

MÉTODOS ÁGEIS E UML: DESENVOLVIMENTO DO APLICATIVO VEG

Marcelo Giacomini Bonato¹, Rodrigo de Oliveira Plotze¹

¹Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

marcelo.bonato@fatec.sp.gov.br,
rodrigo.plotze@fatec.sp.gov.br

Resumo. *O consumo de artigos vegetarianos e veganos cresce anualmente em diversas partes do mundo. Buscando facilitar o comércio destes produtos entre produtores e consumidores, estamos desenvolvendo o aplicativo de comércio eletrônico VEG, e para tal serão utilizadas as metodologias ágeis e a linguagem UML, fornecendo um ambiente de identificação visual das funcionalidades requeridas em um desenvolvimento iterativo e em constante atualização.*

Abstract. *The consumption of vegetarian and vegan items grows annually in various parts of the world. Seeking to facilitate the trade of these products between producers and consumers, we are developing the VEG e-commerce application, and for this will be used the agile methodologies and the UML language, providing an environment of visual identification of the functionalities required in an iterative development and constant updating.*

1. Introdução

Esta pesquisa visa avaliar a viabilidade da aplicação das metodologias ágeis em conjunto com os modelos previstos na UML no desenvolvimento de um aplicativo de *e-commerce* de produtos vegetarianos e veganos, com possibilidade de expansão para outros segmentos.

O aplicativo em desenvolvimento busca atender a demanda de um mercado em plena expansão no mercado brasileiro. Cerca de 14% das pessoas se declararam veganos ou vegetarianos em pesquisa realizada pelo IBOPE em 2018, segundo informações disponibilizadas pelo SVB – Sociedade Vegetariana Brasileira (2020), número este que cresce para 16% nas regiões metropolitanas de São Paulo, Curitiba, Recife e Rio de Janeiro. Isto representa um aumento de 75% em relação aos dados de 2012, quando a mesma pesquisa indicou um total de 8% de pessoas que se declaravam vegetarianas. Levando em números totais, cerca de 30 milhões de brasileiros se identificam com este tipo de alimentação. Mesmo com um cenário de crise econômica enfrentado pelo país, o mercado de produtos veganos cresceu na ordem de 40% ao ano, segundo dados

apontados pelo jornal Folha de São Paulo (2016).

Assim, busca-se o desenvolvimento de um aplicativo que atenda a necessidade deste nicho em expansão e verificar se é possível implementar as soluções presentes nos métodos ágeis de desenvolvimento de software em conjunto com a documentação e diagramação fornecida pela UML, em detrimento do modelo de desenvolvimento tradicional, conforme discutido por Russo Ramos (2020).

1.1. Justificativa

Foi observado que os atuais aplicativos de compra e delivery de mercadorias não fornecem soluções que satisfazem as necessidades deste nicho e identificou-se a necessidade de desenvolvimento de um aplicativo focado neste serviço, com possibilidades de expansão para demais produtos voltados para este mercado além dos alimentícios, como vestuário e produtos para pets, dentre outros. Além disso, pretende-se criar um canal direto entre os pequenos produtores e os clientes que buscam seus produtos, o que atualmente ocorre de maneira deficitária ou inexistente, havendo quase sempre a necessidade de um intermediário, como mercados ou varejões.

Visando ocupar esta lacuna de oportunidade, pretende-se desenvolver um aplicativo que forneça o atendimento desejado por este nicho de mercado de maneira mais efetiva, aplicando as metodologias ágeis e a UML. Para tanto serão buscadas parcerias com pequenos produtores e demais distribuidores de produtos vegetarianos e veganos, assim como de agências de fomento interessadas no desenvolvimento desta tecnologia.

Serão utilizadas as documentações oficiais da UML e de outras metodologias ágeis pertinentes ao desenvolvimento do aplicativo.

1.2. Problema de pesquisa

A presente pesquisa visa responder ao questionamento: é possível a aplicação de metodologias ágeis em conjunto com a UML no desenvolvimento de um aplicativo de delivery de produtos veganos?

1.3. Objetivo geral

A presente pesquisa tem como objetivo geral verificar a aplicabilidade das metodologias ágeis juntamente com a parte de modelos e da documentação prevista na UML, em um projeto de desenvolvimento de um aplicativo para dispositivos móveis e plataforma web para atender a demanda crescente de entregas de produtos, indo desde alimentos e

produtos veganos a produtos para pets, mercados em expansão contínua e regular.

1.4. Objetivos específicos

Para atender ao objetivo geral, buscar-se-á a utilização das metodologias ágeis e analisar se estas metodologias em conjunto com a utilização dos diagramas previstos na UML contribuem para um fluxo de trabalho mais dinâmico e organizado, com entregas contínuas de funcionalidades e atualizações constantes e com uma documentação mais completa para o software em questão.

1.5. Hipótese de pesquisa

Acredita-se sim ser possível a aplicação das metodologias ágeis e da UML em conjunto e que essa junção colaborará para uma maior agilidade no processo de desenvolvimento devido à maior facilidade de comunicação entre a equipe, visto que os modelos da UML fornecem uma visão mais objetiva dos aspectos do sistema em desenvolvimento, e consequentemente nas tomadas de decisões necessárias para que o projeto esteja sempre em evolução, com entregas constantes e revisão periódica do processo, premissas de todas as metodologias ágeis

2. Métodos Ágeis, Scrum e UML

2.1. Métodos Ágeis

Em um mundo cada vez mais conectado e dinâmico, as empresas precisam de respostas mais rápidas e constantes para as novas tecnologias e oportunidades que surgem. Neste contexto, a forma tradicional de desenvolvimento de software, com requisitos muito bem estabelecidos e rígidos, em uma sequência de desenvolvimento que ia somente em uma direção, representado geralmente por uma cascata, com as etapas extremamente detalhadas e planejadas, não atendiam essa mudança de contexto. É nesse cenário que começam a surgir as metodologias ágeis, que propõem uma forma de desenvolvimento mais voltada a entrega de valor imediato ao cliente e com atualizações constantes, com a direção de desenvolvimento se tornando ciclos, onde há uma entrega de funcionalidade, utilização e feedback, que gera um novo ciclo, em um processo que mais parece uma espiral.

A insatisfação com o modelo mais tradicional de desenvolvimento, levou alguns desenvolvedores a, no início da década de 1990, pensar no que ficou conhecido como “métodos ágeis” (SOMMERVILE, 2011). Estes métodos permitiram que as equipes de desenvolvimento focassem mais no software em si, do que em sua concepção e planejamento, permitindo entregas mais rápidas e funcionais. De acordo com Sommerville, “embora todos os métodos ágeis sejam baseados na noção de desenvolvimento e entrega incremental, eles propõem diferentes processos para alcançar tal objetivo.” (SOMMERVILE, 2011, p. 56) Um dos mais conhecidos métodos ágeis é o Scrum.

2.2. Scrum

O Scrum foi trazido em 1995 por Jeff Sutherland e Ken Schwaber em uma conferência realizada em Austin, Texas. Os autores se aproveitaram do termo apresentado por Takeuchi e Nonaka em 1986, no artigo “The New Product Development Game”. Este termo é originário do Rugby, sendo um movimento que valoriza o trabalho em equipe no jogo. Neste artigo, Takeuchi e Nonaka concluem que equipes pequenas e auto-organizadas obtêm um desempenho melhor no desenvolvimento de produtos complexos, definindo tarefas ao invés de objetivos (DE OLIVEIRA, 2016).

Assim, nota-se que Scrum e métodos ágeis não são sinônimos. As metodologias ágeis tiveram como marco inicial o Manifesto Ágil, um artigo publicado inicialmente em 2001 por dezessete desenvolvedores que já utilizavam diferentes metodologias como o próprio Scrum, Extreme Programming, e outras, para documentar alguns princípios identificados por eles como fundamentais para um desenvolvimento de software de qualidade, e o termo ágil acabou sendo utilizado por melhor captar os ideais apresentados. Os valores apresentados são os seguintes, segundo Beck (et al, 2001):

Indivíduos e interações mais que processos e ferramentas
Software em funcionamento mais que documentação abrangente
Colaboração com o cliente mais que negociação de contratos
Responder a mudanças mais que seguir um plano (Manifesto para Desenvolvimento Ágil de Software)

Jeff e Ken estavam entre os dezessete desenvolvedores que idealizaram o Manifesto Ágil, e fundaram diversas instituições para certificar profissionais para a utilização de Scrum, como Scrum Alliance e a Scrum Org. O primeiro Scrum Guide foi publicado em 2010, com atualizações periódicas, sempre buscando o aperfeiçoamento do framework. Neste guia, o Scrum é definido como sendo um “framework leve que ajuda pessoas, times e organizações a gerar valor por meio de soluções adaptativas para problemas complexos.” (Schwaber, Sutherland, p. 4, 2020). Ainda neste guia, os autores afirmam que o framework é incompleto propositalmente, apresentando somente as estruturas básicas de relacionamentos e interações, e deixando que a inteligência coletiva das equipes que o utilizarem sejam aproveitadas.

Atualmente, o framework é utilizado por diversas empresas ao redor do globo, não se limitando a somente empresas de desenvolvimento de software, mas como descrito por Mike Cohn, a mudança para o Scrum não é um processo fácil, pois além de mudar as práticas de uma empresa ele também pede alteração na forma de se pensar (Cohn, 2011, p.25). Dado o tamanho da equipe de desenvolvimento deste projeto e que o framework será utilizado desde seu início, esta parte de adaptação tende a ser menos custosa, o que proporcionará que os ganhos por sua utilização sejam sentidos desde o começo. Segundo Cohn:

Apesar de todas as razões por trás das dificuldades da transição para o Scrum, stakeholder de empresas que fizeram a transição estão felizes porque a fizeram. Uma razão pela qual os stakeholders estão tão satisfeitos é que o time-to-market é reduzido com o uso de um processo ágil como o Scrum. Esse time-to-market mais veloz é gerado pela produtividade mais alta das equipes de desenvolvimento ágil. Já que os funcionários ficam livres para fazer um trabalho de alta qualidade, e já que

veem seu trabalho sendo entregue mais cedo para os usuários que estão esperando, a satisfação no trabalho aumenta. Com uma satisfação no trabalho mais alta vem um maior engajamento dos funcionários, o que leva a mais ganhos de produtividade, iniciando um eficiente ciclo de melhoria contínua. (2011, p.32)

O autor também cita uma última razão para se adotar o Scrum. Segundo ele, o processo de desenvolvimento utilizado até então não está mais funcionando devido à velocidade das mudanças pelas quais o mundo vem passando, o que torna necessário que as equipes se adaptem a esse novo ritmo e busquem cada vez a agilidade em seus processos.

2.3. UML

Como observado por de Oliveira e Vendramel (2021), o escopo da documentação fica em segundo plano quando se fala em metodologias ágeis e, neste contexto, a documentação visual pode ser uma alternativa mais viável. Assim, a UML possui bastante potencial para atender essa finalidade, trazendo vantagens principalmente no compartilhamento de informações entre os participantes do projeto.

A UML é um modelo de linguagem para modelagem de dados orientado a objetos, utilizada para especificar, construir, visualizar e documentar um sistema de software. Com ela podemos entender melhor os relacionamentos e a estrutura do sistema através da visualização de seus componentes.

Podemos utilizar a UML em diversas fases do desenvolvimento, desde sua concepção até a geração de códigos. Pode também ser aplicada em qualquer tipo de sistemas, sendo mais utilizada na modelagem de softwares com o paradigma de orientação a objetos.

Sendo que a UML gera diversos modelos visuais, pode-se facilmente acoplá-los às metodologias ágeis, já que artefatos visuais são uma das formas mais eficientes de se identificar os requisitos. A visualização do modelo resulta em uma imagem da solução, permitindo assim aos stakeholders o entendimento do que será entregue, tornando mais fácil a comunicação entre os membros do time e o compartilhamento de informações sobre o projeto (CHEN; BEATTY, 2012).

3. Materiais e métodos

3.1. Elicitação dos requisitos

Com a realização de entrevistas entre os desenvolvedores e demais stakeholders do projeto, elencamos as necessidades funcionais e não funcionais para o desenvolvimento do aplicativo (SOMMERVILLE, 2007).

3.1.1. Resumo das entrevistas

- O software Veg deverá ser executado em modo web, e ser compatível com os

principais navegadores web disponíveis no mercado

- O aplicativo Veg deverá funcionar em sistema Android e IOS
- O software/aplicativo será uma vitrine para que os lojistas possam vender produtos vegetarianos e veganos
- Deverá conter um termo de uso da plataforma para isentar a plataforma de vínculo empregatício com lojistas e entregadores
- O sistema deverá solicitar o CNPJ dos lojistas e entregadores para concluir o cadastro no sistema
- O sistema deverá recusar o cadastro de lojistas pessoa física
- O sistema deverá cobrar uma taxa de 15% dos lojistas em cada transação
- O sistema deverá ser desenvolvido pensando na experiência dos usuários, lojistas
- As vitrines das lojas deverão conter a logomarca dos lojistas
- Para acessar os produtos os clientes deverão acessar as vitrines
- Os produtos deverão ser exibidos na forma de menu contendo uma figura ilustrativa, os ingredientes, e seu respectivo valor
- Deverão existir valores mínimo e máximo para os cálculos de frete
- O sistema deverá descontar automaticamente dos lojistas as taxas de utilização da plataforma
- O sistema deverá ser integrado para aceitar pagamentos por cartões de crédito, pix e pagamentos em dinheiro

3.2. Desenvolvimento do aplicativo

O aplicativo Veg foi desenvolvido para atender as plataformas Android e IOS utilizando Flutter e o Firebase. Para compartilhamento de código entre os desenvolvedores e para o versionamento do código foi utilizado o GitHub.

Flutter é um framework que possibilita criar aplicativos nativos para dispositivos Android e iOS programando na linguagem Dart. O Flutter compila código Advanced RISC Machine (ARM) nativo para as duas plataformas, que resulta em um aplicativo com alto desempenho (FLUTTER, 2021). A compilação do código, juntamente com todos os recursos utilizados foi feita através do SDK (Software Development Kit), e o resultado dessa compilação foi um arquivo APK (Android Package) que por sua vez possui todas as informações e características do aplicativo. É através do APK que um usuário instala o aplicativo no seu Smartfone (FLUTTER, 2021).

O Firebase, desenvolvido pela Google, é um conjunto de produtos distribuídos gratuitamente, com um limite de utilização (FIREBASE, 2021). Entre esses produtos, existem serviços de hospedagem, armazenamento em nuvem e banco de dados. No caso do banco de dados, a utilização gratuita permite até 100 acessos simultâneos. Essa ferramenta permite, com poucas linhas de código adicionar o banco de dados em aplicações web, Android e iOS para que se conectem ao mesmo banco, sem, contudo, requerer conhecimentos sobre a infraestrutura do sistema. Neste aplicativo utilizamos um bando de dados NoSql.

4. Resultados

O software Astah UML, com suporte a UML 2 (ASTAH, 2021), foi o utilizado para a elaboração dos diagramas. Elaboramos os diagramas de Caso de Uso de Negócio, de Caso de Uso, de Classes e de Sequência, apresentados abaixo, com um exemplo da descrição de cada diagrama:

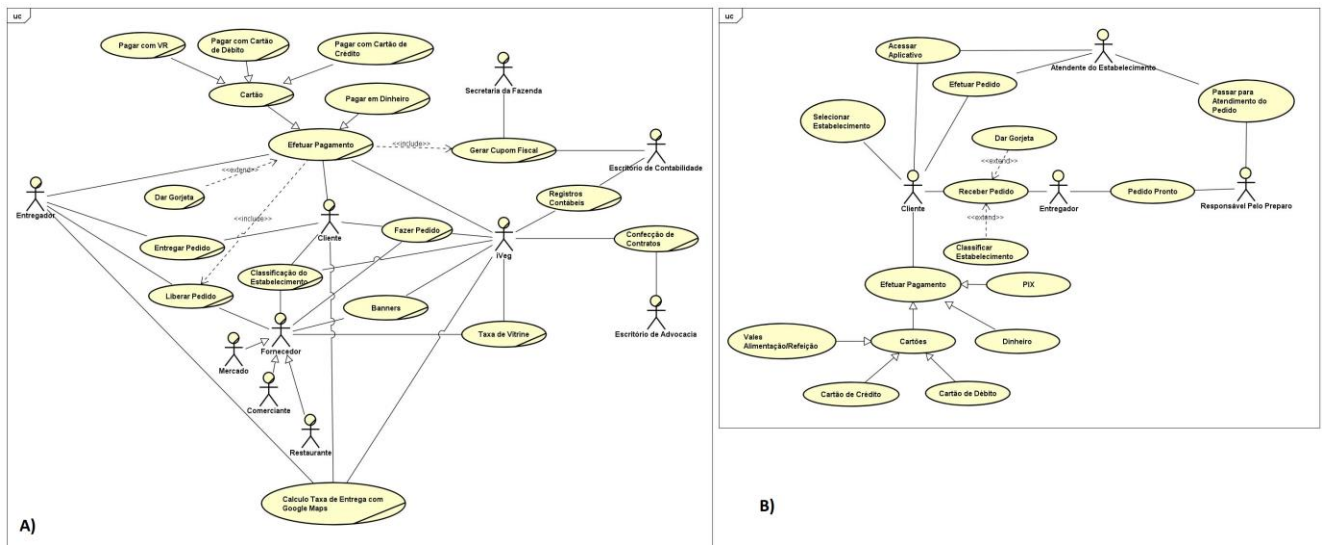


Figura 1: A) Diagrama de Caso de Uso de Negócio Aplicativo Veg; B) Diagrama de Caso de Uso do aplicativo Veg

Fonte: Autoria Própria, 2021

TABELA 01: Exemplo de descrição de caso de uso de negócios

UC001	Fazer Pedido
Atores	Cliente e Veg
Descrição	Para realizar o pedido, o cliente deverá acessar sua conta no aplicativo Veg e selecionar o estabelecimento e produto desejado.
Regras de Negócio	<ul style="list-style-type: none"> O cliente deverá criar uma conta de acesso informando seus dados, tais como nome, CPF, endereço, telefone e senha. O aplicativo deverá exibir a lista de estabelecimentos cadastrados que atendam a região relativa ao CEP informado pelo cliente no cadastro.

Fonte: Autoria Própria, 2021

TABELA 02: Exemplo de descrição de caso de uso

UC001	Efetuar Pedido
Atores	Cliente e Atendente do Estabelecimento
Descrição	Para realizar o pedido, o cliente deverá acessar sua conta no aplicativo Veg e selecionar o estabelecimento e produto desejado. O atendente do estabelecimento também deverá acessar sua conta e verificar os pedidos recebidos.

Regras de Negócio	<ul style="list-style-type: none"> • O cliente poderá incluir produtos de mais de um estabelecimento em seu carrinho, sendo a taxa de entrega calculada isoladamente para cada estabelecimento. • O Atendente deverá verificar a existência de pedidos e repassar para o atendimento.
-------------------	---

Fonte: Autoria Própria, 2021

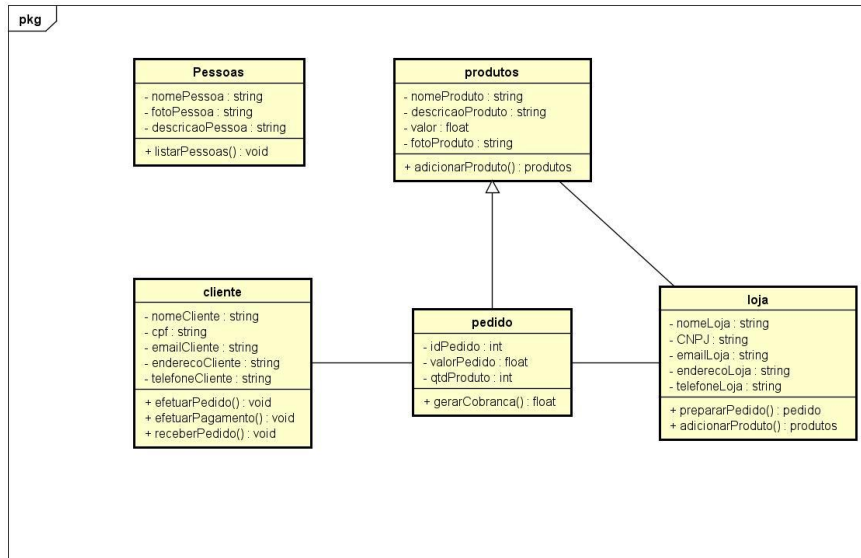


Figura 3: Diagrama de Classes do aplicativo Veg

Fonte: Autoria Própria, 2021

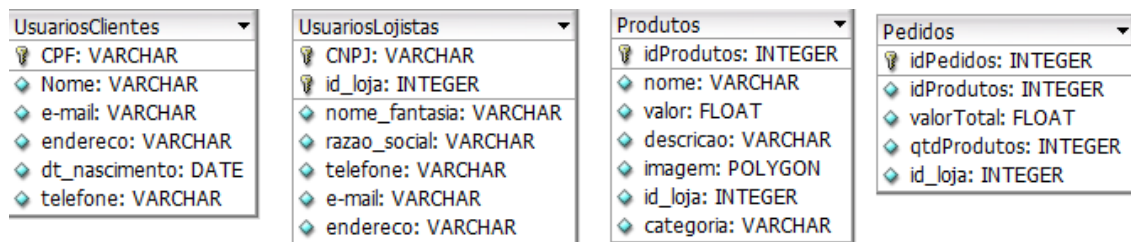


Figura 4: Coleções do Firebase do aplicativo Veg

Fonte: Autoria Própria, 2021

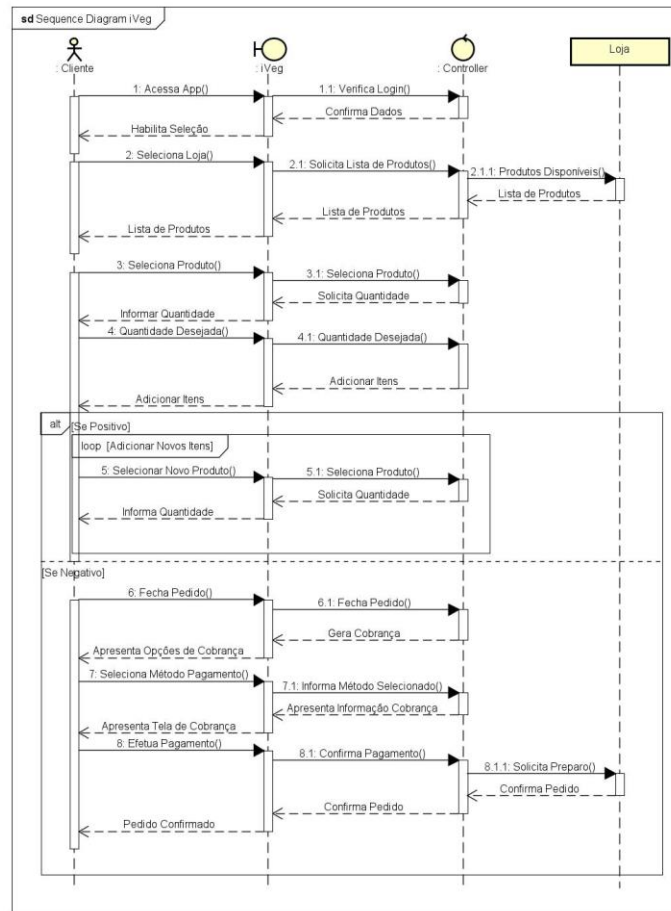


Figura 5: Diagrama de Sequência do aplicativo Veg

Fonte: Autoria Própria, 2021

Com os diagramas definidos foi elaborado o backlog priorizado do produto e definido a duração de quatro semanas para cada sprint.

A primeira sprint teve os requisitos e funcionalidades definidos conforme abaixo:

- Criação do banco de dados no Firebase
- Tela inicial
- Tela institucional
- Tela de cadastro e acesso de usuários
- Tela de cadastro de clientes e parceiros de vendas

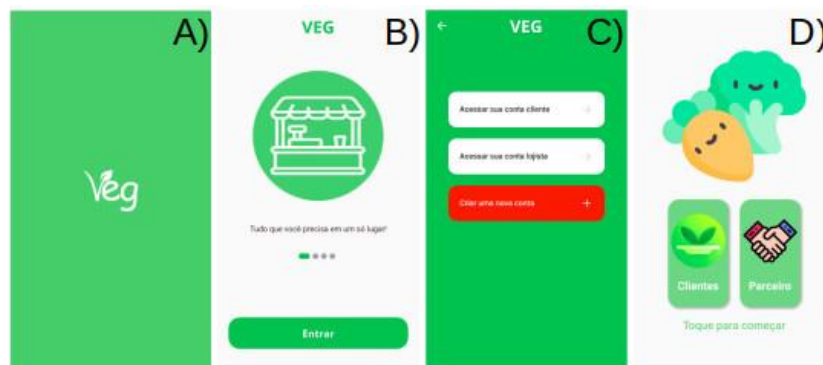


Figura 6 – Apresentação das telas iniciais do aplicativo Veg. A) Tela inicial, B) Tela institucional, C) Tela de cadastro e acesso de usuários, D) Tela de cadastro de clientes e parceiros de vendas.

Fonte: Autoria própria, 2021

Na segunda sprint foram desenvolvidas as telas e funcionalidades abaixo:

- Cadastro de parceiro de vendas
- Login de Parceiro de Vendas
- Tela do painel administrativo
- Tela de visualização e exclusão de produtos
- Tela de cadastro e atualização de produtos

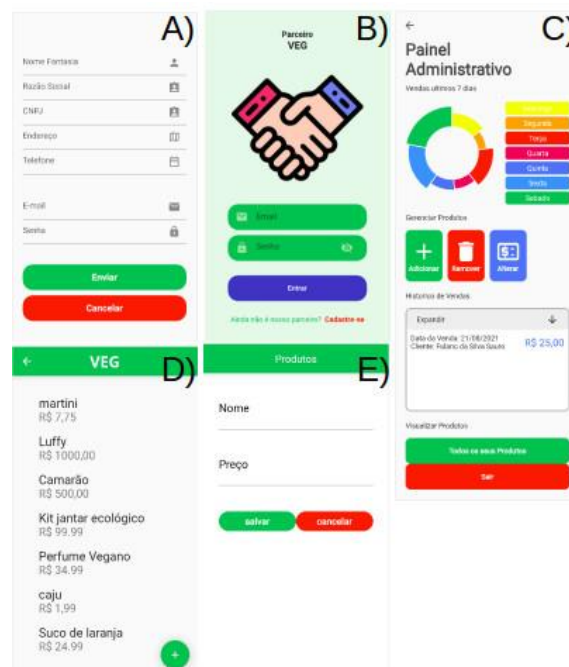


Figura 7 – Apresentação das telas: A) Cadastro de parceiro de vendas, B) Login de parceiro de vendas, C) Tela do painel administrativo, D) Tela de visualização e exclusão de produtos, E) Tela de cadastro e atualização e produtos.

Fonte: Autoria própria, 2021

Assim, sucessivamente as sprints foram sendo desenvolvidas e geraram as demais telas apresentadas abaixo, resultando na versão 0.1.10 do aplicativo:

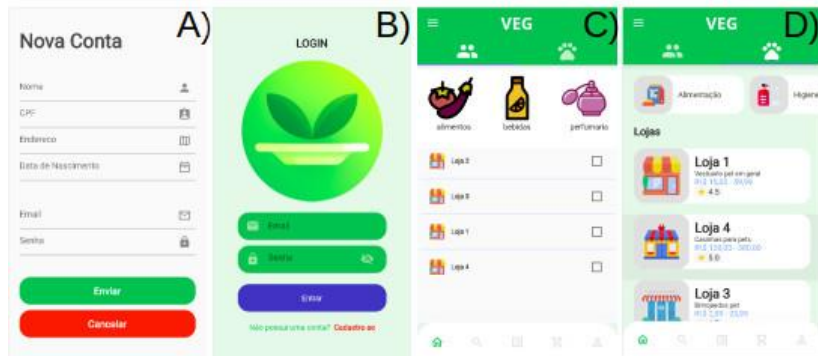


Figura 8 – Apresentação das telas: A) Tela de cadastro de usuários, B) Tela de login, C) Tela de menu de lojas e produtos destinados a humanos, D) Tela de menu de lojas e produtos destinados aos PETs.

Fonte: Autoria própria, 2021



Figura 10 – Apresentação das telas: A) Tela de pesquisa. B) Tela de histórico de compras, C) Tela de perfil.

Fonte: Autoria própria, 2021

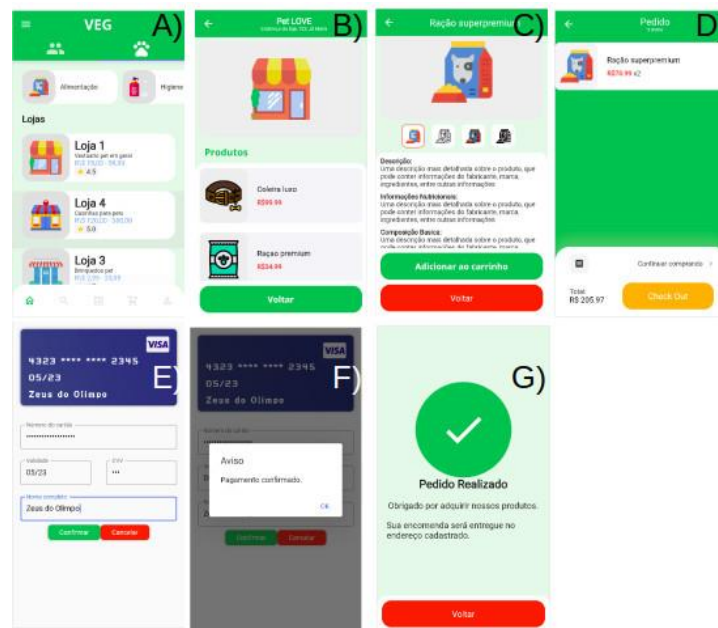


Figura 11 – Apresentação das telas: A) Tela de menu, B) Tela de seleção de produtos, C) Tela de adição de produtos ao carrinho, D) Tela de checkout, E) Tela de pagamento, F) Tela de confirmação de pagamento, G) Tela de confirmação do pedido.

Fonte: Autoria própria, 2021

5. Conclusão

Neste trabalho buscamos analisar a possibilidade da utilização da UML e de metodologias ágeis para o desenvolvimento do aplicativo Veg. Este aplicativo teve sua versão inicial desenvolvida seguindo um fluxo trabalho pertinente às metodologias ágeis, com funcionalidades sendo entregues de maneira incremental e iterativa, buscando sempre continuar aperfeiçoando a experiência do usuário com base nos feedbacks recebidos e trazendo novas funcionalidades de acordo com a prioridade do backlog do produto. Além disso, a utilização dos diagramas da UML permitiu uma noção objetiva inicial do sistema, visão esta que deverá também continuar em evolução conforme as atualizações de funcionalidades e do modelo de negócio sejam implementadas.

Assim sendo, verificou-se que a UML pode acrescentar e muito ao desenvolvimento de softwares desenvolvidos com metodologias ágeis, agregando uma visão geral do sistema e contribuindo para uma documentação mais bem elaborada caso essa junção não fosse realizada.

Valendo-se de ferramentas de comunicação, tais como Whatsapp, Microsoft Teams e Slack, a equipe de desenvolvimento foi capaz de desenvolver uma versão inicial 0.1.10 do aplicativo Veg. Com o foco em entregas de valor ao usuário, esta primeira versão será responsável pela coleta de feedbacks junto aos stakeholders para que as atualizações e adequações sejam realizadas de maneira sucessiva e incremental, como rege a premissa das metodologias ágeis.

6. Referências

- ASTAH. Vantagens com o Astah Pro. Disponível em: <<http://astah.net/editions/uml-new#advantages>>. Acesso em: 20 de setembro de 2021.
- COHN, Mike (2011) Desenvolvimento de Software com Scrum Aplicando Métodos Ágeis com Sucesso. 1ª Edição, Editora Bookman.
- OLIVEIRA, Rosana de. (2016) Introdução ao Scrum. Disponível em: <<https://blog.romarconsultoria.com.br/2016/05/ebook-introducao-ao-scrum.html>>. Acesso em: 15 de agosto de 2021.
- RANGEL, Ana. Pequenas empresas de produtos vegetarianos crescem 40% ao ano. (2016) Folha de São Paulo 04/07/2016. Disponível em: <http://www1.folha.uol.com.br/mercado/2016/07/1787773-pequenas-empresas-de-produtos-vegetarianos-crescem-40-ao-ano.shtml>.> Acesso em: 15 de agosto de 2021.
- SCHWABER, Ken; SUTHERLAND, Jeff; et al. (2001) Manifesto Ágil para o Desenvolvimento de Software, <http://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 20 de agosto de 2021
- SCHWABER, Ken; SUTHERLAND, Jeff. (2020) O Guia do Scrum, <https://scrumguides.org/>. Acesso em: 20 de agosto de 2021.
- FIREBASE (2021) Documentação do firebase. Disponível em:<<https://firebasegoogle.com/docs/>>. Acesso em: 20 setembro 2021.
- FLUTTER. Flutter.Dev. Documentação Flutter. Disponível em: <<https://flutter.dev/>>. Acesso em: 20 setembro 2021.
- Sociedade Vegetariana Brasileira (2018) Mercado Vegetariano. Disponível em: <<https://www.svb.org.br/vegetarianismo1/mercado-vegetariano.html>> Acesso em: 15 de agosto de 2021.
- DE OLIVEIRA, Wellington Kauê Ribeiro; VENDRAMEL, Wilson. (2021) A Relação Entre Metodologias Ágeis e UML no Desenvolvimento de Software: Uma Revisão Sistemática da Literatura. RETEC – Revista de Tecnologias v.13 n.2 (2020)