

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Desafios, Agilidade e Simplicidade: Uma Abordagem para Desenvolvimento *Mobile*

Autor: Leonardo Cambraia Corrêa
Orientador: Profa. Dra. Milene Serrano

Brasília, DF
2018



Leonardo Cambraia Corrêa

Desafios, Agilidade e Simplicidade: Uma Abordagem para Desenvolvimento *Mobile*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2018

Leonardo Cambraia Corrêa

Desafios, Agilidade e Simplicidade: Uma Abordagem para Desenvolvimento *Mobile*/ Leonardo Cambraia Corrêa. – Brasília, DF, 2018-
119 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2018.

1. Abordagem. 2. *Mobile*. 3. Scrum. 4. *Challenge Based Learning*. 5. *Lean Thinking*. I. Profa. Dra. Milene Serrano. II. Prof. Dr. Maurício Serrano. III. Universidade de Brasília. IV. Faculdade UnB Gama. V. Desafios, Agilidade e Simplicidade: Uma Abordagem para Desenvolvimento *Mobile*

CDU 02:141:005.6

Leonardo Cambraia Corrêa

Desafios, Agilidade e Simplicidade: Uma Abordagem para Desenvolvimento *Mobile*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado.

Brasília, DF, 04 de julho de 2018: 08:00.

Profa. Dra. Milene Serrano
Orientador

Prof. Dr. Maurício Serrano
Coorientador

Prof. Msc. Ricardo Ajax Kosloski
Convidado 1

Brasília, DF
2018

Este trabalho é dedicado ao meu avô Sebastião dos Reis Cambraia, meu vô Tião, que mesmo nas mais adversas condições se esforçou para que todos os seus filhos tivessem a melhor educação possível; e mesmo tendo falecido antes que eu entrasse na Universidade, estaria imensamente orgulhoso hoje.

Agradecimentos

Agradeço primeiramente a Deus, por me dar a capacidade de estudar e chegar aonde cheguei.

É impossível agradecer quaisquer pessoas sem antes mencionar os sacrifícios que meus pais, Regina e Cláudio, fizeram para que eu sempre tivesse uma boa educação. Por isso, é com todo o amor do mundo que os agradeço, juntamente a minha irmã Luciana.

Agradeço quem mais me incentivou durante este trabalho, Bruna, a pessoa que amo e escolhi para partilhar os meus sucessos e me apoiar durante momentos difíceis, que tanto cuida de mim e que conquistou seu lugar em minha vida, de uma maneira sem precedentes.

Agradeço com muito carinho meu amigo, Castellar, que desde cedo, em nossa educação, me apoia em todos os aspectos, tornando-se um dos meus melhores amigos ao longo dos anos.

Agradeço minha grande amiga, Júlia, que, apesar de nossos altos e baixos, nunca deixou de torcer e brigar por mim, ao passo que sempre me ajudou nos momentos que precisei.

Agradeço com muito carinho minha amiga, Karol, que sempre esteve ao meu lado, em todos os grandes momentos nos últimos anos, onde crescemos juntos, tanto academicamente como em nossas vidas pessoais, sempre nos apoiando nas idas e vindas da graduação e nos formando pessoas melhores.

Agradeço especialmente meus amigos Fernanda Muro, Geovanni Oliveira, Lucas Rufino, Paulo Markes e Tainara Reis e também outros amigos e colegas que me apoiaram durante a graduação.

Agradeço a todos os meus professores que, ao longo da graduação, me moldaram como engenheiro, em especial aos meus orientadores, Milene e Maurício, que durante essa reta final me auxiliaram e me cobravam excelência.

A todos estes, meu mais sincero muito obrigado.

*"Existe algo mais perigoso do que deuses insatisfeitos
e irresponsáveis que não sabem o que querem?"
(Uma breve história da humanidade - Sapiens, p. 428)*

Resumo

O desenvolvimento de aplicativos tem se tornado cada vez mais frequente. Conforme a crescente demanda, aplicativos *web* nem sempre conseguem suprir as necessidades do mercado. Não apenas desenvolvedores de outras linguagens tem se aventurado pelo mundo *mobile*, mas entusiastas - pessoas sem conhecimento formal da área da Ciência da Computação ou Engenharia de Software - passaram a desenvolver aplicativos. Um problema que estes encontram é relacionado ao acesso e entendimento de materiais de apoio, enquanto o desenvolvedor ainda não adquiriu experiência. Este trabalho se propôs a construir uma abordagem de desenvolvimento simplificada, para apoiar desenvolvedores que ainda não possuem experiência em desenvolvimento ágil e *mobile* - nos seus primeiros passos para a criação de aplicativos voltados para o sistema operacional iOS. O trabalho consistiu de duas fases, sendo que o objetivo da primeira fase foi realizar um mapeamento inicial de literatura, onde, a partir deste buscou-se encontrar as atividades consideradas ideais para guiar o desenvolvimento de um aplicativo por alguém sem experiência, definindo-se uma abordagem preliminar. Na segunda fase, foi realizada a validação da abordagem preliminar, por meio do desenvolvimento de aplicativos - utilizando a linguagem Swift, dentro de ciclos de pesquisa-ação. Após a execução destes ciclos, foi criada a versão final da abordagem.

Palavras-chaves: Abordagem - *Mobile* - Scrum - *Challenge Based Learning* - *Lean Thinking*

Abstract

App development has become more frequent. Accordingly to its crescent demand, web apps not always can achieve market's needs. Not only developers that are used to develop in other languages have been adventuring through the mobile world, but enthusiasts - people without formal knowledge in Computer Science or Software Engineering - started to develop apps. A problem found by those is related to support materials access and understanding, while as a developer, the person has not acquired experience yet. This project proposed to build a simplified development approach, to support developers - which do not have experience in agile and mobile development - in its first steps towards the creation of iOS apps. The project consisted in two phases, being the goal of the first one to create a initial literature map, from which started the search for the ideal activities for guiding someone without experience into the development of an app. From this activities a preliminar approach was defined. In the second phase, the preliminar approach was validated, through the development of apps - using Swift, during action research cycles. After the execution of those cycles, the final version of the approach was created.

Key-words: Approach - Mobile - Scrum - Challenge Based Learning - Lean Thinking

Lista de ilustrações

Figura 1 – Diagrama de causa e efeito (<i>fishbone</i>)	25
Figura 2 – Diagrama de Venn com a localização da abordagem	29
Figura 3 – Processo do <i>framework</i> Scrum	31
Figura 4 – Representação de uma <i>sprint</i>	33
Figura 5 – Representação de um <i>Backlog</i> do produto	34
Figura 6 – Processo do <i>Challenge Based Learning</i>	37
Figura 7 – Subprocesso - <i>Engage</i>	38
Figura 8 – Subprocesso - <i>Investigate</i>	39
Figura 9 – Subprocesso - <i>Act</i>	40
Figura 10 – Tela com o uso da ferramenta Bizagi Modeler	43
Figura 11 – Tela com o uso da ferramenta Reminders, em sua versão <i>Desktop</i>	44
Figura 12 – Tela com o uso da ferramenta XMind	45
Figura 13 – Tela com o uso da ferramenta Xcode	47
Figura 14 – Tela com o uso da ferramenta Atom	48
Figura 15 – Tela com o uso do serviço GitHub	49
Figura 16 – Seleção metodológica	52
Figura 17 – Processo de condução do TCC	53
Figura 18 – Estrutura da pesquisa-ação	57
Figura 19 – Cronograma para o TCC1	59
Figura 20 – Cronograma para o TCC2	60
Figura 21 – Fluxo da abordagem preliminar proposta	63
Figura 22 – Exemplo de comparação entre o protótipo e o aplicativo desenvolvido	66
Figura 23 – Cronograma de validação	72
Figura 24 – Atividades no primeiro ciclo	74
Figura 25 – Relevância das atividades para os desenvolvedores no primeiro ciclo	75
Figura 26 – Fluxo da segunda versão da abordagem	77
Figura 27 – Atividades no segundo ciclo	78
Figura 28 – Relevância das atividades para os desenvolvedores no segundo ciclo	78
Figura 29 – Evolução das atividades ao longo do trabalho	80
Figura 30 – Fluxo da abordagem CBAMD em sua versão final	81
Figura 31 – Nomes dos artigos encontrados	97
Figura 32 – Relação das expressões de busca com os artigos encontrados	98
Figura 33 – Comparação entre o protótipo e o aplicativo desenvolvido	105
Figura 34 – Comparação entre o protótipo e o aplicativo desenvolvido	108
Figura 35 – Comparação entre o protótipo e o aplicativo desenvolvido	112
Figura 36 – Comparação entre o protótipo e o aplicativo desenvolvido	115

Figura 37 – Comparação entre o protótipo e o aplicativo desenvolvido 116

Figura 38 – Comparação entre o protótipo e o aplicativo desenvolvido 119

Lista de abreviaturas e siglas

<i>ACOT²</i>	<i>Apple Classrooms of Tomorrow Today</i>
<i>BPMN</i>	<i>Business Process Modeling Notation</i>
<i>CBAMD</i>	<i>Cambraia's Beginners Approach for Mobile Development</i>
<i>CBL</i>	<i>Challenge Based Learning</i>
<i>CRUD</i>	<i>Create, Read, Update and Delete</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>XP</i>	<i>Extreme Programming</i>
<i>FDD</i>	<i>Feature Driven Development</i>
<i>PO</i>	<i>Product Owner</i>
<i>SM</i>	<i>Scrum Master</i>

Sumário

1	INTRODUÇÃO	23
1.1	Contextualização	23
1.2	Problemática	23
1.3	Justificativa	24
1.3.1	Revisão sistemática	25
1.4	Questão de pesquisa	25
1.5	Objetivos	26
1.5.1	Objetivo geral	26
1.5.2	Objetivos específicos	26
1.6	Organização do trabalho	26
2	REFERENCIAL TEÓRICO	29
2.1	Metodologias ágeis	29
2.1.1	Scrum	30
2.1.1.1	Time Scrum	32
2.1.1.2	Eventos Scrum	32
2.1.1.3	Artefatos Scrum	34
2.2	<i>Lean Thinking</i>	35
2.2.1	Comparativo entre filosofias	36
2.3	<i>Challenge Based Learning</i>	36
2.3.1	Fases	36
2.3.1.1	<i>Engage</i>	37
2.3.1.2	<i>Investigate</i>	38
2.3.1.3	<i>Act</i>	38
2.3.2	Entregáveis	39
2.4	Simplicidade e educação	40
2.5	Considerações parciais	41
3	SUORTE TECNOLÓGICO	43
3.1	Desenho da abordagem	43
3.2	Engenharia de software	44
3.2.1	Gestão do projeto	44
3.2.1.1	Reminders	44
3.2.1.2	XMind	45
3.2.2	Desenvolvimento de software	45
3.2.2.1	Swift	45

3.2.2.2	Xcode	46
3.2.2.3	Atom	46
3.2.3	Gestão de configuração de software	46
3.2.3.1	Git	47
3.2.3.2	GitHub	47
3.2.4	Pesquisa	48
3.2.4.1	LaTeX	48
3.2.4.2	Zotero	48
3.3	Considerações parciais	49
4	METODOLOGIA	51
4.1	Pesquisa	51
4.2	Condução do TCC	52
4.3	Avaliação dos resultados	57
4.3.1	Pesquisa-ação	57
4.3.2	Validação	57
4.3.2.1	Participantes	58
4.3.2.2	Coleta de dados	58
4.3.2.3	Propostas de mudanças	59
4.4	Cronograma	59
4.5	Considerações parciais	59
5	ABORDAGEM PRELIMINAR	61
5.1	Proposta preliminar	61
5.2	Estrutura das atividades	62
5.2.1	Descrição das atividades	62
5.3	Considerações parciais	69
6	RESULTADOS E ABORDAGEM FINAL	71
6.1	Execução por ciclos	71
6.1.1	Ciclo 1	71
6.1.2	Ciclo 2	73
6.2	Análise por ciclos	74
6.2.1	Ciclo 1	74
6.2.2	Ciclo 2	76
6.3	Atividades da abordagem CBAMD final	79
6.3.1	Descrição das atividades	79
6.4	Considerações parciais	85
7	CONSIDERAÇÕES FINAIS	87
7.1	Trabalhos futuros	88

	REFERÊNCIAS	89
	APÊNDICES	93
	APÊNDICE A – REVISÃO SISTEMÁTICA	95
A.1	Protocolo de Revisão Sistemática	95
A.1.1	Perguntas de pesquisa	95
A.1.2	Palavras-chave	95
A.1.3	Expressão de busca	95
A.1.4	Base	96
A.1.5	Seleção de artigos	96
A.1.5.1	Processo de análise de artigos	96
A.1.5.2	Inclusão	96
A.1.5.3	Exclusão	97
A.1.6	Avaliação da qualidade	97
A.2	Expressões de busca e artigos encontrados	97
	APÊNDICE B – TERMO DE CONSENTIMENTO	99
	APÊNDICE C – APLICAÇÃO DA ABORDAGEM	101
C.1	Aplicativo 1 - LetsDraw	101
C.2	Aplicativo 2 - DELStar	105
C.3	Aplicativo 3 - LetsSee	109
C.4	Aplicativo 4 - MeuQR	112
C.5	Aplicativo 5 - 3D Scale	115

1 Introdução

Neste capítulo, é introduzido o trabalho de conclusão de curso. O capítulo está dividido em seções, que melhor detalham o que é apresentado, estando dispostas da seguinte maneira: na seção 1.1, quanto à **contextualização**; na seção 1.2, quanto à **problemática**; na seção 1.3, quanto à **justificativa**; na seção 1.4, quanto à **questão de pesquisa**, bem como aos **objetivos**, na seção 1.5. Por fim, na seção 1.6, é apresentada a **organização dessa monografia**.

1.1 Contextualização

O uso de dispositivos móveis tem crescido com consideráveis proporções e vem tornando-se tendência para as mais diversas atividades, conforme acordado em Santos et al. (2016). Se antes o acesso à informação era mais complexo, o uso de dispositivos móveis facilitou isso e, por sua vez, contribuiu para o crescimento do número de usuários em diversos serviços (SANTOS et al., 2016).

Para suprir essa nova demanda, apenas aplicativos *web* adaptados para navegadores móveis, nem sempre são suficientes. Tornam-se, portanto, necessárias soluções nativas para cada plataforma, os aplicativos. Porém, os aplicativos possuem um nível maior de dificuldade de desenvolvimento em relação a outros tipos de aplicações, como aplicativos *web*, de acordo com Xanthopoulos e Xinogalos (2013).

Ao iniciar o desenvolvimento de aplicativos, é necessária a análise de diversos aspectos do projeto para definir pontos técnicos, como as melhores plataformas e metodologias de desenvolvimento (KALEEL; HARISHANKAR, 2013). Com base em tais pontos, nota-se que é importante o contínuo aprimoramento do modo que aplicativos são desenvolvidos, levando em conta não apenas fatores técnicos, mas também fatores ambientais, como o suporte dado para tal atividade.

1.2 Problemática

Um dos sistemas móveis com mais usuários nos últimos anos é o iOS¹. Trata-se de um sistema operacional utilizado tanto em *smartphones* como em *tablets*.

Conforme (SANTOS et al., 2016), dentre os problemas encontrados no desenvolvimento *mobile*, encontram-se os diferentes tipos de plataformas de desenvolvimento, tendo cada uma seu conjunto específico de hardware e software.

¹ <http://gs.statcounter.com/os-market-share/>. Acesso em: 10/06/2017.

Graças à popularidade do desenvolvimento *mobile* e ao fácil acesso ao mercado, muitas pessoas, sem o conhecimento formal no campo da Ciência da Computação ou Engenharia de Software, dedicam-se ao desenvolvimento de aplicativos para dispositivos móveis, de acordo com Santos et al. (2015).

Apesar de metodologias terem sido criadas com o intuito de facilitar esse desenvolvimento, nem sempre essas metodologias se traduzem em algo fácil para o desenvolvedor que está iniciando seu aprendizado.

Dentro dessa problemática, foram analisadas causas para o problema que se procura uma solução, sendo essas:

- Escassez de materiais de apoio (WELLS, 2015);
- Processo de desenvolvimento indefinido (KALEEL; HARISHANKAR, 2013);
- Quando existente, dificuldade no entendimento do material de apoio (WELLS, 2015) (REBOUÇAS et al., 2016).

Com o intuito de facilitar o entendimento do problema, juntamente com suas causas, representou-se esses aspectos em um diagrama de causa e efeito, também conhecido como *fishbone*, conforme visto na Figura 1.

Para uma descrição mais clara e formal do problema encontrado, em conformidade com Joorabchi, Mesbah e Kruchten (2013), foi utilizada a técnica de declaração do problema, apresentada por IIBA (2009), conforme o quadro a seguir:

O problema: Necessidade de experiência no desenvolvimento iOS;

Afeta: Desenvolvedores que estão iniciando seu aprendizado;

Cujo impacto é: Gasto desnecessário de tempo;

Uma solução bem sucedida seria: A definição de uma abordagem de desenvolvimento simples e sucinta.

1.3 Justificativa

Procurando colaborar com essa problemática, o presente trabalho propôs a construção de uma abordagem de desenvolvimento simplificada, para apoiar desenvolvedores - que ainda não possuem experiência em desenvolvimento *mobile* - nos seus primeiros passos para a criação de aplicativos voltados para o sistema operacional iOS.

A proposta para criação dessa abordagem, inicialmente, apoiava-se em dois pilares: o *Framework* de aprendizado *Challenge Based Learning* (CBL), por Nichols, Cator e Torres (2016) e a metodologia ágil Scrum, por Sharma e Hasteer (2016). Posteriormente, a busca

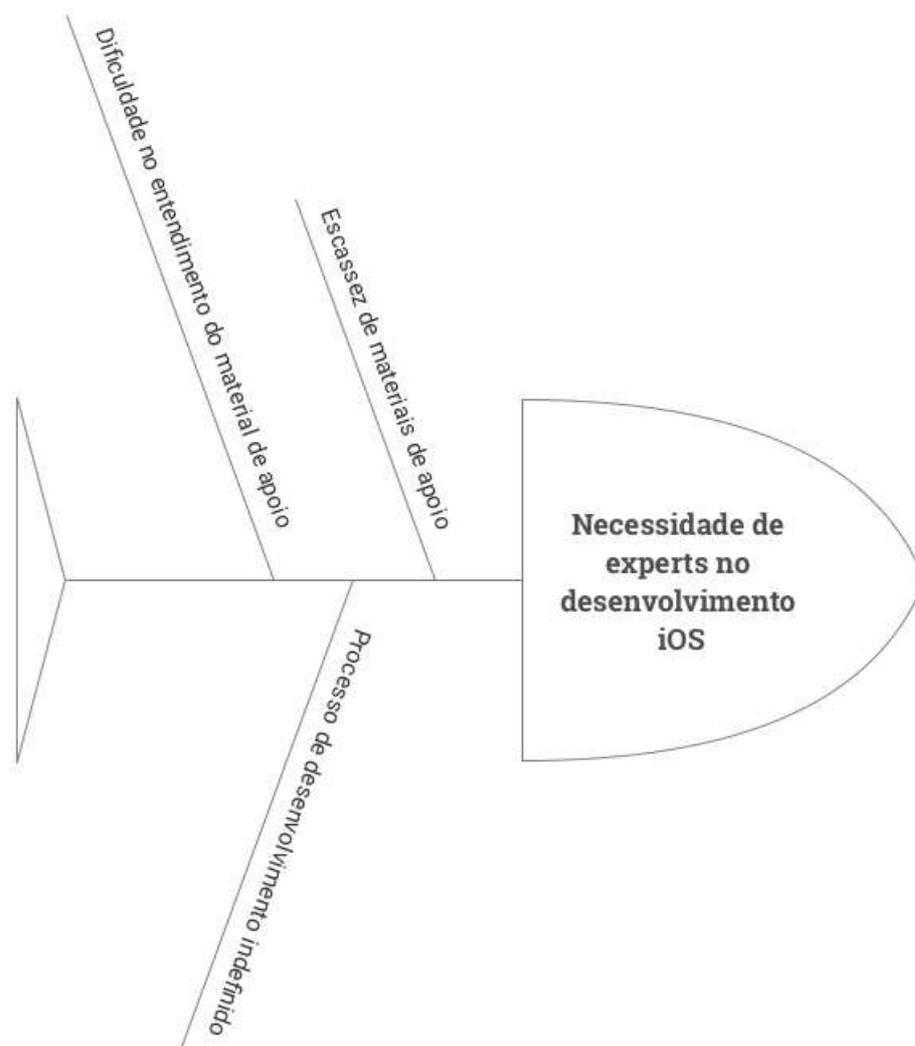


Figura 1 – Diagrama de causa e efeito (*fishbone*). Fonte: Autor.

pela simplicidade no desenvolvimento e aspectos do *Lean Thinking* também foram adicionados.

1.3.1 Revisão sistemática

Para averiguar a existência de estudos e propostas similares à proposta inicial, foi conduzida uma revisão sistemática. Esse tipo de levantamento bibliográfico foi escolhido, principalmente, por conta de seu rigor de pesquisa e possibilidade de replicação.

Para sua execução, foi criado um protocolo de revisão baseado nas pesquisas de Petersen, Vakkalanka e Kuzniarz (2015), Kitchenham (2004) e Dieste e Padua (2007). O protocolo, fruto da adaptação das pesquisas mencionadas, possui seus detalhes no Apêndice A.

1.4 Questão de pesquisa

Tendo como objetivo guiar o trabalho, foi definida a seguinte questão de pesquisa:

Como a aplicação conjunta de conceitos oriundos do Scrum, Lean Thinking e CBL pode contribuir na criação de uma abordagem capaz de apoiar iniciantes no desenvolvimento de aplicativos para dispositivos móveis iOS?

1.5 Objetivos

Este trabalho possuiu um objetivo geral, que por sua vez foi modularizado em objetivos específicos para uma condução mais apropriada do trabalho.

1.5.1 Objetivo geral

Elaboração de uma abordagem capaz de apoiar iniciantes no desenvolvimento de aplicativos para dispositivos móveis iOS, a qual será orientada por Scrum, *Lean Thinking* e CBL.

1.5.2 Objetivos específicos

- Investigar o que já existe sobre a utilização conjunta do CBL e Scrum;
- Definir, por meio de revisão bibliográfica, aspectos do CBL a serem utilizados no desenvolvimento da abordagem;
- Definir, por meio de revisão bibliográfica, aspectos do Scrum a serem utilizados no desenvolvimento da abordagem;
- Definir um conjunto de atividades, por meio de ciclos de pesquisa-ação, que, em conjunto, serão caracterizadas como a abordagem proposta;
- Implementação, de um ou mais aplicativos, pelo Autor e/ou terceiros, orientando-se pela abordagem proposta;
- Avaliação dos resultados coletados durante a implementação.

1.6 Organização do trabalho

Este trabalho está organizado da seguinte forma, estando as seções na ordem citada: **Referencial teórico**, no qual, cada aspecto teórico é previamente abordado visando colaborar com o entendimento da abordagem como um todo; **Suporte tecnológico**, no qual, é explicada cada tecnologia utilizada no decorrer do trabalho; **Metodologia**, na

qual, são detalhadas as metodologias de pesquisa, desenvolvimento e avaliação dos resultados utilizadas no decorrer do trabalho; **Abordagem preliminar**, na qual, é apresentada a estrutura das atividades e a abordagem preliminar; **Resultados e abordagem final**, nos quais, são apresentadas a execução e análise dos ciclos, junto a abordagem proposta neste trabalho e; **Considerações finais**, sendo apresentada a conclusão deste trabalho e possíveis trabalhos futuros.

2 Referencial teórico

Neste capítulo, é apresentado o embasamento teórico utilizado para o desenvolvimento da abordagem. O capítulo está dividido em seções, estando dispostas da seguinte maneira: na seção 2.1, são discutidas as **metodologias ágeis**, apresentando com mais detalhes o *framework* Scrum; na seção 2.2, é apresentado o **Lean Thinking**, junto a seus conceitos; na seção 2.3, é apresentado o *framework* de aprendizado **CBL**; na seção 2.4 é discutida a **simplicidade e educação** no desenvolvimento *mobile*.

A localização da abordagem no referencial teórico, pode ser vista no diagrama de Venn, conforme orientações de Ruskey e Weston (1997), apresentado na Figura 2.

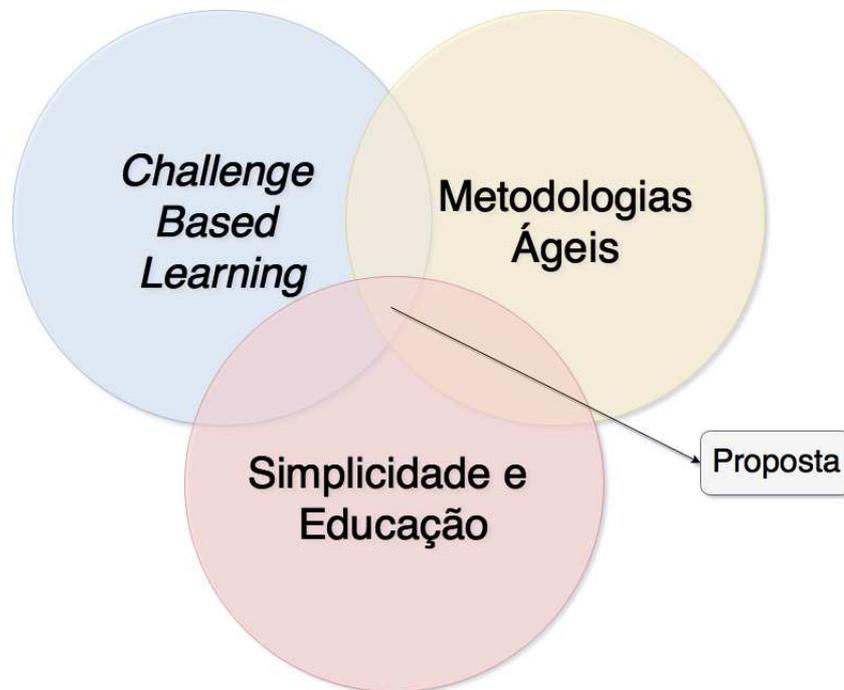


Figura 2 – Diagrama de Venn localizando a abordagem dentro do referencial teórico.
Fonte: Autor

2.1 Metodologias ágeis

De acordo com Bourque, Fairley et al. (2014), metodologias (ou métodos) ágeis são assim consideradas uma vez que são caracterizadas por ciclos de desenvolvimento curtos e iterativos, times autogerenciáveis, envolvimento maior do cliente, ênfase na criação de um entregável a cada ciclo de desenvolvimento e outras características similares.

Desde o Manifesto Ágil¹, as metodologias ágeis tornaram-se tão populares que a

¹ <http://agilemanifesto.org>

indústria de software procura adotar essa abordagem em suas organizações, visando obter benefícios (SHARMA; HASTEER, 2016).

Dentre as metodologias ágeis, é possível notar algumas que se destacaram, tendo sido mais utilizadas pela indústria, como o *Extreme Programming* (XP), Scrum e *Feature Driven Development* (FDD) (AWAD, 2005).

Conforme Awad (2005), o processo do XP pode ser caracterizado por curtos ciclos de desenvolvimento, planejamento incremental, *feedback* contínuo, dependência de comunicação e *design* evolutivo. Deste modo, desenvolvedores XP respondem a mudanças no ambiente com muito mais coragem.

O Scrum é processo iterativo e incremental para o desenvolvimento de qualquer produto ou gestão de qualquer trabalho. Este se concentra no modo em como os membros do time deveriam funcionar para que sejam flexíveis em um ambiente com mudanças. O objetivo é que, ao fim de cada iteração, seja produzido algo com o potencial de uma funcionalidade (AWAD, 2005).

De acordo com Awad (2005), o FDD, diferente das metodologias já mencionadas, não cobre todo o processo de desenvolvimento de software, mas possui o seu foco nas fases de desenho e construção. É composto pelo desenvolvimento de um modelo, construção de uma lista de funcionalidades e planejamento por funcionalidade, tudo ao início do projeto. Após isso, é realizado o desenho e a construção, por funcionalidade e de forma iterativa.

Para este trabalho, o Scrum foi escolhido como a metodologia ágil que serviu de fundação para a abordagem - o que não impediu a possibilidade de influência das outras metodologias mencionadas. A escolha se deu por conta da abrangência do Scrum ao lidar com não somente o desenvolvimento de software, mas o cuidado que este possui para com quem o desenvolve.

2.1.1 Scrum

Como é mencionado por Sharma e Hasteer (2016), o Scrum é um *framework* leve, de simples entendimento e capaz de gerir o desenvolvimento de produtos complexos, dado determinado prazo de entrega.

A entrega de software é incremental, por meio de uma série de iterações, que possuem tempo definido, chamadas *sprints*.

Conforme é possível visualizar na Figura 3, foi criado um diagrama explicativo do *framework*, que aliado ao restante desta subseção, deixa claro o seu funcionamento como um todo; incluindo seus integrantes, os seus eventos e seus artefatos.

Uma das principais vantagens do Scrum para este trabalho é que este não necessita de prática alguma ou método de desenvolvimento específico, nem ao menos provê isso

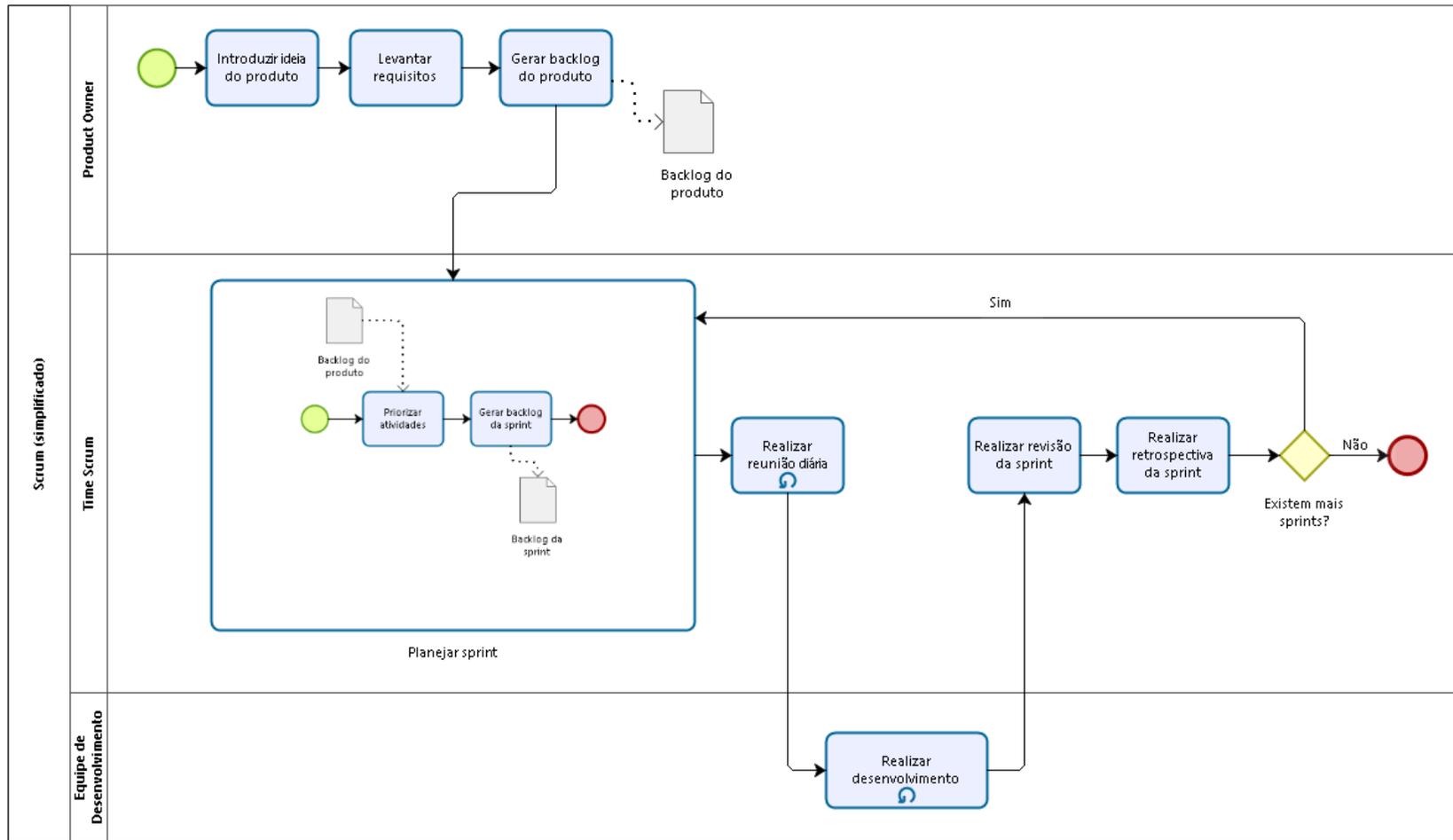


Figura 3 – Processo do *framework* Scrum. Fonte: Autor, baseado em Sharma e Hasteer (2016).

para seu usuário. Ao invés disso, requer algumas práticas relativas à gestão e o apoio de ferramentas em suas diferentes fases, para assim evitar complexidade (RISING; JANOFF, 2000).

O *framework* consiste em um Time Scrum, suas atividades associadas, artefatos e regras. A seguir, o time, eventos e artefatos do Scrum são apresentados com maiores detalhes, em acordo com Sharma e Hasteer (2016) e Schwaber e Sutherland (2016).

2.1.1.1 Time Scrum

O Time Scrum é composto pelo *Product Owner* (PO), *Scrum Master* (SM) e pela equipe de desenvolvimento.

Times Scrum devem ser auto-organizáveis e multifuncionais, sendo que esse modelo foi projetado para aperfeiçoar a criatividade, flexibilidade e, especialmente, a produtividade.

O PO possui a responsabilidade de formular o planejamento conforme o problema em mãos e dividi-lo para a equipe de desenvolvimento, procurando, assim, melhores resultados. Deve maximizar o valor do produto e do trabalho da equipe de desenvolvimento.

O SM é a pessoa que lidera a equipe de desenvolvimento, além de oferecer supervisão, orientação, ajuda, resolução de problemas e demais ajustes necessários. Uma de suas responsabilidades é a condução das reuniões diárias para a obtenção do *status* atual do projeto. Realiza ainda a interação entre o PO e a equipe de desenvolvimento, sendo responsável por garantir que o Scrum seja aplicado.

A equipe de desenvolvimento é um grupo de desenvolvedores qualificados que ajuda atingir os objetivos do projeto e entregar o produto no fim de cada *sprint*. Estes são estruturados e autorizados pela organização para organizar e gerenciar o seu próprio trabalho. Vale notar que uma equipe de desenvolvimento deve ser pequena o suficiente para se manter ágil, e grande o suficiente para desenvolver o que é necessário dentro de cada *sprint*.

2.1.1.2 Eventos Scrum

A *sprint* é um período de tempo, de um mês ou menos, no qual é criada uma versão incremental do produto. Durante uma *sprint*, não são feitas mudanças que possam colocar em risco seu objetivo; metas de qualidade não diminuem, e o escopo pode ser negociado. Isso ocorre para que a definição do que será construído não mude, e a complexidade bem como o risco não aumentem desnecessariamente.

A Figura 4 possui uma representação visual de uma *sprint*.

O trabalho a ser desenvolvido deve ser planejado na reunião de planejamento

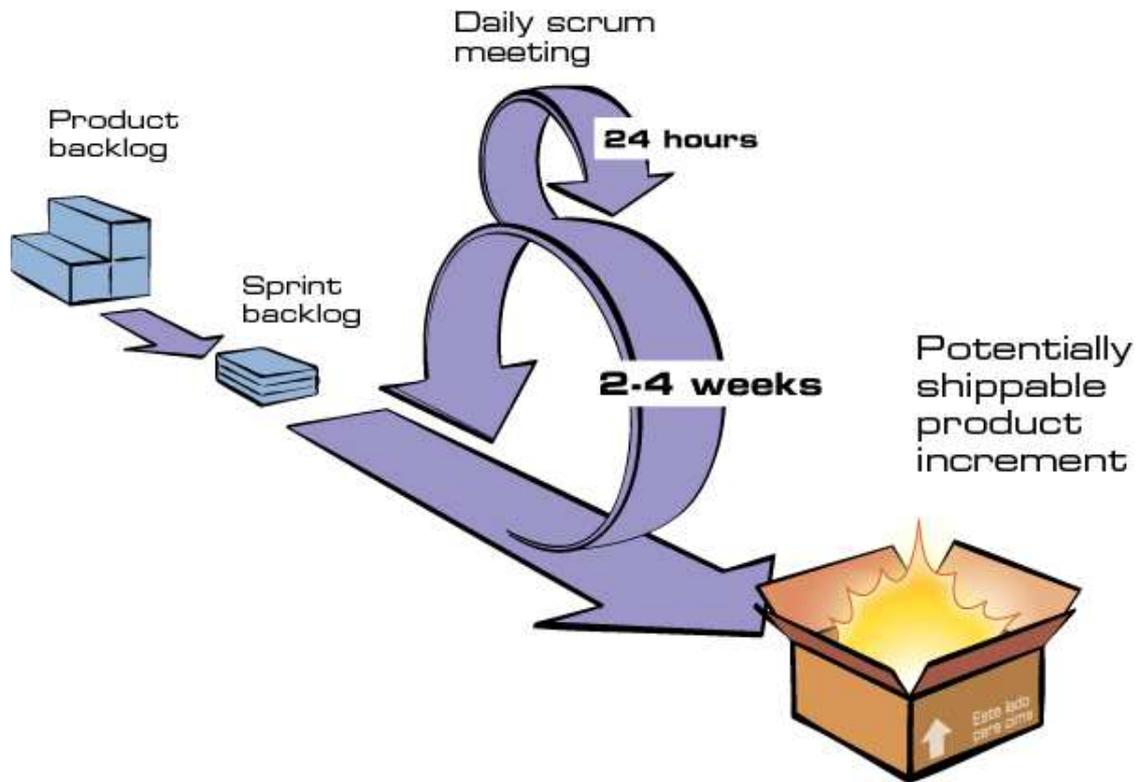


Figura 4 – Representação de uma *sprint*. Fonte: Scrum... (2017).

da *sprint*, sendo um plano colaborativo criado por todo o Time Scrum, não passando de oito horas (para casos de duração maior). Tal plano colaborativo deve responder, necessariamente:

- Qual é o objetivo da *sprint*?
- O que pode ser entregue como resultado do incremento da próxima *sprint*?
- Como o trabalho necessário para entregar o incremento será realizado?

Outro evento muito importante são as reuniões diárias, estas possuem prazo máximo de 15 (quinze) minutos, período no qual a equipe de desenvolvimento deve sincronizar suas atividades realizadas no último dia e criar o planejamento para o próximo dia.

Durante as reuniões, as seguintes perguntas devem ser esclarecidas:

- O que eu fiz ontem que ajudou a equipe de desenvolvimento a atender a meta da *sprint*?
- O que eu farei hoje para ajudar a equipe de desenvolvimento a atender a meta da *sprint*?
- Eu vejo algum obstáculo que impeça a mim ou a equipe de desenvolvimento no atendimento da meta da *sprint*?

Ao final de cada *sprint* é realizada uma revisão para inspecionar o incremento produzido, sendo que nesta reunião, as partes interessadas colaboram sobre o que foi feito na *sprint*. Isso permite identificar pontos que podem ser melhorados para otimizar valor.

Após a revisão, ocorre a retrospectiva da *sprint*, que possui como foco as pessoas, os relacionamentos, os processos e as ferramentas. Seu intuito é identificar melhorias no modo que o Time Scrum trabalha.

2.1.1.3 Artefatos Scrum

O *Backlog* do produto é uma lista ordenada, por prioridade, de tudo o que deve ser necessário no produto. É responsabilidade do PO o seu conteúdo, disponibilidade e ordenação. Seus itens devem possuir atributos de descrição, ordem estimativa e valor.

A Figura 5 possui uma representação visual de um *backlog* do produto.

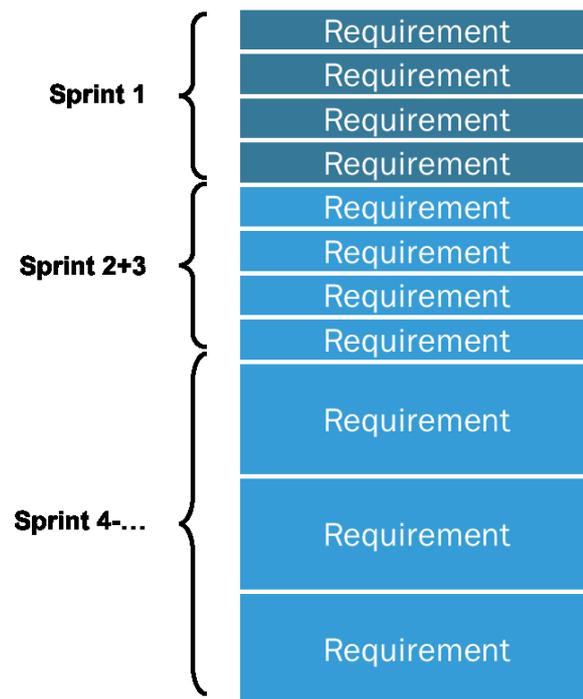


Figura 5 – Representação de um *backlog* do produto. Fonte: (WHAT..., 2017).

Para manter a agilidade, apenas um conjunto de itens específicos do *backlog* do produto é selecionado por *sprint*, formando assim o *backlog* da *sprint*. É composto dos itens que a equipe de desenvolvimento previu que estará desenvolvido naquela *sprint*.

Um fator fundamental para o bom funcionamento do Scrum é a "Definição de Pronto". Simplificadamente, essa definição diz que todo o Time Scrum deve entender o que "Pronto" significa, de forma que seja possível garantir quando o trabalho está realmente completo.

2.2 Lean Thinking

O *Lean Thinking*, é um conjunto de princípios que foram testados ao longo dos anos na manufatura e que, posteriormente, foram traduzidos para o desenvolvimento de software, dando origem ao *Lean Software Development* (POPPENDIECK; POPPENDIECK, 2003).

Tendo seu início com Henry Ford, em 1913, como uso de uma linha de produção, o *Lean Thinking* permitiu a produção em massa de veículos, porém, ainda não possibilitava variedade na criação. Ao perceber a necessidade do mercado, Kiichiro Toyota, junto com outros, revisitaram o pensamento inicial de Ford, e inventaram o sistema de produção Toyota. Esse sistema consistiu na mudança do tamanho das máquinas para o tamanho essencialmente necessário; máquinas que se monitoram; realocação para formar um processo sequencial; configurações rápidas para as máquinas, e a comunicação entre os processos no que diz respeito a necessidade de materiais. Como fruto de tais mudanças, seria possível obter baixo custo, alta variedade e alta qualidade, além de permitir mudanças de acordo com as necessidades do mercado (LEI, 2017).

O *Lean Thinking* não é uma metodologia de desenvolvimento ou gestão em si, mas oferece princípios que aplicados melhoram o desenvolvimento de software (AWAD, 2005).

De acordo com Poppendieck e Poppendieck (2003), os princípios do *Lean Software Development* são:

- Elimine o desnecessário: tudo o que não agrega valor ao produto, por exemplo, funcionalidades duplicadas e/ou desnecessárias em uma aplicação, deve ser eliminado. Valor este o percebido pelo usuário;
- Aumente o aprendizado: tendo em vista que o desenvolvimento é um exercício, este deve ser concebido como um processo de aprendizado com a adição do desafio de lidar com equipes e com o produto final sendo algo complexo;
- Decida o mais tardiamente: o atraso nas decisões tem seu valor no fato de que decisões tomadas mais tardiamente podem ser tomadas com base em fatos e não em especulações;
- Entregue o mais rapidamente: com ciclos de entregas mais rápidos, decisões podem ser tomadas "mais tardiamente", sem prejuízo para o produto, e mais pode ser aprendido a cada ciclo;
- Empodere a equipe: ao invés de apenas obedecer decisões, equipes equipadas com a *expertise* necessária e com um bom apoio de um líder, tomarão melhores decisões técnicas;

- Mantenha a integridade: um software é íntegro quando possui uma arquitetura coerente, é extensível, manutenível e adaptável, além de atingir plenamente o seu objetivo;
- Veja o todo: é necessária uma visão do todo, ao invés de apenas ter uma visão individual e "curta", para que o desempenho do produto seja maximizado.

2.2.1 Comparativo entre filosofias

Dadas suas características inovadoras, é natural que surjam comparações entre as metodologias ágeis e o *Lean Thinking*. Nesse tópico, Nedre (2015) traz algumas comparações:

- Desenvolvimento inteligente (Lean) x (Ágil) Execução mais rápida de tarefas;
- Transforma o processo sustentável (Lean) x (Ágil) Flexibiliza o processo;
- Surgiu na manufatura e se expandiu (Lean) x (Ágil) Iniciou-se em software e depois se expandiu;
- Ciclo: Construir - Medir - Aprender (Lean) x (Ágil) Ciclo: *Backlog* do produto-*Backlog* da *sprint* - *Sprints*;
- Demonstração de progresso por aprendizagem validada (Lean) x (Ágil) Demonstração de progresso por definição de pronto;
- Metodologias: Kanban, Kaizen, entre outras (Lean) x (Ágil) Metodologias: Scrum, XP, entre outras.

Ainda existem outros pontos passíveis de comparação. Entretanto, pode-se perceber que ao invés de se contrariarem, ambas as filosofias se completam ou se sobrepõem, tendo em vista suas similaridades.

Nedre (2015) ainda coloca as metodologias ágeis dentro da filosofia do *Lean Thinking*, já que o termo é mais amplo, envolvendo não apenas o fator do tempo, como algumas metodologias ágeis o fazem; mas também outros fatores como energia, trabalho e capital.

2.3 *Challenge Based Learning*

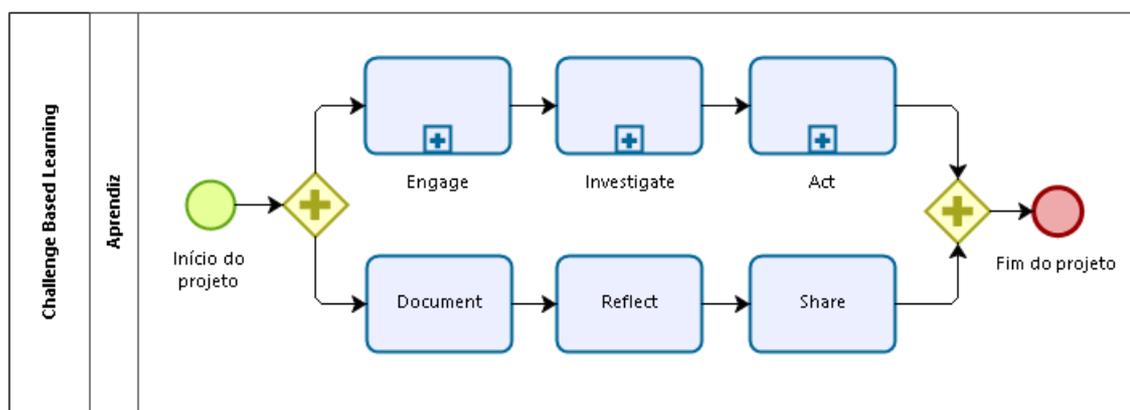
O *Challenge Based Learning*, também conhecido como CBL, é um *framework* voltado para o aprendizado que teve início no projeto *Apple Classrooms of Tomorrow Today (ACOT²)*, em março de 2008. Esse possuía como objetivo a permanência dos alunos nas escolas (NICHOLS; CATOR; TORRES, 2016).

2.3.1 Fases

Atualmente, conforme Nichols, Cator e Torres (2016), o CBL é dividido em três fases interconectadas: **Engage**, **Investigate** e **Act**.

Paralelamente, ao longo do CBL, existe um constante processo de **documentação (Document)**, **reflexão (Reflect)** e **compartilhamento (Share)**.²

Para melhor visualização das fases do CBL, foi criado um diagrama explicativo, conforme é possível visualizar na Figura 6. Seus subprocessos podem ser vistos nas suas respectivas subseções.



Powered by
bizagi
Modeler

Figura 6 – Processo do *Challenge Based Learning*. Fonte: Autor, baseado em (NICHOLS; CATOR; TORRES, 2016).

2.3.1.1 Engage

Por meio de um processo de Questionamento Essencial, nas mãos do Aprendiz³, uma Grande Ideia abstrata torna-se um Desafio concreto e realizável.

1. Grandes Ideias são conceitos amplos, explorados em diversos aspectos e relevantes para o Aprendiz e para a comunidade.
2. O Questionamento Essencial permite que o Aprendiz contextualize e personalize a Grande Ideia, possuindo como produto final uma única Questão Essencial que é relevante para o indivíduo ou grupo.

² Os artefatos apresentados possuem *templates* para download, disponíveis em <<http://cbl.digitalpromise.org/toolkit/>>. Acesso em 24/10/2017.

³ O usuário do *framework* aqui é chamado de Aprendiz.

- Já os Desafios transformam as Questões Essenciais em chamados para ação por participantes que irão aprender acerca do assunto e desenvolver uma Solução. Estes devem ser imediatos e realizáveis.

O subprocesso *Engage* pode ser melhor visualizado na Figura 7.

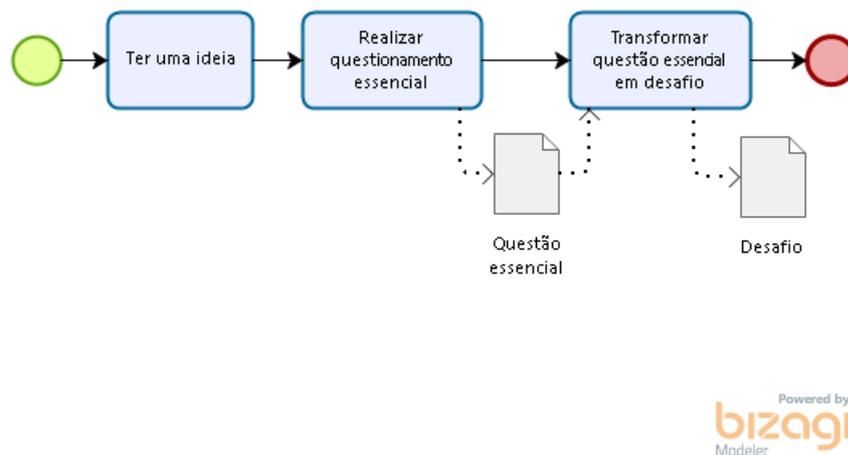


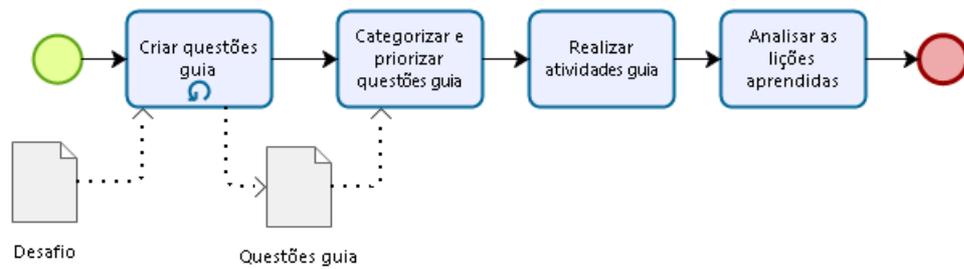
Figura 7 – Subprocesso - *Engage*. Fonte: Autor, baseado em Nichols, Cator e Torres (2016).

2.3.1.2 *Investigate*

Seguindo o que a fase anterior dita, o Aprendiz deve aprender acerca do assunto em questão.

- Questões Guia apontam na direção do conhecimento que o Aprendiz irá precisar para desenvolver uma Solução para o Desafio. Categorizar e priorizar as questões criam uma experiência de aprendizado organizada, e de tal modo Questões Guia continuarão a emergir durante o processo.
- Atividades Guia e Recursos são utilizadas para responder as Questões Guia elaboradas pelo Aprendiz. Tais atividades e recursos incluem todos os métodos disponíveis para o Aprendiz.
- Análise das lições aprendidas durante as Atividades Guias fornece uma fundação para uma eventual identificação de Soluções.

O subprocesso *Investigate* pode ser melhor visualizado na Figura 8.



Powered by
bizagi
 Modeler

Figura 8 – Subprocesso - *Investigate*. Fonte: Autor, baseado em Nichols, Cator e Torres (2016).

2.3.1.3 Act

Com base nas evidências disponíveis, Soluções são desenvolvidas, implementadas e, posteriormente, avaliadas com base nos resultados.

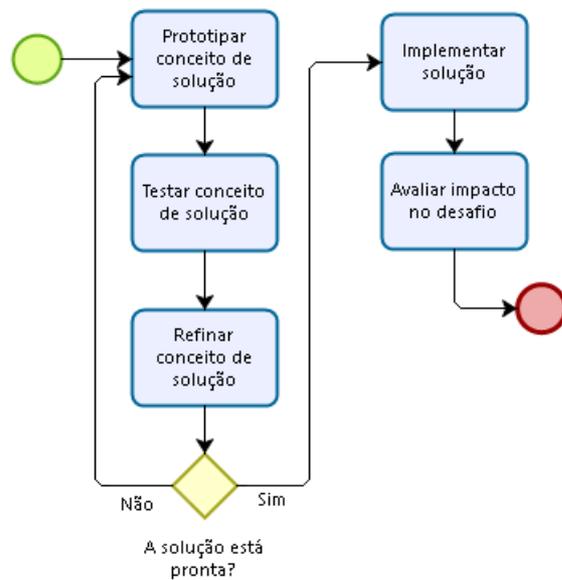
1. Conceitos de Soluções emergem do que foi descoberto durante a fase anterior. Utilizando um ciclo de *design*, o Aprendiz irá prototipar, testar e refinar os conceitos de Solução.
2. A implementação da solução toma lugar em um ambiente real, com um público-alvo real. A quantidade de tempo e recursos disponíveis irão guiar a profundidade que a implementação possuirá.
3. Uma avaliação fornece a oportunidade de avaliar a efetividade da Solução, realizar ajustes e aprofundar o conhecimento do assunto.

O subprocesso *Act* pode ser melhor visualizado na Figura 9.

2.3.2 Entregáveis

Nichols, Cator e Torres (2016) propõem entregáveis para auxiliar na execução do *framework*, sendo alguns destes:

- Questionário de Grandes Ideias - Identificação e pesquisa de Grandes Ideias, com posterior geração de relatórios ou apresentações;
- Proposta de Desafio - Produção de documentos que definem a Grande Ideia, a Questão Essencial, o Desafio e a explicação da importância do Desafio;



Powered by
bizagi
 Modeler

Figura 9 – Subprocesso - *Act*. Fonte: Autor, baseado em Nichols, Cator e Torres (2016).

- Questões Guia - Conjunto de questões que guiará a busca por uma solução;
- Proposta de Solução - Apresentações propondo uma solução que inclui protótipos, conceitos e *feedback* inicial.

2.4 Simplicidade e educação

Olhando um pouco mais para o lado da psicologia, têm-se algumas vertentes de raciocínio que trazem conceitos interessantes para a questão da necessidade de uma abordagem simples.

Dentre essas vertentes, o construtivismo de Piaget, conforme Becker (1992), acorda que nada está pronto e que o conhecimento não é dado, e sim que este deve ser construído. Até mesmo o sujeito humano é um objeto também a ser construído. Entretanto, essa construção não se dá de forma passiva, pois necessita de condições do sujeito, sejam estas: fisiológicas, como estar bem alimentado e ser sadio; condições de seu meio, como um ambiente que permita o acesso a conhecimentos já descobertos, como uma biblioteca e auxílio ferramental para estudos. Da mesma forma, a educação deve ser um processo de construção do conhecimento ao qual ocorrem problemas sociais vivenciados pelo indivíduo e o conhecimento já consolidado. Essa pode se dar por meio da apropriação da

prática, ao construir ou reconstruir as estruturas do pensar do sujeito, assim ampliando sua capacidade de compreensão (BECKER, 1992).

Para (PAPERT; HAREL, 1991), o construcionismo compartilha a conotação do aprendizado como uma construção de estruturas de conhecimento, vinda do construtivismo. Mas, além de compartilhar tal conotação, adiciona a ideia de que isso ocorre caso o sujeito esteja conscientemente engajado na criação deste conhecimento. O conhecimento possui um grande apelo para o contexto do sujeito, sendo importante salientar que, no construcionismo, a expressão dos sentimentos de cada indivíduo é fundamental, visto o impacto que este possui na educação e no auto-aprendizado.

De tal modo, fica evidente que o processo educativo é algo que deve ser desenvolvido, com as devidas ferramentas, orientações e ambiente propício para isso. Ao iniciar o desenvolvimento de software, é importante que a simplicidade seja um dos fatores relevantes para uma experiência de sucesso. Por isso, a simplicidade têm se tornado cada vez mais importante, especialmente no desenvolvimento *mobile*, uma vez que cada vez mais sujeitos, que não possuem instrução formal, ou seja, que ainda precisarão criar seu conhecimento, passam a desenvolver (SANTOS et al., 2015).

Diante do exposto, é fundamental que diretrizes apoiem desenvolvedores, principalmente iniciantes, possibilitando o rápido progresso, mesmo com a necessidade de adquirir novos conhecimentos técnicos (HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2012).

2.5 Considerações parciais

O capítulo abordou temas importantes para a aquisição de conhecimento necessários a este trabalho. Foram abordadas as metodologias ágeis, com foco no Scrum, tendo em vista a sua escolha como um dos principais referenciais deste trabalho; ao passo que houve um comparativo com o *Lean Thinking*. Também foi abordado o *framework* CBL, este que trouxe um olhar novo para a abordagem, no que diz respeito a coleta de conhecimento. Ao fim do capítulo, é trazida uma breve discussão sobre a simplicidade e a educação e o seu impacto no desenvolvimento de software.

3 Suporte tecnológico

Neste capítulo, são apresentadas as tecnologias que foram utilizadas no desenvolvimento deste trabalho. A seção 3.1, confere uma visão sobre a ferramenta que foi utilizada para o **desenho da abordagem**. Já a seção 3.2, apresenta as tecnologias que foram utilizadas nesse trabalho, no escopo da **Engenharia de Software**.

3.1 Desenho da abordagem

No desenho da abordagem proposta, foi utilizado o Bizagi Modeler ([BIZAGI, 2017](#)), em sua versão 3.1.0.011.

Conforme visto na Figura 10, trata-se de um software, gratuito, que possui como objetivo a criação e edição de processos na notação BPMN (*Business Process Modeling Notation*). A escolha dessa ferramenta foi feita tendo-se em vista sua facilidade de uso, como também sua capacidade de modelagem dos fluxos de atividades desejados. Além do uso mencionado, esse software também foi utilizado na criação e no mapeamento de processos auxiliares.

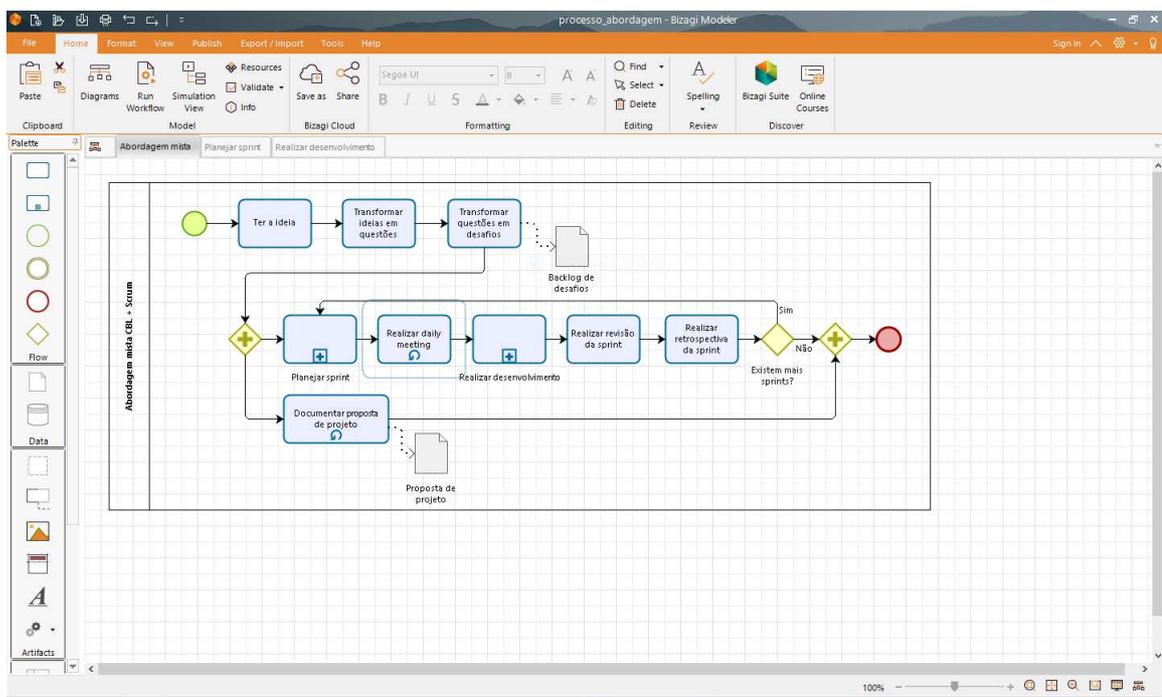


Figura 10 – Tela com o uso da ferramenta Bizagi Modeler.

3.2 Engenharia de software

As ferramentas mencionadas a seguir, da seção 3.2.1 à seção 3.2.4, visaram auxiliar na gestão do projeto, pesquisas para o desenvolvimento do trabalho e implementação das soluções propostas.

3.2.1 Gestão do projeto

Por ser um trabalho desenvolvido por apenas um autor, não foi necessária a utilização de muitas ferramentas para a gestão do projeto. Foram utilizados para este fim o Reminders (REMINDERS, 2017), em sua versão 5.0, e o XMind (XMIND, 2017), em sua versão 8.7.

3.2.1.1 Reminders

Com o objetivo de manter organizado todos os itens necessários para a completude do trabalho, foi utilizada a ferramenta Reminders. Refere-se a um aplicativo multiplataforma, presente no iOS, macOS e acessível via navegador *web*. Conforme visto na Figura 11, a principal vantagem do uso do Reminders é sua simplicidade, oferecendo a opção de criação de listas e atividades, ainda sendo possível detalhar as atividades conforme prioridade e adicionar datas para notificação.

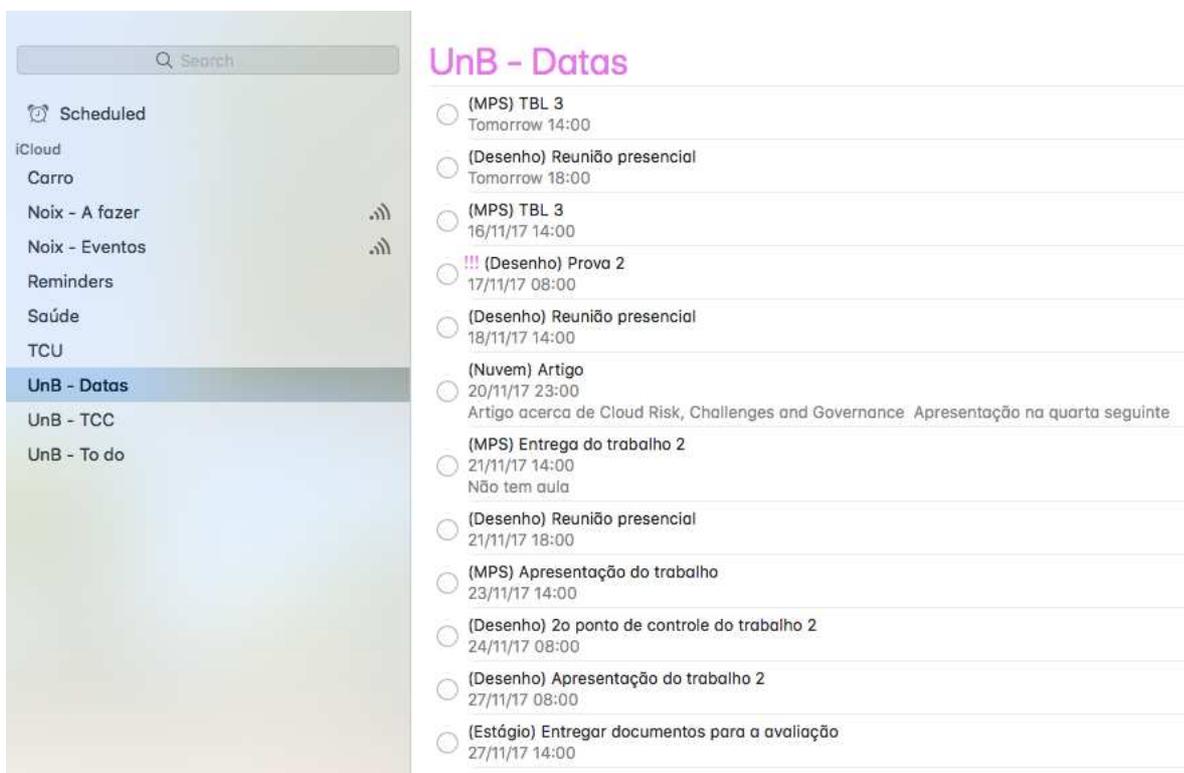


Figura 11 – Tela com o uso da ferramenta Reminders, em sua versão *Desktop*.

3.2.1.2 XMind

Com o objetivo de prover um auxílio visual para a gestão do projeto, foi utilizada a ferramenta XMind. Conforme visto na Figura 12, é uma aplicação para Desktop que permite a criação de diversos gráficos e diagramas como, por exemplo, o de causa e efeito. Além disso, permite a criação de mapas mentais, como o nome sugere.

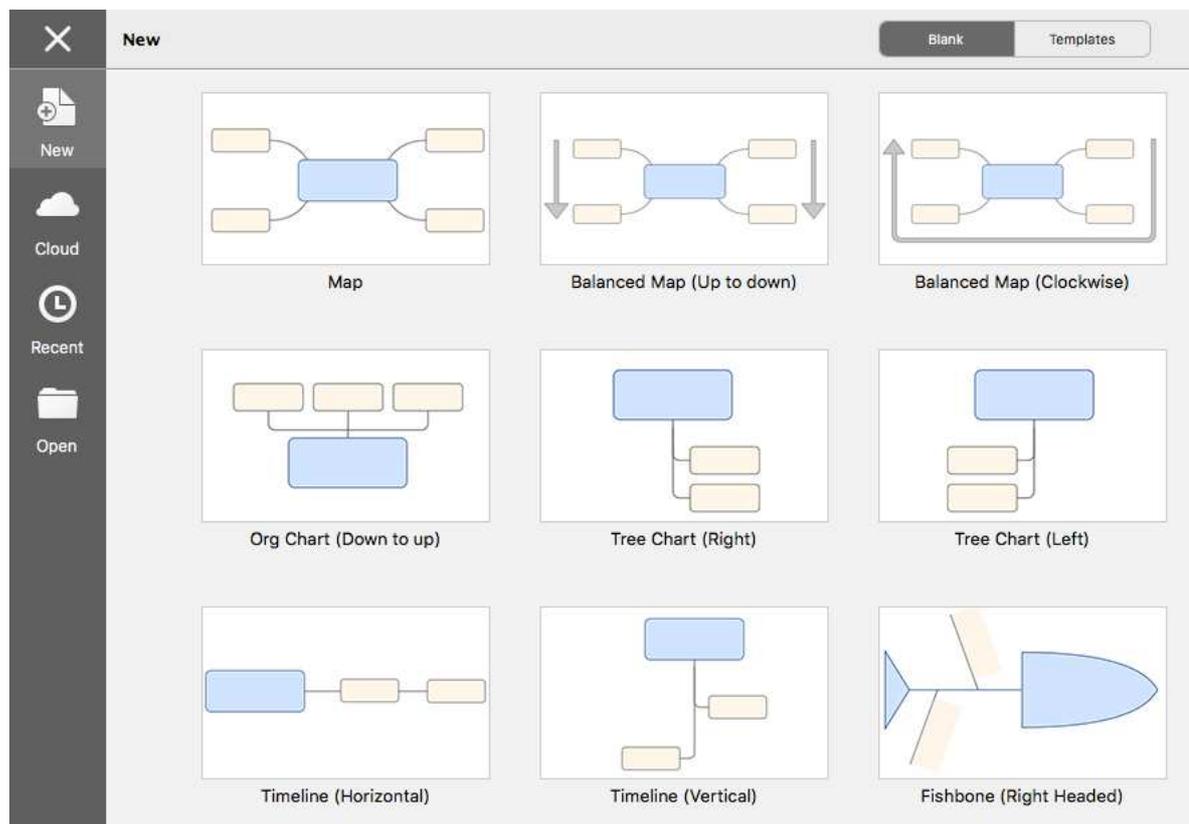


Figura 12 – Tela com o uso da ferramenta XMind.

3.2.2 Desenvolvimento de software

Para a realização do desenvolvimento das aplicações, por intermédio da abordagem proposta, algumas tecnologias foram selecionadas, conforme apresentado nas seções 3.2.2.1 à seção 3.2.2.3.

3.2.2.1 Swift

Swift (SWIFT, 2017), em sua versão 4.0, é uma linguagem de programação construída utilizando uma abordagem moderna para a obtenção de segurança, desempenho e padrões de projeto de software. Permite o desenvolvimento de diversos tipos de aplicações, indo de sistemas, aplicativos - tanto *mobile* como *desktop* - e escalando até serviços em nuvem.

Procura ser uma linguagem segura, rápida, tendo em vista que possui como objetivo ser a sucessora do Objective-C no desenvolvimento de aplicativos. Ainda conforme [Wells \(2015\)](#), mesmo que Swift não tenha muitas funcionalidades originais, possui foco na garantia que as funcionalidades que possui, sejam implementadas de maneira correta e inteligente.

De acordo com ([SOLT, 2015](#)), existem pelo menos dez motivos pelos quais Swift é melhor que sua predecessora, Objective-C, se destacando:

- É mais fácil de se ler;
- É mais fácil de se manter;
- É mais segura;
- Requer menos código, e
- É mais rápida.

Por fim, cabe mencionar que Swift se orienta pela facilidade de uso, conforme visto em [Swift \(2017\)](#).

3.2.2.2 Xcode

O Xcode ([XCODER, 2017](#)), em sua versão 9.3, é um conjunto de ferramentas que visa a criação de aplicativos para Macs, iPhones, iPads, Apple Watch e Apple TV.

Conforme visto na Figura 13, é uma IDE (*Integrated Development Environment*) que traz o *design* de interfaces, codificação, testes, verificação de erros e envio para a App Store - loja de aplicativos da Apple - em um fluxo unificado.

3.2.2.3 Atom

Atom ([ATOM, 2017](#)), em sua versão 1.26, é um editor de texto, gratuito, de código livre, que oferece conveniência no seu uso.

Conforme visto na Figura 14, ao mesmo passo que oferece uma boa experiência para o usuário, é uma ferramenta extensível, permitindo um elevado nível de customização, por meio da instalação de pacotes ([ATOM, 2017](#)).

3.2.3 Gestão de configuração de software

Tendo como objetivo a automação de tarefas inerentes ao desenvolvimento do trabalho, as ferramentas acordadas nas seções 3.2.3.1 e 3.2.3.2 foram utilizadas para a gestão de configuração de software do projeto.

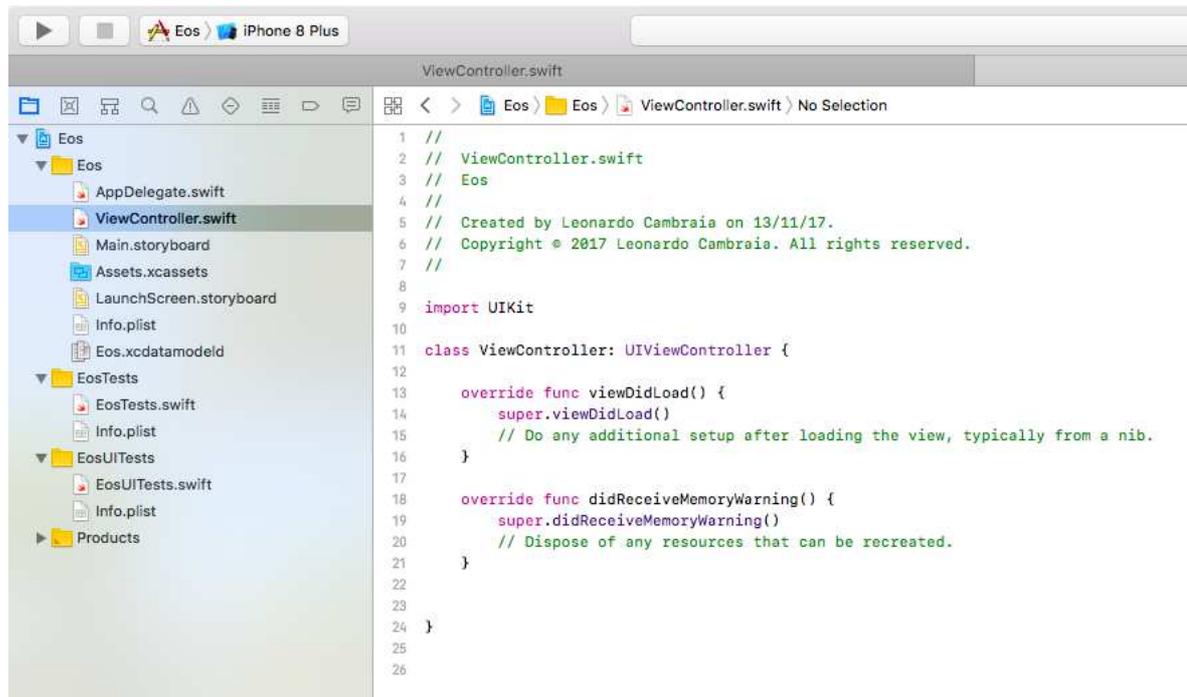


Figura 13 – Tela com o uso da ferramenta Xcode.

3.2.3.1 Git

Para a realização de uma boa gestão de configuração, foi decidido que o Git ([GIT, 2017](#)), em sua versão 2.11.0, seria utilizado para o armazenamento do código, tanto da monografia como dos aplicativos desenvolvidos.

O Git é um sistema de controle de versão distribuído e gratuito, que foi desenhado para lidar com projetos grandes ou pequenos, mantendo a velocidade e eficiência. Possui como atrativos a possibilidade de criação de ramos nos repositórios, simplicidade de uso, confiabilidade e uma licença categorizada como software livre ([GIT, 2017](#)).

3.2.3.2 GitHub

Dentre outras opções disponíveis¹, o GitHub ([GITHUB, 2017a](#)) foi escolhido para ser utilizado neste trabalho. É uma plataforma de armazenamento de código para controle de versão e colaboração, assim permitindo que várias pessoas trabalhem simultaneamente em um mesmo projeto.

Conforme visto na Figura 15, possui como atrativos a possibilidade de criação de repositórios públicos ou privados, revisão de código, documentação e funcionalidades de integração que facilitam o seu uso por comunidades ([GITHUB, 2017b](#)).

¹ <https://about.gitlab.com/>. Acesso em: 10/06/2017.

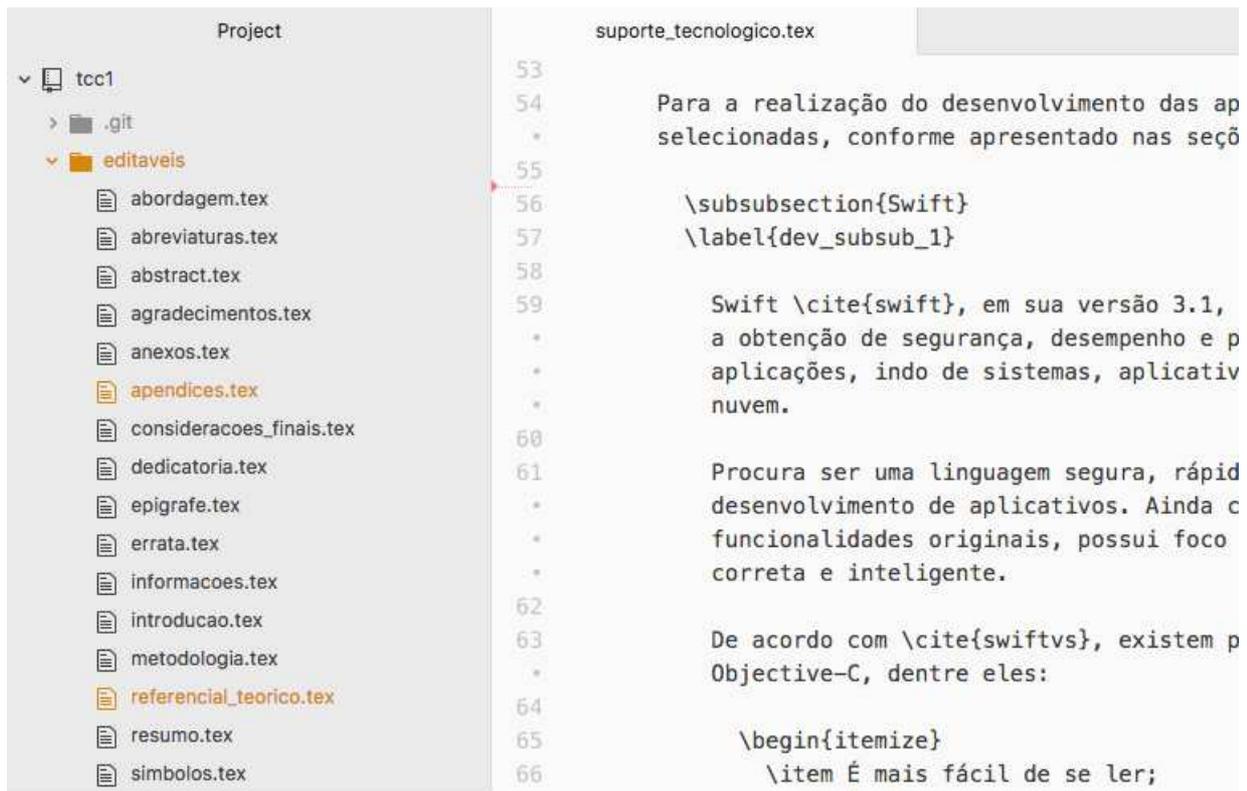


Figura 14 – Tela com o uso da ferramenta Atom.

3.2.4 Pesquisa

Com o intuito de auxiliar na escrita da monografia, foi utilizada a ferramenta \LaTeX (LATEX, 2017), em sua versão 3.14159265; e a ferramenta Zotero (ZOTERO, 2017), em sua versão 4.0.29.15.

3.2.4.1 \LaTeX

Este é um sistema de preparação de documentos com formatação de alta qualidade. É utilizado com frequência para documentos técnicos ou científicos, de médio ou grande porte. Ao mesmo tempo, pode ser utilizado para quase todos os tipos de publicações (LATEX, 2017).

Além da edição dos documentos, o registro das referências utilizadas no trabalho também é realizado por meio do \LaTeX . Para que isso ocorra, a formatação das referências deve seguir o padrão BibTeX, facilmente encontrado na parte de citações do Google Scholar².

² <http://scholar.google.com/>. Acesso em 11/07/2017.

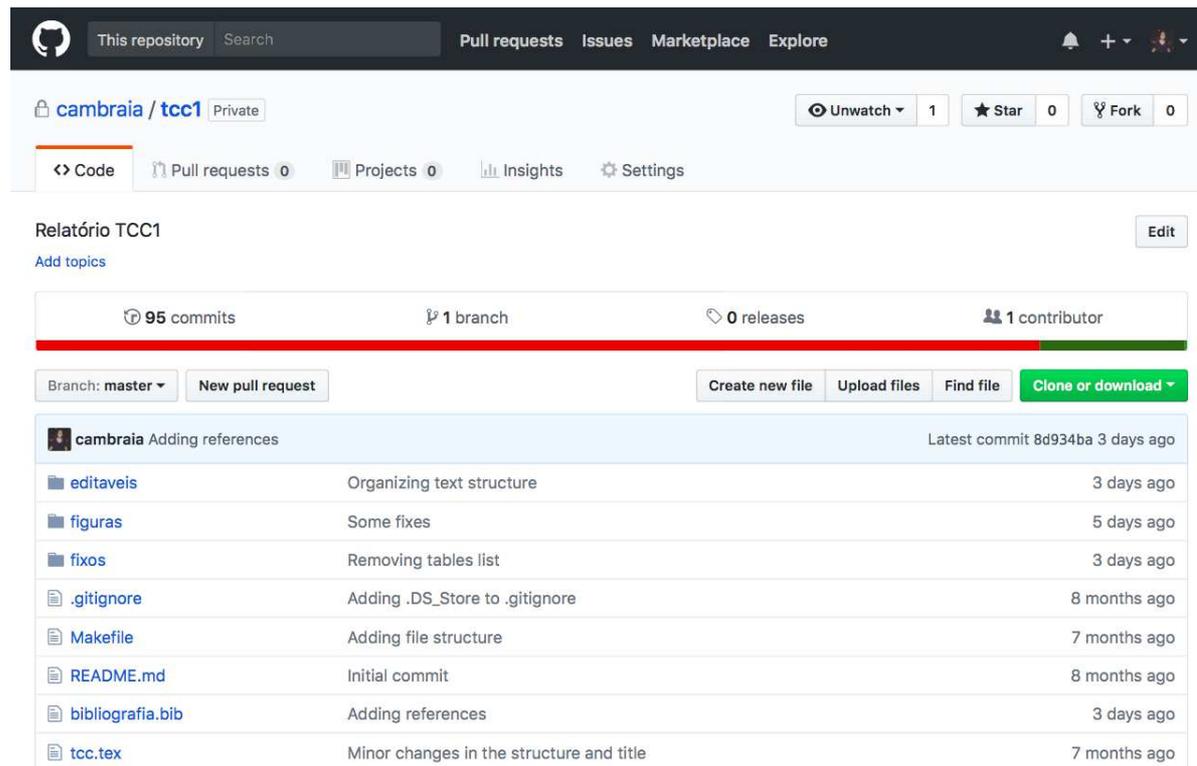


Figura 15 – Tela com o uso do serviço GitHub.

3.2.4.2 Zotero

As referências foram gerenciadas por meio do Zotero, que é uma ferramenta de código aberto, que auxilia na coleta, organização, análise e compartilhamento de pesquisas em diversos modos.

Permite o armazenamento de autores, títulos e outros campos da pesquisa e, posteriormente, a exportação de todos os campos preenchidos como referências formatadas (ZOTERO, 2017).

3.3 Considerações parciais

O capítulo explicou o suporte tecnológico que foi utilizado para a elaboração deste trabalho. Foram exibidas tanto as ferramentas que dizem respeito ao desenho da abordagem, como também as ferramentas que auxiliaram em atividades relativas a Engenharia de Software.

4 Metodologia

Nesse capítulo, é apresentada a metodologia de desenvolvimento do trabalho. Na seção 4.1, a metodologia é detalhada quanto à **pesquisa**; na seção 4.2, é detalhada quanto à **condução do TCC**; na seção 4.3, é detalhada quanto à **avaliação dos resultados**, e na seção 4.4, é apresentado o **cronograma** do trabalho.

4.1 Pesquisa

Conforme Gerhardt e Silveira (2009), a metodologia de pesquisa é classificada conforme a aderência a alguns aspectos, sendo estes: quanto à abordagem, quanto à natureza, quanto aos objetivos, e quanto aos procedimentos. Além destes, ainda conforme Gerhardt e Silveira (2009), foram definidas as técnicas de coleta de dados utilizadas durante o desenvolvimento do trabalho. É importante salientar que a coleta de dados se deu durante a segunda parte do trabalho, após o desenvolvimento inicial, ocorrendo durante a avaliação.

Ainda de acordo com Gerhardt e Silveira (2009), a pesquisa foi definida como uma **pesquisa qualitativa**. A pesquisa qualitativa possui a preocupação com aspectos da realidade que não podem ser quantificados. No caso deste trabalho, levando-se em consideração a questão de pesquisa definida, visou-se descobrir se a aplicação de conceitos oriundos do CBL e do Scrum poderia contribuir na criação de uma abordagem capaz de apoiar iniciantes no desenvolvimento de aplicativos móveis. Dessa forma, um dos objetivos do trabalho foi realizar a avaliação, qualitativa, da contribuição da abordagem proposta.

A natureza da pesquisa foi definida como uma **pesquisa aplicada**. Tal decisão está de acordo com a proposta de uma abordagem com foco no apoio ao desenvolvimento *mobile*, propondo-se, assim, a solução para um problema específico, o que atende aos critérios estabelecidos para esse tipo de pesquisa, conforme definido por Gerhardt e Silveira (2009).

Quanto aos objetivos desta pesquisa, a pesquisa foi definida como uma **pesquisa explicativa**. Gerhardt e Silveira (2009) e Gil (2008) afirmam que este tipo de pesquisa explica o porquê das coisas através dos resultados oferecidos, o que condiz com a proposta do trabalho, que foi determinar a influência da abordagem proposta em certos aspectos já mencionados do aprendizado.

No que diz respeito aos procedimentos escolhidos para o desenvolvimento da pesquisa, dentre os descritos por Gerhardt e Silveira (2009), foram escolhidos **Pesquisa bibliográfica** e **Pesquisa-ação**, além de uma, **Revisão sistemática de literatura**.

A escolha de tais métodos deu-se ao levar em conta a necessidade de um bom levantamento bibliográfico para embasamento da pesquisa, a necessidade de avaliação dos resultados obtidos e a participação do autor e/ou outros grupos na avaliação proposta.

Para colocar em prática a pesquisa proposta, foi necessário que existisse uma coleta de dados condizente com a mesma. Com isso em mente, as seguintes técnicas de coletas de dados propostas por Gerhardt e Silveira (2009) foram definidas: **Pesquisa bibliográfica** e **Pesquisa eletrônica**.

Essas escolhas deram-se com base no levantamento de informações feito tanto em literatura revisada por pares, conforme Fonseca (2002), quanto em materiais encontrados em *sites* confiáveis.

Na Figura 16, é possível visualizar mais claramente a classificação da metodologia de pesquisa escolhida.

Abordagem	Pesquisa qualitativa		Pesquisa quantitativa	
Natureza	Pesquisa básica		Pesquisa aplicada	
Objetivos	Pesquisa exploratória	Pesquisa descritiva	Pesquisa explicativa	
Procedimentos	Pesquisa experimental	Pesquisa bibliográfica	Pesquisa documental	
	Pesquisa de campo	Pesquisa <i>ex-post-facto</i>	Pesquisa de levantamento	
	Pesquisa com <i>survey</i>	Estudo de caso	Pesquisa participante	
	Pesquisa-ação	Pesquisa etnográfica	Pesquisa etnometodológica	
Técnicas de coleta de dados	Pesquisa bibliográfica	Pesquisa documental	Pesquisa eletrônica	
	Questionário	Formulário	Entrevista	
	Observação		Diário de campo	

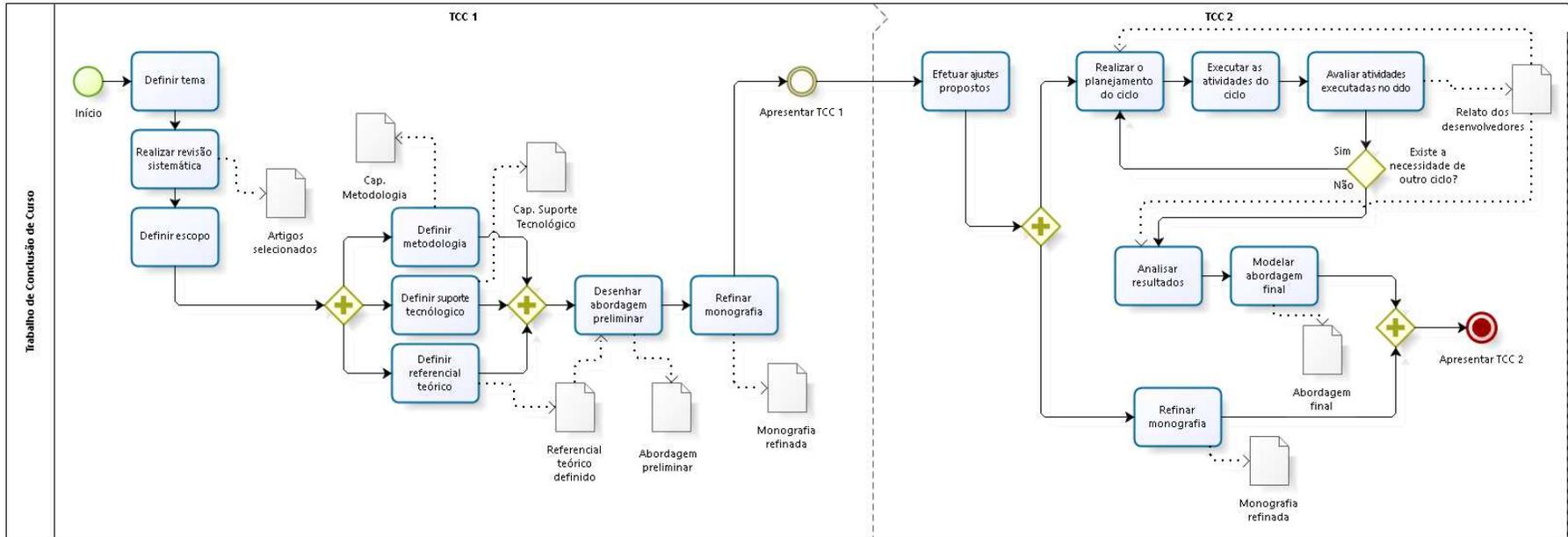
Figura 16 – Seleção metodológica. Fonte: Autor, baseado em (GERHARDT; SILVEIRA, 2009).

4.2 Condução do TCC

Para o desenvolvimento deste trabalho, um processo de condução foi definido.

Na Figura 17, é possível visualizar o processo definido.

Esse processo englobou duas entregas previstas. A primeira parte possui um foco maior na fundamentação exigida para a execução do trabalho e o desenvolvimento da abordagem preliminar proposta. A segunda possuiu como objetivo realizar o aprimora-



Powered by
bizagi
Modeler

Figura 17 – Processo de condução do TCC. Fonte: Autor.

mento e a validação da abordagem proposta, por meio de ciclos de pesquisa-ação, conforme Petersen et al. (2014).

A seguir, são apresentadas as atividades pertinentes ao TCC 1.

- **Definir tema:**

- Descrição: Realizar escolha do tema do trabalho. É necessário que seja realizada com base na percepção de *gap* tecnológico, análise de literatura científica relacionada, afinidade do autor e possibilidade de pesquisa dentro do tempo da disciplina.
- Artefato utilizado: -
- Artefato gerado: -

- **Realizar revisão sistemática:**

- Descrição: Realizar uma revisão sistemática de literatura com o intuito de verificar a existência de pesquisas similares e adquirir conhecimento no tema escolhido;
- Artefato utilizado: -
- Artefato gerado: Artigos selecionados.

- **Definir escopo:**

- Descrição: Delimitação dos limites da pesquisa, com base no tema escolhido. É necessário que seja feita a partir de critérios definidos pelo autor, tais como a relevância da pesquisa para a comunidade e a não abordagem do tema em trabalhos relacionados.
- Artefato utilizado: -
- Artefato gerado: -

- **Definir metodologia:**

- Descrição: Definir o tipo de pesquisa a ser feita, podendo incluir detalhes mais específicos sobre o trabalho;
- Artefato utilizado: -
- Artefato gerado: Capítulo de Metodologia.

- **Definir suporte tecnológico:**

- Descrição: Definir as ferramentas de auxílio ao desenvolvimento do trabalho;
- Artefato utilizado: -

- Artefato gerado: Capítulo de Suporte Tecnológico.

- **Definir referencial teórico:**

- Descrição: Levantar uma fundamentação bibliográfica para o trabalho. É necessário para comprovar que o que está sendo proposto tem real validade;
- Artefato utilizado: -
- Artefato gerado: Referencial teórico definido.

- **Desenhar abordagem preliminar:**

- Descrição: Desenvolver uma versão preliminar da abordagem a ser proposta como produto do trabalho;
- Artefato utilizado: Referencial teórico definido;
- Artefato gerado: Abordagem preliminar.

- **Refinar monografia:**

- Descrição: Refinar o texto do trabalho, que será objeto de avaliação para a disciplina, junto a apresentação para a banca;
- Artefato utilizado: -
- Artefato gerado: Monografia refinada.

- **Apresentar TCC 1:**

- Descrição: Apresentar para a banca. Marcará o fim da primeira etapa do trabalho.
- Artefato utilizado: -
- Artefato gerado: -

A seguir, são apresentadas as atividades pertinentes ao TCC 2.

- **Efetuar ajustes propostos:**

- Descrição: Realizar ajustes propostos pela banca na apresentação da primeira etapa do trabalho.
- Artefato utilizado: -
- Artefato gerado: -

- **Realizar o planejamento do ciclo:**
 - Descrição: Definir o escopo do ciclo e planejar a avaliação;
 - Artefato utilizado: Relato de avaliação do ciclo anterior, caso não seja o primeiro ciclo.
 - Artefato gerado: -
- **Executar atividades do ciclo:**
 - Descrição: Executar as atividades planejadas para o ciclo.
 - Artefato utilizado: -
 - Artefato gerado: -
- **Avaliar atividades executadas no ciclo:**
 - Descrição: Avaliar o que foi desenvolvido e identificar possíveis melhorias;
 - Artefato utilizado: -
 - Artefato gerado: Relato dos desenvolvedores.
- **Analisar resultados:**
 - Descrição: Analisar resultados coletados ao longo dos ciclos;
 - Artefato utilizado: Relato dos desenvolvedores.
 - Artefato gerado: -
- **Modelar abordagem final:**
 - Descrição: Modelar e documentar abordagem final, após avaliações;
 - Artefato utilizado: -
 - Artefato gerado: Abordagem final.
- **Refinar monografia:**
 - Descrição: Refinar a monografia com os resultados obtidos nos ciclos;
 - Artefato utilizado: -
 - Artefato gerado: Monografia refinada.
- **Apresentar TCC 2:**
 - Descrição: Apresentar para a banca. Marcará o fim do trabalho.
 - Artefato utilizado: -
 - Artefato gerado: -

4.3 Avaliação dos resultados

Nesta seção, é apresentado o método seguido para a avaliação dos resultados obtidos ao longo do trabalho.

4.3.1 Pesquisa-ação

Com o intuito de garantir que as atividades definidas atingissem ao seus objetivos, foi realizada uma pesquisa-ação.

A pesquisa-ação definida possui uma estrutura cíclica, composta por etapas de Planejamento, de Execução e de Avaliação, sendo que antes de seu início houve uma única vez a etapa de Diagnóstico, conforme pode ser visto na Figura 18.

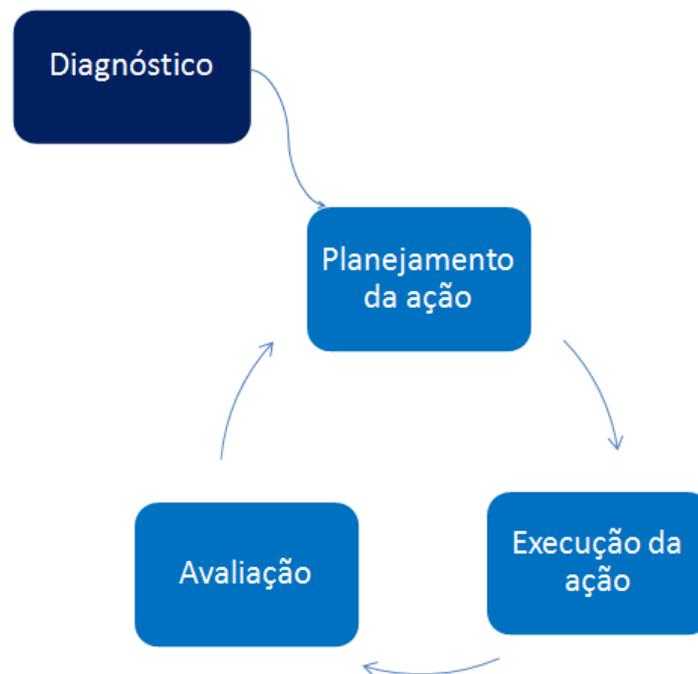


Figura 18 – Estrutura da pesquisa-ação. Fonte: Autor

Este trabalho possuiu dois ciclos de pesquisa-ação, onde tais ciclos, possuíram como objetivo a validação da estrutura de atividades propostas para a abordagem. Durante o desenvolvimento do trabalho, essas atividades foram alteradas, de acordo com a necessidade.

4.3.2 Validação

Para que a validação da abordagem ocorresse de forma bem sucedida, além de utilizar a pesquisa-ação, foi necessário realizar a seleção de participantes para o desen-

volvimento dos aplicativos - utilizando-se da abordagem, definir como seria feita a coleta dos dados e como as mudanças seriam propostas.

4.3.2.1 Participantes

Conforme mencionado na seção 1.3, a abordagem possui como público-alvo desenvolvedores que ainda não possuem experiência em desenvolvimento *mobile*. Com isso em mente, para participar na validação, foram buscadas pessoas que se encontrassem dentro deste perfil.

Além disso, para que a abordagem fosse aplicada com sucesso, foi fundamental que os participantes já tivessem um conhecimento prévio sobre lógica de programação.

Assim, para participar na validação, cada pessoa obedeceu aos seguintes critérios:

- Ter sido aprovado em uma disciplina de Introdução à Ciência da Computação, ou disciplina com ementa similar;
- Ter acesso a um computador com o sistema operacional macOS Sierra 10.13, ou superior;
- Em cada ciclo, possuir disponibilidade de tempo para:
 - Breve treinamento sobre a abordagem e resolução de dúvidas;
 - Desenvolvimento de um aplicativo utilizando a abordagem;
 - Fornecer *feedback* sobre a utilização da abordagem.

Após encontrar participantes interessados, os critérios de seleção foram aplicados e para iniciar a validação, individualmente, foi realizado um breve treinamento na abordagem para que possíveis dúvidas sobre as atividades fossem sanadas.

4.3.2.2 Coleta de dados

Foi solicitado a cada participante que, durante a execução das atividades da abordagem, documentassem todos os resultados obtidos com cada atividade, observações que viessem a ter e possíveis sugestões de mudança. De tal forma, cada participante forneceu um relato qualitativo sobre o uso da abordagem.

Ao fim da execução da abordagem, ocorreram entrevistas com os participantes, onde foi realizado um questionamento sobre a relevância de cada atividade. Este questionamento se deu de forma qualitativa, onde expressavam, em uma escala de um a cinco, a relevância que determinada atividade teve no desenvolvimento de seu aplicativo.

4.3.2.3 Propostas de mudanças

Em dois momentos durante a validação da abordagem, ao fim de cada ciclo de pesquisa-ação, o Autor realizou uma análise com base nos dados coletados, o que levou ao refinamento da abordagem proposta.

Essa análise, possuiu como objetivo verificar, especialmente, a importância de cada atividade e eventuais necessidades de mudança em seu ordenamento ou granularidade.

4.4 Cronograma

Em concordância com o processo de condução definido para o TCC na seção 4.2, foi elaborado um cronograma, dividido em duas partes, vistas nas Figuras 19 e 20, com as atividades desenvolvidas durante o trabalho.

Cronograma para o TCC1				
Atividade	Agosto	Setembro	Outubro	Novembro
Definir tema	x			
Realizar revisão sistemática	x			
Definir escopo	x			
Elaborar proposta inicial	x			
Definir metodologia		x	x	
Definir suporte tecnológico		x	x	
Definir referencial teórico		x	x	
Desenhar abordagem preliminar			x	
Refinar monografia				x
Apresentar TCC 1				x

Figura 19 – Cronograma para o TCC1. Fonte: Autor

4.5 Considerações parciais

O capítulo abordou as metodologias de pesquisa utilizadas neste trabalho, além de detalhar a forma como foi conduzido este trabalho. Além disso, apresenta como este trabalho foi avaliado, utilizando-se de uma pesquisa-ação. Ao fim do capítulo, foram trazidos os cronogramas.

Cronograma para o TCC2				
Atividade	Março	Abril	Maio	Junho
Efetuar ajustes propostos	x			
Realizar o planejamento do ciclo 1	x			
Executar atividades do ciclo 1	x	x		
Avaliar atividades executadas no ciclo 1		x		
Realizar o planejamento do ciclo 2		x		
Executar atividades do ciclo 2			x	
Avaliar atividades executadas no ciclo 2			x	
Analisar resultados			x	x
Modelar abordagem final				x
Refinar monografia	x	x	x	x
Apresentar TCC2				x

Figura 20 – Cronograma para o TCC2. Fonte: Autor

5 Abordagem preliminar

Neste capítulo, é apresentada a abordagem preliminar, que por sua vez orienta-se por atividades do Scrum e do CBL, considerados *frameworks* de referência. Na seção 5.1, será apresentado como foi realizado o **mapeamento entre as atividades** dos *frameworks* de referência; na seção 5.2, será apresentado o **fluxo da abordagem** e a **descrição de suas atividades**.

5.1 Proposta preliminar

Conforme indicado no capítulo 1, esse trabalho de conclusão de curso consistiu na elaboração de uma abordagem de desenvolvimento, a qual possui como intuito principal auxiliar os desenvolvedores, sem experiência em desenvolvimento *mobile*, nos seus primeiros passos para a criação de aplicativos voltados para o sistema operacional iOS.

Para a primeira entrega do trabalho, uma versão preliminar da abordagem foi proposta. Para a construção desta versão, e garantia de coerência com os referenciais utilizados, foi realizado um mapeamento das atividades do Scrum, por [Sharma e Hasteer \(2016\)](#), e do CBL, por [Nichols, Cator e Torres \(2016\)](#). A partir deste mapeamento, foram definidas as atividades propostas neste trabalho.

Ao realizar o mapeamento, nem sempre foi possível manter uma relação de um para um ao mapear as atividades. Em resultado disso, algumas atividades propostas não possuem uma contraparte correspondente em um dos referenciais ou possuem relação com mais de uma delas.

Além disso, um dos principais objetivos das atividades presentes na abordagem proposta, é que possam ser simples, de modo a auxiliar o público-alvo. Os princípios oferecidos pelo *Lean Thinking* auxiliaram neste ponto. Durante a criação das atividades, embora primariamente guiadas pelos referenciais presentes no mapeamento, foi levado em conta cada princípio aplicável do *Lean Thinking*.

A seguir, são apresentados os princípios, junto a identificadores - para que possa manter uma relação entre estes e as atividades criadas, conforme consta na próxima seção, Estes princípios podem ser vistos com mais detalhes na seção 2.2.

- P1 - Elimine o desnecessário;
- P2 - Aumente o aprendizado;
- P3 - Decida o mais tardiamente;

- P4 - Entregue o mais rapidamente;
- P5 - Empodere a equipe;
- P6 - Mantenha a integridade, e
- P7 - Veja o todo.

5.2 Estrutura das atividades

Além do mapeamento inicial, cada atividade possui **Descrição**; **Justificativa**; e **Exemplo ou Sugestão de Uso**.

- A **Descrição** possui como objetivo explicar o que se espera que ocorra naquelas atividades. Dessa forma, são providas mais informações do que o apresentado no nome da atividade;
- A **Justificativa** possui como objetivo explicar para o leitor o motivo da inclusão de cada atividade na abordagem. Essa explicação foi embasada na literatura. A literatura utilizada para a definição de cada atividade é vista na legenda de cada figura sobre cada atividade;
- Os **Exemplos ou Sugestões de Uso** possuem como objetivo auxiliar o leitor, de forma prática, no entendimento do que se espera ou sugerir um método ou ferramenta para a execução daquela atividade.

O fluxo apresentado na Figura 21 representa a abordagem preliminar.

5.2.1 Descrição das atividades

As atividades desenvolvidas são descritas a seguir, utilizando a estrutura acordada anteriormente.

- **Ter uma ideia:**

Orientando-se por [Wooldridge et al. \(2011\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Introduzir ideia do produto;
- Base no CBL: Ter uma ideia;
- Princípio(s) *Lean*: -
- Descrição: Escolher a ideia inicial para o desenvolvimento da aplicação;

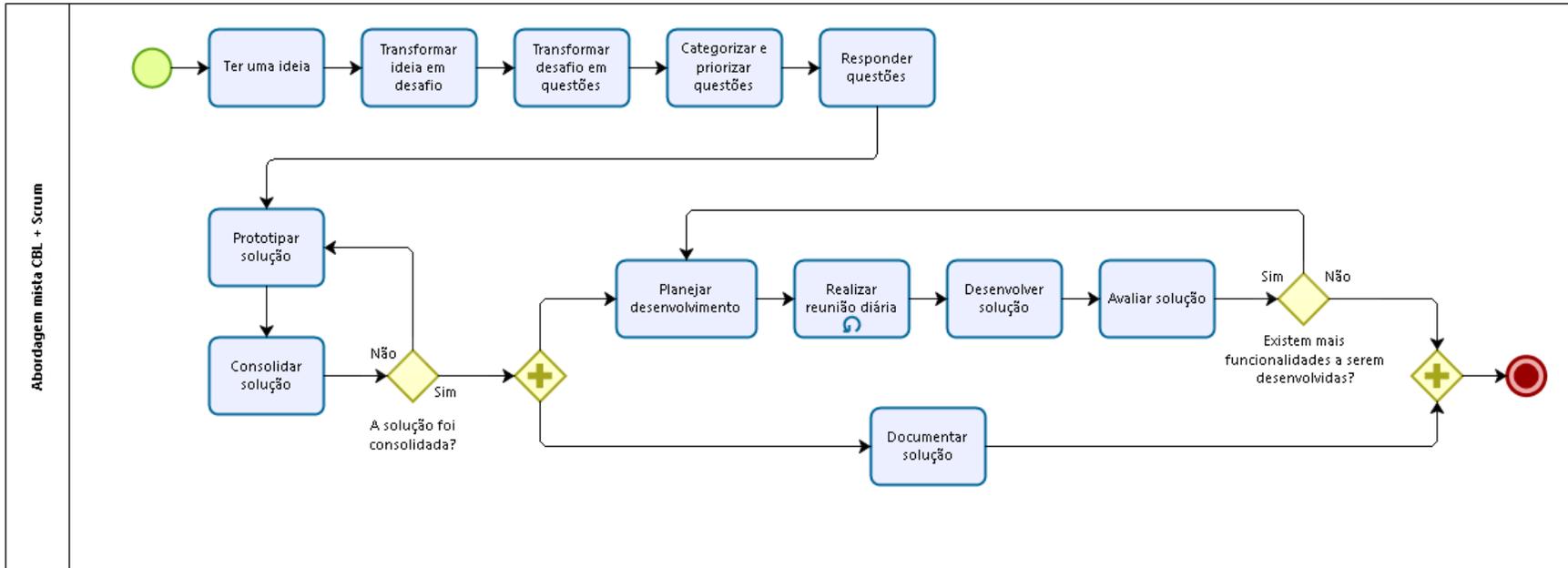


Figura 21 – Fluxo da abordagem preliminar proposta. Fonte: Autor.

- Justificativa: É necessário para que se possa dar início ao projeto;
- Exemplo: Ter uma ideia de aplicativo relacionado com a área da saúde.
- Sugestão: -

- **Transformar ideia em desafio:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Realizar questionamento essencial e Transformar questão essencial em desafio;
- Princípio(s) *Lean*: [P2](#) e [P7](#);
- Descrição: Transformar a ideia inicial em um desafio a ser enfrentado;
- Justificativa: É necessário para que a ideia seja transformada em um objetivo claro;
- Exemplo: Transformar a ideia de um aplicativo relacionado com a área da saúde no desafio “Tornar meu usuário mais saudável!”.
- Sugestão: -

- **Transformar desafio em questões:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Criar questões guia;
- Princípio(s) *Lean*: [P2](#) e [P7](#);
- Descrição: Transformar o desafio em questões;
- Justificativa: É necessário para que os requisitos comecem a ser trabalhados no início do projeto, ao passo que, inicie-se o aprendizado;
- Exemplo: Transformar o desafio de tornar o usuário mais saudável em questões como Q1: “O que preciso fazer para ser mais saudável?”.
- Sugestão: -

- **Categorizar e priorizar questões:**

Orientando-se por [Karlsson, Wohlin e Regnell \(1998\)](#), [Berander e Andrews \(2005\)](#) e [Davis \(2003\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Categorizar e priorizar questões guia;

- Princípio(s) *Lean*: P2 e P7;
- Descrição: Categorizar as questões criadas e priorizá-las;
- Justificativa: É necessário para que a equipe de desenvolvimento identifique o que é mais importante e ordene sua lista de trabalho, tendo em vista limitações do projeto, como, por exemplo, tempo e custos;
- Exemplo: -
- Sugestão: Sugere-se utilizar o teste dos 100 dólares.

- **Responder questões:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Realizar atividades guia;
- Princípio(s) *Lean*: P2 e P7;
- Descrição: Possuindo como objetivo elucidar o que o aplicativo deverá agregar de valor para o usuário, as questões criadas devem ser respondidas;
- Justificativa: É necessário para iniciar o aprendizado e a coleta de conhecimento, ao mesmo tempo que, provê as respostas para as questões levantadas anteriormente.
- Exemplo: Resposta para a Q1: Preciso registrar meus dados de treino.
- Sugestão: -

- **Prototipar solução:**

Orientando-se por [Pressman \(2005\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos e Gerar *backlog* do produto;
- Base no CBL: Analisar as lições aprendidas e Prototipar conceito da solução;
- Princípio(s) *Lean*: P1;
- Descrição: Prototipar solução para validação com os *stakeholders*, utilizando o conceito de *wireframe* para obter apelo visual;
- Justificativa: É necessário para que a consolidação do conhecimento tome forma e o conhecimento possa ser traduzido em algo mais visual, permitindo, assim, a validação com os *stakeholders*;
- Exemplo: Um protótipo de tela inicial pode visto ao lado de seu aplicativo desenvolvido na Figura 22.
- Sugestão: Sugere-se utilizar a prototipação em papel.



Figura 22 – Exemplo de comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Autor

- **Consolidar solução:**

Orientando-se por [Myers, Sandler e Badgett \(2011\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Gerar *backlog* do produto;
- Base no CBL: Testar conceito da solução e Refinar conceito da solução;
- Princípio(s) *Lean*: **P1** e **P6**;
- Descrição: Consolidar a solução com base no *feedback* dos protótipos, validados junto aos *stakeholders*;
- Justificativa: É necessário para que o desenvolvimento não seja iniciado com uma percepção errada acerca da solução;

- Exemplo: Consolidar que será desenvolvido um aplicativo que incentivará o usuário a realizar corridas matinais.
- Sugestão: -

- **Planejar desenvolvimento:**

Orientando-se por [Sharma e Hasteer \(2016\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Planejar *sprint*, Priorizar atividades e Gerar *backlog* da *sprint*;
- Base no CBL: -
- Princípio(s) *Lean*: [P3](#), [P5](#), [P6](#) e [P7](#);
- Descrição: Definir quais serão os seus passos para que a ideia seja executada e se obtenha um aplicativo como produto final, observando a priorização definida anteriormente;
- Justificativa: É necessário para organizar o que é planejado naquela *sprint* e, posteriormente, permitir a análise da relação entre planejado e feito.
- Exemplo: -
- Sugestão: -

- **Realizar reunião diária:**

Orientando-se por [Stray, Moe e Aurum \(2012\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Realizar reunião diária;
- Base no CBL: -
- Princípio(s) *Lean*: [P4](#), [P5](#) e [P6](#);
- Descrição: Realizar uma breve reunião diária, por poucos minutos, entre a equipe de desenvolvimento;
- Justificativa: É necessário para que o status do projeto esteja claro entre os membros da equipe de desenvolvimento.
- Exemplo: Após a realização de uma reunião diária, o responsável pela ata registrou as seguintes observações:
 - * **Ata da reunião do dia 01/01/2045:**
 - * Desenvolvedor #3: Ontem finalizei a segunda versão do protótipo da nova funcionalidade e hoje vou leva-lo para a validação com o cliente;
 - * Desenvolvedor #1: Ótimo! Eu corrigi alguns *bugs* no *login* com o Facebook e hoje irei verificar se restou algum para correção;

- * Desenvolvedor #2: E pra terminarmos eu estou com alguns problemas ao acessar a câmera, mas acredito que hoje mesmo vou conseguir acessa-la, mas se precisar, peço ajuda a um de vocês.

– Sugestão: -

- **Desenvolver solução:**

Orientando-se por [Cockburn \(2002\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Realizar desenvolvimento;
- Base no CBL: Implementar solução;
- Princípio(s) *Lean*: [P1](#), [P4](#) e [P5](#);
- Descrição: Codificar a solução consolidada com base nos protótipos desenhados;
- Justificativa: É necessário para que a solução seja entregue;
- Exemplo:

```
print("Let's get healthy!")
```

– Sugestão: Sugere-se utilizar a linguagem Swift.

- **Avaliar solução:**

Orientando-se por [Pikkarainen et al. \(2008\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Realizar revisão da *sprint* e Realizar retrospectiva da *sprint*;
- Base no CBL: Avaliar impacto no desafio;
- Princípio(s) *Lean*: [P6](#) e [P7](#);
- Descrição: Realizar uma comparação entre o protótipo consolidado e o aplicativo final para avaliação da solução desenvolvida;
- Justificativa: É necessário para garantir que o que foi desenvolvido esteja em conformidade com o esperado.
- Exemplo: -
- Sugestão: -

- **Documentar solução:**

Orientando-se por [Forward e Lethbridge \(2002\)](#) e [Briand \(2003\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: -

- Base no CBL: -
- Princípio(s) *Lean*: P6 e P7;
- Descrição: Manter a solução documentada por meio do registro dos resultados e observações de cada atividade executada;
- Justificativa: É necessário para garantir que o conhecimento adquirido sobre a solução não seja perdido.
- Exemplo: -
- Sugestão: -

5.3 Considerações parciais

O capítulo trouxe a abordagem preliminar proposta na primeira parte deste trabalho. Foi detalhado como ocorreu o mapeamento, considerando os *frameworks* referenciais Scrum e CBL. Esse mapeamento permitiu embasar essa visão preliminar da abordagem bem como estabelecer um fluxo de atividades para a mesma. Adicionalmente, foi apresentado o fluxo e cada atividade em detalhe. A seguir, serão apresentados os resultados obtidos nesse trabalho de conclusão de curso, bem como a versão final da abordagem, após a realização de ciclos de pesquisa-ação junto ao público alvo.

6 Resultados e abordagem final

Nesse capítulo, são apresentados os resultados obtidos durante a execução deste trabalho, bem como a abordagem em sua versão final. O capítulo está organizado em seções. Na seção 6.1, tem-se a descrição das avaliações realizadas na abordagem junto ao público alvo, considerando ciclos de pesquisa-ação. Justificativas para escolha dessa modalidade de pesquisa - pesquisa-ação - foram apresentadas na seção 4.1 dessa monografia. Na seção 6.2, tem-se a análise realizada, com base em cada ciclo. Por fim, na seção 6.3, consta a apresentação da abordagem em sua versão final.

6.1 Execução por ciclos

Visando coletar as primeiras impressões da abordagem proposta, conforme detalhado na seção 4.3.2, foi realizado o desenvolvimento de aplicativos pelo Autor deste trabalho, bem como por terceiros. Esses terceiros foram selecionados em aderência ao perfil alvo da abordagem, ou seja, interessados com pouco conhecimento sobre desenvolvimento em plataformas móveis para o sistema operacional iOS.

Neste trabalho, dois ciclos de pesquisa-ação foram realizados, conforme o cronograma de validação encontrado na Figura 23.

6.1.1 Ciclo 1

Durante o primeiro ciclo, utilizando-se da abordagem preliminar proposta na primeira parte deste trabalho, dois aplicativos foram desenvolvidos.

O primeiro deles, um aplicativo que permite desenhar na tela do dispositivo foi desenvolvido pelo Autor deste trabalho. A intenção era o próprio Autor vivenciar o desenvolvimento de um projeto *mobile* fazendo uso da abordagem. Os detalhes da execução de todas as atividades, para este aplicativo, podem ser encontrados no Apêndice, na seção C.1. É importante ressaltar a aderência do perfil estabelecido na seção 4.3.2.1 com o perfil do Autor, conforme é detalhado a seguir:

- Estudante do 11o semestre de Engenharia de Software;
- Possui experiência com testes de software, estimativa de custos de projetos e melhoria de processos;
- Não possuía experiência prévia com desenvolvimento *mobile*.

Cronograma de validação									
	Abril				Maio				Junho
	Semana 9-13	Semana 16-20	Semana 23-27	Semana 30-4 (Maio)	Semana 7-11	Semana 14-18	Semana 21-25	Semana 28-1 (Junho)	Semana 4-8
Atividades	Ciclo 1				Atividades	Ciclo 2			
	Sprint 0	Sprint 1				Sprint 0	Sprint 1		
<i>Ter uma ideia</i>	x				<i>Ter uma ideia</i>	x			
<i>Transformar ideia em desafio</i>	x				<i>Transformar ideia em desafios</i>	x			
<i>Transformar desafio em questões</i>	x				<i>Transformar desafios em questões</i>	x			
<i>Categorizar e priorizar questões</i>	x				<i>Responder questões</i>	x			
<i>Responder questões</i>	x				<i>Priorizar questões</i>	x			
<i>Prototipar solução</i>	x				<i>Prototipar e consolidar solução</i>	x			
<i>Consolidar solução</i>	x				<i>Planejar desenvolvimento</i>		x		
<i>Planejar desenvolvimento</i>		x			<i>Realizar reunião diária</i>		x	x	x
<i>Realizar reunião diária</i>		x	x	x	<i>Desenvolver solução</i>		x	x	x
<i>Desenvolver solução</i>		x	x	x	<i>Avaliar solução</i>			x	x
<i>Avaliar solução</i>			x	x	<i>Documentar solução</i>	x	x	x	x
<i>Documentar solução</i>	x	x	x	x	-	-	-	-	-

Figura 23 – Cronograma de validação. Fonte: Autor

O segundo aplicativo desenvolvido neste ciclo permite que o usuário realize o cálculo de conversão de valores de resistores e foi desenvolvido pelo Desenvolvedor #1. Os detalhes da execução de todas as atividades, para este aplicativo, podem ser encontrados no Apêndice, na seção C.2. É importante ressaltar a aderência do perfil estabelecido na seção 4.3.2.1 com o perfil do Desenvolvedor #1, conforme é detalhado a seguir:

- Estudante do 11o semestre de Engenharia Automotiva;
- Possui experiência com desenho industrial, usinagem e gestão de projetos;
- Não possuía experiência prévia com desenvolvimento *mobile*.

6.1.2 Ciclo 2

Durante o segundo ciclo, foi utilizada a segunda versão da abordagem. Para se chegar nesta versão, as mudanças propostas pelos participantes do primeiro ciclo foram analisadas, qualitativamente, pelo Autor, conforme apresentado na seção 6.2.

Utilizando-se desta nova versão da abordagem, foram desenvolvidos três aplicativos.

O primeiro deles foi desenvolvido pelo Autor deste trabalho. O objetivo do aplicativo é a identificação de imagens, que por sua vez podem ser capturadas na hora ou carregadas do próprio dispositivo. Para que isso fosse possível de forma *offline*, foi utilizado o *framework* CoreML¹, para que, junto a um modelo de *Machine Learning* já treinado, fosse possível a identificação das imagens e sua possível correspondência. Os detalhes da execução de todas as atividades, para este aplicativo, podem ser encontrados no Apêndice, na seção C.3.

Já o segundo aplicativo foi desenvolvido pelo Desenvolvedor #1, possuindo como objetivo a identificação e leitura de códigos QR². Os detalhes da execução de todas as atividades, para este aplicativo, podem ser encontrados no Apêndice, na seção C.4.

O terceiro aplicativo foi desenvolvido pelo Desenvolvedor #2, e permite que o usuário pese pequenos objetos ou a pressão exercida pelo próprio dedo sobre a tela do dispositivo. Os detalhes da execução de todas as atividades, para este aplicativo, podem ser encontrados no Apêndice, na seção C.5. É importante ressaltar a aderência do perfil estabelecido na seção 4.3.2.1 com o perfil do Desenvolvedor #2, conforme é detalhado a seguir:

¹ <https://developer.apple.com/documentation/coreml>. Acesso em: 18/06/2018.

² <https://www.iso.org/standard/62021.html>. Acesso em: 08/06/2018.

- Estudante do 8o semestre de Engenharia de Software;
- Possui experiência com elicitação de requisitos, desenvolvimento em Java e desenvolvimento embarcado;
- Possuía conhecimentos dos conceitos básicos do desenvolvimento *mobile*, porém nunca desenvolveu antes.

6.2 Análise por ciclos

Na seção 4.3.2.3, é definido que ao fim de cada ciclo de pesquisa-ação, seria realizada uma análise de seus resultados e com base nessa análise, possíveis mudanças nas atividades poderiam ser realizadas.

6.2.1 Ciclo 1

No início do primeiro ciclo, por utilização das atividades provenientes da abordagem preliminar, proposta no capítulo anterior, sua disposição dentro da pesquisa-ação era a encontrada na Figura 24.

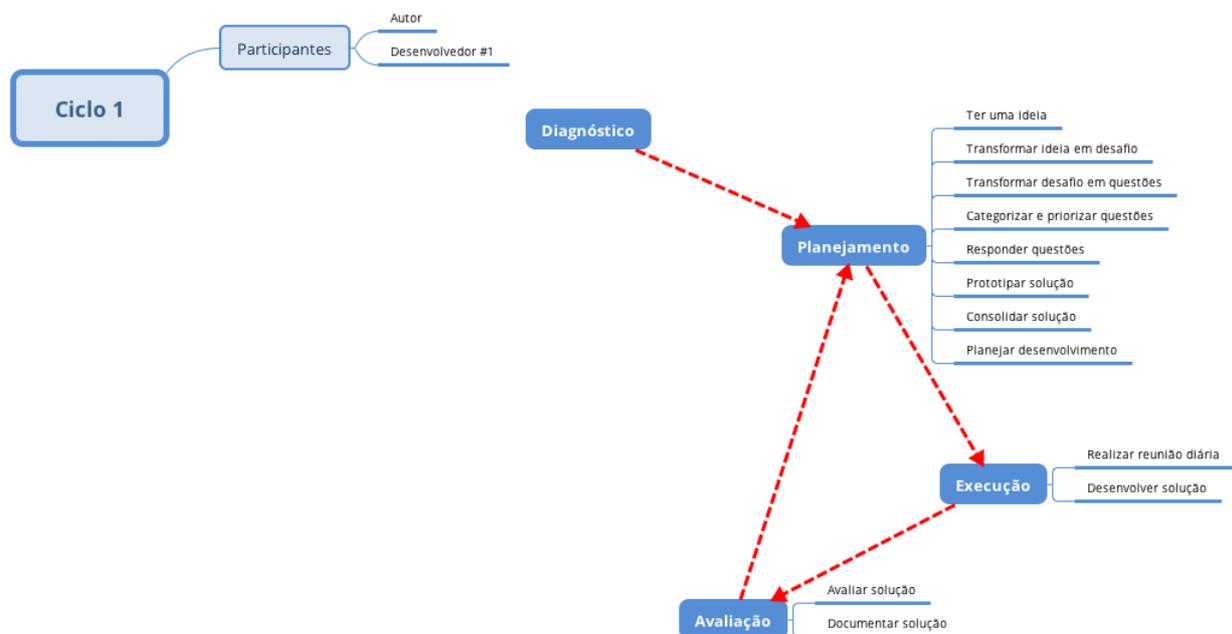


Figura 24 – Atividades no primeiro ciclo. Fonte: Autor

Após a execução do primeiro ciclo, conforme apresentado na seção 6.1.1, foi realizada uma análise qualitativa acerca dos resultados obtidos.

Por meio da análise qualitativa foi possível verificar o nível de relevância de determinada atividade para o desenvolvedor. Para o primeiro ciclo, esse nível pode ser visto na Figura 25.

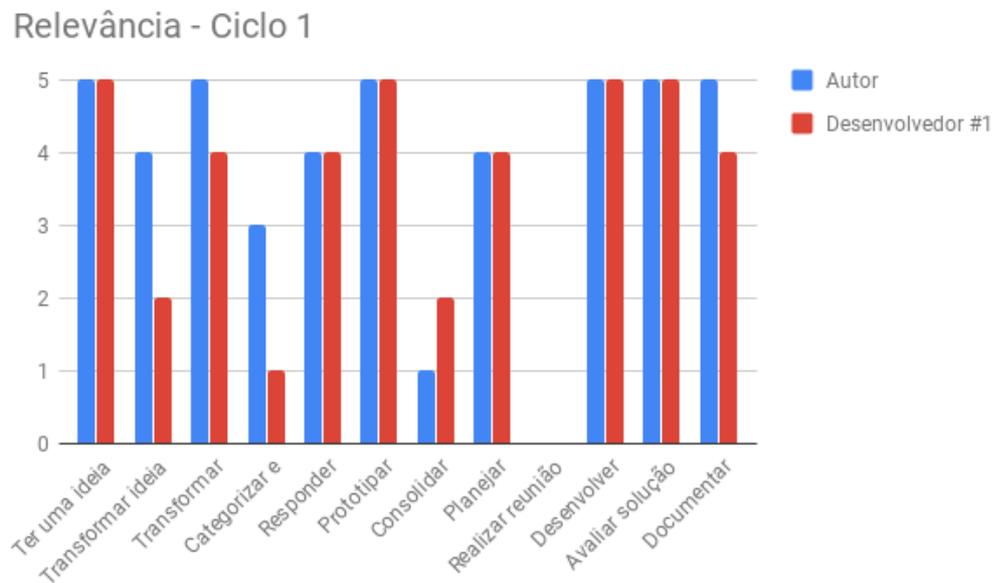


Figura 25 – Relevância das atividades para os desenvolvedores no primeiro ciclo. Fonte: Autor

Ainda de forma qualitativa, uma análise foi realizada com base nas observações documentadas pelos desenvolvedores ao longo da execução das atividades para o desenvolvimento de cada aplicativo. Essas informações podem ser vistas no Apêndice C.

Com base nos dados obtidos, algumas alterações no conjunto de atividades foram realizadas, sendo estas:

- Na atividade "Transformar ideia em desafio":
 - O nome da atividade foi alterado para "Transformar ideia em desafios", tendo em vista a possibilidade de existir mais de um desafio;
 - A descrição da atividade foi alterada para "Transformar a ideia inicial em um desafio a ser enfrentado, o que resultará na definição do objetivo a ser atingido pelo aplicativo.", para explicitar que, neste momento, durante a aplicação da abordagem, o objetivo está sendo definido.
- Na atividade "Transformar desafio em questões":
 - A descrição da atividade foi alterada para "Transformar o desafio em questões relacionadas ao negócio, o que resultará no dimensionamento do escopo.", para explicitar que neste momento o escopo está sendo dimensionado;

- Foi adicionada a sugestão para a atividade "Sugere-se o relacionamento entre desafios, questões e funcionalidades."
- Na atividade "Categorizar e priorizar questões":
 - O nome da atividade foi alterado para "Priorizar questões", pois como todas as questões a serem respondidas neste momento da abordagem são de natureza do negócio do aplicativo a ser desenvolvido, a categorização se torna irrelevante;
 - Para que a priorização ocorra de forma mais natural e precisa, a ordem foi invertida com a atividade "Responder questões" para que, ao chegar na priorização, o desenvolvedor possa ter todas as respostas que precisa para definir o que é mais importante dentro de seu aplicativo.
- Nas atividades "Prototipar solução" e "Consolidar solução":
 - A atividade foi unida a atividade "Consolidar solução", por ambas possuírem objetivos similares e tendo em vista o caráter de confirmação do que é visto nos protótipos.

Após tais alterações, o fluxo da segunda versão da abordagem, a ser utilizada no segundo ciclo de pesquisa-ação, é o que pode ser visto na Figura 26.

6.2.2 Ciclo 2

Após as modificações realizadas ao fim do primeiro ciclo, a disposição das atividades dentro da pesquisa-ação para o segundo ciclo podem ser vistas na Figura 27.

De forma similar ao realizado no fim do primeiro ciclo, após a execução do segundo ciclo, foi realizada uma análise qualitativa sobre os resultados obtidos.

Por meio da análise qualitativa, foi possível verificar o nível de relevância de determinada atividade para o desenvolvedor. No segundo ciclo, já com a adição do novo desenvolvedor, esse nível pode ser visto na Figura 28.

Ainda de forma qualitativa, outra análise foi realizada, com base nas observações documentadas pelos desenvolvedores, que podem ser vistas no Apêndice C.

Por ser o segundo ciclo de pesquisa-ação, já mais estável, com bases em ambas fontes de dados, as alterações realizadas foram estas:

- Na atividade "Transformar desafios em questões":
 - A descrição da atividade foi alterada para "Transformar o desafio em questões relacionadas ao negócio, o que resultará no dimensionamento do escopo.", para

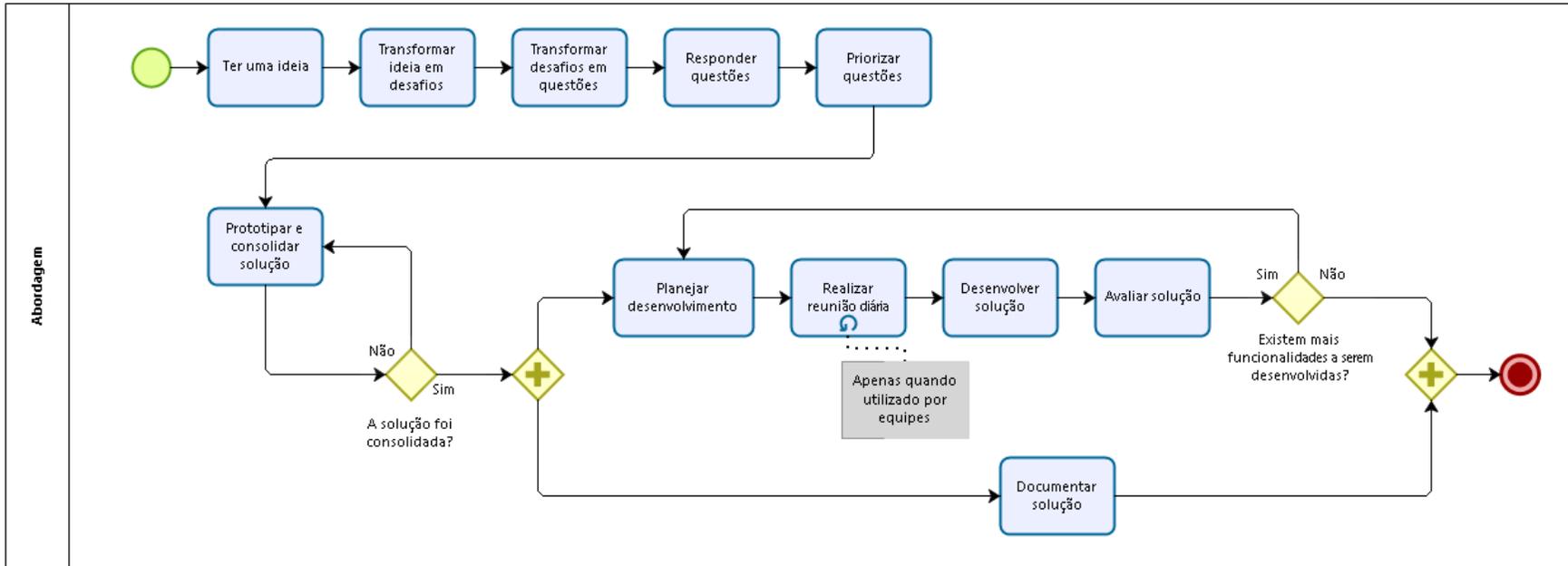


Figura 26 – Fluxo da segunda versão da abordagem. Fonte: Autor.

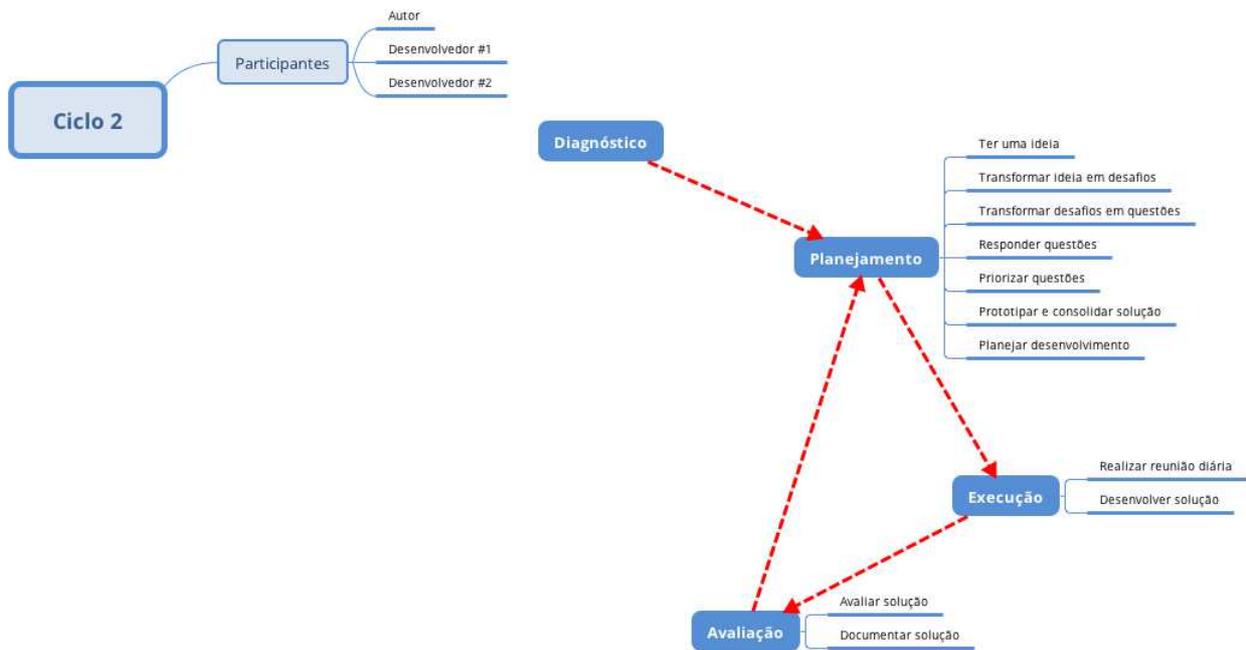


Figura 27 – Atividades no segundo ciclo. Fonte: Autor

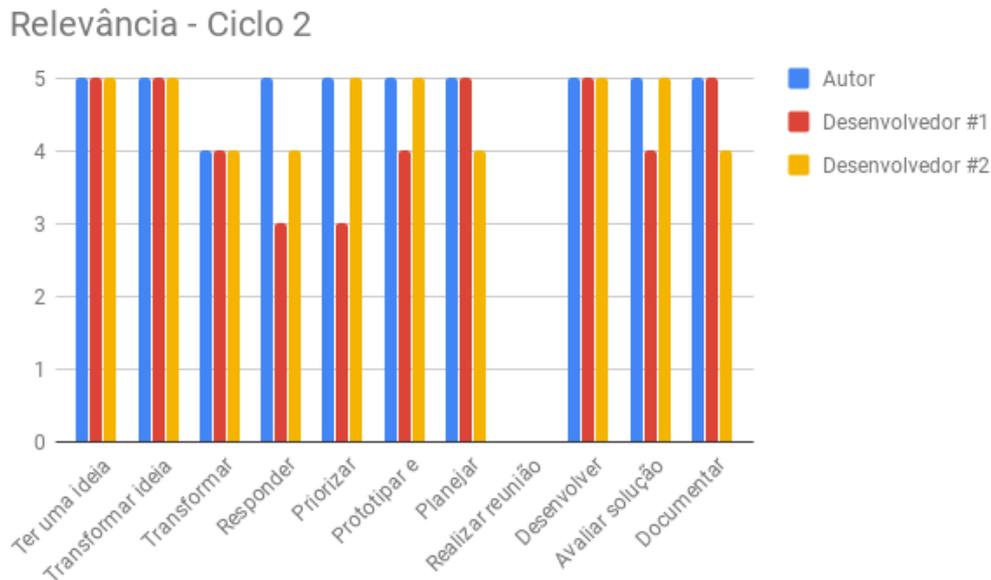


Figura 28 – Relevância das atividades para os desenvolvedores no segundo ciclo. Fonte: Autor

deixar mais claro que as perguntas devem possuir caráter não-técnico e sim dos requisitos do aplicativo.

- Na atividade "Priorizar questões":
 - A descrição da atividade foi alterada para "Priorizar as questões criadas con-

forme sua importância para o usuário.", para deixar mais claro que o peso a ser dado durante a priorização deve refletir a importância para o usuário, não para o desenvolvedor.

- Nas atividades "Realizar reunião diária" e "Desenvolver solução":
 - Como em todos os casos analisados a atividade "Realizar reunião diária" não foi aplicada, por se tratar de desenvolvimento individual, esta foi removida. Em contrapartida, a sugestão da atividade "Desenvolver solução" foi alterada para "Sugere-se utilizar a linguagem Swift. Além disso, ao realizar o desenvolvimento em equipes, sugere-se que sejam realizadas reuniões diárias para transferência de conhecimento dentro da equipe.", de modo a orientar equipes que utilizem a abordagem.

Após tais alterações, o fluxo da terceira versão da abordagem, agora estável, será apresentado e detalhado em sua completude, não apenas nos pontos de alteração, na seção 6.3.

6.3 Atividades da abordagem CBAMD final

Após as modificações das atividades ocorridas durante os ciclos de pesquisa-ação, explicado nas seções anteriores e visível na Figura 29, o fluxo apresentado na Figura 30 representa a abordagem CBAMD (Cabraia's Beginners Approach for Mobile Development) em sua versão final.

6.3.1 Descrição das atividades

Utilizando a estrutura apresentada na seção 5.2, as atividades da abordagem CBAMD, em sua versão final, são detalhadas a seguir.

- **Ter uma ideia:**

Orientando-se por [Wooldridge et al. \(2011\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Introduzir ideia do produto;
- Base no CBL: Ter uma ideia;
- Princípio(s) *Lean*: -
- Descrição: Escolher a ideia inicial para o desenvolvimento da aplicação;
- Justificativa: Tendo em vista seu público-alvo e o contexto em que seu aplicativo irá atuar, ter uma boa ideia é fundamental para que o aplicativo a

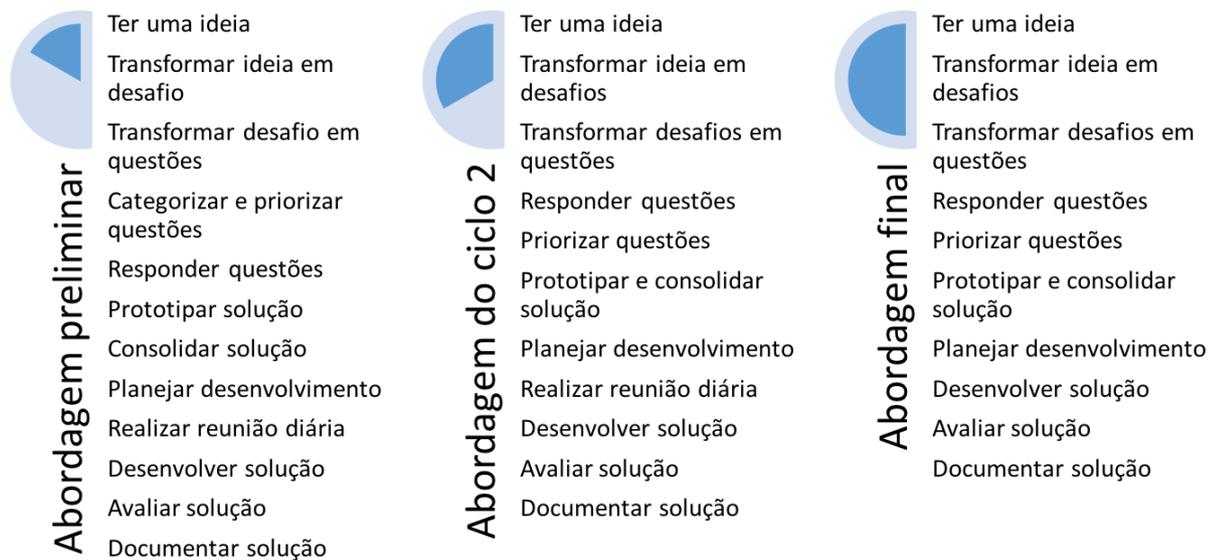


Figura 29 – Evolução das atividades ao longo do trabalho. Fonte: Autor

ser desenvolvido tenha sucesso no mercado. Ao ter tal ideia é dado início ao projeto;

- Exemplo: O desenvolvedor pode ter uma ideia de aplicativo envolvendo seus treinos e saúde.
- Sugestão: -

- **Transformar ideia em desafios:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Realizar questionamento essencial e Transformar questão essencial em desafio;
- Princípio(s) *Lean*: [P2](#) e [P7](#);
- Descrição: Transformar a ideia inicial em um desafio a ser enfrentado, o que resultará na definição do objetivo a ser atingido pelo aplicativo;
- Justificativa: A execução dessa atividade se faz necessária para que a ideia de aplicativo existente seja transformada em um objetivo claro para o desenvolvedor;

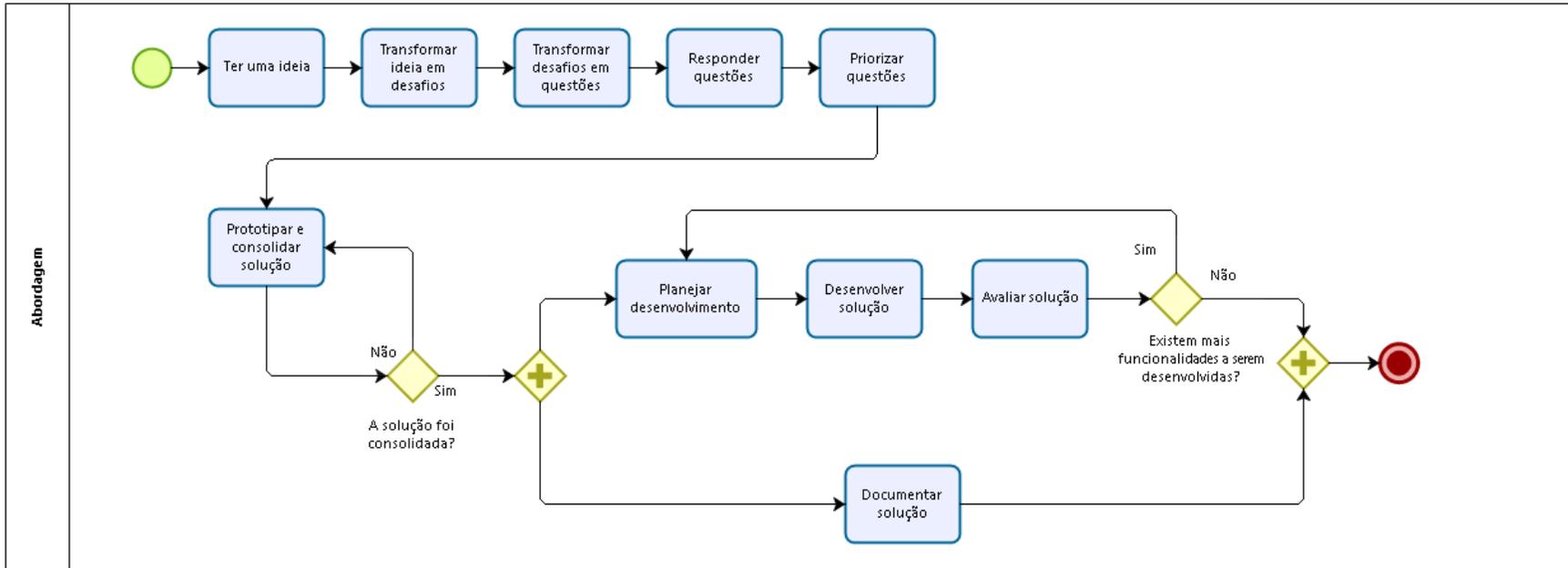


Figura 30 – Fluxo da abordagem em sua versão final. Fonte: Autor.

- Exemplo: Transformar a ideia de um aplicativo relacionado com a área da saúde e que envolve seus treinos em um desafio simples que é "Tornar meu usuário mais saudável!".
- Sugestão: -

- **Transformar desafios em questões:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Criar questões guia;
- Princípio(s) *Lean*: [P2](#) e [P7](#);
- Descrição: Transformar o desafio em questões relacionadas ao negócio, o que resultará no dimensionamento do escopo;
- Justificativa: É uma atividade necessária para que os requisitos comecem a ser trabalhados no início do projeto, ao passo que, se inicia o aprendizado;
- Exemplo: Transformar o desafio de tornar o usuário mais saudável em uma questão como Q1: "O que preciso fazer para ser mais saudável?";
- Sugestão: Sugere-se o relacionamento entre desafios, questões e funcionalidades.

- **Responder questões:**

Orientando-se por [Dey \(2001\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Realizar atividades guia;
- Princípio(s) *Lean*: [P2](#) e [P7](#);
- Descrição: Possuindo como objetivo elucidar o que o aplicativo deverá agregar de valor para o usuário, as questões criadas devem ser respondidas;
- Justificativa: É necessário para iniciar o aprendizado e a coleta de conhecimento, ao mesmo tempo que confere as respostas para as questões levantadas anteriormente, trazendo informações vitais para o funcionamento do aplicativo de forma clara;
- Exemplo: Com base na ideia obtida e desafio definido a resposta para a Q1 pode ser: "Preciso registrar meus dados de treino".
- Sugestão: -

- **Priorizar questões:**

Orientando-se por [Karlsson, Wohlin e Regnell \(1998\)](#), [Berander e Andrews \(2005\)](#), [Davis \(2003\)](#) e [Jacobi et al. \(2014\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos;
- Base no CBL: Categorizar e priorizar questões guia;
- Princípio(s) *Lean*: P2 e P7;
- Descrição: Priorizar as questões criadas conforme sua importância para o usuário;
- Justificativa: É necessário para que a equipe de desenvolvimento identifique o que é mais importante e ordene sua lista de trabalho, tendo em vista possíveis limitações do projeto, como, por exemplo, tempo e custos;
- Exemplo: -
- Sugestão: Sugere-se utilizar o teste dos 100 dólares. Além disso, pode ser analisado o que deve, deveria e poderia ter, além do que não terá por enquanto.

- **Prototipar e consolidar solução:**

Orientando-se por Pressman (2005) e Myers, Sandler e Badgett (2011), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Levantar requisitos e Gerar *backlog* do produto;
- Base no CBL: Analisar as lições aprendidas, Prototipar conceito da solução, Testar conceito da solução e Refinar conceito da solução;
- Princípio(s) *Lean*: P1 e P6;
- Descrição: Prototipar solução para validação com os *stakeholders*, utilizando o conceito de *wireframe* para obter apelo visual, e consolidar a solução com base em seu *feedback*;
- Justificativa: É de suma importância para que a consolidação do conhecimento tome forma e este possa ser traduzido em algo visual, o que irá, posteriormente, permitir a validação com os *stakeholders* de forma simples e intuitiva;
- Exemplo: Um protótipo de tela inicial pode visto ao lado de seu aplicativo desenvolvido na Figura 22.
- Sugestão: Sugere-se utilizar a prototipação em papel.

- **Planejar desenvolvimento:**

Orientando-se por Sharma e Hasteer (2016), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Planejar *sprint*, Priorizar atividades e Gerar *backlog* da *sprint*;
- Base no CBL: -
- Princípio(s) *Lean*: P3, P5, P6 e P7;

- Descrição: Definir quais serão os seus passos para que a ideia seja executada e se obtenha um aplicativo como produto final, observando a priorização definida anteriormente;
- Justificativa: É importante e necessário para organizar o que é planejado naquela *sprint* e, posteriormente, permitir a análise da relação entre planejado e feito.
- Exemplo: -
- Sugestão: -

- **Desenvolver solução:**

Orientando-se por [Cockburn \(2002\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Realizar desenvolvimento;
- Base no CBL: Implementar solução;
- Princípio(s) *Lean*: [P1](#), [P4](#) e [P5](#);
- Descrição: Codificar a solução consolidada com base nos protótipos desenhados;
- Justificativa: Esta é uma atividade fundamental dentro do desenvolvimento do projeto, tendo em vista que o desenvolvimento do aplicativo é o que torna todas as outras atividades necessárias.
- Exemplo:

```
print("Let's get healthy!")
```

- Sugestão: Sugere-se utilizar a linguagem Swift. Além disso, ao realizar o desenvolvimento em equipes, sugere-se que sejam realizadas reuniões diárias para transferência de conhecimento dentro da equipe.

- **Avaliar solução:**

Orientando-se por [Pikkarainen et al. \(2008\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: Realizar revisão da *sprint* e Realizar retrospectiva da *sprint*;
- Base no CBL: Avaliar impacto no desafio;
- Princípio(s) *Lean*: [P6](#) e [P7](#);
- Descrição: Realizar uma comparação entre o protótipo consolidado e o aplicativo final para avaliação da solução desenvolvida;
- Justificativa: Ao fim do desenvolvimento, é comum que algumas pequenas alterações possam ter ocorrido, por isso a atividade é necessária para garantir que o desenvolvido esteja em conformidade com o esperado.

- Exemplo: -
- Sugestão: -

- **Documentar solução:**

Orientando-se por [Forward e Lethbridge \(2002\)](#) e [Briand \(2003\)](#), junto ao mapeamento, esta atividade foi criada.

- Base no Scrum: -
- Base no CBL: -
- Princípio(s) *Lean*: [P6](#) e [P7](#);
- Descrição: Manter a solução documentada por meio do registro dos resultados e observações de cada atividade executada;
- Justificativa: É necessário para garantir que o conhecimento adquirido sobre a solução não seja perdido caso ocorra alguma mudança no responsável por sua manutenção.
- Exemplo: -
- Sugestão: -

6.4 Considerações parciais

Esse capítulo conferiu uma visão mais detalhada sobre como foi conduzida a evolução da abordagem preliminar à abordagem em sua versão final. Com base em pesquisa, foram descritos os ciclos envolvendo as etapas de Diagnóstico, Planejamento, Execução e Avaliação. Por fim, a abordagem CBAMD, em sua versão final, foi apresentada em termos de fluxo bem como em termos descritivos, considerando cada atividade prevista.

7 Considerações finais

Este trabalho aponta que muitas pessoas, sem o conhecimento formal no campo da Ciência da Computação ou Engenharia de Software, dedicam-se ao desenvolvimento de aplicativos para dispositivos móveis. Apresenta que apesar de metodologias serem criadas com o intuito de facilitar esse desenvolvimento, nem sempre essas metodologias se traduzem em algo fácil para o desenvolvedor que está iniciando seu aprendizado.

A abordagem proposta por este trabalho possuiu como objetivo facilitar a entrada de novas pessoas no universo dos dispositivos móveis, procurando ser considerada de fácil aprendizado. A questão de pesquisa definida foi: *Como a aplicação conjunta de conceitos oriundos do Scrum, Lean Thinking e CBL pode contribuir na criação de uma abordagem capaz de apoiar iniciantes no desenvolvimento de aplicativos para dispositivos móveis iOS?* Para procurar responder essa questão foi realizado um mapeamento inicial de literatura, onde, a partir deste, buscou-se encontrar as atividades consideradas ideais para guiar o desenvolvimento de um aplicativo por alguém sem experiência, o que levou a criação da abordagem preliminar, conforme apresentado no capítulo 5. Após a criação da abordagem preliminar, foi realizada a validação desta, por meio do desenvolvimento de aplicativos, utilizando ciclos dentro de uma pesquisa-ação. Dois ciclos foram executados, sendo que após cada um destes, os resultados eram analisados, qualitativamente, para a proposta de melhorias, por sua vez aplicadas na próxima versão da abordagem. Após a execução destes ciclos, foi criada a versão final da abordagem, batizada de CBAMD (acrônimo para *Cambraia's Beginners Approach for Mobile Development*), conforme apresentado no capítulo 6.

Este trabalho possuiu um objetivo geral, que por sua vez foi modularizado em objetivos específicos para uma condução mais apropriada do trabalho, conforme apresentado na seção 1.5. O objetivo geral foi definido como:

Elaboração de uma abordagem capaz de apoiar iniciantes no desenvolvimento de aplicativos para dispositivos móveis iOS, a qual será orientada por Scrum, Lean Thinking e CBL;

Já os objetivos específicos foram:

- Investigar o que já existe sobre a utilização conjunta do CBL e Scrum;
- Definir, por meio de revisão bibliográfica, aspectos do CBL a serem utilizados no desenvolvimento da abordagem;
- Definir, por meio de revisão bibliográfica, aspectos do Scrum a serem utilizados no desenvolvimento da abordagem;

- Definir um conjunto de atividades, por meio de ciclos de pesquisa-ação, que, em conjunto, serão caracterizadas como a abordagem proposta;
- Implementação, de um ou mais aplicativos, pelo Autor e/ou terceiros, orientando-se pela abordagem proposta;
- Avaliação dos resultados coletados durante a implementação.

Após a execução das atividades previstas no cronograma, mencionadas no segundo parágrafo deste capítulo, todos esses objetivos podem ser considerados alcançados.

7.1 Trabalhos futuros

Ainda que os objetivos propostos para este trabalho tenham sido alcançados, o Autor considera que novas pesquisas podem ser realizadas utilizando a abordagem como alicerce.

Seguem algumas sugestões, não devendo-se limitar a estas, para trabalhos futuros:

- Durante a execução deste trabalho, foi analisado como a CBAMD se comporta durante o desenvolvimento de aplicativos no ecossistema iOS. Uma oportunidade de pesquisa consiste na avaliação do seu funcionamento no desenvolvimento para dispositivos Android, o sistema operacional com o maior número de usuários atualmente¹;
- Outro ponto válido de nota, é que durante a execução deste trabalho foi avaliado o uso da CBAMD por desenvolvedores individuais. Outra oportunidade de pesquisa consiste na avaliação de seu uso por equipes de desenvolvimento;
- Outro fator que possui margem para exploração é a avaliação de desempenho e eficiência da CBAMD. Ao coletar dados que auxiliem na aferição desses indicadores, estes podem ser comparados com dados provenientes do uso de outras abordagens já utilizadas pelo mercado;
- Uma das características deste trabalho foi a sua preocupação com a simplicidade, tendo em vista seu público-alvo iniciante. Outra oportunidade de pesquisa pode ser a avaliação da aplicação da CBAMD, em instituições educacionais, com o objetivo de auxiliar no ensino de programação.

¹ <http://gs.statcounter.com/os-market-share/>. Acesso em: 24/06/2018.

Referências

- ATOM. 2017. <<http://flight-manual.atom.io/getting-started/sections/why-atom/>>. Acesso em: 2017-06-03. Citado na página 46.
- AWAD, M. A comparison between agile and traditional software development methodologies. *University of Western Australia*, 2005. Citado 2 vezes nas páginas 30 e 35.
- BECKER, F. O que é construtivismo. *Revista de educação AEC, Brasília*, v. 21, n. 83, p. 7–15, 1992. Citado na página 40.
- BERANDER, P.; ANDREWS, A. Requirements prioritization. *Engineering and managing software requirements*, Springer, v. 11, n. 1, p. 79–101, 2005. Citado 2 vezes nas páginas 64 e 82.
- BIZAGI. 2017. <<http://www.bizagi.com/pt/produtos/bpm-suite/modeler/>>. Acesso em: 2017-05-13. Citado na página 43.
- BOURQUE, P.; FAIRLEY, R. E. et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. [S.l.]: IEEE Computer Society Press, 2014. Citado na página 29.
- BRIAND, L. C. Software documentation: how much is enough? In: IEEE. *Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on*. [S.l.], 2003. p. 13–15. Citado 2 vezes nas páginas 68 e 85.
- COCKBURN, A. *Agile software development*. [S.l.]: Addison-Wesley Boston, 2002. v. 177. Citado 2 vezes nas páginas 67 e 84.
- DAVIS, A. M. The art of requirements triage. *Computer, IEEE*, v. 36, n. 3, p. 42–49, 2003. Citado 2 vezes nas páginas 64 e 82.
- DEY, A. K. Understanding and using context. *Personal and ubiquitous computing*, Springer-Verlag, v. 5, n. 1, p. 4–7, 2001. Citado 4 vezes nas páginas 64, 65, 80 e 82.
- DIESTE, O.; PADUA, A. G. Developing search strategies for detecting relevant experiments for systematic reviews. In: IEEE. *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. [S.l.], 2007. p. 215–224. Citado 2 vezes nas páginas 25 e 96.
- FONSECA, J. J. S. Metodologia da pesquisa científica. 2002. Citado na página 52.
- FORWARD, A.; LETHBRIDGE, T. C. The relevance of software documentation, tools and technologies: a survey. In: ACM. *Proceedings of the 2002 ACM symposium on Document engineering*. [S.l.], 2002. p. 26–33. Citado 2 vezes nas páginas 68 e 85.
- GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de pesquisa*. [S.l.]: Plageder, 2009. Citado 2 vezes nas páginas 51 e 52.

- GIL, A. C. *Métodos e técnicas de pesquisa social*. [S.l.]: 6. ed. Editora Atlas SA, 2008. Citado na página 51.
- GIT. 2017. <<https://git-scm.com/about/>>. Acesso em: 2017-06-03. Citado na página 47.
- GITHUB. 2017. <<https://guides.github.com/activities/hello-world/>>. Acesso em: 2017-06-03. Citado na página 47.
- GITHUB. 2017. <<https://github.com/about/>>. Acesso em: 2017-06-03. Citado na página 47.
- HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. Evaluating cross-platform development approaches for mobile applications. In: SPRINGER. *International Conference on Web Information Systems and Technologies*. [S.l.], 2012. p. 120–138. Citado na página 41.
- IIBA, A. guide to the business analysis body of knowledge (babok guide). *International Institute of Business Analysis (IIBA)*, 2009. Citado na página 24.
- JACOBI, F. et al. Towards a design model for interdisciplinary information systems curriculum development, as exemplified by big data analytics education. 2014. Citado na página 82.
- JOORABCHI, M. E.; MESBAH, A.; KRUCHTEN, P. Real challenges in mobile app development. In: IEEE. *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*. [S.l.], 2013. p. 15–24. Citado na página 24.
- KALEEL, S. B.; HARISHANKAR, S. Applying agile methodology in mobile software engineering: Android application development and its challenges. *Ryerson University, Digital Commons@ Ryerson, Computer Science Tech. Rep., Paper*, v. 4, 2013. Citado 2 vezes nas páginas 23 e 24.
- KARLSSON, J.; WOHLIN, C.; REGNELL, B. An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, Elsevier, v. 39, n. 14-15, p. 939–947, 1998. Citado 2 vezes nas páginas 64 e 82.
- KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004. Citado 2 vezes nas páginas 25 e 95.
- LATEX. 2017. <<https://www.latex-project.org/about/>>. Acesso em: 2017-06-03. Citado na página 48.
- LEI. *A Brief History of Lean*. 2017. <<https://www.lean.org/WhatsLean/History.cfm>>. Acesso em: 2017-10-07. Citado na página 35.
- MYERS, G. J.; SANDLER, C.; BADGETT, T. *The art of software testing*. [S.l.]: John Wiley & Sons, 2011. Citado 2 vezes nas páginas 65 e 83.
- NEDRE, N. *How to choose between Agile and Lean, Scrum and Kanban — which methodology is the best?* 2015. <<https://realtimeboard.com/blog/choose-between-agile-lean-Scrum-kanban/>>. Acesso em: 2017-10-12. Citado na página 36.

- NICHOLS, M.; CATOR, K.; TORRES, M. Challenge based learner user guide. *Redwood City, CA: Digital Promise.*, 2016. Citado 7 vezes nas páginas 24, 36, 37, 38, 39, 40 e 61.
- PAPERT, S.; HAREL, I. Situating constructionism. *Constructionism*, v. 36, n. 2, p. 1–11, 1991. Citado na página 40.
- PETERSEN, K. et al. Action research as a model for industry-academia collaboration in the software engineering context. In: ACM. *Proceedings of the 2014 international workshop on Long-term industrial collaboration on software engineering*. [S.l.], 2014. p. 55–62. Citado na página 52.
- PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Elsevier, v. 64, p. 1–18, 2015. Citado 4 vezes nas páginas 25, 95, 96 e 97.
- PIKKARAINEN, M. et al. The impact of agile practices on communication in software development. *Empirical Software Engineering*, Springer, v. 13, n. 3, p. 303–337, 2008. Citado 2 vezes nas páginas 68 e 84.
- POPPENDIECK, M.; POPPENDIECK, T. *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. [S.l.]: Addison-Wesley, 2003. Citado na página 35.
- PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005. Citado 2 vezes nas páginas 65 e 83.
- REBOUÇAS, M. et al. An empirical study on the usage of the swift programming language. In: IEEE. *Software Analysis, Evolution, and Reengineering (SANER), 2016 IEEE 23rd International Conference on*. [S.l.], 2016. v. 1, p. 634–638. Citado na página 24.
- REMINDERS. 2017. <<https://support.apple.com/en-us/HT205890/>>. Acesso em: 2017-05-13. Citado na página 44.
- RISING, L.; JANOFF, N. S. The scrum software development process for small teams. *IEEE software*, IEEE, v. 17, n. 4, p. 26–32, 2000. Citado na página 32.
- RUSKEY, F.; WESTON, M. A survey of venn diagrams. *Electronic Journal of Combinatorics*, v. 4, p. 3, 1997. Citado na página 29.
- SANTOS, A. et al. Investigating the adoption of agile practices in mobile application development. In: *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016)*. [S.l.: s.n.], 2016. p. 490–497. Citado na página 23.
- SANTOS, A. R. et al. Combining challenge-based learning and scrum framework for mobile application development. In: ACM. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. [S.l.], 2015. p. 189–194. Citado 2 vezes nas páginas 24 e 41.
- SCHWABER, K.; SUTHERLAND, J. The scrum guide. *Scrum Alliance*, 2016. Citado na página 32.
- SCRUM: metodologia ágil para gestão e planejamento de projetos. 2017. <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 2017-11-13. Citado na página 33.

- SHARMA, S.; HASTEER, N. A comprehensive study on state of scrum development. In: IEEE. *Computing, Communication and Automation (ICCCA), 2016 International Conference on*. [S.l.], 2016. p. 867–872. Citado 7 vezes nas páginas 24, 30, 31, 32, 61, 67 e 83.
- SOLT, P. *Swift vs. Objective-C: 10 reasons the future favors Swift*. 2015. <<https://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>>. Acesso em: 2017-10-18. Citado na página 46.
- STRAY, V. G.; MOE, N. B.; AURUM, A. Investigating daily team meetings in agile software projects. In: IEEE. *Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on*. [S.l.], 2012. p. 274–281. Citado na página 67.
- SWIFT. 2017. <<https://swift.org/about/>>. Acesso em: 2017-06-03. Citado 2 vezes nas páginas 45 e 46.
- WELLS, G. The future of ios development: Evaluating the swift programming language. 2015. Citado 2 vezes nas páginas 24 e 46.
- WHAT is a Product Backlog? 2017. <<https://www.scrum.org/resources/what-is-a-product-backlog>>. Acesso em: 2017-11-13. Citado na página 34.
- WOOLDRIDGE, D. et al. The business of iphone and ipad app development. *Making and marketing Apps that succeed*. Apress, New York, Springer, 2011. Citado 2 vezes nas páginas 62 e 79.
- XANTHOPOULOS, S.; XINOGALOS, S. A comparative analysis of cross-platform development approaches for mobile applications. In: ACM. *Proceedings of the 6th Balkan Conference in Informatics*. [S.l.], 2013. p. 213–220. Citado na página 23.
- XCODE. 2017. <<https://developer.apple.com/support/xcode/>>. Acesso em: 2017-06-03. Citado na página 46.
- XMIND. 2017. <<http://www.xmind.net/features/>>. Acesso em: 2017-10-23. Citado na página 44.
- ZOTERO. 2017. <<https://www.zotero.org/about/>>. Acesso em: 2017-06-03. Citado 2 vezes nas páginas 48 e 49.

Apêndices

APÊNDICE A – Revisão Sistemática

A.1 Protocolo de Revisão Sistemática

Para a elaboração do protocolo de revisão sistemática e sua posterior execução, foram seguidas as diretrizes encontradas em [Petersen, Vakkalanka e Kuzniarz \(2015\)](#) e [Kitchenham \(2004\)](#) com adaptações para o contexto de pesquisa.

A.1.1 Perguntas de pesquisa

- Existem trabalhos que relacionam o uso do CBL e metodologias ágeis?
- Quais práticas ágeis são utilizadas em conjunto com o CBL?
- São positivos os resultados dessa combinação?

A.1.2 Palavras-chave

Para a definição das palavras-chave foi utilizado o PICO, proposto por [Kitchenham \(2004\)](#), um método para analisar determinada questão por quatro visões diferentes, sendo estas: População, Intervenção, Comparação e Resultados.

Para facilitar a posterior utilização na expressão de busca, foi definido que as palavras-chave, usando o PICO, seriam apresentadas em inglês.

O resultado da aplicação de tal metodologia encontrou os seguintes resultados:

- População: *Mobile*
- Intervenção: *Agile Software Development, CBL*
- Comparação: -
- Resultados: *Process, Approach, Methodology*

A.1.3 Expressão de busca

Com base nas palavras-chave identificadas, e sua tradução para o português, foi criada uma expressão de busca, tendo esta o objetivo de encontrar artigos relacionados aos temas desejados.

A evolução da expressão de busca pode ser observada na Figura 31, sendo que a expressão final é:

(“Agile Software Development” OR “Agile Methodologies” OR “Desenvolvimento Ágil” OR “Metodologias Ágeis”) AND (CBL OR “Challenge Based Learning” OR “Aprendizado Baseado em Desafios”) AND (Mobile OR “Mobile Development” OR “Desenvolvimento Móvel” OR “Desenvolvimento Mobile” OR “Dispositivos Móveis” OR “Software Engineering” OR “Engenharia de Software” OR Framework OR Process OR Processo OR Approach OR Abordagem OR Methodology OR Metodologia)

A.1.4 Base

Conforme [Dieste e Padua \(2007\)](#), é possível, através de uma escolha do pesquisador, utilizar apenas a base de pesquisa Scopus para a realização da revisão sistemática, tendo em vista que a base possui poucos dos problemas analisados em seu estudo e possui um alto nível de efetividade para encontrar artigos.

Sendo assim, para a realização dessa revisão, foi decidido que apenas a base **Scopus** seria utilizada para a busca de artigos, assim evitando a adaptação da expressão de busca para os padrões de diversas bases.

A.1.5 Seleção de artigos

Foi definido um processo de análise de artigos, adaptando-se o que foi utilizado por [Petersen, Vakkalanka e Kuzniarz \(2015\)](#).

Para realizar a seleção dos artigos a serem lidos, foram definidos critérios de inclusão e exclusão de artigos, sendo que sua aplicação é necessária para evitar que artigos indesejados sejam incluídos na pesquisa.

A.1.5.1 Processo de análise de artigos

1. Busca na base
2. Leitura do título e resumo com aplicação dos critérios de seleção
3. Leitura do artigo completo
4. Avaliação da qualidade, no que diz respeito a *clareza, completeza e pertinência*

A.1.5.2 Inclusão

- Artigos que tratam de metodologias ágeis e CBL;
- Artigos no contexto da Engenharia de Software;
- Artigos publicados a partir de 2008.

A.1.5.3 Exclusão

- Artigos que não se encontram em inglês ou português;
- Artigos indisponíveis para leitura.

A.1.6 Avaliação da qualidade

Ao ler os artigos selecionados, é necessário analisar a qualidade destes, sendo que para isso foram definidas perguntas a serem respondidas ao ler os artigos, conforme Petersen, Vakkalanka e Kuzniarz (2015).

As perguntas de avaliação definidas foram:

- O artigo apresenta de forma clara o uso do CBL junto a metodologias ágeis?
- O artigo apresenta em detalhes a sua realização?
- O artigo apresenta resultados relacionados ao tema?

A.2 Expressões de busca e artigos encontrados

Durante a realização da revisão sistemática, a expressão de busca passou por adaptações para que fossem encontrados o maior número possível de artigos relacionados ao tema.

A Figura 31 apresenta os nomes e ID atribuídos dos artigos. detalhes sobre os artigos Já a Figura 32 mostra as variações da expressão de busca, sua evolução e os identificadores dos artigos encontrados com o uso de cada uma.

Nomes dos artigos encontrados	
ID	Nome do artigo
[1]	<i>A comprehensive study on state of Scrum development</i>
[2]	<i>Investigating the adoption of agile practices in mobile application development</i>
[3]	<i>Combining challenge-based learning and scrum framework for mobile application development</i>

Figura 31 – Nomes dos artigos encontrados. Fonte: Autor.

Relação das expressões de busca com os artigos encontrados	
Expressões de busca	Artigos encontrados
("agile software development" OR "desenvolvimento ágil") AND (CBL OR "challenge based learning" OR "aprendizado baseado em desafios") AND (mobile OR "mobile development" OR "desenvolvimento móvel" OR "dispositivos móveis") AND ("software engineering" OR "engenharia de software") AND (framework OR process OR processo OR approach OR abordagem OR methodology OR metodologia)	[1] [2]
("agile software development" OR "agile methodologies" OR "desenvolvimento ágil" OR "metodologias ágeis") AND (CBL OR "challenge based learning" OR "aprendizado baseado em desafios") AND (mobile OR "mobile development" OR "desenvolvimento móvel" OR "dispositivos móveis") AND ("software engineering" OR "engenharia de software") AND (framework OR process OR processo OR approach OR abordagem OR methodology OR metodologia)	[1] [2] [3]
("agile software development" OR "agile methodologies" OR "desenvolvimento ágil" OR "metodologias ágeis") AND (CBL OR "challenge based learning" OR "aprendizado baseado em desafios") AND (mobile OR "mobile development" OR "desenvolvimento móvel" OR "desenvolvimento mobile" OR "dispositivos móveis" OR "software engineering" OR "engenharia de software" OR framework OR process OR processo OR approach OR abordagem OR methodology OR metodologia)	[1] [2] [3]

Figura 32 – Relação das expressões de busca com os artigos encontrados. Fonte: Autor.

APÊNDICE B – Termo de consentimento

Você foi convidado a participar da pesquisa de validação da abordagem de desenvolvimento mobile criada pelo aluno de Engenharia de Software, Leonardo Cambraia Corrêa, como resultado da pesquisa desenvolvida na disciplina Trabalho de Conclusão de Curso 1, no campus Gama da Universidade de Brasília.

Essa pesquisa vai ocorrer durante a disciplina Trabalho de Conclusão de Curso 2, e possui como objetivo a melhoria e a validação da abordagem criada durante o trabalho. Para que este objetivo seja atingido, os participantes irão desenvolver um aplicativo.

Por esta razão, seu consentimento é solicitado para a realização desta pesquisa. Para tanto, é importante que você tenha algumas informações:

- O aplicativo deverá ser desenvolvido na plataforma iOS e linguagem Swift;
- Todos os resultados, positivos ou negativos, decorrentes do desenvolvimento do aplicativo deverão ser compartilhados com o avaliador;
- Os dados coletados durante a validação destinam-se apenas a atividades de pesquisa e de desenvolvimento;
- Os resultados encontrados durante esta pesquisa serão publicados como monografia e em eventos científicos da área, com a garantia de anonimato;
- O consentimento para a realização da pesquisa é uma escolha livre, feita mediante a prestação de todos os esclarecimentos necessários sobre a mesma;
- A realização da avaliação pode ser interrompida a qualquer momento, conforme a vontade do participante. Neste caso, os dados coletados até o momento serão descartados.

De posse das informações acima:

Autorizo a minha participação.

Não autorizo minha participação.

Nome: _____

CPF: _____

Assinatura: _____

Brasília, _____ de Março de 2018.

APÊNDICE C – Aplicação da abordagem

C.1 Aplicativo 1 - LetsDraw

- Atividade 1 “Ter uma ideia”
 - **Resultado:**

Um aplicativo que me permita desenhar.
 - **Observações:**

É uma atividade simples, onde se procura entender o que se quer como resultado;

Sugere-se renomear a atividade para "Definir o problema".

- Atividade 2 “Transformar ideia em desafio”
 - **Resultado:**

Permitir que eu desenhe;

Permitir que eu comece meu desenho do zero;

Permitir que eu compartilhe meu desenho.
 - **Observações:**

Nessa atividade, foi realizada uma certa priorização sobre as necessidades do usuário, o que não ocorreu de forma exclusiva;

Sugere-se que a atividade seja renomeada para "Transformar ideia em desafios", já que mais de um desafio foi identificado.

- Atividade 3 “Transformar desafio em questões”
 - **Resultado:**

Q1: Devo permitir que o desenho ocorra em toda a tela?

Q2: Para que eu comece meu desenho do zero, devo apenas ter uma borracha ou um botão que faça isso automaticamente?

Q3: Como devo compartilhar meu desenho?
 - **Observações:**

Após essa atividade, foi possível imaginar melhor as funcionalidades do aplicativo e seu funcionamento;

Para exemplificar algumas das funcionalidades, foi comum a menção de aplicativos conhecidos e seu funcionamento;

Sugere-se que seja explícita a relação entre desafios, questões e futuras funcionalidades. Por exemplo, para o Desafio 1, temos a Questão 1 e a Funcionalidade 1.

- Atividade 4 “Categorizar e priorizar questões”

- **Resultado:**

- Ao utilizar o teste dos 100 dólares, obteve-se a seguinte priorização:

- Q1: 60 dólares, ou 60% do esforço;

- Q2: 20 dólares, ou 20% do esforço;

- Q3: 20 dólares, ou 20% do esforço.

- **Observações:**

- A priorização das questões em si, sem as respostas, se mostrou um pouco estranha, de modo que a atividade só pôde ser realizada ao já imaginar as respostas para as perguntas definidas.

- Sugere-se que essa atividade seja realizada após a atual atividade 5 "Responder questões", tendo em vista que as respostas dadas na próxima atividade poderiam influir na quantificação do esforço;

- Como todas as questões levantadas e, conseqüentemente, suas respostas são de natureza do negócio do aplicativo, a categorização parece desnecessária, mesmo com a priorização possuindo o seu devido peso e importância;

- Sugere-se que a atividade seja modificada e seu nome seja alterado para "Priorizar questões".

- Atividade 5 “Responder questões”

- **Resultado:**

- Q1: O desenho deve ser realizado em toda a tela disponível, de modo que apenas abaixo dos seletores de cores este não seja exibido;

- Q2: Tanto a opção de borracha como a opção de limpar toda a tela devem existir, tendo em vista que apenas alguns trechos do desenho podem ser apagados ou este em sua completude;

- Q3: É necessário que seja possível salvar o desenho no rolo da câmera, porém, o envio para outros via aplicativos de mensagens pode ser algo interessante.

- **Observações:**

- Sugere-se alterar a ordem da atividade com a atividade 4 "Categorizar e priorizar questões".

- Atividade 6 “Prototipar solução”
 - **Resultado:**

O protótipo foi desenhado em uma folha de papel e pode ser visto na Figura 33.
 - **Observações:**

Não se mostrou necessário a utilização de um aplicativo para automatização do protótipo, tendo em vista que este possuiu apenas uma tela.

- Atividade 7 “Consolidar solução”
 - **Resultado:**

O protótipo foi validado.
 - **Observações:**

Essa atividade se mostrou um pouco redundante, dada que a aprovação do protótipo é realizada em sua própria atividade.

Sugere-se que ambas as atividades "Prototipar solução" e "Consolidar solução" sejam unidas em uma única atividade englobando ambos os objetivos.

- Atividade 8 “Planejar desenvolvimento”
 - **Resultado:**

Com base nas três questões definidas, três funcionalidades foram planejadas. Além das próprias funcionalidades, alguns critérios de aceitação foram adicionados, quando necessários. As funcionalidades e critérios definidos são:

F1: Permitir desenhar na tela:

F1C1: Permitir desenhar utilizando a cor preta por padrão;

F1C2: Permitir desenhar utilizando algumas cores ao selecionar seus devidos botões.

F2: Permitir apagar o desenho presente na tela:

F2C1: Permitir apagar o desenho por meio de uma borracha;

F2C2: Permitir apagar o desenho de toda a tela por meio de um botão.

F3: Permitir compartilhar o desenho presente na tela:

F3C1: Permitir o salvamento do desenho presente na tela;

F3C2: Permitir o compartilhamento do desenho presente na tela.
 - **Observações:**

Seguindo uma das sugestões apresentadas em atividades anteriores, foi definido que existiria uma funcionalidade para cada questão definida.

Sugere-se que essa orientação esteja explícita no comando das atividades.

- Atividade 9 “Realizar reunião diária”

- **Resultado:**

- Não foi realizada por ser apenas um desenvolvedor.

- **Observações:**

-

- Atividade 10 “Desenvolver solução”

- **Resultado:**

- Todas as funcionalidades foram desenvolvidas com sucesso¹.

- O desenvolvimento iniciou-se com a adição dos componentes da tela, utilizando o protótipo como base. Após a adição destes componentes, foi realizada a criação da comunicação entre tela e código. Em seguida, as funcionalidades começaram a ser desenvolvidas, conforme a importância e priorização dadas.

- **Observações:**

-

- Atividade 11 “Avaliar solução”

- **Resultado:**

- Analisando de forma qualitativa, é possível verificar que a solução desenvolvida cumpre o estabelecido nas atividades anteriores e, ao analisar sua aderência com o protótipo criado, a solução foi aprovada.

- O aplicativo desenvolvido pode ser visto ao lado de seu protótipo na Figura 33.

- **Observações:**

- Após o desenvolvimento, testes funcionais foram realizados em um dispositivo físico para atestar a conformidade com os requisitos, foi neste momento que alguns ajustes tiveram de ser feitos, como, por exemplo, a adição de solicitação de acesso ao dispositivo para o salvamento de imagens no rolo da câmera.

- Sugere-se que seja tornado explícito na atividade a necessidade da realização de testes da aplicação no próprio dispositivo.

- Atividade 12 “Documentar solução”

- **Resultado:**

- Por conter os resultados, junto a algumas observações, o conteúdo de cada seção deste apêndice é considerado a documentação de seu respectivo aplicativo desenvolvido.

¹ O código-fonte deste aplicativo encontra-se disponível no seguinte repositório: <https://github.com/cabraia/AppsTCC>.

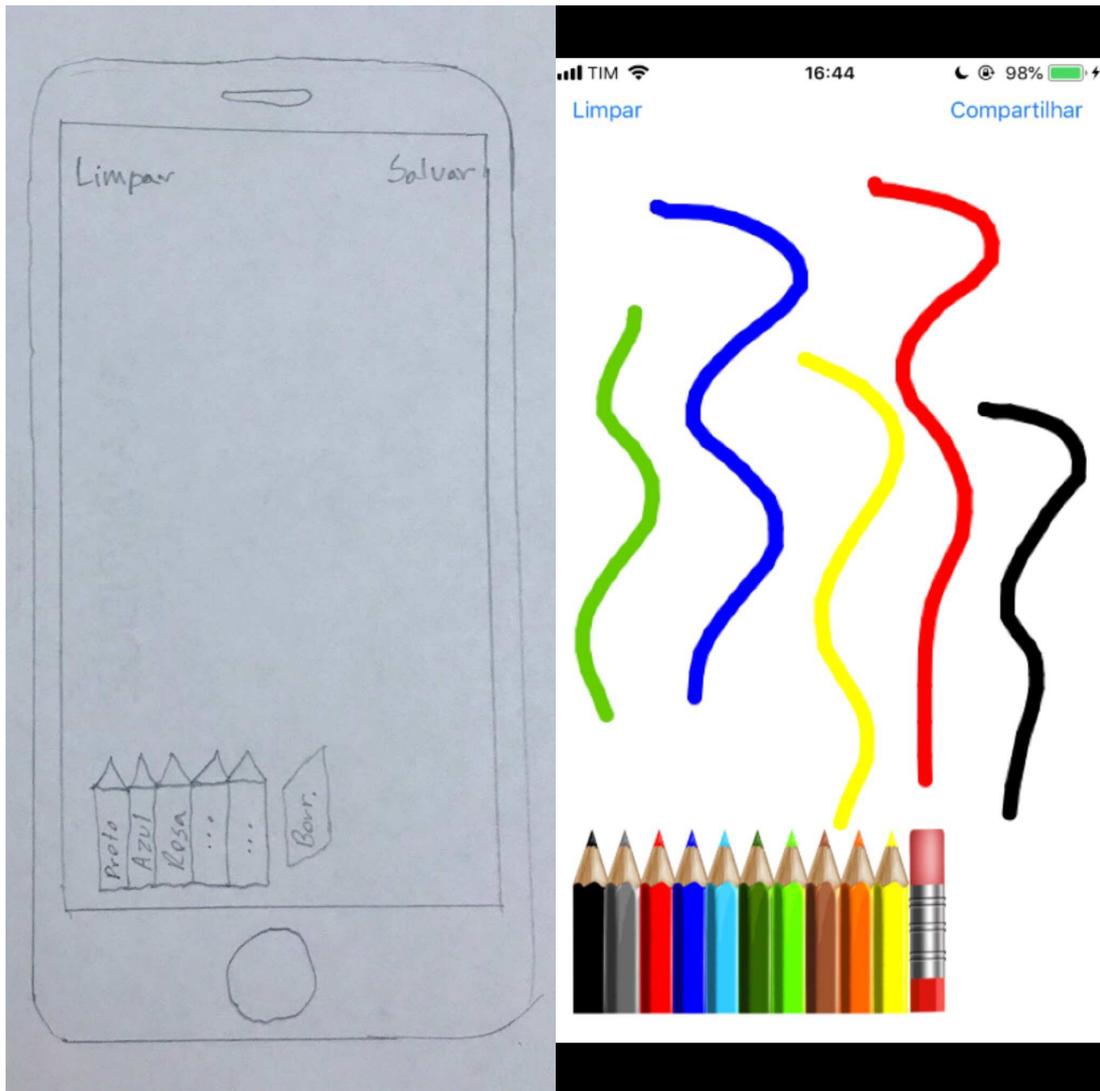


Figura 33 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Autor

– **Observações:**

Durante toda a aplicação da abordagem foi realizada a coleta de informações sobre cada atividade executada, de modo que seus resultados e observações foram registrados.

C.2 Aplicativo 2 - DELStar

● Atividade 1 “Ter uma ideia”

– **Resultado:**

Ferramenta para converter valores iniciais de resistores R_1 , R_2 e R_3 configurados na forma estrela para R_i , R_j e R_k na configuração delta, ou vice versa.

– **Observações:**

Descrição suficiente para o entendimento.

- Atividade 2 “Transformar ideia em desafio”
 - **Resultado:**
Tornar prático e fácil o cálculo das conversões delta-estrela de resistores.
 - **Observações:**
Deixar mais claro que essa etapa envolve a definição do objetivo.
- Atividade 3 “Transformar desafio em questões”
 - **Resultado:**
Q1: Qual é a relação de transformação que se aplica na conversão delta-estrela?
 - **Observações:**
Deixar mais claro que essa etapa envolve o dimensionamento do escopo.
- Atividade 4 “Categorizar e priorizar questões”
 - **Resultado:**
Q1: Qual é a relação de transformação que se aplica na conversão delta-estrela?
 - **Observações:**
Lembrar de fazer as questões primeiro para em seguida encontrar as respostas;
Atividade só é necessária para múltiplas questões.
- Atividade 5 “Responder questões”
 - **Resultado:**
O aplicativo deverá obedecer as seguintes fórmulas:
 - * $Z_a = (Z_1 Z_2 + Z_2 Z_3 + Z_1 Z_3) / Z_1$
 - * $Z_b = (Z_1 Z_2 + Z_2 Z_3 + Z_1 Z_3) / Z_2$
 - * $Z_c = (Z_1 Z_2 + Z_2 Z_3 + Z_1 Z_3) / Z_3$
 - * $Z_1 = (Z_b Z_c) / (Z_a + Z_b + Z_c)$
 - * $Z_2 = (Z_a Z_c) / (Z_a + Z_b + Z_c)$
 - * $Z_3 = (Z_a Z_b) / (Z_a + Z_b + Z_c)$
 - **Observações:**
-
- Atividade 6 “Prototipar solução”
 - **Resultado:**
O protótipo foi desenhado em um editor de texto e pode ser visto na Figura 34.

- **Observações:**
Preferi criar o protótipo no computador.
- Atividade 7 “Consolidar solução”
 - **Resultado:**
-
 - **Observações:**
Não se aplica a todos os casos;
Juntar à atividade anterior.
- Atividade 8 “Planejar desenvolvimento”
 - **Resultado:**
Criar itens de interface, botões e caixa de texto;
Conectar interface ao código;
Codificar.
 - **Observações:**
-
- Atividade 9 “Realizar reunião diária”
 - **Resultado:**
Não foi realizada por ser apenas um desenvolvedor.
 - **Observações:**
-
- Atividade 10 “Desenvolver solução”
 - **Resultado:**
Foi desenvolvido o aplicativo².
 - **Observações:**
-
- Atividade 11 “Avaliar solução”
 - **Resultado:**
Analisando de forma qualitativa, é possível verificar que a solução desenvolvida cumpre o estabelecido nas atividades anteriores e, ao analisar sua aderência com o protótipo criado, a solução foi aprovada.

² O código-fonte deste aplicativo encontra-se disponível no seguinte repositório: <https://github.com/cambraia/AppsTCC>.

O aplicativo desenvolvido pode ser visto ao lado de seu protótipo na Figura 34.

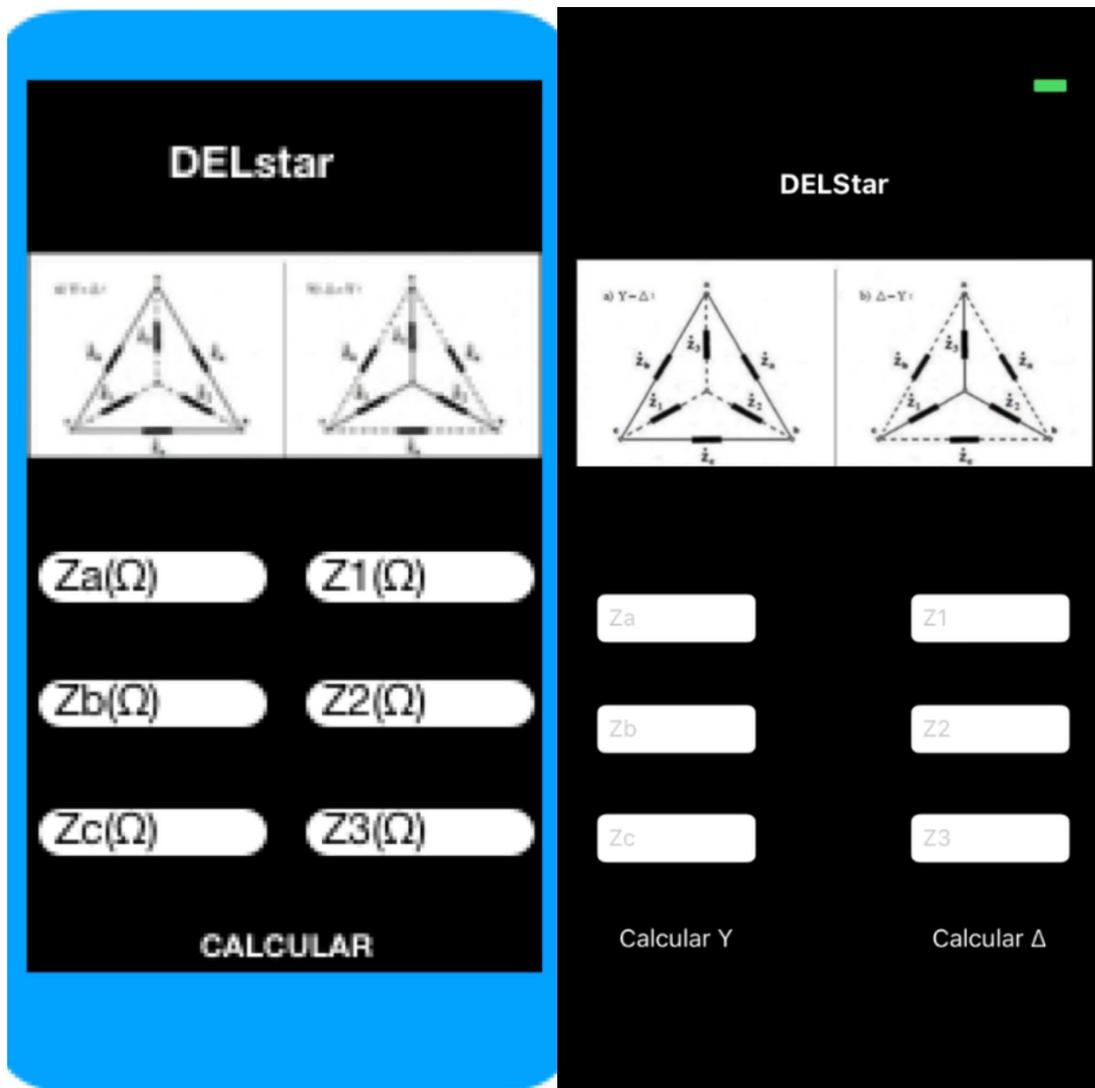


Figura 34 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Desenvolvedor #1

– **Observações:**

Durante o desenvolvimento, foram feitas algumas alterações com o interesse de facilitar o entendimento do usuário, como, por exemplo, dois botões para calcular as duas operações distintas que o aplicativo pode fazer.

• Atividade 12 “Documentar solução”

– **Resultado:**

Por conter os resultados, junto a algumas observações, o conteúdo de cada seção deste apêndice é considerado a documentação de seu respectivo aplicativo desenvolvido.

– **Observações:**

Todas as etapas foram seguidas, gerando esta documentação.

C.3 Aplicativo 3 - LetsSee

● Atividade 1 “Ter uma ideia”

– **Resultado:**

Um aplicativo para identificar objetos.

– **Observações:**

-

● Atividade 2 “Transformar ideia em desafios”

– **Resultado:**

Ver objetos;

Reconhecer os objetos analisados;

Exibir para o usuário os possíveis objetos.

– **Observações:**

-

● Atividade 3 “Transformar desafios em questões”

– **Resultado:**

Q1: Devo utilizar apenas imagens da câmera ou também imagens já salvas?

Q2: Devo limitar os resultados encontrados?

– **Observações:**

Ao transformar os desafios em questões, é comum pensarmos em questões técnicas. Porém o foco da atividade são as perguntas que ajudam a definir as regras de negócio;

Sugere-se adicionar uma explicação extra para essa descrição indicando que isso pode ocorrer.

● Atividade 4 “Responder questões”

– **Resultado:**

Q1: Podem ser utilizadas imagens tiradas naquele momento ou imagens já salvas;

Q2: Um dos métodos que podem ser utilizados para a resolução deste problema é a utilização de *Machine Learning* (ML). Ao utilizar modelos treinados por

ML, é comum encontrar mais de uma associação a determinada imagem. Logo, é interessante que todas as possíveis opções sejam exibidas.

– **Observações:**

-

● Atividade 5 “Priorizar questões”

– **Resultado:**

Utilizando-se do teste dos 100 dólares, a priorização foi a seguinte:

* Q1: 30 dólares, ou 30% do esforço;

* Q2: 70 dólares, ou 70% do esforço.

– **Observações:**

-

● Atividade 6 “Prototipar e consolidar solução”

– **Resultado:**

O protótipo foi desenhado em uma folha de papel e pode ser visto na Figura 35.

– **Observações:**

O protótipo em papel tem se mostrado muito bem sucedido, pois, de maneira rápida e simples, é possível obter uma boa noção de como o aplicativo deve funcionar.

● Atividade 7 “Planejar desenvolvimento”

– **Resultado:**

As funcionalidades e critérios de aceitação definidos são:

F1: Obter as imagens:

F1C1: Permitir capturar imagens com a câmera;

F1C2: Permitir carregar imagens do dispositivo.

F2: Reconhecer as imagens:

F2C1: Enviar a imagem escolhida para análise com o modelo de ML;

F2C2: Obter uma *string* com os possíveis resultados.

F3: Exibir os resultados:

F3C1: Exibir os resultados para o usuário.

– **Observações:**

-

- Atividade 8 “Realizar reunião diária”
 - **Resultado:**

Não foi realizada por ser apenas um desenvolvedor.
 - **Observações:**

-

- Atividade 9 “Desenvolver solução”
 - **Resultado:**

Seguindo o protótipo de papel, foram inseridos todos os itens pertencentes à interface do aplicativo. Logo em seguida, estes foram conectados ao código³.
Para que o reconhecimento de objetos fosse possível, a maneira escolhida para realizar isso envolveu a identificação de imagens pelo dispositivo, envio delas para comparação com um modelo treinado de ML, por meio do *framework* CoreML, da própria Apple e a posterior exibição dos resultados devolvidos.
 - **Observações:**

Os passos a serem desenvolvidos foram bem lineares, de modo que todas as atividades mencionadas acima foram realizadas de forma sequencial.

- Atividade 10 “Avaliar solução”
 - **Resultado:**

Analisando de forma qualitativa, é possível verificar que a solução desenvolvida cumpre o estabelecido nas atividades anteriores e, ao analisar sua aderência com o protótipo criado, a solução foi aprovada.
O aplicativo desenvolvido pode ser visto ao lado de seu protótipo na Figura 35.
 - **Observações:**

-

- Atividade 11 “Documentar solução”
 - **Resultado:**

Por conter os resultados, junto a algumas observações, o conteúdo de cada seção deste apêndice é considerado a documentação de seu respectivo aplicativo desenvolvido.
 - **Observações:**

-

³ O código-fonte deste aplicativo encontra-se disponível no seguinte repositório: <https://github.com/cambraia/AppsTCC>.

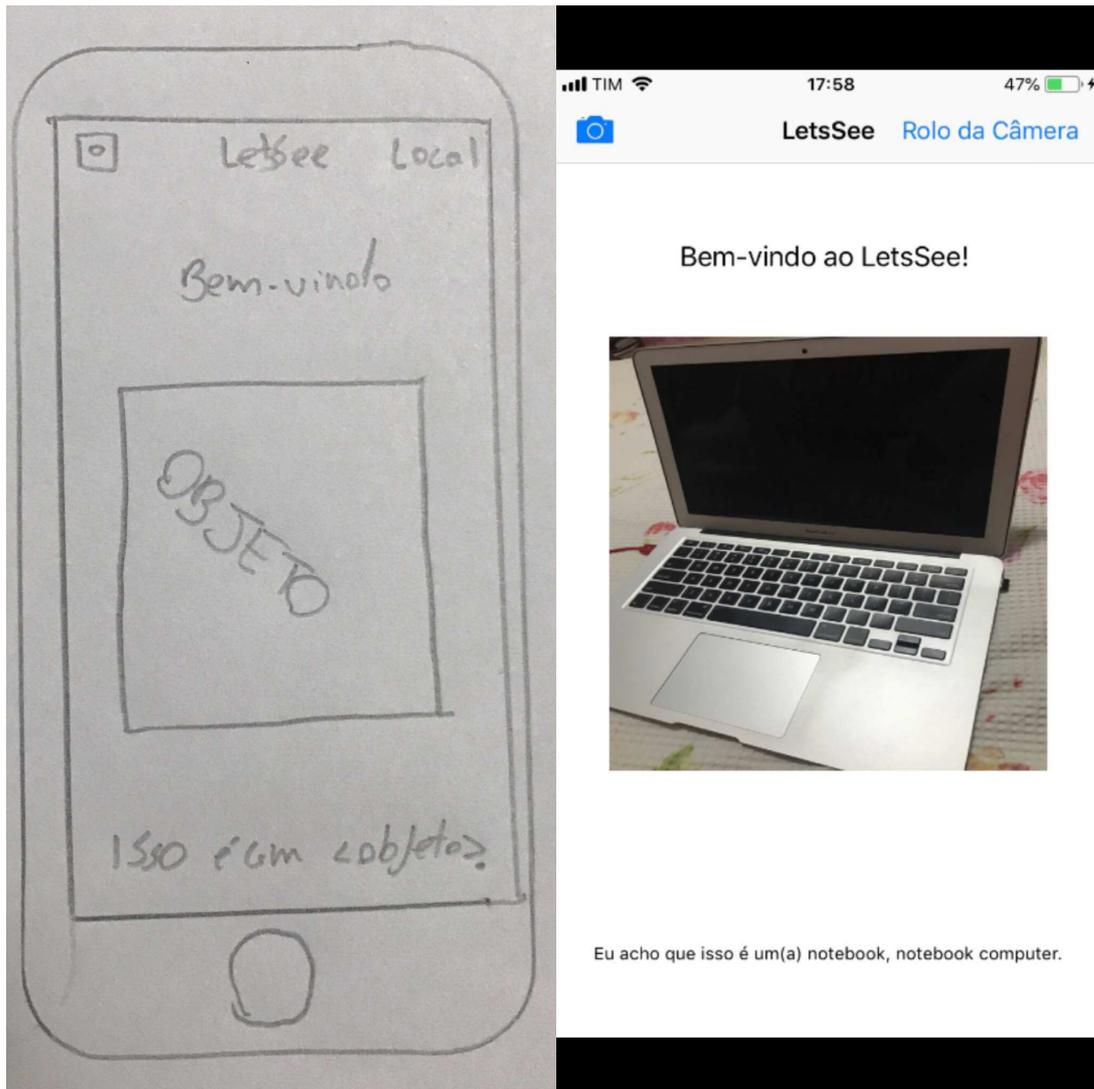


Figura 35 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Autor

C.4 Aplicativo 4 - MeuQR

- Atividade 1 “Ter uma ideia”
 - **Resultado:**
Criar um leitor de códigos QR.
 - **Observações:**
Decidi utilizar a pesquisa para compreender o nível de dificuldade da implementação de diferentes tipos de funcionalidades.
- Atividade 2 “Transformar ideia em desafios”
 - **Resultado:**
Capturas informações da câmera;
Ler e interpretar essas informações;

Exibir resultados para o usuário.

– **Observações:**

-

● Atividade 3 “Transformar desafios em questões”

– **Resultado:**

Q1: Qual câmera do dispositivo utilizar?

Q2: Onde exibir o resultado?

– **Observações:**

Para mim, as perguntas que devemos nos fazer referem-se a finalidade do aplicativo e definição do escopo. Porém, algumas perguntas técnicas podem surgir.

● Atividade 4 “Responder questões”

– **Resultado:**

Q1: Apenas a câmera traseira;

Q2: Por meio de uma caixa de texto no canto inferior da tela.

– **Observações:**

Ao chegar nessa atividade, mostra-se o quão vital é realizar boas perguntas para se ter respostas úteis.

● Atividade 5 “Priorizar questões”

– **Resultado:**

Q1: 55 dólares, ou 55% do esforço;

Q2: 45 dólares, ou 45% do esforço.

– **Observações:**

As porcentagens empregadas descrevem a relevância prática (para o usuário) e não a dificuldade técnica (para o desenvolvedor).

● Atividade 6 “Prototipar e consolidar solução”

– **Resultado:**

O protótipo foi desenhado em uma folha de papel e pode ser visto nas Figuras [36](#) e [37](#).

– **Observações:**

-

- Atividade 7 “Planejar desenvolvimento”
 - **Resultado:**

As funcionalidades definidas são:
F1: Ler a imagem da câmera;
F2: Reconhecer o que significa o código.
 - **Observações:**

-

- Atividade 8 “Realizar reunião diária”
 - **Resultado:**

Não foi realizada por ser apenas um desenvolvedor.
 - **Observações:**

-

- Atividade 9 “Desenvolver solução”
 - **Resultado:**

O aplicativo foi desenvolvido, estava compilando e funcionando⁴.
 - **Observações:**

-

- Atividade 10 “Avaliar solução”
 - **Resultado:**

Analisando de forma qualitativa, é possível verificar que a solução desenvolvida cumpre o estabelecido nas atividades anteriores e, ao analisar sua aderência com o protótipo criado, a solução foi aprovada.
O aplicativo desenvolvido pode ser visto ao lado de seu protótipo nas Figuras 36 e 37.
 - **Observações:**

-

- Atividade 11 “Documentar solução”
 - **Resultado:**

Por conter os resultados, junto a algumas observações, o conteúdo de cada seção deste apêndice é considerado a documentação de seu respectivo aplicativo desenvolvido.

⁴ O código-fonte deste aplicativo encontra-se disponível no seguinte repositório: <https://github.com/cabraia/AppsTCC>.

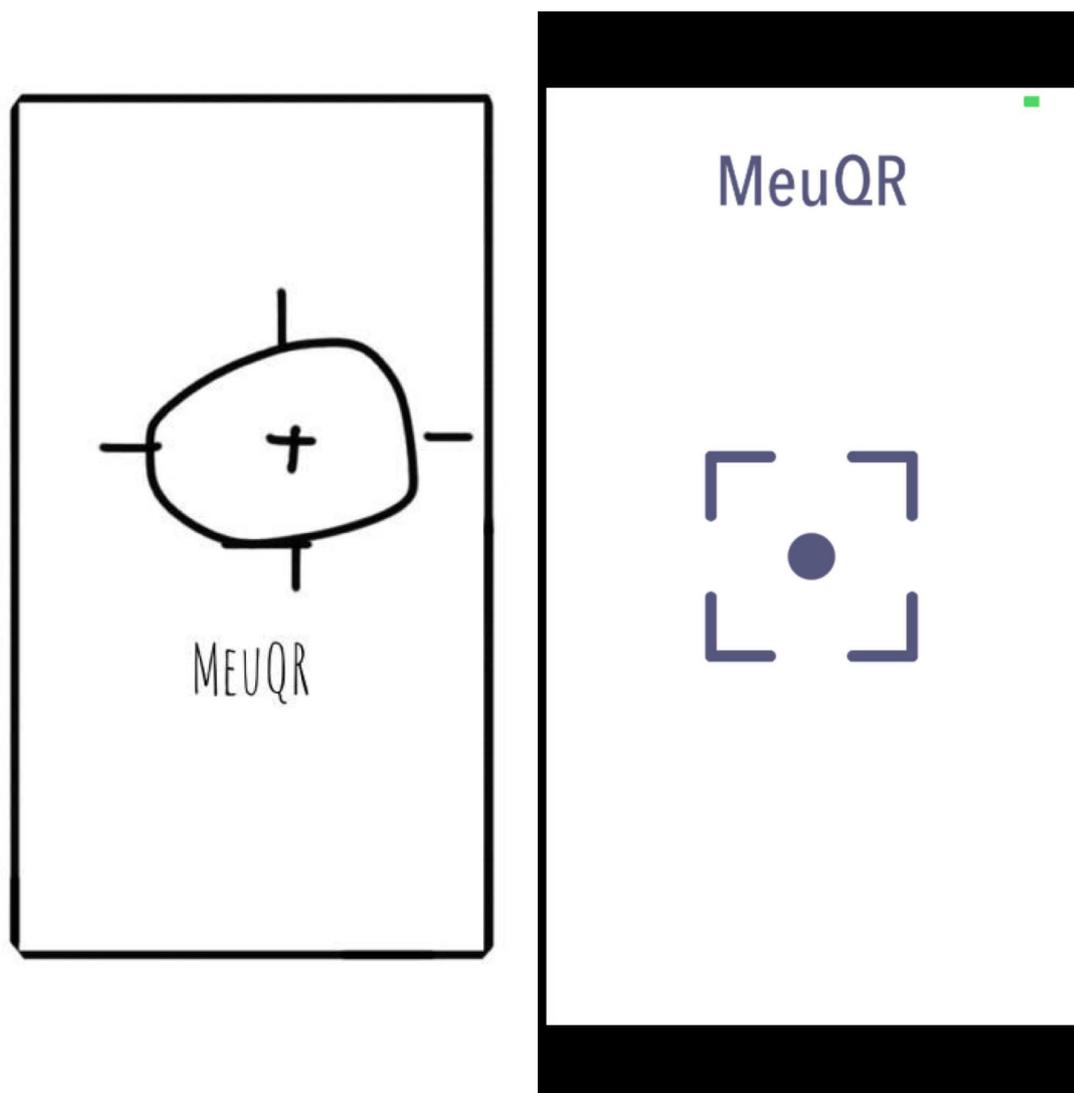


Figura 36 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Desenvolvedor #1

– Observações:

-

C.5 Aplicativo 5 - 3D Scale

- Atividade 1 “Ter uma ideia”

– Resultado:

Criar um aplicativo que utiliza o 3D touch da tela do celular.

– Observações:

-

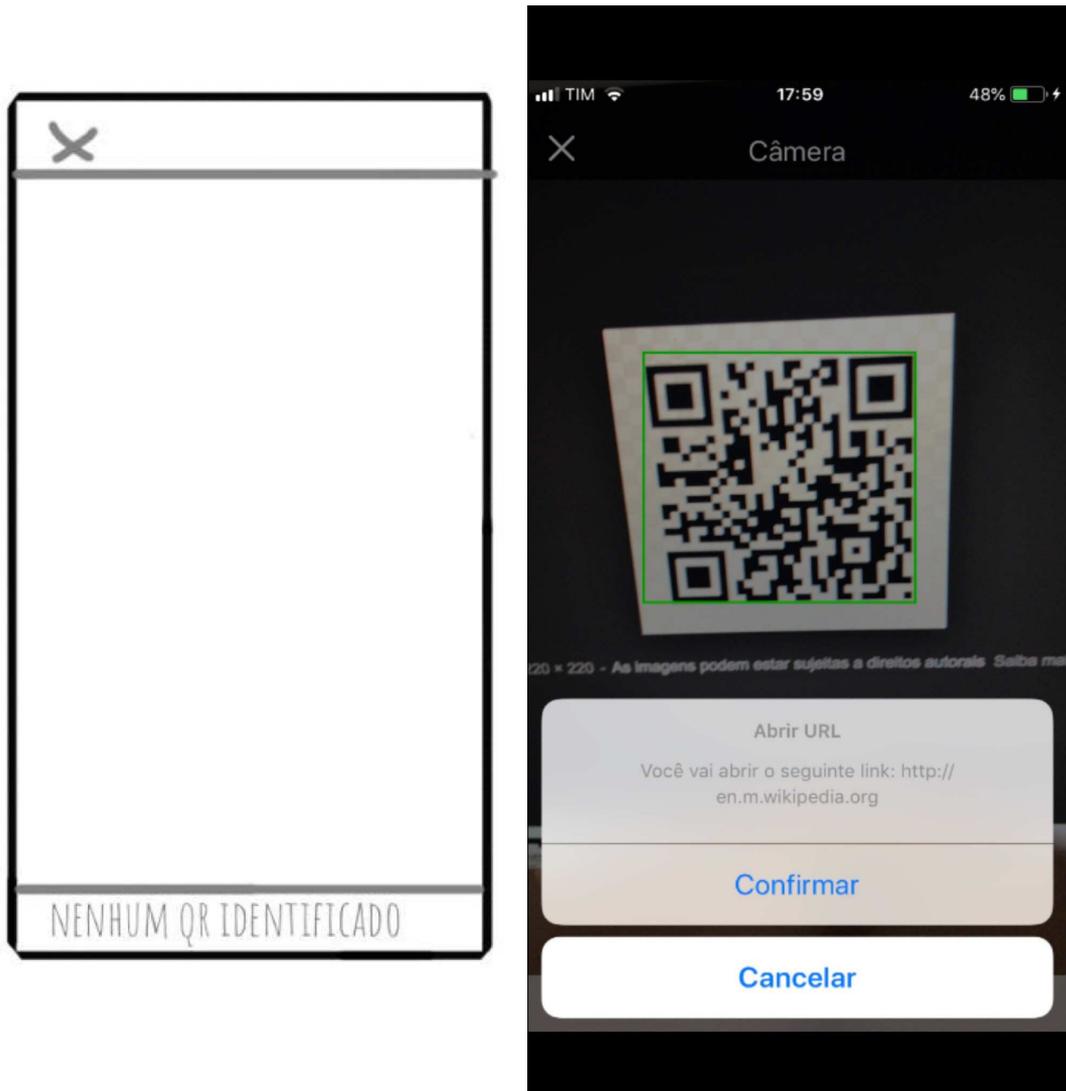


Figura 37 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Desenvolvedor #1

- Atividade 2 “Transformar ideia em desafios”
 - **Resultado:**
Através do 3D Touch do celular transformar a pressão aplicada em peso.
 - **Observações:**
-
- Atividade 3 “Transformar desafios em questões”
 - **Resultado:**
Q1: Mostrar o resultado sempre em gramas?
Q2: Devo transformar a pressão aplicada em peso?
 - **Observações:**
-

- Atividade 4 “Responder questões”
 - **Resultado:**
 - Q1: Sim, dada a precisão necessária;
 - Q2: Sim, utilizando uma fórmula.
 - **Observações:**
 -

- Atividade 5 “Priorizar questões”
 - **Resultado:**
 - Q1: 30 dólares, ou 30% do esforço;
 - Q2: 70 dólares, ou 70% do esforço.
 - **Observações:**
 -

- Atividade 6 “Prototipar e consolidar solução”
 - **Resultado:**
 - O protótipo foi desenhado em uma folha de papel e pode ser visto na Figura [38](#).
 - **Observações:**
 -

- Atividade 7 “Planejar desenvolvimento”
 - **Resultado:**
 - Montar interface;
 - Ligar o *asset* de texto que mostra as gramas a *controller*;
 - Captar o evento de *touch* na tela;
 - Quantificar pressão e transformar em peso;
 - Mostrar resultado na tela atualizando o *asset* de texto.
 - **Observações:**
 -

- Atividade 8 “Realizar reunião diária”
 - **Resultado:**
 - Não foi realizada por ser apenas um desenvolvedor.
 - **Observações:**
 -

- Atividade 9 “Desenvolver solução”
 - **Resultado:**

Foi desenvolvido utilizando como IDE o XCode e como linguagem o Swift⁵.
 - **Observações:**

-

- Atividade 10 “Avaliar solução”
 - **Resultado:**

Analisando de forma qualitativa, é possível verificar que a solução desenvolvida cumpre o estabelecido nas atividades anteriores e, ao analisar sua aderência com o protótipo criado, a solução foi aprovada.

O aplicativo desenvolvido pode ser visto ao lado de seu protótipo na Figura 38.
 - **Observações:**

O que foi desenvolvido era o esperado. Além disso, foi adicionado um texto na tela de começo para obter uma melhor experiência de usuário.

- Atividade 11 “Documentar solução”
 - **Resultado:**

Por conter os resultados, junto a algumas observações, o conteúdo de cada seção deste apêndice é considerado a documentação de seu respectivo aplicativo desenvolvido.
 - **Observações:**

A anotação dos resultados e algumas observações se qualifica como documentação.

⁵ O código-fonte deste aplicativo encontra-se disponível no seguinte repositório: <https://github.com/cambraila/AppsTCC>.

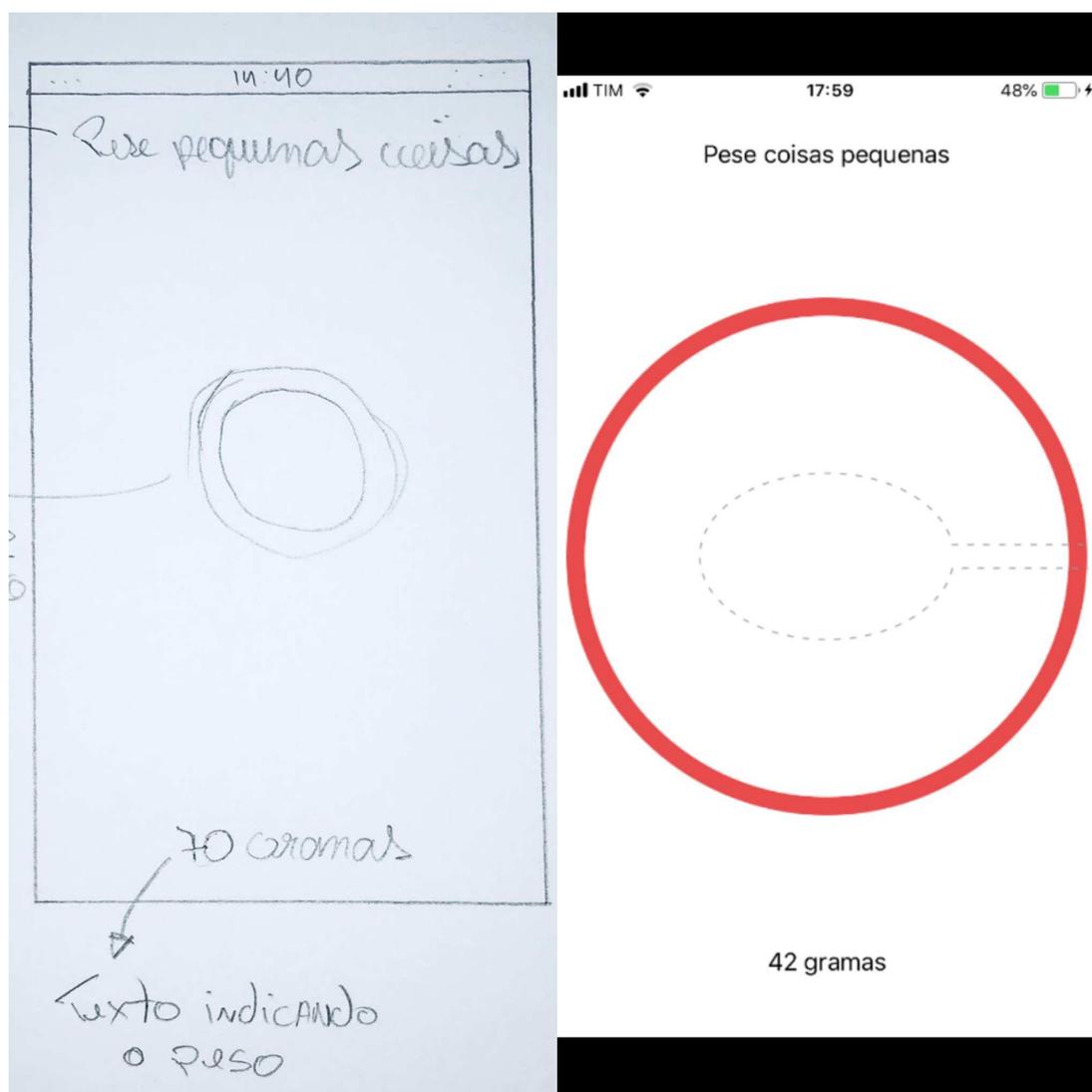


Figura 38 – Comparação entre o protótipo e o aplicativo desenvolvido. Fonte: Desenvolvedor #2