



DIAN LUCA FERNANDES DI GRAZIA

**RELATÓRIO DE ESTÁGIO
NA EMPRESA EQUALS S/A**

LAVRAS – MG

2022

DIAN LUCA FERNANDES DI GRAZIA

RELATÓRIO DE ESTÁGIO
NA EMPRESA EQUALS S/A

Relatório de Estágio Supervisionado
apresentado à Universidade Federal de Lavras
como parte das exigências do curso de
Sistemas de Informação, para obtenção do
título de Bacharel.

Prof. Heitor Augustus Xavier Costa
Orientador

LAVRAS - MG
2022

DIAN LUCA FERNANDES DI GRAZIA

RELATÓRIO DE ESTÁGIO
NA EMPRESA EQUALS S/A

Relatório de Estágio Supervisionado
apresentado à Universidade Federal de Lavras
como parte das exigências do curso de
Sistemas de Informação, para obtenção do
título de Bacharel.

Aprovado em 05 de Maio de 2022

Prof. Antonio Maria Pereira Resende	ICET/UFLA
Prof. Paulo Afonso Parreira Júnior	ICET/UFLA
Prof. Heitor Augustus Xavier Costa	ICET/UFLA

Prof. Heitor Augustus Xavier Costa
Orientador

LAVRAS - MG

2022

AGRADECIMENTOS

Certa vez o escritor e filósofo Ralph Waldo Emerson escreveu em um de seus livros o seguinte ensaio sobre o ato da gratidão: “Cultive o hábito de ser grato por todas as coisas boas que vierem a você e de agradecer continuamente. E porque todas as coisas contribuíram para o seu progresso, você deve incluir todas as coisas em sua gratidão.”. Não obstante a esse pensamento, irei resumir em poucos parágrafos todo o sentimento de gratidão que posso expressar pelas pessoas as quais fizeram parte dessa jornada.

Agradeço à toda a minha família, em especial meus pais, Viviane e Fernando, que foram o suporte por todo o período de vida que tive em Lavras. Agradeço também aos meus avós Fernando, Ilíria, Virgínia e Paulo que com sua experiência e sabedoria nunca deixaram de me aconselhar sobre os fenômenos da vida. Sem essas pessoas, principalmente, eu jamais teria deixado minha cidade natal rumo a outro estado e todos os relatos descritos aqui teriam sido apenas hipóteses de um futuro inexistente.

Durante a passagem por Lavras, conheci dezenas de pessoas, das mais variadas origens e sob os mais diversos contextos. Não posso deixar de agradecer todos aqueles com quem dividi um lar, um ambiente de trabalho ou mesmo momentos de lazer. Agradeço aos meus colegas de república Alysson e Deyller que foram como uma família para mim em Lavras. Agradeço também a toda a Equals e aos meus colegas de time, que sempre me prestaram suporte em cada dia que passei atuando na empresa. Meus agradecimentos ao Marcelo, que me abriu as portas da Equals, também ao Jonhy, Rodrigo, Nikolas, Igor, André e Gustavo. Agradeço também aos amigos que fiz aqui, em especial a Arthur, Bruna, Douglas, Emanuelle, Paulo, Rafael e Thiago.

Agradeço também à oportunidade e o privilégio de ter cursado minha graduação em uma das melhores universidades públicas do Brasil, a UFLA. Não apenas por seu título, mas pelas pessoas que a compõem, em especial meu agradecimento aos docentes Heitor, Renata e Juliana que dentre todos os docentes do DCC foram aqueles com quem mais tive uma aproximação e de quem mais levei puxões de orelha.

Por fim, e em especial, agradeço a Mateus e Leticia por tudo o que fizeram por mim durante minha passagem por essa cidade, vocês dois foram os maiores presentes que pude receber desde que cheguei aqui em 2017. Estarão para sempre em meu coração e memória.

LISTA DE FIGURAS

Figura 1 – Interface do produto Equals Core	12
Figura 2 – Interface do produto Raio-X App	13
Figura 3 – Interface do produto OPS-Monitor	13
Figura 4 – Modelo comparativo entre <i>Scrum</i> , <i>Kanban</i> e <i>Scrumban</i>	18
Figura 5 – Exemplo de <i>Product Backlog</i> Documentado na Plataforma Jira Software	21
Figura 6 – Exemplo de Quadro de <i>Sprint</i> Retrospective	23
Figura 7 – Exemplo de Quadro Kanban no Jira Software	26
Figura 8 – Exemplo de Construção de um Painel de BI no Metabase	27
Figura 9 – Exemplo de um Painel de BI no Metabase	28
Figura 10 – Exemplo de Documentação com Confluence	31
Figura 11 – Exemplo de Documentação de uma API com Apiary	31
Figura 12 – Exemplo de uma Automação de Processo utilizando o JMWE	33

SUMÁRIO

INTRODUÇÃO	8
SOBRE A ORGANIZAÇÃO	9
EQUALS S/A	9
ESTRUTURA ORGANIZACIONAL	10
PRODUTOS DA ORGANIZAÇÃO	11
CONCEITOS E ÁREAS DE ATUAÇÃO	15
ENGENHARIA DE REQUISITOS	15
MANIFESTO ÁGIL E O FRAMEWORK SCRUMBAN	16
O PAPEL DO ANALISTA DE NEGÓCIOS	18
PROCESSOS DE TRABALHO	19
PRODUCT DISCOVERY	20
PRODUCT BACKLOG	20
SPRINT PLANNING	21
SPRINT REVIEW	22
SPRINT RETROSPECTIVE	22
TECNOLOGIAS UTILIZADAS	24
API	24
MICROSSERVIÇOS	25
JIRA SOFTWARE	26
METABASE	27
PROJETOS DESENVOLVIDOS	29
ANÁLISE DE REQUISITOS DE SOFTWARE	29
DOCUMENTAÇÃO DE API E MICROSSERVIÇOS	30
AUTOMAÇÃO DE PROCESSOS NO JIRA SOFTWARE	32
CRIAÇÃO DE PAINÉIS DE BUSINESS INTELLIGENCE NO METABASE	33
CONSIDERAÇÕES FINAIS	35
REFERÊNCIAS	37

1 INTRODUÇÃO

Durante o curso Bacharelado em Sistemas de Informação (BSI), da Universidade Federal de Lavras (UFLA), os estudantes participam de aulas teóricas e práticas recebendo formação em múltiplas áreas relacionadas à tecnologia da informação. Os componentes curriculares abarcam áreas como lógica, algoritmos, engenharia de *software*, banco de dados, estatística, economia e administração. Os alunos formados recebem o título de bacharel em Sistemas de Informação podendo criar, contratar, projetar, desenvolver e utilizar sistemas de informação para solucionar problemas tecnológicos e administrativos.

O Componente Curricular Estágio Supervisionado/TCC no Curso de Bacharelado em Sistemas de Informação da UFLA pode ser realizado em três modalidades: Estágio em uma Organização, Projeto Acadêmico Orientado¹ ou Projeto Empreendedor². Este trabalho caracteriza-se como um Relatório de Estágio em uma Organização, o qual entende-se por “qualquer atividade desenvolvida por um discente em um ambiente de trabalho que visa à preparação para o trabalho produtivo na sua área de formação” (UFLA, 2020)³. O estágio realizado pelo autor se concretizou atuando como Analista de Negócios, entre Julho de 2019 e Setembro de 2020, na empresa Equals S/A. O recorte apresentado possui o objetivo de sintetizar o ambiente em que o trabalho foi executado, bem como as vertentes do ensino do curso de Sistemas de Informação que se conectam com as atividades práticas desenvolvidas.

Além deste capítulo, este documento está organizado como segue. O Capítulo 2 apresenta uma visão da organização em que o trabalho foi realizado. O Capítulo 3 aborda os conceitos e as respectivas áreas de atuação das atividades executadas. O Capítulo 4 descreve as tecnologias utilizadas. O Capítulo 5 dispõe os principais projetos desenvolvidos. O Capítulo 6 apresenta as considerações finais acerca das atividades desempenhadas durante o estágio.

¹ Atividade de pesquisa, de desenvolvimento técnico, de ensino ou de participação em empreendimentos e projetos de interesse social realizada pelos discentes do curso de graduação.

² Desenvolvimento de um Plano de Negócios que descreve os objetivos de um negócio na área de Sistemas de Informação e quais ações são necessárias para que esses objetivos sejam alcançados.

³ UNIVERSIDADE FEDERAL DE LAVRAS. **Resolução CBSI nº 01. 2020**. Disponível em: https://icet.ufla.br/images/Legislacao_graduacao/Sistemas_Informacao/Resolucao-CBSI-n1_17-01-2020_Estagi-o-TCC.pdf. Acesso em: 09 abr. 2022.

2 SOBRE A ORGANIZAÇÃO

Este capítulo tem como objetivo descrever a organização em que o trabalho de estágio foi realizado, mencionando uma breve contextualização de sua história, momento atual, processos organizacionais, bem como os respectivos produtos relacionados à área de atuação do autor, a área de engenharia de requisitos.

2.1 Equals S/A

A Equals é uma *fintech*, ou seja, uma empresa que utiliza-se de tecnologia aplicada ao setor financeiro, mais especificamente dentro da área de meios de pagamento. Foi fundada em 2011, na cidade de Araçatuba, interior de São Paulo, mas desde 2017 possui escritórios em Lavras-MG e São Paulo-SP. A Equals desenvolve soluções capazes de auxiliar outras empresas a fazerem a gestão financeira de forma inteligente e com o uso de tecnologia. Atualmente, a empresa conta com cerca de 200 colaboradores distribuídos por diferentes estados do Brasil, atuando de forma híbrida entre os modelos presenciais e remoto.

A proposta da empresa, desde sua criação, é reinventar a forma como cada transação eletrônica é apurada e auditada por lojas físicas e virtuais, empoderando clientes grandes e pequenos a entender suas transações e conversar de igual para igual com os bancos e adquirentes que dominam o cenário de cartão no Brasil. A missão da empresa é resolver três principais dores do mercado, sendo: garantir a gestão dos meios de pagamento, ou seja, agregar as informações de todos os meios de pagamento de uma loja em uma única plataforma; realizar a auditoria de taxas e contratos, evitando cobranças indevidas e abusivas fora dos termos de uso de “maquininhas” de cartão; promover a governança corporativa nos balanços financeiros de uma empresa, fazendo uso de automação e rastreabilidade de transações monetárias. Em números, a Equals dispõe de mais de 20 mil⁴ estabelecimentos ativos, conciliando cerca de 170 milhões de transações por mês, que equivalem a cerca de 26 bilhões de reais por meio de mais de 100 meios de pagamentos diferentes integrados.

A organização em sua vertente de tecnologia, dispõe de três grandes áreas de atuação, sendo elas: Operações, Meios de Pagamento e Desenvolvimento de Produtos de Software. O presente trabalho possui ênfase na área de Operações, tendo em vista o fato de que a atuação

⁴ Todos os valores dispostos nesta seção foram fornecidos diretamente pela empresa mediante consulta prévia à realização deste trabalho.

do autor esteve dentro deste ambiente, contando com a presença de cerca de 28 colaboradores envolvidos diretamente e onde o autor exerceu o papel de Analista de Negócios.

2.2 Estrutura Organizacional

A estrutura organizacional da Equals, dentro da área de tecnologia, é definida a partir de times, também chamadas de *squads*⁵, que possuem uma área de atuação ou produto específicos. Cada time possui uma pessoa que assume a função de líder, pelo menos, uma pessoa de negócios (comumente chamada de Analista de Negócios), duas pessoas de desenvolvimento de *software* (desenvolvedores) e, pelo menos, uma pessoa que assume a função de Analista de Qualidade de Software⁶. A quantidade de pessoas em um mesmo time varia de acordo com a necessidade, bem como podem ocorrer transições de pessoas entre diferentes times para suprir demandas específicas.

O modelo processual que a Equals adota gira em torno de um modelo híbrido entre os *frameworks* Scrum⁷ e Kanban⁸, o que popularizou dentro da empresa o termo Scrumban⁹. Contudo, vale destacar que, de maneira geral, a estrutura organizacional é aplicada a partir de processos que costumam seguir um mesmo fluxo, composto pelas seguintes etapas: Levantamento de Requisitos, Desenvolvimento de *Software* e Testes de Qualidade de Software.

A primeira etapa, denominada Levantamento de Requisitos, é conduzida em sua maior parte pelos Líderes e Analistas de Negócios dos times, podendo envolver diversas outras funções, direta ou indiretamente. Esse momento é oportuno para fortalecer a interface de comunicação entre as necessidades dos usuários e clientes e como elas podem ser supridas do ponto de vista de equilíbrio entre negócios, tecnologia e financeiro. Durante a etapa de Levantamento de Requisitos geralmente são definidas as regras de negócio e os artefatos necessários, com o objetivo de formar a documentação inicial do produto de *software*. Os frutos gerados nessa etapa estão voltados para uma listagem ordenada das demandas a serem encaminhadas para a próxima etapa, Desenvolvimento de *Software*.

⁵ Modelo organizacional que distribui os colaboradores da empresa em pequenos grupos multidisciplinares com objetivos específicos.

⁶ Profissional responsável por garantir que o *software* final desenvolvido esteja de acordo com as diretrizes de requisitos inicialmente acordadas.

⁷ Metodologia de trabalho. Cf. Sec. 3.2 desta obra.

⁸ Metodologia de trabalho. Cf. Sec. 3.2 desta obra.

⁹ Termo criado unindo as palavras Scrum e Kanban, representa a aplicação conjunta desses dois tipos de metodologias de trabalho. Cf. Cap. 3.2 desta obra.

A segunda etapa do processo refere-se ao processo de implementação das necessidades de usuários, Desenvolvimento de *Software*. A lista de demandas encaminhada para essa etapa é recebida pelo Time de Desenvolvimento e passa por uma análise técnica antes da implementação de códigos. Os times costumam realizar ciclos de duas semanas corridas ou dez dias úteis, para a lista de tarefas prioritizada ser concluída. Ao final do processo de desenvolvimento do *software*, o Time de Desenvolvimento deve encaminhar os produtos gerados para a última etapa de preparação das evoluções do *software*, para análise da qualidade do que foi desenvolvido.

A terceira grande etapa do processo consiste na avaliação realizada pelos Analistas de Qualidade de Software, responsáveis por garantir que os requisitos levantados na primeira etapa foram efetivamente implementados no *software*. A partir de técnicas de análise e de posse da lista de requisitos, a validação pode acontecer de diferentes formas, incluindo etapas automatizadas para avaliação. Ao final da análise, cada demanda pode ser definida como concluída sem problemas ou não concluída com êxito. Para o caso positivo em que tudo está correto, a demanda é liberada para seguir adiante. Para o caso negativo, ou seja, quando há algum problema na demanda, ela é retornada para a etapa anterior para ser revista pelo Time de Desenvolvimento.

Ao final da jornada, quando uma demanda de *software* está pronta para ser entregue aos clientes e usuários, ela é enviada pela liderança do time para o time técnico responsável entregar. Uma entrega consiste em disponibilizar no *software* o que foi elencado como requisito para a melhora do produto no quesito atendimento das necessidades dos clientes e usuários.

2.3 Produtos da Organização

A Equals possui dois grandes produtos vendidos para o mercado, sendo um deles categorizado como Aplicação Web¹⁰ e o outro como Aplicativo Móvel¹¹. O produto de categoria Aplicação Web é chamado de Equals Core e foi concebido para utilização de grandes empresas por meio de computadores e navegadores de Internet. Em contrapartida, o produto de categoria Aplicativo Móvel é nomeado como Raio-X e possui maior foco em micro, pequenas e médias empresas e seu uso é feito por meio de um aplicativo *mobile*

¹⁰ Uma Aplicação Web é aquela que é desenvolvida inicialmente para ser utilizada pelo navegador de um dispositivo, podendo ser um computador ou um smartphone.

¹¹ Uma Aplicação Móvel é um software desenvolvido para ser instalado em um dispositivo eletrônico móvel, como um smartphone.

instalado em *smartphones*. Além deste dois produtos, a Equals detém o desenvolvimento de um produto de uso interno voltado para o Time de Operações. Esse produto é conhecido como OPS-Monitor e possui a missão de auxiliar o dia a dia dos responsáveis por garantir que o processamento de arquivos dos clientes esteja ocorrendo conforme as normas estabelecidas. A Figura 1 mostra o produto Equals Core, a Figura 2 mostra o aplicativo Raio-X e a Figura 3 mostra o produto OPS-Monitor.

Figura 1. Interface do produto Equals Core



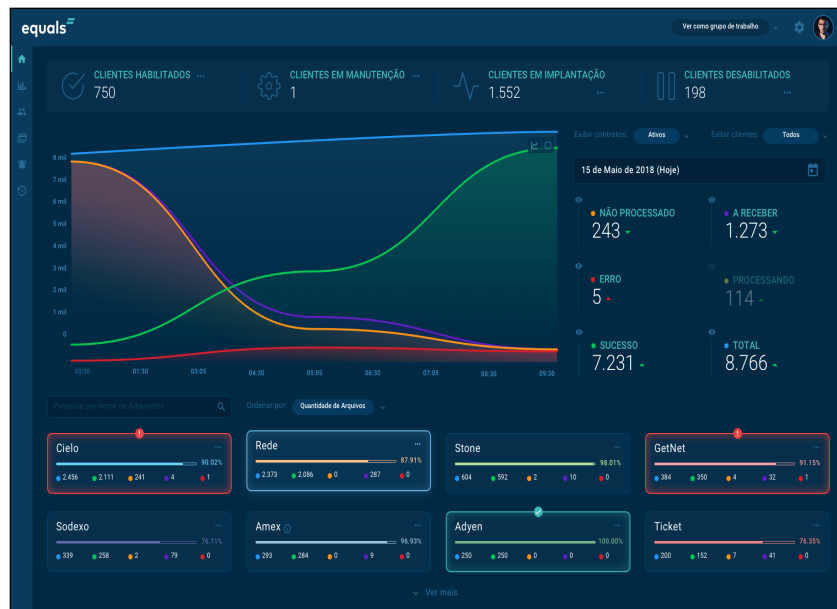
Fonte: Website da empresa Equals (2022).

Figura 2. Interface do produto Raio-X App



Fonte: Website da empresa Equals (2022).

Figura 3. Interface do produto OPS-Monitor



Fonte: Website da empresa Equals (2022).

Os três produtos que a Equals detém em seu portfólio possuem características diferentes, porém com missões muito semelhantes e alinhadas diretamente com a missão da empresa. Durante o trabalho de estágio supervisionado executado na empresa, o autor atuou diretamente com o produto OPS-Monitor, sendo o responsável pela sua parte de negócios. Contudo, assim como parte da missão do produto em si, bem como do Time de Operações em que o autor estava inserido, o trabalho executado refletiu efeitos diretos e indiretos nos demais produtos da empresa. Além disso, os itens descritos no Capítulo 4 complementam os artefatos trabalhados durante a jornada de estágio supervisionado, APIs¹² e Microsserviços¹³ desenvolvidos durante o trabalho em questão. O foco deste trabalho está voltado para o produto OPS-Monitor e as respectivas tecnologias que auxiliam no seu bom funcionamento (APIs e Microsserviços).

¹² Abreviação de *Application Programming Interface*. Cf. Cap. 4.1 desta obra.

¹³ Modelo de arquitetura de software. Cf. Cap. 4.2 desta obra.

3 CONCEITOS E ÁREAS DE ATUAÇÃO

Este capítulo busca trazer os conceitos de técnicas, metodologias e áreas de atuação dentro do contexto do Time de Operações em que o autor atuou na empresa Equals. Esses conceitos foram parte da jornada de trabalho desenvolvida e não se restringem apenas ao ambiente interno da organização, sendo aplicados em diferentes empresas do mercado de trabalho. O capítulo está subdividido em três tópicos, sendo o primeiro deles o da Engenharia de Requisitos, seguido pela explicação do conceito de Metodologia Ágil e da forma de trabalho Scrumban implementada na empresa e, por fim, como se desdobra a função do Analista de Negócios, exercida pelo autor durante o estágio supervisionado.

3.1 Engenharia de Requisitos

De acordo com a ISO/IEC/IEEE 24765 (2010), um requisito pode ser caracterizado como uma condição ou capacidade que deve ser alcançada ou possuída por um sistema, produto, serviço, resultado ou componente para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto. Requisitos incluem as necessidades quantificadas e documentadas, desejos e expectativas do patrocinador, clientes e outras partes interessadas.

Segundo Vazquez e Simões (2016), a Engenharia de Requisitos pode ser definida como uma disciplina da Engenharia de Software que consiste no uso sistemático e repetitivo de técnicas para cobrir atividades de obtenção, documentação e manutenção de um conjunto de requisitos para *software* que atendam aos objetivos de negócio. A especificação de requisitos resulta em um contrato entre clientes e Time de Desenvolvimento. Ela deve esclarecer aos clientes o que será entregue como produto do trabalho do time. Dessa forma, o principal objetivo da especificação é documentar as necessidades dos clientes e obter um aceite (aprovação) sobre o que se está propondo entregar em termos de produto. Além disso, a especificação tem por objetivo permitir que o Time de Desenvolvimento consiga compreender exatamente o que os clientes desejam. Ela é um importante instrumento de comunicação entre clientes e Time de Desenvolvimento.

O desenvolvimento de *software* é, em geral, um processo iterativo, no qual uma iteração do ciclo de desenvolvimento de *software* acontece uma ou mais vezes ao longo de cada fase do seu ciclo de vida. Para o desenvolvimento de um *software* bem sucedido, é imprescindível uma completa compreensão dos seus requisitos. Com isso, nota-se que a

melhor solução para satisfazer as necessidades do cliente e usuário final consiste em um processo sólido de engenharia de requisitos (LOPES et al., 2003).

Dentro do contexto desta obra, o termo “requisito” refere-se aos artefatos base dos projetos e produtos desenvolvidos pelo autor durante o trabalho exercido enquanto Analista de Negócios. Os requisitos descrevem, sob a ótica de diversas partes interessadas, como um produto de *software* deve se comportar, buscando a satisfação dos clientes e dos usuários da Equals. Além disso, possuem a missão de orientar o desenvolvimento do *software*.

A atuação do autor mediante a aplicação do conceito de engenharia de requisitos na empresa Equals possui caráter fundamental para o sucesso do exercício da função de Analista de Negócios, descrita na seção 3.3 deste capítulo. Esse aspecto pode ser abordado pois, segundo a própria cultura interna da empresa, não importa quão bem codificados e testados os produtos da Equals sejam feitos, se eles não atenderem às reais necessidades dos usuários finais, todo o esforço exercido foi em vão.

3.2 Manifesto Ágil e o *Framework* Scrumban

A estrutura de processos aplicada dentro do ambiente da empresa Equals se baseia no produto formado entre duas diferentes vertentes de formas de trabalho em times ágeis: o *framework* Scrum e o método Kanban. O termo “ágil” em vigor provém do chamado Manifesto Ágil (2001) e se propõe a disponibilizar valores e princípios para que times atuem em seus respectivos contextos. O Scrum e o Kanban não são implementados em sua totalidade dentro do ambiente da empresa em questão, mas adaptados da melhor forma para suprir a necessidade de cada time.

O Manifesto Ágil foi criado por um time composto de diversos autores provenientes de diferentes escolas de pensamento voltados para o desenvolvimento de *software*. Embora tenha sido criado há vinte anos da data de escrita deste trabalho, os valores e os princípios propostos perduram até os dias atuais e seguem sendo fundamentais para diferentes metodologias de trabalho usadas em empresas de tecnologia. “Indivíduos e interações mais que processos e ferramentas”, “*Software* em funcionamento mais que documentação abrangente”, “Colaboração com o cliente mais que negociação de contratos” e “Responder a mudanças mais que seguir um plano” são os valores principais propostos pelo Manifesto Ágil, os quais são melhor detalhados em outros doze pilares.

O Scrum é um *framework* ágil de gerenciamento de projetos que ajuda pessoas, times e organizações a gerar valor com de soluções adaptadas para problemas complexos e é

composto por artefatos, eventos e pelo time Scrum (SCHWABER; SUTHERLAND, 2020). O Scrum propõe a divisão em times pequenos, multifuncionais e auto-organizados. Esses times possuem papéis pré-definidos, como o *Product Owner*, *Scrum Master* e o Time de Desenvolvimento e trabalham a partir de uma lista de entregáveis pequenos e concretos, onde são estimados e priorizados. Para que esses entregáveis priorizados sejam entregues, o tempo do projeto é dividido em iterações curtas, denominadas *Sprints* com período fixo (de 1 a 4 semanas). Ao final de cada ciclo, o plano de *release* é adaptado de acordo com os *insights* obtidos com as entregas, além da melhoria de processo realizada por meio de retrospectivas (PM3, 2020).

Segundo a Agile Alliance (2020), o termo “Kanban” é de origem japonesa, possui um sentido de placa ou cartaz e deriva de raízes que se traduzem literalmente como “placa visual”. Sua aplicação dentro do contexto de desenvolvimento de *software* se traduz em um método para projetar, gerenciar e melhorar os sistemas de fluxo de trabalho. O método Kanban também permite que as organizações comecem com seu fluxo de trabalho existente e conduzam mudanças evolutivas no processo. Ele é aplicado utilizando formas visuais para elencar o trabalho em andamento, definir limites de trabalho em progresso e delimitar os momentos de início e fim de uma tarefa. Aplicando maior pragmatismo, quando o método Kanban é utilizado, o trabalho é dividido em cartões expostos em um painel o qual é fracionado em colunas nomeadas a partir dos estágios de trabalho que compõem o fluxo.

Partindo dos conceitos do Scrum e do Kanban, a Equals passou a implementar uma forma de trabalho que une o que de melhor ambos possuem para suprir as necessidades internas e externas à organização. Dessa forma, o Scrumban é um híbrido entre as melhores características entre o Scrum e o Kanban, mas utilizando o Scrum como base e complementando com o Kanban (PM3, 2020). Essa característica faz com que o time utilize alguns dos papéis e cerimônias do Scrum e a evolução do processo e cadência do Kanban, como a visualização do fluxo de trabalho e o limite de tarefas em andamento. No caso do time em que o autor atuou, o papel do *Product Owner*, proposto originalmente no Scrum, é denominado Analista de Negócios e assume as responsabilidades do *Scrum Master*. Além disso, vale destacar que o Time de Desenvolvimento é composto por Desenvolvedores de Software e Analistas de Qualidade de Software. A Figura 4 mostra um modelo comparativo entre o Scrum, o Kanban e o Scrumban.

Figura 4. Modelo comparativo entre Scrum, Kanban e Scrumban

	Scrum	Kanban	Scrumban
Papéis	Product Owner, Scrum Master e Time de Desenvolvimento	Não possui papéis	Time e os papéis de acordo com a necessidade (variáveis)
Cerimônias	Daily, sprint planning, sprint review e retrospectiva	Não possui reuniões	Daily, sprint planning, sprint review e retrospectiva (variáveis)
Fluxo	Somente o que foi planejado para a sprint	Contínuo	Contínuo e com limite de trabalho em andamento em cada etapa
Artefatos	Board, backlog e burndown	Board	Board e backlog
Estimativas	Possui	Não possui	Pode ou não possuir
Time	Multifuncional	Especializado	Pode ser multifuncional ou especializado

Fonte: Acervo do autor.

3.3 O Papel do Analista de Negócios

Tendo em vista o fato da Equals não utilizar por completo dos papéis do Scrum, o Analista de Negócios dentro do contexto a empresa é a pessoa responsável por cuidar dos interesses do negócio em consonância com os interesses de tecnologia e *stakeholders*¹⁴. Dessa forma, a função exercida exige que o profissional atue como um agente de conexão entre diferentes perspectivas, transitando entre diversas áreas da empresa e facilitando a integração da visão de todas elas. De acordo com o *International Institute of Business Analysis* (IIBA, 2017), “os analistas de negócios são responsáveis por descobrir, sintetizar e analisar informações de uma variedade de fontes dentro de uma corporação, incluindo ferramentas, processos, documentação e stakeholders”. Expandindo o conceito para a área de Análise de Negócios em si, diz-se que ela “é a prática de habilitar mudanças em uma empresa, definindo necessidades e recomendando soluções que ofereçam valor aos stakeholders” (IIBA, 2017).

Dentro da atuação de um Analista de Negócios, é possível traçar um perfil fortemente generalista com relação às diferentes abordagens que precisam assumir com diversos times da empresa. Dessa forma, é importante que um profissional dessa área seja capaz de transitar entre diferentes níveis da empresa e comunicar-se com uma linguagem única, os objetivos do negócio. Um dos pilares do trabalho de um Analista de Negócios é o processo de captação das necessidades dos usuários e a conversão de tais necessidades em requisitos de *software* para o Time de Desenvolvimento. Dessa forma, para atuar nessa função, é preciso apresentar um

¹⁴ Um grupo ou indivíduo com uma relação com a mudança, com a necessidade, ou com a solução.

perfil comunicador, negociador, saber se relacionar com diferentes públicos e, principalmente, trabalhar em time. Adotar uma postura pró-ativa na tomada de decisão consciente é uma característica importante para ter sinergia no trabalho executado, assim como exercer a capacidade de negociar prazos, escopo e responsabilidades juntamente com os times envolvidos em um produto. Acima de tudo, um profissional que atue como Analista de Negócios deve ser capaz de ter uma visão global da organização para somar esforços e atingir os objetivos do produto e da empresa. Vale destacar, ainda, que a Equals disponibilizou uma plataforma de cursos, além de mentorias e *feedbacks* contínuos com outros Analistas de Negócio mais experientes buscando auxiliar no desenvolvimento das habilidades necessárias para a atuação do autor. Ademais, durante o curso de Sistemas de Informação, algumas disciplinas cursadas pelo autor tiveram papel importante na formação teórica de base, para aplicação no dia a dia de trabalho na empresa.

Exercendo a função de Analista de Negócios dentro da Equals, o autor atuou diretamente no relacionamento com outros times da organização com o intuito de evoluir os serviços e produtos prestados por seu time. Por se tratar de um produto de uso interno, o OPS-Monitor foi um produto de constante evolução durante o estágio supervisionado do autor, buscando sempre satisfazer as necessidades do Time de Operações. A transparência e a agilidade dos recursos dispostos no produto permitiam que as ações diárias dos demais colaboradores ganhassem tração e celeridade. Com os conceitos aplicados da Engenharia de Requisitos, o autor realizou projetos de evolução do produto unindo os interesses de negócios e tecnologia sob a ótica dos interessados.

3.4 Processos de Trabalho

Nesta seção, são apresentadas as principais atividades gerais realizadas pelo autor no dia a dia de trabalho, as quais fundamentam a metodologia de trabalho utilizada. Tais atividades abrangem métodos e processos os quais a Equals utiliza para definir padrões e rituais e manter um equilíbrio entre os diferentes times que atuam na empresa. Dentre esses aspectos, destacam-se os principais ritos do Scrum executados de diferentes formas, em diferentes periodicidades durante os ciclos de trabalho realizados. Além disso, compromissos cumpridos com outros colegas de trabalho são destacados, pois foram de suma importância no processo de desenvolvimento pessoal e profissional do autor.

3.4.1 *Product Discovery*

A primeira etapa das principais tarefas exercidas pelo autor em sua atuação enquanto Analista de Negócios consistiu na realização de um processo denominado *Product Discovery*. Segundo a PM3 (2022), a missão do *Product Discovery* é descobrir o que e porquê, com base nas necessidades dos usuários. Ou seja, é investigar informações, oportunidades e soluções que são valiosas para seu público-alvo e, ao mesmo tempo, viáveis para a sua empresa. É entender profundamente o problema antes de pensar em como resolvê-lo. Dessa forma, o principal objetivo do autor era descobrir oportunidades e propor soluções capazes de entregar valor para os usuários das soluções desenvolvidas pelo Time de Desenvolvimento.

Dentro do contexto de trabalho do autor, a realização de sessões de *Product Discovery* foram fundamentais para aprendizado rápido e eficaz de oportunidades de soluções com experimentação. O principal objetivo deste tipo de ação era reduzir o risco de investimento de tempo e esforço do Time de Desenvolvimento com soluções que não atendiam às reais necessidades dos usuários do produto trabalhado. Para a execução dessa tarefa, o autor utilizou algumas etapas como entrevistas com usuários, dinâmicas de levantamento de oportunidades, testes de interface de protótipo com usuários e mapeamento de jornadas de usuário. Tais sessões de trabalho foram realizadas em conjunto com o Líder do Time e os clientes do produto, os quais eram os integrantes de outros times de Desenvolvimento da Equals. O produto final desse processo era um documento em formato diverso, contendo: as hipóteses e declarações do problema tratado; narrativas de tentativa de solução para os problemas declarados; variados cenários de implementação; protótipos de interface de *software*; requisitos de *software* e histórias de usuário.

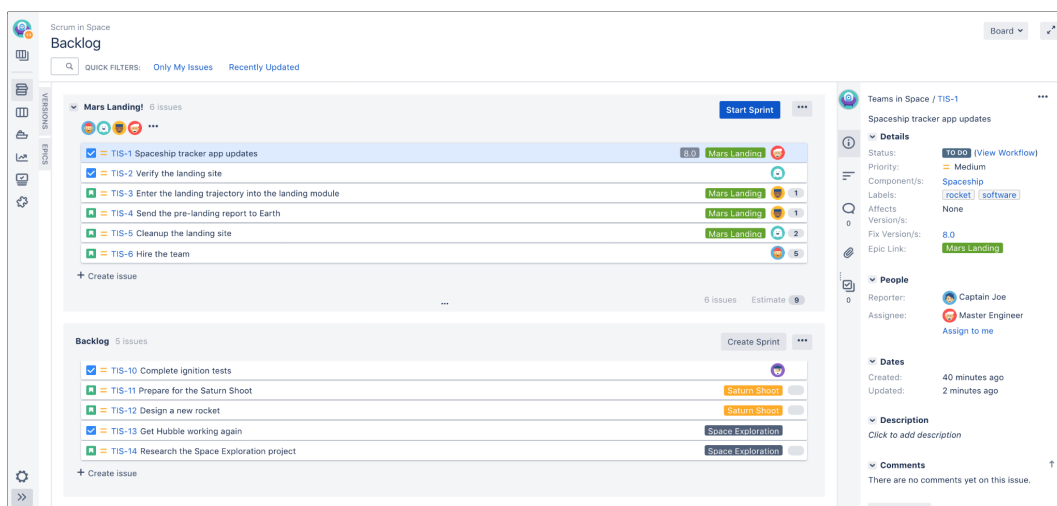
3.4.2 *Product Backlog*

Após a realização de qualquer sessão de *Product Discovery*, a sequência imediata do trabalho do autor consistia na elaboração de um *Product Backlog* a ser trabalhado pelo Time de Desenvolvimento. Segundo a Schwaber e Sutherland (2020), o *Product Backlog* é uma lista de afazeres que deve estar de posse do Time de Desenvolvimento de produtos. A partir disso, as pessoas responsáveis poderão se guiar e entender quais devem ser as prioridades no fluxo de trabalho, considerando recursos, detalhes importantes e características. É a única fonte de trabalho realizado pelo Time de Desenvolvimento.

O autor, enquanto Analista de Negócios, foi o responsável por elaborar e priorizar as atividades contidas no *Product Backlog*. Para exercer tal função, foi utilizado um método

peçoal de trabalho, que consistia em buscar equilíbrio entre três pilares de priorização, sendo: atendimento imediato aos desejos dos diversos clientes; entrega de valor para os objetivos da empresa; e critérios de entrega do maior valor no menor tempo possível. A manutenção da lista de itens contidas no *Product Backlog* foi a principal tarefa exercida pelo autor no dia a dia de trabalho na Equals, por se tratar da principal tangibilização enquanto resultado do trabalho exercido como Analista de Negócios. Para consolidar a descrição dos itens do *Product Backlog*, o autor implementou um modelo construído a partir de um consenso envolvendo o Time de Desenvolvimento e o Líder do Time, o qual continha os requisitos necessários, detalhamento técnico, critérios de aceitação e cenários de teste. A Figura 5 mostra um exemplo de como os artefatos podem ser organizados dentro de um *Product Backlog* com o uso da Plataforma Jira Software.

Figura 5. Exemplo de *Product Backlog* Documentado na Plataforma Jira Software



Fonte: Atlassian (2022).

3.4.3 *Sprint Planning*

A *Sprint Planning* é uma das cerimônias propostas pela metodologia Scrum de trabalho utilizada pelos Times de Desenvolvimento da Equals. A *Sprint Planning* inicia a *Sprint* estabelecendo o trabalho a ser executado para a *Sprint*. Esse plano resultante é criado pelo trabalho colaborativo do Time de Desenvolvimento (SCHWABER; SUTHERLAND, 2020). Essa é uma reunião onde é feita uma priorização das histórias de usuário presentes no *Product Backlog* elaborado pelo autor. Essa reunião foi importante para o bom resultado ao final da *Sprint*, pois é nela em que o autor tinha a oportunidade de transmitir para o Time de

Desenvolvimento o que havia sido descoberto durante o *Product Discovery* e iria nortear as próximas semanas de trabalho.

Ao final da *Sprint Planning* deve gerar o plano de trabalho para aquela *Sprint*, ou seja, quais atividades devem ser executadas no decorrer da *Sprint* (SCHWABER; SUTHERLAND, 2020). O principal papel do autor durante este tipo de cerimônia era facilitar o processo de leitura e entendimento dos artefatos presentes da lista de tarefas que fora priorizada anteriormente. Ao final da *Sprint Planning*, o autor detinha a tarefa de consolidar os tópicos discutidos dentro de cada item do *Product Backlog*, bem como organizá-los no quadro Kanban utilizado pelo Time de Desenvolvimento ao longo da *Sprint*. Todo esse processo era realizado em conjunto com o Líder do Time como medida de garantir um processo completo e fluído, além de auxiliar o autor em eventuais dúvidas durante o processo.

3.4.4 *Sprint Review*

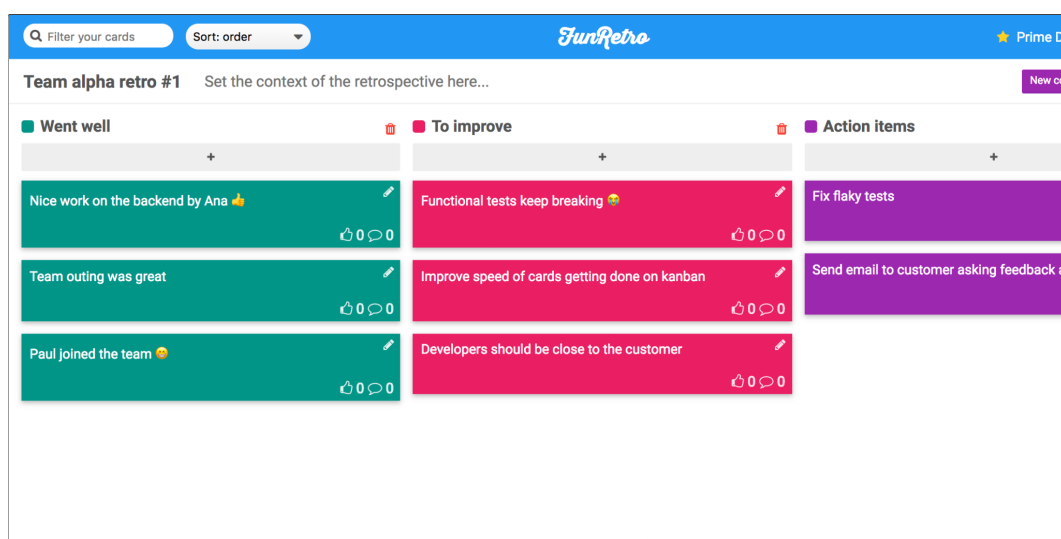
Tendo em vista que a *Sprint Planning* datava o início de uma *Sprint*, a cerimônia que representava o final do ciclo de desenvolvimento era a *Sprint Review*. De acordo com Schwaber e Sutherland (2020), o objetivo da *Sprint Review* é inspecionar o resultado da *Sprint* e determinar futuras adaptações. O Time de Desenvolvimento apresenta os resultados de seu trabalho para as principais partes interessadas e o progresso em direção ao objetivo do produto é discutido. Ainda conforme dito por Schwaber e Sutherland (2020), durante o evento, o Time de Desenvolvimento e as partes interessadas revisavam o que foi realizado na *Sprint* e o que mudou em seu ambiente. Com base nessas informações, os participantes colaboravam sobre o que fazer em seguida. O *Product Backlog* também pode ser ajustado para atender a novas oportunidades. O autor teve a oportunidade de participar de diversas *Sprint Reviews* ao longo do tempo de trabalho, principalmente por ser um dos principais integrantes. Durante este tipo de interação com o Time de Desenvolvimento e demais *stakeholders*, as habilidades de negociação e entendimento abrangente do contexto apresentado foram fatores chave para o sucesso da atuação do autor.

3.4.5 *Sprint Retrospective*

Dentro dos ciclos de desenvolvimento de *software* os quais o autor atuou, o último ritual executado ao final de cada *Sprint* foram as Retrospectivas ou *Sprint Retrospective* (no termo original do Scrum). Conforme dito por Schwaber e Sutherland (2020), o objetivo da

Sprint Retrospective é planejar maneiras de aumentar a qualidade e a eficácia. O Time de Desenvolvimento inspeciona como a última *Sprint* foi em relação aos indivíduos, interações, processos, ferramentas e discute o que correu bem durante a *Sprint*, quais problemas foram encontrados e como esses problemas foram (ou não) resolvidos. A atuação do autor nesse tipo de cerimônia era central, auxiliando na facilitação das reuniões e documentando os resultados das discussões realizadas pelo Time de Desenvolvimento. Dentre as diversas reuniões realizadas, o autor exerceu inúmeras vezes sua liberdade de atuação propondo diferentes dinâmicas para tal reunião, buscando aumentar sua eficácia, além de proporcionar momentos de reflexão e união do Time de Desenvolvimento. Após a saída da empresa, o autor ainda documentou um guia com as variantes de condução desse tipo de reunião, disponibilizando ao seu sucessor um material para reutilização nas reuniões subsequentes à sua saída. A Figura 6 abaixo mostra um exemplo de como o autor conduziu algumas das *Sprint Retrospectives*, registrando os tópicos elencados pelo Time de Desenvolvimento usando a ferramenta FunRetro¹⁵.

Figura 6. Exemplo de Quadro de *Sprint Retrospective*



Fonte: Website FunRetro (2022).

¹⁵ Ferramenta para aplicação de diversas técnicas de *Sprint Retrospective*. Disponível em: <https://www.funretrospectives.com/funretro/>

4 TECNOLOGIAS UTILIZADAS

Neste capítulo, são apresentadas as principais tecnologias utilizadas pelo autor durante o exercício da função de Analista de Negócios na Equals. Tais tecnologias foram precursoras de grande parte dos projetos desenvolvidos, bem como da evolução profissional do autor mediante desafios vivenciados e do aprendizado extraído a partir da vivência prática com elas. A primeira seção aborda o conceito de API e exemplifica como ela se inseriu no processo de operações da empresa. A segunda seção descreve os princípios e exemplos de Microsserviços. A terceira seção ilustra as principais funções e aplicações do Jira Software, principal ferramenta de trabalho do autor. A quarta seção descreve como e porquê o Metabase esteve presente durante o estágio supervisionado do autor.

4.1 API

Segundo Robillard (2009), uma API (*Application Program Interface*) é um mecanismo de *software* que permite a disponibilização de serviços, por meio de interfaces, para usuários externos, ou seja, uma API é um conjunto de rotinas e padrões de programação para acesso a um serviço de *software* qualquer. Dessa forma, um *software* moderno não necessita ser desenvolvido contendo apenas suas próprias funcionalidades, mas utilizando funcionalidades disponibilizadas por outros *softwares*.

Dentro do contexto da Equals, as APIs desenvolvidas pelos times de Desenvolvimento têm como principal objetivo garantir comunicação fluida entre diferentes frentes de trabalho. Considerando a área de Operações, onde o autor atuou, as APIs disponibilizadas para as demais áreas da empresa tinham por objetivo permitir que dados fossem transitados entre os produtos Equals Core e Raio-X App. Essa estratégia é possível pois, segundo a IBM Cloud Education (2020), uma API permite que os serviços e os produtos se comuniquem entre si e aproveitem os dados e a funcionalidade uns dos outros por meio de uma interface documentada, o que garante maior segurança e celeridade nos processos internos da empresa. Dessa forma, ainda segundo a IBM Cloud Education (2020), os desenvolvedores não precisavam saber como uma API é implementada; eles simplesmente usavam a interface para se comunicar com outros produtos e serviços.

Exemplos reais de APIs desenvolvidas e mantidas pelo time em que o autor esteve diretamente envolvido podem ser destacadas aquelas que permitiam: manipulação de dados de usuários; checagem de regras de negócios a partir de dados de arquivos externos; transição de

dados entre diferentes aplicações e contextos; e gatilhos para disparo de e-mails internos e externos. Ainda segundo a *IBM Cloud Education* (2020), as APIs oferecem segurança por *design* porque sua posição como intermediárias facilita a abstração da funcionalidade entre dois sistemas - o terminal da API separa o aplicativo de consumo da infraestrutura que fornece o serviço. Tais aspectos justificam a sua importância no cenário da empresa. Além disso, como ainda reforça a *IBM Cloud Education* (2020), a segurança exercida dentro de um contexto de APIs é de grande valor visto, que as chamadas de API geralmente incluem credenciais de autorização para reduzir o risco de ataques ao servidor, e um *gateway* de API pode limitar o acesso para minimizar ameaças à segurança.

4.2 Microsserviços

Segundo Stoiber (2017), um microsserviço é uma unidade de *software* autônoma que, juntamente com muitas outras, compõem uma grande aplicação. Ao dividir seu aplicativo em unidades pequenas, cada parte pode ser independentemente implantada e escalada, pode ser escrita por diferentes times de desenvolvimento, em diferentes linguagens de programação e pode ser testada individualmente. O desenvolvimento de *software* realizado no time em que o autor atuou baseia-se em uma arquitetura de microsserviços, desenvolvida a partir da linguagem de programação Java, o que significa que pequenos serviços independentes se comunicam entre si usando as APIs descritas na seção 4.1. Tal estratégia de desenvolvimento foi implementada, pois arquiteturas em microsserviços, de acordo com a Amazon Web Services (2020), tornam os aplicativos mais fáceis de escalar e mais rápidos de desenvolver, permitindo a inovação e acelerando o tempo de colocação no mercado de novos recursos. Segundo a *IBM Cloud Education* (2020), uma arquitetura em microsserviços, diferentemente de uma arquitetura do tipo monolítica¹⁶, possui diversos benefícios para a empresa pois requer menos infraestrutura do que os aplicativos monolíticos, além de permitir ajustes em partes independentes e específicas, reduzindo o custo de evoluções e manutenções no *software*.

Na Equals, os microsserviços possuem papel fundamental no bom funcionamento e comunicação dentre diferentes produtos da empresa, além de atuarem diretamente com as APIs desenvolvidas. Esses microsserviços são responsáveis por grande parte do fluxo de arquivos recebidos diariamente dos milhares de clientes que a empresa possui, além de

¹⁶ “Arquitetura monolítica, relativo à monólito, refere-se ao modelo de desenvolvimento de uma aplicação dentro de uma única estrutura executável, onde o comportamento obtido no resultado final do seu desenvolvimento pode-se encontrar uma única entidade, compacta, indivisível e impenetrável, que posteriormente será executada.” (LUCIO, 2017, p. 21)

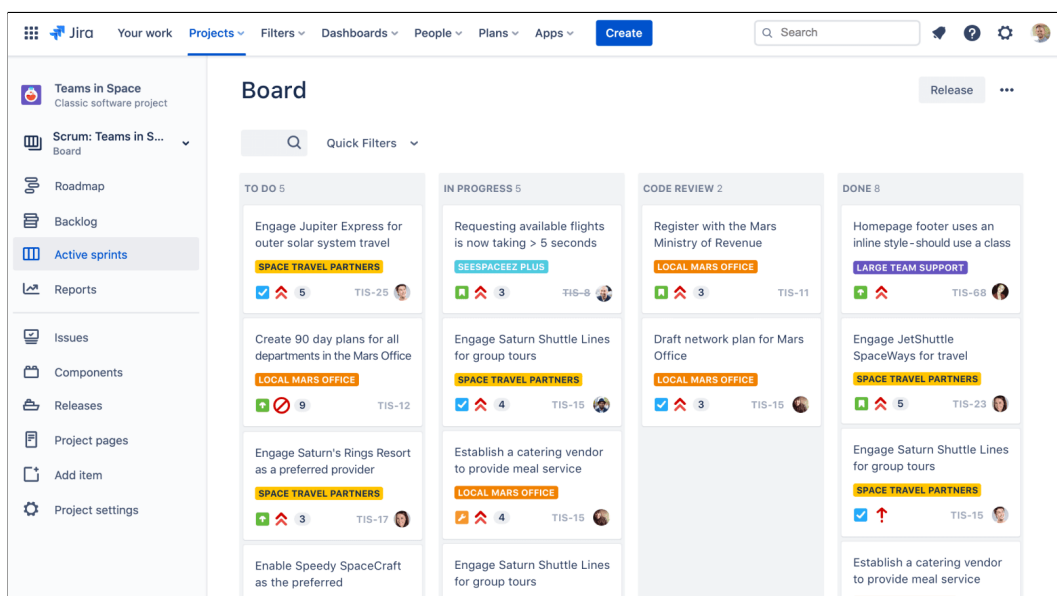
permitirem maior celeridade em processos automatizados como o disparo de e-mails para clientes. O autor esteve em constante trabalho de manutenção, evolução e criação de novos microsserviços durante sua trajetória na empresa.

4.3 Jira Software

O Jira¹⁷ é uma das principais ferramentas utilizadas dentro do ambiente interno da Equals e possui a missão de auxiliar grande parte dos times da organização a gerenciarem seu trabalho. Segundo o *site* da empresa, o Jira é uma ferramenta de gestão ágil de projetos que oferece suporte a qualquer metodologia ágil, como Scrum, Kanban ou qualquer outra e dada a sua versatilidade é capaz de agregar de diferentes formas para o trabalho cotidiano dos times da organização.

Durante o trabalho de estágio supervisionado, o autor trabalhou diariamente com essa ferramenta a partir de quadros ágeis de Kanban, seções de documentação de *software* e regras de negócio, criação de *roadmaps* de entregas, além de integrações com outras ferramentas. As principais ações executadas pelo autor durante o trabalho exercido estão dispostas na seção 5.3. A Figura 7 ilustra um exemplo de quadro utilizado pelo autor durante o estágio supervisionado.

Figura 7. Exemplo de Quadro Kanban no Jira Software



Fonte: Website Atlassian (2021).

¹⁷ ATLISSIAN. Jira. [S.l.]. Atlassian, 2022. Disponível em: <https://www.atlassian.com/br/software/jira>. Acesso em: 21 mar. 2022.

4.4 Metabase

O Metabase¹⁸ é uma ferramenta de *Business Intelligence* (BI) de código aberto que permite criar gráficos e painéis usando dados de uma variedade de bancos de dados e fontes de dados. Essa ferramenta é utilizada pela Equals para fornecer, de forma transparente, o desempenho dos times para toda a organização. Como o Metabase necessita de bases de dados para alimentar seu funcionamento, os times de Operações e Tecnologia, principalmente, dispõem de uma integração dessa ferramenta com o Jira Software. Como as demandas do time que o autor trabalhou eram concentradas no Jira Software, após a integração com o Metabase, tornou-se possível a geração de quadros contendo gráficos para medir o desempenho do time e de cada integrante, do ponto de vista dos produtos de *software* gerados ao longo do tempo. O principal projeto implementado pelo autor fez uso dessa ferramenta, sendo uma dos primeiros times a implantar o uso contínuo durante os ciclos de desenvolvimento de *software*. As Figuras 8 e 9 ilustram como são construídos os quadros do Metabase.

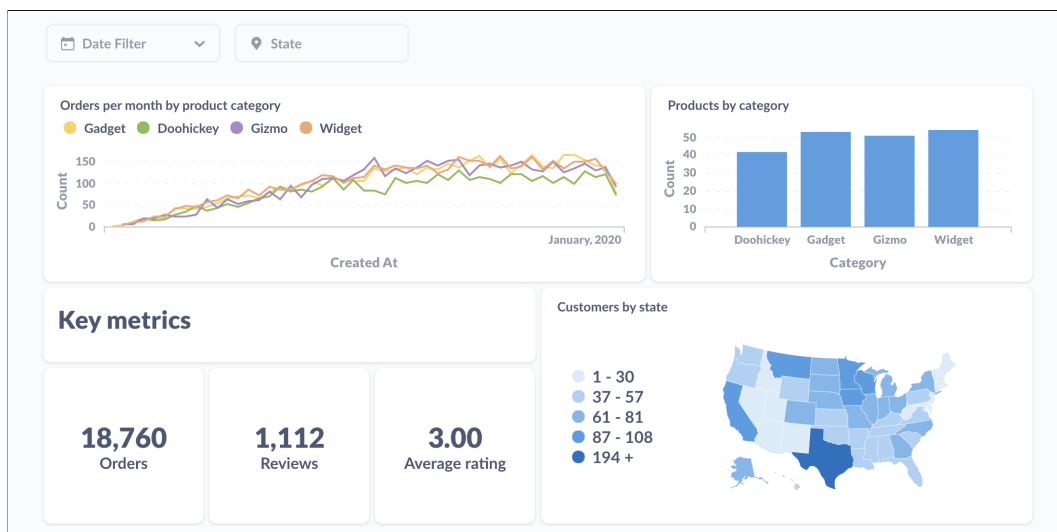
Figura 8. Exemplo de Construção de um Painel de BI no Metabase



Fonte: Website Metabase (2022).

¹⁸ METABASE. Metabase. Metabase, 2022. Disponível em: <https://www.metabase.com/product/>. Acesso em 21 mar 2022.

Figura 9. Exemplo de um Painel de BI no Metabase



Fonte: Website Metabase (2022).

5 PROJETOS DESENVOLVIDOS

Neste capítulo, são apresentadas as atividades desenvolvidas e as oportunidades experienciadas pelo autor desde o início de sua carreira. A Seção 5.1 descreve a experiência do autor ao realizar as atividades de levantamento e análise de requisitos. A Seção 5.2 relata o processo de documentação de APIs e microsserviços implementados.

O Capítulo segue com a descrição de outros dois projetos realizados de grande importância e relevância durante a experiência do autor enquanto trabalhou na empresa. A Seção 5.3 aborda o projeto de automação de processos implementado utilizando a ferramenta Jira Software. A Seção 5.4 traduz as etapas de implantação de painéis de *Business Intelligence* utilizando a ferramenta Metabase.

5.1 Análise de Requisitos de Software

Segundo Vazquez e Simões (2016), um produto de *software* pode ser considerado de qualidade se suas características estão compatíveis com sua especificação, isto é, cumpre com o seu propósito o qual os requisitos foram levantados. Dessa forma, seja o requisito uma necessidade a ser satisfeita ou uma propriedade de um produto existente, em ambos os casos podem ter suas capacidades especificadas em um documento. Ainda de acordo com Vazquez e Simões (2016), as especificações de requisitos de um *software* devem fornecer um escopo definido como macrofunções, embutindo vários itens a definir, por exemplo, quais tarefas dos fluxos operacionais no negócio em específico serão total ou parcialmente transformadas em *software*. Das partes a serem codificadas, por exemplo, é essencial delimitar o comportamento que se espera do *software* quando funcionando em cada uma das tarefas definidas para ele executar.

A etapa de análise e definição de requisitos é realizada de maneira subsequente à de *Product Discovery* e pode ser considerada como uma parte dela. Contudo, o nível de detalhamento nessa etapa é maior e mais específico para que sejam gerados itens que alimentem o *Product Backlog*. Durante o processo de levantamento e análise de requisitos, o autor sempre pautou-se em uma estratégia de comunicação com todos os envolvidos para obter o máximo de conhecimento a respeito do contexto. O autor não utilizou um modelo único para a descrição dos requisitos, como mencionado na seção 3.4.2. Fruto de um trabalho conjunto com os membros do Time de Desenvolvimento e o Líder do Time, cada documento gerado a partir do levantamento de requisitos detinha tópicos específicos e inerentes ao

contexto em que aquele documento estava inserido. Como exemplo, um documento com o objetivo de analisar e elicitar requisitos para um microsserviço era descrito a partir de um modelo próprio, bem como um outro específico para uma API detinha seu próprio modelo. Cada um desses modelos era baseado nas principais características de cada tecnologia, quais dados eram necessários para uma completa compreensão dos envolvidos e execução das demandas por parte do Time de Desenvolvimento. Os requisitos de *software* possuem um papel de grande importância para a construção eficiente e eficaz de produtos como o Equals Core, Raio-X App e OPS-Monitor, além das APIs e Microsserviços que trabalham em consonância com estes produtos. Eles podem ser vistos como gatilhos para a definição de que o produto correto está sendo desenvolvido da maneira correta, ou seja, permitem avaliar se o processo e o artefato final estão de acordo com o esperado. Requisitos de *software* possuem forte apelo para criar estimativas de escopo e prazos que norteiam a cadeia produtiva de um Time de Desenvolvimento de *software*.

5.2 Documentação de API e Microsserviços

O processo de levantamento e análise de requisitos de *software* é seguido necessariamente por uma etapa de documentação e especificação, de acordo com as políticas e boas práticas adotadas pela Equals. No contexto vivenciado pelo autor, foram utilizadas duas principais ferramentas para a documentação dos requisitos: Jira Confluence¹⁹ e Apiary²⁰. A primeira ferramenta detinha os dados referentes aos produtos e microsserviços desenvolvidos pelo Time de Desenvolvimento a qual o autor atuou. A segunda ferramenta possuía um foco exclusivo na documentação das rotas desenvolvidas para a API fornecida pelo mesmo time em questão.

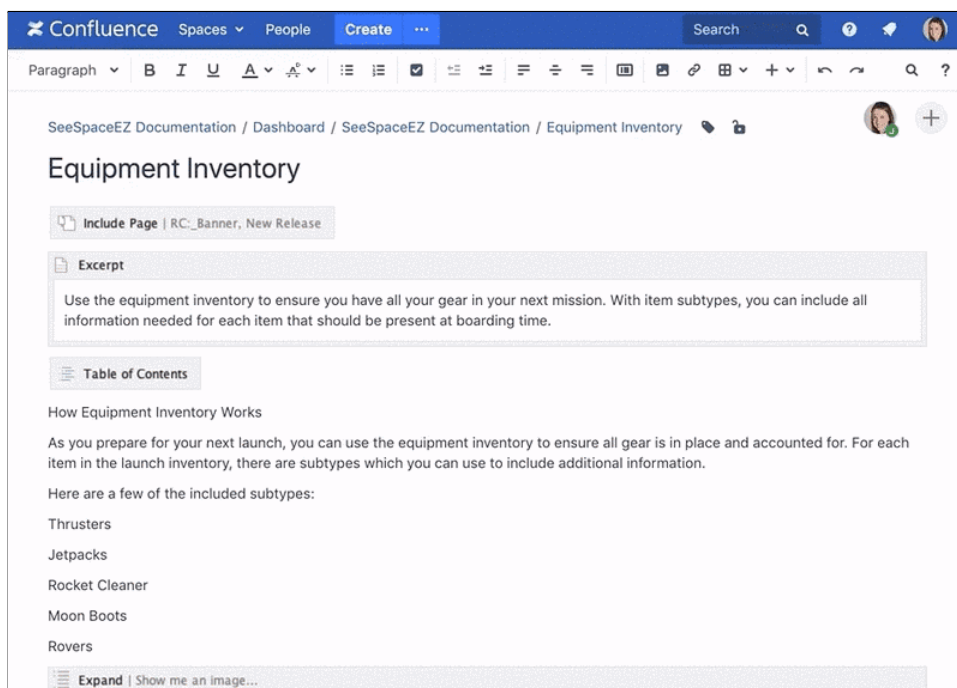
Ainda que não seja a norma padrão utilizada pela Equals, a norma ISO/IEC/IEEE 24765 (2010) aborda o fato de que o documento de requisitos objetiva incluir declarações não ambíguas dos requisitos e ser completo. Outras características importantes do documento é ser verificável, consistente, modificável, rastreável e usável durante as fases do ciclo de vida do requisito. Cabe ressaltar que a atuação do autor durante o processo de documentação sempre visou abordar diferentes perspectivas e atender às mais variadas aplicações dos artefatos de *software* documentados. Como os documentos gerados eram utilizados por diferentes áreas, um dos pilares do trabalho executado era garantir que os materiais fossem claros e aplicáveis

¹⁹ ATLASSIAN. Confluence. Atlassian, 2022. Disponível em: <https://www.atlassian.com/br/software/confluence>. Acesso em: 21 mar. 2022.

²⁰ ORACLE. Apiary. Oracle, 2022. Disponível em: <https://apiary.io/>. Acesso em: 21 mar. 2022.

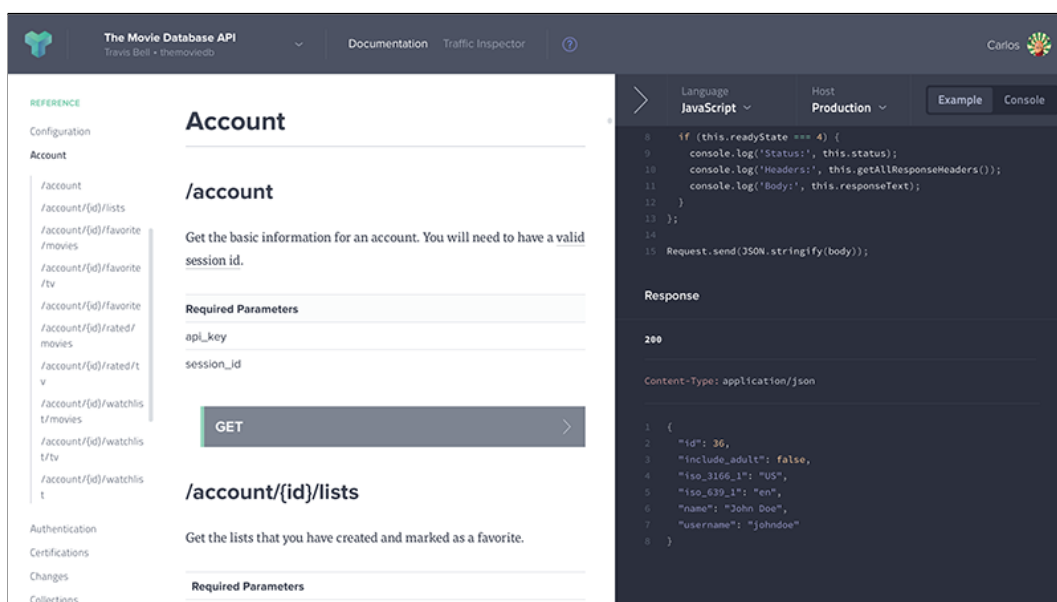
a tais contextos, de maneira a abranger os cenários de uso dos dados relatados. As Figuras 10 e 11 exemplificam como são elaborados os documentos de APIs e Microserviços usando as ferramentas Confluence e Apiary, respectivamente.

Figura 10. Exemplo de Documentação com Confluence



Fonte: Atlassian - Confluence (2022).

Figura 11. Exemplo de Documentação de uma API com Apiary



Fonte: Oracle - Apiary (2022).

5.3 Automação de Processos no Jira Software

Segundo Hannana e Freeman, (1983), dentro de qualquer organização, são encontradas inúmeras rotinas de trabalho repetitivas. Tais ações são a base daquilo que a empresa prega para atingir seus objetivos, o que chama-se de jeito de ser da organização, forma de pensar e agir de seus colaboradores. Por se tratarem de um aspecto fundamental do trabalho organizado dentre os fatos humanos, as rotinas organizacionais também são uma fonte extensa de análises de inércia. De acordo com Feldman (2003), cada uma das rotinas organizacionais são definidas como padrões de ações: “repetitivas, padrões de ações interdependentes, executadas por múltiplos atores”.

Durante sua atuação enquanto Analista de Negócios, o autor teve a oportunidade de vivenciar um projeto de otimização da jornada de trabalho de diferentes participantes do Time de Desenvolvimento. Esse projeto baseava-se na premissa de que, durante a execução de tarefas do dia a dia, o time despendia uma quantidade de tempo considerável e completamente desnecessária executando ações facilmente automatizáveis. A partir dessa premissa, foi feito um levantamento e diversas análises de rotinas executadas dentro da plataforma Jira Software que poderiam ser automatizadas com o auxílio de tecnologia, buscando tornar o dia a dia das pessoas mais produtivo.

Como meio de melhorar a garantia de execução dos processos necessários dentro da plataforma Jira Software, foi proposta a utilização de um *plugin* acoplado a essa plataforma, JMWE (*Jira Misc Workflow Extensions*)²¹. Seu principal objetivo é criar e automatizar fluxos de trabalho dentro do Jira Software com um arsenal grande de possibilidades de configurações a serem implementadas em diferentes funções da ferramenta. O trabalho realizado atingiu resultados além da otimização de rotinas manuais, mas se estenderam para outras áreas como análise de medidas e geração de relatórios. Tendo em vista que ações manuais são passíveis de erros, como muitas das tarefas automatizadas representavam o preenchimento de dados importantes para documentações e geração de painéis de negócios, a partir da implementação do projeto, esses erros foram reduzidos a zero. A Figura 12 mostra um processo de criação de uma automação de processo utilizando a ferramenta JMWE na plataforma Jira Software.

²¹ ATLISSIAN. Confluence. Atlassian, 2022. Disponível em: <https://innovalog.atlassian.net/wiki/spaces/JMWE/overview>. Acesso em: 21 mar. 2022.

Figura 12. Exemplo de uma Automação de Processo utilizando o JMWE.

Transition	When	By	Time in status
→ TO DO	Nov 10, 2017 1:24 PM	Radhika Vijji.	
TO DO → IN PROGRESS	Jun 10, 2019 10:04 PM	Inna Gordin	2 years
IN PROGRESS → DONE	Jun 10, 2019 10:06 PM	Inna Gordin	2 minutes
DONE → TO DO	Jun 10, 2019 10:06 PM	Inna Gordin	a minute

Fonte: Atlassian - JMWE (2022).

5.4 Criação de Painéis de *Business Intelligence* no Metabase

Segundo Turban *et al.* (2008, p. 27), *Business Intelligence* (BI) é um termo guarda-chuva que inclui arquiteturas, ferramentas, bancos de dados, aplicações e metodologias. Ainda segundo Turban *et al.* (2008, p. 27), os principais objetivos do BI são permitir o acesso interativo aos dados (às vezes, em tempo real), proporcionar a manipulação desses dados e fornecer aos gerentes e analistas de negócios a capacidade de realizar a análise adequada. Dentre os principais benefícios da aplicação de ferramentas de BI, podem-se destacar: economia de tempo e custos; centralização de dados em uma fonte segura; maior transparência quanto aos dados relevantes para o negócio; auxílio na tomada de decisões baseadas em dados; e suporte para a construção de processos mais eficazes.

A partir do trabalho realizado pelo autor com a automação de processos na plataforma Jira Software, deu-se início um processo de aplicação dos dados gerados com o Jira Software. Como a maior parte do trabalho realizado pelo Time de Desenvolvimento gerava um histórico utilizando as interações com a plataforma e à medida que essas interações acarretavam ações automáticas de captação de dados, tornou-se possível realizar a transformação de dados em informações, depois em decisões e finalmente em ações (TURBAN *et al.*, 2008, p. 33).

A ferramenta utilizada pelo autor para a geração de painéis de BI foi o Metabase, uma solução *open-source* integrada com o Jira Software utilizando um banco de dados e API. Com

o Metabase, foi possível construir diversos painéis para diferentes finalidades a partir dos dados captados da integração. Esses painéis foram utilizados em diversos rituais adotados pelo Time de Desenvolvimento e para análises de gestores, tornando possível a análise do cenário atual e realização de projeções futuras, pois os dados eram confiáveis. Como resultado desse projeto, pode-se destacar que muitas decisões tomadas, em diferentes camadas da empresa, mediante o trabalho do Time de Desenvolvimento no qual o autor atuou, tornaram-se mais assertivas e fáceis de serem tomadas, pois foram baseadas em dados e fatos reais e não apenas em opiniões ou hipóteses não fundamentadas do trabalho que estava sendo desenvolvido.

6 CONSIDERAÇÕES FINAIS

O começo de uma carreira, em qualquer área de atuação, é sempre um grande desafio para as pessoas. Por se tratar de uma jornada totalmente em estágio inicial, muitas dúvidas e anseios são gerados e, conseqüentemente, um cenário nebuloso é oferecido para profissionais que enfrentam esse tipo de momento. Ter a oportunidade de vivenciar as experiências mencionadas neste trabalho, assim como muitas outras não descritas, foram de grande impacto e importância para o início da trajetória profissional do autor. Não apenas mediante desafios que representavam a conversão de uma teoria aprendida em sala de aula, mas de uma exposição constante a diversas situações do cotidiano as quais exigem uma postura e tomada de decisão mediante seus valores e experiências passadas. A vivência de uma cultura forte e acolhedora como é a da Equals possui uma relevância importante de ser destacada dentro desse contexto. Dar a oportunidade para uma pessoa que ainda não concluiu seu processo de formação completo e entender as exigências e limitações cabíveis a esse profissional são atitudes nobres e devem ser reforçadas como conclusão dessa jornada.

O autor iniciou seu estágio no segundo semestre de 2019, enquanto vivenciava seu sexto período do curso de Sistemas de Informação, o que representa o fato de que muitas disciplinas importantes para a formação profissional sequer haviam sido cursadas. Contudo, a vivência prática, somada à base teórica obtida até então e, principalmente, todo o suporte e dedicação de outras pessoas presentes no dia a dia de trabalho do autor, foram de suma importância para uma experiência enriquecedora. Dentre os mais de 200 colaboradores que trabalhavam na Equals, o autor teve a oportunidade de ser mentorado por diversas pessoas, de diferentes áreas da empresa, as quais abriram caminhos e deram suporte e confiança para que o autor pudesse extrair o melhor de si e aprimorar sua atuação.

A formação de um profissional passa não apenas por seus conhecimentos teóricos e práticos com relação a aspectos técnicos no exercício de sua função, mas por seu caráter, valores e postura adotada mediante as mais variadas situações às quais ele é exposto. Sem sombra de dúvidas, o período vivido pelo autor, atuando enquanto Analista de Negócios na Equals, foi de essencial importância para a fundamentação do profissional que o direcionou para se tornar durante e após tal experiência.

O ofício de Analista de Negócios exige forte habilidade de comunicação, entendimento de um mesmo cenário sob diversas perspectivas e muita dedicação para equilibrar as necessidades e as possibilidades existentes entre os pilares de clientes, negócio e

tecnologia. As bases obtidas durante diversas disciplinas cursadas na Universidade Federal de Lavras proporcionaram caminhos menos árduos ao autor, por se tratarem de aspectos antes estudados e debatidos com docentes de grande qualificação. Vale mencionar que as disciplinas de Administração Estratégica, Gestão de Tecnologia da Informação, Engenharia de Software, Processos de Software e Gerência de Projetos de Software, cursadas pelo autor, destacam-se dentre aquelas que deram maior embasamento teórico para o exercício da função de Analista de Negócios. Além disso, o fato de a empresa proporcionar acesso a diversos cursos de formação especializada para a área de atuação em questão também deve ser destacada pois auxiliou de forma imediata na ampliação dos horizontes possíveis da atuação enquanto Analista de Negócios.

Ademais aos fatores mencionados, é preciso considerar aspectos pouco tangíveis e mais abstratos, mas que possuem grande relevância no contexto abordado. Esses aspectos não necessariamente estão relacionados ao curso ou ao exercício da profissão em si, mas possuem forte relevância para a formação completa de um profissional moderno. Cabe destacar que a experiência vivida pelo autor o auxiliaram a desenvolver habilidades como comunicação, inteligência emocional, organização e antifragilidade para lidar com adversidades da rotina. Conciliar os estudos do curso de graduação com a rotina de trabalho e os demais compromissos pessoais foi um desafio superado pelo autor mediante esforço e persistência e, sem o suporte de inúmeras pessoas presentes nesse contexto, jamais teriam sido vivências tão completas e enriquecedoras como foram.

Dessa forma, compreendendo as variáveis e situações descritas neste documento, é passível de concluir que o saldo da experiência de realização de um estágio supervisionado foi extremamente positivo para o autor. Muito além de um registro no currículo, um cargo ou um salário oferecido pela empresa, ter vivido cada um dos dias e situações mencionados foram de suma importância para uma construção sólida das bases do profissional que começou a ser moldado durante esse período. Embora o autor não tenha seguido na empresa após o término do contrato estipulado inicialmente, não restam dúvidas de que as oportunidades que se abriram após a experiência na Equals foram diretamente relacionadas ao crescimento adquirido durante essa passagem pela empresa. O profissional que se forma na graduação é o reflexo do conhecimento teórico, prático, os momentos vivenciados, aprendizados adquiridos e as pessoas que fizeram parte dessa jornada.

REFERÊNCIAS

- AGILE ALLIANCE. **What is a Kanban Board?** 2021. Disponível em: <https://www.agilealliance.org/glossary/kanban-board>. Acesso em: 27 mar. 2022.
- BECK, Kent *et al.* **Manifesto for Agile Software Development**. 2001. Disponível em: <http://agilemanifesto.org>. Acesso em: 21 mar. 2022.
- FELDMAN, Martha S. **A performative perspective on stability and change in organizational routines**. *Industrial and Corporate Change*, v. 12, n. 4. p. 727 - 752, 2003.
- HANNAN, Michael T., FREEMAN, John. **Structural inertia and organizational change**. *American Sociological Review*, v. 49, p.149-64, 1984.
- ISO/IEC/IEEE. **Std. 24765-2010 IEEE Guide for Developing System Requirements Specification**. The Institute of Electrical and Electronics Engineers. Piscataway, NJ. Guide, EUA, 2010. Disponível em: <https://www.iso.org/standard/50518.html>. Acesso em: 11 abr. 2022.
- IBM CLOUD EDUCATION. **What is an Application Programming Interface (API)?** Disponível em: <https://www.ibm.com/cloud/learn/api>. Acesso em: 27 mar. 2022.
- IBM CLOUD EDUCATION. **What are Microservices?**. Disponível em: <https://www.ibm.com/cloud/learn/microservices>. Acesso em: 27 mar. 2022.
- INTERNATIONAL INSTITUTE OF BUSINESS ANALYSIS (IIBA). **Padrão Essencial Global do IIBA para Análise de Negócios**. IIBA, 2017. Disponível em: <https://www.iiba.org/globalassets/career-resources/business-analysis-resources/iiba-bookstore/files/global-business-analysis-core-standard---portuguese.pdf>. Acesso em: 21 mar. 2022.
- LEWIS, J.; FOWLER, M. **Microservices: a definition of this new architectural term**. Disponível em: <https://martinfowler.com/articles/microservices.html>. Acesso em: 27 mar. 2022.
- LOPES, Leandro *et al.* **Uma proposta para processo de requisitos em ambientes de desenvolvimento distribuído de software**, Piracicaba. Brasil. 2003. p. 329-342. Disponível em: <https://1library.org/document/z15l6pdy-uma-proposta-processo-requisitos-ambientes-desenvolvimento-distribuido-software.html>. Acesso em: 21 mar. 2022
- LOPES, Leandro T. **Um Modelo de Processo de Engenharia de Requisitos para Ambientes de Desenvolvimento Distribuído de Software**. Porto Alegre, 2004. 142f.. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul. Disponível em: <http://tede2.pucrs.br/tede2/bitstream/tede/5300/1/384394.pdf>. Acesso em: 21 mar. 2022.

LUCIO, João Paulo Duarte. **Análise comparativa entre arquitetura monolítica e de microsserviços**. Florianópolis, 2017. p. 21. Dissertação (Bacharelado em Sistemas de Informação) - Departamento de Informática e Estatística, Universidade Federal de Santa Catarina Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/182309/TCC_corrigido.pdf?sequence=1&isAllowed=y. Acesso em 07 mai. 2022

KHAN, Zahoor A. **Scrumban-adaptive agile development process**: Using scrumban to improve software development process. 2014. Disponível em: https://www.theseus.fi/bitstream/handle/10024/77014/Khan_Zahoor.pdf. Acesso em: 27 mar. 2022.

PAULA FILHO, W. P. **Engenharia de software: fundamentos, métodos e padrões**. Rio de Janeiro: Livros Técnicos e Científicos, 2001.

PM3. **O que é Scrumban?** 2020. Disponível em: <https://www.cursospm3.com.br/blog/o-que-e-scrumban>. Acesso em: 21 mar. 2022.

PM3. **Product Discovery**: o que é, importância e como fazer. Disponível em: <https://www.cursospm3.com.br/blog/product-discovery-guia>. Acesso em: 27 mar. 2022.

ROBILLARD, Martin P. **What makes apis hard to learn?** Answers from developers. IEEE Software, 26(6):27–34, 2009.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**. The Definitive Guide to Scrum: The Rules of the Game. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Acesso em: 21 mar. 2022.

STOIBER, Max. **Build your first Node.js microservice**. Disponível em: <https://mxstbr.blog/2017/01/your-first-node-microservice>. Acesso em: 21 mar. 2022.

TURBAN, Efraim *et al.* **Business intelligence: um enfoque gerencial para a inteligência do negócio**. São Paulo: Bookman, 2009.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. **Engenharia de Requisitos: software orientado ao negócio**. São Paulo: Brasport, 2016.