



BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

**USO DE SCRUM PARA DESENVOLVIMENTO FRONTEND DE
APLICAÇÃO DE CONTAGEM DE ALEVINOS**

LUCAS OLIVEIRA PEREIRA DA SILVA

Rio Verde, GO

2023



INSTITUTO FEDERAL GOIANO - CAMPUS RIO VERDE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

**USO DE SCRUM PARA DESENVOLVIMENTO FRONTEND DE
APLICAÇÃO DE CONTAGEM DE ALEVINOS**

LUCAS OLIVEIRA PEREIRA DA SILVA

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Goiano - Campus Rio Verde, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dra. Heyde Francielle do Carmo França

Rio Verde, GO

Maio, 2023

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

SSI586 Silva, Lucas Oliveira Pereira da
u Uso de Scrum para desenvolvimento frontend de
aplicação de contagem de alevinos / Lucas Oliveira
Pereira da Silva; orientadora Heyde Francielle do
Carmo França. -- Rio Verde, 2023.
50 p.

TCC (Graduação em Ciência da Computação) --
Instituto Federal Goiano, Campus Rio Verde, 2023.

1. Desenvolvimento web. 2. Metodologia ágil. 3.
Scrum. 4. Vue.js. 5. Aplicação frontend. I. França,
Heyde Francielle do Carmo , orient. II. Título.

Responsável: Johnathan Pereira Alves Diniz - Bibliotecário-Documentalista CRB-1 n°2376



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES
TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

Identificação da Produção Técnico-Científica (assinale com X)

- Tese
- Dissertação
- Monografia - Especialização
- Artigo - Especialização
- TCC - Graduação
- Artigo Científico
- Capítulo de Livro
- Livro
- Trabalho Apresentado em Evento
- Produção técnica. Qual: _____

Nome Completo do Autor: Lucas Oliveira Pereira da Silva

Matrícula: 2019102201940134

Título do Trabalho: Uso de Scrum para desenvolvimento de aplicação de contagem de alevinos

Restrições de Acesso ao Documento [Preenchimento obrigatório]

Documento confidencial: Não Sim, justifique: _____

Informe a data que poderá ser disponibilizado no RIIF Goiano: 20/06/2023

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

1. O documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
2. Obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
3. Cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Rio Verde, 21 de junho de 2023

Lucas Oliveira Pereira da Silva

Assinado eletronicamente pelo o Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

Heyde Francielle do Carmo França

Assinatura eletrônica do(a) orientador(a)

Documento assinado eletronicamente por:

- Lucas Oliveira Pereira da Silva, 2019102201940134 - Discente, em 21/06/2023 12:49:52.
- Heyde Francielle do Carmo França, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 21/06/2023 12:18:36.

Este documento foi emitido pelo SUAP em 21/06/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 506330
Código de Autenticação: 8d8a490ed4



INSTITUTO FEDERAL GOIANO
Campus Rio Verde
Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970
(64) 3624-1000



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Ata nº 7/2023 - GEPTNM-RV/DE-RV/CMPRV/IFGOIANO

ATA DE DEFESA DE TRABALHO DE CURSO

Ao(s) 29 do mês de maio de 2023, às 19 horas, reuniu-se a banca examinadora composta pelos docentes: Heyde Francielle do Carmo França, Valber Sardi Lopes, Douglas Cedrim Oliveira, para examinar o Trabalho de Curso intitulado “USO DE SCRUM PARA DESENVOLVIMENTO FRONTEND DE APLICAÇÃO DE CONTAGEM DE ALEVINOS” do(a) estudante Lucas Oliveira Pereira da Silva, Matrícula n.º 2019102201940134 do Curso de Ciência da Computação do IF Goiano – Campus Rio Verde. A palavra foi concedida ao(a) estudante para a apresentação oral do TC, houve arguição do(a) candidato pelos membros da banca examinadora. Após tal etapa, a banca examinadora decidiu pela APROVAÇÃO do(a) estudante. Ao final da sessão pública de defesa foi lavrada a presente ata que segue assinada pelos membros da Banca Examinadora.

(Assinado Eletronicamente)

Heyde Francielle do Carmo França

Orientador(a)

(Assinado Eletronicamente)

Valber Sardi Lopes

Membro

(Assinado Eletronicamente)

Douglas Cedrim Oliveira

Membro

Documento assinado eletronicamente por:

- Douglas Cedrim Oliveira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 05/06/2023 08:57:21.
- Heyde Francielle do Carmo Franca, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 29/05/2023 20:27:31.

Este documento foi emitido pelo SUAP em 29/05/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifgoiano.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 499125
Código de Autenticação: a9f567ac08



INSTITUTO FEDERAL GOIANO
Campus Rio Verde
Rodovia Sul Goiana, Km 01, Zona Rural, 01, Zona Rural, RIO VERDE / GO, CEP 75901-970
(64) 3624-1000



Documento assinado digitalmente
VAL BER SARDI LOPES
Data: 06/06/2023 09:04:20-0300
Verifique em <https://validar.iti.gov.br>

Dedico este trabalho a todos aqueles que foram fundamentais em minha jornada acadêmica e pessoal, contribuindo de diferentes formas para o meu crescimento e amadurecimento.

Aos meus pais, pela constante dedicação, amor incondicional e pelos valores que me foram transmitidos. Vocês são minha inspiração e o alicerce que sustenta meus sonhos.

Aos meus professores e orientadora, pela oportunidade de aprendizado e por compartilharem seu conhecimento comigo. Suas orientações e desafios contribuíram significativamente para minha formação e me ajudaram a alcançar resultados além das minhas próprias expectativas.

Aos meus colegas de curso, pelos momentos de troca, colaboração e amizade. A jornada ao lado de vocês foi repleta de desafios, mas também de muitas conquistas e memórias inesquecíveis.

Agradeço também a todos aqueles que duvidaram e duvidam do meu trabalho. Suas dúvidas e questionamentos me impulsionaram a persistir e a provar que sou capaz. Minha história está apenas começando e cada obstáculo superado me fortalece ainda mais.

Por fim, dedico este trabalho a todos que acreditam em mim e no meu potencial. Sua confiança e apoio são combustíveis para que eu siga em frente, buscando sempre novos desafios e superando limites.

Que este trabalho seja uma pequena amostra do que está por vir. Minha dedicação e compromisso com o conhecimento e o desenvolvimento são constantes, e esta é apenas a primeira etapa de uma jornada promissora.

A todos que fazem parte da minha história, meu sincero agradecimento.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que contribuíram para a realização deste trabalho e para a minha jornada acadêmica.

Agradeço primeiramente aos meus estimados professores, cuja dedicação e conhecimento foram fundamentais para o meu desenvolvimento intelectual ao longo do curso. Suas orientações, ensinamentos e desafios foram essenciais para que eu pudesse ampliar meus horizontes e aprofundar meu conhecimento na área.

Um agradecimento especial vai para minha orientadora, Heyde Francielle do Carmo França, pela sua orientação cuidadosa, paciência e apoio ao longo de todo o processo de elaboração deste trabalho. Sua expertise e insights foram de grande valor, guiando-me na direção certa e ajudando-me a aprimorar minhas habilidades de pesquisa e escrita.

Não posso deixar de mencionar meus colegas de curso, que compartilharam comigo desafios, experiências e momentos inesquecíveis ao longo desses anos. A troca de conhecimentos e a colaboração mútua foram fundamentais para o meu crescimento acadêmico e pessoal. Sou grato por cada momento compartilhado e por ter tido a oportunidade de aprender com cada um de vocês.

Quero expressar meu profundo agradecimento à minha família, pelo amor incondicional, apoio e incentivo constantes. Vocês foram minha base durante toda essa jornada acadêmica, e sou imensamente grato por todo o suporte emocional e encorajamento que sempre me proporcionaram.

A todos vocês, meu mais profundo agradecimento. Sem o apoio e contribuição de cada um, este trabalho não seria possível. Vocês fizeram a diferença em minha vida acadêmica e pessoal, e sou eternamente grato por isso.

Muito obrigado!

RESUMO

OLIVEIRA, Lucas. **Uso de Scrum para desenvolvimento frontend de aplicação de contagem de alevinos.** Maio, 2023. 48 f. Monografia – (Curso de Bacharel em Ciência da Computação), Instituto Federal Goiano - Campus Rio Verde. Rio Verde, GO.

Este trabalho de conclusão de curso aborda a implementação de uma aplicação *frontend* destinada a resolver um problema prático na área da piscicultura, aproveitando os avanços contínuos das tecnologias de software e a crescente demanda na indústria de programação. A motivação para esse desenvolvimento surgiu na empresa OnPeixes, que possui uma rede neural especializada na contagem de peixes, mas carecia de uma interface gráfica eficiente para permitir uma interação adequada dos usuários com essa tecnologia.

Com o intuito de atender a essa demanda, foi criada uma aplicação web interativa utilizando o *framework* Vue.js, com o projeto sendo embasado nos princípios da metodologia ágil Scrum no que diz respeito ao *design* e à estrutura. Essa aplicação, além de fornecer uma interface amigável ao usuário, estabelece uma comunicação efetiva com a rede neural localmente.

O desenvolvimento deste projeto ressaltou a eficácia das metodologias ágeis na gestão e no desenvolvimento de soluções de software em um cronograma específico e orientado a objetivos, em consonância com o planejamento dos *sprints*.

Palavras-chave: Desenvolvimento web, Metodologia ágil, Scrum, Vue.js, Aplicação *frontend*.

LISTA DE FIGURAS

Figura 1 – Descrição da figura.	5
Figura 2 – Quadro O Manifesto Ágil.	6
Figura 3 – Funcionamento XP simplificado.	8
Figura 4 – Ciclo do Scrum simplificado.	9
Figura 5 – Gerenciamento Kanban simplificado.	10
Figura 6 – Comparativo Scrum x Kanban simplificado.	11
Figura 7 – Modelo clássico simplificado.	12
Figura 8 – Quadros criados na plataforma trello.	14
Figura 9 – Landing Page que estava sendo desenvolvida.	15
Figura 10 – Tela de login desenvolvida no sprint 1.	16
Figura 11 – Calendário de atividades desenvolvido no trabalho.	17
Figura 12 – Estrutura em árvores de nós na DOM.	20
Figura 13 – Caso de uso do projeto.	25
Figura 14 – Demonstração de uso da aplicação.	25
Figura 15 – Tela de login no figma.	27
Figura 16 – Tela do criação de conta no figma.	27
Figura 17 – Criação do projeto.	29
Figura 18 – Instalando Quasar.	30
Figura 19 – Instalando VueRouter.	30
Figura 20 – Arquivos criados durante o projeto.	32
Figura 21 – Landing Page mobile parte 1.	33
Figura 22 – Landing Page parte 2.	34
Figura 23 – Landing Page parte 3.	35
Figura 24 – Tela de criação de conta.	36
Figura 25 – Tela de login.	37
Figura 26 – Navegador pedindo permissão para acessar a câmera do dispositivo que acessa o site.	38
Figura 27 – <i>Hash</i> da foto tirada pelo botão de acessar a câmera.	39
Figura 28 – Script Vue.js que realiza comunicação com rede neural na porta 5000.	40
Figura 29 – Funcionamento da rede neural em execução na porta 5000.	41
Figura 30 – Formato de imagem que a rede neural espera receber da aplicação frontend	41
Figura 31 – Parte do código que chama função de envio e cria o campo de submissão de imagem	42
Figura 32 – Retorno no <i>browser</i> após envio da imagem para rede neural.	43

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
	SPA Single Page Application
	SSR Server Side Rendering
	NPM Node Package Manager
	JSON JavaScript Object Notation
	YAML Ain't Markup Language
	HTML HyperText Markup Language
	API Application Programming Interface
	RUP Rational Unified Process
	XP Extreme Programming
	PWA Progressive Web App
	DOM Document Object Model
	HTTP Hypertext Transfer Protocol

SUMÁRIO

1 – INTRODUÇÃO	1
2 – REVISÃO DE LITERATURA	4
2.1 Desenvolvimento de software	4
2.2 RUP - Rational Unified Process	4
2.3 Manifesto Ágil	6
2.4 Desenvolvimento ágil de software	7
2.5 Extreme Programming(XP)	7
2.6 Scrum	9
2.7 Kanban	10
2.8 Comparação Kanban x Scrum	11
2.9 Comparação entre metodologias ágeis x tradicionais	12
3 – TRABALHOS RELACIONADOS	14
4 – MATERIAIS E MÉTODOS	19
4.1 Frontend	19
4.2 HyperText Markup Language (HTML)	19
4.3 Cascading Style Sheets(CSS)	19
4.4 Figma	19
4.5 Document Object Model(DOM)	20
4.6 JavaScript	21
4.7 Vue.js	21
4.8 Quasar	21
4.9 PWA	22
4.10 Mobile First	22
4.11 Componentes	22
4.12 Node.js	22
4.13 Node Package Manager(NPM)	23
4.14 Visual Studio Code	23
5 – DESENVOLVIMENTO WEB	24
5.1 Resultados	32
5.2 Telas desenvolvidas com Vue.js e Quasar	35
6 – CONCLUSÃO	44
6.1 Trabalhos Futuros	44
Referências	45

1 INTRODUÇÃO

A computação evoluiu nos últimos anos, tanto em questões de infraestrutura de hardware quanto em uso de software dentro do âmbito corporativo, principalmente com advento de diferentes aplicabilidades que a programação pode oferecer às empresas. Dentro desse contexto, o software se torna uma ferramenta de apoio ao trabalho, possibilitando o facilitamento de recursos tanto na forma de planejamentos quanto na maneira que a empresa pode gerir o fluxo de trabalho dos colaboradores.

Com o passar do tempo e aumento do uso das tecnologias, a criação de software tornou-se cada vez mais complexa, com acúmulo de tarefas a serem desenvolvidas em pequenos espaços de tempo para os funcionários, pois como a tecnologia surge na expectativa de automatizar processos, o time de desenvolvedores acaba sendo sobrecarregado de atividades. Organizar o período de trabalho e a maneira que deve seguir as metas dos projetos torna-se uma tarefa complexa, uma vez em que estiver sendo desenvolvido um produto com pouco entendimento dos requisitos ou não exista a possibilidade de correção de bugs criados durante o desenvolvimento, isso poderá causar danos a empresa, como, por exemplo, a perda de clientes caso a empresa atue em um segmento sólido, no qual os dados precisam ser reajustados imediatamente caso surja um bug em produção, como no caso empresas bancárias.

Na tentativa de auxiliar a execução e planejamento dos projetos de software no qual envolvem processos e demandas com prazos, surge a proposta de utilizar metodologias ágeis, a fim de facilitar na administração tanto do tempo de cada tarefa quanto na formulação de planejamentos juntamente com prazos mais realistas para os colaboradores. (LIMA, 2022).

Uma das características que mais se destaca o scrum está relacionado a comunicação, reduzindo assim burocracias, uma vez em que existe o entendimento do que está sendo trabalhado, as periódicas reuniões facilitam o processo de entendimento do produto trabalhado (NASCIMENTO, 2021).

Desse modo, utilizar o scrum em processos empresariais tanto da programação de um produto quanto no planejamento das etapas necessárias a conclusão do projeto, poderá ocasionar uma maior produtividade entre os colaboradores, facilitando a comunicação e ampliando a compreensão de como solucionar de forma eficiente as necessidades do cliente final que utilizará do software.

Assim, as metodologias ágeis dentro do desenvolvimento de software, são abordagens que buscam facilitar o trabalho interno de empresas, criando fluxos de trabalho e planejamentos aos quais o cliente participa ativamente do projeto que estará sendo desenvolvido, um dos princípios das metodologias ágeis é tornar o trabalho incremental a cada período e manter regulares reuniões a fim de que exista o completo entendimento por parte do time de desenvolvedores da empresa a respeito das necessidades reais e do propósito que a aplicação precisará atingir.

Na tentativa de contornar o tempo de trabalho no projeto a fim de otimizar etapas no mercado de trabalho, surgem as diversas metodologias ágeis, tais como, Scrum, Kanban, XP, Lean, Smart. Todas possuem em comum a capacidade de aumentar a produtividade do time de desenvolvimento, assim como possibilitando um planejamento prévio das tarefas a serem realizadas em um período limitado de tempo, dentro desse período, existindo a possibilidade de incrementar o que há de ser feito nas subsequentes semanas ou meses até ser atingido os objetivos de conclusões dos projetos.

Métodos tradicionais de gerência de projetos podem apresentar acentuados riscos

de insucesso, sendo relacionado do tempo de planejamento, que podem ser solucionados por uso e aplicação de metodologias ágeis (OLIVEIRA et al., 2020). As metodologias ágeis mais utilizadas empresarialmente são o Scrum e o Kanban, dentro do contexto de desenvolvimento de software, são ferramentas que possibilitam o relacionamento entre pessoas, buscando eficiência através da execução a medida em que surgem novos requisitos. Ambas são métodos dinâmicos de trabalho, justificando o amplo uso e aplicabilidade dos conceitos aos quais podem utilizados além de trabalhos relacionados a software.

A principal abordagem que as metodologias ágeis são capazes de trazer para um projeto está ligado a capacidade de montar um time multidisciplinar e fazer com que seja trabalhado e fragmentado o projeto independente do tamanho, dividindo tarefas em pequenas partes, de forma com que incrementalmente o produto no qual os desenvolvedores ou a empresa está trabalhando possa progredir até que todos os requisitos funcionais sejam atendidos. E se houver necessidade de alteração, também existirá um período de inferência, principalmente por parte do cliente para avaliar e realizar testes enquanto a aplicação se desenvolve, para validar a expectativa com a realidade do que a aplicação poderá proporcionar.

Associado ao desenvolvimento de software, o presente trabalho busca solucionar uma necessidade da empresa On-Peixes. A empresa possui uma aplicação de rede neural que realiza a contagem de peixes através fotos, isso é, a rede recebe a imagem do tanque de peixes e a partir de então retorna a quantidade existente de peixes no tanque. No entanto, a funcionalidade existe em um ambiente controlado no Google Collab, ambiente integrado com editor de código na qual é submetido as imagens de forma manual pelo programador, sem alguma possibilidade de interação com cliente final. Existindo a necessidade de implementação do produto da empresa, através um site web que posteriormente poderá se conectar à Inteligência Artificial para que o cliente possa utilizar do produto construído. Após a aplicação de *Machine Learning* se conectar ao frontend web, o cliente final poderá utilizar a aplicação de fácil acesso, remotamente usando celular pessoal.

O método tradicional de contagem de peixes, é realizado por o piscicultor de forma unitária, cada peixe é contado individualmente, sendo necessário retirar cada um do tanque a fim de se obter o número exato quando for necessário efetuar a venda do lote dos peixes. Tal processo é muito custoso, pois a quantidade de tempo gasto na contagem é proporcional ao número de peixes e ao tamanho do tanque. A aplicação web servirá para tornar o produto da empresa On-Peixes mais completo, no quesito de funcionalidades para o auxílio ao software da inteligência artificial.

Nesse contexto, esse trabalho visa como principal projetar e implementar o frontend de uma aplicação web utilizando como base a metodologia scrum para o planejamento de *sprints* e levantamento de requisitos a fim de estruturar as tarefas do presente trabalho. Será utilizado modernas tecnologias emergentes no mercado de desenvolvimento de software no contexto de frontend, em específico, o *framework* Vue.js e Quasar para construção da interface.

A metodologia ágil Scrum será a técnica escolhida para o desenvolvido deste trabalho, contendo planejamentos, levantamento de requisitos, implementação em ferramenta estática de protótipo de design até programação dinâmica para suprir as necessidades específicas do projeto abordado, em conjunto com técnicas modernas de desenvolvimento de software.

As *sprints* foram divididas de forma em que o trabalho foi fragmentado em pequenas atividades, e a partir de então, realizar levantamentos a respeito se o feedback do cliente final está sendo positivo de forma em que a aplicação está fazendo sentindo e satisfazendo a necessidade real que existe por trás da ferramenta de uma aplicação web.

Portanto, espera-se desenvolver uma aplicação web frontend, que atenda aos requisitos identificados na seção de levantamento de requisitos, incluindo, portanto, o cadastro, login e envio de imagens da própria aplicação para a rede neural que realizará a contagem de peixes e retornará ao usuário a quantia exata com base na foto submetida, paralelo, portanto, ao desenvolvimento, será utilizado a metodologia ágil scrum para estruturar o trabalho. Espera-se deste modo desenvolver uma aplicação web, para que esta possa se conectar com a rede neural, de forma em que o usuário envie uma imagem do tanque de peixes e seja retornado a tela do usuário a quantidade exata de peixes contida no tanque, sendo resultado da contagem da rede neural, devolvida a aplicação web frontend.

2 REVISÃO DE LITERATURA

Nesta seção serão detalhadas métodos e conceitos aos quais fundamentam e auxiliam o desenvolvimento de software, iniciando desde métodos tradicionais como desenvolvimento em cascata até implementações modernas como o Scrum.

2.1 Desenvolvimento de software

Indústrias de software estão em evolução de forma crescente, possuindo sistemas cada vez mais abrangentes e complexos, necessitando, portanto, de uma atenção especial em projetos a fim de que seja atendido as necessidades dos clientes (MARTINS et al., 2017).

Dentro do contexto moderno, o software se torna, portanto, um facilitador de relações, quer sejam estas de trabalho ou pessoais, expandido-se então em diversas áreas e auxiliando em diferentes ramos de trabalhos. Sua principal relevância foi facilitar trabalhos repetitivos, como, por exemplo, serviços de companhia telefônica ou agendamentos em aplicativos que eliminam a existências de filas de pessoas físicas para o serviço bancário. Desse modo, softwares podem fornecer auxílio tanto no processo de ensino e aprendizado, quanto para profissionais da saúde a fim de facilitar o tratamento com pacientes, conforme exemplificado por (PINCELLI-NETTO, 2016).

O uso de tecnologias da informação foi alterado durante as últimas décadas, sendo, portanto expandidos a diferentes âmbitos sociais, causando avanços em setores como acadêmicos, industriais e financeiros (PRAZERES, 2015).

Em 1968 e 1969 foi quando surgiu o termo "Engenharia de software" durante as conferências realizadas pelo Comitê de Ciências da Organização do Tratado do Atlântico Norte (OTAN) (SATO, 2007). Desde então, foram criadas discussões a respeito de como estruturar um produto aos quais se utilizara de recursos tecnológicos, buscando uma padronização teórica para o que seria criado e estruturado dentro do universo da programação.

A Engenharia de software surgiu para corrigir impasses internos de softwares (LESSA; JUNIOR, 2009). A fim de tornar escalável as aplicações criadas oriundas da tecnologia, a Engenharia de Software agregou uma série de recursos e formulações técnicas a fim de que fossem criadas aplicações robustas e com manutenções de forma simplória.

Conforme citado por (AMORIM; GUEDES, 2017) devido ao aumento da criação de softwares, fora preciso definir regras dentro da Engenharia de software, tais como o desenvolvimento em cascata, prototipação e espiral.

1. Modelo em Cascata: Inicia a modelagem da aplicação partindo do pressuposto da orientação a objetos.

2. Modelo de prototipação: Fornece um rascunho, a fim de esclarecer ideias para o processo de criação do software.

3. Modelo Incremental: Sendo uma evolução do modelo em cascata, as modelagens para programar a solução baseiam-se de forma interativa.

2.2 RUP - Rational Unified Process

Rup é um acrônimo, em Português significa Processo Unificado Rational, sendo uma camada de processos, aos quais existem níveis elevados de organização para times de

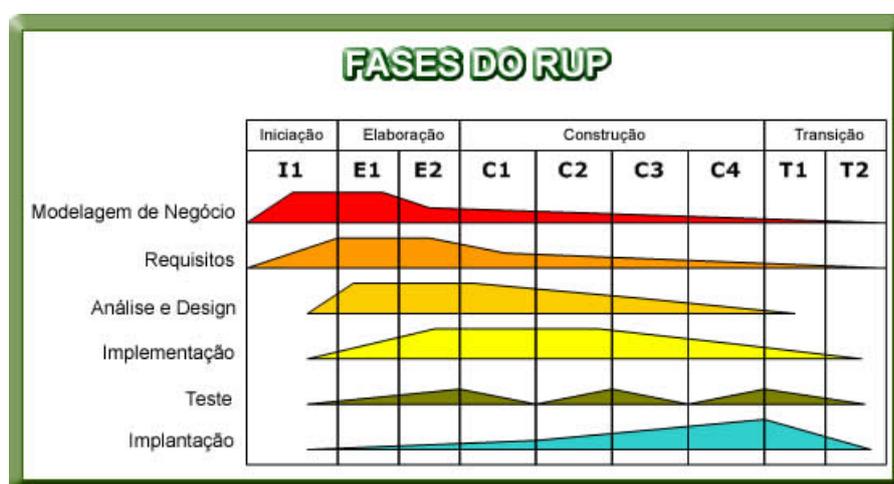
desenvolvimento no qual são divididos entre muitos colaboradores.

Segundo (PEREIRA, 2017) a metodologia RUP é iterativa de desenvolvimento, metodologia que pode ser adaptada com base nas necessidades e tamanho dos projetos. Com base na característica supracitada, contém aspectos semelhantes a uma metodologia ágil.

A principal característica do RUP quando surgiu, foi essencialmente trazer ao mercado, na década de 90, uma metodologia que tivesse como base os conceitos da linguagem UML(Unified Model Language), modelando sistemas utilizando o paradigma de Orientação a objetos. Causando diretamente, portanto, um aumento na produtividade da equipe de trabalho, pois permite com que a equipe que estiver trabalhando no produto interno, poderá acessar os modelos de atividades a serem criadas. (NONEMACHER et al., 2003).

As fases do Processo Unificado da Rational podem ser observados na figura 1

Figura 1 – Descrição da figura.



Fonte: Infoescola (2021)

1. Fase de iniciação: planejamento técnico voltado a reunião com cliente, elaborando, portanto um plano de execução do projeto;
2. Fase de Elaboração: inicializando a modelagem do produto, de forma genérica;
3. Fase de construção: sendo responsável pela maior parcela de contribuição com código, portanto, inicializando o setup dos componentes funcionais dos sistemas;
4. Fase de transição: entrega do software e colhimento de feedbacks a respeito de quais utensílios podem ser melhorados.

Conforme descrito por (MANIKA,) e com base em figura 1, durante as fases, utiliza-se de seis *workflows* de processos, o eixo horizontal representando o tempo e o ciclo de vida que aquele processo leva para se desenvolver, enquanto no eixo vertical, é representado as atividades que precisarão serem desenvolvidas, tais como, componentes, fluxos de trabalho e atividades.

Desse modo, as fases da metodologia RUP podem ser consideradas cíclicas, uma vez em que as tarefas são realizadas e supervisionadas a medida em que há uma progressão nos trabalhos executados, sempre seguindo de forma cíclica o planejamento inicial levantado.

2.3 Manifesto Ágil

Com base no livro Processos Ágeis de Desenvolvimento de Software de (MEDEIROS; NETO,), em 2001 iniciaram discussões a fim de definir padrões e princípios aos quais poderiam gerar valores para a criação de um produto programável e poderia auxiliar a velocidade com que as entregas de softwares eram realizadas, baseando-se na proposição de que os times de desenvolvimento cada vez mais possuíam burocracias e empecilhos em seus trabalhos.

A partir desse movimento foi possível solidificar as ideias e conceitos a respeito do desenvolvimento de software orientado a uma maneira ágil tanto de pensamento quanto de planejamento e execução de tarefas no dia-dia, sendo possível criar a formulação de uma *squad*, sendo um grupo de colaboradores multifuncionais trabalhando no mesmo projeto. Esse movimento deu origem, portanto ao Manifesto Ágil.

Figura 2 – Quadro O Manifesto Ágil.

Quadro 2.1 O Manifesto Ágil. Adaptado de [AGILE MANIFESTO 2009]

Nós estamos descobrindo melhores maneiras de desenvolver software, fazendo e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:

Indivíduos e interação entre eles 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

Mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Assinam este manifesto:
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas.

Fonte: Adaptado de [AGILE MANIFESTO 2009]

Como reação do movimento, foi gerado o quadro proposta abaixo, fundamentando, portanto, doze princípios conforme a listagem referenciada por (MEDEIROS; NETO,) no capítulo 2 do livro Processos Ágeis de Desenvolvimento de Software.

1. Antes de tudo é essencial priorizar entregas contínuas e de valor;
2. O projeto pode estar sujeito a alterações;
3. De duas em duas semanas é necessário realizar uma entrega;
4. *Squads* multidisciplinares;
5. O ambiente de trabalho deve ser propício à produtividade;
6. Priorização de conversas;
7. O software deve ser funcional;
8. Os processos devem fluir, mesmo que este esteja lento;
9. Atenção, adjunta a habilidade técnica, maximizam a metodologia;
10. Maximização do trabalho de forma simplória;
11. Escolha da melhor arquitetura para os projetos;

12. Reflexão do time, a fim de melhorar habilidades pessoais;

A partir dos conceitos supracitados, surgiram diversas metodologias ágeis, as quais dominaram o mercado de trabalho não somente no âmbito de tecnologia, mas também dominou o segmento de RH para uma melhoria nos fluxos internos de execuções de tarefas, sendo planejadas e estruturadas.

2.4 Desenvolvimento ágil de software

As metodologias ágeis são maneiras de organizar a criação de um software utilizando de ciclos flexíveis, facilitando a pressão de criar um produto em detrimento de alta demanda no mercado de trabalho. (FARIAS, 2022). Com a criação das *squads*, sendo grupo de colaboradores que corroboram e trabalham em prol do desenvolvimento de um projeto, tornou-se necessário formular ciclos de reuniões, assim como períodos estimados para que as tarefas internas fossem resolvidas entre os times de desenvolvedores e engenheiros de software, para maximizar as entregas e aumentar o campo de compreensão entre o que está sendo criado e o que precisará ser implementado futuramente.

Mediante o uso de metodologias ágeis, são priorizados atividades a serem desenvolvidas, bem como o planejamento a respeito de como solucionar tarefas, fragmentando-as em períodos, juntamente com reuniões. Deste modo, as técnicas ágeis causaram uma maior produtividade entre os times de desenvolvimento, além de beneficiar também outras áreas, as quais vão além de codificações, sendo estas, habilidades interpessoais como comunicação e facilitação na troca de experiências(PONTES; ARTHAUD, 2018).

Empresas nas quais sempre existe uma mudança a ser realizada enquanto uma aplicação está em produção e possui como base milhares de usuários, como, por exemplo, startups, adotando a metodologia de trabalho ágil, é possível lidar e prever as mudanças que surgirão em necessidades de serem tratadas.

Assim, o fluxo de tratamento de um software poderá ser incrementado ou decrementado, a medida em que surgir mudanças repentinas a serem realizadas, agilizando a qualidade do produto final que estará sendo entregue ao usuário. O desenvolvimento de software orientado de forma ágil trouxera avanços para que a tecnologia acompanhasse a crescente demanda de entrega requerida pelo mercado de trabalho.

Com uma abordagem flexível e descentralizada, fora proposto, assim, a minimização de gastos desnecessários, bem como burocracias para a entrega de valor a um cliente que irá consumir um software.(SATO, 2007). Maximizando então processos que outrora foram considerados por serem ultrapassados devido o desgaste ocasionado pela falta de planejamento em processos internos empresariais.

2.5 Extreme Programming(XP)

O processo do Extreme Programming é baseado em torno de práticas que buscam reduzir o tempo de criação do software alinhado a um produto que satisfaça o cliente e gere valor (BORTOLI, 2017). O XP é uma metodologia que buscou solucionar o impasse de longos ciclos de trabalho dentro da criação e elaboração de um software, propondo, portanto, uma abordagem mais flexível de descentralizada para curtos períodos de trabalho em uma determinada tarefa específica dentro do que fora proposto no escopo do projeto.

Nas últimas décadas, mostrou-se uma metodologia com impacto positivo de produção de software, adotando uma postura de eficiência entre os times de desenvolvimento, conforme analisou (FADEL; SILVEIRA, 2010). Conforme referenciado outrora pelo autor,

as ideologias do XP são: 1. Comunicação, que consiste na comunicação ativa entre os colaboradores do projeto.

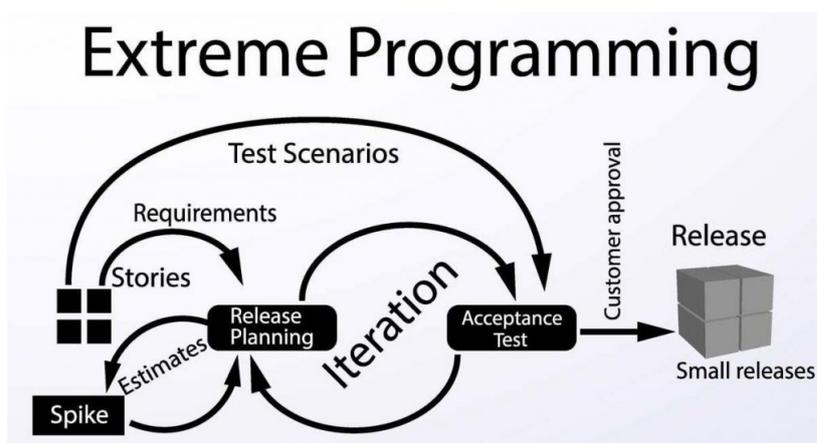
2. Simplicidade, consiste no princípio de que a aplicação desenvolvida precisa estar funcional.

3. Feedback, entre as reuniões, deve ser relatado impasses existentes.

4. Respeito, sendo necessário manter a comunicação entre o time para que os colaboradores estejam em harmonia.

Segundo (SILVA; SANTOS; SHIBAO, 2019), o XP deve acontecer somente quando houver uma demanda explícita por parte do cliente, obrigando a não ser criado software desnecessário, mas seguindo as necessidades reais a serem implementadas. Desse modo, é notório que a metodologia XP apresenta um ganho de desempenho em projetos a medida em que estes possuem fluxos aos quais os colaboradores podem decidir de uma maneira menos rígida quando comparada ao nascimento da área de desenvolvimento e criação de softwares. Segundo (RAMOS et al.,) a principal vantagem está relacionada a possibilidade incremental no quesito melhoria do software que estiver sendo desenvolvido. Como grande parte do projeto depende de uma etapa direta de feedback, a metodologia acaba se prendendo a necessidade de produtividade entre os desenvolvedores, para que nenhuma das etapas iniciais da ideologia XP se prenda na execução.

Figura 3 – Funcionamento XP simplificado.



Fonte: (Nimblework, 2023)

Por se tratar de uma metodologia tão eficiente, esta não poderá se restringir somente a criação de softwares, mas também poderá ser utilizada para equipes que trabalham de forma estrutural, por exemplo, na área de *marketing*, no qual os colaboradores buscam melhorar um produto existente ou elaborar um que satisfaça o gosto dos públicos finais, sendo possuindo em mente que o produto criado poderá ser alterado e conforme o tempo, melhorado.

O XP não é considerado necessariamente uma metodologia ágil, pois as ações desenvolvidas no projeto estão acima de pessoas, logo os processos internos ocupam o principal cenário da criação final, o que pode causar uma resistência aos desenvolvedores, uma vez em que, independente do que houver, o projeto precisará andar conforme o planejamento inicial traçado e este deverá ser seguido até a aplicação ser entregue ao destino.

A principal característica da metodologia supracitada, é acreditar que o software pode ser desenvolvido de forma rápida, de modo em que cada alteração criada possa ser exposta ao cliente a fim de colher feedbacks e planejar novas implementações, sempre se

baseando completamente sobretudo nas necessidades individuais do alvo que utilizará o produto programado.

2.6 Scrum

O scrum consiste em uma das mais populares metodologias ágeis de desenvolvimento, trazendo consigo todas as características e organizações conceituais fundamentadas na seção anterior. Sendo uma metodologia ágil e flexível, o scrum define um processo iterativo e incremental, passível de ser utilizado em todo e qualquer processo complexo em um time de desenvolvedores (BISSI, 2007).

Sendo idealizado em 1993, o scrum ainda é uma ideologia atual para lidar com grandes projetos e protótipos de projetos dentro do escopo de squads. Causando deste modo, uma maneira inteligente e ágil de gerenciar os projetos, considerando acima de tudo, reuniões e feedbacks constantes.

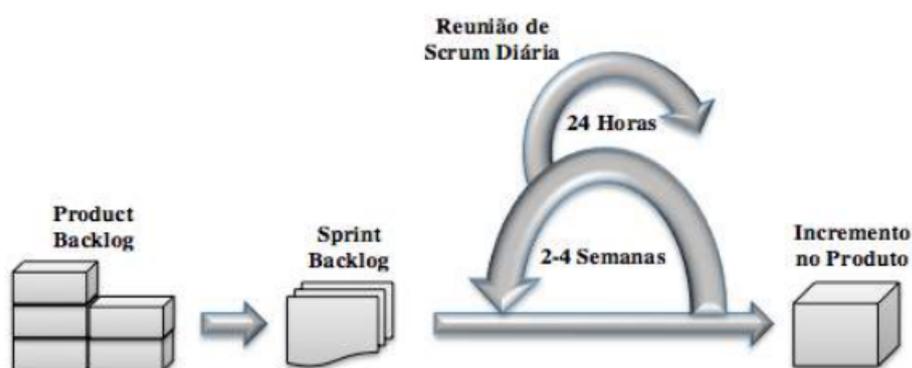
Um dos problemas que este *framework* soluciona é a transparência e aumento do rendimento entre os agentes que estiverem participando do projeto planejado, causando um relacionamento ativo tanto do cliente quanto da equipe. Este pode ser aplicado a qualquer produto orientado as atividades (BISSI, 2007).

Para o scrum, é necessário ser definido ciclos de tarefas para serem desenvolvidas, usualmente estes ciclos são curtos, pois se haverá de ser realizado alguma alteração, esta poderá ser realizada de forma natural e adaptativa. Deste modo, esse foi o ponto fundamental que alavancou o *framework* para ser utilizado no mercado de trabalho.

Essa quebra de muitas tarefas para a fazeres curtos, permite alavancar uma aplicação, de modo em que a complexidade será descentralizada.

Os ciclos de vida do scrum são baseados em *sprints*, sendo estas reuniões curtas durante as etapas e semanas de execução do trabalho que estiver em demanda. A duração não poderá ser superior a 30 dias de execuções diretas do projeto, a cada *sprint* é definido um novo escopo de atuação de trabalho e se possível, será proposto novas alterações.

Figura 4 – Ciclo do Scrum simplificado.



Fonte: (CASTRO et al., 2017)

O Product Backlog consiste em uma importante etapa do scrum, uma vez que são definidos a listagem de prioridades que precisarão serem executados, a fim de que gere algum valor de negócio.(CASTRO et al., 2017). A partir dessa lista, a equipe de trabalho irá definir as metas para as, sprints que o fluxo de execução deverá seguir, separando também as prioridades imediatas.

Entre as *sprints*, deverão ocorrer reuniões diárias (*Daily Meeting*), sendo encontros no qual o encontro deverá ter duração de no máximo 15 minutos, a fim de revisar, planejar ou implementar o escopo no qual estiver sendo trabalhado ou discutido, a fim de incrementar a visão do projeto e sanar dúvidas específicas caso exista.

2.7 Kanban

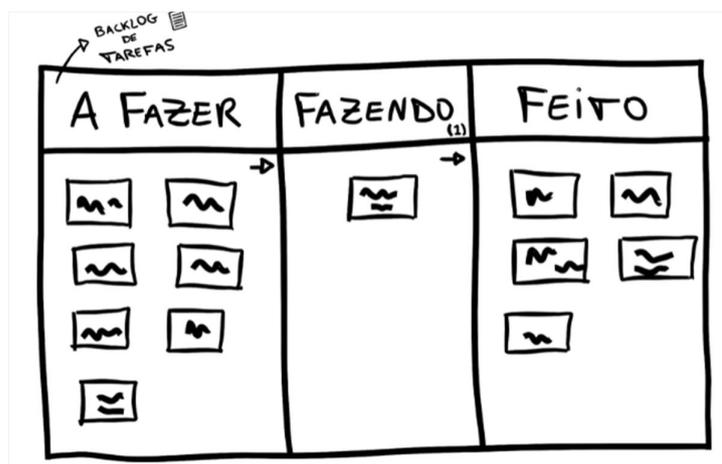
Kanban consiste em uma metodologia ágil de criação de software, com enfoque para o processo prático, no qual o objetivo está em melhorar um produto que será criado ou mantido. (ARRUDA, 2012). É uma metodologia bastante utilizada em processos no qual existem diferentes etapas de realizações, principalmente nas áreas em que os produtos passam por uma série de detalhes antes de serem entregues ao consumidor final.

As bases do Kanban são semelhantes as do manifesto ágil, logo, é valorizado o progresso do valorizado, portanto, enquanto este ainda há de ser executado e melhorado. Um dos exemplos práticos é a linha de produção do Sistema Toyota, que foi a criadora do método Kanban na década de 1960, assim foi surgindo a necessidade de incrementação a cada ciclo de trabalho que este fora executado por trabalhadores.

Como parte incremental dos processos, o Kanban trouxe a necessidade de computar o tempo necessário para realização de uma tarefa e seguir o mesmo, para manter o controle do que estaria sendo criado. (ARRUDA, 2012).

Dentro do contexto de software, o kanban serve com grande utilidade, pois permite com que os "cartões" sendo as tarefas ou "*tasks*", podem ser alteradas e até mesmo evidenciadas ao cliente no mesmo momento em que a aplicação está sendo desenvolvida, isto é, o cliente pode incrementar, bem como a equipe de desenvolvimento, diretamente no valor em que o produto irá gerar para o consumidor final e/ou o próprio cliente.

Figura 5 – Gerenciamento Kanban simplificado.



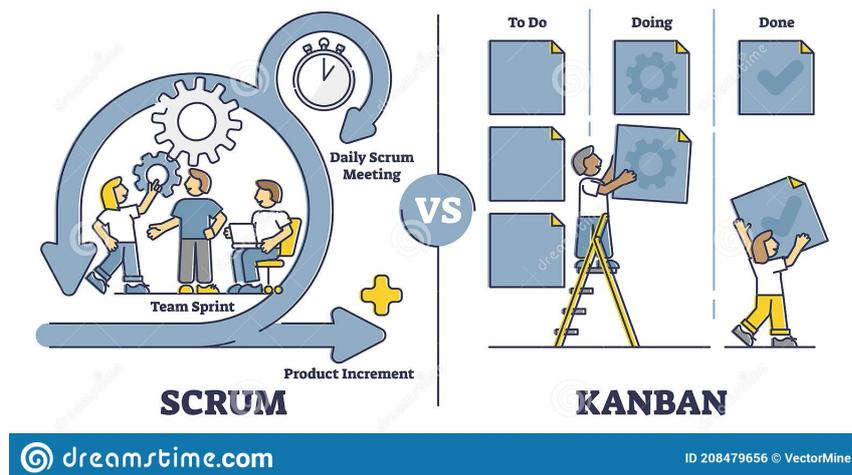
Fonte: (NEOVERO, 2023)

Não se atendo somente a planejamentos de projetos do zero, o kanban possui bastante expressividade em projetos aos quais são legados e apresentariam um árduo planejamento para codificação de novas funcionalidades, trazendo, portanto, uma agilidade em processos e execuções de complexas tarefas (SILVA; SANTOS; NETO, 2012).

2.8 Comparação Kanban x Scrum

Ambas as metodologias supracitadas são amplamente utilizadas em diversos setores do mercado de trabalho, sendo, portanto, as mais recomendadas para desenvolvimento incremental e escaváveis de um software. Em especial, cada uma possui particularidades e serão comparadas, portanto, a fim de se aproveitar o melhor de cada tecnologia em volto do trabalho criado.

Figura 6 – Comparativo Scrum x Kanban simplificado.



Fonte: (DREAMSTIME, 2023)

O Scrum permite uma hierarquia dentro da metodologia de planejamento e execução, sendo inicialmente estruturada em: Scrum master, sendo o responsável por manter a estrutura e planejamento do Scrum em execução, o *Product Owner*, sendo responsável pelo levantamento de requisitos e o time de desenvolvimento, sendo, portanto, colaboradores para que o projeto esteja sempre em execução. (GENARI; FERRARI, 2016).

Enquanto isso, o Kanban não utiliza dos mesmos métodos hierárquicos para a organização dos, a fazeres dentro do contexto de planejamento e aplicação para a criação de uma solução programável. Todos os membros da equipe são de suma e igualitária importância. (GENARI; FERRARI, 2016)

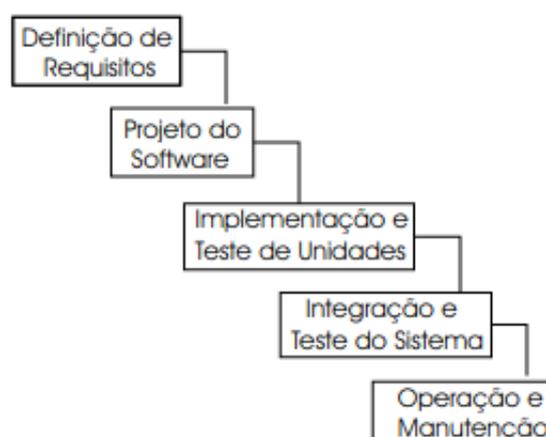
No Scrum, a equipe irá se basear em *sprints* curtas, a fim de que o time sempre seja alerta a respeito das implementações, contudo, se a *squad* de trabalho for muito grande, poderá existir falha na comunicação e compreensão das atividades, mas este pode ser um fator facilitador para a organização a longo prazo das tarefas. Contudo, se a necessidade da equipe for se adaptar a repentinas mudanças, como, por exemplo, em um bug em produção para milhares de pessoas, o Kanban torna-se a melhor alternativa, uma vez em que aquele problema poderá ser tratado de forma imediata, todavia, essa interrupção imediata pode ocasionar atrasos para futuras tarefas que haverão de serem implementadas.

Com base nisso, é importante salientar que toda tecnologia possui vantagens e desvantagens e ambas as metodologias servem para formular uma base para um produto melhor, no qual este passará por diferentes etapas incrementais, a fim de que a empresa possa acompanhar o ritmo frenético de entrega de software e mesmo assim poderá inovar de forma ágil e otimizada. Como são metodologias ágeis, qualquer que seja uma das duas, haverá um ganho de desempenho para a equipe de trabalho.

2.9 Comparação entre metodologias ágeis x tradicionais

Dentro do contexto de criação e mercado de trabalho na Engenharia de Software, otimizar o tempo e funções de trabalho, sobretudo específicas de cada tarefa, é uma parte extremamente fundamental, tão quanto codificar e criar soluções é pensar em como orquestrar o tempo a se fazer. Cada vez mais os Engenheiros de Software precisam entregar mais demandas em um tempo menor, visto que há muito trabalho a ser feito.

Figura 7 – Modelo clássico simplificado.



Fonte: (SOARES, 2004)

Pensando nisso, metodologias ágeis conforme supracitadas facilitam o ganho de desempenho tanto individuais, quanto no valor agregado ao produto que há de ser entregue.

No modelo clássico de desenvolvimento, as tarefas eram dispostas de forma sequenciais, para que ao terminar uma tarefa, outra tarefa haveria de vir, com novas prerrogativas e necessidades (SOARES, 2004). Logo, se uma tarefa não progredisse, todo o projeto acabaria em atraso, pois a sequência de execução estaria congestionada, além disso, o time irá se tornar improdutivo e não apresentará crescimento expressivo em *hard skills*.

As metodologias tradicionais vêm aumentando sua popularidade, em ambientes onde o escopo trabalhado não é alterado enquanto os colaboradores exercem sua função, diferentemente da indústria de software (PERIDES; BARROTE; SBRAGIA, 2021). Na área de desenvolvimento, é essencial em que os colaboradores estejam cientes do que estão construindo e cientes também que as funções e ordens de operações podem ser alteradas, sempre se baseando no escopo do projeto atual e na metodologia empregada para acelerar o processo criativo dos colaboradores.

Segundo o trabalho de (SOARES, 2004), a ideia por trás em implementar metodologias ágeis se concentra em pessoas no detrimento de processos, as pessoas são colocadas em prioridades para que o fluxo de trabalho seja executado. Assim, cabe ao time escolher dentre todas as metodologias a que for mais suave a sua implementação e de fácil adaptação entre os *squads*. Se a metodologia escolhida resolver problemas internos e de falhas de comunicação já apresentará um avanço para que as entregas estejam em uma abordagem moderna, funcional e progressiva.

Dentro do mesmo trabalho, o autor referencia que o uso de metodologias ágeis modernas irá ampliar os resultados internos empresariais e sua adoção poderá também

acelerar as datas de entregas e aumentar a confiabilidade entre colaboradores, dentro do requisito da Engenharia e Arquitetura de Software.

Em tecnologia é muito comum as empresas serem fundadas inicialmente como startup para conseguir recursos aos seus projetos e fundamentalmente a velocidade com que os desenvolvedores têm precisam ser acentuados, desse modo, surge a necessidade de se desenvolver um *MVP* de um produto cada vez melhor em um pequeno prazo para agregar valor ao projeto.

Assim, a escolha na metodologia que a empresa irá seguir diz muito a respeito sobre a longevidade do produto em desenvolvimento e sobre a capacidade em que as *squads* conseguirão progredir dentro da *feature* escolhida. Se o produto for bom e inovador, mas levar muito tempo a ser desenvolvido, o valor em que a startup conseguirá agregar a empresa poderá ser menor do que se este fosse desenvolvido de uma maneira mais ágil e rápida.

3 TRABALHOS RELACIONADOS

O trabalho de (OLIVEIRA; FRANÇA; ARAÚJO, 2022) se baseou em uma abordagem de scrum e kanban, nomeada de ScrumBan, uma escolha híbrida que recorre aos dois principais *frameworks* de metodologia ágil, o autor criou uma aplicação web completa, proposta de uma plataforma para adoção de animais online, tanto frontend quanto backend formulando o planejamento sobre as metodologias supracitadas, sendo a principal ação destinada ao planejamento e calendários a serem incrementados.

O autor usou cartões na plataforma Trello, que é uma plataforma que auxilia no gerenciamento de trabalho, facilitando, portanto, a distribuição de tarefas e trabalhos em aberto, especificamente formulou-se oito quadros que significam os tempos e tarefas a serem completadas tanto ao nível de programação quanto ao nível de estudo das ferramentas e até reuniões a respeito de marketing.

Cada quadro na plataforma Trello significa as tarefas de cada área de trabalho, assim como a disposição de planejamentos de execução e reuniões a respeito do projeto em desenvolvimento.

Figura 8 – Quadros criados na plataforma trello.



Fonte: (OLIVEIRA; FRANÇA; ARAÚJO, 2022)

O trabalho do projeto foi dividido em:

1. Levantamento de requisitos.
2. Gerenciamento da metodologia (ScrumBan).
3. Modelagem e casos de uso.
4. Prototipação.
5. Desenvolvimento.

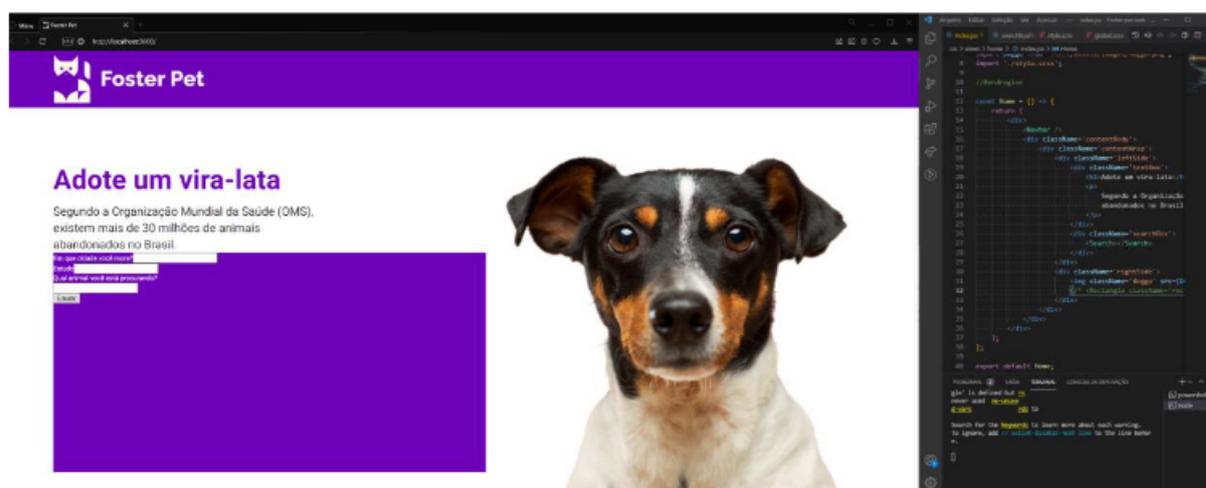
A partir de então, é possível afirmar que as metodologias ágeis, principalmente o scrum traz a perspectiva sobre produtividade ligada a equipe de trabalho, uma vez em que é possível pegar um projeto extenso e conseguir produzir módulos eficientes e condizentes com os levantamentos da necessidade dos usuários, pois o desenvolvimento sempre será contínuo e incremental, possibilitando reajustes no decorrer de cada processo fragmentado em cartões ou tarefas em aberto.

A principal vantagem apresentada pelo artigo citado está ligado ao fato de que o desenvolvedor do projeto não iniciou programando de fato, iniciou escolhendo uma

metodologia de planejamento e trabalho e a partir de então montou um cronograma dos requisitos funcionais ao projeto e não funcionais.

Logo nota-se também que não se obtinha todo o conhecimento a respeito das tecnologias empregadas no projeto, sendo React.js para o frontend e Java para o backend, contudo houve um período de estudo até o entendimento da *stack* de desenvolvimento, isso possibilitou a capacitação e preparo para que as tecnologias fossem aprendidas em um prazo específico assim como implementadas na prática tudo dentro do período necessário e destinado para cada tarefa.

Figura 9 – Landing Page que estava sendo desenvolvida.



Fonte: (OLIVEIRA; FRANÇA; ARAÚJO, 2022)

Existe a modularidade nas metodologias ágeis, facilitando tanto o processo de aprendizagem quanto o processo de planejamento, de modo em que cada tarefa ou cada cartão escolhido precisa de um tempo a ser gasto para estar em perfeito estado.

O trabalho de (CARNEIRO et al., 2022) consistiu em analisar os impactos que práticas ágeis causam ao cliente, bem como o valor gerado quando o desenvolvimento de sistemas é orientado ao scrum, se tornando, portanto, diferente do anterior trabalho mencionado, uma vez em que fora utilizado um estudo de caso para compreender as expectativas e necessidades de um cliente de software quando a gestão ágil está em execução de projetos reais.

Existiu uma coleta de informações nas dependências da empresa escolhida que trabalha com desenvolvimento de software voltado à logística empresarial na cidade de Maringá, estado do Paraná, contando com a participação direta do gerente de projetos, montando, portanto, um estudo qualitativo descritivo. Abordagem escolhida foi uma entrevista semiestruturada, aplicada por funcionários locais e foi seguido um roteiro no qual fornece um caminho para indicadores fundamentais para compreensão da pesquisa.

Sobre os questionamentos, a duração foi de duas horas, utilizando de quatorze questões abertas sobre o *framework* scrum, contando com um quadro de colaboradores, sendo, programadores, consultores e administradores qualificados na qual juntos em um time formulam soluções para a empresa de logística, desenvolvendo sistemas de gestão corporativo.

O uso do scrum na empresa foi adaptado de modo a suprimir as necessidades da empresa, projetos mensais contando com *sprints* no meio e no final do período dos projetos, existindo ainda, sim, a possibilidade de readaptação para situações específicas.

Desse modo, foi observado que não necessariamente o colaborador que entrega as tarefas mais rápidas é melhor que o colaborador que entrega as *tasks* com mais tempo.

Os gestores do projeto puderam analisar e compreender também que quando existem muitos requisitos de forma contínua no projeto, é promovido quebras a fim de ser dividido em tarefas menores, facilitando assim o planejamento e execução para a equipe de desenvolvimento.

Segundo a empresa estudada, quanto menor as tarefas a serem desenvolvidas, mais existirá motivação por parte dos funcionários de realizar entregas contínuas, concluindo, portanto, que projetos longos, com *tasks* extensas causam uma menor produtividade, afetando a qualidade do produto final.

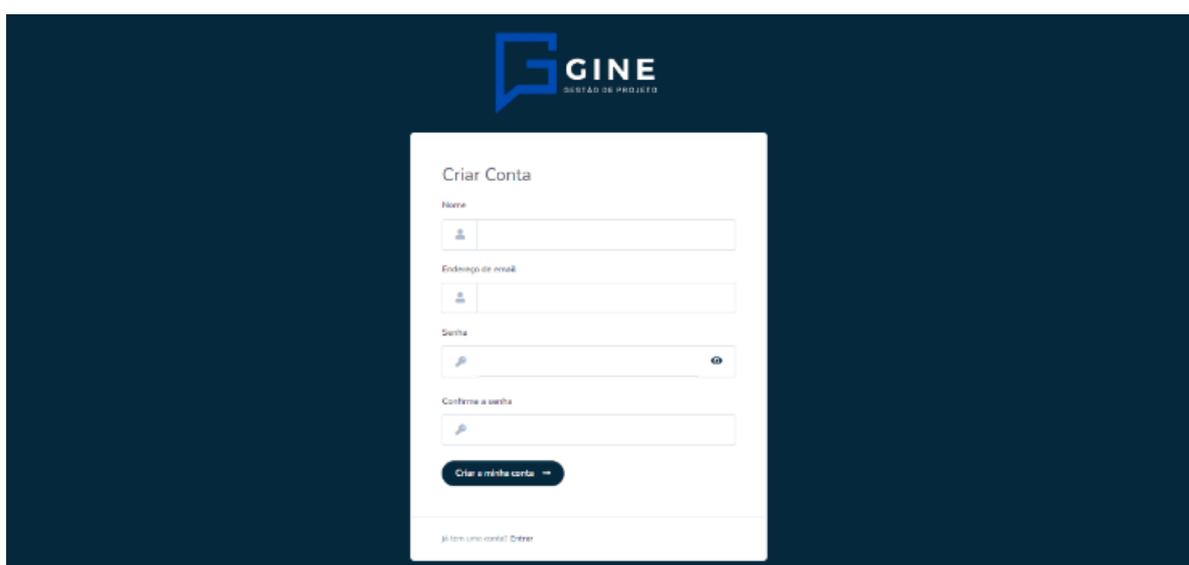
A conclusão da empresa alvo do estudo de (CARNEIRO et al., 2022) revela que o *framework* scrum contribui positivamente para construção de um produto onde o foco está ligado diretamente ao cliente, com a intenção de criar um produto que possa agregar valor ao consumidor final do software. Destaca-se que pode existir impasse na fase de criação e formulação do software, contudo é necessário manter sempre o detalhamento máximo das atividades e necessidades reais.

Uma maneira de contornar o impasse supracitado, a empresa cria uma Sprint nomeada de "Cerimônia de Homologação" no qual o líder do projeto analisa se o produto a ser entregue está compatível com o levantamento de requisitos, caso não esteja, é gasto um tempo maior para reajustar e alinhar as expectativas.

Como resultado, a empresa recorre ao scrum por ter consciência da qualidade que o *framework* proporciona tanto para criação do produto programável quanto para interação dos colaboradores a fim de promover o completo entendimento dos requisitos que serão desenvolvidos, uma vez que a metodologia aproxima o cliente do produto trabalhado.

Como impasse, está na falta de clareza e entendimento nas fases iniciais da aplicação do scrum, na prática, sendo necessário assim uma conversa direta com o cliente a fim de compreender a necessidade e implementar através da programação todos os requisitos, na prática.

Figura 10 – Tela de login desenvolvida no sprint 1.



Fonte: (MAIA, 2022)

No trabalho de (MAIA, 2022) fora desenvolvido uma ferramenta para fornecer suporte a equipe de criação de software na gestão de atividades, essencialmente, a ferramenta

foi desenvolvida utilizando uma adaptação do scrum para equipes unitárias, sendo deste modo uma abordagem diferente com relação às tradicionais, uma vez que o scrum é abordado em um time multidisciplinar com mais de um colaborador por projeto.

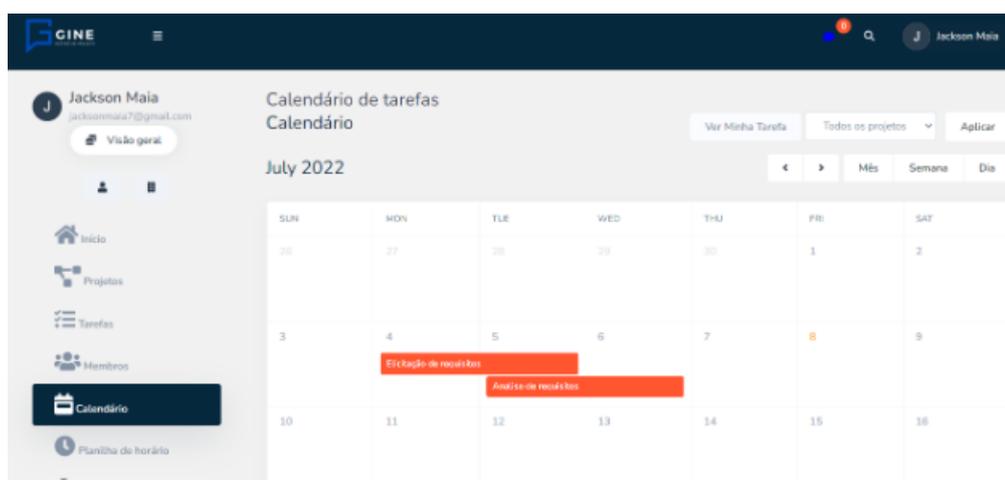
O autor supracitado, estudou ferramentas de planejamento e gestão para estruturação do trabalho, como o Trello, aplicações de chat como o Slack e entre outros. A partir de então, houve fundamentação o suficiente para modelar e criar uma solução adaptada ao "Scrum Solo" nomeado pelo autor, como uma alternativa ao uso do scrum sendo adaptado para eliminação de reuniões diárias adjunto a entrega de parte do projeto ao final do escopo estipulado como prazo.

Em contínuo, o autor dividiu o trabalho em *sprints*, contendo incrementalmente novas atividades a serem desenvolvidas, uma vez em que o protótipo e o levantamento de requisitos já foram realizados. Progressivamente, a primeira atividade consiste na mais básica, o cadastro de times desenvolvimento, sendo responsabilidade do líder do time, aumentando então as implementações a medida em que novas necessidades eram levantadas.

Na fase final do "Scrum Solo" o projeto foi apresentado ao cliente a fim de ser realizado validações e testes de funcionalidades da ferramenta criada, surgindo necessidades de simular testes de cada sprint para comparar se os resultados estavam compatíveis com o produto durante as fases de *sprints*. Como foi criado um planejamento dos requisitos válidos a serem implementados, se tornou facilitada a verificação de como o trabalho progrediu.

A *task* inacabada no projeto foi justamente a que lidava com validações de cronômetro por cada tarefa, além de passar do prazo estipulado de trabalho da *task*, fora relatado um nível de complexidade acentuado, uma das explicações do autor, consiste no fato de cada equipe poder atuar em fuso-horários e horários diferentes.

Figura 11 – Calendário de atividades desenvolvido no trabalho.



Fonte: (MAIA, 2022)

O projeto "Gine" criado por (MAIA, 2022) usou variadas funcionalidades, desde gerenciamento de projeto com criações de projetos, relatórios, até mesmo seções de gerenciamento de equipe, permitindo criação de perfis e também seções de gerenciamento de tarefas. Permitindo a gerência de um trabalho em equipe por meio de um escopo bem definido de funções e requisitos necessários à implementação.

Como feedback do citado autor, o "Scrum Solo" se mostrou uma abordagem eficiente, justamente por ser dinâmica, contando com reduções de burocracias, contudo fora criado

ressalvas, como, por exemplo, a inexistência de padronização para lidar com as atividades das *sprints* semanais, existiu uma divisão de tarefas e do que precisaria ser implementado, contudo, não se apresentou uma separação clara e objetiva de como lidar com eventuais desafios, uma vez em que mais de uma *sprint* exigiu mais trabalho do desenvolvedor, ocasionando mais horas de trabalho do que o planejado.

Como alternativa à solução do imbróglio supracitado, o criador do "Gine" optou por separar as execuções das tarefas dos sprints por grau de dificuldade, em cada nova funcionalidade fora utilizado testes unitários e o final de cada *sprint* foi o momento de tratar eventuais bugs ou eventos inesperados das *sprints* inacabadas.

Outro impasse identificado pelo autor, foi devido à escassez de modelos prontos para facilitar o processo de criação do software, a base do trabalho de metodologias ágeis está em conseguir entregar um produto dentro do prazo estipulado, deste modo, *templates* prontos, fáceis de alteração facilitam o processo de criação e reduz o tempo de customização, uma vez em que tarefas de frontend podem ser complexas caso exista muito detalhamento e diversas seções a serem construídas. O autor contornou o problema buscando um *template* que fornecia implementações pré-prontas, a fim de otimizar o processo de programação, bem como possuir maior velocidade na construção da aplicação e incrementar novas funcionalidades para melhorar a fluidez e organização do software.

Como o projeto foi desenvolvido na fase da pandemia da Covid-19, o autor referencia a ferramenta desenvolvida como uma abordagem e utilização para projetos que precisam ser desenvolvidos a distância, centralizando todas as informações e trabalhos pertinentes a um projeto de software em uma única ferramenta, evidenciando, portanto, a eficiência e utilidade de se utilizar metodologias ágeis para criação de um produto com necessidade real no mercado de trabalho.

4 MATERIAIS E MÉTODOS

Nesta seção serão detalhadas todas as ferramentas e metodologias que foram utilizadas para desenvolvimento da proposta, como: HTML, CSS, Figma, JavaScript, Vue.js, Quasar, NPM e entre outros.

4.1 Frontend

O frontend é a camada da aplicação responsável por lidar e estar diretamente disponível ao cliente, em uma aplicação web, o frontend será interpretado por um *browser* que permitirá interação do usuário com o backend (VALERO, 2019). Todo o desenvolvimento da *UI/UX* da aplicação, se baseando, portanto, em componentes modernos e reutilizáveis, sendo a principal seção da aplicação no qual o usuário poderá inferir e navegar, logo é necessário desenvolver um produto intuitivo e de fácil uso para ser entregue uma experiência satisfatória ao usuário final. Uma característica da área é simular ações do usuário a fim de que seja construído e previsto todas as ações úteis ao cliente final, bloqueando, portanto, ações desnecessárias, bem como confusas.

4.2 HyperText Markup Language (HTML)

HTML é uma linguagem de marcação, diferenciando das linguagens de programação, pois é possível criar seções e estruturas para conteúdos da web, uma maneira de padronizar e organizar as informações dentro de uma página web. Se tornando uma série de elementos que agrupam um conteúdo, através das *tags* (LEITZKE; AMARAL, 2022). Não é considerado uma linguagem de programação porque não é possível realizar lógica dentro do próprio recurso HTML, todavia, permite a criação de estruturas da página que posteriormente sofrerão alterações ou inserções dentro dos elementos que o HTML demarca. Deste modo, é possível criar o esqueleto de aplicações web com padronizações necessárias da web para funcionamento em diferentes navegadores e dispositivos.

4.3 Cascading Style Sheets(CSS)

O CSS é a linguagem que define o padrão dos documentos que estarão visíveis ao usuário em uma página web (RABETTI, 2019). Assim como HTML não é considerado uma linguagem de programação, contudo é uma linguagem de estilos que permite a customização da linguagem de marcação HTML, com CSS é possível fornecer estilo à página web, permitindo alteração de características, tais como cores, fontes e sobretudo adaptabilidade o display do dispositivo com a aplicação desenvolvida.

Com o CSS a página web estática pode adaptar à diferentes dispositivos, trazendo uma experiência do usuário mais agradável, facilitando a usabilidade do software. Inferindo diretamente no conteúdo e demarcação de tags HTML, possibilitando até mesmo a criação de animações nas páginas web.

4.4 Figma

O Figma é uma ferramenta na qual é possível realizar prototipações de design de telas para diferentes dispositivos e tamanhos dos mesmos, sendo uma ferramenta de

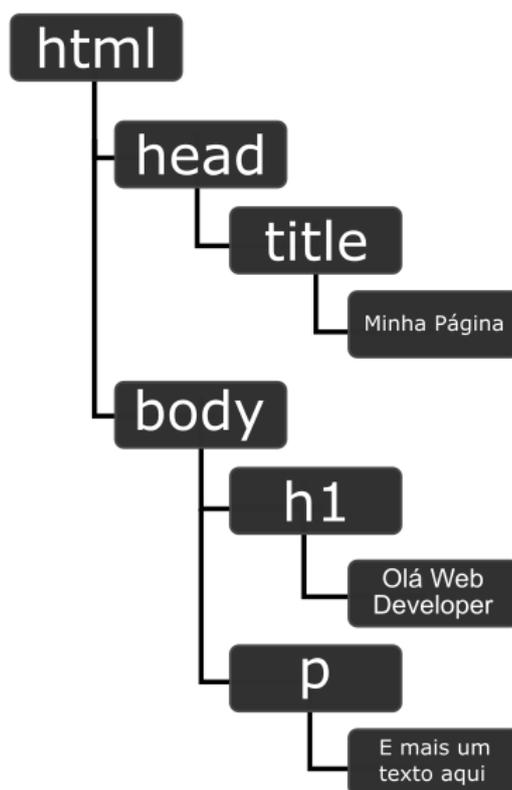
design, esta permite com que todos os, micro *componentes de UI* sendo as iniciativas do CSS e HTML sejam definidas de forma clara e objetiva. Servindo portanto para a criação do *MVP* visual da aplicação, no qual o cliente precisa validar e enviar feedbacks caso for necessário o reajuste. É uma ferramenta de design que permite o uso colaborativo e incremental em projetos em desenvolvimento (OLIVEIRA, 2022).

Toda a parte de frontend da aplicação será primeiramente prototipada no figma para posteriormente ser programada, esse método acelera bastante a produtividade na programação, uma vez em que o escopo está definido e foi implementado os levantamentos exatos, é desenvolvido uma aplicação centrada no escopo e com base nos feedbacks recebidos pelo líder do projeto.

4.5 Document Object Model(DOM)

A figura 12 representa a estrutura da DOM e alguns de seus elementos.

Figura 12 – Estrutura em árvores de nós na DOM.



Fonte: (TREINAWEB, 2023)

DOM consiste em uma representação estruturada de um documento e estrutura HTML na web, sendo uma representação é possível, portanto, modificar a estrutura de uma página através do uso de JavaScript, através de eventos aceitos pelo navegador, o programador consegue tornar uma página web interativa, com mudanças exibidas em tempo real a medida em que são usados estados. Usando eventos, o DOM cria uma árvore de objetos correspondentes a elementos HTML que permite operações assíncronas na web. Através da representação da estrutura HTML em objetos, o JavaScript consegue alterar e monitorar as mudanças na tela do navegador (SANTOS, 2022).

4.6 JavaScript

JavaScript é uma linguagem de programação criada para executar *scripts* em navegadores, atualmente sendo comumente utilizada de forma assíncrona (COSTA, 2022). Uma linguagem interpretada e estruturada em alto nível, baseada no *ECMAScript* sendo uma linguagem de *script* utilizada para construção de páginas web modernas e com o ambiente integrado aos *browsers* modernos, suportando, portanto, diferentes estilos de codificação, tais como programação funcional e programação orientada a objetos. Sua popularidade consiste em trazer aos desenvolvedores recursos prontos e nativos mediante funções aos quais modificam a *DOM* do *browser* de uma maneira modular e escalável.

Os navegadores modernos são em parte construídos em JavaScript, facilitando, portanto, a integração com recursos assíncronos, causando alterações na tela para o usuário sem que este fique necessariamente com recursos bloqueados para uso. Através do JavaScript surgem diferentes bibliotecas e *frameworks* a fim de formular boas práticas de programação tanto para a área de backend quanto para frontend.

4.7 Vue.js

O Vue.js é um *framework* JavaScript utilizado para construção de interfaces progressivas e modernas, sendo utilizado em toda a parte do frontend da aplicação, este *framework* recorre a melhores práticas estruturais para que o código final que será lido pelo navegador fique o mais conciso possível. Utilizando de métodos de manipulação da *DOM* do navegador, o Vue.js traz todo um arsenal funcional para o suporte de aplicações escaláveis e modernas, sem deixar de lado a segurança da aplicação. A aplicação consiste em componentes, se baseando em HTML, CSS e JavaScript em pequenas partes do desenvolvimento da interface. Com o *framework* Vue.js os componentes podem ser compartilhados de forma facilitada, sem a necessidade de ser criadas coleções complexas para comunicação dos dados (LIMA; PETRUCCELLI; SANTO, 2019).

4.8 Quasar

O Quasar consiste em um *framework* de Vue.js para criação de aplicações responsivas e que possua um *design* simples base para criação de componentes em Vue.js, a base do quasar está voltada para construção de *PWAs*, todavia também existe a possibilidade de ser utilizado todo o seu *superset* para construção de aplicações web também. A estrutura com Vue.js permite a criação de sites responsivos tanto renderizados ao lado do servidor quanto aplicações de página única (TRINDADE, 2020). Deste modo, é facilitando tanto a parte de interface do usuário, quanto utilização, suporte para diferentes navegadores e até mesmo uma fácil integração para que a aplicação tenha o comportamento de uma *PWA* ou recorra a alguma aplicação *SSR*. Dentro do contexto da aplicação desenvolvida, fora tratado o quasar como um integrador para construção de uma aplicação *SPA*, possuindo como base uma aplicação de página única, isto é, possuindo comportamento integrador de todo o conteúdo e funcionalidades em um único bloco visual.

Com o Quasar é desenvolvido uma aplicação completa de modo em que esta poderá ser reaproveitada pequenos componentes úteis entre si, tais como botões, *headers*, *footers*, ícones de tela que estarão presentes em todo o sistema, sendo deste modo um facilitador tanto para construção quanto para manutenções no futuro.

Caso seja necessário transformar a aplicação web para uma versão mobile, será um passo facilitado, uma vez em que todos os componentes utilizados, ícones, fonte de

texto do projeto estão em Quasar.

4.9 PWA

Progressive Web App (*PWA*) é uma aplicação que possui características semelhantes a um aplicativo nativo dos dispositivos móveis, incluindo algumas operações *offline* como o uso de notificações. O avanço da Progressive Web App se deve ao crescimento do desenvolvimento web, tornando recursos compartilhados acessíveis às plataformas mobile (TRINDADE, 2020).

A base é construída a partir dos conhecimentos de HTML, CSS e JavaScript, assim, é uma aplicação web que pode se comportar como um aplicativo mobile. Com relação ao uso, a *PWA* é instalada no ambiente do dispositivo móvel sem distinção de forma imediata da diferença para um aplicativo nativo.

4.10 Mobile First

Mobile First é uma abordagem na qual o desenvolvimento inicia primeiramente para telas menores, indo para *displays* maiores, se adaptando a tela do dispositivo usado na aplicação (PIETRANTONIO, 2021). Quando a aplicação é projetada para ser *Mobile First*, aumenta a adequação para diferentes tipos de dispositivos, permitindo uma usabilidade agradável ao usuário final, facilitando até mesmo no entendimento da aplicação, uma vez em que tanto imagens quanto textos são adaptados ao tamanho satisfatório para experiência do usuário.

4.11 Componentes

Componentes em desenvolvimento de software, são blocos de códigos escritos que podem ser utilizados uma ou mais vezes em uma aplicação, facilitando tanto a escrita quanto a manutenção de projetos, serve também para separar a estrutura da aplicação, aumentando deste modo a organização do software, definindo prioridades. Além disso, *componentes* são recursos de diferentes arquiteturas, o conjunto de componentes em diretórios de pastas na aplicação pode formular uma arquitetura de software. Uma vez criado *componentes*, estes poderão ser compartilhados em todo escopo do projeto, tanto para inserção de dados quanto para alteração.

Através da reusabilidade dos *componentes*, os desenvolvedores conseguem criar um produto mais seguro, seguindo padrões de usabilidade e qualidade na aplicação (SANTIAGO et al., 2020).

4.12 Node.js

Node.js é definido como um ambiente de execução JavaScript surgindo em 2009, possibilitando, portanto, a criação de aplicações destinadas para o lado do servidor, isso é, um ambiente que irá funcionar todos os métodos e protocolos de uma aplicação backend estruturada. A principal vantagem de trabalhar com Node.js está ligado a forma com que é possível programar de forma funcional, além de trabalhar com recursos de forma assíncrona, sem que existam recursos bloqueantes no percurso do backend da aplicação. Além disso, eficiência do Node, está ligado a sua forma de lidar com eventos.

A partir do Node é possível criar *APIs Rest* e *Rest Full* de formas escaláveis, sendo o padrão *Real-Time* e modelar arquiteturas modernas utilizando como base o mesmo

conhecimento de JavaScript adquirido no frontend de aplicação web. O Node oferece APIs extra que podem manipular entrada e saída, tais como *Sockets*, sistemas de arquivos e entre outros (PISSURNO, 2022).

A sua utilidade será necessária para executar a aplicação em vue.js, forçando um servidor local em uma porta da máquina de teste para simular como a aplicação iria sobressair em um ambiente de produção real. O node executa a aplicação através do pacote *NPM* e então sobe automaticamente uma *URL* para acesso do projeto, através do node e *NPM* também é possível entender o tempo que a aplicação frontend levará para responder quando estiver em ambiente de produção.

4.13 Node Package Manager(NPM)

NPM é um gerenciador de pacotes para aplicações JavaScript, este fornece todos os pacotes necessários para que um projeto importe dependências necessárias, sendo útil tanto para inclusão de bibliotecas e ferramentas para o projeto quanto para manutenções, pois é o gerenciador capaz de executar a instância de projetos locais ou remotos. A ideia do *NPM* é facilitar o desenvolvimento, unindo ferramentas à programação da aplicação, trazendo pacotes necessários para aplicação (FILHO, 2022).

4.14 Visual Studio Code

O Visual Studio Code(VS Code) é um editor de texto desenvolvido para Microsoft, Linux e MacOs, suportando diversas linguagens de programação, bem como *plugins* específicos de cada uma (FILHO, 2022). Contando com a possibilidade de instalação de extensões e utensílios para tornar a experiência de programação mais dinâmica e organizada. Contando com *plugins* do Vue.js e do *autoimport* e *autoclose tags*, além de permitir trabalhar com *eslint* habilitado ao projeto, facilitando a compreensão e edentação do código.

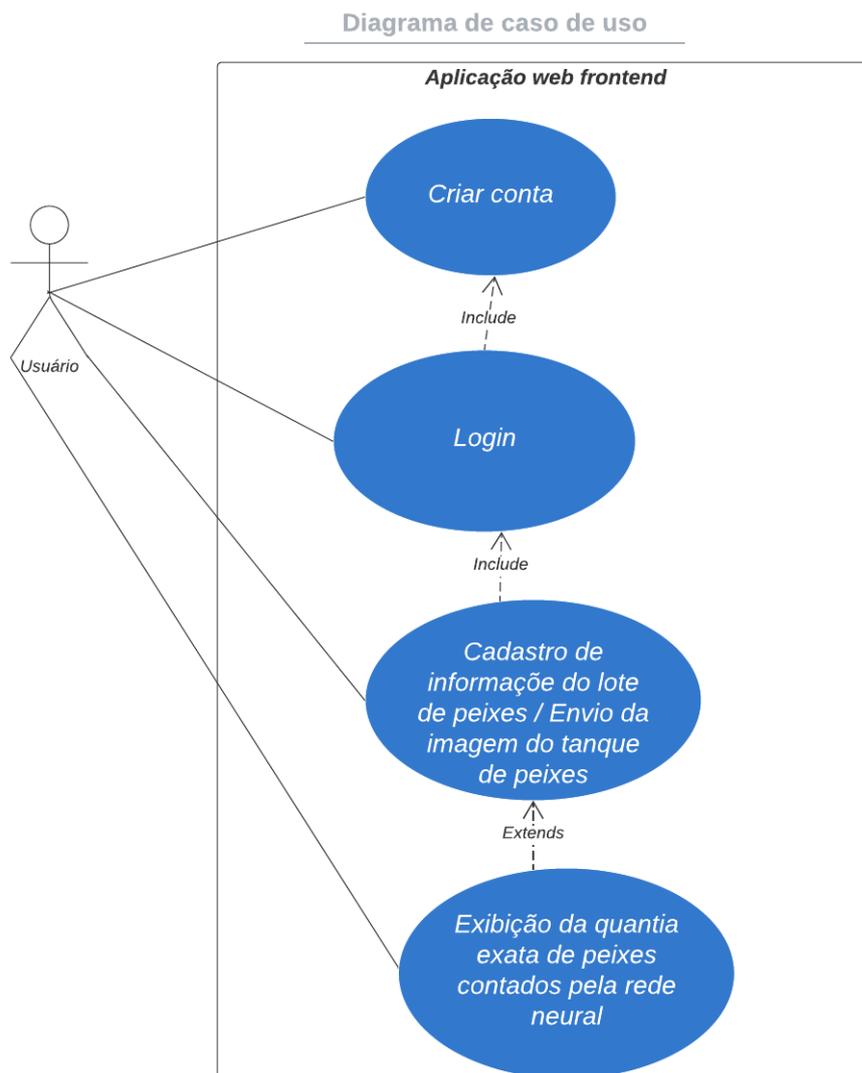
5 DESENVOLVIMENTO WEB

Nesta seção, será apresentada a abordagem para solucionar o problema existente na empresa OnPeixes, que envolve o desenvolvimento de uma aplicação web frontend integrada à rede neural interna. A rede neural já está em funcionamento e foi criada por colaboradores da empresa. A aplicação frontend estabelecerá conexão com essa rede neural, enviando uma imagem do tanque de peixes para obter a contagem precisa dos peixes e exibi-la na própria aplicação. Essa contagem requer o envio da imagem do tanque de peixes à rede neural, possibilitando o retorno do número exato de peixes presentes no espaço. Além disso, serão abordados os métodos utilizados na construção da aplicação web, com potencial para se tornar um produto completo. Assim, a aplicação web se conectará a uma inteligência artificial especializada em contagem de peixes, utilizando as imagens fornecidas pela própria aplicação web como base.

Após o período de definição de tecnologias, foi realizada a análise de requisitos, primordialmente a aplicação precisa ter um local de envio de imagens, para que as imagens sejam enviadas da aplicação frontend até a rede neural que receberá essa imagem. Contudo, para a aplicação ser mais completa, foi necessário compreender os aspectos pertinentes ao trabalho do piscicultor, precisando incluir, portanto, o lote de peixes, sexo, espécie, e alguma informação adicional que o usuário preferir inserir para que esta informação fique salva no perfil da conta do usuário que estará utilizando aplicação.

Conforme a figura 13, fora elaborado o caso de uso a respeito das necessidades do cliente, bem como quais recursos são necessários para compor a aplicação web frontend, de modo que faça sentido para incrementar e integrar futuramente à rede neural, conforme citado na seção de introdução. O caso de uso é utilizado a fim de se obter o conhecimento do acesso por parte do utilizador da aplicação, bem como compreender quais recursos são requisitos funcionais para seções do frontend.

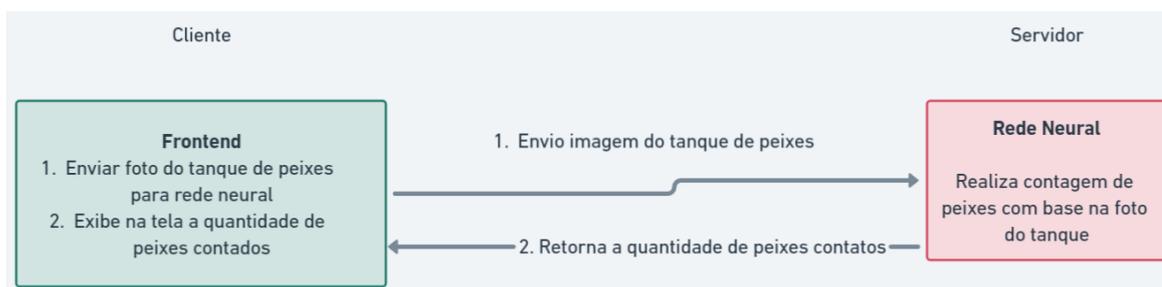
Figura 13 – Caso de uso do projeto.



Fonte: Próprio autor.

Conforme a figura 14, o comportamento da aplicação segue este padrão, o frontend da aplicação enviando uma imagem de um tanque de peixes e a rede neural será encarregada por realizar a contagem com base na foto subetida anteriormente, após essa etapa, será devolvido pela rede neural até o frontend da aplicação apenas a contagem de peixes contados, sendo portanto um número inteiro.

Figura 14 – Demonstração de uso da aplicação.



Fonte: Próprio autor.

Após os requisitos necessários compreendidos, como forma de estruturação do trabalho e desenvolvimento, fora utilizado o *framework* Scrum para planejamento e gestão das tarefas a serem desenvolvidas durante o curto período de tempo de três *sprints*, cada uma contando com duas semanas de execução. Como o foco foi desenvolver, cada semana o software foi incrementado, tanto em programação quanto de adaptação do código já desenvolvido.

A decisão de adotar o scrum baseia-se na sua capacidade de proporcionar uma organização estruturada do trabalho, permitindo incrementar a construção do software e receber feedback para melhorar o produto. Essa abordagem permite estar mais alinhado com as necessidades da empresa. Por meio das sprints, o scrum fragmenta o trabalho, o que possibilita uma compreensão abrangente dos requisitos que a aplicação precisa atender.

A proposta consiste em uma aplicação web, *mobile-first* na qual é construída aplicação do mobile para o desktop, usando o scrum como motivação e gerenciamento do projeto. A aplicação irá ser utilizada para que o usuário possa realizar operações, tais como criação de conta, login e cadastramento de informações com envio de foto, para ser enviada de forma padronizada de uma maneira que a rede neural trate das informações recebidas e retorne ao usuário a contagem de peixes.

A aplicação será um frontend web criado em Vue.js e Quasar e acessado exclusivamente por meio de um navegador com acesso à internet. Através dessa aplicação, o usuário terá a possibilidade de enviar uma foto para a rede neural, que retornará o número exato de peixes contidos no tanque, exibindo-o na tela ao usuário.

Desse modo, a ferramenta de Inteligência artificial realiza contagens de peixes, uma vez que é submetido uma imagem de um tanque de peixes, contudo funcionando apenas em ambiente de testes dentro do Google Collab, carecendo, portanto, de novas ferramentas que faça a contagem de peixes se tornar um produto da programação e aprendizado de máquina.

Inicialmente foi realizado o estudo a respeito de metodologias ágeis e do uso de scrum em projetos de software. Em seguinte foi definido a escolha das tecnologias úteis para criação da aplicação web, por fim, foi definido os prazos de trabalho para conclusão do projeto, separando-o assim, em *sprints* semanais.

Cada *sprint* possuía responsabilidades individuais, existindo, contudo, a possibilidade de alterações, a fim de que o produto desenvolvido esteja validado e conforme as necessidades reais do cliente final. Inicialmente foi realizado o levantamento de quais itens a aplicação web precisaria conter, identificado, portanto, seções de criação de conta, login, cadastramento de informações para ser enviado a rede neural, estes foram os pontos mais urgentes e essenciais levantados.

As *sprints* contaram com periódicas reuniões diárias a fim de ser compreendido os requisitos do projeto e delimitar quais os próximos passos do trabalho em que estiver sendo executado. A maioria das *sprints* foi remotamente, com a intenção de ser construído um produto que satisfaça as necessidades do projeto, inferindo, assim, pontuais alterações e alinhamentos do que fora desenvolvido e planejado. O escopo de trabalho foi coordenado pela responsável do projeto, organizando os prazos de entrega e acompanhando de perto as alterações a cada ciclo de desenvolvimento iterativo. Após a prototipação inicial, foram definidos o uso da *stack* de tecnologias para programação do software, definindo, desta maneira, prazos tanto de período de programação quanto de estudo de ferramentas necessárias a execução do projeto.

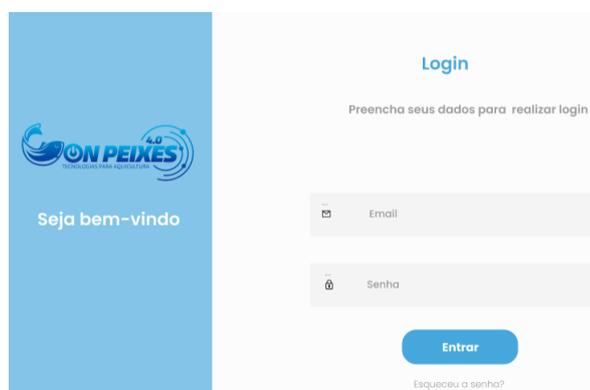
A primeira *sprint* foi destinada a estudos do ecossistema Vue.js e JavaScript para criação de aplicações responsivas na web.

Sprint 1:

Levantamento a respeito das tecnologias utilizadas a fim de promover um software com ferramentas atuais e que pudesse contar com futuras manutenções, sendo necessário, portanto, uso de ferramentas recentes e com amplo suporte para facilitar a construção da aplicação. Foi definido, um estudo do *framework* Vue.js, uso de *componentização*, entendimento de renderizações através de estados, estudo de *props* e das mudanças do ecossistema ES-6 no Javascript para programação funcional. Adjunto a protótipos de design na ferramenta Figma para melhor compreensão do produto desenvolvido, sendo utilizados apenas como motivação ao desenvolvimento.

Conforme a figura 15 e figura 16 foram realizados protótipos de testes para entendimento de quais campos de *input* e de informações a aplicação precisará.

Figura 15 – Tela de login no figma.



Fonte: Próprio autor

Figura 16 – Tela do criação de conta no figma.



Fonte: Próprio autor

Sprint 2:

Início da configuração do repositório para desenvolvimento, com ferramentas de dependência e plugins para funcionamento de utilitários. O *setup* dos primeiros componentes da página web foi iniciado com Vue.js e o Quasar como fonte de *UI* para aplicação, usando, portanto, elementos prontos de dentro da própria ferramenta, tais como botões, *footer*, *inputs* e *cards*. O Quasar permitiu também a responsividade da aplicação, sendo adaptado tanto ao display de celulares pequenos, quanto de monitores maiores para facilitar adaptabilidade aos diversos dispositivos.

Seguindo a documentação oficial do Vue.js (YOU, 2023) é possível iniciar o desenvolvimento sendo orientado a depender dos recursos necessários ao desenvolvimento,

para isso, foi necessário instalar o Vue.js de forma global na máquina e instalar também o Node.js, juntamente com *NPM*, a fim de ser utilizado o gerenciador de pacotes do *NPM* para criação do projeto.

Com *NPM*, foi criado o projeto com a estrutura básica somente do Vue, sem nenhum gerenciador de rotas ou estrutura de UI. Em seguida, foi necessário usar o gerenciador do *NPM* para adicionar o *Vue-Router* (YOU, 2022), sendo o roteador oficial do *framework* progressivo. Após o projeto semi-configurado, faltava somente adicionar a fonte de UI, ícones e fonte de texto do Quasar (SRL, 2022), bem como *plugins* necessários para integração com o *framework*.

Conforme figura 17 fora criado o projeto em Vue.js após instalado de forma global na máquina de testes.

Figura 17 – Criação do projeto.

```
lucas@lucas ~ > vue create project 5929 09:43:16
Vue CLI v5.0.6
New version available 5.0.6 → 5.0.8
Run npm i -g @vue/cli to update!

? Please pick a preset: Default ([Vue 3] babel, eslint)

Vue CLI v5.0.6
🌟 Creating project in /home/lucas/project.
📁 Initializing git repository...
⚙️ Installing CLI plugins. This might take a while...

> yorkie@2.0.0 install /home/lucas/project/node_modules/yorkie
> node bin/install.js

setting up Git hooks
done

> core-js@3.27.2 postinstall /home/lucas/project/node_modules/core-js
> node -e "try{require('./postinstall')}catch(e){}"

added 854 packages from 465 contributors and audited 855 packages in 31.365s

91 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

🚀 Invoking generators...
📦 Installing additional dependencies...

added 101 packages from 63 contributors and audited 956 packages in 9.786s

102 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

🔗 Running completion hooks...
📄 Generating README.md...

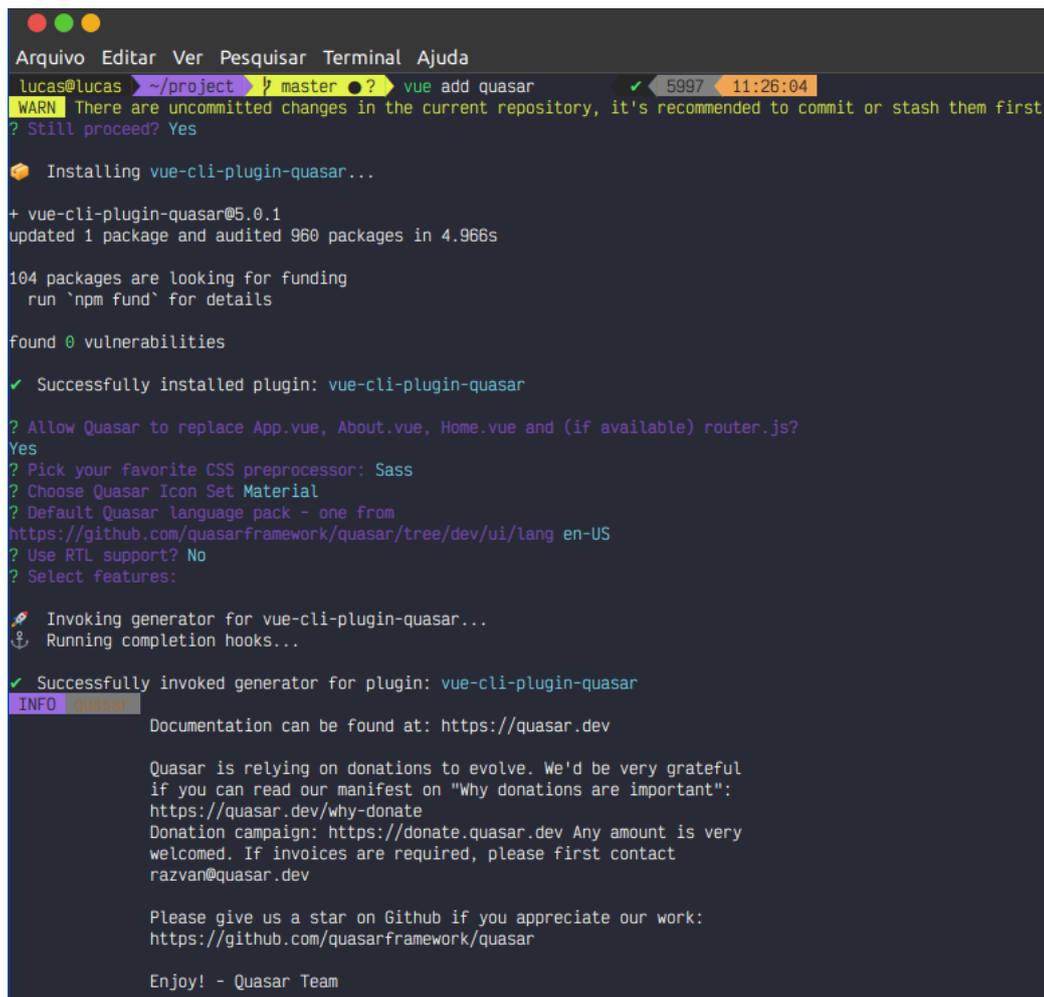
🎉 Successfully created project project.
👉 Get started with the following commands:

$ cd project
$ npm run serve
```

Fonte: Próprio autor.

Instalando o Quasar, figura 18 no diretório do projeto para utilizar todos os recursos de *UI* e ícones para a aplicação.

Figura 18 – Instalando Quasar.



```
Arquivo Editar Ver Pesquisar Terminal Ajuda
lucas@lucas > ~/project > ? master ● ? vue add quasar ✓ 5997 11:26:04
WARN There are uncommitted changes in the current repository, it's recommended to commit or stash them first.
? Still proceed? Yes

📦 Installing vue-cli-plugin-quasar...

+ vue-cli-plugin-quasar@5.0.1
updated 1 package and audited 960 packages in 4.966s

104 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

✓ Successfully installed plugin: vue-cli-plugin-quasar

? Allow Quasar to replace App.vue, About.vue, Home.vue and (if available) router.js?
Yes
? Pick your favorite CSS preprocessor: Sass
? Choose Quasar Icon Set Material
? Default Quasar language pack - one from
https://github.com/quasarframework/quasar/tree/dev/ui/lang en-US
? Use RTL support? No
? Select features:

🔧 Invoking generator for vue-cli-plugin-quasar...
📌 Running completion hooks...

✓ Successfully invoked generator for plugin: vue-cli-plugin-quasar
INFO Quasar
Documentation can be found at: https://quasar.dev

Quasar is relying on donations to evolve. We'd be very grateful
if you can read our manifest on "Why donations are important":
https://quasar.dev/why-donate
Donation campaign: https://donate.quasar.dev Any amount is very
welcomed. If invoices are required, please first contact
razvan@quasar.dev

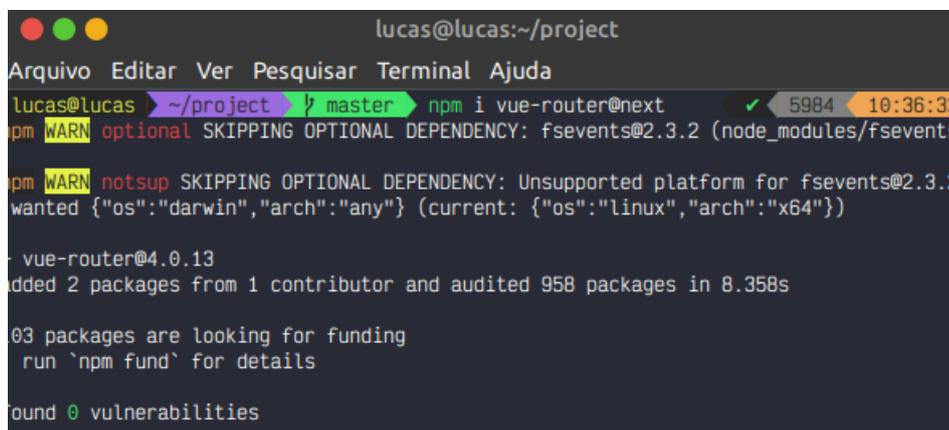
Please give us a star on Github if you appreciate our work:
https://github.com/quasarframework/quasar

Enjoy! - Quasar Team
```

Fonte: Próprio autor.

Em seguida é preciso instalar o *VueRouter* figura 19 para gerenciar toda a seção de rotas que a aplicação utilizará.

Figura 19 – Instalando VueRouter.



```
lucas@lucas:~/project
Arquivo Editar Ver Pesquisar Terminal Ajuda
lucas@lucas > ~/project > ? master > npm i vue-router@next ✓ 5984 10:36:32
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules/fsevents)
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2
wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

- vue-router@4.0.13
added 2 packages from 1 contributor and audited 958 packages in 8.358s

03 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Fonte: Próprio autor.

Assim, desenvolvido, portanto, as telas de criação de conta e login, nessa *sprint* foi necessário validar o *template* anteriormente projetado no figma, pois a cor do *layout*

dividida ao meio não funciona para facilitar a experiência do usuário no mobile, conforme exemplificado nas figuras 15 e figura 16. Nas aplicações *mobile first*, os elementos precisam ser centralizados a partir do topo da tela até o final, facilitando deste modo a divisão dos elementos que precisarão ser exibidos na tela.

Sprint 3:

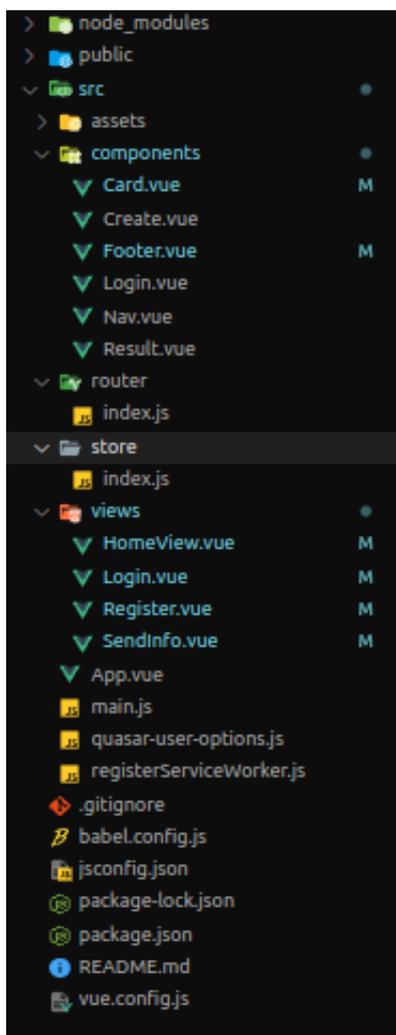
Nessa Sprint foi necessário realizar os levantamentos a respeito de quais dados a aplicação precisaria receber para ser enviada à rede neural para um funcionamento adequado. A partir de então foi desenvolvido os campos de *input* dentro de um *componente* apenas de envio de informações, com os campos de envio de formulário para que posteriormente possa ser recebido pela rede neural. Além dos campos de *input*, foi necessário na *sprint*, criar um campo de utilização de câmera, testado na aplicação como um elemento permitido pela própria *DOM* do navegador, sendo um evento de câmera do dispositivo que estará sendo acessado pela web.

No período da terceira sprint, foi realizado levantamento das prioridades a serem programadas com feedback de que os campos de *input* precisariam conter campos de lote, espécie do peixe, sexo do peixe, idade em dias, alguma informação ou descrição que o usuário quiser adicionar e por fim o campo para tirar a foto. A aplicação também tinha que contar com um botão de *upload* para caso a foto já estivesse sido tirada, só bastava, portanto, fazer o *submit* da mesma, escolhendo na própria galeria do celular.

A ideia por trás dos elementos preenchidos nos inputs e o acionamento da câmera, é para quem for utilizar a aplicação fornecer os dados parametrizados para a rede neural realizar a contagem de peixes, assim haverá um cadastramento na aplicação de forma assertiva. Assim, houve o fechamento do escopo da aplicação, validando o que foi desenvolvido e adaptando o que precisaria ser mudado para ser realizado a entrega do frontend dentro do prazo estipulado nas *sprints*.

A figura 20 evidencia a estrutura de arquivos utilizados no desenvolvimento do projeto.

Figura 20 – Arquivos criados durante o projeto.



Fonte: Próprio autor.

5.1 Resultados

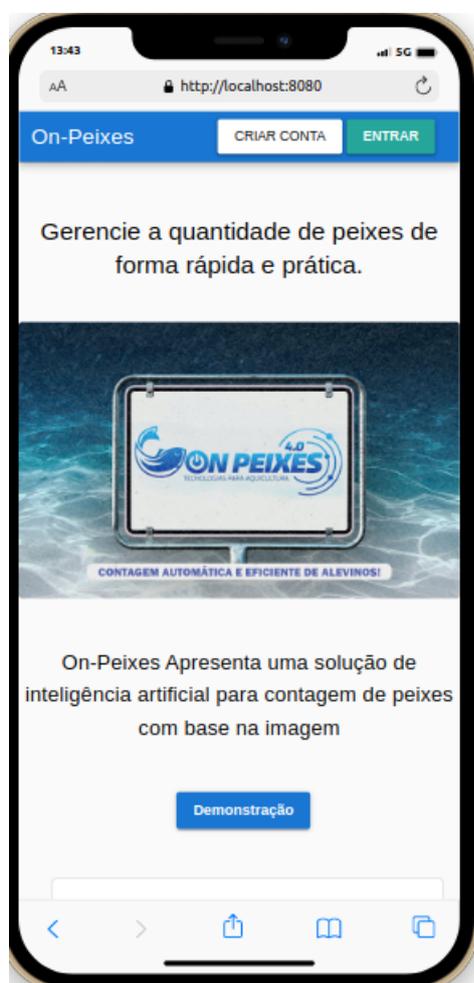
Com base na necessidade da empresa OnPeixes de ter uma aplicação web que se conecte ao produto interno de sua rede neural, foi proposta a implementação de um frontend web utilizando recursos modernos de desenvolvimento frontend. Essa ferramenta terá a funcionalidade de enviar uma imagem de um tanque de peixes para a rede neural, além de requisitos úteis sobre os peixes contidos no tanque. Após a conexão entre o frontend e a rede neural, as informações relacionadas à contagem de peixes estarão disponíveis na aplicação frontend, facilitando a visualização dos resultados fornecidos pela rede neural, para o usuário final visualizar no celular através do site.

Essa seção apresenta implementações e o desenvolvimento proposto nos capítulos iniciais do trabalho. Utilizando metodologia scrum foi possível realizar o planejamento e separar os estudos da prática de programação, conforme supracitado, fora criado uma aplicação web para servir futuramente de base para reformulação de um produto, integrado com uma rede neural que realiza contagem de peixes utilizando aprendizado de máquina.

Logo após o projeto configurado, foi inicializado a programação dos *componentes* da aplicação, sendo inicialmente formulada e desenvolvida a página de início do usuário, em ordem de prioridade, assim sendo, a primeira nas figuras 21, figura 22 e figura 23. Como

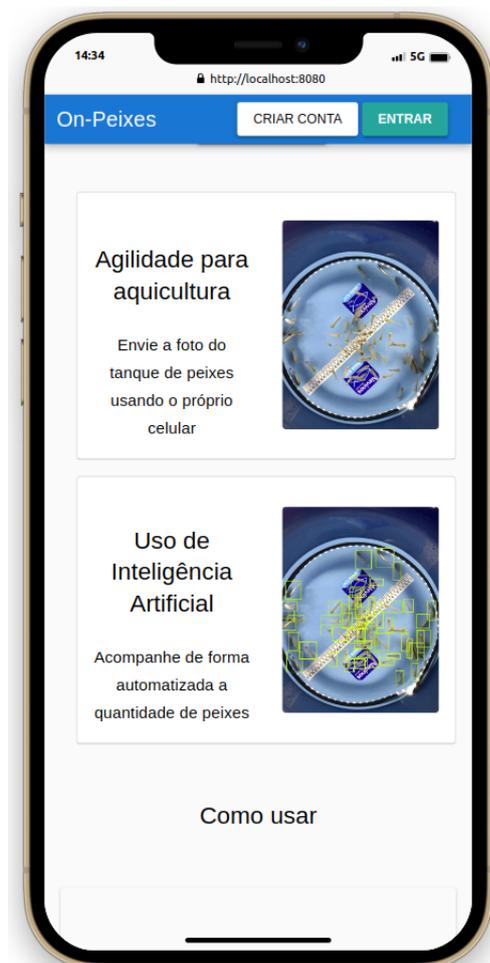
a página é uma *landing page*, sendo uma seção de propriedade única, é a primeira página a ser demonstrada ao usuário a respeito da aplicação, representa a primeira impressão do usuário em relação à aplicação, mesmo antes de criar uma conta para utilizá-la.

Figura 21 – Landing Page mobile parte 1.



Fonte: Próprio autor.

Figura 22 – Landing Page parte 2.



Fonte: Próprio autor.

Figura 23 – Landing Page parte 3.



Fonte: Próprio autor.

5.2 Telas desenvolvidas com Vue.js e Quasar

Após ambiente configurado e preparado para desenvolvimento, o projeto passou por *sprints* de execução e ajustes no *layout*, focando na experiência do usuário, a primeira tela desenvolvida foi a parte de *landing page* de apresentação do software, contando com seções de demonstrações e explicação a respeito do produto, assim, sendo criada somente a seção de página de única de apresentação, a *landing page*.

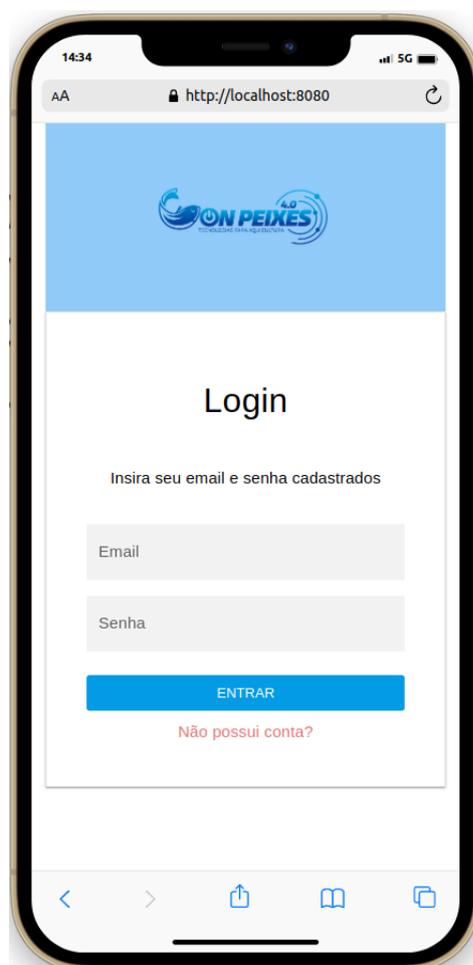
A segunda etapa de desenvolvimento foi conduzida na entrada do usuário à aplicação web, sendo programado, portanto, as telas de Login conforme demonstrado nas figuras 25, 24 a tela de login e Registro, bem como suas *rotas* particulares.

Figura 24 – Tela de criação de conta.



Fonte: Próprio autor.

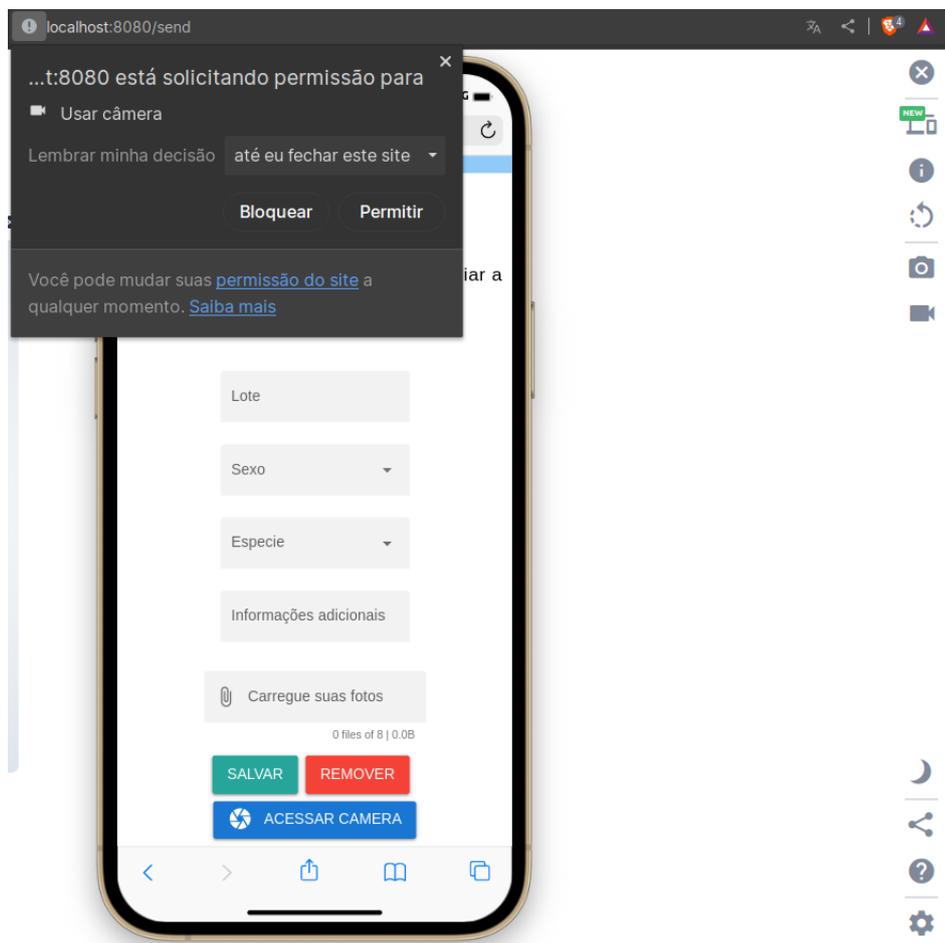
Figura 25 – Tela de login.



Fonte: Próprio autor.

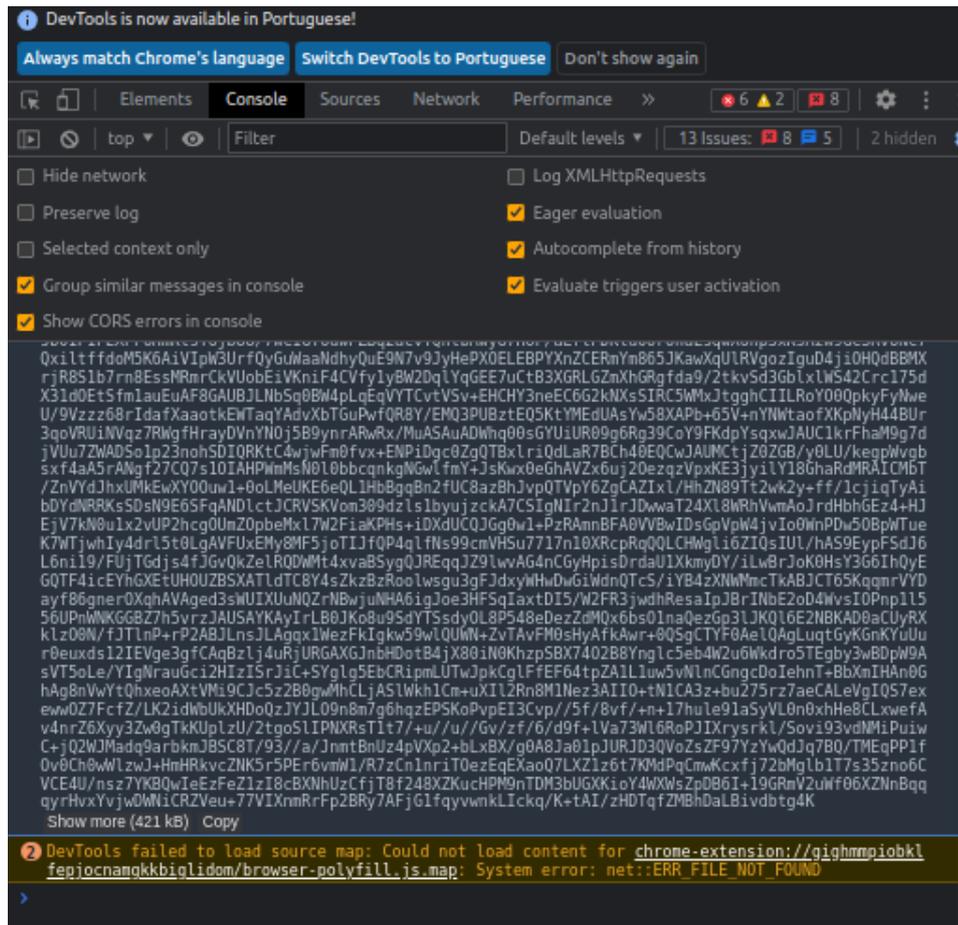
Com base no levantamento de requisitos, foi solicitado que o usuário precisaria cadastrar algumas informações para ser enviado a rede neural, tais como, lote, sexo, espécie do peixe e alguma informação adicional como campo de texto, adjunto a botões de *submit* e de acessar a câmera, sendo uma tela única na qual usuário poderá inserir informações pertinentes ao lote de peixes. Assim foi desenvolvido, como alternativa ao acessar a câmera foi utilizado recursos próprios do *browser* para ser simulado o acesso à câmera em figura 26 no qual o navegador solicitará uma permissão para carregar a câmera do dispositivo que acessa o site, gerando, portanto, uma *hash* da foto tirada através do dispositivo acessado, demonstrado figura 27.

Figura 26 – Navegador pedindo permissão para acessar a câmera do dispositivo que acessa o site.



Fonte: Próprio autor.

Figura 27 – Hash da foto tirada pelo botão de acessar a câmera.



Fonte: Próprio autor.

Através da *hash* exemplificada no console do navegador figura 27, é possível pegar a *url* da foto e renderizá-la na tela do dispositivo, uma vez que é gerado um objeto com um *link* para uma imagem, conforme tirada pela câmera do dispositivo.

Tendo em vista o front-end montado a estrutura que será útil para comunicação com a rede neural, fora proposto inicialmente testes de modo em que através da interface seja enviada uma imagem para a rede neural trar e retornar a contagem de peixes que estão presentes na foto submetida.

Para realizar o teste de comunicação entre o frontend e a rede neural, foi preciso executar tanto o frontend quanto a rede neural localmente, para que ambas possam se comunicar, conforme demonstrado na figura 31. A fim de facilitar a leitura, foram retirados partes de código na intenção de proporcionar o contato com o código desenvolvido, a ferramenta de *UI* utilizada é o *carbon*.

A simulação de envio de fotos e retorno da rede neural para o frontend consiste em uma abordagem útil para futuras implementações do projeto, uma vez em que fora testado tanto as aplicações isoladas quanto a comunicação entre duas partes distintas: o frontend da aplicação e a rede neural se comunicando.

Figura 28 – Script Vue.js que realiza comunicação com rede neural na porta 5000.



Fonte: Próprio autor.

O trecho mostrado na figura 31 utiliza o *axios* para fazer a requisição, um cliente *HTTP* que facilita a comunicação entre as *rotas* da aplicação, fornecendo todas as configurações necessárias para as *requisições*, incluindo também recursos de segurança. A comunicação é estabelecida localmente na porta 5.000, com a criação de um método *post* para a rota */count*, responsável por enviar a imagem. Além disso, é adicionado um cabeçalho padrão para requisições *HTTPs* locais. Após a execução da função chamada *send data*, a imagem é enviada para a rede neural.

Na figura 31, é ilustrada a comunicação entre a rede neural e o frontend. Podemos observar que um arquivo *jpg* é recebido da porta 5.000, que é o local onde a aplicação web está hospedada.

Antes de realizar o envio da imagem do tanque de peixes teste, é necessário executar a rede neural que ficará aguardando o recebimento da imagem para devolver uma resposta ao frontend da aplicação, conforme demonstrado na figura 29 no qual a imagem retrata a execução e funcionamento da rede neural, que recebe uma imagem *jpg* e retorna um *status 200* de sucesso ao receber uma imagem.

Em figura 29 a foto foi submetida de forma manual pelo usuário, mediante um campo de *input* no qual existe a opção de ser realizado o *upload* através do *click* no campo de carregamento de imagem, após a localização da imagem na máquina do usuário, resta somente clicar em salvar para ser submetido, portanto, à rede neural.

A rede neural, espera receber uma imagem de um tanque de peixes, para poder

Figura 29 – Funcionamento da rede neural em execução na porta 5000.

```
lucas@lucas > ~/Documentos/bcc-2/peixes/backend/onback > main ● ? ↓7 > python src/main.py  
[ '/home/lucas/Documentos/bcc-2/peixes/backend/onback/src', '/usr/local/lib/python310.zip', '  
Documentos/bcc-2/peixes/backend/onback/yolov7' ]  
* Serving Flask app 'main'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a produc  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
127.0.0.1 - - [20/Mar/2023 15:54:02] "OPTIONS /count HTTP/1.1" 200 -  
ImmutableMultiDict([('image', <FileStorage: 'peixes.jpeg' ('image/jpeg')>)])  
YOLOR 🚀 v0.1-116-g8c0bf3f torch 1.11.0+cu102 CPU
```

Fonte: Próprio autor.

então ser mapeado os peixes e contados, conforme exemplificado na figura 30. Uma imagem do tanque de peixes, na qual a câmera possa estar posicionada de forma superior com relação aos peixes, para que todos os elementos estejam visíveis.

Figura 30 – Formato de imagem que a rede neural espera receber da aplicação frontend



Fonte: Próprio autor.

Para que a função seja chamada, foi necessário anexar esta ao componente de formulário utilizado pelo *framework* Quasar, sendo o *Q-form*, sendo, portanto, deste componente é chamado o método de *submit*, referenciando na figura 31.

Após o envio no frontend da aplicação, é possível notar a resposta retornada no *console* do *browser*, foi programado para ser retornado ao *console* somente para fim de testes a contagem de peixes, com base na imagem submetida pelo usuário no navegador, demonstrado em figura 32. Adjunta a resposta no console, foi realizado a *interpolação*, que é uma técnica usada para inserir variáveis dinamicamente em uma *string*, sendo nesse caso, a quantidade de peixes demonstrado na tela para o usuário. Assim sendo, a rede neural possui a função de retornar ao frontend o número exato da contagem, nos

Figura 31 – Parte do código que chama função de envio e cria o campo de submissão de imagem

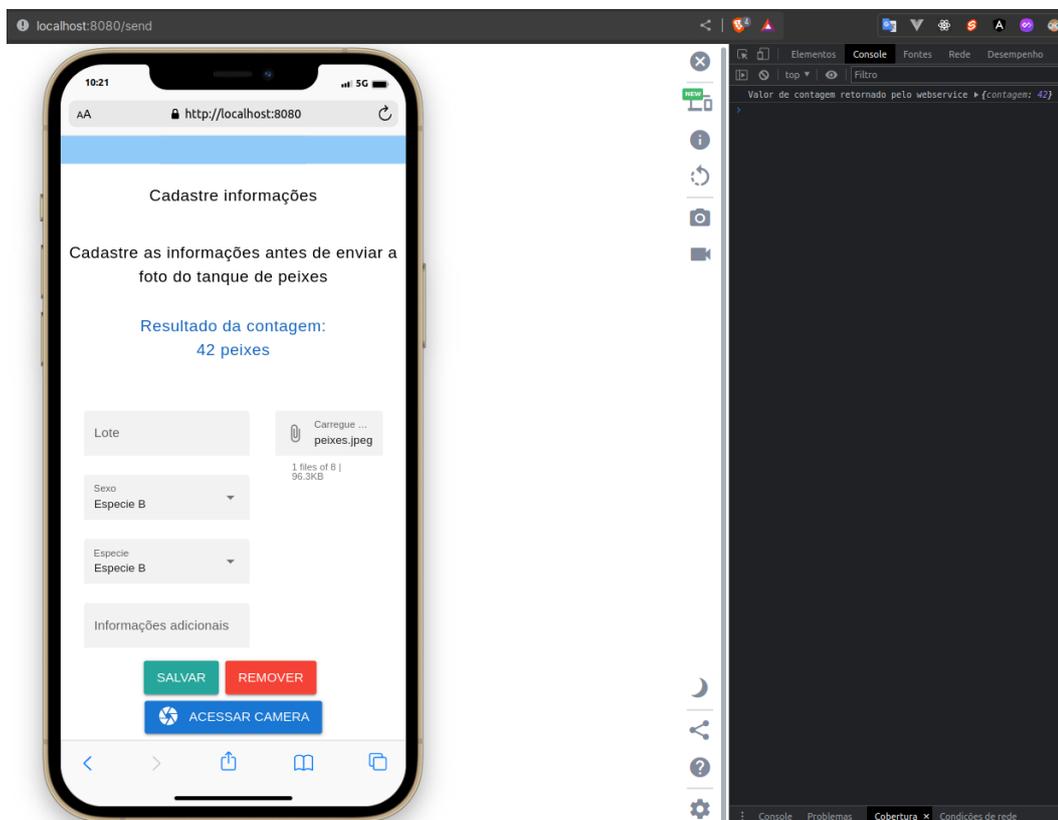
```
1 <template>
2   <q-form @submit.prevent="send_data">
3     <div class="col-12 items-center">
4       <div class="flex justify-center ml-2 mr-2">
5         <h5
6           class="text-center text-weight-regular font-poppins text-body1 q-mt-lg"
7         >
8           Cadastre informações
9         </h5>
10      </div>
11      <p
12        class="text-weight-regular text-center text-poppins q-pa-sm text-body1"
13      >
14        Cadastre as informações antes de enviar a foto do tanque de peixes
15      </p>
16      <p
17        class="text-blue-9 text-weight-regular text-center text-poppins q-pa-sm text-body1"
18        v-if="showResult"
19      >
20        Resultado da contagem: <br />
21        {{ count.contagem }} peixes
22      </p>
23      ...
24      <div class="q-pa-md">
25        <div class="q-gutter-md row items-start">
26          <q-file
27            v-model="files"
28            label="Carregue suas fotos"
29            filled
30            counter
31            type="file"
32            name="image"
33            ref="fileAdd"
34            :counter-label="counterLabelFn"
35            max-files="8"
36            multiple
37            style="max-width: 300px"
38            accept="image/*"
39          >
40            <template v-slot:prepend>
41              <q-icon name="attach_file" />
42            </template>
43          </q-file>
44        </div>
45      </div>
46    </q-form>
47  </template>
```

Fonte: Próprio autor.

exemplos abordados no presente trabalho, é retornado somente o número inteiro respectivo

ao resultado da contagem da inteligência artificial.

Figura 32 – Retorno no *browser* após envio da imagem para rede neural.



Fonte: Próprio autor.

6 CONCLUSÃO

O presente trabalho foi construído com base em metodologias ágeis, em específico, desenhado e programado utilizando scrum como princípio de estruturação e planejamento, a partir do mesmo foi possível programar uma solução web a fim de atender as necessidades do produto para empresa On-Peixes, sendo, portanto, uma aplicação web *frontend* que realiza a comunicação de forma local com a rede neural em figura 31, o qual é a responsável pela contagem de peixes.

Assim, foi possível atingir o resultado esperado para este trabalho, evidenciando o uso de metodologias ágeis em um projeto real com escopo de tempo de desenvolvimento definido orientado a metas conforme estabelecido no planejamento das *sprints* para a criação de uma aplicação web frontend que atendesse as necessidades na empresa On-Peixes.

A aplicação foi conectada à rede neural através de protocolos HTTP, possibilitando a interação entre elas. O frontend envia uma imagem para a rota 5.000, onde a rede neural está localizada, e recebe da rede neural a contagem exata dos peixes. Dessa forma, é viável enviar uma imagem de um tanque de peixes e obter na tela a contagem precisa dos peixes realizada pela inteligência artificial.

6.1 Trabalhos Futuros

Como trabalho futuro ficarão pendências com relação ao *deploy* da aplicação, mudar toda a estrutura local para uma arquitetura distribuída e descentralizada, de modo em que o usuário poderá interagir com a aplicação, necessitando apenas de acesso à internet e um celular.

O sistema pode ser incrementado a fim de separar as responsabilidades do software, como adaptação da *feature* de tirar uma foto instantaneamente e ser enviada à rede neural, existir um *backend* específico para armazenamento de dados de usuário e permitir a gerência das informações enviadas, bem como um histórico de informações computadas pela rede neural, estando, portanto, de fácil acesso para o usuário.

Como um facilitador, seria transpilar o projeto criado para uma PWA, possuindo comportamento como um aplicativo ao ambiente desktop *mobile* do usuário, ou transformar a ideia do produto criada para um aplicativo nativo e separar as responsabilidades sendo somente o ambiente web para divulgação e demonstração do produto enquanto o ambiente *mobile* nativo sendo a parte de usabilidade e venda do software.

Referências

- AMORIM, N.; GUEDES, G. Aplicando técnicas de v&v na fase de projeto do modelo em cascata. In: SBC. *Anais da I Escola Regional de Engenharia de Software*. [S.l.], 2017. p. 129–136. Citado na página 4.
- ARRUDA, L. V. Desenvolvimento ágil de software: uma análise sintética a partir da metodologia kanban. In: *VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação*. [S.l.: s.n.], 2012. Citado na página 10.
- BISSI, W. Metodologia de desenvolvimento ágil. *Revista Campo Digital*, v. 2, n. 1, 2007. Citado na página 9.
- BORTOLI, L. de. Experiência de uso do extreme programming (xp) em disciplina de engenharia de software. In: SBC. *Anais da I Escola Regional de Engenharia de Software*. [S.l.], 2017. p. 113–120. Citado na página 7.
- CARNEIRO, R. A. et al. O valor percebido pelo cliente na gestão de projetos de software, utilizando o framework scrum. *Revista de Gestão e Projetos*, v. 13, n. 3, p. 149–176, 2022. Citado 2 vezes nas páginas 15 e 16.
- CASTRO, R. M. de et al. Agility scrum-um jogo para ensino da metodologia scrum. In: SBC. *Anais do XXV Workshop sobre Educação em Computação*. [S.l.], 2017. Citado na página 9.
- COSTA, J. d. S. *Arquitetura de um sistema para automação residencial baseado em IoT*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2022. Citado na página 21.
- DREAMSTIME. *Modelo de comparação: diferenças entre desenvolvimento de software Scrum vs Kanban. Metodologia e estratégia do quadro de gestão de projetos*. 2023. Acessado em: 4 de abril de 2023. Disponível em: <<https://encurtador.com.br/ampG3>>. Citado na página 11.
- FADEL, A. C.; SILVEIRA, H. d. M. Metodologias ágeis no contexto de desenvolvimento de software: Xp, scrum e lean. *Monografia do Curso de Mestrado FT-027-Gestão de Projetos e Qualidade da Faculdade de Tecnologia-UNICAMP*, v. 98, p. 101, 2010. Citado na página 7.
- FARIAS, L. H. C. R. Estudo comparativo da utilização de design patterns no desenvolvimento de aplicação web utilizando frameworks front-end. 2022. Citado na página 7.
- FILHO, C. I. V. Aplicação web para avaliação de bares e restaurantes. Pontifícia Universidade Católica de Goiás, 2022. Citado na página 23.
- GENARI, J. O. S.; FERRARI, F. C. Times de alto desempenho no contexto das metodologias scrum e kanban. *Revista TIS*, v. 4, n. 3, 2016. Citado na página 11.
- Infoescola. 2021. Acessado em 25 de março de 2023. Disponível em: <<https://www.infoescola.com/engenharia-de-software/rup/>>. Citado na página 5.

- LEITZKE, D. M.; AMARAL, M. A. Html básico. *PROJETOS E RELATÓRIOS DE ESTÁGIOS*, v. 4, n. 1, p. 18–18, 2022. Citado na página 19.
- LESSA, R. O.; JUNIOR, E. O. lessa. Modelos de processos de engenharia de software. *Link para o PDF: http://xps-project.googlecode.com/svn-history/r43/trunk/outros/02_Artigo.pdf*, 2009. Citado na página 4.
- LIMA, L. G.; PETRUCCELLI, E. E.; SANTO, F. do E. Visão geral sobre o gerenciamento de estado no vue. js com a biblioteca vuex. *Revista Interface Tecnológica*, v. 16, n. 1, p. 56–66, 2019. Citado na página 21.
- LIMA, W. L. d. S. *Aplicando BDD em testes de REST API: uma experiência prática*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2022. Citado na página 1.
- MAIA, J. J. T. Desenvolvimento de uma ferramenta web para apoio a gerência de equipes de desenvolvimento distribuídas de software. 2022. Citado 2 vezes nas páginas 16 e 17.
- MANIKA, E. R. Desenvolvimento de um sistema de controle de distribuidora de doces utilizando a tecnologia java e rational unified process-rup como metodologia de desenvolvimento. Citado na página 5.
- MARTINS, R. et al. Práticas, técnicas e ferramentas de gestão do conhecimento para a definição de requisitos de software: um estudo de caso no laboratório bridge. Florianópolis, SC, 2017. Citado na página 4.
- MEDEIROS, M. A. de; NETO, M. M. C. Processos ágeis de desenvolvimento de software. Citado na página 6.
- NASCIMENTO, W. Um estudo de viabilidade de implementação do scrum em uma empresa de desenvolvimento de software. 2021. Citado na página 1.
- NEOVERO. *Método Kanban: Gerenciando projetos e tarefas com Neovero*. 2023. Acessado em 4 de abril de 2023. Disponível em: <<https://www.neovero.com/metodo-kanban-gerenciando-projetos-e-tarefas-com-neovero/>>. Citado na página 10.
- Nimblework. *Programação Extrema (XP) - Metodologia Ágil*. 2023. Disponível em: <<https://www.nimblework.com/pt-br/agile/programacao-extrema-xp/>>. Citado na página 8.
- NONEMACHER, M. L. et al. Comparação e avaliação entre o processo rup de desenvolvimento de software e a metodologia extreme programming. Florianópolis, SC, 2003. Citado na página 5.
- OLIVEIRA, G. M. d. Desenvolvimento e avaliação do plugin para o figma para documentação de acessibilidade para interfaces-dai. 2022. Citado na página 20.
- OLIVEIRA, R. A. de et al. Desafios no uso de metodologias ágeis de gestão de projetos em órgãos públicos:: um estudo de caso da receita estadual do paraná. *Gestão e Projetos: GeP*, Universidade Nove de Julho, v. 11, n. 2, p. 12–36, 2020. Citado na página 2.
- OLIVEIRA, T. M. de; FRANÇA, G. M.; ARAÚJO, M. A. P. Desenvolvimento ágil de aplicações com react e java-foster pet. *ANALECTA-Centro Universitário Academia*, v. 7, n. 2, 2022. Citado 2 vezes nas páginas 14 e 15.

- PEREIRA, D. J. Comparação entre metodologias de desenvolvimento de software baseadas nos métodos rup e xp. *Revista de Tecnologia Aplicada*, v. 5, n. 3, p. 46–52, 2017. Citado na página 5.
- PERIDES, M. P. N.; BARROTE, E. B.; SBRAGIA, R. As competências de gestores de projetos que atuam com métodos ágeis e tradicionais: um estudo comparativo. *Revista de Gestão e Projetos*, v. 12, n. 1, p. 11–38, 2021. Citado na página 12.
- PIETRANTONIO, E. G. *Desenvolvimento de rubricas para validação de design responsivo*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2021. Citado na página 22.
- PINCELLI-NETTO, M. Proposta para a criação de software educativo: diagrama de ishikawa para edema macular cistóide. Universidade Federal de São Paulo (UNIFESP), 2016. Citado na página 4.
- PISSURNO, P. M. A. Geração automática de anotações jsdoc para consultas sql raw em javascript. 2022. Citado na página 23.
- PONTES, T. B.; ARTHAUD, D. D. B. Metodologias ágeis para o desenvolvimento de softwares. *Ciência E Sustentabilidade*, v. 4, n. 2, p. 173–213, 2018. Citado na página 7.
- PRAZERES, A. P. Princípios para o desenvolvimento de software seguro. *Engenharia de Projetos de Software-Florianópolis*, 2015. Citado na página 4.
- RABETTI, R. d. S. Comparativo entre as metodologias atômica e tradicional no desenvolvimento de css. 2019. Citado na página 19.
- RAMOS, D. F. et al. Software de qualidade: Metodologias e desafios. Citado na página 8.
- SANTIAGO, C. P. et al. Desenvolvimento de sistemas web orientado a reuso com python, django e bootstrap. *Sociedade Brasileira de Computação*, 2020. Citado na página 22.
- SANTOS, J. T. S. d. *Desenvolvimento de uma interface gráfica para um robô de conversação (Chatbot) no contexto educacional*. Dissertação (B.S. thesis) — Universidade Federal do Rio Grande do Norte, 2022. Citado na página 20.
- SATO, D. T. Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. *Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo*, v. 139, 2007. Citado 2 vezes nas páginas 4 e 7.
- SILVA, D. V. d. S.; SANTOS, F. A. d. O.; NETO, P. S. Os benefícios do uso de kanban na gerência de projetos de manutenção de software. In: SBC. *Anais do VIII Simpósio Brasileiro de Sistemas de Informação*. [S.l.], 2012. p. 715–725. Citado na página 10.
- SILVA, P. R. da; SANTOS, M. R. dos; SHIBAO, F. Y. Desenvolvimento de softwares: Cmmi e metodologias ágeis. *Revista Livre de Sustentabilidade e Empreendedorismo*, v. 4, n. 3, p. 157–184, 2019. Citado na página 8.
- SOARES, M. dos S. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, v. 3, n. 2, p. 8–13, 2004. Citado na página 12.

SRL, R. S. P. *Documentação oficial do Quasar*. 2022. Url:<https://quasar.dev/>. Citado na página 28.

TREINAWEB. *O que é DOM, Virtual DOM e Shadow DOM*. 2023. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-dom-virtual-dom-e-shadow-dom>>. Citado na página 20.

TRINDADE, P. E. O progressive web apps–pwa: como ferramenta para a produção audiovisual. Universidade Estadual Paulista (Unesp), 2020. Citado 2 vezes nas páginas 21 e 22.

VALERO, D. M. Negação de serviço em aplicações node. js. Faculdade de Tecnologia de Americana, 2019. Citado na página 19.

YOU, E. *Documentação oficial do Vue.js*. 2023. Url:<https://vuejs.org/>. Citado na página 27.

YOU, E. S. M. M. E. *Documentação oficial do Vue Router*. 2022. Url:<https://router.vuejs.org/>. Citado na página 28.