaborações em equipe e resolução de demandas em uma empresa de tecnologia.

2.2 Objetivos específicos

- Participar de todos os ciclos de desenvolvimento de aplicativos;
- Aplicar de forma prática os conhecimentos adquiridos na graduação;
- Desenvolver habilidades e atitudes necessárias para aquisição de competências profissionais;
- Auxiliar no aprendizado com outros desenvolvedores de alto desempenho para um ambiente colaborativo.

3 A EMPRESA

O objetivo desta seção é apresentar algumas informações gerais da empresa onde o estágio foi realizado, denotando os projetos principais e o mercado onde ela está inserida. Vide figura 01 para visualizar o logo da empresa Akross.

Figura 01 – Logo Akross



Fonte: Site Oficial Akross¹

3.1 Origem

A Akross é uma empresa de tecnologia fundada em 2018, no Rio de Janeiro/RJ, como *spin-off* da Mobicare, que estava ativa no mercado desde 2010, desenvolvendo soluções de WiFi, autoatendimento e gestão de serviços para grandes empresas. A Akross foi criada para assumir alguns projetos existentes da Mobicare e criar novos produtos, como o *Profit Place*, a maior plataforma de gestão e entrega de serviços digitais e serviços por assinatura da América Latina (Akross, 2024). Eventualmente, devido a alterações estratégicas, a Mobicare foi unificada à Akross, que assumiu todos os seus projetos. A empresa possui sua sede localizada na Av. Barão de Tefé, 27 - 12º andar - Saúde, Rio de Janeiro - RJ, 20220-460.

3.2 Perfil da empresa

A Akross se autodenomina como uma *Tech Boutique* por atender diferentes áreas do mercado de tecnologia com seus produtos e projetos oferecidos (Akross, 2024). Seus maiores clientes são empresas de telecomunicação, com o objetivo de facilitar tanto processos internos para os colaboradores, quanto externos para o cliente final. A maioria dos colaboradores trabalha no modelo *home office*, mas existem escritórios no Rio de Janeiro e São Paulo, onde ficam a maioria dos gerentes de negócios e projetos para realizar reuniões presenciais com os clientes.

¹ Disponível em: <https://akross.com.br/pt/>. Acesso em: 11 set. 2024.

3.3 Estrutura organizacional

A Akross, em sua estrutura organizacional, opera usando a metodologia de gerenciamento ágil *Scrum*, que descreve um conjunto de reuniões, ferramentas e papéis que funcionam juntos para que as equipes organizem e gerenciem o próprio trabalho (Drumond, s.d.). Geralmente, as entregas do *Scrum* ocorrem em sessões de trabalho que duram duas semanas, chamadas *Sprints*. Antes do início de cada *Sprint*, são definidas as tarefas a serem realizadas e os artefatos que devem ser entregues no fim do período. Durante o decorrer da *Sprint*, são realizados encontros diários chamados de *Daily*, onde a equipe pode comunicar seu progresso e apontar possíveis impedimentos. No final da *Sprint*, é realizado um evento chamado de *Retrospective*, onde a equipe tem a oportunidade de refletir sobre a *Sprint* passada e propor melhorias para *Sprints* futuras.

Na escala de projetos, a empresa trabalha com uma divisão em times de atuação separados por projetos, chamadas *SQUADs*. Cada *SQUAD* é composta geralmente por um time de desenvolvimento, um time focado em *Quality Assurance*, uma equipe de *DevOps*, um *Scrum Master* e um *Product Owner*. A empresa utiliza a ferramenta *Slack* para comunicação interna entre os membros da equipe, o *Gmail* para contato entre os colaboradores, com fornecedores, clientes e outros públicos, *Google Forms* para realizar pesquisas rápidas, *Qulture Rocks* para acompanhamento do clima da empresa e feedbacks, o *Google Meet* para reuniões e o *Nuclino* para organizar e compartilhar conhecimento em equipe.

O discente atuou em dois projetos diferentes durante o período de estágio: o projeto *Nomad Business*, uma solução de Wi-Fi que rentabiliza redes de forma segura e eficaz, através de portais cativos e rede *community*; e um projeto pertencente a uma grande empresa cliente de telecomunicação, um aplicativo *mobile* que permite que seus clientes gerenciem seus contratos de Internet Banda Larga e realizem autoatendimento sobre questões técnicas, financeiras e contratuais. Estes projetos e a participação do discente em cada um serão abordadas nos seguintes tópicos.

4 RELATO DE ESTÁGIO

Os capítulos a seguir têm como objetivo descrever a experiência do discente, durante seu estágio na Akross, abrangendo alguns aspectos como o processo seletivo, o *Onboarding* e o exercício do discente nos projetos mencionados anteriormente. Serão também apresentadas, de maneira breve, algumas das ferramentas e tecnologias utilizadas pelo discente, ao longo de sua atuação.

4.1 Primeiro contato

A vaga de estágio foi indicada ao aluno por um de seus colegas de faculdade que trabalhava na antiga Mobicare e também foi divulgada no grupo de e-mails do curso de Análise e Desenvolvimento de Sistemas do IFPB Campus Monteiro pelo então coordenador do curso. A vaga e o processo seletivo foram gerenciados pela plataforma *Gupy*, que atua como intermediária entre os candidatos e as empresas em busca de estagiários, onde também foram apresentados mais detalhes sobre a vaga, como as responsabilidades do candidato contemplado e o modelo de trabalho, que seria completamente remoto.

4.2 Processo seletivo

Pouco mais de uma semana após a inscrição do discente, a empresa enviou um *e-mail* detalhando a primeira etapa do processo seletivo, que se tratava de uma reunião com uma representante do RH, com o objetivo de apresentar alguns detalhes da empresa como: o mercado de atuação, uma breve descrição de alguns projetos e também conhecer as habilidades e competências do discente.

A segunda etapa envolveu duas provas que testaram o conhecimento do discente em SQL, em relações e estruturas de Bancos de Dados relacionais e na linguagem de programação Java. Por fim, a última etapa se deu em uma entrevista individual com o líder técnico principal da empresa, onde foram feitas perguntas mais específicas sobre a linguagem Java.

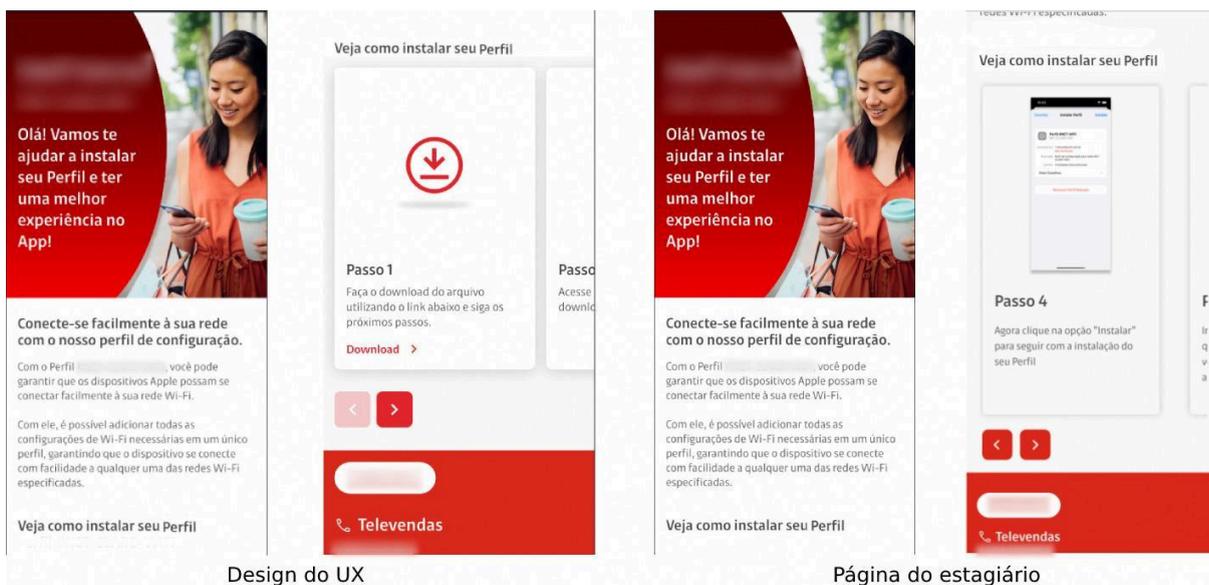
Passado um tempo após a última etapa, a Akross entrou em contato para comunicar que o discente havia sido selecionado para a vaga, e que o estágio seria iniciado uma semana após o envio dos documentos de admissão.

4.3 Onboarding e a primeira atividade

O discente iniciou suas atividades no dia 14 de Fevereiro de 2023, quando a representante de RH lhe enviou um e-mail de boas-vindas com informações essenciais para o primeiro dia de estágio, incluindo o horário de trabalho e a senha de sua conta corporativa. No primeiro dia, o estagiário participou de uma breve sessão de orientação sobre as políticas da empresa, os benefícios do colaborador e as expectativas de performance dos estagiários. A equipe de operações também forneceu ao discente um laptop pré-configurado com algumas ferramentas de gerenciamento e monitoramento da empresa, que foi enviado para sua casa logo após sua admissão.

Inicialmente, o discente não foi designado a nenhum projeto, e passou um tempo ocioso. Alguns dias após a admissão, o gerente de projetos entrou em contato com o estagiário para lhe atribuir sua primeira demanda: criar uma *landing page* destinada a instruir clientes de um determinado provedor de serviços de internet que possuem celulares da Apple a instalar um perfil de configuração necessário para se conectar ao serviço de Wi-Fi. É importante frisar que o discente possuía pouquíssima experiência em desenvolvimento *frontend* no momento, e o fato de sua primeira tarefa ser justamente focada em *frontend* o deixou apreensivo. Apesar disso, o discente aceitou a demanda e começou a configurar sua máquina com as ferramentas necessárias para desenvolvimento em React, que foi o *framework* de sua escolha por ter alguma familiaridade devido a experiências obtidas nas disciplinas de Desenvolvimento de Aplicações Corporativas e Projeto 1.

Logo após o aceite, o *UX Designer* enviou ao estagiário o design da *landing page* criado na ferramenta Figma, e o seu objetivo era recriar o design em código, de forma que o produto final ficasse o mais próximo possível do design. O discente pesquisou cursos e tutoriais *online* para se familiarizar melhor com o *framework* React para então começar a implementação do código da página. Cerca de uma semana depois, o estagiário chegou a um resultado similar ao design fornecido (Figura 02), o que foi sinônimo de aprendizado com sua tentativa e erro durante o desenvolvimento. Isso resultou, por fim, em um produto funcional e que proporciona uma excelente experiência ao usuário.

Figura 02 – Resultado da *Landing Page*

Fonte: Autoria própria

Após finalizar o código e realizar alguns ajustes finais, o estagiário apresentou o resultado final ao *UX Designer*, que aprovou e elogiou seu trabalho. Logo em seguida, o estagiário foi introduzido a um integrante da equipe de *DevOps* para que os dois preparassem a página para ser disponibilizada publicamente. Alguns membros da equipe de desenvolvimento visitaram o site e identificaram pequenos problemas no design responsivo da página, como textos desalinhados em celulares com telas menores. O estagiário corrigiu estes problemas e recebeu auxílio do membro de *DevOps* para subir a nova versão. Posteriormente, o gerente de projetos parabenizou o estagiário por seu empenho e dedicação, informando que entraria em contato assim que houvesse uma vaga disponível em algum projeto.

4.4 Vivência na SQUAD do projeto Nomad Business

A intenção dos próximos capítulos é relatar o aprendizado adquirido pelo discente operando na *SQUAD* do projeto Nomad Business, apresentando o projeto em si, as atuações feitas pelo discente em *backend* e alguns episódios excepcionais que testaram e ampliaram seu conhecimento.

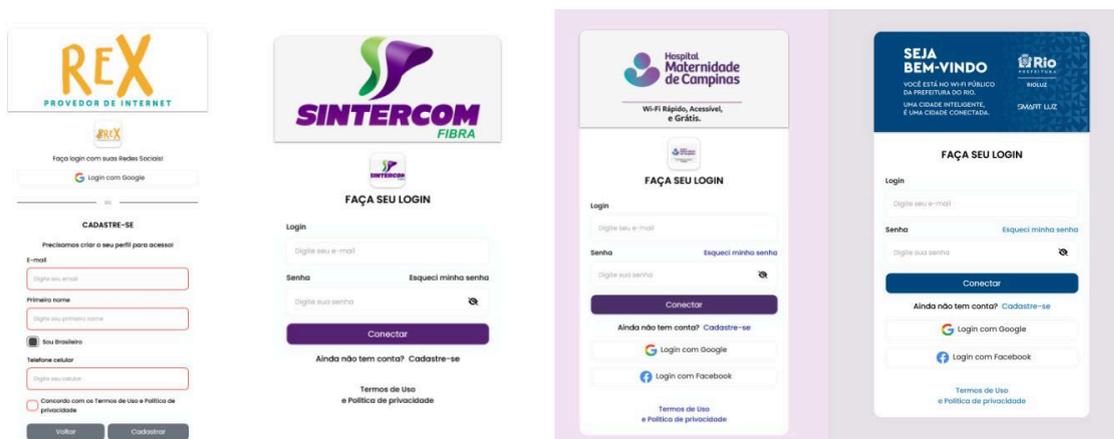
4.4.1 Sobre o projeto

O projeto Nomad Business oferece uma solução *B2C* de portais cativos para fornecedores de equipamentos Wi-Fi, permitindo que eles vendam a plataforma de gerenciamento de acesso junto com seus aparelhos, atendendo às necessidades de indivíduos e empresas que buscam gerenciamento de acesso em seus pontos de Wi-Fi. A solução atende diversos clientes, comportando uma média de 15 mil acessos diários. Este foi o primeiro projeto onde o estagiário atuou durante seu período na empresa.

A arquitetura do projeto consiste em várias tecnologias para garantir uma operação transparente entre o usuário e as controladoras de Wi-Fi. As controladoras são instaladas junto com o roteador no ponto de acesso e têm a função de gerenciar e acompanhar a sessão de Wi-Fi de cada usuário, enviando solicitações de acesso à rede para o portal cativo e transmitindo diferentes sinais que refletem o status atual da sessão do cliente.

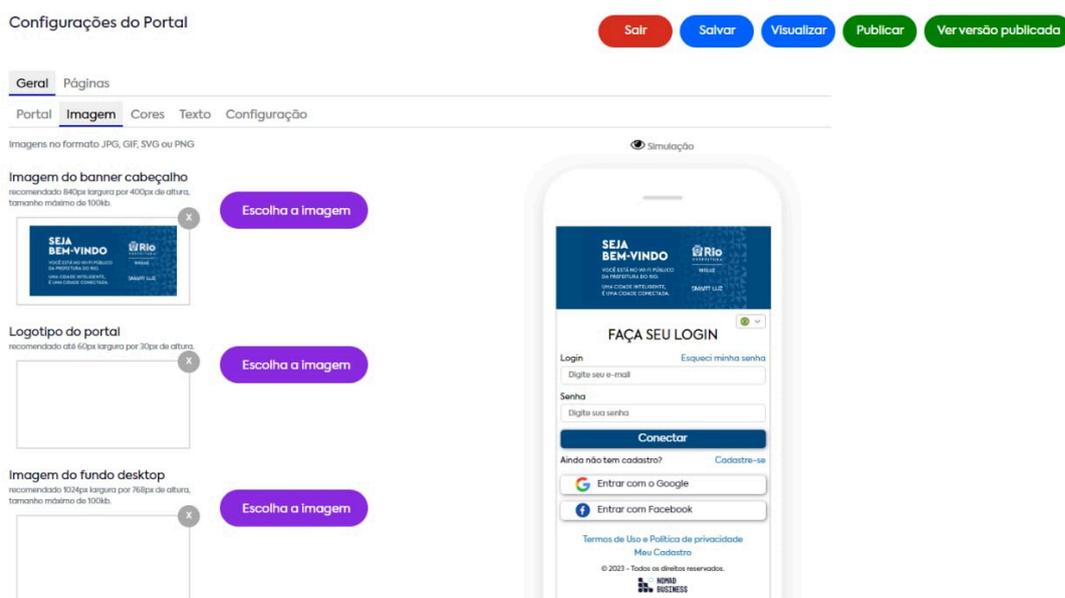
O fluxo do projeto se inicia no *frontend* com o portal cativo desenvolvido em *React*, proporcionando uma interface de usuário dinâmica que se adapta com base nas configurações definidas na plataforma de gerenciamento do portal. Essa flexibilidade permite que os clientes personalizem o *layout* e os recursos disponíveis em seus portais, como o logotipo e o banner da empresa, cores do portal, logins sociais e os dados de cadastro necessários, para atender às suas necessidades específicas. Para alguns exemplos de diferentes designs de portais cativos e da interface de gerenciamento e customização de portais utilizada pelos clientes, vide figuras 3 e 4 respectivamente.

Figura 03 – Quatro designs únicos de portais cativos de clientes diferentes



Fonte: Compilação do autor²

Figura 04 – Interface de customização e gerenciamento do portal cativo



Fonte: Interface de gerenciamento *Nomad Business*

A estrutura do *backend* inicia com um *API Gateway* que atua como o principal ponto de entrada para solicitações feitas ao *backend*. O *API Gateway* injeta um identificador único em cada solicitação para facilitar o rastreamento, e as encaminha para o microsserviço apropriado. Um Banco de Dados *Redis* serve como *cache* e repositório para manter informações das sessões ativas, enquanto um Banco de

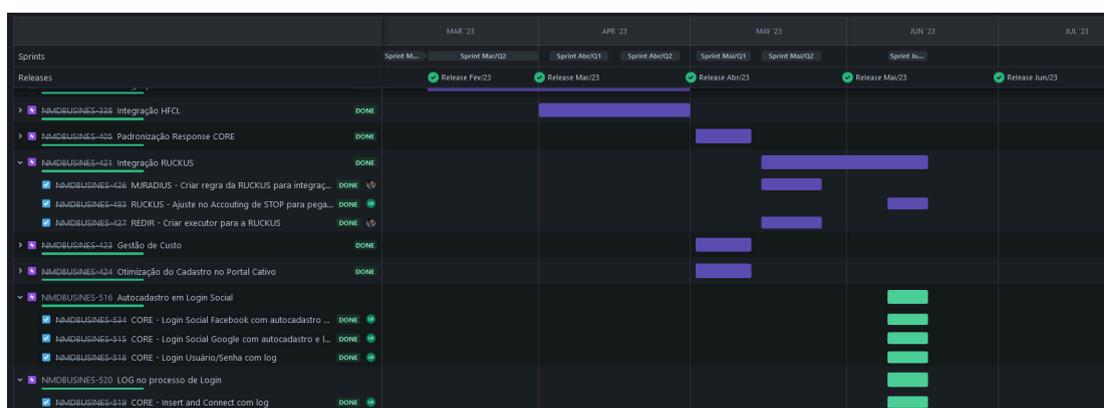
² Montagem com imagens de portais cativos de diferentes clientes, cada um com seu respectivo design. Da esquerda para a direita: REX Provedor de Internet, Sintercom Fibra, Hospital Maternidade de Campinas e Prefeitura do Rio de Janeiro

Dados relacionais armazenam outros dados importantes, como usuários cadastrados, customizações feitas pelos clientes em seus portais, controladoras Wi-Fi registradas, entre outros.

O *backend* em si é composto por alguns microsserviços escritos com a linguagem *Java* utilizando o framework *Spring*. Tudo se inicia com uma biblioteca desenvolvida internamente que recebe e decodifica as solicitações enviadas pelas controladoras. Em seguida, um microsserviço chamado *MJRadius* trata essas solicitações e gerencia os dados da sessão do usuário no *Redis*. Outros microsserviços do *backend* incluem o *Core*, responsável pelo login, cadastro de usuários e gerenciamento dos portais cativos, e o *Redir*, que monta *URLs* de redirecionamento final para diferentes marcas de controladoras para conceder acesso à Internet ao usuário.

A metodologia de desenvolvimento ágil usada no projeto é o *Scrum*. Novos recursos (denominados de *Features*), problemas a serem corrigidos e suas respectivas tarefas são gerenciados pelo *Jira* (Figura 05), uma plataforma para acompanhamento de problemas e demandas, que ajuda as equipes a planejar, acompanhar e gerenciar seus trabalhos e projetos. Cada *Feature* individual tem um identificador, uma descrição do problema ou do recurso, critérios de aceitação e uma possível implementação de código escrita pelo *Product Owner/Tech Leader* da *SQUAD*. Em seguida, as *Features* são priorizadas e recebem uma estimativa de tempo dos desenvolvedores durante a *Sprint Planning* usando pontos denominados pela sequência de *Fibonacci*, variando de 1, que geralmente é um dia, a 13, que é a *Sprint* inteira.

Figura 05 – Timeline de Features e subtarefas no Jira



Fonte: Jira Akross

Depois de priorizadas e estimadas, as *Features* são divididas em subtarefas menores, se necessário. As *Features* e suas subtarefas são então atribuídas a desenvolvedores individuais com base em suas habilidades, disponibilidade e na carga de trabalho. Cada desenvolvedor é responsável por concluir uma ou mais subtarefas ou uma *Feature* inteira dentro do prazo da *Sprint*. Quando um desenvolvedor conclui o trabalho que lhe foi atribuído, ele envia uma solicitação de revisão de código, na qual outro membro da *SQUAD* revisará o código para verificar a qualidade, *bugs*, adesão aos padrões de código e conformidade com os critérios de aceitação.

Após a conclusão do desenvolvimento, o *Product Owner/Tech Leader* testa o resultado final em um ambiente de desenvolvimento controlado e informa quaisquer problemas ou ajustes necessários ao desenvolvedor responsável. O desenvolvedor então corrige os problemas encontrados e reenvia o código para novos testes. Assim que todos os critérios de aceitação são totalmente atendidos, a *Feature* ou a subtarefa é marcada como "Concluída" no *Jira*. A cada duas semanas, as *Features* marcadas como concluídas são designadas para uma *GMUD*, uma abordagem sistemática para implementar, controlar e acompanhar as alterações feitas em um sistema de *software* para garantir a estabilidade e a melhoria contínua. O código de todas as *Features* é então reunido, liberado para o ambiente de produção e disponibilizado para os usuários finais.

4.4.2 Atuações em backend

Os seguintes capítulos descrevem algumas atuações do discente nos sistemas de *backend* do projeto, indo de uma simples contagem de usuários ativos até uma refatoração completa de uma regra de negócio.

4.4.2.1 Configuração multilíngue dos portais

A primeira tarefa do estagiário envolveu a criação de um novo *endpoint* que atualizava as configurações de um denominado portal cativo do sistema acessado por um usuário, e gerava os textos da tela em todas as três linguagens disponíveis: Português, Inglês e Espanhol. De acordo com a solução dada pelo *Product Owner/Tech Leader*, já existia um *endpoint* responsável por gerar esta configuração para uma linguagem, e a solução se dava em chamar este *endpoint* uma vez para cada linguagem. Ao finalizar as chamadas, o *backend* deveria retornar um código *HTTP* 204 se pelo menos uma das chamadas concluíssem sem erro, ou um código *HTTP* 400 caso todas as três falhassem.

Seguindo então a solução dada pelo *Product Owner/Tech Leader*, o estagiário criou o novo *endpoint* que realizava as três chamadas para gerar os arquivos de configuração, reaproveitando a implementação do *endpoint* existente (Figura 06).

Figura 06 – Pseudocódigo do Portal Config multilíngue

```
HttpStatus responseStatus = HttpStatus.NO_CONTENT;

Stream<String> languages = Stream.of("pt", "en", "es");
List<HttpStatus> statusCodes = languages.map(lang ->
    generatePortalConfig(portalId, lang)
).toList()

if (statusCodes.stream().allMatch(HttpStatus.BAD_REQUEST::equals) {
    responseStatus = HttpStatus.BAD_REQUEST;
}
```

Fonte: Autoria própria

4.4.2.2 Mensagens de erro internacionalizadas

A página *web* dos portais cativos possui uma caixa de seleção que permite que o usuário escolha a linguagem em que os textos da página são exibidos, entre Português, Inglês e Espanhol. Esta configuração, porém, não era totalmente obedecida pelo *backend*, que sempre retornava suas mensagens de erro em Português, visto que o *frontend* não comunicava ao *backend* qual linguagem o usuário escolheu. Com isso, observou-se a necessidade de criar mensagens de erro no *backend* em todas as três línguas disponíveis, além de implementar uma forma de receber a linguagem escolhida pelo usuário para retornar a mensagem apropriada.

No caso desta *Feature*, a solução foi sugerida pelo próprio estagiário e refinada pelo *Product Owner/Tech Leader*, resumindo-se então às seguintes tarefas:

1. Salvar todas as mensagens de erro no Banco de Dados com todas as três linguagens, juntamente com um identificador único para cada erro e o código *HTTP* a ser retornado;
2. Padronizar todos os retornos de erro do *backend* para utilizarem esse novo formato de mensagem, levando em conta o identificador único de cada erro;
3. Adaptar o *frontend* para enviar um novo *Request Header* em todas as requisições ao *backend* indicando a linguagem escolhida pelo usuário; e
4. Implementar uma tratativa que obtenha e retorne a mensagem correspondente ao erro ocorrido e a linguagem escolhida do Banco de Dados, mantendo o resultado dessa consulta em *cache* no *Redis*.

O primeiro esforço do estagiário nessa tarefa se deu em vasculhar o *backend*, documentar todas as mensagens e diferentes tipos de erro que podem ser retornados ao *frontend* e listá-los em uma tabela do *Excel* (demonstrada na Figura 07) para que o *Product Owner/Tech Leader* pudesse padronizá-las no Banco de Dados e atribuir um identificador único para cada mensagem.

Figura 07 – Tabela de mensagens internacionalizadas

A	B	C	D	E
Código de retorno	Status Code	Mensagem PT	Mensagem ES	Mensagem EN
GENERAL_ERROR	500	Não foi possível realizar sua solicitação!	¡Su solicitud no pudo ser cumplida!	Your request could not be fulfilled!
SESSION_NOT_FOUND	404	Sessão não encontrada!	¡Sesión no encontrada!	Session not found!
USER_NOT_FOUND	404	Usuário não encontrado!	¡Usuario no encontrado!	User not found!
USER_EXISTS	400	Usuário já existe, favor informar outro e-mail	El usuario ya existe, por favor ingrese	User already exists, please enter another email
INVALID_CODE	401	O código de verificação está inválido!	¡El código de verificación no es válido!	The verification code is invalid!
INVALID_EMAIL	400	Email não encontrado, verifique se foi digitado corretamente.	Correo electrónico no encontrado, verifique si fue ingresado correctamente.	Email not found, please check if it was typed correctly.
INVALID_ANSWER	400	Resposta da frase secreta incorreta, tente novamente.	Respuesta de frase de contraseña incorrecta, por favor intente nuevamente.	Incorrect passphrase answer, please try again.
INVALID_PASSWORD	400	A nova senha e a confirmação da nova senha não são iguais.	La nueva contraseña y la confirmación de la nueva contraseña no coinciden.	New password and new password confirmation are not the same.
PORTAL_NOT_FOUND	400	Portal indisponível no momento	Portal actualmente no disponible	Captive Portal currently unavailable

Fonte: Autoria própria

Após todas as mensagens serem mapeadas e criadas no Banco de Dados, o estagiário criou uma entidade de persistência no código utilizando o framework *JPA*, que facilita criar e ler objetos *Java* em um Banco de Dados com o mínimo de intervenção necessária do programador. Cada entidade possui os atributos necessários: um identificador, o código *HTTP* e as três mensagens.

O próximo passo se deu em refatorar todos os pontos do código do *backend* que podem jogar uma *Exception* (que, em programação, simboliza um erro) e substituí-las por uma nova *Exception* que contém o identificador correspondente ao erro ocorrido. Por exemplo: em todos os pontos do código que podem retornar um erro quando uma sessão no portal cativo não for encontrada, substituir por uma *Exception* que contém o identificador correspondente ao erro “Sessão não encontrada” (Figura 08).

Figura 08 – Exemplo de *Exception* padronizada

```

public class SessionNotFoundException extends BaseException {
    public SessionNotFoundException() { super(ReturnCode.SESSION_NOT_FOUND); }
}

```

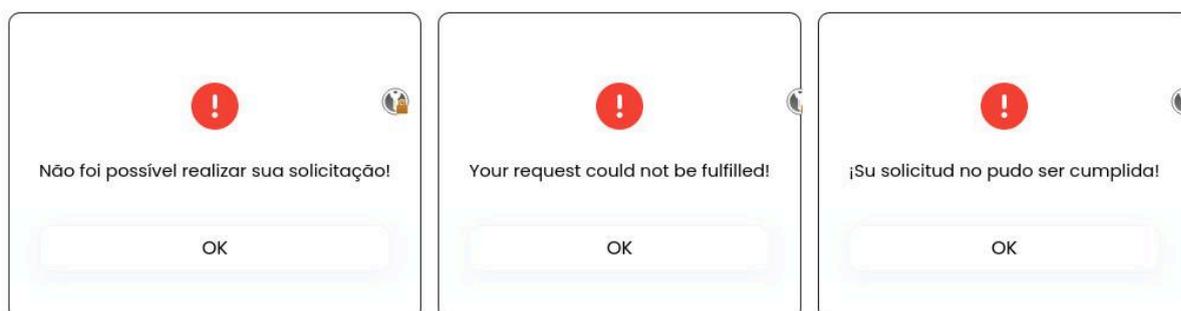
Fonte: Autoria própria

A última parte da implementação do lado do *backend* constituiu em criar um tratamento global que captura todas estas *Exceptions* padronizadas, obtém o identificador da *Exception*, busca a mensagem de erro correspondente no Banco de Dados e a retorna ao *frontend*. Para isso, o estagiário utilizou um novo *Exception*

Handler, que captura erros no código para que o programador crie um tratamento customizado. Foram criados também métodos para obter as mensagens do Banco de Dados utilizando o identificador da *Exception* lançada e filtrar pela mensagem correspondente à língua recebida no *Request Header*.

Com a implementação do *backend* pronta, o desenvolvedor de *frontend* da equipe ficou encarregado de adaptar os componentes do código para exibir as mensagens de erro utilizando o novo padrão de resposta do *backend* e implementar a lógica que envia a linguagem escolhida pelo usuário em todas as requisições com um novo *Request Header* chamado 'lng'. Com isso, as mensagens de erro do portal cativo sempre serão exibidas na linguagem que o usuário escolheu, como no exemplo da Figura 09.

Figura 09 – Mensagens de erro internacionalizadas



Fonte: Autoria própria

4.4.2.3 Refatoração do serviço de portais

No projeto Nomad Business, são aceitas solicitações de diversas controladoras Wi-Fi de diferentes marcas, cada uma usando nomes distintos para alguns de seus atributos. Por exemplo, uma controladora pode conter o seu endereço *MAC* em um atributo com nome de "sysname", enquanto outra pode usar um atributo de nome "ap_mac". Portanto, é necessário desenvolver diferentes versões do serviço de redirecionamento do *Redir* que utilizem os atributos apropriados para cada marca de controladora.

Durante a análise do código, enquanto o estagiário trabalhava na implementação da funcionalidade de mensagens de erro padronizadas mencionada anteriormente, ele notou que as implementações dos serviços de todas as

controladoras (com exceção de uma) continham uma quantidade considerável de código repetido. A única variação estava em algumas poucas linhas para atender aos diferentes atributos de cada controladora. Das mais de 200 linhas presentes em cada serviço, aproximadamente 6 eram distintas, enquanto o restante consistia em lógica duplicada, o que impactava significativamente a compreensão e a manutenção do código.

O estagiário apresentou este ponto de melhoria para o *Product Owner/Tech Leader*, que resolveu criar uma tarefa para realizar essa refatoração. O estagiário então refatorou os serviços, movendo a lógica duplicada para a classe mãe, e garantiu que o código continuava operando corretamente, porém agora com a legibilidade e manutenção muito melhoradas. Para visualizar um exemplo do código final de um dos serviços depois da refatoração, vide Figura 10.

Figura 10 – Resultado da refatoração do *PortalService*

```
if (!queryParams.containsKey("██████████")) {  
    throw new MissingResourceException("██████████ not received",  
    }  
  
if (!queryParams.containsKey("██████████")) {  
    throw new MissingResourceException("██████████ not received", "  
    }
```

Fonte: Autoria própria

4.4.3 Reunião de investigação técnica em Inglês

Durante uma fase da Sprint, o *Product Owner/Tech Leader* identificou um problema nos portais cativos que utilizavam controladoras de uma determinada marca. Qualquer usuário que tentasse acessar uma rede com uma controladora desta marca se deparava com um erro de *timeout*, já que a controladora não conseguia liberar o acesso à Internet. O *Product Owner/Tech Leader* examinou o ambiente, os logs e as configurações, mas não conseguiu identificar qualquer problema técnico do lado da Akross.

O *Product Owner/Tech Leader* resolveu agendar uma chamada com a equipe de suporte da fornecedora da controladora para investigar e tentar solucionar o problema. Contudo, isso apresentou um novo obstáculo: ele não tinha fluência em

Inglês, e toda a comunicação com a equipe de suporte deveria ser feita em Inglês. Normalmente, outro desenvolvedor fluente o apoiaria nessas conversas, mas ele havia deixado a empresa recentemente e sugeriu que o estagiário assumisse essa função. O estagiário se sentiu um pouco ansioso por ter que traduzir uma discussão técnica tão complexa em tempo real, mas mesmo assim aceitou o desafio.

Na data agendada, o estagiário e o *Product Owner/Tech Leader* entraram em uma chamada no *Microsoft Teams* juntamente com dois engenheiros de suporte. O estagiário se esforçou ao máximo para traduzir o que estava sendo discutido e apresentar a questão do erro de timeout. Após discutir possíveis pontos de falha e realizar algumas alterações nas configurações por aproximadamente uma hora, os engenheiros da equipe de suporte conseguiram identificar e resolver o problema, que realmente era originado do lado deles. Os portais com controladoras daquela marca imediatamente restabeleceram o acesso à Internet sem apresentar erros.

Ver o problema sendo resolvido foi gratificante. Contudo, o que mais se destacou foi o orgulho do estagiário, pois suas competências em Inglês foram fundamentais para superar a barreira da comunicação, possibilitando uma colaboração eficaz entre o *Product Owner/Tech Leader* e a equipe de suporte técnico da fornecedora. Essa experiência trouxe um aprendizado significativo que desafiou as habilidades do estagiário.

4.4.4 Descoberta de falha nas regras de negócio

Como mencionado anteriormente, o *backend* possui diferentes versões do serviço de redirecionamento implementadas para cada marca de controladora Wi-Fi, devido a diferenças nos nomes dos atributos e em outras lógicas no código. Para identificar qual marca de controladora fez uma solicitação, o projeto realizava um processo de validação que se baseava na existência de um determinado atributo na solicitação vinda da controladora. Nesse procedimento, o atributo era decodificado pelo *backend* e utilizado para reconhecer a marca da controladora.

O estagiário percebeu, no entanto, que este atributo não é obrigatório na especificação do protocolo usado pelas controladoras, o que implica que alguns modelos podem não incluí-lo nas solicitações. Quando esse atributo estava ausente, o método de decodificação do *backend* desconsiderava todas as validações vinculadas a ele, fazendo com que o código assumisse a marca da primeira

controladora registrada no sistema. Assim, o *backend* criava um serviço de redirecionamento da primeira marca, tentando recuperar os atributos necessários por meio dos nomes reconhecidos dessa marca e falhando caso a controladora fosse de outro fabricante.

Para solucionar essa questão, o *Product Owner/Tech Leader* pediu ao estagiário para desenvolver um processo de validação em duas etapas. Na primeira etapa, o código tentaria realizar a validação existente por meio do atributo decodificado na solicitação. Caso haja uma correspondência, a marca da controladora é identificada. Caso contrário, o código então captura o endereço IP da controladora e realiza uma comparação com uma matriz de endereços IP conhecidos para cada marca de controladora.

O *Product Owner/Tech Leader* já havia salvo uma lista de endereços IP para cada marca de controladora no Banco de Dados, e era apenas uma questão de fazer referência a essa lista. Isso permitiu que o sistema determinasse a marca correta da controladora Wi-Fi com base no endereço IP, caso a validação por meio do atributo falhasse. As controladoras que enviavam o atributo não foram afetadas e as que não enviavam passaram a funcionar após a lista de endereços IP ser implementada.

4.5 Vivência na SQUAD do projeto Codinome Fibra

Os próximos capítulos têm como objetivo compartilhar as experiências de aprendizado vivenciadas pelo estudante enquanto atuava na *SQUAD* do projeto Codinome Fibra. Serão apresentados alguns detalhes sobre o projeto e as contribuições do discente na área de *backend*.

4.5.1 Sobre o projeto

Por volta do segundo trimestre de 2023, a Akross firmou um contrato de consultoria com uma das maiores empresas de telecomunicações do Brasil, no qual a Akross disponibilizaria profissionais para atuarem diretamente em projetos relacionados ao seu plano de Internet Banda Larga, abrangendo o desenvolvimento de aplicativos móveis e a operação do *Call Center*.

O estagiário foi designado para atuar no projeto denominado neste trabalho de Codinome Fibra, um aplicativo móvel que permite que os assinantes de Internet Banda Larga realizem pagamentos de contas, adquiram produtos adicionais (também conhecidos como *addons*), alterem a velocidade de sua conexão e solicitem assistência técnica.

O *frontend* do projeto se trata de um aplicativo móvel desenvolvido em React Native, permitindo que os desenvolvedores utilizem a mesma base de código para as plataformas Android e iOS. Na camada de negócios, se encontram dois *backends*, além de um Banco de Dados *Redis* utilizado como *cache* e um Banco de Dados Mongo que armazena parâmetros e *logs* do *backend*. Além disso, há uma série de microsserviços que suportam o *backend*, disponibilizando informações relevantes sobre os clientes, como dados pessoais, produtos contratados, contas e outras informações essenciais para o funcionamento do projeto. Para monitoramento, além dos *logs* armazenados no Mongo, o projeto utiliza também a *stack* de tecnologia Kibana, Logstash e Elasticsearch (abreviada ELK)³ para oferecer uma melhor experiência de visualização e filtragem de *logs*.

A equipe do projeto adota a metodologia ágil de desenvolvimento *Scrum*, similar a experiência anterior do estagiário no projeto Nomad Business. A principal

³ Disponível em <https://www.elastic.co/elastic-stack>. Acesso em 26 fev. 2025.

distinção é que, além das *Sprint Plannings*, também são realizadas *Program Increment Plannings* (abreviadas de *PI Plannings*). Esse processo inclui não apenas a equipe de desenvolvimento, mas também o *Product Owner* e o *Product Manager*, que colaboram na priorização de funcionalidades para as próximas *Sprints* até a próxima *PI Planning*, em vez de apenas a próxima *Sprint*. As *PI Plannings* costumam ocorrer a cada 8 a 12 semanas e têm uma duração variando de 1 a 3 dias.

4.5.2 Onboarding e insegurança inicial

Ao iniciar sua atuação na empresa proprietária do Codinome Fibra, o estagiário não pôde evitar de se sentir ansioso e um pouco preocupado. Afinal, ele estava prestes a prestar serviço para uma das maiores e mais reconhecidas empresas do Brasil, com menos de 6 meses de experiência de mercado. Para agravar a situação, a reunião inicial de *Onboarding*, onde foram apresentadas as políticas, projetos e camadas de negócios da empresa, o deixou se sentindo sobrecarregado. Como na reunião também estavam presentes pessoas que foram contratadas para trabalhar em outros projetos, a apresentação da camada de negócios incluía serviços de todos os projetos da empresa. O estagiário não se deu conta de que apenas alguns poucos daqueles serviços importariam para o projeto que ele foi designado, e acreditou que teria que participar ativamente no desenvolvimento de todas as camadas de serviços mostradas, o que gerou insegurança e dúvidas sobre sua capacidade de colaborar com a equipe.

Diante de uma situação estressante, o estagiário decidiu abordar seu Gerente de Projetos para compartilhar suas preocupações. Ele comentou que se sentia sobrecarregado e estava em dúvida se conseguiria lidar com as responsabilidades relacionadas ao projeto, e chegou a questionar a possibilidade de se afastar do projeto Codinome Fibra e explorar outras oportunidades dentro da Akross.

O Gerente de Projetos ouviu atentamente as preocupações do estagiário e ofereceu-lhe palavras encorajadoras. Ele esclareceu que não se esperava que o estagiário tivesse domínio de todos os serviços e projetos apresentados durante a reunião de *Onboarding*, assegurando que ele se concentraria apenas em um subconjunto específico de serviços que eram relevantes para o projeto que ele foi designado. O Gerente tomou também um tempo para detalhar a camada de serviços

e ressaltar as principais áreas em que o estagiário atuaria. Ele mencionou que, embora os outros serviços fossem relevantes, não eram diretamente aplicáveis ao projeto, e que o estagiário não precisava se preocupar com eles. Essa explicação contribuiu para atenuar as ansiedades do estagiário, proporcionando-lhe uma sensação de clareza e direção.

4.5.3 Atuações em backend

Os capítulos a seguir abordam diversas atividades realizadas pelo estagiário no sistema *backend* do projeto Codinome Fibra, abrangendo desde a análise de logs até a continuidade da implementação da mudança de endereço de instalação.

4.5.3.1 Estudo dos logs

O projeto Codinome Fibra conta com uma equipe de *Data Analytics* encarregada de monitorar os *logs* de transações gerados pelo projeto e de usar essas informações para desenvolver uma interface com diversas métricas, como taxa de erro por requisição, número de chamadas em cada fluxo, localização dos usuários, entre outros. A primeira tarefa do estagiário foi realizar uma análise do *backend* do projeto para compreender o código que obtém os atributos e gera os *logs*, além de sugerir quais atributos seriam valiosos para que a equipe de *Data Analytics* pudesse criar sua interface de métricas. O estagiário também ficou encarregado de identificar todos os fluxos nos quais o aplicativo poderia exibir uma tela chamada internamente de Tela de Guindaste, que aparece quando ocorre um erro em um fluxo essencial para o funcionamento do aplicativo, como no caso do Login, por exemplo.

Depois de compreender o funcionamento do processo de geração de logs e identificar atributos úteis (como o nome interno e a URL do microsserviço utilizado no fluxo, CPF/CNPJ do cliente, versão do aplicativo instalada, código HTTP do resultado da chamada, entre outros), o *Product Owner* marcou uma reunião com a equipe de *Data Analytics* para que o estagiário pudesse expor os atributos listados por ele e as formas de acessá-los para o desenvolvimento da interface. Assim, a equipe de *Data Analytics* deu início ao trabalho de elaborar um mecanismo para

extrair as informações dos *logs* do aplicativo e construir a interface de métricas no Kibana para que elas possam ser facilmente acessadas e interpretadas. Para um exemplo de duas interfaces de métricas criadas no Kibana utilizando as informações obtidas pelo estagiário, vide figura 11.

Figura 11 – Exemplos de interfaces de métricas no Kibana



Fonte: Autoria própria

4.5.3.2 Bloqueio financeiro na mudança de endereço

O projeto Codinome Fibra conta com um recurso de Mudança de Endereço, permitindo que os clientes solicitem a alteração do local de instalação de sua Internet Banda Larga, caso estejam mudando de casa, por exemplo. Entretanto, a implementação realizada pela equipe de consultoria anterior apresentava falhas e esse recurso estava desativado no ambiente de produção. Assim, a *SQUAD* do estagiário assumiu a responsabilidade de dar continuidade ao desenvolvimento e correção deste fluxo, visando torná-lo funcional.

A segunda atividade designada ao estagiário consistiu em desenvolver uma verificação do *status* financeiro da Banda Larga do cliente sempre que houvesse um pedido de alteração do endereço de instalação do produto. Para isso, o estagiário precisaria estudar o *backend* para identificar qual microsserviço fornecia a informação que indicava se o produto do cliente estava com um Bloqueio Parcial ou Bloqueio Total e, caso estejam presentes, barrar o usuário de iniciar o fluxo. Esses

status de bloqueio indicam um atraso no pagamento da fatura e podem resultar na redução da velocidade e/ou cancelamento do serviço.

Após analisar o *backend*, o estagiário identificou que o serviço responsável por fornecer dados sobre os produtos contratados por um cliente específico incluía o atributo de Bloqueio Parcial/Total relacionado ao produto de Banda Larga do cliente. Com essa informação em mente, o estagiário desenvolveu um novo *endpoint* para realizar uma consulta ao serviço identificado, filtrar o produto selecionado pelo cliente com base no ID do contrato enviado pelo *frontend*, verificar a presença do atributo que indica bloqueio e, por fim, gerar uma resposta adequada esclarecendo se é viável ou não realizar a mudança de endereço para aquele produto (Figura 12). O *frontend* então chamaria esse *endpoint* logo que o cliente solicitar uma mudança de endereço para barrar ou prosseguir com o fluxo de acordo com a resposta do *backend*.

Figura 12 – Pseudocódigo do bloqueio financeiro

```
const selectedContract = (await this.getUserContracts(userId)).find(c => c.id === contractId);

const { blockType } = selectedContract.products.find(p => p.type === ProductTypes.BANDA_LARGA);
if (blockType === BlockTypes.PARTIAL_BLOCK || blockType === BlockTypes.TOTAL_BLOCK) {
  return { isEligible: false, reason: AddressChangeEligibilityReasons.HAS_FINANCIAL_BLOCK };
}

return { isEligible: true };
```

Fonte: Autoria própria

4.5.3.3 Correção na migração de plano

Para sua próxima tarefa, o estagiário foi encarregado de alterar a quantidade máxima de dias em que um cliente inadimplente pode alterar a velocidade de seu plano de Internet após o vencimento de sua fatura. Essa mudança exigiu a alteração de um parâmetro no Banco de Dados (Figura 13), que continha um número representando a quantidade máxima de dias utilizada na regra para ser comparado a quantidade de dias decorridos desde o vencimento da conta do cliente. O estagiário atualizou esse parâmetro no Banco de Dados, possibilitando que clientes inadimplentes realizassem uma alteração de velocidade dentro do novo prazo.

Figura 13 – Variável armazenada no Banco de Dados

```
1  _id: ObjectId('')
2  name: "MAX_ALLOWED_DELAYED_DAYS"
3  value: Array (1)
4    0: Object
5      name: "CONFIG"
6      value: Object
7        daysAllowed: X
```

Fonte: Autoria própria

Essa tarefa, apesar de tecnicamente simples, proporcionou um ensinamento valioso ao estagiário sobre a importância de parametrizar variáveis no Banco de Dados, em vez de mantê-las fixas no código. Como o limite de dias era um parâmetro ajustável no Banco de Dados, a regra de negócio do sistema pôde ser alterada rapidamente e sem necessidade de uma nova implantação do *backend*. Essa abordagem permite maior flexibilidade e reduz a necessidade de reimplementações custosas e demoradas.

Ao definir parâmetros para todos os aspectos possíveis no Banco de Dados, o sistema se torna mais adaptável e mais simples de ser administrado. Essa vivência ressaltou as vantagens de projetar sistemas tendo em vista a flexibilidade e a escalabilidade, proporcionando ao estagiário uma compreensão mais aprofundada da importância da configuração centrada no Banco de Dados no processo de desenvolvimento de *software*.

4.5.3.4 Impedimentos na mudança de endereço

Continuando a implementação do recurso de Mudança de Endereço existente no projeto, novas verificações foram identificadas como necessárias para evitar que clientes iniciassem solicitações de mudança de endereço nos seguintes cenários:

- Mudança de endereço para um local com Fibra ativa;
- Solicitações de clientes com pedido de mudança de endereço em aberto.

Para implementar essas verificações, o *backend* deveria interagir com alguns microsserviços, cada um responsável pela verificação de um impedimento específico. A função do *backend* era fornecer os dados necessários ao

microserviço correspondente, que realizaria a verificação, e então retornar o resultado para o *frontend*. Além disso, para garantir a flexibilidade e facilidade de manutenção, todas as verificações relacionadas a endereços deveriam ser parametrizadas no Banco de Dados, permitindo que elas pudessem ser facilmente ligadas ou desligadas a qualquer momento.

O estagiário iniciou seu trabalho desenvolvendo as integrações com os novos microserviços, enviando o novo endereço informado pelo cliente para os serviços responsáveis por verificar as bases de Fibra ativas, além de encaminhar o CPF/CNPJ do cliente para o microserviço que analisa pedidos de mudança de endereço em aberto. Ele também implementou parâmetros para permitir a ativação e desativação de cada chamada de forma independente. Em seguida, foi preciso criar diferentes respostas para cada um dos serviços integrados, para permitir que o *frontend* apresente telas distintas conforme o cenário em que o fluxo caiu. Por exemplo, uma tela que informa: “Você já possui uma ordem de mudança de endereço em aberto” para uma situação em que o cliente possui um pedido ativo.

4.5.3.5 Mudança de endereço com migração de oferta

A implementação existente da mudança de endereço incluía uma lógica destinada a apresentar ofertas disponíveis na nova localização ao cliente, com o intuito de incentivá-lo a optar por um plano mais rápido, caso houvesse opções com velocidades maiores disponíveis. Alternativamente, a lógica deveria forçar o cliente a selecionar uma oferta inferior se seu plano atual não estivesse acessível no novo endereço e não houvessem ofertas maiores disponíveis. No entanto, assim como muitas partes da implementação existente, essa lógica estava desatualizada e não operava corretamente. A próxima tarefa do estagiário consistiu em reescrever esta parte do fluxo para que os clientes pudessem realizar a migração de suas ofertas juntamente com o pedido de mudança de endereço de instalação, reforçando as regras de negócio mencionadas anteriormente.

A documentação da tarefa indicava que o *backend* precisava consultar um microserviço chamado RTD para obter as ofertas disponíveis no novo endereço, e durante uma de suas análises para realizar outras atividades anteriores, o estagiário notou que esse microserviço já era utilizado pelo processo de migração de oferta do sistema. Essa implementação realizava o envio do endereço e do plano atual do

cliente, a separação das ofertas que eram maiores, iguais ou inferiores à oferta atual, consulta da disponibilidade de *upgrades*, entre outras lógicas que também seriam necessárias para o fluxo de mudança de endereço com migração de oferta.

Com essas informações em mente, o estagiário fez adaptações no código existente (Figura 14) para que ele pudesse receber o novo endereço escolhido pelo cliente e encaminhá-lo ao RTD. Além disso, o estagiário examinou o fluxo de migração no *frontend* e notou que também seria possível reaproveitar boa parte da implementação para o fluxo de mudança de endereço.

Figura 14 – Pseudocódigo do envio do novo endereço

```
const isForAddressChange = !!newAddress
if (isForAddressChange) {
  address = newAddress.address;
  complements = newAddress.complements;
} else {
  address = currentProduct.address;
  complements = currentProduct.address.complements;
}
```

Fonte: Autoria própria

5 CONSIDERAÇÕES FINAIS

Ao longo de sua jornada de estágio na Akross, o estagiário teve a oportunidade de vivenciar uma experiência prática na criação de soluções de *software*, elicitação de requisitos e arquitetura de projetos, aplicando todo o seu conhecimento obtido ao longo do curso de Análise e Desenvolvimento de Sistemas.

A trajetória acadêmica do discente demonstra que diversas disciplinas da grade curricular foram fundamentais para aprimorar as competências necessárias para lidar com os desafios do estágio. Disciplinas como Programação Orientada a Objetos e Banco de Dados forneceram a base teórica para o desenvolvimento de um *software* de qualidade, enquanto outras, como Processos de Desenvolvimento e Análise e Projeto, abordaram os ciclos de entrega de demandas e a elicitação de requisitos. Por fim, as disciplinas de Desenvolvimento de Aplicações Corporativas, Projeto 1 e Projeto 2 permitiram a aplicação prática desses conceitos adquiridos. Isso evidencia que o curso está bem estruturado para atender às exigências do mercado e preparar programadores altamente capacitados.

Durante o programa de estágio supervisionado, o discente se deparou com diversas atividades desafiadoras, muitas vezes sem a supervisão de profissionais mais experientes. Essas situações, embora não sejam ideais, mostraram-se extremamente proveitosas, uma vez que permitiram que o discente desenvolvesse sua autonomia e independência técnica como desenvolvedor. Contudo, é importante salientar que essa abordagem tem suas limitações, já que o estagiário era muitas vezes responsabilizado por entregar resultados que normalmente seriam esperados de um profissional mais experiente e com senioridade superior.

O período de estágio de dez meses e meio provou ser uma experiência transformadora, repleta de desafios e oportunidades de resolução de problemas que contribuíram significativamente para o crescimento profissional do estagiário. O ponto culminante desse crescimento ficou evidente quando o discente recebeu uma oferta de emprego da Akross para fazer parte do time como Desenvolvedor *Backend* Júnior, destacando as competências e a experiência que ele adquiriu durante seu estágio na empresa.

Em conclusão, sua vivência na Akross influenciou significativamente seu crescimento profissional, fornecendo-lhe as habilidades e o conhecimento necessários para prosperar no mundo dinâmico e em constante evolução do

desenvolvimento de *software*. O discente expressa sua profunda gratidão à empresa e a seus colegas de trabalho por terem proporcionado um ambiente acolhedor e colaborativo, onde ele pôde expandir seus conhecimentos e compartilhar sua paixão pela inovação. O programa de estágio não apenas atendeu como também superou suas expectativas em vários aspectos, fornecendo habilidades técnicas e valiosas lições sobre trabalho em equipe, comunicação e flexibilidade. Ao dar os primeiros passos em sua carreira, ele sem dúvida aproveitará o conhecimento, as habilidades e as experiências obtidas durante seu estágio na Akross, que servirão de alicerce para seu crescimento e sucesso contínuos.

REFERÊNCIAS

A Importância da Capacitação Profissional em TI: Um Investimento Estratégico para Empresas de Tecnologia. **VISIONNAIRE**, 01 de Outubro de 2024. Disponível em: <https://www.visionnaire.com.br/A-Importancia-da-Capacitacao-Profissional-em-TI-Um-Investimento-Estrategico-para-Empresas-11-16312.shtml>. Acesso em 08 de Nov. de 2024.

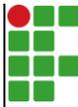
AKROSS. Akross, 2024. Página Inicial. Disponível em: <https://www.akross.com.br>. Acesso em: 7 de Set. de 2024.

COPELAND, JACK. **Alan Turing: The codebreaker who saved 'millions of lives'**
19 de Junho de 2012
Disponível em: <https://www.bbc.com/news/technology-18419691>. Acesso em 10 de Mar. de 2024.

DRUMOND, CLAIRE. **O que é scrum e como começar** S.d.
Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em 13 de Mar. de 2024.

PORTAL DO ESTUDANTE, IFPB. **Análise e Desenvolvimento de Sistemas**. S.d.
Disponível em: <https://estudante.ifpb.edu.br/cursos/13>. Acesso em 11 de Mar. de 2024.

QUEIROZ, JOÃO VICTOR LACERDA DE. **Relato de Experiência Profissional no Processo de Desenvolvimento de Software na Empresa Mobicare**.
Instituto Federal da Paraíba, Campus Monteiro, 2023.

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

Documento Digitalizado Ostensivo (Público)

Relatório de Estágio Final

Assunto:	Relatório de Estágio Final
Assinado por:	Jan Joris
Tipo do Documento:	Relatório
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Jan Joris Tomé de Lira, DISCENTE (202025020010) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - MONTEIRO**, em 14/04/2025 08:14:40.

Este documento foi armazenado no SUAP em 14/04/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1459444

Código de Autenticação: ef8c520e4f

