



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS
CURSO DE ENGENHARIA DE PRODUÇÃO MECÂNICA

Felipe da Rosa Kuse

***Aplicação do framework Scrum em uma startup de desenvolvimento de aplicativos web
de fidelização de clientes***

Florianópolis

2024

Felipe da Rosa Kuse

**Aplicação do *framework Scrum* em uma *startup* de desenvolvimento de aplicativos *web*
de fidelização de clientes**

Trabalho de Conclusão de Curso submetido ao curso de Engenharia de Produção Mecânica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica com habilitação em Produção.

Orientador: Prof. Artur Santa Catarina.

Florianópolis

2024

Kuse, Felipe da Rosa

Aplicação do framework Scrum em uma startup de desenvolvimento de aplicativos web de fidelização de clientes / Felipe da Rosa Kuse ; orientador, Artur Santa Catarina, 2024.

91 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Produção Mecânica, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Produção Mecânica. 2. Scrum. 3. Gerenciamento de Projetos. 4. Trabalho Remoto. I. Catarina, Artur Santa. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Produção Mecânica. III. Título.

Felipe da Rosa Kuse

**Aplicação do *framework Scrum* em uma *startup* de desenvolvimento de aplicativos *web*
de fidelização de clientes**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Engenheiro Mecânico com habilitação em Produção e aprovado em sua forma final pelo Curso de Engenharia de Produção Mecânica

Florianópolis, 25 de junho de 2024.



Prof^ª. Mônica Maria Mendes Luna, Dra.

Coordenação do Curso

Banca examinadora



Prof. Artur Santa Catarina, Dr.

Orientador

Universidade Federal de Santa Catarina



Prof. Diego de Castro Fettermann, Dr.(a)

Universidade Federal de Santa Catarina



Felipe Goes

Universidade Federal de Santa Catarina

Florianópolis, 2024

RESUMO

A utilização de metodologias ágeis como ferramenta para gestão de projetos vem sendo difundida e ganhando espaço nos últimos anos, principalmente no âmbito de desenvolvimento de produtos tecnológicos, onde há rápidas evoluções no mercado. As ferramentas de metodologia são uma alternativa útil que tem como premissa trazer mais contato entre clientes e desenvolvedores, com respostas e entregas em pequenas etapas ao longo do projeto. Essa maior interação e pequenas entregas permitem que alterações sejam feitas no decorrer do projeto, moldando, assim, o produto ao longo do seu desenvolvimento. O presente trabalho de conclusão de curso tem como objetivo aplicar e avaliar o resultado das metodologias ágeis, especificamente o *Scrum*, em uma *startup* de desenvolvimento de *software* de fidelização na cidade de Florianópolis. O estudo visou melhorar a gestão de projetos e a eficiência no desenvolvimento de produtos na empresa analisada. Para poder avaliar os resultados, inicialmente foi relatado o cenário encontrado antes da implementação da ferramenta *Scrum*, aplicadas entrevistas com membros da *Startup* e utilizado um mapeamento dos processos. Encontrados então os pontos negativos no processo de desenvolvimento, o *Scrum* foi aplicado de maneira prática com um time de desenvolvedores e *designers*. Os resultados indicaram que a adoção do *Scrum* trouxe benefícios significativos, como maior organização das atividades, proatividade da equipe e uma dinâmica mais eficiente nas entregas. As cerimônias do *Scrum* foram fundamentais para manter o foco da equipe e resolver dúvidas rapidamente, promovendo um ambiente de *feedback* constante e colaboração. As principais dificuldades observadas estavam relacionadas à adaptação das metodologias ágeis ao contexto remoto, exigindo ajustes específicos para manter a eficácia dos processos. A experiência mostrou que tais adaptações são possíveis e podem conduzir a uma melhor gestão de projetos, mesmo em cenários diferentes dos encontrados na base da literatura das metodologias ágeis.

Palavras-chave: *Scrum*; Gestão de projetos; Trabalho remoto; Fidelização.

ABSTRACT

The use of agile methodologies as a tool for project management has been widespread and gaining ground in recent years, mainly in the field of technological product development, where there are rapid developments in the market, methodology tools are a useful alternative that can bring benefits more contact between customers and developers, with responses and deliveries in small stages throughout the project. This greater interaction and small deliveries allow changes to be made throughout the project, thus shaping the product throughout its development. This final course completion project aims to apply and evaluate the results of agile methodologies, specifically Scrum, in a loyalty software development startup in the city of Florianópolis. The study aims to improve project management and efficiency in product development at the proven company. In order to evaluate the results, the scenario found before implementing the Scrum tool was initially reported, interviews were carried out with members of the Startup and the use of a process mapping. Once the negative points in the development process were found, Scrum was applied in a practical way over time by developers and designers. The results indicated that the adoption of Scrum brought benefits, such as greater organization of activities, team proactivity and more efficient delivery dynamics. The Scrum meetings were essential for maintaining the team's focus and resolving doubts quickly, promoting an environment of constant feedback and collaboration. The main difficulties observed were related to adapting the methodologies to the remote context, requiring specific adjustments to maintain the effectiveness of the processes. Experience has shown that such adaptations are possible and can lead to better project management, even in scenarios different from those found in the methodological literature.

Keywords: Scrum; Project management; Remote work; Loyalty.

SUMÁRIO

1. INTRODUÇÃO	9
1.1 OBJETIVOS.....	10
1.1.1 <i>Objetivo Geral</i>	10
1.1.2 <i>Objetivos Específicos</i>	10
1.2 JUSTIFICATIVA.....	10
2. REFERENCIAL TEÓRICO	11
2.1 DESENVOLVIMENTO DE <i>SOFTWARES</i>	11
2.1.1 <i>Evolução Tecnológica</i>	11
2.1.2 <i>Empresas desenvolvedoras de Softwares</i>	13
2.1.3 <i>Tipos de Softwares</i>	16
2.2 GERENCIAMENTO DE PROJETOS	17
2.2.1 <i>Extreme Programming (XP)</i>	21
2.2.2 <i>Scrum</i>	23
2.2.3 <i>Método Kanban</i>	30
2.3 O ÁGIL NO TRABALHO REMOTO	31
2.3.1 <i>Dificuldades do Ágil no trabalho remoto</i>	31
2.3.2 <i>Vantagens do ágil do trabalho remoto</i>	32
3. METODOLOGIA.....	33
3.1 TIPO DE PESQUISA.....	33
3.2 A EMPRESA.....	34
3.2.1 <i>Situação da empresa</i>	34
3.2.2 <i>O produto da empresa</i>	34
3.2.3 <i>Estrutura organizacional</i>	36
3.3 PROCEDIMENTOS PARA O DESENVOLVIMENTO DO TRABALHO	37
3.3.1 <i>Entrevistas com a equipe</i>	37
3.3.2 <i>Mapeamento de processos</i>	40
3.3.3 <i>Aplicação de metodologia Scrum</i>	40
3.3.4 <i>Entrevistas finais e comparação de resultados após aplicação do Scrum</i>	40
4. RESULTADOS	40

4.1	SITUAÇÃO INICIAL.....	41
4.1.1	<i>Busca por processos ágeis</i>	<i>41</i>
4.1.2	<i>Fluxo de processos no momento inicial.....</i>	<i>42</i>
4.2	ENTREVISTAS	43
4.2.1	<i>Formulário com escala Likert.....</i>	<i>43</i>
4.2.2	<i>Entrevistas semiestruturadas</i>	<i>45</i>
4.3	DEFINIÇÃO DO ESCOPO	45
4.4	APLICANDO TÉCNICAS DO SCRUM.....	47
4.4.1	<i>Time Scrum</i>	<i>47</i>
4.4.2	<i>Planejamento a nível macro dos entregáveis no trimestre 2.....</i>	<i>48</i>
4.4.3	<i>Software de acompanhamento das atividades</i>	<i>50</i>
4.4.4	<i>Sprints</i>	<i>52</i>
4.4.5	<i>Cerimônias Scrum.....</i>	<i>67</i>
4.5	QUESTIONÁRIO SOBRE APLICAÇÃO DO SCRUM	71
4.6	DIFICULDADES ENCONTRADAS PARA MANTER OS PROCESSOS	74
4.7	DISCUSSÃO SOBRE RESULTADOS.....	76
5.	CONCLUSÃO.....	80
6.	REFERÊNCIAS.....	83
7.	APÊNDICES	87

1. INTRODUÇÃO

A evolução tecnológica vem tomando grande espaço nas relações humanas nas últimas décadas, principalmente após os anos 2000 com a popularização dos telefones celulares, computadores pessoais e smartphones. O celular deixou de ser apenas um aparelho para fazer ligações telefônicas e se tornou um assistente pessoal com diversas funcionalidades. Aproveitando-se disso, o setor comercial passou a aproveitar esse nicho para atingir seus clientes por meio digital.

Cresceu, então, a necessidade de desenvolvimentos de *Softwares*, sites, plataformas *web*, aplicativos e todo tipo de sistema que possa atingir os diversos segmentos de clientes que utilizam seus *smartphones* diariamente. Os *e-commerce*, por exemplo, cresceram cerca de 367% entre 2011 e 2020, segundo a pesquisa Webshoppers (EBIT, 2021), chegando à marca de 87 bilhões de reais em vendas em 2020. Muito desse crescimento veio impulsionado pela pandemia do Covid-19 que iniciou em 2019 e revolucionou o mundo com seus *lockdowns*, fazendo-se cada vez mais necessário a inserção das empresas no meio digital para continuarem vivas entre os *players* do mercado.

O desenvolvimento de plataformas digitais sentiu a necessidade de evoluir na inércia das constantes mudanças tecnológicas. Hoje em dia é necessário que a gestão dos projetos possa adaptar-se aos requisitos do cliente, recebendo recorrentes *feedbacks* positivos e negativos, e a partir daí agir em ações de correção instantânea. Os projetos estão mais dinâmicos, mais participativos e por isso a aplicação de novas metodologias pode ser bem-vinda.

As metodologias ágeis têm esse objetivo de trazer uma interação recorrente com as partes envolvidas, fazendo entregas menores, possibilitando alterar ou inserir novas etapas no decorrer do projeto. Porém é necessário entender se a utilização de uma metodologia ágil, como o *Scrum*, se aplica ao projeto em questão, bem como, é necessário que o time tenha conhecimento das cerimônias e etapas desta metodologia, se aplicada.

Baseando-se na necessidade de adaptação do desenvolvimento de produtos digitais e na evolução de novas metodologias de gestão, buscou-se aplicar o *framework Scrum* em uma *startup*, que passou pela transição do escritório para o *home-office* e ainda não possui um método de gerenciamento de projetos definido. A questão de pesquisa desse trabalho pode ser definida da seguinte forma: Como melhorar a comunicação e fluxo de trabalho de um time de desenvolvimento de *webapps* que trabalha de forma remota e com equipe reduzida através do *framework Scrum*?

1.1 OBJETIVOS

Nesse tópico são apresentados os objetivos gerais e específicos do trabalho, que serviram como base para a consulta do referencial teórico e para o desenvolvimento.

1.1.1 Objetivo Geral

Aplicar a ferramenta *Scrum* no desenvolvimento de *webapps* de fidelização de clientes em uma *startup* de tecnologia de Florianópolis, no contexto pós-pandemia, onde toda a equipe trabalha de forma remota.

1.1.2 Objetivos Específicos

De forma a seguir um fluxo de conhecimento dos processos da empresa, para em seguida chegar ao objetivo geral, podemos definir objetivos específicos no decorrer do processo:

- a) Realizar mapeamento dos processos no fluxo de desenvolvimento;
- b) Gerar *product backlog* do projeto;
- c) Obter uma melhoria do fluxo de desenvolvimento através das ferramentas utilizadas;
- d) Comparar a visão da equipe sobre a comunicação e fluxos antes e depois da aplicação das ferramentas.

1.2 JUSTIFICATIVA

O Manifesto Ágil trouxe importantes visões do desenvolvimento de *Softwares*, principalmente mostrando que é possível iterar com os *stakeholders* no processo, focando em entregáveis que evoluem no decorrer do desenvolvimento, desde uma versão primária básica até chegar no produto funcional e completo. Porém, um dos doze princípios do Manifesto é priorizar a transmissão de informações face a face (Beck *et al.*, 2001). Necessita-se adaptar os princípios do *Agile* à realidade de uma empresa desenvolvedora de *Software* no pós-pandemia, onde os desenvolvedores residem em diferentes regiões do Brasil e o contato deixa de ser face a face, e passa a ser através de reuniões e trocas de mensagens *online*.

A comunicação e o tempo disponível da equipe são fatores que trazem diversas dificuldades no andamento dos projetos. É necessário ter uma visão geral do andamento das

tarefas, o que muitas vezes é feito de forma assíncrona através de mensagens no *workspace* da empresa. Esses processos podem não aproveitar o tempo dos desenvolvedores da melhor maneira, então uma oportunidade de melhorar a comunicação e o fluxo de tarefa é a aplicação dos princípios da metodologia ágil, bem como *frameworks* que permitam a desenvolvedores e demais integrantes da equipe enxergarem o projeto como um todo, além de “puxar” as tarefas específicas, numa espécie de *Lean* de tarefas.

A oportunidade de aplicação de metodologias ágeis, em específico o *Scrum*, em uma equipe que encontra impasses de comunicação e o contexto de trabalho remoto que implica em uma adaptação dos princípios *Agile* são os fatores que justificam esse trabalho.

2. REFERENCIAL TEÓRICO

A aplicação de conceitos de gerenciamento de projetos, passa pelo conhecimento e significado dos diferentes tipos de projeto, das metodologias avaliados e das dificuldades encontradas relacionadas à aplicação destas metodologias em ambientes de *startups*. Este capítulo descreve os principais conceitos necessários para compreender a estrutura do trabalho.

2.1 DESENVOLVIMENTO DE *SOFTWARES*

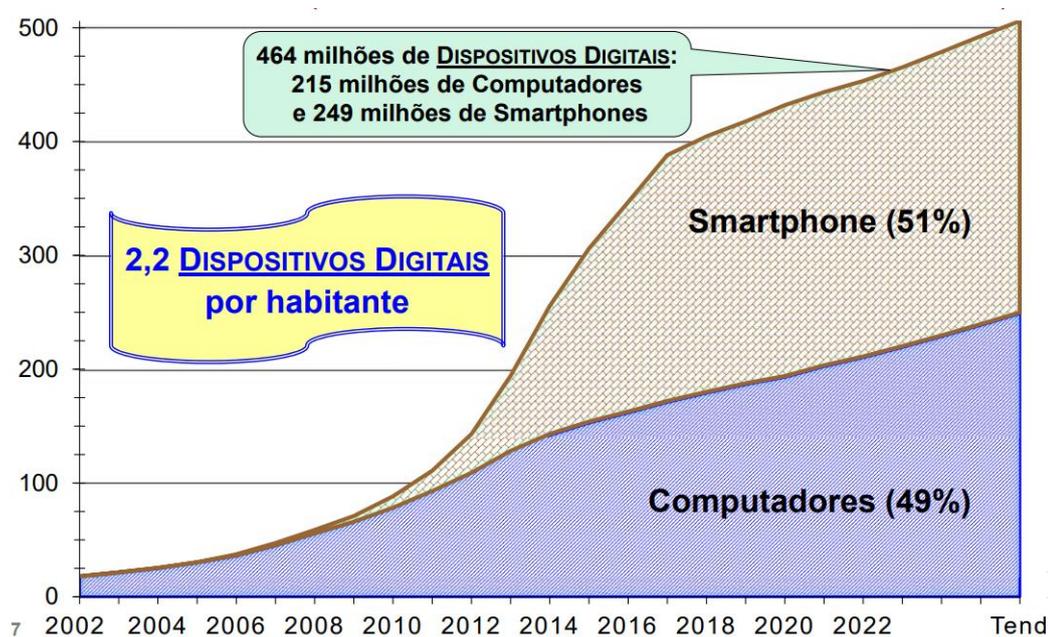
O desenvolvimento de softwares pode ser realizado com foco em diferentes tipos de softwares e também diferentes maneiras de se conduzir a sua gestão. Estes pontos são abordados neste tópico.

2.1.1 Evolução Tecnológica

O desenvolvimento de *Softwares* está intimamente ligado à evolução tecnológica que a humanidade conheceu no último século, principalmente após a Segunda Guerra Mundial. Ao pensarmos em tecnologia, temos a visão de supercomputadores e evoluções como a Inteligência Artificial que ganha espaço no mundo atual com sua “vida própria” através da automação de diversos sistemas interligados à banco de dados, mas a tecnologia esteve presente desde os primeiros seres-humanos que habitaram a Terra. De acordo com IPEA (2019), o desempenho científico e tecnológico da humanidade é o principal responsável por tudo que foi construído até hoje. O domínio das tecnologias está presente desde o domínio do fogo até a tecnologia moderna, sempre ligado à melhora na qualidade de vida da população. Segundo Lorenzetti

(2012), porém, a evolução tecnológica atual está além da produção industrial. “Podem-se citar inovações em serviços e no setor público e privado, a exemplo de negócios baseados na *Internet*, que mudaram a forma de vender produtos pelo uso da inovação tecnológica introduzida pela *Internet*” (Lorenzetti, 2012). Nesse contexto a utilização de *Softwares* ganha grande importância, pois segundo Fernandes (2023), “[...] *Software* é visto como um conjunto ordenado de descrições ou instruções, capazes de direcionar a máquina possuída pelo usuário (MPU) para a realização de tarefas que satisfazem às necessidades do último”. A utilização de *Softwares* com o domínio de computadores e celulares na sociedade veio de encontro às necessidades encontradas pelos usuários no decorrer dos anos, desde os primeiros *Softwares* desenvolvidos como calculadoras, passando pela máquina de Turing que quebrou a cifra alemã da Máquina Enigma, chegando aos computadores pessoais da IBM e Apple, e finalmente aos *smartphones* atuais. A pesquisa de uso de tecnologias de informação nas empresas, realizada pela FGV (2023), mostra a evolução da utilização de computadores e *smartphones* no Brasil nos últimos 20 anos, conforme a Gráfico 1.

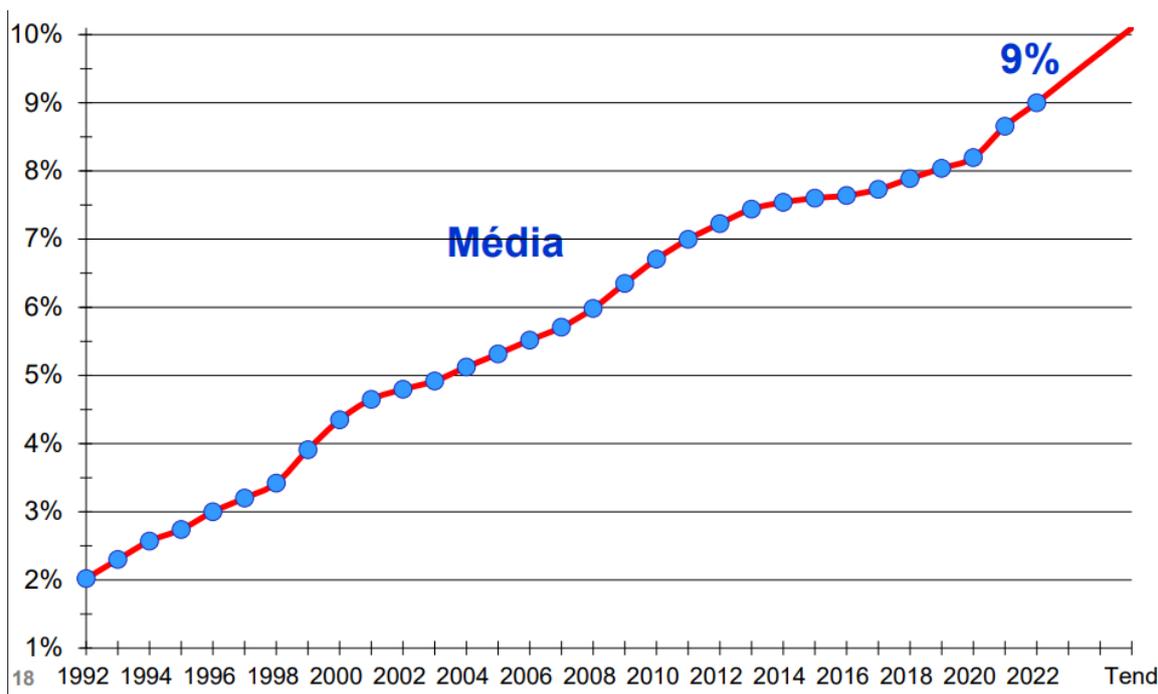
Gráfico 1 – Dispositivos digitais no Brasil até Maio/2023



Fonte: FGV (2023)

Outro ponto importante a se notar em relação aos *Softwares* no mercado, é o investimento que empresas dispõem para tecnologia, principalmente após a pandemia do COVID-19. A pesquisa de uso de TI em empresas, da FGV (2023) mostra essa evolução, com um impulso nos anos 2021 e 2022, conforme o Gráfico 2.

Gráfico 2 – Gastos de empresas com TI sobre faturamento



Fonte: FGV (2023)

Essa era uma realidade já presenciada no início da pandemia de COVID-19, de acordo com Paladini (2020):

[...] na verdade, a pandemia vai consolidar a tendência da competência digital. Apenas organizações que detém plataformas realmente eficazes sobreviverão, independentemente do porte delas. Todos (todos, sem exceção) os setores da economia formal (e até informal) vão ter que investir em modelos digitais de negócios.

Apesar disso, de acordo com a Associação Brasileira das Empresas de *Software* (ABES), o crescimento na produção de TI no Brasil em 2022 foi abaixo do esperado, apresentando um crescimento de 3% frente aos 7,4% mundiais, e com um investimento de 247.4 bilhões de reais. Mas, dentro desse mercado, o setor de *Software* foi o que manteve seu crescimento mais acentuado, principalmente por parte das empresas ainda atuarem em um sistema de trabalho home-office ou híbrido. (ABES, 2023)

2.1.2 Empresas desenvolvedoras de *Softwares*

A produção de *Softwares* tem algumas diferenças entre a produção de bens industriais e até mesmo de serviços, por isso existem diferentes maneiras de se olhar para esse setor. “O processo de desenvolvimento de *Software* difere dos processos de prestação de serviços e manufatura por causa das suas propriedades essenciais” (Fleury *et al.*, 2013). Kubota (2006),

também afirma que “o mercado de *Software* é complexo, pois abrange tanto serviços como produtos. E mesmo os produtos são atípicos: têm um caráter intangível, semelhante ao dos serviços”. Empresas desenvolvedoras de *Software* atuam em diferentes segmentos de acordo com o seu escopo. Existem empresas voltadas a produtos, que visam desenvolver sistemas para utilização em larga escala, e empresas mais voltadas ao serviço de desenvolver um sistema específico contratado por um cliente, visando atender apenas a sua necessidade.

Porém, ao compararmos com outros setores, Fleury *et al.* (2013) afirmam que as empresas de *Software* não se encaixam em desenvolvedora de produtos nem em prestação de serviços, pois não passam por alguns processos genéricos dos setores de manufatura e serviços. Os autores ainda destacam três diferentes tipos de empresas desenvolvedoras de *Software*, de acordo com o foco em que atuam:

- Empresas com foco em cliente: Projetos únicos para objetivos específicos de cada cliente.
- Empresas com foco em serviços: Alguns projetos que atendem uma gama maior de clientes.
- Empresas com foco em produtos: Poucos projetos, mas com pacotes suficientes para atenderem necessidades de grande parte dos clientes, visando comercialização em massa.

Em geral, empresas desenvolvedoras de *Softwares* focam em soluções para nichos diferentes no mercado, desde o sistema único para um cliente até sistemas para todo o mercado. Um exemplo famoso é o da Microsoft, a maior desenvolvedora de *Softwares* no mundo (Warren, 2023). A empresa é responsável pelo desenvolvimento do sistema operacional Windows, presente em 97% dos dispositivos empresariais, e do pacote Office, presente em 90% dos dispositivos empresariais (FGV, 2023). São números impressionantes que demonstram o domínio da Microsoft em *Softwares* de uso operacional e de uso integrado. A Microsoft é uma empresa que atua com foco em produtos, os seus projetos visam atender a grande maioria do mercado, criando pacotes completos nos seus *Softwares* que integrados formam uma conexão para suprir as necessidades das mais básicas até algumas complexas, sentidas pelos usuários. Para Fleury *et al.* (2013), a base do mercado de *Softwares* é composta pelas empresas focadas em produtos, pois são elas que compõem os aplicativos ou *Softwares* utilizados tanto pelos usuários finais, quanto como meio de produzir outras ferramentas pelas demais empresas. Já as empresas focadas em serviços e clientes, utilizam-se de produtos desenvolvidos por outros fornecedores, para realizarem seus novos desenvolvimentos específicos, atuando em nichos

menores. A Tabela 1 apresenta uma comparação entre os tipos de empresas desenvolvedoras de *Softwares*.

Tabela 1 – Características das empresas desenvolvedoras de *Softwares*

	Empresas orientadas a clientes	Empresas orientadas a serviços	Empresas orientadas a produtos
Descrição	Desenvolvedores de <i>software</i> de acordo com as especificações únicas do cliente	Prestadores de serviços relacionados com <i>software</i>	Desenvolvedores de produtos de <i>software</i> inovadores
Pesquisa e desenvolvimento	Portfólio de clientes	Portfólio de serviços	Portfólio de produtos
Sistemas desenvolvidos	Sistemas de <i>software</i> únicos, desenvolvidos de acordo com as especificações dos clientes	Serviços relacionados com <i>software</i> , incluindo: - Customização - Implantação - Treinamento - Operação	<i>Software</i> “pacote”, incluindo aplicativos, ferramentas, componentes e produtos complexos
Qualidade	Qualidade de processo	Qualidade de projeto	Qualidade de produto
Riscos principais	Mudança nos requisitos e duração do projeto	Custo e duração do projeto	Qualidade do produto final e duração do projeto
Foco do gerenciamento de projetos	Gerenciamento das expectativas do cliente	Gerenciamento das restrições do projeto	Gerenciamento da qualidade do produto
Treinamento	Programação e gerenciamento de configurações	Programação e modelagem de sistemas	Programação e testes de sistemas

Fonte: Fleury *et al.* (2013)

Um tipo de empresa que atua fortemente em tecnologia são as *startups*, e com isso, existem muitas *startups* desenvolvedoras de *Software*. Cagan (2021), define *startups* como uma nova empresa de produtos buscando um espaço no mercado antes que os recursos acabem, portanto existe um foco inicial muito forte em entregar um produto que atue nas necessidades dos consumidores de forma imediata. Com o produto aceito pelo mercado, o desafio passa a ser o de escalabilidade. Para Ries (2019), “uma *startup* é uma instituição humana projetada para criar um novo produto ou serviço sob condições de incerteza extrema”. O autor, porém, alerta que essa definição omite informações importantes como o tamanho da instituição, atividades, e fatores que tornam a definição de *startup* muito abrangente. Qualquer empresa assumindo um processo de alta incerteza se tornaria uma *startup*. Porém, um ponto chave a ser considerado é que *startups* atuam em inovação. “A inovação é um diferencial das *startups* em relação às empresas tradicionais, pois é o que movimenta o rápido crescimento desse modelo de negócio” (Palazi, 2022).

Esse contexto de inovação e incertezas, aliado a produtos que atendam às necessidades de nichos diferentes do mercado, estão atrelados ao desenvolvimento de *Softwares*. *Startups* carregam consigo a utilização da tecnologia como base para o desenvolvimento escalável de produtos, pois a utilização de tecnologia da informação e a internet permitem a replicação dos seus produtos à menores custos (ABSTARTUPS, 2023).

2.1.3 Tipos de *Softwares*

Em consequência às dinâmicas necessidades de novas soluções apresentadas pelo mercado, bem como pelas diferentes estruturas de empresas desenvolvedoras, surgem então alguns tipos de *Software*, cada qual com suas características. Fleury *et al.* (2013), apontam em sua pesquisa seis sistemas genéricos de *Softwares*:

- *Aplicativos*: *Software* que pode ser usado diretamente pelo cliente final sem necessidade de treinamento ou customização como, por exemplo, planilhas de cálculo ou processadores de texto;
- *Produtos complexos*: *Software* que precisa ser customizado antes de ser utilizado pelo cliente final como, por exemplo, sistemas de gestão empresarial;
- *Ferramentas*: *Software* que será utilizado para construir novos *Softwares*, como, por exemplo, aplicativos de desenvolvimento e editores de imagens;
- *Componentes*: *Software* que será incorporado na construção de outro *Software*, tais como rotinas e classes de objetos;
- *Software original*: *Software* especificamente desenvolvido para o cliente, de acordo com as suas especificações únicas; e
- *Software customizado*: produto complexo adaptado para o cliente, pronto para entrar em operação.

Rezende e Abreu (2013) classificam de forma semelhante os tipos de *Software*. Eles destacam a utilização de *Softwares* de rede para a integração de diversas fontes de informação, os *Softwares* de sistema operacional que “podem ser vistos como o administrador geral do computador” (Rezende e Abreu, 2013); os *Softwares* aplicativos que unem diferentes instruções através de uma linguagem de programação para realizar tarefas específicas; *Softwares office* que auxiliam na automação de tarefas de escritório, os utilitários que atuam como um suporte dos demais *Softwares* como antivírus, compactadores e afins. O autor também faz o registro dos *Softwares* de automação industrial que trouxeram maior precisão, tecnologia e segurança às indústrias através de máquinas com comandos programáveis.

Conforme descrito anteriormente, aplicativos são *Softwares* destinados à utilização pelo cliente final. Hoje em dia todo smartphone possui uma loja de aplicativos vinculada ao seu sistema operacional, onde é possível fazer o download dos mais diversos tipos de app, desde jogos até ferramentas digitais para o trabalho. Nesse cenário de aplicativos, surgiram os *Webapps* ou PWA (*Progressive Web Application*), que são *Softwares* aplicativos utilizados em

navegadores web, sem necessidade de download para o seu uso. Souza e Cintra (2021), definem: “[...] o PWA é uma combinação de uma aplicação web com uma aplicação *mobile*, utilizando o melhor de ambas”. Essa visão também é relatada por Neto (2020), que afirma que os aplicativos *web* se assemelham em interface com aplicativos tradicionais, mas tem o objetivo de utilizar os recursos oferecidos pelos navegadores.

Os *webapps*, portanto, apresentam algumas vantagens em relação aos aplicativos convencionais, conforme descreve Murakami (2023):

- Melhora a experiência do usuário com rapidez, responsividade e funcionamento *offline*.
- Fácil de atualizar, sem a necessidade de atualizar a loja de aplicativos.
- Menor custo de desenvolvimento e manutenção, pois pode ser criado com tecnologias *web* padrão.
- Fácil de ser descoberto através dos mecanismos de busca.
- Mais seguro, pois é executado em HTTPS para manter a segurança dos dados do usuário e do aplicativo.

Dessa maneira, algumas empresas desenvolvedoras de *Software*, principalmente voltadas a aplicativos, estão optando por seguir o padrão PWA. Além das vantagens citadas, também pesa o conceito de *Mobile First*, isso é, aplicativos que priorizam os *layouts* e funcionalidades para usuários *mobile*, em contrapartida aos *desktops*. Conforme visto anteriormente na pesquisa da FGV, existe hoje em dia mais de um smartphone por pessoa no Brasil (Meireles, 2023), e a responsividade dos aplicativos em PWA, isto é, adaptação do *layout* do sistema aos diferentes tamanhos de telas de celular, fazem com que esses aplicativos possam ser acessados e utilizados de qualquer aparelho *mobile*, qualquer navegador, permitindo o acesso desses aplicativos à um maior número de pessoas.

2.2 GERENCIAMENTO DE PROJETOS

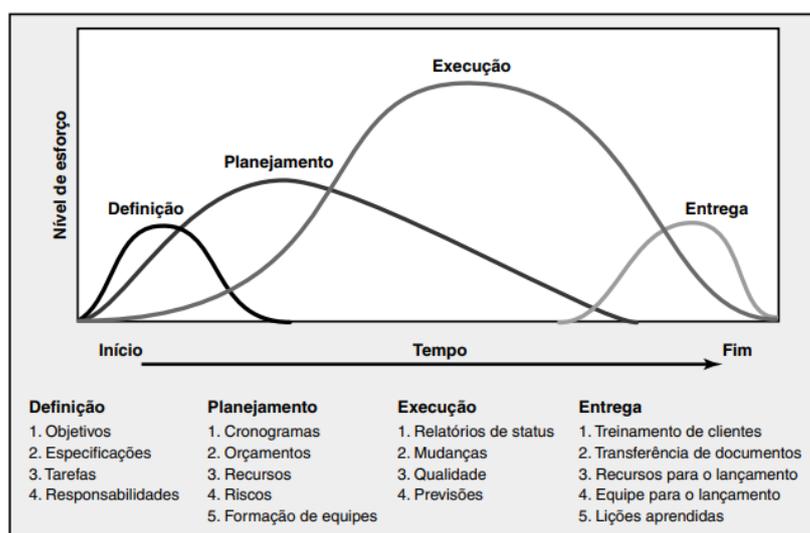
Wysocki (2009) define projeto como “uma sequência de atividades únicas, complexas e conectadas, que tem um objetivo ou propósito e que devem ser concluídas em um tempo específico, dentro do orçamento e de acordo com as especificações”¹. O Project Management Institute, PMI, também define no livro PMBOK (Project Management Body of Knowledge) que projetos diferem de operações ou processos por conter atividades únicas que tem um início e fim definidos. O gerenciamento de projetos, portanto, é “[...] a aplicação de conhecimentos,

¹ “a sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific time, within budget, and according to specification”.

habilidades, ferramentas e técnicas nas atividades do projeto a fim de atender aos seus requisitos” (PMI, 2004).

Gerenciamento de projetos, portanto, envolve a prática de diferentes métodos para atingir o seu objetivo, dentro do tempo estipulado, cumprindo os requisitos de projeto e dentro do orçamento. Um projeto passa por diferentes fases no seu desenvolvimento, chamado de ciclo de vida do projeto, conforme Gray e Larson (2009): “O ciclo de vida de um projeto passa comumente por uma sequência em quatro fases: definição, planejamento, execução e entrega”. Os autores também mostram que os esforços em cada etapa do ciclo de vida do projeto variam ao longo do tempo, visto na Figura 1.

Figura 1 – Ciclo de vida do projeto



Fonte: Gray e Larson (2009)

Wysocky (2009) também define cinco fatores importantes no ciclo de vida de um projeto, que em diferentes combinações definem seu resultado: escopo, qualidade, custo, tempo e recursos. Projetos que demandam qualidade, em geral dispendem maior tempo e mais recursos, e conseqüentemente, maiores custos. Por outro lado, projetos podem ser feitos de forma rápida e com menos recursos, mas pecam em qualidade. Por isso a utilização de métodos de gerenciamento de projetos é importante para trabalhar com essas fases do ciclo de projeto, ao mesmo tempo em que alguns recursos são limitados.

O gerenciamento de projetos tradicional, conforme descrito por Wysocky (2009), está enraizado desde os anos 1950, e segue um modelo linear, onde o planejamento tem grande importância pois as atividades de desenvolvimento do projeto são completamente conhecidas. Berni (2010) afirma que “os métodos clássicos têm foco maior no planejamento e controle e

qualquer alteração no planejamento é considerada uma anomalia que precisa ser corrigida”. Conhecendo os métodos para se chegar ao resultado desejado, o planejamento é essencial para garantir que todos os passos seguirão corretamente, bem como estimar os custos e tempos de desenvolvimento, depois disso, a parte de execução segue de forma regular. Um exemplo do modelo tradicional de projetos é a construção civil, onde os cálculos de fundação, tensões, materiais, métodos de trabalho e demais fatores são conhecidos há anos, e fazem com que o planejamento seja longo e burocrático, mas após isso, a execução segue (em teoria) uma sequência padronizada de processos. Em outras palavras, existem poucas incertezas.

O desenvolvimento de *Softwares*, por outro lado, é rodeado de incertezas, conforme Berni (2010):

O estilo antecipatório ou prescritivo para o desenvolvimento de *Software*, representado pelas metodologias planejadas ou tradicionais como o modelo de desenvolvimento em cascata, desenvolvimento evolucionário [...], dentre outras, pregam a definição dos requisitos de *Software* nas etapas iniciais do processo de desenvolvimento. Ou seja, devem-se prever com antecedência todas as prováveis iterações que irão ocorrer ao longo da execução do projeto. Essas técnicas são muito importantes para definição de custos, riscos e demais contingências que possam influenciar o andamento do projeto, mas raramente são capazes de prever com precisão todas as questões relacionadas ao processo de *Software*.

O autor ainda descreve que por esses motivos, novas metodologias de gerenciamento para o desenvolvimento de *Softwares* surgiram, pois, o processo não deve estar associado a prazos e cronogramas apertados, mas sim deve fluir de uma forma criativa. Nesse contexto de um processo interativo, o gerenciamento ágil surge como uma proposta de desenvolvimento de *Softwares*. O gerenciamento ágil, apesar de inicialmente ser concebido para o desenvolvimento de *Softwares*, está sendo utilizado nas mais diversas áreas, isso por que ele é útil em projetos onde não temos objetivos definidos inicialmente, assim como os requisitos de projeto (Neto, 2020)

Algumas práticas do desenvolvimento ágil foram compiladas e publicadas em 2001, dando mais conhecimento ao assunto a nível mundial, e que ganhou força nos anos seguinte. O “manifesto ágil” (Beck *et. al*, 2001), como ficou conhecido esse movimento, descreve em sua publicação “estamos descobrindo maneiras melhores de desenvolver *Softwares*, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo”. E ele também definem quatro princípios de valorização no desenvolvimento de *Softwares*:

- Indivíduos e interações mais que processos e ferramentas;
- *Software* em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Segundo Wazlawick (2013), “os modelos ágeis de desenvolvimento de *Software* seguem uma filosofia diferente da filosofia dos modelos prescritivos. Em vez de apresentar uma ‘receita de bolo’ [...], eles focam em valores humanos e sociais”. Ou seja, os conceitos de gerenciamento ágil priorizam a interação, o *feedback* e a colaboração com o cliente, respondendo a mudanças ao longo tempo, aprimorando o projeto ao invés de seguir uma sequência estrita e ordenada. Essas interações permitem a evolução dos produtos a serem desenvolvidos, erros podem ser identificados e corrigidos no início do processo. O manifesto ágil (Beck et al, 2001) destaca doze princípios seguidos pelos desenvolvedores de *Softwares* com gerenciamento ágil:

1. Prioridade em satisfazer o cliente com entregas contínuas e *Softwares* com valor agregado;
2. Mudanças durante o projeto são bem-vindas, sendo para entregar um produto melhor;
3. Fazer entregas de *Software* funcionando, por mais simples que sejam, em curtos espaços de tempo;
4. Desenvolvedores e outros setores devem trabalhar juntos diariamente;
5. Entregar ambiente e suporte necessário para manter a equipe motivada;
6. Conversas face a face são mais eficientes na troca e informações;
7. *Software* funcionando é a medida primária de progresso;
8. Os processos ágeis promovem desenvolvimento sustentável, a equipe deve manter um ritmo constante;
9. Atenção constante ao design e requisitos técnicos tornam as entregas mais ágeis;
10. Simplicidade é essencial;
11. As melhores arquiteturas e soluções surgem de equipes auto-organizáveis;
12. A equipe reflete sobre como ser mais eficaz e ajustar o comportamento e intervalos regulares.

Alguns modelos de gerenciamento ágil surgiram ao longo do tempo. Wazlawick (2013) destaca o Feature-Driven Development (FDD), que foi apresentado em 1997 e possui apenas duas fases: 1 a 2 semanas de concepção e planejamento, e 1 a 2 semanas de desenvolvimento iterativo do *Software*. O FDD é composto por etapas que priorizam a identificação das funcionalidades desejadas no *Software*, e posteriormente, o desenvolvimento do design e a construção do *Software* cumprindo os requisitos de funcionalidades. O autor apresenta também o Dynamic Systems Development Method (DSDM), que é “um modelo ágil baseado em desenvolvimento incremental, com participação ativa do usuário”. O DSDM consiste de três grandes fases: pré-projeto, ciclo de vida, pós projeto. Sendo que o ciclo de vida envolve a análise de viabilidade, análise de negócio, as iterações do desenvolvimento e, por fim, a implantação. Este método “procura iniciar pelo estudo e implementação dos 20% dos requisitos que serão mais determinantes para o sistema como um todo” (Wazlawick, 2013). O autor destaca também que a iteração com o cliente, a autonomia dos desenvolvedores e as entregas iniciais com os pontos mais importantes do projeto são fatores importantes para determinar a eficácia do método. Outras metodologias descritas pelo autor serão abordadas de forma mais aprofundada a seguir.

Segundo a *State of Agile Report*, pesquisa publicada pela *State of Agile* (2022), 80% dos respondentes afirmam utilizar metodologias ágeis em seus projetos, sendo as mais utilizadas o *Scrum*, *Kanban*, *Iterative* e *XP*, e também alguns híbridos entre essas metodologias. Os subcapítulos seguintes fornecem informações sobre os métodos mais utilizados.

2.2.1 Extreme Programming (XP)

De acordo com Pressman (2010), “o XP inclui um conjunto de regras e práticas que ocorrem no contexto de quatro atividades de arcabouço: planejamento, projeto, codificação e teste”. Wazlawick (2013), salienta que o XP também segue uma série de valores, como “simplicidade, respeito, comunicação, *feedback* e coragem”. O autor destaca que os valores do XP são importantes em projetos de *Software*, que requerem coragem para executar as mudanças necessárias e simplicidade para desenvolver o que realmente será utilizado. Além disso, respeito, comunicação e *feedback* andam lado a lado de forma a trazer a interação saudável entre equipe e clientes.

Planejamento: etapa onde são escritas as histórias de usuário. “[...] as histórias de usuários devem ser escritas pelos usuários considerando-se que elas são os itens de que eles precisam que o sistema faça pra eles” (Wazlawick, 2013). Estas histórias de usuário são uma

descrição das atividades esperadas pelo usuário final escritas pelo cliente, portanto, os desenvolvedores conseguem desenvolver sabendo exatamente o que o usuário deseja fazer/ver no sistema. A equipe XP avalia histórias entre uma e três semanas de desenvolvimento, e elas são priorizadas juntamente com o cliente, a fim de tomar a decisão de quais atividades devem ser concluídas até a próxima entrega combinada com o cliente (Pressman, 2010). O XP prioriza entregas, então as histórias maiores devem ser subdividas, com objetivo de desenvolvimentos funcionais concluídos nas datas combinadas entre as partes. Além disso, planejar com muita antecedência pode ser considerado desfavorável, devido às dinâmicas do desenvolvimento, “planejamento just-in-time é uma forma de estar sempre sintonizado com as mudanças de requisitos e arquitetura” (Wazlawick, 2013).

Projeto: Os projetos em XP devem seguir a lógica de simplicidade, e fornecer apenas as informações necessárias para a implementação das histórias. Caso problemas complexos sejam encontrados, é recomendado imediatamente de uma solução operacional desta parte do projeto. Refatorações também são encorajadas no XP, isto é, produzir um código melhor, mais eficiente, mas sem alterar a percepção do usuário final. A utilização de cartões CRC (classe, responsabilidade, colaboração), para identificar responsabilidades e colaborações de uma atividade a ser desenvolvida (Pressman, 2010).

Codificação: Pressman (2010) e Wazlawick (2013) apontam que a codificação deve ser precedida pela definição dos testes unitários, isso otimiza o tempo de codificação para ser desenvolvido apenas aquilo que foi planejado e de acordo com os testes estipulados. Os autores também definem que a codificação em XP deve ser feita em pares, duas pessoas codificando na mesma máquina. Isto faz com que haja uma dupla verificação do que está sendo desenvolvido, e sempre seguindo as padronizações necessárias no código para que toda a equipe tenha conhecimento do que foi feito. Essas etapas desenvolvidas devem ser incorporadas no projeto, o que é chamado de integração, então cada dupla de desenvolvedores integra sua parte ao projeto maior, evitando que os demais integrantes da equipe atuem em códigos desatualizados. Para isso ocorrer, é estimulado a integração constante, adicionando ao projeto de forma frequente as pequenas partes de código produzidas.

Teste: conforme descrito na etapa de codificação, testes unitários são definidos antes do início da codificação. Então, estes testes unitários devem ser executados constantemente, idealmente se de forma automatizada. Os testes unitários são validações de pequenas etapas do código escritas, demonstrando que um efeito ou atividade desejada está sendo realizada. Após os testes unitários devem ser feitos testes de integração, isto é, testes com o código integrado ao projeto como um todo. Testes também podem ser criados ao se notar um novo erro ou

problema, um comportamento inesperado pode ser sanado ao criar novos testes a partir destes erros. Por fim, testes de usabilidades são executados, estes são testes funcionais a partir da perspectiva da história do usuário, ou seja, são testes que garantem que está sendo entregue o que o usuário espera para aquela atividade (Pressman, 2010; Walzawick, 2013).

O Quadro 1 apresenta de forma resumida os principais pontos da metodologia XP.

Quadro 1 – Principais pontos da metodologia XP

Etapa	Resumo
Planejamento	Escrever histórias de usuário juntamente com o cliente, definir a prioridade das atividades, estipulando tempos de uma a três semanas de desenvolvimento.
Projeto	Focar na simplicidade e entrega de itens funcionais. Recomendado a utilização de cartões CRC e estimulado refatorações.
Codificação	Definir testes unitários antes da codificação. Recomenda-se a codificação em pares e a integração frequente do código.
Testes	Executar testes unitários, testes de integração, criar novos testes em caso de identificação de erros inesperados, finalizar com os testes de usabilidade baseados nas histórias de usuários.

Fonte: elaborado pelo autor

2.2.2 *Scrum*

O *scrum* é uma metodologia ágil criada nos anos 90 por Jeff Sutherland juntamente com sua equipe, e que utiliza muitos princípios do manifesto ágil como a aceitação a mudanças, produzir incrementos que podem ser testados e expandidos, testes e documentação durante o desenvolvimento, visando maximizar a comunicação entre a equipe (Pressman, 2010).

O Guia do *Scrum* (Sutherland e Schwaber, 2013), fornece informações sobre a metodologia. Os autores definem o *scrum* como:

Scrum é um framework estrutural que está sendo usado para gerenciar o desenvolvimento de produtos complexos desde o início de 1990. *Scrum* não é um processo ou uma técnica para construir produtos; em vez disso, é um *framework* dentro do qual você pode empregar vários processos ou técnicas. O *Scrum* deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las.

O *scrum* é baseado nos processos empíricos e na iteração e incremento tradicionais do Agile, e apoiado em três pilares: transparência, inspeção e adaptação. Transparência para que todos os participantes do processo tenham noção dos aspectos definidos. Inspeção para garantir que os processos ocorrem corretamente. Adaptação para corrigir possíveis desvios encontrados nas inspeções (Schwaber; Sutherland, 2013).

2.2.2.1 Papeis do Scrum

O *Scrum* é composto por algumas cerimônias, processos e diferentes papeis na equipe. O time *Scrum* é composto de perfis multifuncionais e flexíveis, de modo que possam tomar decisões e se adaptar durante o processo. Dentro do time *Scrum* destacam-se três papeis:

Product Owner: responsável por entregar valor ao produto. O “dono do produto” descreve as atividades compostas no *backlog*, conforme os requisitos que ele avaliou com o cliente, e também pode definir as prioridades. Em outras palavras, o *product owner* converte os requisitos do cliente em atividades claras e com prioridades para o time de desenvolvimento trabalhar (Schwaber; Sutherland, 2013).

Scrum master: responsável por garantir o entendimento do *Scrum* a toda a equipe. Wazlawick (2013), descreve o *Scrum master* como: “[...] não é um gerente no sentido dos modelos prescritivos. Não é um líder, já que as equipes são auto-organizadas, mas um facilitador [...] e um solucionador de conflitos”. O *Scrum master* auxilia o *Product Owner* a criar itens de *backlog* de maneira clara, além de mostrar maneiras de trabalhar com a equipe de desenvolvimento respeitando os eventos e praticando agilidade. Auxilia também o time de desenvolvimento à trabalhar sua interdisciplinaridade e autogerenciamento, criando produtos que entreguem valor e facilitando os eventos *Scrum* (Schwaber; Sutherland, 2013).

Desenvolvedores: são as equipes responsáveis pelo desenvolvimento e entrega dos incrementos ao final de uma “*Sprint*”. Equipes de desenvolvimento são autogerenciáveis e decidem, como um todo, a melhor maneira de construção do incremento que será feito, respeitando os requisitos do cliente anteriormente descritos. As equipes devem ser diversificadas, contendo pessoas com diferentes habilidades para que se construa aquilo que está designado para a equipe. Os times de desenvolvimento não são recomendados à serem menores do que 3 pessoas, pois pode haver restrições de conhecimentos necessários no desenvolvimento, ao mesmo tempo em que times com mais 9 pessoas pode necessitar de maiores esforços de coordenação. Portanto, times entre 3 e 9 pessoas de diferentes áreas interagindo entre si é o mais habitual (Schwaber; Sutherland, 2013).

2.2.2.2 Backlog do produto

“As funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de *product backlog*” (Wazlawick, 2013). “O *Backlog* do Produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões” (Schwaber; Sutherland, 2013). Portanto, o *backlog* é onde todas as informações do produto são listadas, e transformadas em atividades que serão desenvolvidas. Schwaber e Sutherland (2013), porém, alertam que um *backlog* não é criado com todas as atividades definidas, ao contrário disso, ele evolui com o produto.

Refinamento: “[...] é a ação de adicionar detalhes, estimativas e ordem aos itens no *Backlog* do Produto. Este é um processo contínuo em que o *Product Owner* e o Time de Desenvolvimento colaboram nos detalhes dos itens do *Backlog* do Produto” (Schwaber; Sutherland, 2013).

Kniberg (2007) define seis campos importantes que uma atividade deve conter no *backlog*:

1. Id: identificação única da atividade em formato numérico
2. Nome: descrição curta da atividade
3. Importância: um fator que demonstre a importância daquela atividade em comparação às demais. A escala não é importante, desde que fique claro que números maiores representam atividades mais importantes.
4. Estimativa inicial: estimativa do esforço a ser empreendido na atividade. Normalmente essa estimativa é determinada em *story points*, onde cada ponto representa uma pessoa trabalhando integralmente entre 6 e 8 horas naquela atividade. Dessa maneira, se uma atividade será desenvolvida por 3 pessoas ao longo de 4 dias, estimam-se $3 \times 4 = 12$ pontos.
5. Como demonstrar: descrição de qual o comportamento esperado ao final da atividade para ser considerada concluída.
6. Notas: demais descrições e anotações que possam ser importantes na história.

Quadro 2 – Demonstração de *backlog* de produto

Product Backlog					
ID	Nome	Imp.	Story points	Como demonstrar	Notas
1	Depósito	30	5	Fazer login, abrir página de depósito, depositar \$10,00, ir para página de saldo, verificar que o depósito foi incrementou \$10,00 no saldo	Necessita um diagrama de sequência UML. Não necessita se preocupar com criptografia por enquanto.
2	Ver histórico de transações	10	8	Fazer login, clicar em “transações”, fazer um depósito. Voltar às transações e checar se um novo depósito aparece.	Utilizar paginação para evitar grande pesquisa no Banco de Dados. Design similar à visualização de páginas de usuário

Fonte: adaptado de Kniberg (2007)

2.2.2.3 Eventos do Scrum

Os eventos no *Scrum* são ferramentas que visam manter o padrão de transparência e adaptabilidade do time, definindo reuniões com tempos fixados para discussão de diversos pontos desde o planejamento do desenvolvimento até as revisões pós desenvolvimento. Esses padrões evitam que o time desperdice tempo em outras reuniões de decisão no decorrer do processo, garantindo ao time de desenvolvimento as decisões necessárias para produzirem os incrementos durante seu tempo de trabalho (Schwaber; Sutherland, 2013).

Sprint: “o *Sprint* é o ciclo de desenvolvimento de poucas semanas de duração sobre o qual se estrutura o *Scrum*” (Walzawick, 2013). Schwaber e Sutherland (2013) definem o *Sprint* como “o coração do *Scrum*”. Os autores afirmam que *sprints* são pequenos projetos com duração de cerca de um mês, onde ao final desse *Sprint* o objetivo da equipe é entregar algo “pronto”. As tarefas a serem executadas durante o *sprint* são definidas em uma reunião de planejamento, onde as atividades descritas no *backlog* do produto passam a fazer parte do *backlog* do *Sprint*, isto é, a lista de atividades e suas descrições que deverão ser concluídas no *Sprint*, desde que o tempo estimado de execução dessas tarefas pela equipe compreenda o espaço de tempo planejado para o *sprint*. Ao concluir um *Sprint*, sequencialmente outro é iniciado, retomando o ciclo de planejamento e entregas.

Plannings: As reuniões de planejamento são o momento onde todo o time *scrum* discute quais atividades devem entrar em desenvolvimento no próximo *Sprint*. Estas reuniões abordam dois tópicos principais: o que pode ser entregue como resultado ao final do *Sprint* e

como isso será desenvolvido. O time de desenvolvimento é responsável por elencar quantas atividades consegue concluir dentro do período da *Sprint*, enquanto o *product owner* define a prioridade das atividades e qual o objetivo final esperado como resultado do *Sprint*, com isso as atividades passam do *backlog* do produto para o *backlog* do *Sprint*. Após a tomada de decisão das atividades a serem desenvolvidas, a equipe de desenvolvimento pode decompor as atividades do *Sprint* em tarefas menores a fim de obter maiores detalhes e trabalhar na sua auto-organização de como prosseguir com as tarefas ao longo do *Sprint*. Ao final, espera-se que a reunião de planejamento consiga trazer o *backlog* do *Sprint*, o que vai ser entregue, os esforços necessários e maiores detalhes das atividades. (Schwaber; Sutherland, 2013).

Daily: “[...] o objetivo é fazer que cada membro da equipe fale sobre o que fez no dia anterior, o que vai fazer no dia seguinte e, se for o caso, o que impede de prosseguir” (Wazlawick, 2013). As reuniões diárias têm o objetivo de verificar com todo o time *scrum* o andamento das atividades durante o *sprint*, de forma com que todos os membros falem o que fizeram e o que pretendem fazer num espaço de 24 horas, além de possíveis problemas que encontraram e o impediram de continuar na sua atividade. O tempo da reunião é de 15 minutos, apenas para atualizações rápidas entre a equipe. Essa reunião é fundamental no *scrum* pois diariamente a equipe tem o contato com os demais membros e consegue acompanhar as atividades em andamento, evitando assim, que outras reuniões ocupem tempo de desenvolvimento. Muitas vezes, porém, a equipe pode se reunir após as *dailys* para ter discussões técnicas sobre algum desenvolvimento ou problema encontrado (Schwaber; Sutherland, 2013).

Review: A reunião de revisão é feita ao final de cada *Sprint*, com o objetivo de revisar o que foi desenvolvido e validar se o andamento das atividades, bem como o resultado esperado, foram concluídos de acordo com o planejamento. É uma reunião que pode contar com a presença de *stakeholders* para discutir o que foi entregue, e tende a ser mais leve, pois pretende motivar e trazer colaboração à equipe (Schwaber; Sutherland, 2013). Os autores definem elementos que devem ocorrer na *review*:

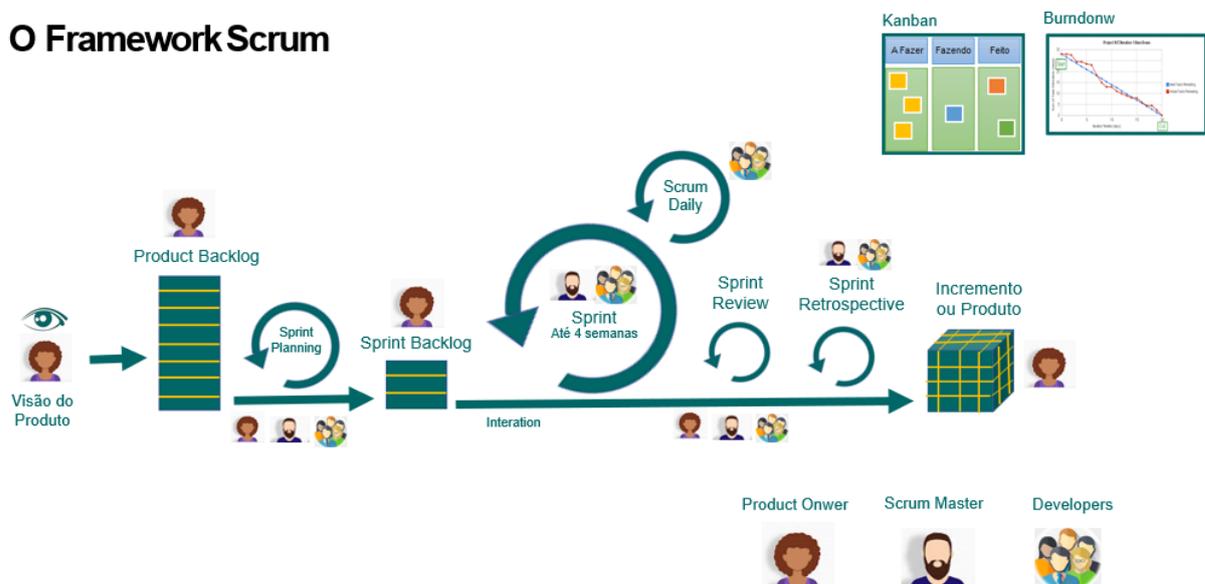
- Os participantes incluem o Time *Scrum* e os *Stakeholders* chave convidados pelo *Product Owner*;
- O *Product Owner* esclarece quais itens do *Backlog* do Produto foram “Prontos” e quais não foram “Prontos”;
- O Time de Desenvolvimento discute o que foi bem durante a *Sprint*, quais problemas ocorreram dentro da *Sprint*, e como estes problemas foram resolvidos;

- O Time de Desenvolvimento demonstra o trabalho que está “Pronto” e responde as questões sobre o incremento;
- O *Product Owner* discute o *Backlog* do Produto tal como está. Ele (ou ela) projeta as prováveis datas de conclusão baseado no progresso até a data (se necessário);
- O grupo todo colabora sobre o que fazer a seguir, e é assim que a Reunião de Revisão da *Sprint* fornece valiosas entradas para a Reunião de Planejamento da próxima *Sprint*;
- Análise de como o mercado ou o uso potencial do produto pode ter mudado e o que é a coisa mais importante a se fazer a seguir; e,
- Análise da linha do tempo, orçamento, potenciais capacidades, e mercado para a próxima versão esperada do produto.

Retrospectiva do *Sprint*: As reuniões de retrospectiva ocorrem após uma *review* e antes de uma *planning*, e tem como objetivo discutir os processos e o relacionamento da equipe durante o *Sprint*. Estas reuniões visam identificar processos que ocorreram bem e devem ser mantidos, e também processos que falharam e necessitam de melhorias. As melhorias devem ser identificadas e traçados planos de ação para mitigar as falhas encontradas, tornando os próximos *sprints* mais eficazes e agradáveis para o time *scrum* (Schwaber; Sutherland, 2013).

Figura 2 – Framework *Scrum*

O Framework *Scrum*



Fonte: Silva (2023)

2.2.2.4 Resumo scrum

Conforme as descrições dos eventos e dos perfis encontrados no método *Scrum*, o quadro 3 traz de forma objetiva as informações encontradas no tópico 2.2.1.2, enquanto a figura 2 ilustra o ciclo das *sprints*.

Quadro 3 – Papeis, eventos e atributos do *Scrum*

Papeis do Scrum	
Product Owner	Responsável pelo produto, faz a ponte entre os requisitos dos clientes e time de desenvolvimento. Inclui as atividades no <i>backlog</i> e auxilia na priorização.
Scrum master	Responsável por garantir o entendimento do <i>Scrum</i> e manter o time <i>Scrum</i> seguindo as cerimônias da metodologia.
Time de desenvolvimento	Equipe geralmente entre 3 e 9 pessoas, responsáveis pela execução dos itens do <i>backlog</i> . Composta de pessoas com diferentes habilidades, capazes de cumprir os requisitos e possuir um auto-gerenciamento
Artefatos do Scrum	
Product Backlog	Lista de itens com requisitos para o projeto, inicialmente conta apenas com os requisitos principais e vai se decompondo no decorrer do projeto.
Sprint Backlog	Lista dos itens do <i>backlog</i> que serão desenvolvidos durante o <i>Sprint</i> . Precisam passar por refinamento para conter mais detalhes e informações pertinentes a atividade a ser desenvolvida.
Eventos do Scrum	
Sprint	Período onde itens do <i>backlog</i> são desenvolvidos pelo time de desenvolvimento, com duração entre 1 e 4 semanas, e com objetivo de promover uma entrega de incremento ao final do <i>Sprint</i> .
Planning	Reunião de planejamento onde são definidos os itens a serem executados no <i>Sprint</i> , o PO auxilia na priorização das atividades enquanto o time de desenvolvimento atua nas estimativas de esforço.
Daily	Reuniões diários de 15 minutos com objetivo de cada membro do time de desenvolvimento informar o que fez e o que irá fazer no projeto num

	período de 24 horas. Têm como objetivo manter a equipe ciente do andamento das atividades.
Review	Reuniões de revisão das atividades realizadas ao final de um <i>Sprint</i> , tem como objetivo discutir o que foi entregue e o que ficou pendente para o próximo <i>Sprint</i> .
Retrospectiva	Reunião realizada após a <i>review</i> , com objetivo de discutir os processos realizados e o relacionamento entre a equipe, com objetivo de traçar planos de ação de melhorias para trazer maior qualidade aos próximos desenvolvimentos.

Fonte: elaborado pelo autor

2.2.3 Método *Kanban*

De acordo Anderson e Carmichael (2016), “*Kanban* é um método para definir, gerenciar e melhorar serviços profissionais, atividades criativas e o *design* de produtos físicos e de *Software*”. O método *Kanban* deriva do sistema *kanban*, criado pela Toyota na década de 50, onde cartões eram utilizados para apresentar de maneira visual a situação de estoques de diferentes setores do chão-de-fábrica, promovendo assim a produção puxada. De acordo com o Guia Oficial do Método *Kanban* (*Kanban University*, 2021), o *Kanban* é confundido com metodologia ou framework na engenharia de *Software*, porém o *Kanban* é um método que deve ser trabalhado juntamente com outras metodologias:

Kanban não é uma metodologia nem um framework de processo. Trata-se, antes, de um método ou de uma abordagem de gestão que deve ser aplicada a um processo ou método de trabalho já existente. Nunca há uma questão de usar *Kanban* versus uma metodologia ou framework. Em vez disso, o *Kanban* sempre é adicionado a uma metodologia, framework, ou a maneira de trabalhar já existente. *Kanban* tem a intenção de ajudá-lo a gerenciar melhor o trabalho e melhorar a prestação de serviços ao ponto de atender consistentemente as expectativas dos clientes. *Kanban* é um meio para melhorar o que e como você já faz as coisas, e não um substituto.

“O método *Kanban* é baseado em tornar visível o que é de outra forma, trabalho intangível de conhecimento, para garantir que o serviço funcione com a quantidade adequada de trabalho” (Anderson e Carmichael, 2016). O *Kanban*, portanto, atua em tornar visual aquilo que está sendo feito, e garantir que as atividades sejam puxadas pelos responsáveis, movimentando os cartões nas diferentes raias, principalmente utilizando os quadros *Kanban*.

2.3 O ÁGIL NO TRABALHO REMOTO

A utilização de metodologias ágeis, conforme descrita por Beck *et. al* (2001), prioriza a comunicação face a face. Durante a pandemia do Covid-19, porém, inúmeras empresas precisaram se adaptar ao formato de trabalho remoto. Algumas dessas, que já utilizavam metodologias ágeis no seu dia-a-dia, precisaram alterar a forma de trabalho, enquanto outras utilizaram esse momento para a implementação do ágil. Diante disso, buscou-se conhecer o processo de adaptação do ágil no âmbito remoto, onde a comunicação precisa passar por mensagens de textos, reuniões virtuais, e-mails e ligações.

2.3.1 Dificuldades do Ágil no trabalho remoto

Maffei (2021), destaca em seu artigo algumas dificuldades encontradas na adaptação do trabalho presencial para o remoto em uma equipe que utiliza o *Scrum*. A autora menciona a necessidade de compra de materiais por conta dos funcionários, como computadores, cadeiras ergonômicas, fones de ouvido, e até a necessidade de aumentar a velocidade de internet para poder se conectar nas diversas reuniões que a equipe tem semanalmente. Outro ponto importante encontrado no trabalho foi a dificuldade da *Scrum master* em reunir toda a equipe, onde precisa criar *links* para encontros *online*, e depende da resposta da equipe para confirmação de disponibilidade, ou até mesmo para entrar em alguma reunião mais urgente.

A sobrecarga de alguns participantes de cada equipe também foi fator relatado por Maffei (2021), Generoso (2022) e Oliveira (2022). Os três autores apresentam informações de que integrantes da equipe, utilizando o *Scrum*, relataram sobrecargas de trabalho em algum momento. Seja pelo curto tempo de entrega de alguma tarefa, pela dificuldade de comunicação (Maffei, 2021), ou pela equipe de tamanho reduzido acarretando em diversas funções para a mesma pessoa dentro das cerimônias de cada metodologia (Generoso, 2022; Oliveira, 2022).

O estudo de Bataglia (2020), reúne outros três pontos importantes destacados por autores que publicaram sobre o assunto. Os três principais pontos de dificuldades apontados pelo autor são a falta de disciplina dos membros da equipe, atuando em horários incomuns de trabalho. A falta de digitalização das informações do projeto, deixando de usar ferramentas digitais que possam agregar valor ao projeto. E, por fim, a dificuldade de adaptação com as tecnologias disponíveis para o contato remoto da equipe e do controle das atividades por conta dos responsáveis.

Os desafios encontrados em trabalhos correlatos mostram a complexidade da adaptação dos membros de equipes em metodologias ágeis ao contexto remoto no que tange a usabilidade das tecnologias e sequência das cerimônias que antes ocorriam presencialmente. Porém na bibliografia também são encontradas sugestões de mitigar os problemas relatados e outras vantagens que a utilização do ágil no *home-office* apresentou.

2.3.2 Vantagens do ágil do trabalho remoto

Generoso (2022) e Oliveira (2022) destacam que a implementação do *Scrum* no ambiente de trabalho remoto em que se encontravam trouxeram algumas vantagens, relatadas pelos próprios participantes das equipes. Os pontos principais em destaque pelos autores foram a utilização do *backlog* do produto que trouxe maior organização ao projeto, e a utilização das cerimônias como as *daily*s e as *plannings* que trouxeram maior acompanhamento e redução de dúvidas no andamento das atividades.

A possibilidade de integração entre os desenvolvedores, permitindo uma ajuda mútua mesmo no ambiente remoto, foi apontada por Maffei (2021). Segundo a autora, os desenvolvedores conseguiram criar uma forma de ajuda, mesmo que por mensagens, permitindo que não haja desenvolvedor parado esperando a solução de algum problema em uma tarefa, gerando maior agilidade ao projeto.

Os autores supracitados destacam ainda um dos principais pontos do ágil: a possibilidade de uma alteração no projeto de acordo com os *feedbacks* recebidos após cada entrega, ou até mesmo no andamento de um *Sprint*. A comunicação com clientes e entre os membros da equipe, e a troca de informações pelos meios remotos ou durante as cerimônias do *scrum* permitiram às equipes a atualização de atividades a fim de atingir um requisito específico do produto levantado pelo cliente.

Apesar das dificuldades encontradas, os resultados dos trabalhos correlatos estudados foram em sua maior parte positivos. A utilização de metodologias ágeis em ambientes remotos trouxe maior organização, um horizonte com direcionamento da sequência de tarefas e auxiliou na comunicação entre membros da equipe. Além disso, foi possível atingir a dinâmica dos requisitos do produto através dos *feedbacks* recebidos durante o andamento ou conclusão das atividades.

3. METODOLOGIA

Este capítulo tem por objetivo apresentar o tipo de pesquisa, os métodos de abordagem no trabalho e a apresentação da empresa onde está sendo realizado o trabalho. A Figura 3 ilustra em fluxo da metodologia utilizada, que é mais aprofundada nos demais itens deste capítulo.

Figura 3 – Metodologia utilizada



Fonte: elaborado pelo autor

3.1 TIPO DE PESQUISA

Segundo Gil (2008), a pesquisa pura visa desenvolver os conhecimentos científicos, porém, sem envolvimento com a aplicação direta desses conhecimentos. Por outro lado, a pesquisa aplicada utiliza-se da pesquisa pura, mas também obtém os resultados da aplicação como conhecimento. Dessa forma, a pesquisa realizada nesse trabalho é considerada de natureza aplicada.

A classificação quanto aos objetivos da pesquisa é classificada como exploratória, pois têm como objetivo tornar um problema mais explícito utilizando ferramentas como a pesquisa bibliográfica e a entrevista com pessoas da área. (Gil, 2008)

Em relação à abordagem da pesquisa, o presente trabalho é considerado como abordagem qualitativa, pois segundo Miguel (2012), a abordagem qualitativa está inserida no meio de pesquisa, observando o ambiente e os indivíduos. A pesquisa, porém, de forma alguma deve desconsiderar dados quantitativos, mas sim, incorporar esses dados às observações, que podem oferecer informações importantes para o resultado.

O método de pesquisa é considerado pesquisa-ação, pois de acordo com Miguel (2012) a pesquisa-ação apresenta a interação entre pesquisador e o objeto pesquisado, produzindo conhecimento e concomitantemente resolvendo um problema real.

3.2 A EMPRESA

A LiHai Tech, empresa que é objeto de estudo desse trabalho, nasceu em 2022 após um processo de transição com outra empresa que englobava seus projetos, a Getmore, que nasceu em 2015 explorando o *cashback*. O *cashback*, em tradução literal do inglês, significa “dinheiro de volta”. O mecanismo consiste em devolver ao cliente uma pequena porcentagem do valor gasto por ele em seu comércio, poderia ser entendido como um desconto, mas se diferencia geralmente pela seguinte premissa: o dinheiro que você ganha de volta é utilizado em uma próxima compra no mesmo comércio. A empresa cresceu através do desenvolvimento de serviços de *cashback* e fidelização para plataformas digitais, entregando desde a criação de *layout*, estrutura do programa até os suportes técnicos. Situou-se com escritório em Florianópolis, com cerca de 30 funcionários, atendendo clientes como Bradesco, Itaú, Ambev, Ultragaz, Polishop, entre outros. Em 2021 a Getmore foi vendida. Nesse cenário, um dos seus criadores deu sequência à dois projetos que não estavam mais ligados à *cashback*, e sim às mecânicas de fidelização, nascendo assim a LiHai Tech.

3.2.1 Situação da empresa

Durante o processo de transição dos projetos entre as *startups*, entre 2021 e 2022 houve redução de grande parte da equipe, que passou a ser gerida pela equipe da nova empresa compradora. Os projetos desenvolvidos pela empresa também reduziram de escopo, restando dois projetos ativos com clientes, e a intenção de um novo desenvolvimento próprio. Novos desenvolvedores foram contratados nesse processo de transição, onde ainda possuíam contato com os desenvolvedores anteriores, para entrarem no contexto dos projetos. Por fim, no final de 2022 a empresa se estabeleceu definitivamente, findando o processo de transição de documentações e demais configurações operacionais que antes tinham relação com a empresa Getmore.

3.2.2 O produto da empresa

A empresa realiza desenvolvimento de *webapps* para fidelização de clientes através do acúmulo e resgate de pontos e também de *gamificação*. Para ficar mais claro, é necessário entender o conceito de fidelização e *gamificação*.

Fidelização: Segundo a TOTVS (2022), “Programa de fidelidade é um conjunto de ações empreendidas pela empresa com o objetivo de recompensar, reter e fidelizar os clientes. Para isso, as companhias oferecem benefícios para quem mantém ou aumenta sua frequência de compras em seus estabelecimentos.”. Desta forma, a empresa que adota programas de fidelização utiliza estratégias de benefícios e recompensas para atrair os seus clientes, e se tornar mais vantajosa em relação aos concorrentes. Segundo o Panorama da Fidelização no Brasil 2022, uma pesquisa realizada em todo o país pela Tudo Sobre Incentivos (2022), 73% dos entrevistados veem nos programas de fidelidade a oportunidade de receber benefícios por uma compra que precisam fazer e 66% entendem que é uma forma de economizar dinheiro. São informações que impactam na hora da tomada de decisão sobre onde realizar a sua compra.

Gamificação: Segundo a FIA (2022), a *gamificação* é adaptado do inglês – *gamification* – que utiliza técnicas de jogos em situações de não-jogo, adaptando características de jogos em um ambiente distinto, com objetivo de motivar e melhorar a experiência do usuário. É uma forma de atrair os usuários em tarefas ou atividades que fazem normalmente, mas de forma a atribuir metas, pontos, *rankings*, elementos normalmente encontrados em jogos, mas que permite o simples ato de realizar uma compra, por exemplo, se tornar uma forma evoluir em uma categoria e obter benefícios futuros. O grande exemplo disso são as diferentes categorias, e suas respectivas vantagens, para os pontos acumulados em passagens aéreas.

Dessa forma, a empresa se baseia em projetos que entregam valor tanto ao usuário final, quanto ao cliente que está solicitando o desenvolvimento destas plataformas. Hoje em dia três projetos estão em andamento, conforme o Quadro 4.

Quadro 4 – Projetos em andamento na empresa

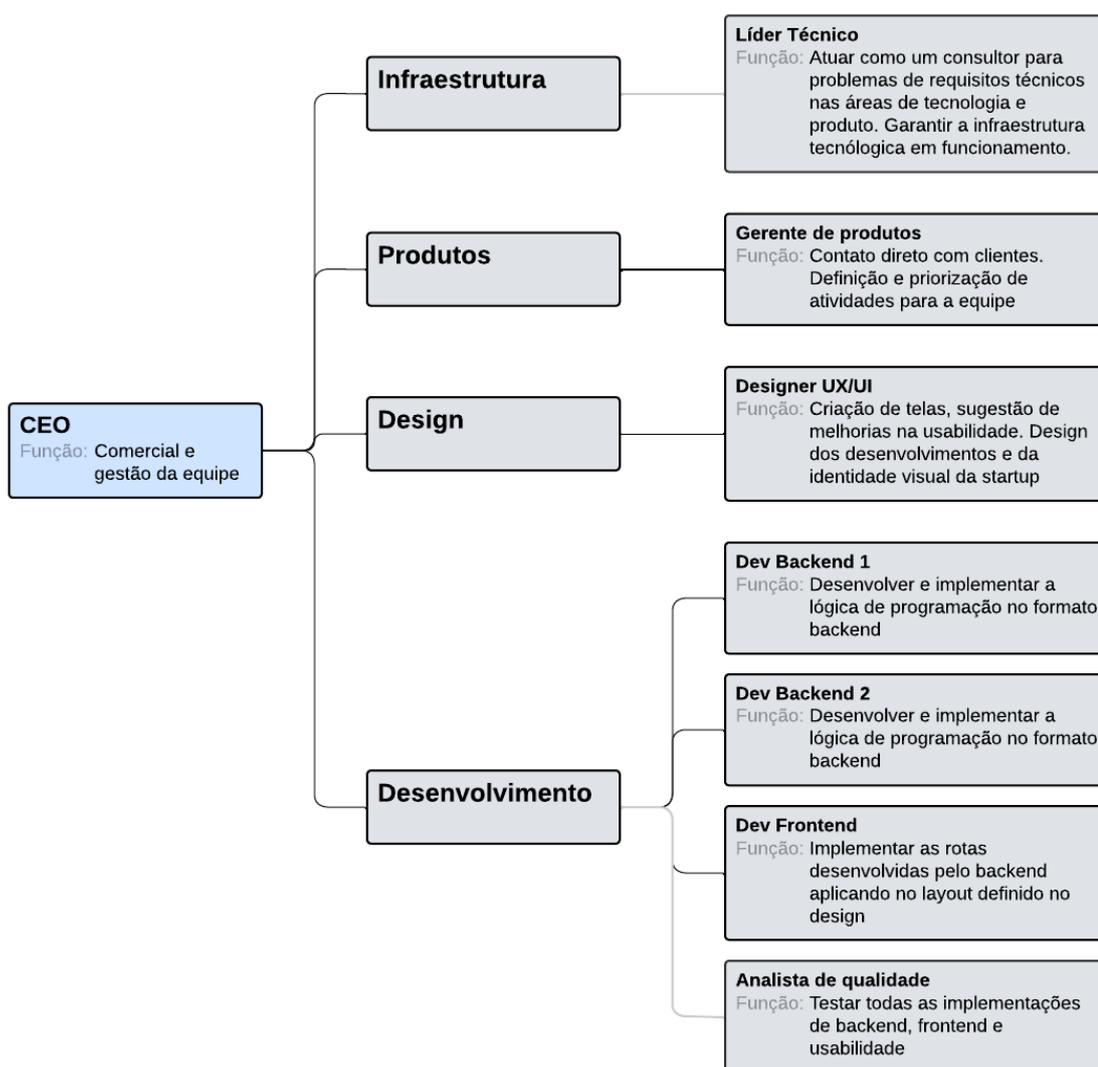
Nome	Descrição	Tipo	Situação
Projeto Diamante	Fidelização através de desafios onde revendedores de gás de uma marca nacional acumulam pontos no decorrer do ano. Ao final do ano, os melhores são premiados.	Gamificação	Desenvolvimento concluído, atuando em suportes e melhorias
Projeto Clube Ciência	Fidelização através da atribuição de pontos com a lógica “a cada 100 reais em compras ganhe 5 pontos”. Os pontos acumulados podem ser resgatados por vouchers de desconto ou brindes.	Recompensas	Desenvolvimento em conclusão, aplicando melhorias antes de finalizar em definitivo.
Projeto Loyalty	Desenvolvimento de plataforma dinâmica onde os administradores do programa podem definir regras, recompensas e desafios para o usuário final.	Ambos segmentos	Definição de escopo e início de versões iniciais para validar hipóteses.

Fonte: elaborado pelo autor

3.2.3 Estrutura organizacional

A *startup* encontra-se com uma equipe que trabalha junto há cerca de um ano, tem pouca rotatividade e os desenvolvedores muitas vezes intercambiam entre os projetos, ajudando onde há mais necessidade no momento. O proprietário e fundador da *startup* é responsável pela área comercial, documentações e contratos, e pelo bom andamento dos projetos. O restante da equipe é encontrado no organograma da Figura 4.

Figura 4 – Organograma da empresa



Fonte: elaborado pelo autor

3.3 PROCEDIMENTOS PARA O DESENVOLVIMENTO DO TRABALHO

O desenvolvimento do trabalho passa por alguns processos de identificação das dificuldades encontradas, mapeamento de processos, definição do escopo e aplicação do *Scrum*. Primeiramente, entrevistas semiestruturadas e um questionário com perguntas no modelo de escala *Likert* foram respondidas pela equipe LiHai. Posteriormente foram definidos os processos atuais da empresa, as dificuldades encontradas nos processos, tanto na visão de fora, quanto na visão dos participantes após a resposta à pesquisa. A partir daí o *Scrum* é aplicado em um escopo restringido a apenas um dos três projetos da empresa. Por fim, o registro das implementações é realizado, com objetivo de conhecer a eficiência das ferramentas aplicadas.

3.3.1 Entrevistas com a equipe

Entrevistas com os integrantes da empresa foram utilizadas com o objetivo de verificar internamente quais processos estão correndo bem, e quais apresentam maiores dificuldades. Essas informações são importantes para definir onde há maior necessidade de atuação, e assim, verificar quais técnicas podem ser utilizadas a fim de mitigar os erros encontrados. As entrevistas ocorreram de forma semiestruturada, pois este formato permite ao entrevistador fazer perguntas pré-estabelecidas, mas ainda assim abrir a conversa para perguntas adicionais a fim de garantir novas informações do entrevistado (Coelho, 2020).

As entrevistas seguiram com todos os participantes atuais da equipe da empresa e o CEO, que também pode fornecer informações importantes pela “visão de fora”, de alguém que está acompanhando as tarefas, mas não está de fato desenvolvendo. A estrutura das perguntas seguiu o fluxo de informações e tarefas na empresa. O contexto das entrevistas pode ser visto no Quadro 5.

Quadro 5 – Contexto das entrevistas

Contexto	Diante da possibilidade de aplicação de metodologias ágeis na startup, conhecer os principais desafios da empresa e onde/como aplicar esses métodos
Objetivos	Encontrar as oportunidades de melhoria através de metodologias ágeis
Metodologia	Entrevistas qualitativas semiestruturadas
Participantes	Funcionários da empresa LiHai Tech

Dúvidas	<ul style="list-style-type: none"> • Quais processos apresentam gargalo? • Quais processos apresentam riscos de segurança para a empresa? • Quais processos apresentam falhas que afetam o desempenho da equipe? • Onde é possível ganhar agilidade? • Onde é possível atribuir ações de documentação e qualidade ao desenvolvimento do <i>Software</i>? • Onde é possível agir para entregar mais valor ao produto?
----------------	--

Fonte: elaborado pelo autor

O roteiro pré-definido das entrevistas contempla as seguintes perguntas:

1. Há quanto tempo você trabalha na LiHai Tech?
2. Qual sua área de atuação?
3. Há quanto tempo você atua nessa área (mesmo que em outras empresas)?
4. Quais as principais dificuldades que você encontra para realizar seu trabalho internamente?
5. Quais as principais dificuldades que você encontra para realizar o seu trabalho em conjunto com a equipe?
6. Existem processos que considera que travam ou atrapalham o fluxo do andamento das atividades (gargalos)? Quais?
7. Existem processos que considera como fator de risco em relação à segurança de dados, servidores e informações sensíveis?
8. Existem processos que considera como negativos em relação ao desempenho da equipe?
9. Considera que há clareza na definição e andamento das atividades?
10. Considera suficiente a forma de registro das atividades atuais, futuras e passadas da empresa? (Registro de informações, logs, *backlogs*...)
11. Considera o tempo diário em que atua como suficiente para cumprir as atividades no prazo?
12. Em relação as atuais cerimônias realizadas pela empresa, considera que são úteis ao bom andamento das atividades e à boa convivência em equipe? (Atualmente temos *weekly* e *daily*, além da utilização do *Jira* e *drive*)
13. Acredita que existem pontos de oportunidade para ganhar agilidade dentro do fluxo? Onde? Como?
14. Acredita que existem pontos de oportunidade para melhorar a documentação das atividades? Onde? Como?

15. Acredita que existem pontos de oportunidade para melhorar a qualidade dos desenvolvimentos? Onde? Como?
16. Considera que existem possíveis processos, dinâmicas, técnicas que podemos aplicar para agregar valor aos desenvolvimentos?
17. Existem outras sugestões ou pontos de atenção que deseja destacar?

Além das perguntas abertas, perguntas de resposta “quantitativa” foram usadas com objetivo de realizar uma comparação entre início e fim da aplicação dos métodos ágeis. Esse formato seguiu um modelo de escala *Likert*, conforme mostra a Figura 4. Os valores foram mostrados em uma média geral da equipe, e a possibilidade de verificar as informações da entrevista aberta, com os números obtidos na média da escala *Likert*, auxiliaram na decisão dos processos a serem utilizados. Para não influenciar as respostas dos participantes, essa pesquisa foi realizada de forma anônima através dos formulários do *google*. As perguntas e a escala de resposta são visualizadas na Figura 5.

Figura 5 – Modelo do questionário com escala *Likert*

Pesquisa processos Lihai

Pesquisa com objetivo de encontrar pontos com oportunidades de melhoria nos processos da LiHai e atuar com planos de ação nesses quesitos. Também será possível realizar a comparação dessas informações após aplicação de alguns processos de metodologia ágil, verificando se de fato houve uma evolução em relação ao fluxo de processos, ambiente, documentação e comunicação dentro da LiHai. A pesquisa é anônima, não há coleta de e-mail.

felipe.kuse@gmail.com [Mudar de conta](#) 🔒 Rascunho salvo.

🔒 Não compartilhado

* Indica uma pergunta obrigatória

Como considera o ambiente e a convivência da equipe? *

Muito ruim 1 2 3 4 5 Muito bom

Como você considera a clareza do fluxo de atividades a serem realizadas? *

Muito ruim 1 2 3 4 5 Muito bom

Fonte: elaborado pelo autor

3.3.2 Mapeamento de processos

Após isso foi feito o mapeamento de processos atual da empresa, que juntamente com os dados qualitativos obtidos na etapa anterior, serviu como base para entender o fluxo atual de informações dos projetos e poder encontrar possíveis gargalos ou processos ineficientes, seja desperdiçando tempo, recursos ou até mesmo com deficiência nas trocas de informações.

Por fim, foram definidos os principais problemas encontrados e foi feita a implementação do *Scrum*, então nesse caso é preciso definir o time *scrum*, com o *Product Owner*, *Scrum master* e os desenvolvedores, e também seguindo o cronograma de reuniões e criando o *backlog* do produto.

3.3.3 Aplicação de metodologia *Scrum*

Após as entrevistas e a definição de quais processos podem ser utilizados, foi aplicado ao dia-a-dia da empresa as cerimônias do *Scrum*, seguindo as dinâmicas características do método. Algumas adaptações podem ser observadas devido às necessidades do home-office e da disponibilidade da equipe, servindo também como forma de verificar se as metodologias selecionadas são válidas em ambientes remotos ou apenas para o modelo de trabalho presencial.

3.3.4 Entrevistas finais e comparação de resultados após aplicação do *Scrum*

Por fim, novamente as entrevistas com os participantes foram feitas com o objetivo de comparar, na visão dos participantes, se houve uma evolução no andamento das atividades e nos pontos encontrados como gargalo nos processos, após a utilização do *Scrum*. De forma qualitativa, pode-se avaliar as respostas nas questões abertas. E de forma quantitativa, pode-se comparar os valores encontrados na primeira versão do questionário com a escala *Likert*.

4. RESULTADOS

Este capítulo tem como objetivo apresentar o desenvolvimento da metodologia definida no capítulo anterior, juntamente com a revisão bibliográfica do capítulo 2. Dessa maneira, o capítulo apresenta a execução das entrevistas com a equipe de desenvolvedores, designer e CEO da LiHai, a verificação dos processos encontrados na literatura para a metodologia ágil em

desenvolvimento de *Software* e o comparativo entre eles, verificando se há aplicabilidade no contexto da empresa e a visualização dos fluxos de processos encontrados no início do trabalho e após as mudanças aplicadas. O desenvolvimento em ordem cronológica apresenta dados e figuras do que foi aplicado, e por fim, com uma nova pesquisa entre os integrantes da equipe, espera-se comparar os resultados com as informações iniciais e analisar se houve ou não, de fato uma melhoria nos processos diários da empresa.

4.1 SITUAÇÃO INICIAL

A empresa, nos últimos meses, vinha buscando definir metodologias e processos para utilizar como padrão no seu dia-a-dia. Houve uma migração das plataformas de registro de atividades passando pelo Asana, GitHub e Notion. Também, a criação de atividades, definição de responsáveis e sequência de atividades a serem seguidas também possuía grande variação. Alguns processos estavam sendo utilizados misturando técnicas do ágil de diferentes segmentos, como o *Kanban* e reuniões diárias do *Scrum*, mas que na prática, muitas vezes se assemelhavam aos métodos em cascata devido à pouca iteração nas atividades, e a demora para concluir entregáveis.

4.1.1 Busca por processos ágeis

A empresa buscava no passado a utilização de algumas formas de metodologia ágil, porém, sem nenhuma com processo de fato sendo seguido à risca. Uma das técnicas encontradas eram as *Daily Scrums*, onde a equipe se reunia online diariamente, durante cerca de 15 minutos, para discutir as atualizações dos 3 projetos que são executados atualmente. A *daily* passou por diversas mudanças de horário devido à disponibilidade da equipe, visto que a maior parte exerce atividades paralelas. Essa questão apresentou uma dificuldade em manter a constância das participações, tendo agendas com apenas 2 ou 3 participantes da equipe, e sem a dissipação das informações conforme se esperava.

Outro processo utilizado eram os testes com um analista de qualidade, onde após qualquer desenvolvimento, apenas uma pessoa visualizava o funcionamento da nova *feature*, tanto no código nativo (*backend*), quanto na usabilidade das plataformas (*frontend*). Em muitos momentos esses testes não se mostraram efetivos, pois a inclusão do analista de qualidade no final do processo, sem entender todo o contexto da atividade, muitas vezes fazia com que o analista aprovasse ou reprovasse um desenvolvimento sem analisar todos os cenários. Esse

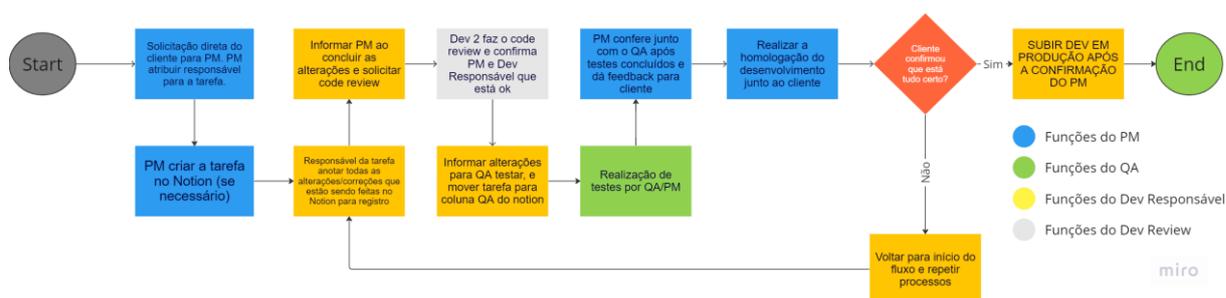
problema também estava atrelado a descrição das atividades, que algumas vezes não continham os critérios de aceite para ser checados. Apesar desse ponto ser defendido no modelo XP, que afirma que além dos testes unitários, os testes de usabilidade realizados devem ser baseados na história do usuário, a falta de mensuração de diferentes cenários apresentava a aprovação de atividades que poderiam apresentar erros no futuro, ao encontrar um cenário não testado.

Assemelhando-se também ao XP, um dos processos implementados na codificação foi o “*code review*”, onde uma nova atualização do código só é aplicada após a revisão de um segundo desenvolvedor. Esse processo tem como objetivos reduzir erros de codificação, gerar padronização da linguagem utilizada, dar agilidade nos desenvolvimentos evitando o retrabalho. Esse processo se apresentou como vantajoso no tempo observado, porém notou-se que em alguns momentos de pressa para subir alterações, os *code reviews* eram aprovados sem a revisão minuciosa das alterações, ou seja, apenas um processo a mais na codificação envolvendo dois desenvolvedores sem trazer os benefícios esperados.

4.1.2 Fluxo de processos no momento inicial

O mapa de processos da Figura 6 apresenta o fluxo definido pela empresa no ano de 2023 para atividades solicitadas por clientes. É possível verificar que o processo passa em diversos momentos pelo Gerente de Produtos (PM), e que o andamento dos processos ocorre de forma manual, onde a cada etapa, é necessário acionar um desenvolvedor, ou Gerente de Produtos, ou Analista de Qualidade (QA). Nesse processo é possível ver a utilização da plataforma *Notion*, onde um quadro *Kanban* é utilizado para definir o status de cada atividade, porém a simples movimentação no quadro não é suficiente, fazendo com que a comunicação entre os responsáveis ocorra de forma desajustada através dos chats de comunicação. Apesar da sugestão de anotação das alterações no *Notion*, na prática, as conversas em agendas e chats para tirar dúvidas acabavam sendo úteis para resolução de problemas, porém poucos registros eram feitos.

Figura 6 – Mapa de processos atual



Fonte: elaborado pelo autor

4.2 ENTREVISTAS

As entrevistas com os participantes foram realizadas entre os dias 11/04/24 e 18/04/24, com o objetivo de identificar de forma qualitativa as dificuldades encontradas nos processos adotados pela LiHai naquele momento, e buscando encontrar lacunas onde o *Scrum* pode atuar e ser mais efetivo. Através do questionário anônimo, houveram 5 respondentes: o CEO, 1 designer e 3 desenvolvedores *Backend*. O desenvolvedor *Frontend* estava de férias e não participou das entrevistas. Além disso, o analista de qualidade foi desligado dias antes, então também não participou das entrevistas e acabou acarretando em mudanças nos fluxos de atividades.

4.2.1 Formulário com escala *Likert*

Na Tabela 1 encontram-se as perguntas feitas e as respostas dos 5 participantes. O registro foi feito visando estabelecer valores de comparação do antes e depois da aplicação do *Scrum*. Podemos perceber que as únicas questões que seguem uma linearidade maior são em relação ao tempo de entrega as atividades e à convivência da equipe. As demais questões possuem maior variação na visão da equipe. Na Tabela 2 essa comparação fica mais evidente.

Tabela 1 – Resultado do questionário da situação inicial da empresa

Pergunta	Resp	Resp	Resp	Resp	Resp
	1	2	3	4	5
Como considera o tempo para a entrega das atividades?	5	4	4	5	4
Como considera a importância das atividades desenvolvidas? (Em relação à entrega de valor ao cliente)	5	5	5	3	5

Como você considera a clareza do fluxo de atividades a serem realizadas?	4	3	5	5	2
Como você considera o andamento das atividades da equipe? (tempos, informações da atividade, comunicação durante a execução)	5	4	4	4	3
Como você considera o registro das informações de atividades futuras?	5	3	4	4	2
Como você considera o registro das informações das atividades que estão sendo executadas?	5	4	3	5	2
Como você considera o registro de atividades passadas?	5	3	3	4	2
Como você considera o registro de processos e informações padrão da empresa?	4	5	3	3	2
Como considera o ambiente e a convivência da equipe?	5	5	5	4	4
Como considera a comunicação assíncrona da equipe?	4	5	4	1	4
Como considera a execução dos eventos que temos atualmente?	5	5	4	3	3

Fonte: elaborado pelo autor

De forma resumida, comparando a maior e menor resposta e a média, podemos ver que há uma discrepância em alguns aspectos. As perguntas relacionadas ao registro das atividades obtiveram as menores médias, destacando um problema sentido pela equipe. Além disso, a comunicação das atividades também foi um ponto onde, em média, houve uma menor satisfação, sendo a única pergunta onde obteve uma resposta com o valor mínimo. Desta forma, a transição da informação, tanto em questões assíncronas, quanto na descrição e fluxo das atividades seguidas, merece atenção.

Tabela 2 – Resumo do resultado do questionário da situação inicial da empresa

Pergunta	Maior	Menor	Média
Como considera o tempo para a entrega das atividades?	5	4	4,4
Como considera a importância das atividades desenvolvidas? (Em relação à entrega de valor ao cliente)	5	3	4,6
Como você considera a clareza do fluxo de atividades a serem realizadas?	5	2	3,8
Como você considera o andamento das atividades da equipe? (tempos, informações da atividade, comunicação durante a execução)	5	3	4,0
Como você considera o registro das informações de atividades futuras?	5	2	3,6
Como você considera o registro das informações das atividades que estão sendo executadas?	5	2	3,8
Como você considera o registro de atividades passadas?	5	2	3,4

Como você considera o registro de processos e informações padrão da empresa?	5	2	3,4
Como considera o ambiente e a convivência da equipe?	5	4	4,6
Como considera a comunicação assíncrona da equipe?	5	1	3,6
Como considera a execução dos eventos que temos atualmente?	5	3	4,0

Fonte: elaborado pelo autor

4.2.2 Entrevistas semiestruturadas

Nas entrevistas semiestruturadas, através do roteiro definido anteriormente, foi possível entender mais a fundo a percepção da equipe em relação aos processos, convivência da equipe, problemas ou sugestões. Três desenvolvedores foram entrevistados, o desenvolvedor *Frontend* estava de férias, o designer não pode participar da agenda marcada.

Baseado nas respostas dos desenvolvedores, conforme o Apêndice, podemos perceber que a LiHai é o primeiro contato empregatício dos 3, e que ambos tinham um contato prévio com a programação anteriormente, mas não em relação à trabalho. Todos também já integram a equipe há 2 anos ou mais, tendo conhecimento dos projetos e processos adotados pela empresa. Em relação às dificuldades encontradas, internamente o maior problema foi a gestão ou falta de tempo. Esse fator tempo também foi mencionado e influenciou nas dificuldades encontradas em atuar com a equipe. A falta de comunicação ou agendas para tirar dúvidas acaba sendo um impeditivo para manter o bom andamento das atividades.

Em relação aos gargalos das atividades, as respostas não seguiram um padrão. Cada desenvolvedor trouxe um ponto relevante, entre não testar diferentes cenários por conta da pressão de entrega e acabar tendo que retrabalhar em uma tarefa, ou a falta de *feedback* de clientes que acaba travando o andamento da tarefa – fugindo totalmente dos conceitos do manifesto ágil, que defende constantes *feedbacks*. Ainda nesse ponto de dificuldades encontradas, a falta de disponibilidade e um pouco da distância da equipe foram mencionadas como fatores que são negativos para o fluxo, o que acarreta em muitas vezes, marcar agendas com a equipe fora do horário de trabalho, que tem impacto na vida pessoal dos participantes.

4.3 DEFINIÇÃO DO ESCOPO

O objetivo deste tópico é definir um projeto para a aplicação do *Scrum* dentro da empresa, visto que a empresa atualmente trabalha com dois projetos voltados para cliente, e um projeto próprio com objetivo escalável. Os dois projetos com clientes, estão em fases mais

avançadas, sendo um deles que já está em funcionamento há três anos e apenas são feitos suportes e manutenções evolutivas. O projeto de outro cliente está em fase de finalização, pois os ambientes estão prontos e funcionais, apenas alguns ajustes de usabilidade solicitados pelo cliente estão em andamento para poder lançar ao público.

O projeto próprio é a evolução dos serviços ofertados aos clientes atualmente. Porém, necessita reescrever o código adaptando e melhorando os problemas encontrados em projetos anteriores. A ideia consiste em uma ferramenta de Fidelização dinâmica, onde os administradores possam configurar as regras de recompensa e resgate. Isso permite a escalabilidade do produto, pois pode ser maleável a diferentes tipos de empresas, e concede aos administradores o controle do seu programa de fidelidade. Além de recompensas por transações (compras e-commerce ou sistêmicas), o programa deve oferecer recompensas baseadas em *gamificação*, como cumprir metas dentro da plataforma.

Desde 2023 este assunto vem sendo tratado na LiHai. Em alguns momentos foram definidas as funcionalidades esperadas do Novo Produto, e a definição de requisitos mais gerais, como o dinamismo e controle do administrador. Recentemente foram definidas em equipe as principais atividades e metas para entregar o programa, que agora vem com foco total da equipe. Dessa forma, por ser uma sequência de atividades que está no início do seu desenvolvimento, e por contar com o foco principal da equipe, optou-se por reduzir o escopo da atuação de metodologias ágeis para este projeto em específico.

Com isso, os demais projetos da empresa seguem com o mesmo fluxo que já tem atualmente, visto que as atividades destes em grande parte são de suportes ou manutenções, não tendo uma sequência bem definida. Enquanto que projeto do Novo Programa de Fidelidade da empresa vai passar a seguir as cerimônias do *Scrum*.

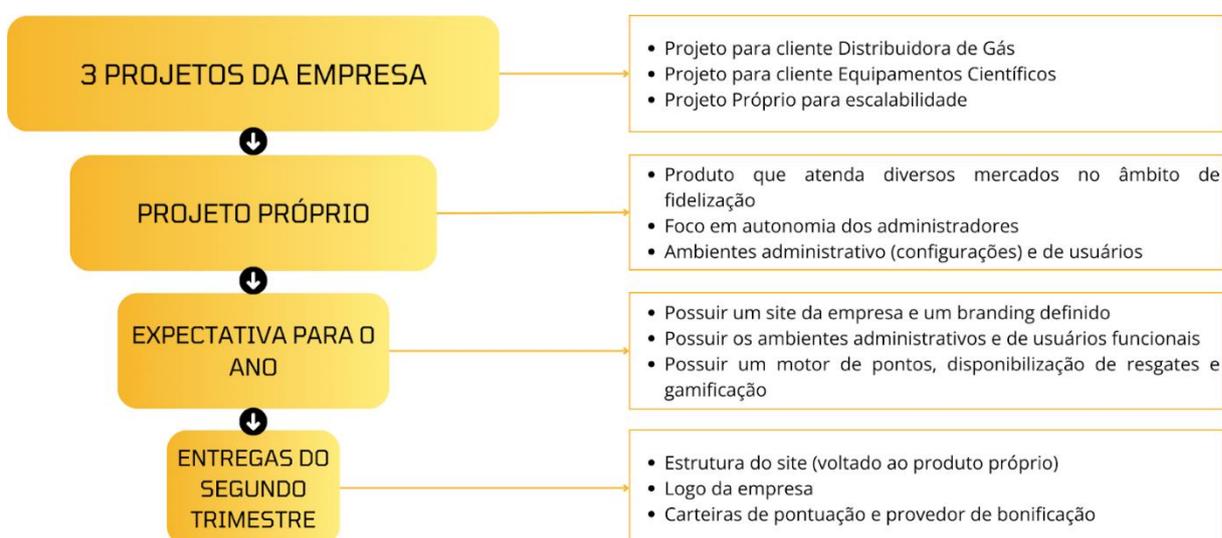
No final de março de 2024, a equipe toda se reuniu durante 3 dias em Florianópolis. Como alguns desenvolvedores moram fora de Santa Catarina, o objetivo foi fazer um encontro descontraído entre colegas de trabalho, porém, o dia 22/03/2024 foi reservado para discutir o produto e quais as expectativas para o ano. Então, através de *brainstorming* e uma dinâmica de grupos, foram separados dois times que apresentaram a sua visão e expectativa do programa de fidelidade. Ao final, os dois grupos se uniram e definiram em conjunto as macro entregas, separadas por trimestre.

Essas entregas foram o ponto inicial da separação de atividades nas *sprints*, destacadas nos próximos tópicos. O foco inicial foi trabalhar as questões de marca da empresa em paralelo com as atividades de infraestrutura do programa. Apesar de as atividades de *branding* e *site* estarem definidas para o trimestre 1, isso demonstrava um objetivo da equipe, mas que

difícilmente seria realizado. Por tanto, no *backlog*, essas atividades acabaram se encaixando no trimestre 2. As demais atividades dos trimestres 3 e 4, foram deixadas em espera, justamente pelo foco das tarefas do trimestre 2, com a criação da carteira, motor de pontos e os ambientes de usuário e administrador. Portanto, essas atividades do trimestre 2, e as de *design* foram as escolhidas como um escopo para visualizar a implementação do *Scrum* inicialmente, visto que eram as prioritárias para serem trabalhadas no momento.

A Figura 7 ilustra a definição de escopo da aplicação do *Scrum*, reduzindo o foco dos três projetos da empresa, passando para um projeto em fase inicial, e priorizando a aplicação do *Scrum* nas entregas para o segundo trimestre de 2024.

Figura 7 – Definição de atividades para Trimestre 1 e Trimestre 2



Fonte: elaborado pelo autor

4.4 APLICANDO TÉCNICAS DO *SCRUM*

Mesmo antes da definição de quais os melhores processos, algumas mudanças já começaram a ser implementadas na empresa, visto a oportunidade e necessidade de melhorar a comunicação da equipe.

4.4.1 Time *Scrum*

Para iniciar a aplicação do *Scrum* na empresa, foi necessário definir o time *Scrum*. Conforme Schwab (2004), o time *Scrum* deve possuir todas as competências necessárias para trabalhar em atividades multifuncionais. O time se completa e se organiza para trabalhar de

maneira ágil, sem necessidade de apoio externo. Dessa maneira, por ser uma empresa com 8 funcionários, foi definida apenas uma equipe *Scrum*.

Os funcionários já possuíam um conhecimento prévio da metodologia, pois alguns já haviam trabalhado com este formato anteriormente, outros, tiveram contato através de cursos e também por ser uma cultura disseminada no ambiente de desenvolvimento tecnológico. Os perfis do time, conforme o Quadro 6, foram definidos pelo *Scrum master*, que é o responsável pela empresa. O Gerente de Produtos foi responsável pelo perfil de *Product Owner*. Os 4 desenvolvedores e o designer foram alocados no perfil do time de desenvolvimento.

Quadro 6 – Time *Scrum*

<i>Scrum master</i>	CEO da Empresa
<i>Product Owner</i>	Gerente de Produtos
Desenvolvimento	<ul style="list-style-type: none"> • Desenvolvedor Front • Desenvolvedor <i>Backend</i> 1 • Desenvolvedor <i>Backend</i> 2 • Líder Técnico • Designer <i>Branding</i> • Designer Site

Fonte: elaborado pelo autor

4.4.2 Planejamento a nível macro dos entregáveis no trimestre 2

Para acompanhar o andamento das atividades, baseado no escopo principal do projeto definido anteriormente para os três trimestres finais de 2024, foram separadas tarefas definidas nos objetivos e separadas por áreas. A decisão de como separar em tarefas foi feita em agendas entre o *Product Owner* e o Líder técnico. O líder técnico, por possuir o conhecimento técnico necessário, auxiliou com os detalhes de infraestrutura necessários para o andamento das atividades que envolviam criação de banco de dados, lógicas para funcionamento das regras, carteiras de pontos, ambientes de usuários. O *Product Owner* contribuiu com os desejos funcionais do programa, a fim de trazer valor ao produto final como usado pelos diferentes usuários (administradores e usuários finais). Além disso, o designer também apresentou suas divisões de atividades.

A divisão das atividades, então, ocorreu em 5 principais aspectos:

- Website da empresa
- *Branding* e *layout* para o programa de incentivos e fidelização

- Infraestrutura do programa
- Ambiente administrativo (*Backoffice*)
- Ambiente de usuários (*Member portal*)

Como cada área tem especificações diferentes, elas foram desdobradas em menores tarefas como aquilo que se espera ao final do Trimestre 2, que foi utilizado como *backlog* do produto, e posteriormente atribuído às *Sprints*. O Quadro 7 mostra as principais divisões de tarefas em cada área, conforme agendas citadas anteriormente:

Quadro 7 – *Backlog* em nível Macro do Segundo Trimestre

Site	Tempo	Responsável
<i>Benchmarking</i> e idealização do site	1 semana	Designer Site
Criar estrutura do site	1 semana	Designer Site
Desenhar modelo de site (<i>wireframes</i>)	1 semana	Designer Site
Compilar dados relevantes da empresa	1 semana	Designer Site
Criar página LiHai (aplicando dados anteriores)	1 semana	Designer Site
Aplicar identidade visual da empresa no site	1 semana	Designer Site
Infraestrutura	Tempo	Responsável
Criar lógicas de carteira	2 semanas	Líder Técnico
Criar lógicas de tier e <i>ranking</i>	2 semanas	Dev 1
Criar lógica de regras de campanhas	2 semanas	Líder Técnico
Brand	Tempo	Responsável
Logo e tipografia da empresa	4 semanas	Designer Brand
Refinamento da identidade visual	4 semanas	Designer Brand
Definição do Tom de Voz	1 semana	Designer Brand
Brandbook da empresa	8 semanas	Designer Brand
<i>Layout</i> e estruturas do <i>Backoffice</i>	2 semanas	Designer Brand
<i>Layout</i> e estruturas do <i>Member portal</i>	2 semanas	Designer Brand
Portal de usuários (<i>Member portal</i>)	Tempo	Responsável
Criar ambiente para <i>Member portal</i>	1 semana	Líder Técnico

Aplicar carteira e tier no <i>Member portal</i>	1 semana	Dev 1
Aplicar regras de campanhas no <i>Member portal</i>	1 semana	Dev 1
Criar cadastro de usuários no <i>Member portal</i>	1 semana	Dev 1
Aplicar <i>layouts</i> do Brand no Portal de Usuários	2 semanas	Dev Front
Portal Administrativo (<i>backoffice</i>)	Tempo	Responsável
Criar <i>backoffice</i> provisório (<i>windmill</i>)	2 semanas	Líder Técnico
Aplicar dados de usuários fictícios para teste	1 semana	Dev 1
Aplicar regras de campanhas no <i>windmill</i>	1 semana	Dev 1

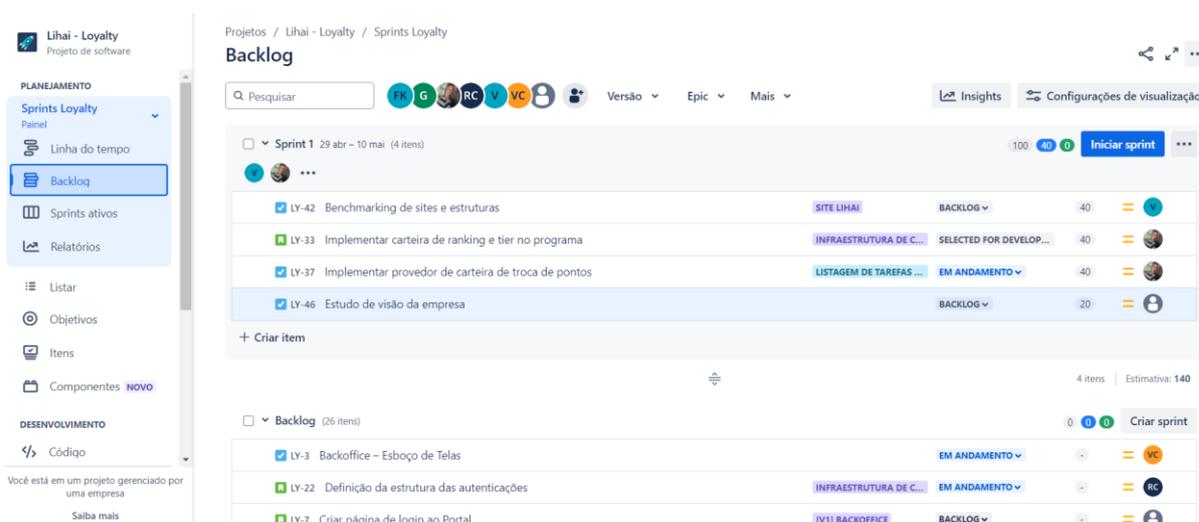
Fonte: elaborado pelo autor

Apesar do time *Scrum* conter 2 desenvolvedores *backend*, um dos desenvolvedores teve que se manter focado em entregas de outro projeto. Portanto, nesse caso, ele não teve entregas atribuídas. Porém, um novo integrante passou a participar da empresa em abril, focado no desenvolvimento de um site para promover o produto

4.4.3 *Software* de acompanhamento das atividades

Com o objetivo de organizar e acompanhar as entregas estipuladas, utilizou-se o *Software Jira*, devido à sua grande utilização no mercado de desenvolvimento de *Softwares*, e o conhecimento prévio de alguns integrantes. O *Jira* permite o acompanhamento das tarefas através de um quadro *Scrum*, mas também possui outros quadros como *Kanban* e até a possibilidade de criação de páginas de documentação vinculadas ao projeto.

A Figura 8 apresenta a aplicação do *Software* voltada ao gerenciamento de projetos utilizando *Scrum*. É possível adicionar itens no *backlog*, como épicos, histórias de usuários, tarefas e sub-tarefas. Em cada item é possível descrever os detalhes da atividade, incluir os *Story points*, datas de início e fim, dependências, responsáveis e outros campos personalizáveis. Na figura, nota-se a divisão entre *backlog* e *sprints*, com a facilidade de transitar um *card* do *backlog* para a *Sprint*, além de verificar os responsáveis, *story points*, nome da atividade e situação.

Figura 8 – Processo de montar o *backlog* e *sprints*

Fonte: elaborado pelo autor

O *backlog* acabou sendo composto pelo desmembramento das atividades a nível macro, pois a maior parte das atividades envolvem sub-tarefas que foram sendo compostas no *backlog* ao tempo que aconteciam refinamentos. Os refinamentos foram feitos em agendas online ou através de conversa no *google workspace*, onde um tópico específico era debatido a fim de definir quais sub-tarefas iriam compor aquela entrega, bem como estimar esforços. Inicialmente, os refinamentos ocorreram apenas para as entregas do segundo trimestre, pois o dinamismo do projeto, envolvendo também o aspecto da metodologia ágil, faz com que tomadas de decisões precipitadas de tarefas de longo prazo possam ser alteradas no futuro, gerando um retrabalho da equipe.

Nesse momento, portanto, foram acrescentadas as atividades envolvendo as lógicas de carteiras de pontos, carteiras de *ranking*, páginas de usuário e administrador, site e *Branding* da empresa.

A definição do tipo de atividade ocorreu de forma a definir se a entrega envolve a interação do usuário final, sendo definida como “história de usuário”, ou se é uma atividade necessária apenas para compor uma estrutura do programa, seja em *backend*, *frontend*, banco de dados e afins. Para essas atividades estruturais, adotou-se o termo “tarefas”, pois não envolvem a participação do usuário final.

4.4.4 Sprints

Conforme mencionado anteriormente, a definição de prazos de entregas e separação de diferentes níveis a serem desenvolvidos ocorreu ao final de março de 2024. Entre março e abril a empresa teve dificuldade em seguir imediatamente com o desenvolvimento do seu programa de fidelização devido à demanda urgente dos dois clientes fixos. Em um dos clientes o projeto estava em fase final de desenvolvimento, necessitando configurações de sistemas de segurança para as transações de pontos. O outro cliente passou por uma mudança em um parceiro de auditorias externas, necessitando esforço dos desenvolvedores para realizar as integrações e alterações nas rotas necessárias para que no final de abril o programa estivesse funcionando normalmente. Dessa maneira, apenas ao final de abril de 2024 que as atividades do programa de fidelidade LiHai conseguiram avançar.

As *sprints* foram definidas com um ciclo de duas semanas. Passando pelos eventos da reunião de planejamento, as agendas diárias, e por final a revisão e retrospectiva. Os *Story points* foram acertados com a equipe com a seguinte lógica:

- Cada funcionário atua aproximadamente 4 horas/dia na empresa.
- Algumas vezes pode atuar menos ou mais, mas em geral cada fluxo de serviço gira em torno de 2 horas.
- Então, cada *Story point* corresponde a duas horas de trabalho
- Atividades que duram uma semana, então são definidas como 10 *Story points*, pois:

$$\circ 4 \frac{\text{horas}}{\text{dia}} * 5 \frac{\text{dias}}{\text{semana}} * \frac{1}{2} \frac{\text{Story points}}{\text{hora}} = 10 \frac{\text{Story Points}}{\text{semana}}$$

4.4.4.1 Sprint 1

A primeira *Sprint* ocorreu entre os dias 29/04 e 10/05/2024. As atividades que compuseram o *backlog* da *Sprint* foram definidas em agenda de planejamento com a equipe no dia 25/04. A Figura 9 apresenta como ficou definido o *backlog* da primeira *Sprint* no momento em que foi iniciada, a Figura 10 apresenta a *Sprint* durante seu andamento, após a conclusão das primeiras atividades. É possível perceber que basicamente todos os desenvolvedores e designers do time *Scrum* foram alocados em tarefas. Com exceção de um desenvolvedor *backend*, que ainda possuía entregas com maior urgência para outro cliente.

Figura 9 – Backlog da primeira *Sprint*

Item	Descrição	Componente	Status	Pontos	Atribuído
LY-42	Benchmarking de sites e estruturas	SITE LIHAI	EM ANDAMENTO	20	VC
LY-33	Implementar carteira de ranking e tier no programa	INFRAESTRUTURA DE C...	EM ANDAMENTO	20	VC
LY-37	Implementar provedor de carteira de troca de pontos	LISTAGEM DE TAREFAS ...	EM ANDAMENTO	20	VC
LY-46	Estudo de visão da empresa	BRANDING LIHAI	EM ANDAMENTO	10	VC
LY-44	Definição de Tipografia	BRANDING LIHAI	SELECTED FOR DEVELOP...	10	VC

Fonte: elaborado pelo autor

Figura 10 – Quadro com atividades da primeira *Sprint*

Coluna	Item	Descrição	Pontos	Atribuído
EM ANDAMENTO 3	LY-33	Implementar carteira de ranking e tier no programa	20	VC
	LY-37	Implementar provedor de carteira de troca de pontos	20	VC
	LY-44	Definição de Tipografia	10	VC
ITENS CONCLUÍDOS 2	LY-42	Benchmarking de sites e estruturas	20	VC
	LY-46	Estudo de visão da empresa	10	VC

Fonte: elaborado pelo autor

No dia 29/04 começaram a ser desenvolvidas as atividades da *Sprint*. O acompanhamento seguiu através das reuniões diárias. As questões voltadas à *Branding* e ao site da empresa seguiram sem maiores problemas, com algumas agendas ao final da *Sprint* entre Gerente de Produto e designers para validação do andamento das entregas desenvolvidas antes de apresentar ao restante da equipe. As questões envolvendo carteira de pontos e de *ranking*, apresentaram algumas dúvidas no decorrer das duas semanas. Para sanar as dúvidas, assincronamente ocorreram conversas entre os responsáveis pelo desenvolvimento, com o time de produto, nos chats do *google workspace*. Também foi realizada uma agenda para mostrar a situação das carteiras e obter maior celeridade nas dúvidas em relação às estruturas desenhadas.

Ao final da *Sprint*, no dia 09/05/2024 foi realizada uma agenda com toda a equipe para seguir a cerimônia da retrospectiva da *Sprint*. E no final do dia, uma agenda com objetivo de fazer um rápido *review* e também planejar as tarefas da próxima *Sprint*.

4.4.4.1.1 *Benchmarking de sites e estruturas*

Como resultados da primeira *Sprint*, tivemos a definição da estrutura do site. Ele segue padrões identificados em *benchmarking* com empresas desenvolvedoras de programas de incentivo de fora do Brasil, além de necessidades e sugestões apontadas pelo designer responsável. Os principais apontamentos da estrutura do site foram apresentar quem a empresa é, quais soluções ela possui, e também destacar a experiência com outros clientes de nível nacional. Além disso, identificar os diferentes pacotes de serviço e suas vantagens, para atender diferentes níveis de empresas. Por fim, uma plataforma estilo *blog* foi sugerida, para contribuir com o cenário de fidelização trazendo conteúdos baseados na experiência da LiHai no mercado.

4.4.4.1.2 *Atividades de Branding*

Basicamente, a estrutura do site pretende seguir no modelo de Scroll, onde em apenas uma página contém todas as informações relevante ao possível cliente. Mostrando principalmente quem somos, nossas soluções, nossas experiências com outros clientes. Mas principalmente, incentivando o possível cliente a buscar contato com a empresa, onde pode ser apresentada de forma mais detalhada as soluções, com contato direto com responsáveis da LiHai, apresentando uma visão de aproximação mais individual e atenciosa ao cliente.

As atividades de *Branding* passaram pelo primeiro ponto de entender a visão de toda a equipe em relação à empresa e ao que desenvolvemos. Portanto o designer responsável por essa tarefa realizou entrevistas com todos os participantes da empresa, e ao final, apresentou a visão da empresa, e também aplicando alguns conceitos de design com relação às cores que poderiam ser trabalhadas no *Branding*. Além disso, também iniciou o processo de definição da tipografia a ser usada pela LiHai em sua identidade visual. O resultado foi uma apresentação e documentação dos processos utilizados pelo designer, onde as cores preto e amarelo foram selecionadas, também a identificação dos participantes com personalidade de “sinceridade e competência”, tudo isso atrelado a conceitos da cultura Chinesa, origem do nome da empresa, que significa “incrível, sério, capaz”.

A Figura 11 apresenta um exemplo do material desenvolvido pelo designer, com a aplicação dos conceitos de design voltados à identificação de marca. A segunda atividade para o designer de *Branding*, de realizar a seleção da tipografia, foi realizada sem nenhum problema reportado, utilizando uma Matriz de seleção tipográfica, onde como resultado foi apresentada a definição da fonte “*Comfortaa*” como a selecionada para utilização na identidade visual da empresa. O trabalho foi realizado definindo pesos a questões como legibilidade, qualidade, investimento e outros aspectos com algumas fontes sugeridas. Os pesos e valores de cada critério, bem como a matriz utilizada não pôde ser apresentada nesse trabalho por pedido dos responsáveis.

Figura 11 – Apresentação dos elementos utilizados no *Branding*



Fonte: elaborado pelo designer de *Branding*

4.4.4.1.3 Carteira de pontos de troca

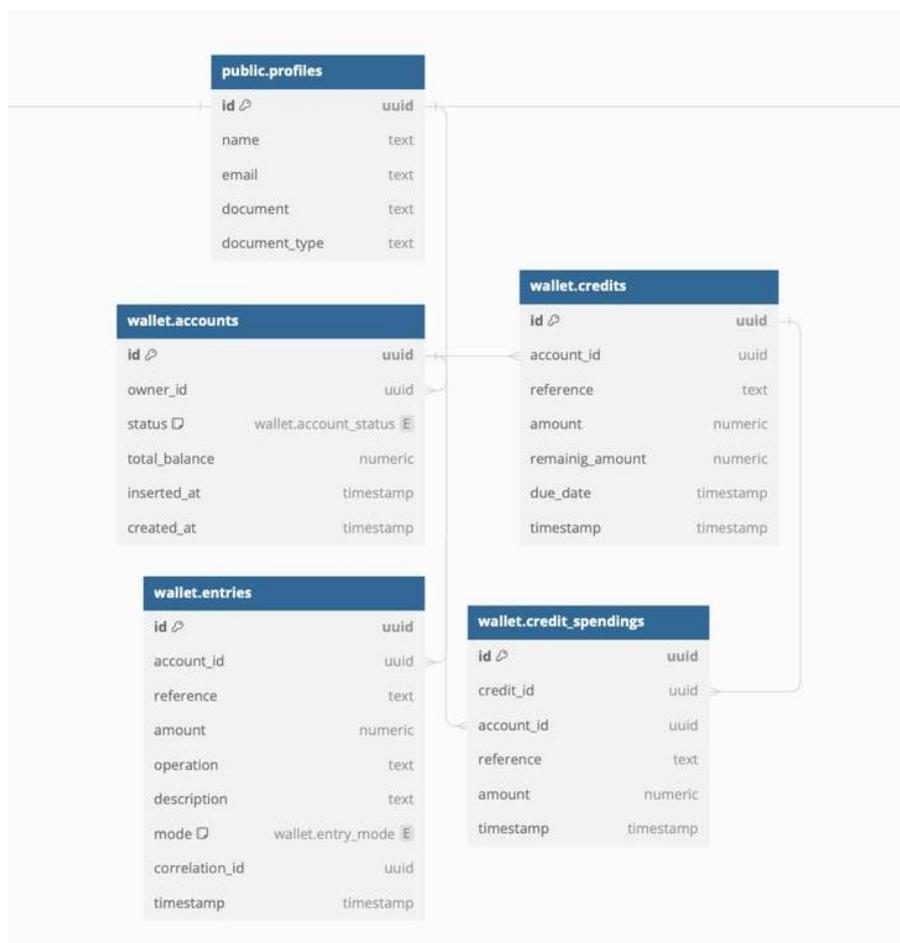
As tarefas que envolvem de fato a estrutura do programa, foram realizadas de maneira a criar as tabelas necessárias para armazenar as informações de carteira de pontos e carteira de *ranking*. Também foi desenvolvido em um ambiente “local” a estrutura do código. O desenvolvimento local consistiu na codificação na própria máquina do desenvolvedor, sem vincular o código desenvolvido em nuvem inicialmente, trazendo possibilidade de alterações e testes individuais sem risco de afetar outros pontos do projeto. Por fim, após o desenvolvimento

e testes unitários, as duas carteiras desenvolvidas foram atribuídas à um ambiente de nuvem, onde poderiam “conversar” entre si.

A estrutura das tabelas de carteira de pontos ficou definida conforme a Figura 12. Cada tabela contém suas informações necessárias o registro da movimentação de pontos, conforme especificações abaixo:

- a) Perfil (`public.profiles`): Tabela a ser utilizada de modo geral no programa, onde armazena informações do usuário. Inicialmente foram definidas apenas informações básicas como nome, documento e e-mail.
- b) Carteira de pontos (`wallet.accounts`): Tabela responsável por registrar o saldo e *status* das carteiras do clientes. Cada carteira é referenciada por um número identificador único, e também ligada à o número identificador do usuário. Sendo assim possível rastrear a qual usuário pertence cada carteira criada no banco. Além disso, o status serve para definir se a carteira está apta ou não a receber transações, visando casos onde usuários podem se desligar do programa, ou ficarem inativos/bloqueados por motivos que sejam de interesse da administradora.
- c) Movimentação de pontos (`wallet.entries`): Tabela responsável por registrar a movimentação de pontos que não possuem validade dentro da carteira de um usuário. Cada transação possui um identificador único, e também o identificador do usuário. A transação também registra as informações de valor, referência (serve como uma breve descrição da ação, como “compra no e-commerce, resgate de brinde”...), uma possível descrição recebida na transação, data, operação (entrada ou saída) e um identificador de correlação, que serve para identificar um evento que gerou aquela transação.
- d) Entrada de créditos (`wallet.credits`): Tabela responsável por registrar entrada de pontos que possuem uma determinada validade definida pelo administrador. Esses pontos são registrados em forma de um crédito vinculado à um identificador de um cliente, apresentando também o valor do crédito, o valor que ainda resta (para identificar se aquele referido crédito já foi consumido total ou parcialmente, ou se ainda não foi consumido), a data de validade, e a referência da sua transação.
- e) Gastos de créditos (`wallet.credits_spending`): Tabela responsável por registrar o consumo de um crédito que o usuário possui em validade, sempre priorizando o formato “primeiro que entra, primeiro que sai”. Como as demais tabelas, ela é vinculada à um número identificador de um usuário, mas também precisa registrar o identificador de qual crédito está consumindo. Também registra valor consumido, data e referência.

Figura 13 – Tabelas de carteira de pontos para troca



Fonte: elaborado pelo autor

Para verificar de forma funcional a estrutura de carteira de pontos, o desenvolvedor responsável aplicou as tabelas e o código de pontos em um ambiente provisório utilizado para testes, onde anteriormente já havíamos utilizados para validar a aplicação de informações de rotas com outros clientes. Então, utilizando uma página que remete à um extrato, é possível ver um exemplo de movimentação de pontos de carteira à um usuário que ganhou créditos. Na Figura 13 é possível verificar que o usuário ganhou “bônus”, referenciando à um tipo de transação que pode ocorrer. Além disso esses pontos ganhos possuem uma data de validade, significando que são créditos com uma data limite para serem resgatados, e também é possível visualizar o saldo final do usuário.

Figura 13 – Aplicação da carteira em um ambiente de testes

Data	Origem	Pontos	Saldo
08/05/2024	Bônus	+100 Válidos até 08/05/2025	7.400 total de pontos
08/05/2024	Bônus	+200 Válidos até 08/05/2025	7.200 total de pontos
08/05/2024	Bônus	+200 Válidos até 08/05/2025	7.000 total de pontos
08/05/2024	Bônus	+2.000 Válidos até 08/05/2025	5.000 total de pontos
08/05/2024	Bônus	+5.000 Válidos até 08/05/2025	0 total de pontos

Fonte: elaborado pelo autor

4.4.4.1.4 Carteira de pontos de ranking

A atividade de criar uma carteira para *ranking* seguiu o mesmo modelo da criação de carteira para troca de pontos, visto que a dinâmica é praticamente a mesma: possuir tabelas que registram a pontuação do usuário em um intervalo de tempo. Porém, no caso de *ranking*, os pontos não são utilizados para resgate de benefícios, mas sim, para diferenciar os usuários participantes do programa de incentivo em categorias e posições em *ranking*, facilitando alguns modelos de *gamificação*.

A Figura 15 apresenta as tabelas criadas e suas ligações. Todas elas, assim como a carteira de pontos, são vinculadas à um perfil de usuário com seu número identificador, a fim de registrar sua pontuação obtida e poder definir sua categoria. As quatro tabelas e suas características são detalhadas abaixo:

- a) Temporada (tier.seasons): A tabela de temporada tem como objetivo registrar a criação de uma campanha para validar as pontuações de *ranking* e categoria. Visto que os programas de incentivo podem ser anuais, mensais ou até ter um intervalo para atender campanhas de datas comemorativas como natal e páscoa, é necessário que o administrador crie uma temporada com data de início e fim. Além disso a temporada deve ter um nome e categoria, valores que serão utilizados na maior parte de forma interna para garantir o controle dos pontos, já que os nomes e descrições dessas temporadas podem ser acrescidos de maiores informações e peças de marketing desenvolvidos pela empresa administradora.

- b) Carteiras de categoria (tier.accounts): Esta tabela serve como uma carteira de pontos para o usuário, vinculado às temporadas que este participa. A tabela registra também a soma de pontos obtidos na temporada, a situação da carteira (em caso de estar aberta, fechada, bloqueada), as datas de criação e última atualização, e se o usuário é elegível. O campo de elegível é importante pois a dinâmica do programa pode permitir que os administradores criem temporadas para um nicho específico de usuários, visando atingir alguma meta específica (por exemplo, criar uma campanha apenas para usuários de São Paulo, a fim de aumentar venda de X produto na região). Portanto, baseando-se nos critérios definidos na criação da temporada, é possível identificar se um usuário é elegível ou não.
- c) Tipos de categoria (tier.labels): A tabela de tipos de categoria é vinculada apenas à uma temporada, pois em cada temporada o administrador pode criar categorias diferentes. Cada categoria possui um nome, uma pontuação mínima e máxima, e suas datas de criação e atualização. A Figura 14 ilustra um exemplo de cadastro de 4 categorias, no qual ao entrar no programa o participante está na categoria azul, e pode evoluir, de acordo com as pontuações máximas e mínimas das demais categorias, atingir o nível ouro.

Figura 14 – Exemplo de cadastro de categorias

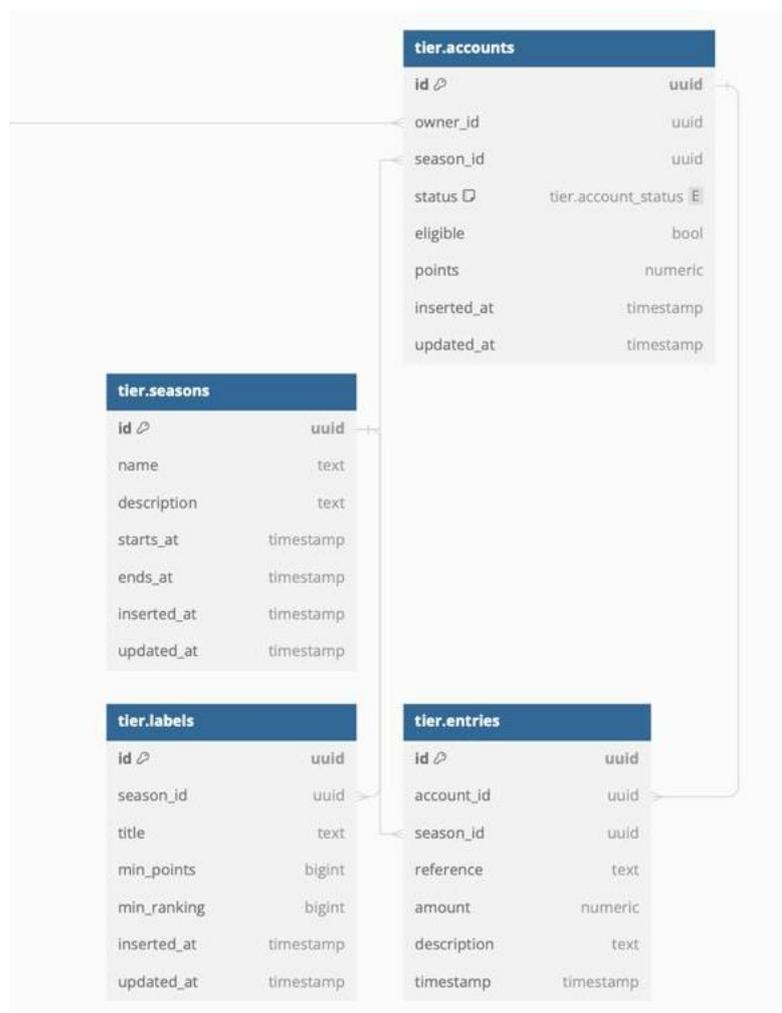
ID	Name	Min	Max
c7adf7ed-884c-49b8-8f9c-5e52eeca2ae0	azul	-9999999	2999
e61a211c-18d1-494c-952e-c1bc0f539531	bronze	3000	5999
c6226e83-66b3-4c7d-ae81-77e7d6b8a2e5	prata	6000	7999
2c2297ba-b9ae-406f-85ba-4e1de78de7af	ouro	8000	9999999

Fonte: Banco de dados de testes da Empresa

- d) Movimentação de pontos de categoria (tier.entries): Por fim, a tabela de movimentação de pontos de categoria é semelhante à tabela de movimentação de pontos de troca. Vinculada à uma conta de usuário e a uma temporada, ela registra todos os pontos atribuídos à usuários após atingir determinados critérios (compras em e-commerce, atingir metas, desafios de auditoria...). Também registra uma referência da

movimentação de pontos, e uma descrição que pode ser atribuída pelo administrador, além da data da transação.

Figura 15 – Tabelas de *ranking* e categorias



Fonte: elaborado pelo autor

As entregas da *Sprint*, apesar de muito técnicas, ocorreram conforme o esperado. Essa base de infraestrutura é necessária para poder seguir com o desenvolvimento das funcionalidades que vão ser renderizadas nas telas para os clientes finais. Também foi possível verificar que a dinâmica de criação de temporadas de pontos para categorias pode ser aplicada em diferentes aspectos de campanhas para os administradores, dando a eles o controle sobre a criação de categorias e campanhas para atingir seu público alvo, que varia de empresa a empresa. Como o programa de fidelidade da LiHai pretende ser escalável, ter esse poder de se adaptar à diferentes cenários é um destaque.

4.4.4.2 Sprint 2

Na segunda *Sprint* foi dado sequência aos três âmbitos que iniciaram na *Sprint* 1: Site, *Branding* e estrutura do programa. O planejamento da *Sprint*, conforme já citado anteriormente, ocorreu na quinta-feira anterior ao início da *Sprint*, que teve como data inicial o dia 13/05/2024. A Figura 16 apresenta as atividades que foram alocadas no *backlog* da *Sprint*, bem como seus responsáveis e os *story points*.

Figura 16 – *Backlog* segunda *sprint*

ID	Descrição	Responsável	Story Points
LY-36	Implementar provedores para bonificação das campanhas	WC	10
LY-35	Ajustar tipagem dos dados nas condições das campanhas	WC	10
LY-45	Logo LiHai	WC	20
LY-50	Estrutura do site	VC	20
LY-51	Wireframe (figma)	VC	10
LY-52	Dados de utilização no site	VC	10

Fonte: elaborado pelo autor

Nessa *Sprint*, porém, só uma parte da equipe foi utilizada. Um desenvolvedor *backend* ficou responsável pela estrutura de bonificação de campanhas, ou seja, o motor de pontos do programa, porém com auxílio do líder técnico para dúvidas. Os designers seguiram atuando nas frentes de site e *Branding*. A redução da equipe ocorreu devido à finalização de uma entrega de outro projeto paralelo, onde requisitava um desenvolvedor *backend*, um desenvolvedor *frontend*, além de maior atenção do líder técnico para realizar a integração de um produto desenvolvido pela LiHai a um terceiro, requisitando testes de segurança e homologação.

O andamento, apesar da redução da equipe, seguiu normalmente. Algumas agendas de dúvidas novamente ocorreram, além de conversas diárias pelo espaço do *google*. Próximo do término da *Sprint*, no dia 23/05/2024, foram realizadas a retrospectiva e a revisão. Ainda nesse meio tempo, no mesmo dia, ocorreu um encontro presencial entre o CEO da empresa, o gerente de produtos e os dois designers. O objetivo do encontro foi validar as evoluções em relação ao site e *Branding*, e pensar os próximos passos e quais os desejos da empresa nesse âmbito.

As entregas da *Sprint* foram positivas pois trouxeram a evolução esperada, conforme pré definição do esforço dos participantes. Principalmente em relação à implantação de bonificação, o motor de pontos do programa, visto que é ponto chave do desenvolvimento.

4.4.4.2.1 Provedor de bonificações

O desenvolvimento do provedor de bonificações, assim como as carteiras de pontos, ocorreu em formato local, visando permitir a facilidade de alterações e testes pelo desenvolvedor. Ao final da *Sprint* o desenvolvedor apresentou à equipe o código desenvolvido simulando uma transação financeira para exemplificar o processo de bonificação. No caso, foram criadas “rotas” que executam funções no programa de acordo com as informações recebidas. É possível verificar algumas ações e configurações do código conforme descrito abaixo:

- a) **Tópico:** inicialmente é criado um tópico, isso é, um formato de evento que o programa vai estar preparado para receber de parceiro externos. Como exemplo, foi criado o tópico “*ecommerce shopping*”, com uma breve descrição e as informações que ele espera receber, como identificador de usuário e um valor de compra. Esse tópico, então, é responsável por receber a informações de possíveis compras em uma loja virtual de uma empresa parceira. A empresa, via integrações, nos envia o identificador do usuário e o valor de sua compra, após isso, são acionadas as regras de bonificação.
- b) **Campanha:** Dentro de um tópico, as campanhas são de fatos as configurações que vão ativar uma bonificação ao usuário. A campanha, portanto, deve ter um identificador de tópico, também é atribuída um nome, descrição e suas estratégias. No nosso caso, a campanha pertence ao tópico de “*ecommerce_shopping*”, utilizando o identificador desse tópico, também apresenta o nome e descrição como um exemplo, e são atribuídas duas estratégias com um nível de prioridade.
- c) **Estratégias:** São chamadas de “condições e efeito”, pois de fato, compara-se condições de um evento, e aplicam-se efeitos. Como um exemplo, podemos supor que a primeira condição é verificar se a compra do usuário é maior que \$400,00, se sim, então aplicam-se como efeito a adição 100 pontos ao usuário, com validade de 30 dias. Caso a compra seja menor que \$400,00, então o efeito é a entrega de 25% do valor da compra em pontuação, conforme o multiplicador “0.25” aplicado no efeito.
- d) **Evento:** é a ação realizada pelo usuário que desencadeia os processos de bonificação. Um evento recebido, busca o tópico em que se encaixa, e então aplica as regras de campanha (estratégias). Podemos ter um evento de compra em uma loja virtual com um valor de \$100,00, conforme já descrito pelo seu tópico. Nesse caso, baseando nas campanhas e estratégias citados anteriormente, esse cliente se encaixa na condição de compra abaixo de \$400,00, tendo como efeito a multiplicação do valor de sua compra

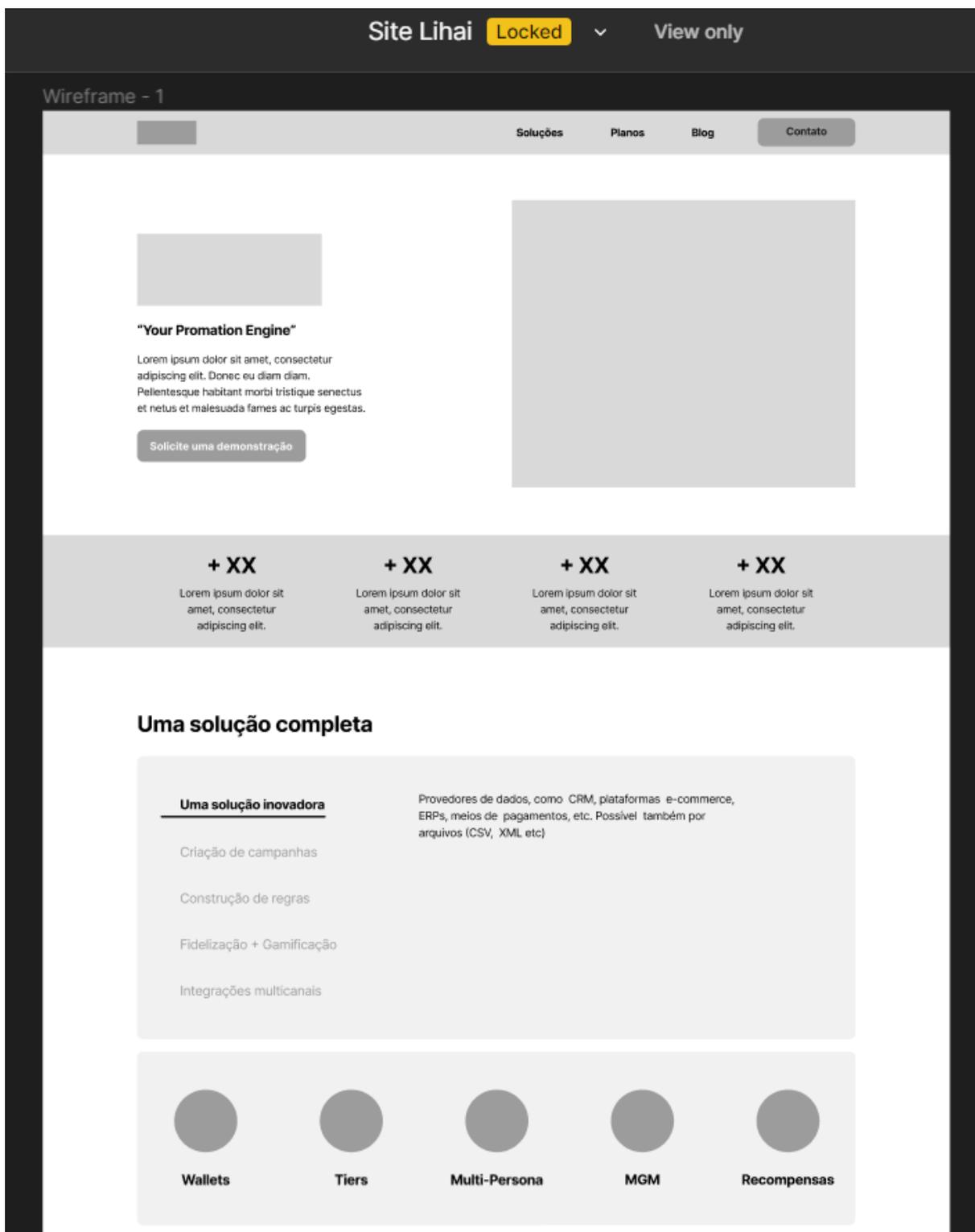
por 25% como atribuição de pontos. Então, o cliente obteve uma bonificação de 25 pontos na sua carteira.

Todas essas informações de tópicos, campanhas e estratégias são salvas em tabelas no banco de dados, pois vão ser utilizadas também para referenciar as carteiras dos usuários. O exemplo acima mostra como é possível ter uma dinâmica na configuração de regras do programa. Além de regras de transações financeiras, também vão ser utilizadas regras com condições nos perfis de usuário, inclusive às suas categorias definidas também pelos administradores.

4.4.4.2.2 *Protótipo do site*

A atividade de prosseguimento para criação do site da empresa como foco em oferecer o produto seguiu bem na primeira sub-tarefa, referente à montar uma estrutura base através da ferramenta *figma*, um *Software* utilizado para ilustrações e design. Ao final da *Sprint*, foi apresentado um protótipo do site, baseando-se nas tarefas realizadas na *Sprint* anterior como o *benchmarking* e a definição da estrutura. O protótipo seguiu um modelo mais a nível macro, valorizando as estruturas geométricas e a disposição das informações. O segundo passo, que é a definição dos dados a serem utilizados no site, não foi concluído na sua totalidade. Na agenda de revisão da *Sprint*, a equipe verificou que pelo fato do programa estar no início, ainda há algumas dúvidas de quais informações disponibilizar no quesito de integrações e precificação. A Figura 16 apresenta uma parte do protótipo do site apresentado pelo designer responsável no *Software figma*, onde constam as principais informações que serão disponibilizadas no primeiro contato do usuário, dando destaque às soluções da empresa. A Figura 17 por outro lado, apresenta o protótipo onde ainda há dúvidas das informações, portanto as estruturas estão sem maiores informações. Essas dúvidas de precificação dos pacotes e parceiros para integração serão definidas na próxima *Sprint*, com participação do CEO da empresa que tem contato direto com a parte comercial.

Figura 17 – Protótipo site LiHai



Fonte: elaborado pelo Designer de site

Figura 17 – Protótipo site LiHai

Site LiHai **Locked** View only

Plataformas integradas



Conheça nossos pacotes

Starter	Premium	Business	Corporate
---------	---------	----------	-----------

Vamos conversar?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eu diam diam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec eu diam diam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Nome completo*

Email*

Telefone*



Empresa*

Todos os campos com (*) são obrigatórios

Fonte: elaborado pelo Designer de site

4.4.4.2.3 Logo LiHai

Ao final da *Sprint*, o designer de *Branding* apresentou o logo desenhado para a empresa, conforme visto na figura 18, utilizando a fonte já selecionada em atividade da *Sprint* anterior, e utilizando ainda elementos da cultura chinesa conforme apresentado nas definições de elementos visuais. O logo desenvolvido, apesar de representar a empresa LiHai, tem o objetivo de passar a ideia do produto principal da empresa aos usuários, ou seja, seu programa de incentivos.

O logo apresentou um formato circular remetendo ao *Wuxing*, uma teoria da cultura chinesa que considera 5 elementos da natureza, e que sua interação faz com que o sistema fique em constante movimento. Além disso, é possível verificar uma noção de ciclo sendo realizado. De fato, o programa pretende trazer essa ideia de ciclo, pois a fidelização e incentivo de clientes visa trazer o retorno do cliente à empresa em que está gerando incentivo. Em um ciclo de benefícios multilaterais: tanto a empresa quanto cliente saem ganhando na fidelização, com a recorrência de compras, disponibilização de benefícios ao cliente, e crescimento da receita da empresa.

Figura 18 – Logo LiHai



Fonte: elaborado pelo Designer de *Branding*

4.4.5 Cerimônias *Scrum*

As cerimônias definidas pelo *Scrum*, conforme definidas na revisão bibliográfica, visam trazer agilidade e interação entre os participantes da equipe *Scrum* e os *stakeholders*. Visando dividir em pequenas entregas de itens onde possa ser verificado se atendem à expectativa do cliente em questão de usabilidade, melhorias e infraestrutura dos *Softwares*. As cerimônias de planejamento, reunião diária, revisão e retrospectiva, cada qual com sua função, foram adaptadas à realidade da LiHai. Principalmente pelo tempo disponível dos participantes do time *Scrum*.

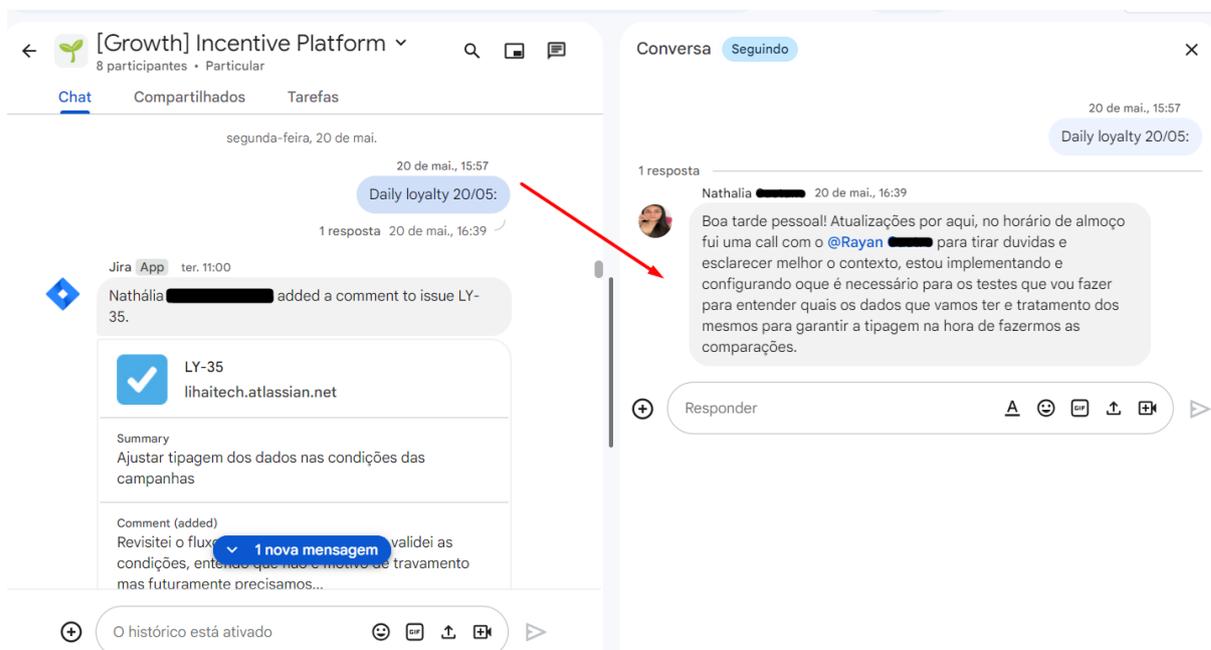
4.4.5.1 Reuniões diárias assíncronas

Historicamente, a empresa já havia tentado trazer o formato de reuniões diárias, porém tinham pouca adesão devido à dificuldade de horários entre os membros da equipe, dessa forma, uma sugestão foi a opção de uma *daily* assíncrona. Ou seja, ao invés de 15 minutos de agenda (geralmente com poucos participantes), optamos por criar uma sequência de conversa nos espaços do *google*, chamadas de “*thread*”, isto é, um fio de mensagens entre os participantes. Nessas conversas são solicitadas as atualizações diárias dos membros da equipe, incentivando que todos participem apontando as evoluções e dúvidas nos horários que tem disponíveis. Os pontos de dificuldades encontrados nas *dailys* direcionam para agendas síncronas, nesse caso, focado em entender e resolver o problema específico, apenas com os membros interessados, liberando o tempo dos demais membros do grupo.

Na Figura 19 é possível visualizar um exemplo da utilização dos espaços do *google workspace*, onde a mensagem inicial do fio é criada, obtendo a atualização dos participantes sobre as atividades que estão exercendo. É possível ver que após o Gerente de produto enviar a mensagem “*Daily loyalty 20/05*”, no espaço reservado para discussão sobre o programa de incentivos, houve uma resposta de um desenvolvedor da equipe sobre o andamento na sua atividade em um espaço secundário. Essa visão de reunião diária assíncrona é encontrada como realidade já em outras empresas, como é o caso definido por Mota (2021), que apresenta as diferenças de horário a equipe e interrupção do foco de atividades realizadas durante o dia como motivadores para implementação desse formato. A autora destaca como benefícios da reunião diária assíncrona a facilidade em participação da equipe, pois não há horário específico, e sim, apenas a necessidade de atualização durante o dia; o registro das atividades através das *threads* e a não interrupção de alguma atividade que um participante está exercendo, podendo este

terminar o que considera necessário sem interrupções, e após isso – ou antes – discorrer as suas atualizações diárias.

Figura 19 – Utilização de *dailys* no formato assíncrono



Fonte: elaborado pelo autor

Além do registro das atualizações pelos próprios membros do time LiHai, também é possível perceber uma notificação do *Jira* no espaço de conversa sobre o Programa de incentivo. Através de uma integração entre o *Jira* e o *google*, foi determinada que cada alteração em atividades no *Jira* seja refletida como uma notificação no espaço. No caso, podemos ver um comentário do desenvolvedor na tarefa em que estava atuando no momento. Então, qualquer participante do projeto consegue facilmente acompanhar o andamento das atividades.

4.4.5.2 Retrospectiva

Em relação às retrospectivas, que são importantes no modelo *Scrum*, foi sugerido por membros da equipe um processo de retrospectiva em 3 passos. Sendo dois passos assíncronos e 1 passo síncrono, novamente, devido à falta de sincronicidade nos horários disponíveis pela equipe. A ideia da retrospectiva é trazer melhoria contínua no processo, aplicando os planos de ação assim que eles sejam definidos, e visualizando novamente em retrospectivas ou com métricas, se aquela ação foi efetiva ao problema em questão. A utilização de retrospectiva assíncrona é um assunto já discutido no ambiente do desenvolvimento ágil. De acordo com

Khlon (2023) a retrospectiva assíncrona pode reduzir as interrupções diárias no trabalho dos desenvolvedores, além de poder incluir aqueles que têm dificuldades de participar das agendas síncronas, como participantes em diferentes fusos. Cotrel (2022) também destaca a questão do fuso horário e da dificuldade de agendar reuniões síncronas com toda a equipe. O autor ainda declara que é possível dividir as etapas da retrospectiva entre assíncronas ou síncronas, de acordo com a disponibilidade da equipe.

Adaptando esse formato conforme descrito anteriormente, a retrospectiva seguiu 3 etapas ao final de cada *Sprint*, visando manter essa rotina de melhorias. Um sistema online de retrospectiva é utilizado como base para puxar essas informações com a equipe.

- a) No primeiro dia, os membros acessam o sistema e criam cartões com pontos positivos e negativos que estão encontrando nos processos, esses cartões são anônimos, e o usuário não pode ver os demais cartões criados, sendo assim algo que garante o sigilo de quem está solicitando melhorias, processo que pode ser visualizado na Figura 20.
- b) No segundo dia, os membros acessam o sistema, visualizam todos os cartões criados e podem votar nos que consideram mais importantes, um número limitado de votos é definido, para incentivar aos membros uma priorização dos assuntos tratados. Na Figura 21 é apresentada o quadro do segundo dia, com a visualização dos cartões e votações.
- c) No terceiro dia, uma agenda síncrona é realizada com objetivo de traçar planos de ação para os cartões mais votados. Dessa forma, são ordenados os cartões pela quantidade de votos, discutido com a equipe quais os problemas encontrados em relação ao assunto encontrado, e definidas ações para mitigar esses problemas. Os planos de ação podem ser revisados no decorrer da execução, a fim de trazer melhoria contínua ao processo, e também podem ser discutidos nas próximas retrospectivas caso os cartões ainda estejam no quadro. A Figura 22 apresentar o resultado final da retrospectiva, com os cartões mais votados entrando na coluna do plano de ação.

Quadro 8 – Planejamento da Restrospectiva

Dia	Terça	Quarta	Quinta
Ação	Criar cartões	Votar nos cartões	Definir planos de ação
Formato	Assíncrono	Assíncrono	Síncrono
Onde	Plataforma online	Plataforma online	Reunião

Fonte: elaborado pelo autor

Um resumo dos passos utilizados da retrospectiva está listado no Quadro 8. Ao final do processo de retrospectiva, espera-se obter resolução nos problemas encontrados pela equipe, revisão dos planos de ação, e ao longo do tempo, ir “limpando” o quadro, obtendo menores problemas encontrados, e preenchendo com os processos que a equipe considera importante na coluna “O que foi bom”.

4.5 QUESTIONÁRIO SOBRE APLICAÇÃO DO SCRUM

Após duas *sprints* um questionário sobre a implantação do *Scrum* no desenvolvimento do programa de incentivos da empresa foi aplicado para todo o time *Scrum*. Apesar de possuir atividades no *backlog* e uma expectativa de mais entregas durante o ano, o questionário foi aplicado a fim de obter os resultados quantitativos da percepção da equipe com relação às cerimônias aplicadas. Mesmo após pouco tempo de implantação da metodologia ágil na empresa, as perguntas visam comparar os valores encontrados com o resultado do questionário e entrevista aplicados no início deste trabalho, antes da aplicação do *Scrum*.

As perguntas foram feitas através de um formulário no *google forms*, e enviada aos participantes do time *scrum* para responderem de forma anônima, para dar maior liberdade do participante responder sem receios. Abaixo é possível verificar as perguntas aplicadas, sendo que todas as perguntas respeitam o modelo de escala *Likert*, variando de 1 a 5, onde 1 é considerado como “Ruim ou Pouco Importante”, e 5 é considerado “Bom ou Muito importante”, a depender do contexto. A única exceção da escala *Likert* é a última pergunta, que foi deixada como um campo aberto para os participantes escreverem sua opinião sobre a implantação do *Scrum* e poder pegar pontos de vista a fim de obter também respostas qualitativas.

Perguntas realizadas no questionário:

1. **Como considerou o tempo para a entrega das atividades dentro de cada trimestre, definidos no encontro da equipe em Março?**
2. **Como você considera a clareza do fluxo de atividades no *Jira*?**

3. **Qual o grau de importância da utilização do *Jira* na sua visão?**
4. **Como você considera o registro das informações de atividades futuras no *Jira*?**
5. **Como você considera a evolução do registro de processos e informações da empresa? (*Confluence, threads, drive* padronizado)**
6. **Como considera os processos de comunicação assíncrona da equipe?**
7. **Como considera a implantação da cerimônia de retrospectiva?**
8. **Qual o grau de importância da retrospectiva na sua visão?**
9. **Como considera a implantação da cerimônia de *daily* assíncrona?**
10. **Qual o grau de importância da *daily* na sua visão?**
11. **Como você considera a utilização de um checkpoint para revisar as entregas?**
12. **Qual a importância você dá para a utilização dos checkpoints?**
13. **Qual o grau de importância para a utilização do *Scrum* em geral?**
14. **Campo aberto para considerações finais.**

Participaram do questionário 5 integrantes, e os resultados são visualizados na Tabela 4. É possível verificar que os valores médios encontrados em relação às cerimônias do *scrum* obtiveram uma média alta de importância na visão da equipe, sendo a reunião diária e o checkpoint (onde eram realizadas revisão da *Sprint* atual e planejamento da próxima) os mais destacados. Em geral, a equipe teve uma boa aceitação dos procedimentos de reuniões assíncronas, planejamento, revisão e retrospectivas, e consideraram processos importantes na execução de suas atividades.

Por outro lado, dois pontos que foram apontados no início deste trabalho como problemáticos, ainda apresentam um grau de insatisfação pela equipe: O registro das atividades e a comunicação da equipe. Apesar de uma pequena evolução nos números da média destes itens, em comparação com a pesquisa realizada no início do trabalho, eles permanecem como pontos de atenção e melhoria. Apesar da implantação da ferramenta *Jira* para uso diário, das notificações de cada ação realizada no *Jira* refletida no *google Workspace*, ainda existem dúvidas na execução de atividades, que passam principalmente pelo registro das mesmas no *Jira*. Muitas vezes, as atividades de nível técnico, se redigidas pelo *Product Owner*, podem não conter todas as especificações necessárias para o bom cumprimento da funcionalidade desejada. Isto pode ser mitigado através das agendas de refinamento entre *Product Owner* e desenvolvedores, porém, nesse caso, mais uma vez caímos na dificuldade de horários entre as equipes. Dessa maneira, algumas vezes as tarefas foram criadas na ferramenta *Jira*, sem haver refinamento, apenas descrevendo de forma superficial a atividade, e isso trazia dúvidas aos desenvolvedores.

O registro de informações paralelas, como o Termo de Abertura do Projeto, ambientes utilizados pelos desenvolvedores, referências de *benchmarking*, além de documentação técnica,

foram agrupados em uma página do *Confluence*, uma ferramenta de edição de páginas e documentos vinculada ao *Jira*. Também foi utilizado um drive compartilhado com toda a equipe para agrupar documentos importantes. Mas ainda assim a equipe não sentiu uma evolução nesse âmbito, mostrando que ainda há possibilidades de evolução na documentação do projeto. De fato, por estar no início e haver muitos pontos de incerteza, a documentação é mais restringida aos requisitos do projeto em nível macro, participantes e tempos estipulados, mas os detalhes de cada parte do projeto estão sendo definidos com a equipe *Scrum* ao decorrer que as atividades avançam. Espera-se, então, que com maiores decisões sendo tomadas em relação à detalhes do programa, infraestrutura, peças de marketing e questões de segurança, possa haver uma documentação mais detalhada do projeto disponibilizada à toda a equipe, e posteriormente aos futuros cliente.

Mais informações, como a discussão de respostas qualitativas da pergunta 14 são discutidas no item 4.7 do trabalho.

Tabela 4 – Resultado do questionário de aplicação do *Scrum*

	Resp 1	Resp 2	Resp 3	Resp 4	Resp 5	Média
Como considerou o tempo para a entrega das atividades dentro de cada trimestre, definidos no encontro da equipe em Março?	4	5	4	3	4	4
Como você considera a clareza do fluxo de atividades no <i>Jira</i>?	4	5	3	4	4	4
Como você considera o registro das informações de atividades futuras no <i>Jira</i>?	4	4	4	4	4	4
Como você considera a evolução do registro de processos e informações da empresa? (<i>Confluence, threads, drive padronizado</i>)	4	4	4	2	4	3,6
Como considera os processos de comunicação assíncrona da equipe?	3	4	5	4	4	4
Como considera a implantação da cerimônia de retrospectiva?	4	4	3	4	5	4
Qual o grau de importância da retrospectiva na sua visão?	5	5	2	3	5	4
Como considera a implantação da cerimônia de <i>daily</i> assíncrona?	4	5	3	5	5	4,4
Qual o grau de importância da <i>daily</i> na sua visão?	5	5	4	4	5	4,6
Como você considera a utilização de um checkpoint para revisar as entregas?	5	5	3	5	5	4,6

Qual a importância você dá para a utilização dos checkpoints?	5	5	3	5	5	4,6
Qual o grau de importância teve a utilização do <i>Scrum</i> em geral?	4	5	4	3	4	4

Fonte: elaborado pelo autor

4.6 DIFICULDADES ENCONTRADAS PARA MANTER OS PROCESSOS

Conforme mencionado em alguns momentos, o processo de aplicação do *Scrum* passou por algumas dificuldades devido à fatores como tempo dos participantes, situação do projeto, divisão da atenção do desenvolvimento do programa de incentivos com projetos de outros clientes. Abaixo estão listadas as principais dificuldades encontradas:

1) Falta de disponibilidade de horários dos membros da equipe: este item foi um dos principais fatores que influenciam nas dificuldades de seguir cerimônias de forma síncrona. A equipe é composta por estudantes, *freelancers* e membros fixos. Cada membro da equipe tem uma grade de horários muito variada durante o dia, de acordo com seus compromissos pessoais ou definição de horários de trabalho (a grande maioria trabalha 4 horas por dia). Marcar agendas com todos participantes é muito difícil, tendo que na maioria das vezes utilizar horários fora do comum no ambiente comercial. As agendas de revisão e planejamento, por exemplo, ocorriam entre as 18h e 19h. Esse fator foi o principal motivador a adotar processos assíncronos dentro das cerimônias do *Scrum*.

2) Comunicação: O ambiente remoto faz com que muitas vezes haja uma defasagem de comunicação, principalmente em relação a dúvidas que possam existir no decorrer de alguma atividade, fazendo com que ela fique parada até que seja esclarecida. Em alguns momentos, houveram travas em atividades justamente pelas dúvidas que surgiram demorarem a ser sanadas. As dúvidas eram compartilhadas no espaço de conversa sobre o produto no *google workspace*, porém a resposta ocorre apenas quando a pessoa questionada estiver disponível, e esse processo fez com que alguns desenvolvedores ficassem um tempo “parado” até sanar sua dúvida.

3) Priorização de atividades: Outro ponto importante é a priorização de atividades dentro de uma entrega, seja considerando utilizar um *Sprint*, ou os modelos de entregas incrementais. Pois devido a necessidade de atender clientes externos, em alguns momentos os suportes ou melhorias solicitadas pelos clientes se tornaram prioridade em relação às atividades que estão sendo executadas dentro da sequência definida no primeiro momento. Isso aconteceu diversas

vezes durante o andamento do projeto. Como exemplo, é possível citar que o início das atividades se deu apenas ao final de Abril, quando o planejado era iniciar as *sprints* no início de Abril. Esse atraso ocorreu devido à grandes entregas de outros projetos da empresa à clientes externos que demandaram maior atenção da equipe. Com isso, o projeto do programa de incentivos da LiHai se tornava o menos prioritário nesse momento.

Além disso, a equipe *Scrum* definida não utilizada em sua totalidade nas primeiras *Sprints*, isso porque um desenvolvedor *backend* e um desenvolvedor *frontend* ainda tinham como prioridade a conclusão de entregas aos demais clientes.

A alteração da prioridade de tarefas é um problema encontrado em outras empresas, conforme Moreira (2018), que destaca em uma pesquisa com participantes de diferentes áreas no desenvolvimento de *Software*, as dificuldades encontradas para se adaptar às mudanças no decorrer de uma *Sprint*, abandonando tarefas em andamento para resolver urgências trazidas pelo *Product Owner*. O autor também apresenta os resultados na visão do PO, que destaca a dificuldade de repriorização de tarefas após a mudança de escopo trazida por *stakeholders*. Em relação ao *Scrum master*, Moreira (2018) descreve que “a maior dificuldade é de manter o processo funcionando com o time quando as demandas em caráter de urgência surgem, como SLA's, por exemplo”.

4) Documentação: Este foi um ponto destacado pela equipe em retrospectiva e também nas entrevistas iniciais. De fato, os primeiros passos do projeto estavam com uma documentação simples, sem descrever maiores processos do projeto, as questões técnicas e requisitos de produto. Em alguns momentos a equipe sentiu falta de um ambiente concentrando informações do projeto do programa de incentivos, pois cada desenvolvedor estava desenvolvendo de forma local, além dos designers estarem atuando no desenvolvimento do *Branding* e site também de maneira individual. Faltava, portanto, um local para reunir todas essas informações.

5) Estruturas de início do projeto: No início, o projeto de programa de incentivos da empresa teve foco em montar as estruturas base para funcionamento do programa futuramente, como as carteiras e motor de pontos. Além disso, também estava trabalhando paralelamente com o desenvolvimento da imagem da empresa, através de definições de logo, tipografia, identidade visual; e também a área comercial com desenvolvimento do site da empresa com foco no programa de incentivos. Nesse momento, portanto, haviam poucas histórias de usuário de fato. As atividades eram muito mais voltadas pra questões internas do programa, do que para funcionalidades que podem ser testadas/usufruídas pelos usuários finais.

O *Backlog* do produto ganhou desdobramento de atividades que compõem histórias de usuários e que serão desenvolvidas ao longo das próximas *sprints* do programa, porém nesse trabalho, o contexto foi composto em sua maior parte de atividades a nível de entregas internas. Mas, apesar disso, o *scrum* pode ser utilizado e serve como um amadurecimento da equipe para as próximas etapas do projeto com foco em entregas aos usuários finais.

O que se pode observar, de maneira geral, é que essas dificuldades encontradas se devem muito ao ambiente de trabalho remoto em que a empresa se encontra. Portanto, entende-se que a aplicação das metodologias ágeis no ambiente remoto, devido aos problemas encontrados, pode ser passiva de adaptações, a fim de obter melhores resultados dentro do cenário encontrado. E, de fato, adaptações foram realizadas em relação ao *Scrum* original, mas com base em experiências de membros da equipe e também de referências encontradas na literatura que destacavam essas adaptações devido às diferenças do trabalho presencial e remoto.

4.7 DISCUSSÃO SOBRE RESULTADOS

Além do questionário aplicado ao final das duas primeiras *Sprints*, o *Product Owner* procurou os participantes da equipe *Scrum* e também o *Scrum master* para discutir suas visões através de agendas particulares ou conversas pelo chat.

De maneira geral, a perspectiva dos participantes é que a aplicação do *scrum* trouxe benefícios para o andamento do projeto. Este é um projeto que desde 2023 vinha sendo discutido na empresa, porém muitas vezes despriorizado em detrimento das entregas que a equipe tinha que dividir com os outros clientes. A adoção do *Scrum* passou por dois grandes pontos que foram considerados cruciais: montar uma sequência de entregas a nível macro, que serviram como base para criar um *backlog* do produto; e organizar as atividades no *Jira* para depois criar e iniciar as *Sprints*.

A organização e sequenciamento das entregas trouxe maior dinâmica na equipe para evoluir com suas atividades e entregas nos tempos pré-estabelecidos. As cerimônias do *Scrum* também foram cruciais para o andamento do projeto, pois as reuniões diárias, planejamento e retrospectiva discutiam de forma recorrente o produto, fazendo com que as dúvidas que foram surgindo no caminho, tivessem respostas mais rápidas e definidas em equipe. O ágil é sinônimo de participação e *feedback* constante, o que foi um fator positivo para amadurecer o produto nesse início. É um momento de grandes incertezas, onde é preciso validar hipóteses de forma

rápida e seguir com o desenvolvimento. Esses pontos foram verificados tanto no desenvolvimento das carteiras de ponto e de *ranking*, como nos provedores de benefícios do programa. Existem inúmeras possibilidades de informações, detalhes e lógicas a se aplicar nesses contextos, mas o objetivo do *Product Owner* e da equipe de desenvolvimento foi de encontrar uma estrutura que seja ao mesmo tempo simples e funcional, onde valide as hipóteses de atribuição de pontos ao usuário após cumprir determinadas regras, e essas hipóteses foram validadas nas entregas realizadas. Foi possível verificar o controle de administradores para criar diferentes categorias de usuários, criar tópicos para bonificação, criar campanhas e estratégias de bonificação.

Além disso, algo menos comum do *Scrum*, mas que foi trazido para esse contexto, foi a utilização dos designers para desenvolvimento do site e identidade visual da empresa. Os designers puderam criar um *backlog* próprio, e definidos os seus esforços, as entregas de logo, tipografia, estruturas e informações de site foram colocadas da mesma maneira dentro de uma *Sprint*. E ao final das *sprints* pode-se conferir as entregas dos designers, não como uma funcionalidade do programa que vai impactar o usuário final, mas como uma entrega de demanda interna que servirá como agente comercial do programa de bonificação da LiHai. A ideia de escalonamento do programa passa pelo processo de ter uma identidade visual definida, e uma plataforma onde os possíveis interessados possam encontrar informações da solução apresentada pela LiHai. Esses fatores, além da qualidade do *Software* desenvolvido pela empresa, farão diferença futuramente para a escalabilidade do produto.

O resultado, apesar de positivo, não foi totalmente satisfatório devido aos cenários encontrados que geraram um atraso para o início das *sprints*, uma incerteza na definição do *backlog* inicial, e uma necessidade de adaptação de processos como as reuniões diárias e a retrospectiva. O *Product Owner* percebeu grande dificuldade de seguir com as cerimônias à risca devido ao trabalho remoto. O próprio Manifesto Ágil cita as reuniões e contatos face a face, obtendo maior agilidade nas respostas, o que não ocorreu nesse caso, pois os processos assíncronos algumas vezes apresentaram demora na obtenção de respostas.

Mas de modo geral, a comunicação da equipe e o andamento das atividades em um horizonte de entregas teve grande evolução. Houve um início e andamento do projeto depois de meses em que estava apenas “no papel”, cresceram as discussões da equipe sobre a visão do projeto de um programa de incentivos, quais pontos devem ser destacados como um agregador de valor ao cliente, e como vamos trabalhar as próximas entregas. Outro ponto a se observar é que mesmo considerando 4 desenvolvedores na equipe *Scrum*, apenas dois trabalharam no primeiro momento. Esse é mais um fator que pode ser considerado “positivo”, pois a equipe

ganha capacidade nas próximas *sprints*, contando com a participação de mais um desenvolvedor *backend* e um *frontend*, podendo evoluir em outras frentes do programa, principalmente na criação dos ambientes de utilização dos administradores (Portal administrativos) e dos usuários (Portal de usuários). É nesse momento em que de fato, as histórias de usuário serão incorporadas na sua forma pura, trazendo as funcionalidades que o programa espera ter na ponta, em detalhe, de maneira que a experiência do usuário seja a melhor possível. E também, reproduzindo em um ambiente navegável, as lógicas desenvolvidas nessas primeiras entregas, que ainda estão em um formato muito técnico, utilizando testes em ambientes locais.

O Quadro 9 apresenta um resumo dos pontos positivos e negativos discutidos nesse tópico e também se baseando nos *feedbacks* dos participantes do time *Scrum* sobre uma visão geral da ferramenta.

Quadro 9 – Pontos positivos e negativos da aplicação do *Scrum*

Contexto	Positivos	Negativos
Sequência de atividades	Organização das atividades através de um <i>backlog</i> trouxe uma melhora na visão do andamento do projeto à equipe, além de auxiliar na definição de quais são prioritárias para desenvolvimento.	Necessidade de detalhamento das atividades pode tomar um tempo da equipe. Perceptível muitas dúvidas em relação ao início do projeto, em alguns momentos com dificuldade para definir por onde começar.
Comunicação	Trouxe maior proatividade e envolvimento da equipe no foco de um novo projeto, principalmente através das reuniões diárias.	Defasagem na comunicação em alguns momentos, devido à necessidade de utilização de meios assíncronos como o chat do <i>google workspace</i> .
Entregas	Melhora na dinâmica de entregas incrementais, com auxílio das agendas de revisão e da divisão de entregas a cada final de Sprint.	Algumas entregas não passaram por processo de testes, apenas pela aprovação da equipe, pois não se encaixavam como história de usuário. Nesses casos, apenas havia o marco de obter uma entrega ao final de uma Sprint (processo de criação da logo, por exemplo), com a aprovação do CEO, PO e designers.
Detalhamento de atividades	A necessidade de definição de um <i>backlog</i> fez com que houvesse refinamento das atividades, podendo trazer mais detalhes de funcionalidades/requisitos, que impactam a utilização pelo cliente final. Também houve interação	Pouco detalhamento de atividades nos momentos iniciais, pois envolviam mais questões de infraestrutura do que histórias de usuário.

	para resolver dúvidas e tomadas de decisão de atividades através do <i>feedback</i> entre <i>Product Owner</i> e desenvolvedores, tanto assíncrono quanto em agendas de refinamento.	
Cerimônias	A utilização das cerimônias propôs um ambiente mais colaborativo e participativo, além de propor uma rotina de entregas à equipe. Apresentou uma evolução em relação ao andamento do projeto em geral.	Dificuldade de conciliar horários com todos os membros da equipe, trazendo a necessidade de adaptações em algumas cerimônias. Apesar de funcionais, agendas assíncronas perdem a essência do ágil.
Jira	Conduziu toda a equipe como referência para documentação das atividades. A ferramenta de divisão em <i>sprints</i> proporcionou destacar um <i>backlog</i> e “mover” as atividades para a Sprint. Trouxe a visão do andamento em quadro <i>Kanban</i> , que também permite de maneira rápida e visual entender a situação da Sprint. Os cartões de atividades possibilitaram incluir responsável, valor de <i>story points</i> , anexos, texto para detalhamento da atividade, divisão em tipos de entrega e datas. Teve boa aceitação da equipe, que se adaptou facilmente ao uso da ferramenta.	Nem toda a equipe utilizou ativamente a ferramenta. Então, apesar de ter a definição da entrega, responsável e esforço, as atualizações ocorriam em grande parte ao ser questionado pelo <i>Product Owner</i> ou <i>Scrum Master</i> , e também no chat de acompanhamento diário, mas essas atualizações não refletiam no <i>Jira</i> . É necessário um trabalho de incentivo para introduzir a ferramenta e torna-la participativa dentro da equipe.
Scrum	Trouxe, como um todo, maior agilidade e interação entre a equipe. A divisão de atividades em entregas a cada duas semanas apresentou proporcionou acompanhar de forma ativa o andamento do projeto, mesmo que algumas entregas sejam apenas de itens estruturais, mas que são cruciais para o bom funcionamento do software no futuro. Além disso, foi possível verificar a flexibilidade das metodologias ágeis, podendo ser adaptadas à diferentes contextos de desenvolvimento.	Dificuldade maior em seguir todas as cerimônias, e de trazer a cultura <i>Scrum</i> para a empresa. Apesar de bem recebida, houve ainda resistência de participação de alguns membros, onde era preciso fazer o trabalho de incentivo à participação nas cerimônias e de utilização da ferramenta <i>Jira</i> .

5. CONCLUSÃO

A evolução no gerenciamento de projetos, desde os seus primórdios com a divisão de trabalho, o sequenciamento *Gantt* e a ampla utilização dos modelos cascata, mostraram a necessidade de ter ferramentas para planejar e acompanhar projetos de qualquer nível. O desenvolvimento tecnológico trouxe a necessária evolução para modelos dinâmicos, e fez com que a divulgação dos princípios das metodologias ágeis se tornasse evidente devido aos seus resultados positivos. Nesse contexto, o desenvolvimento de *softwares* se beneficiou dos conceitos do *Agile*, devido à sua necessidade de constantes alterações de acordo com as necessidades dos clientes, atendendo a novos requisitos, permitindo testar situações e subir novas funcionalidades de maneira rápida. Através disso, o *Scrum* se tornou amplamente difundido no ambiente de desenvolvimento de softwares no mundo todo, tendo notáveis resultados positivos.

O presente trabalho objetivou a aplicação da ferramenta *Scrum* dentro de uma *startup* de desenvolvimento de aplicativos Web, com soluções voltadas à fidelização e incentivo de clientes. O escopo do *Scrum* ocorreu na intenção de criar uma solução própria para abranger diferentes cenários no âmbito da fidelização de clientes, possuindo apenas uma ferramenta adaptável e escalável, sem a necessidade de desenvolver softwares voltados à especificação de cada cliente. Com isso, foram definidos os participantes do *Scrum*, uma definição de entregas esperadas para o projeto a nível macro, e posteriormente foram definidos o *backlog* e as entregas dentro de duas *Sprints*.

Com o auxílio do *Scrum* o projeto saiu do papel e tomou forma, evoluindo não apenas em funcionalidades da ferramenta, mas também em questões comerciais e de marketing da empresa, como definição de identidade visual e criação de site, além de definir alguns pontos cruciais que serão focados como a estruturas do programa. Esses são fatores que auxiliarão a escalar o produto, assim que possua uma versão funcional que atenda às necessidades de controle e dinâmica na criação de ferramentas de campanhas de fidelização. Essas necessidades dos clientes de possuir uma variedade de opções na idealização de ferramentas de fidelização, passa também pelas experiências prévias da empresa através do contato com diversos parceiros, que colocam isso como um dos pontos chave no software. O desenvolvimento do software, então, passou pelas principais questões estruturais dos programas de fidelização que a empresa atua e pretende disponibilizar também no seu novo produto que é a de possuir uma carteira de pontos, onde são registradas as “transações” de pontos dos usuários, assim com um extrato

bancário. Além disso, também passou pelo desenvolvimento de uma carteira para utilizar pontuações em questões relacionadas à gamificação, podendo dividir os usuários em categorias ou até mesmo utilizar um *ranking* de pontos. Mas o ponto chave de desenvolvimento é a dinâmica de atribuição de pontos, que será a lógica utilizada para atribuir pontos à usuários de acordo com os eventos realizados pelos mesmos.

Os resultados foram em grande parte positivos. A equipe passou a ter maior contato e rotina através das cerimônias do *Scrum*. O planejamento de atividades permitiu ver um horizonte de entregas e trazer estas entregas para dentro de uma *Sprint*. Com a utilização da ferramenta *Jira*, as tarefas listadas eram compostas por maiores informações como responsáveis e esforço necessário para conclusão, servindo tanto como uma referência ao desenvolvedor das suas responsabilidades e entregas, quanto ao Gerente de Produtos para acompanhamento do projeto como um todo, e também das *sprints*. Ao final de duas *sprints*, a equipe avaliou a evolução da ferramenta e os resultados, mesmo que muito iniciais, da sua aplicação. A utilização das agendas de retrospectiva e de revisão, chamada de checkpoints, e também o acompanhamento diário, foram consideradas de grande importância dentro do cenário de desenvolvimento. A ferramenta *Jira* também foi aprovada pela equipe e comprovadamente acrescentou resultados positivos no detalhamento e acompanhamento das atividades. Por fim, as entregas direcionadas nas *sprints* 1 e 2 ocorreram dentro do prazo, seguindo o cerimonial definidos pelo *Scrum*, e a troca de feedback entre os participantes proporcionou a equipe como um todo uma visão geral do produto, aprofundando, quando necessário, em maiores detalhes técnicos.

Por outro lado, o ambiente remoto apresentou um cenário em que as atividades “face a face” do ágil não tinham como ser realizadas. Apesar da evolução da utilização das reuniões online no ambiente profissional, o *Scrum* utiliza ferramentas que focam no contato pessoal, fazendo-se cenário uma adaptação para o ambiente de trabalho remoto encontrado na empresa. Outro ponto que trouxe grande dificuldade foi a disponibilidade de horário dos membros da equipe que raramente encontravam um horário em comum para realizar as reuniões, o que vai de encontro com Maffei (2021), que relatou essa dificuldade como uma das principais no ambiente remoto. Foi definido como muito importante a participação de todos nas agendas de revisão e na retrospectiva, procurando horários alternativos para a sua realização, porém em algumas agendas ainda foi possível perceber que nem toda a equipe estava presente. Também a utilização de conversa assíncrona teve que ser implementada, ainda pela dificuldade na disponibilidade de horários. As reuniões diárias tiveram um formato assíncrono através de um chat na plataforma *google workspace*, o espaço serviu como atualização diária das atividades

pelos usuários, e também como registro de dúvidas e informações. Conforme Bataglia (2020), a falta de digitalização do projeto pode representar um problema, então obter esses registros foi um ponto positivo pois houve cenários em que o desenvolvedor conseguiu procurar na plataforma alguma dúvida antiga que possuía e a sua resposta, além de trazer maior participação da equipe do que quando houve a tentativa de aplicar a reunião diária em reuniões online.

De forma geral, apesar das necessárias adaptações, dos cenários encontrados de divisão de tarefas com outros projetos, e da implementação em um processo inicial, a aplicação do *Scrum* se mostrou uma ferramenta útil e que auxiliou a equipe no andamento do projeto do programa de incentivos e fidelização. O uso contínuo da ferramenta, com a adaptação das cerimônias por parte dos usuários e a implantação da cultura *Scrum* dentro da empresa pode trazer resultados positivos e a entrega do produto com todos os requisitos exigidos ao fim do projeto, tornando uma ferramenta escalável e que agregue valor aos usuários.

Uma sugestão de trabalhos futuros, seria realizar a comparação entre dois times *Scrum*, um deles utilizando o *Scrum* tradicional, e outro utilizando as agendas assíncronas, retrospectiva em duas ou 3 partes. Isso pode trazer uma visão dos efeitos dessas adaptações de forma mais concisa. Outra sugestão seria a de utilizar outra ferramenta de gerenciamento de projetos no desenvolvimento de software, em comparação com o *Scrum*. Essas sugestões seriam melhor aplicadas em empresas que possuem mais que um time *Scrum*, pois os cenários serão muito parecidos, apenas alterando a metodologia como trabalham. Por fim, uma sugestão voltada ao objetivo do programa, seria analisar a evolução dos programas de fidelidade e seus diferentes modelos. E também, validar se os requisitos desse projeto, como dar controle ao administrador do programa para criar campanhas com diferentes bonificações, por exemplo, de fato será algo efetivo futuramente.

6. REFERÊNCIAS

A CIÊNCIA e a tecnologia como estratégia de desenvolvimento. **IPEA. Centro de Pesquisa em Ciência, Tecnologia e Sociedade.** Disponível em:

<<http://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/116-a-ciencia-e-a-tecnologia-como-estrategia-de-desenvolvimento>>. Acesso em: 3 nov. 2023.

ABSTARTUPS. **O que é uma startup? Tudo o que você precisa saber.** Disponível em: <<https://abstartups.com.br/o-que-e-uma-startup/>>. Acesso em: 7 nov. 2023.

ANDERSON, David J; CARMICHAEL, Andy. **Kanban essencial condensado: do inglês para o português** por José jr. Seattle: Lean Kanban University Press, 2016. 77 p. Disponível em: https://andresuman.com/wp-content/uploads/2022/04/KanbanEssencialCondensado_Portugues.pdf. Acesso em: 14 nov. 2023.

BATAGLIA, Ariel Guilherme. **Os desafios do home office e a potencial solução nas metodologias ágeis.** 2020. 26 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Pontifícia Universidade Católica de Goiás, Goiânia, 2020. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1172/1/ArielBatagliaTCC.pdf>. Acesso em: 10 dez. 2023.

BECK, K. et al. **Princípios por trás do Manifesto Ágil.** Disponível em: <<https://agilemanifesto.org/iso/ptbr/principles.html>>. Acesso em: 9 out. 2023.

BERNI, Jean Carlo Albiero. **Gestão para o processo de desenvolvimento de Software científico, utilizando uma abordagem ágil e adaptativa na microempresa.** 2010. 77 f. Dissertação (Doutorado) - Curso de Engenharia de Produção, Centro de Tecnologia, Universidade Federal de Santa Maria, Santa Maria, 2010. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/8132/BERNI%2C%20JEAN%20CARLO%20ALBIERO.pdf?sequence=1&isAllowed=y>. Acesso em: 10 nov. 2023.

CAGAN, Marty. **Inspirado: como criar produtos de tecnologia que os clientes amam.** 2. ed. Rio de Janeiro: Alta Books, 2021. 368 p.

COTREL, Benjamin. **How to Run an Async Retrospective?** 2022. Disponível em: <https://www.neatro.io/blog/asynchronous-retrospective/>. Acesso em: 24 maio 2024.

EBIT | NIELSEN. **43ª Ed. Webshoppers.** 2021. 40 slides. Disponível em: <<https://www.negociossc.com.br/materiais-para-download/?categoria=pesquisa&busca=Estudo+Webshoppers>>. Acesso em: 7 out. 2023.

FERNANDES, Jorge Henrique Cabral. Qual a prática do desenvolvimento de *Software*?. **Cienc. Cult.**, São Paulo, v. 55, n. 2, p. 29-33, Abril. 2003. Disponível em: <http://cienciaecultura.bvs.br/scielo.php?script=sci_arttext&pid=S0009-67252003000200021&lng=en&nrm=iso>. Acesso em: 13 nov. 2023.

FIA. **Gamificação: o que é, vantagens e como implementar.** FIA, 30 jul. 2020. Disponível em: <<https://fia.com.br/blog/gamificacao/>>. Acesso em: 17 out. 2023

FLEURY, André Leme *et al.* Alinhando objetivos estratégicos e processo de desenvolvimento em empresas de *Software*. **Production**, [S.L.], v. 24, n. 2, p. 379-391, 24 maio 2013. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0103-65132013005000025>.

GENEROSO, Jackson Lucrécio. **Implementação da metodologia ágil Scrum em uma empresa de marketing digital**. 2022. 60 f. TCC (Graduação) - Curso de Engenharia de Produção Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 2022. Disponível em:

[https://repositorio.ufsc.br/bitstream/handle/123456789/243382/Implementacao_da_Metodologia_Agil_Scrum_em_Uma_Empresa_de_Marketing_Digital_-_final2_assinado%20\(1\).pdf?sequence=1](https://repositorio.ufsc.br/bitstream/handle/123456789/243382/Implementacao_da_Metodologia_Agil_Scrum_em_Uma_Empresa_de_Marketing_Digital_-_final2_assinado%20(1).pdf?sequence=1). Acesso em: 10 dez. 2023.

GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2008. 220 p.

GRAY, Clifford F.; LARSON, Erik W.. **Gerenciamento de projetos: o processo gerencial**. 4. ed. São Paulo: Amgh, 2009. 601 p.

<https://eaesp.fgv.br/sites/eaesp.fgv.br/files/u68/pesti-fgvicia-2023-resumoppt.pdf>

KHLON, Roman. **Retrospectives reinvented: an asynchronous approach for distributed teams**. 2023. Disponível em: <https://www.linkedin.com/pulse/retrospectives-reinvented-asynchronous-approach-teams-roman-khlon>. Acesso em: 24 maio 2024.

KNIBERG, H. **Scrum and XP from the trenches: how we do Scrum; [an agile war story]**. s.l: C4Media, 2007. Acessado em <https://www.agileleanhouse.com/lib/lib/People/HenrikKniberg/ScrumAndXpFromTheTrenchesonline07-31.pdf>

KUBOTA, L. C. TEXTO PARA DISCUSSÃO No 1150. n. 1150, [s.d.]. https://portalantigo.ipea.gov.br/agencia/images/stories/PDFs/TDs/td_1150.pdf

LORENZETTI, J. et al. **Tecnologia, inovação tecnológica e saúde: uma reflexão necessária**. Texto & Contexto - Enfermagem, v. 21, n. 2, p. 432–439, jun. 2012. Disponível em: <https://www.scielo.br/j/tce/a/63hZ64xJVrMf5fwsBh7dnnq/?format=pdf&lang=pt>. Acesso em: 13 nov. 2023

MAFFEI, Debora Cristina. **Uma avaliação do impacto do trabalho remoto no desenvolvimento de Software por meio do scrum**. 2021. 124 f. TCC (Graduação) - Curso de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, 2021. Disponível em: https://repositorio.ufscar.br/bitstream/handle/ufscar/14558/DEBORA%20MAFFEI%20-%20Vers%C3%A3o%20final%2007.07_final.pdf?sequence=1&isAllowed=y. Acesso em: 10 dez. 2023.

MEIRELLES, F. S. USO DE TI NAS EMPRESAS. [s.d.]. <https://eaesp.fgv.br/sites/eaesp.fgv.br/files/u68/pesti-fgvicia-2023-resumoppt.pdf>

MICROSOFT: a origem, o crescimento e os motivos para investir na empresa. **Warren**. Disponível em: <<https://warren.com.br/magazine/microsoft/>>. Acesso em: 4 nov. 2023

MIGUEL, Paulo Augusto Cauchick (org.). **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. 2. Ed. São Paulo: Elsevier, 2012

MOREIRA, Luan de Oliveira. **Desafios na implantação de metodologia ágil SCRUM para gestão e planejamento de projetos de TI**. Gerência de Projetos de Tecnologia da Informação-Unisul Virtual, 2018.

MOTA, Ana Paula. **Daily Assíncrona: Porque você deveria dar uma chance**. 2021. Disponível em: <https://anapaulamota.me/2021/03/23/daily-assincrona-porque-voce-deveria-dar-uma-chance/>. Acesso em: 06 abr. 2024.

MURAKAMI, M. **PWA (Progressive Web Apps): o que é, vale a pena? We-Led Growth**, 2023. Disponível em: <https://weledgrowth.com/blog/pwa-progressive-web-app/>. Acesso em: 9 nov. 2023

NETO, Azuel Aguiar da Silva. **Desenvolvimento de uma aplicação PWA que comporte outras aplicações usando arquitetura de microfrontend**. 2020. 40 f. TCC (Graduação) - Curso de Ciência da Computação, Pontifícia Universidade Católica de Goiás, Goiânia, 2020. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/1037/1/TCC%20-%20Azuel.pdf>. Acesso em: 14 nov. 2023.

OLIVEIRA, Luana Amorim de. **Aplicação da ferramenta Scrum no desenvolvimento de um sistema para automatizar o processo de validação de autodeclaração de candidatos optantes pelas Ações Afirmativas no vestibular na UFSC**. 2022. 80 f. TCC (Graduação) - Curso de Engenharia de Produção Mecânica, Universidade Federal de Santa Catarina, Florianópolis, 2022. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/232226/TCC_Luana_Amorim_de_Oliveira.pdf?sequence=1&isAllowed=y. Acesso em: 10 dez. 2023.

PALADINI, Edson Pacheco. **Recuperação do Brasil Pós COVID-19**. 2020. Entrevista realizada pela ABQ. Disponível em: <https://abqualidade.org.br/academico-edson-pacheco-paladini/>. Acesso em: 04 nov. 2023.

PALAZI, A. **Tudo Inovação | Conheça 5 características essenciais que definem uma startup**. Inova, 27 out. 2022. Disponível em: <https://www.inova.unicamp.br/2022/10/tudo-inovacao-conheca-5-caracteristicas-essenciais-que-definem-uma-startup/>. Acesso em: 7 nov. 2023

REZENDE, Denis Alcides; ABREU, Aline França de. **Tecnologia da Informação: aplicada a sistemas de informação empresariais**. São Paulo: Editora Atlas S.A., 2013. 376 p.

RIES, Eric. **A startup enxuta: como usar a inovação contínua para criar negócios radicalmente bem-sucedidos**. Rio de Janeiro: Sextante, 2009. 288 p. Tradução de: *The lean startup*.

SCHWABER, K.; SUTHERLAND, J. **O Guia do Scrum**. 2013. Disponível em: <https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 14 nov. 2023.

SILVA, Giana. **SCRUM. Analista expert**, 2023. Disponível em: <<https://analistaexpert.com.br/scrum/>>. Acesso em: 14 nov. 2023

SOUZA, Camila Brandina Crispin de; CINTRA, Fausto Gonçalves. PWA: Tecnologia da informação a serviço de pequenas empresas. **Revista Edufatec**, Franca, v. 2, n. 1, p. 1-21, jul. 2018. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2019/03/Camila-B.-Crispin-Souza.pdf>. Acesso em: 15 nov. 2023.

STATE OF AGILE. **SOA16.pdf**. Disponível em: <<https://info.digital.ai/rs/981-LQX-968/images/SOA16.pdf>>. Acesso em: 13 nov. 2023.

TOTVS, E. **Programa de fidelidade: como torná-lo estratégico para a sua empresa?** Disponível em: <<https://www.totvs.com/blog/gestao-varejista/programa-de-fidelidade/>>. Acesso em: 15 out. 2023.

TUDO SOBRE INCENTIVOS. **Pesquisa Panorama da Fidelização no Brasil 2022**. Disponível em: <<https://materiais.tudosobreincentivos.com.br/pesquisa-panorama-da-fidelizacao-no-brasil-2022>>. Acesso em: 15 out. 2023.

WAZLAWICK, Raul Sidnei. **Engenharia de Software: conceitos e práticas**. Rio de Janeiro: Elsevier, 2013. 343 p.

WYSOCKI, Robert K.. **Effective Project Management: traditional, agile, extreme**. 5. ed. Indianapolis: Wiley, 2009. 796 p

7. APÊNDICES

Pergunta	Dev 1	Dev 2	Dev 3
1. Há quanto tempo você trabalha na LiHai Tech?	3 anos	Quase 2 anos	2 anos
2. Qual sua área de atuação?	Desenvolvedor e tech lead	Desenvolvedor <i>backend</i>	Desenvolvedor <i>Backend</i>
3. Há quanto tempo você atua nessa área (mesmo que em outras empresas)?	3 anos	Cerca de 3 anos	Em questão profissional fazem 2 anos, anteriormente programava com projetos menores em escola de programação
4. Quais as principais dificuldades que você encontra para realizar seu trabalho internamente?	Gestão do tempo e priorização de atividades (roadmap)	Não possui dificuldades.	As vezes ter uma visão focada no problema e não no geral (exemplo: onde uma alteração no código pode afetar varios ambitos, e não um só). Não faz testes unitários devido à falta de tempo.
5. Quais as principais dificuldades que você encontra para realizar o seu trabalho em conjunto com a equipe?	Comunicação e falta de padronização. Engajamento da equipe.	Dificuldade para tirar dúvidas com pessoas da equipe de forma rápida.	Agendas para tirar dúvidas dependem do tempo da equipe. As vezes ficar isolado em um único projeto, sem ter conhecimento dos demais projetos para ajudar, quando é necessário. Dependência do Tech Lead para resolver problemas de infraestrutura, e as vezes ele não está disponível.
6. Existem processos que considera que travam ou atrapalham o fluxo do andamento das atividades (gargalos)? Quais?	A busca de qualidade da entrega acaba gerando atraso. Projetos com grande incerteza precisam ser feitos de forma a testar diversos cenários. Da mesma forma, features entregues de forma rápida e com poucos cenários, tem chance de retrabalho pois	Alguns processos de testes atrapalhavam pois o desenvolvedor não testava atividades, então o QA pegava as tasks sem contexto e devolvia com sugestões ou	Em algumas situações, a dependência do Tech Lead para resolver problemas de infraestrutura, e as vezes ele não está disponível. A questão de homologação ou <i>feedback</i> dos clientes que trava o andamentos das atividades.

	quebram em cenários não previstos.	correções que não faziam parte da atividade.	
7. Existem processos que considera como fator de risco em relação à segurança de dados, servidores e informações sensíveis?	Por ser startup, ainda existe um "relaxamento" em relação à processos de segurança. Mas quanto aos dados, as aplicações tem ambientes separados, com ferramentas de criptografia no acesso.	Clientes que possuem acesso aos Bancos de Dados, e também dados do cliente que são compartilhados com toda a equipe (exemplo da chave dos correios).	Baixar dados de produção para criar um ambiente local e resolver os problemas.
8. Existem processos que considera como negativos em relação ao desempenho da equipe?	Disponibilidade da equipe é complicada, então a comunicação da forma como temos hoje é um processo que atrasa. As vezes a distância de acompanhamento dos processos por algumas pessoas também torna difícil manter todos na mesma página dos desenvolvimentos.	Não.	Marcar agendas "fora" do horário de trabalho, pode acabar atrapalhando as questões pessoais.

9. Considera que há clareza na definição e andamento das atividades?	Considera que não está funcionando legal pois muitas vezes não há a informação sobre o andamento das atividades. Mas diz que é uma opinião baseada em comportamento pessoal.	<i>Jira</i> muitas vezes não atualiza, não manda notificação. E as vezes as informações não são passadas completas na <i>daily</i> , mas no geral consegue acompanhar bem.	Fica focado nas atividades apenas dele, então não presta muita atenção nas demais atividades envolvidas. <i>Jira</i> é um bom facilitador na organização das atividades.
10. Considera suficiente a forma de registro das atividades atuais, futuras e passadas da empresa? (Registro de informações, logs, <i>backlogs</i>...)	Registro de informações no código através do GitHub salva versionamentos, então auxilia muito. Ainda há espaço para gerar uma cultura de discussão e registro de atividades.	Nas tarefas atuais o <i>Jira</i> auxilia bem. Em relação à documentação de processos passados ainda falta uma evolução.	Considera que com o que está no <i>Jira</i> está Ok.
11. Considera o tempo diário em que atua como suficiente para cumprir as atividades no prazo?	Geralmente os tempos definidos são suficientes, mas pode ser afetado pela disponibilidade da equipe.	Sim.	Considera que sim.
12. Em relação as atuais cerimônias realizadas pela empresa, considera que são úteis ao bom andamento das atividades e à boa convivência em equipe? (Atualmente temos <i>weekly</i> e <i>daily</i>, além da utilização do <i>Jira</i> e <i>drive</i>)	Sim, está trazendo um acompanhamento das atividades, mas pode ser melhor em relação à participação.	Sim, considera bem importante essas agendas para troca de informação. Gosta bastante.	Sim, considera bem importante essas agendas para troca de informação. Gosta bastante.

13. Acredita que existem pontos de oportunidade para ganhar agilidade dentro do fluxo? Onde? Como?	Descentralização de Responsabilidades em questão de infraestrutura. Utilização da cultura DevOps Cultura de proatividade.	Em relação à troca de informações para tirar dúvidas de atividades, no demais está ok.	GitHub ter um formato de subir automático alterações pra sandbox após code <i>review</i> . Pois traz agilidade pra testes.
14. Acredita que existem pontos de oportunidade para melhorar a documentação das atividades? Onde? Como?	Buscar formalizações porém sem tanta burocracia (pra não formar gargalo). Descrições de Pull Requests ajudam, formalização da discussão em agendas.	Existem pontos, sugestão de criar uma board no <i>Jira</i> com tarefas de documentação dos processos e regras de negócio de cada cliente. É necessário documentar as rotas utilizadas e seus efeitos.	Sumário ao final de uma task, informando o problemas encontrados e como foi resolvido. Isso traria uma quantidade de situações e resoluções encontradas ao longo do tempo.
15. Acredita que existem pontos de oportunidade para melhorar a qualidade dos desenvolvimentos? Onde? Como?	Testes unitários, melhoria contínua olhando dados e otimizar o processo baseado em métricas. Proatividade da equipe na busca de melhorias.	Time participar mais da concepção dos desenvolvimentos, pois agrega valor com conhecimento técnico juntamente ao conhecimento de produto que é empregado na criação das features.	Compilação de "peças" no Lapidar. No caso de uma informação ou comportamento ser utilizado em vários componentes diferentes. Utilizar um "mapeamento" para situações generalizadas.
16. Considera que existem possíveis processos, dinâmicas, técnicas que podemos aplicar para agregar valor aos desenvolvimentos?	Não desenvolver baseado somente no desejo do cliente, mas sim, propor soluções em contrapartida que possam ser mas úteis. Principalmente com	Inclusão a cultura DevOps como um todo pois atua na qualidade do código, na fluidez das	Precisamos de dois pilares: Controle por parte dos administradores e independência para dinamizar o programa.

	<i>feedbacks</i> de outras entregas semelhantes.	entregas e no valor entregue ao produto.	
17. Existem outras sugestões ou pontos de atenção que deseja destacar?	Não	Considera necessário a troca de informação entre as pessoas que entendem mais de algum assunto com os demais, principalmente trazendo novidades no âmbito de tecnologia e de produto. Deixar a equipe na mesma página das atividades que são desenvolvidas ou possíveis processos que pretendemos puxar.	Trazer algumas vezes uma "informalidade" para eventos da empresa ou até mesmo nos processos internos.
