



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I
CENTRO DE CIÊNCIA E TECNOLOGIA
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO
CURSO DE CIÊNCIAS DA COMPUTAÇÃO**

BIANCA LAYSE SOARES FURTADO

**RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES: ESTUDO DE
CASO NO OCARIOT DASHBOARD**

**CAMPINA GRANDE - PB
2019**

BIANCA LAYSE SOARES FURTADO

**RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES: ESTUDO DE
CASO NO OCARIOT DASHBOARD**

Trabalho de Conclusão de Curso apresentado a Universidade Estadual da Paraíba, como requisito parcial à obtenção do título em bacharel em Computação.

Área de concentração: Engenharia de Software.

Orientadora: Profa. Dra. Sabrina de Figueirêdo Souto

**CAMPINA GRANDE - PB
2019**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

F992r Furtado, Bianca Layse Soares.
Rastreabilidade de requisitos guiada por testes
[manuscrito] : Estudo de caso no OCARIoT Dashboard /
Bianca Layse Soares Furtado. - 2019.
105 p. : il. colorido.
Digitado.
Trabalho de Conclusão de Curso (Graduação em
Computação) - Universidade Estadual da Paraíba, Centro de
Ciências e Tecnologia , 2019.
"Orientação : Profa. Dra. Sabrina de Figueirêdo Souto ,
Coordenação do Curso de Computação - CCT."
1. Engenharia de Software. 2. Engenharia de Requisitos.
3. Rastreabilidade de Requisitos. I. Título
21. ed. CDD 005.1

BIANCA LAYSE SOARES FURTADO

**RASTREABILIDADE DE REQUISITOS GUIADA POR
TESTES: ESTUDO DE CASO NO OCARIOT DASHBOARD**

Trabalho de Conclusão de Curso de Graduação em Ciência da Computação da Universidade Estadual da Paraíba, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 21 de Agosto de 2019.



Profa. Dra. Sabrina de Figueirêdo Souto (UEPB)
Orientador(a)



Prof. Dr. Paulo Eduardo e Silva Barbosa (UEPB)
Examinador(a)



Profª. Dra. Kézia de Vasconcelos Oliveira Dantas (UEPB)
Examinador(a)

Ao meu pai, Meinardo Furtado (*in Memoriam*)
pelo exemplo de garra, amor e caráter,
DEDICO.

AGRADECIMENTOS

Ao final dessa longa jornada, somente há gratidão.

Grata ao Pai, que me sustentou até aqui. Durante os dias difíceis Ele fortalece a minha fé para perseverar, não tenho como encerrar esta fase sem honrar Aquele que fez da minha fraqueza, a força que eu precisava para continuar.

Grata ao Filho Jesus, por seu amor redentivo, que com sua vida me ensina que neste mundo existirão aflições, mas me incentiva a ter bom ânimo e assim como Ele, prova que as vencerei neste mundo também!

Grata ao Espírito Santo, que é um amigo fiel, consolador em todo tempo. E assim agradeço e louvo acima de tudo pela Trindade, base para minha fé, na qual me assegura de confiança, força, motivação.

Grata em especial e com demasiada importância a minha família, esta que permanece presente em todos os momentos, assiste de perto todas as lutas, esforços e é alicerce para os vãos mais altos. São os que primeiro acreditam em mim, incentivam e vibram, sinto o amor, cuidado, admiração. Cito o nome de Andreza Furtado, aquela na qual me inspiro. Mãe, obrigada por estar aqui! A senhora é um porto seguro, e sua garra e valores, me ensinam todos os dias o sentido da vida e das conquistas. As minhas irmãs, Brenda e Anna Beatriz, vocês são amor, companhia, sempre me encorajam com o olhar, com uma palavra amiga, com um abraço sincero, com vocês posso contar em todo tempo. Que privilégio tê-las comigo a vocês dedico esta conquista. Vocês são combustíveis de motivação para mim, sou feliz em compartilhar desta vida e o que nela adquiri com vocês.

Aos amigos que muitas vezes se tornam irmãos, que decidem viver parte dos seus momentos, seja vibrando, chorando junto, compartilhando das experiências, ou apenas sendo um bom ouvinte. Vocês também fazem parte de mim! Em especial, agradeço a Priscilla e a Flávia, vocês sabem o quanto foram importantes para que eu chegasse até aqui, desprenderam seu tempo para me auxiliar, para estar junto, para encorajar, só há gratidão em meu coração, que vocês sejam retribuídas em tudo no qual me abençoaram.

Grata aos amigos dessa jornada acadêmica, aos companheiros de curso, aqueles com os quais dividi todos os momentos, os desafios, as lutas diárias e as vitórias também. Aprendi a admirá-los, aprendi com seus valores, aprendi com a convivência. Me despeço dessa jornada, embora sozinha, mas torcendo por vocês que são minha turma. Verei o tempo dessa conquista na vida de vocês e estarei na platéia vibrando por cada um. Em especial, agradeço a

Davi no qual foi minha dupla em todas as atividades seculares, dividimos muito conhecimento, experiência, partilhamos conversas, sonhos. E hoje comemoro este grande sonho, sendo grata a vocês que me impulsionaram até aqui.

A todos os colegas de trabalho, em especial aqueles que se tornaram amigos, e grandes incentivadores. Foi nesta jornada de experiências, no início imaturas, que me lancei bem antes de ter o diploma, mas que me trouxeram até aqui. Uma profissional, dividindo momentos de grandes aprendizagens, desafios, e que bom que tive pessoas como vocês que são referências, para compartilhar da prática. Vocês são admiráveis, obrigada por todo carinho que depositam.

Aos mestres da instituição, vocês são os protagonistas da concretização deste sonho. Compartilhando as experiências, ensino, e todo conhecimento que me fizeram hoje estar no fim. Aos primeiros, que acreditaram em mim e me lançaram os desafios da pesquisa e inovação, ao professor Daniel Scherer o meu muito obrigada, o senhor me mostrou o caminho para as primeiras conquistas acadêmicas. E a todos aqueles que são dignos de reverência nesta jornada, meu muito obrigada.

Agradeço em especial a minha orientadora, Sabrina Souto, por chegar até o fim. Pelo carinho, pela paciência, por não ter desistido. Esse tempo muito me ensinou a suportar, a desejar vencer, e a senhora fez parte dessa etapa final, sendo mediadora para realização do sonho mais esperado de qualquer jovem universitário. Grata, por tudo!

Dedico a todos vocês esta grande conquista. Em especial ao meu querido, amado, e exemplo de pai (*in Memoriam*). Aquele no qual tenho demasiadas e inenarráveis saudades, que jamais serão supridas. Que travam meu coração, mas que me faz transbordar força, garra, e me impulsiona para toda conquista. Você foi, é, e sempre será a minha maior e melhor referência de pessoa, homem, profissional. Seu legado foi deixado, e até hoje faz história. Pai essa conquista tem seu nome! Obrigada pela vida!

Grata a vida! Grata aos sonhos que me alimentam de motivação. Finalizo aqui a etapa que mais me treinou na perseverança, foram dias difíceis, mas em todos eles não estive só. E hoje posso celebrar o fim de uma jornada para início de muitas outras que virão!

RESUMO

Atualmente há no mercado de software várias metodologias de desenvolvimento que se adaptam ao modelo do processo do negócio. A metodologia ágil está em constante uso neste meio, por promover interação e colaboração com os *Stakeholders* do sistema e uma rápida resposta a mudanças. Diante deste cenário é comum observar que quando não aplicada com eficácia, pode desencadear deslizamentos principalmente na gerência e controle destas mudanças que ocorrem nos requisitos que definem as funcionalidades e características do software, conhecidas e tratadas no processo de engenharia de requisitos. Estas mudanças quando não planejadas impactam o projeto e o deixa vulnerável ao insucesso, pois há abertura para inconsistências, falhas e ausência de informações nos artefatos que definem e guiam a orientação do desenvolvimento do software. Como proposta de solução aos problemas citados, foi elaborada uma abordagem que permite rastrear requisitos de um escopo do projeto por meio de testes. O caso específico é tratado para rastrear requisitos para com outros requisitos, quando há versionamentos diferentes de documentações. Portanto o objetivo deste trabalho é melhorar a qualidade de software através da produção de testes e da atualização de seus requisitos. A abordagem foi aplicada em um estudo de caso no projeto OCARIoT Dashboard, que se adequa ao contexto da problemática citada. Como resultado a aplicação da abordagem proposta, obteve-se para o OCARIoT Dashboard uma matriz de rastreabilidade de seus requisitos guiados por casos de testes, que promoveu a aplicação de um processo de validação para suas funcionalidades, além da reescrita de seus requisitos como sugestão de controle e gerência para as mudanças decorridas durante seu processo de desenvolvimento. Possibilitando o bom funcionamento do ciclo de vida do software guiado por uma metodologia ágil, rastreável e com validações. Permitindo ainda ser aplicada e utilizada no contexto de qualquer desenvolvimento de software que se assemelhe ao descrito nesta pesquisa, contribuindo positivamente para a evolução do projeto sob a ótica da rastreabilidade de requisitos guiada por testes.

Palavras-Chave: Rastreabilidade de Requisitos. Testes de Software. Engenharia de Requisitos.

ABSTRACT

There are currently several software development methodologies in the market that adapt to the business process model. In this context, agile methodology is in constant for promoting interaction and collaboration with both system StakeHolders, and rapid response to changes. Given this scenario, it is common to observe that when not applied effectively, it can trigger problems, mainly in the management and control of these changes that occur in the requirements that define the functionality and characteristics of software, known and addressed in the requirements engineering process. These unplanned changes impact the project and leave it vulnerable to failure, as there is openness to inconsistencies, failures, and a lack of information in the artifacts that define and guide software development guidance. As a solution to the problems mentioned, we developed an approach that allows the tracking requirements of project scope through testing. The specific case is handled to track requirements against other requirements when there are different versions of the documentation. Therefore the objective of this work is to improve the software quality by producing tests and updating its requirements. The approach was applied in a case study in the OCARIoT Dashboard project, which fits the context of the aforementioned problem. As a result of the application of the proposed approach, we obtained for OCARIoT Dashboard a traceability matrix of its requirements guided by test cases, which promoted the application of a validation process for its functionalities, as well as the rewriting of its requirements as a control suggestion and management for changes that have occurred during its development process. Making possible the proper operation of the software life cycle guided by an agile, traceable and validated methodology. Also allowing to be applied and used in the context of any software development that resembles that described in this research, contributing positively to the evolution of the project from the perspective of requirements traceability guided by tests.

Keywords: Requirements Traceability. Software testing. Requirements Engineering.

LISTA DE ILUSTRAÇÕES

Figura 1: Engenharia de Requisitos.....	19
Figura 2: Processo de Engenharia aliado a gerência de requisitos	20
Figura 3: Direções de rastreabilidade de requisitos.....	21
Figura 4: Exemplo de mapa mental para captura de testes de aceitação.....	24
Figura 5: Visão geral da abordagem proposta para a rastreabilidade de requisitos guiadas por testes	29
Figura 6: Fase da abordagem proposta - Análise de Requisitos.....	31
Figura 7: Exemplo de criação de casos de testes utilizando mapa mental	32
Figura 8: Fase da abordagem proposta - Geração da Matriz de rastreabilidade.....	33
Figura 9: Fase da abordagem proposta - Aplicação de testes para validação a partir do mapa mental	34
Figura 10: Casos de testes utilizando mapa mental.....	35
Figura 11: Menu de especificação de casos testes na ferramenta TestLink	36
Figura 12: Exemplo de especificação de um caso de teste na ferramenta TestLink	37
Figura 13: Exemplo de visualização de casos de testes disponíveis para execução na ferramenta TestLink	38
Figura 14: Visualização de um caso de teste a ser executado na ferramenta TestLink.....	38
Figura 15: Fase da abordagem proposta - Reescrita de requisitos	39
Figura 16: Como funciona o projeto OCARIoT.....	41
Figura 17: Mapa mental dividido por tela (funcionalidade) para criação de casos de testes ...	44
Figura 18: Tela de Login da aplicação OCARIoT- Dashboard.....	45
Figura 19: Tela Home da aplicação OCARIoT- Dashboard	45
Figura 20: Definição de casos de testes por requisito para a funcionalidade de Login da aplicação	47
Figura 21: Definição de casos de testes por requisito para funcionalidade da Home Screen da aplicação - Parte 1.....	48
Figura 22: Definição de casos de testes por requisito para a funcionalidade da Home Screen da aplicação - Parte 2.....	49
Figura 23: Criação dos testes para a funcionalidade de Login do OCARIoT-Dashboard	51
Figura 24: Criação dos testes para a funcionalidade de Home Screen do OCARIoT-Dashboard na ferramenta TestLink	52

Figura 25: Exemplo 1 - cenário de caso de teste construído para a funcionalidade de Login do dashboard.....	53
Figura 26: Exemplo 2 - cenário de caso de teste construído para a funcionalidade de Login .	54
Figura 27: Exemplo 1 - cenário de caso de teste construído para a funcionalidade de Home screen do dashboard.....	55
Figura 28: Exemplo 2 - cenário de caso de teste construído para a funcionalidade de Home screen do dashboard.....	56
Figura 29: Matriz de resultado de execução de testes	57
Figura 30: Matriz de resultado de execução de testes para um plano do login em específico os requisitos de Screen Resolutions	57
Figura 31: Matriz de resultado de execução de testes para um plano da Home Screen	58
Figura 32: Métricas gerais para execução de testes do plano de Login	59
Figura 33: Métricas gerais para execução de testes do plano de Login específico para screen resolutions.....	59
Figura 34: Métricas Gerais do plano de teste específico para home screen	60
Figura 35: Relatório Geral de métrica de execução para os casos de testes que contemplam LOGIN e HOME	60
Figura 36: Comparação entre protótipo e tela funcional implementada no OCARIoT - referente ao R1.....	63
Figura 37: Processo de compilação de requisitos para a versão de um requisito R1 final.....	64
Figura 38: Comparação entre protótipo e tela funcional implementada no OCARIoT - referente ao R2.....	65
Figura 39: Processo de compilação de requisitos para a versão de um requisito R2 final.....	66

LISTA DE TABELAS

Tabela 1: Modelo de matriz de rastreabilidade entre requisitos guiada por casos de teste	33
Tabela 2: Matriz de rastreabilidade de requisitos guiada por testes para o OCARIoT	50
Tabela 3: Modelo de template gerado para estruturação da reescrita de requisitos	61

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Problema	14
1.2	Solução	15
1.3	Objetivos	16
1.4	Estrutura da monografia	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Processos Ágeis	17
2.2	Engenharia de Requisitos	18
2.2.1	<i>Processo de Gerência de Requisitos</i>	18
2.2.2	<i>Rastreabilidade de Requisito</i>	20
2.2.3	<i>Documento de Requisitos</i>	22
2.3	Teste de Software	23
2.3.1	<i>Casos de Teste</i>	23
2.3.2	<i>Uso de Mapa Mental</i>	24
2.3.3	<i>Técnicas de Teste (Black-box)</i>	25
2.3.4	<i>Testes Manuais</i>	25
2.3.5	<i>TestLink</i>	26
2.4	Considerações Finais	26
3	RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES	28
3.1	Abordagem	28
3.1.1	<i>Análise de Requisitos</i>	30
3.1.2	<i>Definição de Casos de Teste através de Mapa Mental</i>	31
3.1.3	<i>Matriz de Rastreabilidade</i>	32
3.1.4	<i>Testes</i>	34
3.1.5	<i>Reescrita de Requisitos</i>	39
3.2	Considerações Finais	40
4	ESTUDO DE CASO: RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES NO OCARIOT DASHBOARD	41
4.1	OCARIoT Dashboard	41
4.2	Rastreabilidade de Requisitos Guiada por Testes no OCARIoT Dashboard ..	42
4.2.1	<i>Análise de Requisitos</i>	43
4.2.2	<i>Definição de Casos de Teste através de Mapa Mental</i>	44
4.2.3	<i>Matriz de Rastreabilidade</i>	49
4.2.4	<i>Testes</i>	51
4.2.4.1	<i>Especificação</i>	52
4.2.4.2	<i>Execução</i>	56
4.2.4.3	<i>Resultados</i>	58
4.2.5	<i>Reescrita de Requisitos</i>	61
4.3	Considerações finais	66
5	CONCLUSÃO	67
	REFERÊNCIAS	69
	APÊNDICE A – REQUISITOS ANTIGOS DO DASHBOARD	73
	APÊNDICE B – REQUISITOS NOVOS DO DASHBOARD	77
	APÊNDICE C – CASOS DE TESTES DESENHADOS NO MAPA MENTAL PARA AS FUNCIONALIDADES DO OCARIOT	79
	APÊNDICE D – PROCESSO DE REESCRITA DE REQUISITOS PARA AS FUNCIONALIDADES DO OCARIOT	89

APÊNDICE E – COMPILAÇÃO DA REESCRITA DE REQUISITOS102

1 INTRODUÇÃO

Processo de desenvolvimento de software é um conjunto de atividades e resultados associados que vão auxiliar na geração de um produto de valor, ou seja, o resultado do processo é um produto que irá refletir a forma como seu processo foi conduzido.

Embora na atualidade existam diversas formas de processos na geração de software as atividades que os gerenciam são basicamente as mesmas, com foco em: especificação de software na definição de suas funcionalidades e restrições, projeto e implementação de software onde o produto está em construção de acordo com sua especificação, validação de software que garante que as funcionalidades estão de acordo com o esperado e planejado e a evolução do software que permite que o mesmo seja atualizado para continuar a ser útil ao seu cliente (SOARES, 2004).

O que observamos nisto é que muitas organizações criam seu próprio processo, ou adaptam as metodologias existentes ao seu negócio. Dentre elas, existe a metodologia ágil de desenvolvimento, que vem ganhando espaço neste meio por promover dinamização e flexibilidade.

Esse estilo de desenvolvimento ágil aborda diretamente os problemas da mudança rápida, como se adaptar a elas de forma eficiente e trabalha para que haja mais interações entre as pessoas envolvidas, mais aproximação. Substituindo documentações excessivas por diálogo entre as partes, extraindo do cliente suas necessidades, visando melhorar a qualidade da comunicação entre a equipe para que seja facilitadora da informação gerando um produto de valor que atenda as expectativas do cliente (COCKBURN; HIGHSMITH, 2001).

É sabido então, que durante o desenvolvimento do software mudanças podem vir a ocorrer em sua especificação de requisitos, devido a diversos fatores incluindo adaptações às necessidades do cliente. No desenvolvimento ágil, a preocupação é gerenciar e promover estas mudanças rapidamente, de forma que não venha impactar no produto final. Quando não bem gerenciadas, problemas podem ser acarretados resultando no possível fracasso do produto.

É diante deste cenário que identifica-se o importante papel da engenharia de requisitos (ER), ela é responsável por utilizar técnicas sistemáticas e repetíveis para garantir que os requisitos do software estejam sempre completos, consistentes e relevantes. Seu principal artefato é o documento de requisitos, que é uma declaração formal daquilo que deve ser

desenvolvido. Então, o processo aplicado da ER é um conjunto de atividades que devem criar, validar e manter um documento de requisitos.

Especialmente no desenvolvimento ágil, voltado para responder a mudanças no software, deve ser bem aplicado a atividade de gerência de requisitos que promove a configuração e o controle do versionamento do documento, além de verificar inconsistências nas especificações após a evolução e alteração no que está sendo produzido.

Percebe-se que mudanças de requisitos são inevitáveis, logo é uma boa prática da ER manter o controle através da gerência dos mesmos. Esta gerência permitirá o rastreamento dos requisitos, sendo possível determinar quem o definiu, porque ele existe, se possui relacionamento com outros requisitos ou outros artefatos do projeto, e assim quando mudanças forem solicitadas o seu impacto na documentação será reduzido, porque facilmente poderá ser rastreado para se fazer as devidas atualizações (ESPINDOLA; MAJDENBAUM; AUDY, 2014).

A rastreabilidade de requisitos é favorável então para manter a validação dos requisitos que estão sendo implementados e acompanhá-los durante o processo de desenvolvimento, e até seu ciclo de vida útil. Está estritamente ligada a qualidade do software, que só é possível manter quando se tem um escopo de requisitos bem definidos e gerenciados. Assim, testes podem ser gerados para validar os critérios de aceitação para o projeto e os seus requisitos, mantendo a qualidade do produto e aumentando a confiança no mesmo.

1.1 Problema

Apesar de saber a importância de um processo de desenvolvimento de software bem gerenciado, e de que mudanças nos requisitos do software são imprescindíveis. A insegurança está quando não há esta gerência, e em meio a mudanças para atender as demandas do cliente o seu acompanhamento de gestão for sendo negligenciado.

É comum organizações falharem na gestão dos seus processos e retomá-los durante um ciclo de desenvolvimento é extremamente mais difícil, oneroso, e requer tempo. Porém, um ciclo de desenvolvimento ágil deve ser bem planejado, para que as entregas de valor não sejam atrasadas a cada iteração. Se tudo isso vier a falhar, o software provavelmente virá ao fracasso como consequência.

Desta forma se torna impraticável manter a qualidade de um software que falha na sua gerência, resultando geralmente em documentos contratuais obsoletos, elicitação de requisitos

sem a declaração formal e devidas atualizações, um verdadeiro aglomerado de problemas com consequências irreversíveis para o projeto.

O problema aumenta em projetos de desenvolvimento ágil e distribuído, pois há facilidade de mais de uma versão de requisitos está sendo utilizada pelas partes envolvidas, diante das mudanças ocorridas. Aplicação de testes se torna incabível, pelo fato de não saber o que testar, pois os requisitos não mostram a veracidade do desenvolvimento, por não ter acompanhado as atualizações, e com isso não retomam a segurança dos critérios de aceitação necessários para validação.

Pelas razões supracitadas, é fato que o processo de ER, gerência e demais práticas gestões que possam vir a ser adotadas, incluindo a rastreabilidade de requisitos para projetos de desenvolvimento ágil voltados para responder à mudanças rápidas, precisam ser de alguma forma aplicados. Só assim as demais atividades do ciclo de vida de um software podem vir a ter sucesso.

Para suprir a análise de impacto das mudanças que podem ocorrer no decorrer do projeto, existe uma solução que corresponde a este contexto, porém aplicando a rastreabilidade dirigida por testes automatizados, seu objetivo é tornar rastreável o requisito ao seu código fonte, para compreender a cobertura de código para com o requisito, ou seja o que está sendo invocado no código fonte que corresponde ao requerido (LUCASSEN et al., 2017). Porque não é suficiente diante do aspecto de rastrear requisitos para com outros requisitos diversões documentadas distintas através de testes manuais com a intenção de gerar um único artefato completo e preciso de documentação contratual para garantir as regras de negócio do produto.

1.2 Solução

Dada a elevada necessidade de se manter rastreável o documento de requisitos e a especificação das funcionalidades do software, e a real dificuldade de manter essa gestão em desenvolvimento ágil e distribuído. Uma solução seria utilizar casos de testes manuais já criados para uma versão da documentação de requisitos como elo de relacionamento entre os requisitos elicitados em todas as versões disponíveis no processo de desenvolvimento, assim a rastreabilidade seria guiada pelos testes, e o conhecimento das regras de negócio em uso e as que são essenciais para o projeto seria descoberto. Só assim um documento de requisitos

poderia ser reformulado como resultado, para atender as necessidades de desenvolvimento das equipes responsáveis, que teriam equivalência nas regras de negócio do produto.

Facilitando a implantação correto de uma abordagem de qualidade, na especificação, execução de testes manuais e report de bugs para correção. E assim, o ciclo de vida do software funcionando da maneira esperada resultará em um produto de sucesso.

1.3 Objetivos

O objetivo geral deste trabalho é promover a qualidade de software através da produção de testes manuais e da atualização de seus requisitos. Para isso, foi proposta uma abordagem que faz uso da escrita de casos de testes para rastrear requisitos de uma documentação mais atual (porém incompletos) para com outros já elicitados em versões distintas, durante o ciclo de desenvolvimento do mesmo.

Com base no objetivo geral, os objetivos específicos são:

- Propor uma abordagem para fazer rastreamento de requisitos através de casos de testes manuais;
- Aplicar a abordagem proposta a um estudo de caso com o OCARIoT Dashboard.

1.4 Estrutura da monografia

A monografia segue a estrutura descrita pelos capítulos citados a seguir:

- Capítulo 2: fornece embasamento teórico para compreensão do problema contextualizado neste trabalho;
- Capítulo 3: apresenta a abordagem para rastreabilidade de requisitos guiada por testes e como ela se aplica;
- Capítulo 4: descreve a aplicação da abordagem de rastreabilidade proposta aplicada ao objeto de estudo OCARIoT-Dashboard;
- Capítulo 5: apresenta as conclusões assumidas após a problematização, seu contexto e a aplicação da abordagem proposta como solução, no objeto de estudo OCARIoT-Dashboard, e suas contribuições para o meio de desenvolvimento de software.

2 FUNDAMENTAÇÃO TEÓRICA

Essa seção tem como objetivo estabelecer uma base teórica para a compreensão de conceitos que são fundamentais para o entendimento do contexto a ser abordado na pesquisa realizada. Inicialmente serão apresentados os conceitos relacionados a Processos ágeis; Engenharia de Requisitos e por fim, Testes de software, na qualidade do processo.

2.1 Processos Ágeis

Até o final do século XX, as metodologias para o desenvolvimento de softwares eram baseadas em métodos rígidos, os quais não se adaptam a mudanças nos requisitos de negócio do software. Não havendo clareza no processo para as partes envolvidas, o cliente muitas vezes se sente inseguro quanto ao produto que será entregue. Nesse contexto, a fim de solucionar estes problemas, surgem as metodologias ágeis de desenvolvimento de software (GONÇALVES, 2017).

De acordo com Pressman e Maxim (2016) um grupo de indivíduos renomados que representavam as metodologias de desenvolvimento mais utilizadas, decidiram discutir e identificar padrões e valores comuns entre elas. A partir dessas discussões foi assinado um acordo de um conjunto de valores e princípios que ficou conhecido como o Manifesto ágil, este contém quatro declarações de valores, que valorizava: Indivíduos e interações acima de processos e ferramentas; software funcionando acima de documentação mais abrangente; colaboração com o cliente acima de negociação de contratos; resposta às mudanças acima de seguimento de um plano.

Nesse contexto, as metodologias ágeis promovem em sua aplicação, interação e colaboração com a equipe de desenvolvimento e *stakeholders* (partes interessadas) facilitando a compreensão do negócio e seus requisitos para o sucesso do projeto. Com isso o processo se torna transparente, já que todos acompanham o desenvolvimento do software, seu andamento, as funcionalidades que estão sendo entregues e os seus custos.

Durante esse processo, a flexibilidade na aceitação de mudanças tem importância acentuada, e por se tratar de um processo iterativo, a cada nova iteração um novo planejamento é orçado, riscos podem ser previstos e evitados de maneira eficaz, falhas são identificadas com antecedência e podem ser corrigidas no intervalo das iterações de

desenvolvimento. Portanto, é possível controlar os custos das funcionalidades ao se planejar e desenvolver com prioridade as que são de mais valia para o cliente, somado a isto a aplicação dos valores e princípios do ágil, acompanhados de testes que garantam a qualidade do software (GONÇALVES, 2017).

2.2 Engenharia de Requisitos

Pressman e Maxim (2016) afirmam que projetar e construir um software é desafiador, pois envolve criatividade e demanda a tomada de decisões importantes. Na prática, ao se projetar um software, os desenvolvedores iniciam imediatamente a sua construção, e por vezes é falho o entendimento do que se deve realizar com clareza, sem a compreensão necessária das regras de negócio do produto do cliente. Alguns argumentam acreditar que só conseguirão compreender os requisitos à medida que forem progredindo com o desenvolvimento, em meio às entregas nas iterações do projeto, no qual com o feedback do cliente as soluções vão se tornando melhores e mais precisas. Embora esse discurso seja atraente por conter informações verídicas, ele pode tornar um projeto vulnerável ao fracasso.

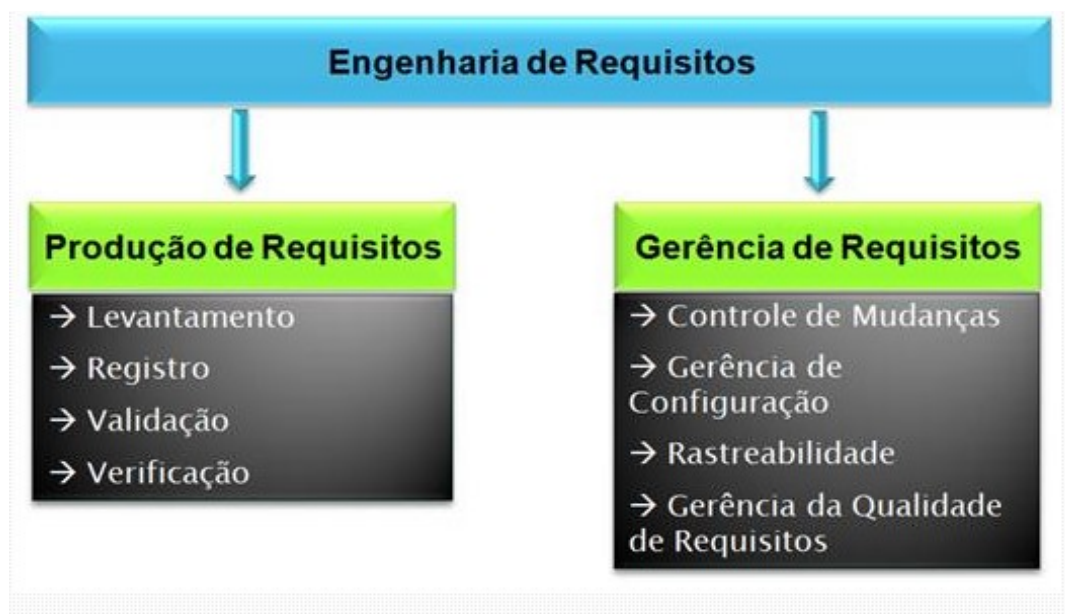
Diante deste cenário é de grande valia a função da Engenharia de Requisitos (ER), ela é ideal no levantamento das necessidades e operacionalidades do software, se tornando o elo entre o projeto e sua construção. Assim, a mesma é denominada como a representação de um conjunto de procedimentos e ferramentas que objetivam identificar as reais necessidades do software, estabelecendo suas características e requisitos para que o mesmo seja aceito por parte do cliente. Entretanto, se as necessidades do cliente forem mal interpretadas, o projeto será mal desenvolvido, culminando em um produto de baixa qualidade, e sendo suscetível a mudanças no decorrer de seu desenvolvimento (MENEZES et al., 2015).

2.2.1 Processo de Gerência de Requisitos

Atividades relacionadas à produção e especificação de requisitos são definidas na engenharia de requisitos, mas o processo de gerência destes também deve ser considerado. Logo, podemos definir a engenharia de requisitos (Figura 1) como um conjunto de atividades relacionadas a produção (levantamento, registro, validação e verificação) e a gerência (controle de mudanças, gerência de configuração, rastreabilidade, e gerência de qualidade dos

requisitos) de requisitos. Ambos conceitos devem ser tratados em conjunto em processos de desenvolvimento de software (DEVMEDIA, 2008).

Figura 1: Engenharia de Requisitos

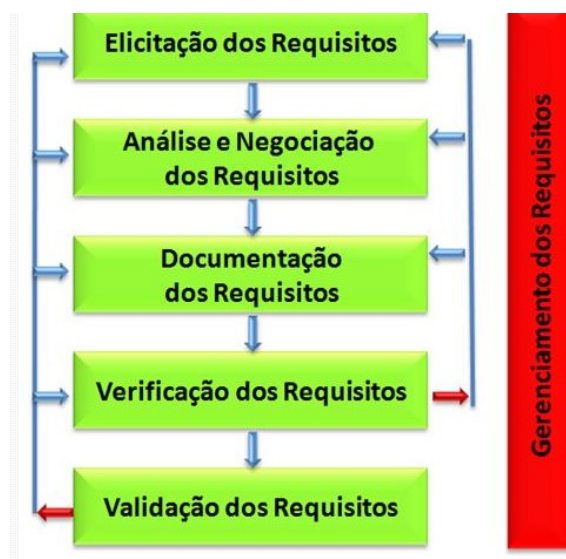


Fonte: <https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-aengenharia-de-requisitos/8034>.

De acordo com Sommerville (2011), alguns dos principais objetivos da gerência de requisitos é registrar e acompanhar os requisitos ao longo de todo o processo de desenvolvimento, documentar e fazer o controle dos requisitos alocados a fim de estabelecer uma *baseline* para uso gerencial e manter os artefatos consistentes com os requisitos. Para dar assistência a estes objetivos, é importante estabelecer um processo de engenharia bem definido, incluindo as atividades descritas na Figura 2.

De acordo com a Figura 2, as atividades: elicitação de requisitos, análise e negociação dos requisitos, documentação dos requisitos, verificação dos requisitos e validação dos requisitos, são descritas no processo de engenharia e estão sendo apresentadas com um comportamento sequencial, na prática pode existir interação e sobreposição entre elas. E deve ocorrer em paralelo a gerência de requisitos atuando nos endereçamentos de modificações dos requisitos. (DEVMEDIA, 2008).

Figura 2: Processo de Engenharia aliado a gerência de requisitos



Fonte: <https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>.

2.2.2 Rastreabilidade de Requisito

A rastreabilidade de Requisitos (RR) está relacionada com a qualidade do produto, e é considerada uma atividade essencial para o gerenciamento de requisitos. De acordo com Sayão e Leite (2005), o rastreamento de requisitos é utilizado para fornecer relacionamentos entre eles, entre a arquitetura e implementação final do sistema, possibilitando uma adequada compreensão sobre dependência entre ambos através dos artefatos do projeto. Ela pode ser implementada por um agrupamento de elos ou ligações (links) entre requisitos inter-relacionados, entre requisitos e suas fontes, e entre requisitos e componentes que os implementam.

A literatura aponta a rastreabilidade de requisitos como um fator importante e favorável no processo de desenvolvimento de sistemas com qualidade. E pode ser considerada como um requisito não funcional no projeto. Os relacionamentos que a rastreabilidade promove facilita os projetistas identificarem a cobertura de requisitos que estão sendo atendidos, e também apoia a detecção dos que não estão sendo, podendo tomar medidas para sanar problemas em tempo hábil (SAYÃO E LEITE, 2005).

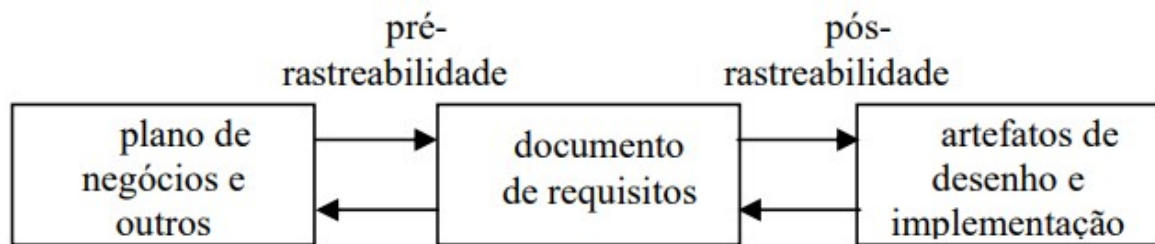
Ramesh e Jarke (2001) afirmam que apesar da importância da RR, nem todas as organizações envolvidas com desenvolvimento de sistemas a aplicam de forma adequada.

Pois, muitas vezes os esforços e valores para tal atividade são dados ao acaso, e os modelos por vezes são mal compreendidos e interpretados.

Alguns dos fatores para este cenário são explicados pela falta de conhecimento sobre as práticas, modelos de rastreabilidade e o receio que estas sejam complexas para aplicação, além da influência do tempo dedicado para isto, que pode ser um caso típico em processos de desenvolvimento, quando não planejado (GENVIGIR, 2009).

De acordo com Sayão e Leite (2005) a rastreabilidade pode ser vista como habilidade de assistir e apresentar a vida de um requisito em ambas as direções, que podem ser explicadas pela figura a seguir:

Figura 3: Direções de rastreabilidade de requisitos



Fonte: Sayão e Leite (2005).

No primeiro relacionamento da Figura 3, temos a rastreabilidade *forward-to* (para frente) que faz a ligação de documentos obtidos no processo de elicitação e da origem do projeto a requisitos relevantes, e a rastreabilidade *backward-from* (para trás) que liga os requisitos as suas fontes de elicitação.

No segundo relacionamento da Figura 3, temos a rastreabilidade *forward-from* que liga requisitos a seus artefatos de desenho e implementação e a rastreabilidade *backward-to* que liga artefatos de volta a requisitos (SAYÃO E LEITE, 2005).

Logo, rastreabilidade também pode ser definida como uma habilidade de descobrir e rastrear a história de toda a característica do sistema, identificando impactos de mudanças nos requisitos ou qualquer outro artefato.

2.2.3 Documento de Requisitos

O documento de requisito é uma descrição oficial de o que os desenvolvedores do sistema devem implementar, que devem conter requisitos do usuário para o sistema e exposição dos requisitos de sistema. Serve como um meio de comunicação entre o projetista do software, a equipe de desenvolvimento e o usuário, a fim de estabelecer oficialmente um contrato de funcionalidades, características, e comportamentos acerca do software a ser desenvolvido, além de ajudar quando precisa adaptar o sistema a novos requisitos (SOMMERVILLE, 2011).

A quantidade de detalhes no documento irá depender do tipo de sistema em desenvolvimento e do processo utilizado. Sistemas mais críticos necessitam de maior número de detalhes, porque a segurança e a proteção devem ser analisadas detalhadamente. Caso os sistemas sejam desenvolvidos por uma corporação separada exigirá especificações detalhadas e precisas, diferente se um processo interno de desenvolvimento iterativo é usado, o documento usado será menos detalhado (SOMMERVILLE, 2011).

Para os requisitos de usuário e de sistema é essencial que sua escrita seja legível facilitando a compreensão dos envolvidos, além de serem completos e consistentes, pois evitará que problemas e erros surjam durante a fase de implementação do software. Os Requisitos definidos irão guiar o processo de desenvolvimento do software (SOMMERVILLE, 2011).

De acordo com Sommerville (2011) o documento de requisitos deve ser composto por sentenças em linguagem natural, seguindo como sugestão determinadas diretrizes:

- 1 Crie um formato padrão e certificar que todas as definições de requisito estejam neste formato, um identificador único torna mais fácil a verificação e diminui risco de omissões.
- 2 Usar uma linguagem densa para diferenciar os requisitos obrigatórios e os desejáveis. Os obrigatórios por vezes são escritos usando-se ‘deve’, os quais os sistemas têm que dar suporte. Os desejáveis são escritos usando-se ‘pode’ e estes não são indispensáveis.
- 3 Usar uma estratégia para destacar as partes essenciais do requisito (negrito, itálico ou cores).
- 4 Evitar o uso de jargões, siglas e acrônimos, por vezes são mal interpretados. Associar uma lógica a cada requisito de usuário, que são proveitosos quando há alteração dos requisitos para decisão sobre quais mudanças são indesejáveis.

2.3 Teste de Software

Testar um software pode ser definido como verificar através de uma execução controlada se o desempenho equivale ao especificado, com intenção de revelar o máximo de falhas com o mínimo de esforço (IEEE, 1990). Um teste é eficaz quando este tem uma elevada probabilidade de revelar um erro ainda não descoberto, são utilizados para garantir a sua qualidade, aumentando assim a confiança do mesmo.

Contudo, apesar de revelarem falhas no software, as atividades de teste não possuem a capacidade de mostrar a ausência delas, só pode certificar se erros existem e estão presentes (DIJKSTRA, 1972). Seria impraticável verificar todas as possibilidades de combinações de testes em um software e validar que o mesmo está em estado perfeito, além do que as atividades de desenvolvimento são executadas por pessoas, e pessoas são passíveis de falhas.

Nesta atividade, o principal objetivo é projetar testes que descubram sistematicamente diferentes classes de erros e que sejam executados em uma quantidade de tempo e esforços mínimos. Se a atividade de teste for conduzida com sucesso, ela descobrirá erros no software, que são imprescindíveis (TOMELIN, 2001).

2.3.1 Casos de Teste

Durante a análise de requisitos é definido o que testar do sistema, identificando as condições de teste, cada condição é definida como um item que pode ser verificado por um ou mais casos de teste. Assim, estabelecer a rastreabilidade destas condições para com as especificações e requisitos, permite identificar a cobertura de testes para determinado requisito, e ainda analisar o impacto quando requisitos mudam (BLANCO, 2012).

Casos de testes também podem ser definidos como cenários de validação das funcionalidades e requisitos do software. Eles geralmente analisam as informações desde o documento de requisito até a sua implementação, desse modo é verificado se a regra de negócio requisitada está sendo implementada no software.

As informações contidas em um caso de teste costuma ser: nome do projeto; número identificador - ID (que deve ser único); módulo, ou funcionalidade, ou requisito que está sendo testado (dependendo do modelo implementado pelos responsáveis); descrição sucinta

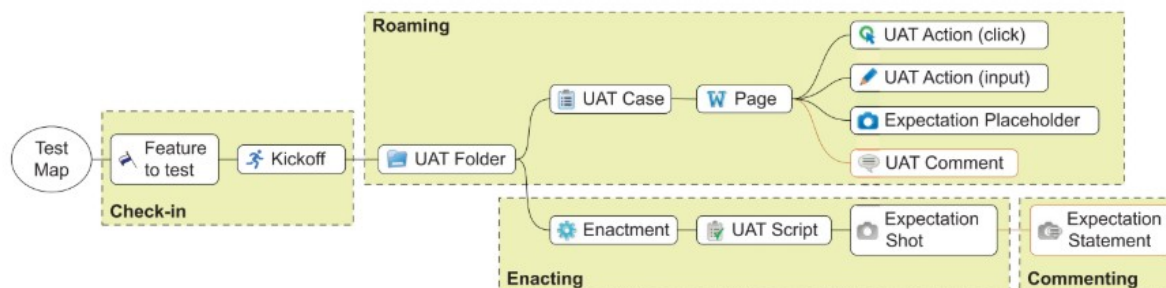
sobre o caso especificado; pré-condições (definem um estado anterior, ou comportamento estável para que o caso possa ser executado); o roteiro do teste (que define os passos a serem seguidos, e o resultado esperado para cada ação; e o resultado final do teste quando este é executado (com data de execução, e nome do testador) (BLANCO, 2012).

2.3.2 Uso de Mapa Mental

No processo de desenvolvimento ágil é comum a colaboração com as partes interessadas. Por ser um modelo iterativo a cada ciclo é feito a entrega de um produto de valor, ou seja, alguma funcionalidade do sistema com estado de ‘pronto’, isto combinado com testes de aceitação que tem a função de validar os critérios de aceitação do software para garantir a qualidade do mesmo, antes dele estar disponível para o cliente. Porém, os mesmos critérios de aceitação podem ser utilizados para que o cliente tenha a oportunidade de compreender os requisitos que estão sendo cobertos, acompanhar o desenvolvimento e fazer a sua própria validação, ao utilizar o sistema.

Assim, de acordo com Otaduy e Diaz (2017), os testes de aceitação deve permitir aos clientes a oportunidade de interagir com o software por conta própria e descobrir se tudo está funcionando como convém. Contudo, deve haver uma orientação para que eles possam compreender como deve ser feito os testes, por isto os mapas mentais são importantes. São conhecidos por seu benefício em ambientes e cenários que necessitam de colaboração, principalmente quando esta colaboração parte de locais geográficos distintos. Com eles é possível descrever cenários, apresentar relacionamentos de dependência e especificar as regras de negócio, como mostra a Figura 4 a seguir:

Figura 4: Exemplo de mapa mental para captura de testes de aceitação



Fonte: Otaduy e Diaz (2017).

A Figura 4 faz a utilização de mapas mentais para capturar as preocupações dos testes de aceitação. Os nós do mapa estão sobrecarregados com a sua significância, uma anotação ou ícone de representação. Cada desenho de nó possui uma dependência no fluxo.

Assim os mapas mentais promovem o acesso às sessões de testes de aceitação e compreensão do fluxo da sua aplicação, com uma estratégia visual e ao mesmo tempo formal, estruturada o suficiente para ser aplicada (OTADUY e DIAZ, 2017).

2.3.3 Técnicas de Teste (Black-box)

Atualmente existem muitas maneiras de se testar um software, porém existem técnicas que guiam os tipos de testes e o seu objetivo é o mesmo: encontrar falhas no software. As técnicas são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste, contemplando diferentes perspectivas do software e impondo a necessidade de se estabelecer uma estratégia de teste que considera as vantagens e os aspectos complementares dessas técnicas (DELAMARO et al., 2007). Um das técnicas existentes é: Black-box (Testes de caixa-preta / funcionais).

- Black-box (Teste Funcional): é uma técnica de teste em que o componente a ser testado é abordado como uma caixa-preta, onde os detalhes de implementação não são considerados e é avaliado conforme o ponto de vista do usuário (DELAMARO; MALDONATO; JINO, 2016). O testador não conhece sobre o código-fonte do mesmo. É necessário apenas fornecer dados de entrada para o teste ser executado, e o resultado obtido é comparado ao resultado esperado e previamente conhecido. O teste terá sucesso quando os resultados comparados são iguais (NETO, 2015).

2.3.4 Testes Manuais

Testes manuais fundamentam-se no desenvolvedor, testador, ou até mesmo no próprio usuário final do software, ao utilizar o sistema a fim de encontrar anomalias no seu funcionamento.

Constata-se que é o tipo de teste menos eficiente, por ser impossível uma pessoa testar exaustivamente um sistema de forma a não ter mais nenhuma possibilidade de falha. Embora

tenha essa fragilidade em relação a cobertura, os testes manuais permanecem sendo utilizados pela facilidade da aplicação, simbolizando o ambiente em produção e simulando cenários que em testes manuais são mais precisos (LEAL, 2009).

Portanto os testes manuais são aqueles em que o testador realiza o passo a passo do fluxo, geralmente guiado por um roteiro que informa o que deve ser realizado, o comportamento do Software e o resultado esperado para suas ações. Eles são considerados úteis quando se trata de subjetividade ou análise de um ser humano nos resultados obtidos, porém não são recomendados quando se necessita executar o mesmo testes várias vezes (NETO, 2019).

2.3.5 *TestLink*

O TestLink é um projeto de código aberto licenciado pela General Public License - GPL (é simplesmente a designação de licença de software para software). É OpenSource, com o código-fonte disponível na Web. É uma ferramenta de gerenciamento de testes e execução de testes, que permite às equipes de garantia de qualidade criar e gerenciar seus casos de teste, bem como organizá-los em planos de teste. Os planos criados permitem aos membros da equipe executar os casos de teste e acompanhar seus resultados dinamicamente (TESTLINK, 2019).

O TestLink ajuda a manter o processo de teste sob controle, ele forma um repositório para requisitos e casos de teste e os relaciona a compilações, plataformas e equipes. É possível através dele alocar testes para a equipe que os executa e registrar os resultados. Uma ampla variedade de relatórios fornece informações sobre o que foi feito e o que ainda precisa ser feito (TESTLINK, 2019).

2.4 Considerações Finais

Neste capítulo foi apresentado o embasamento teórico para o contexto da pesquisa elaborada, envolvendo os conceitos de processos ágeis de desenvolvimento de software, engenharia de requisitos e suas subseções (processo de gerência de requisitos, rastreabilidade de requisitos e documento de requisitos), e por fim o entendimento sobre testes de software e

suas subseções (casos de testes, o uso de mapa mental na diagramação de testes, técnicas de testes, testes manuais e sobre a ferramenta de gerência de testes: Testlink).

3 RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES

Nesta seção serão apresentadas as etapas de elaboração deste trabalho, contendo a explicação da abordagem desenvolvida como solução para a rastreabilidade de requisitos guiada por testes. Faz-se entender que elos de rastreabilidade contribuem para mostrar que todos os requisitos estão sendo atendidos no software em desenvolvimento, e que nenhuma informação está sendo omitida neste processo.

3.1 Abordagem

Como atualmente os processos ágeis de desenvolvimento têm ganhado cada vez mais espaço no mercado de sistemas computacionais, os mesmos apresentam uma metodologia focada nas interações entre as partes interessadas mais do que nos processos, colaboração com o cliente e respostas as mudanças mais do que documentações excessivas. Com isso, a rastreabilidade de requisitos pode ser considerada onerosa, exaustiva, dispendiosa.

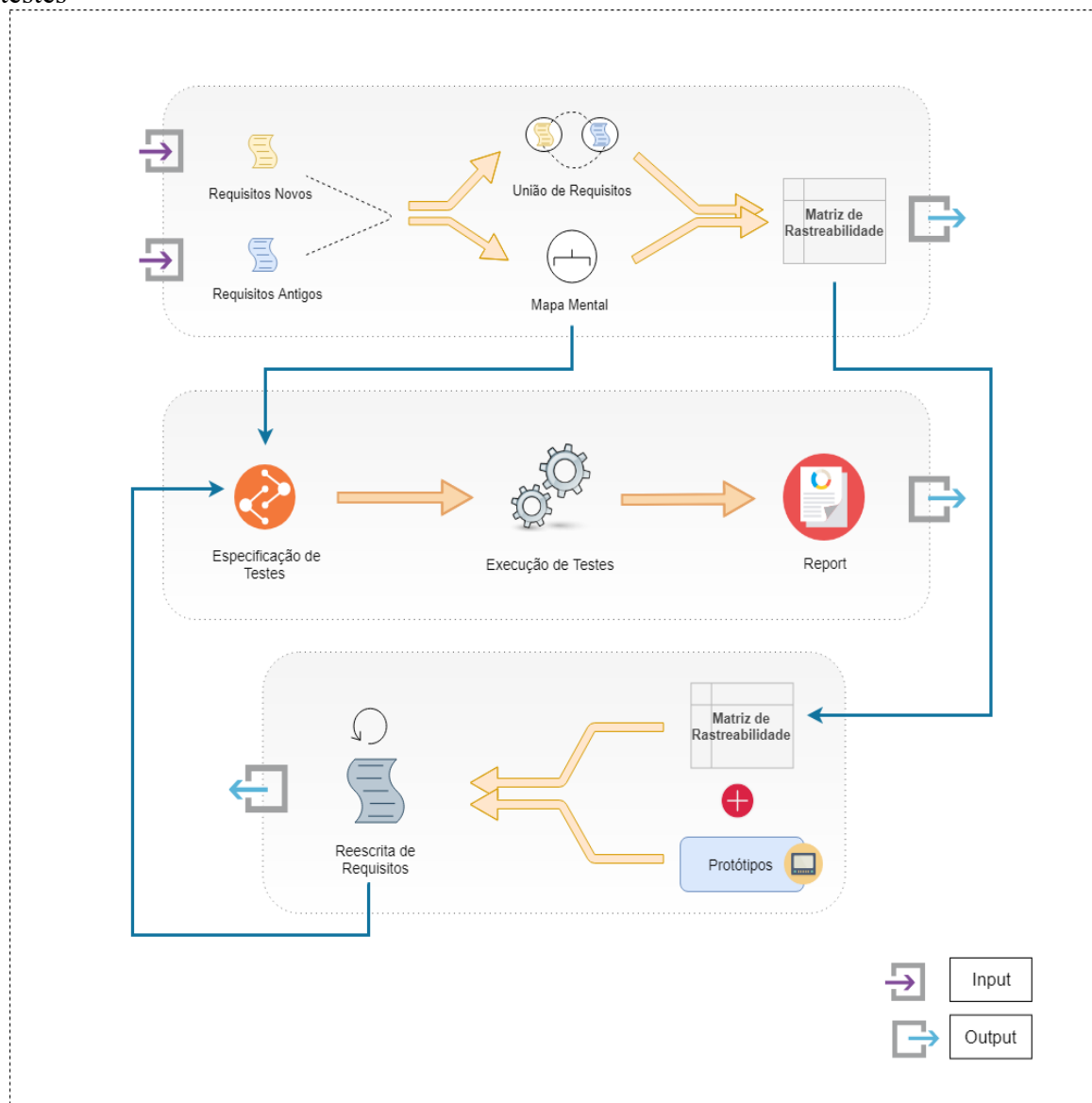
Embora a rastreabilidade de requisitos sofra uma resistência por parte dos processos ágeis, ela é necessária, pois através dela é possível manter os requisitos sempre atualizados, permitindo um melhor entendimento de como o sistema funciona para que o mesmo possa ser testado de forma eficaz, garantindo a qualidade do produto. É impraticável testar o sistema que contém requisitos desatualizados, desta forma mudanças de requisitos também devem ser acompanhadas e registradas durante todo o ciclo de vida do software, o contrário disto resultará em testes obsoletos e a garantia de qualidade baixa ou inexistente, culminando em retrabalho para os responsáveis, ou ainda no acúmulo de inconsistências levando ao fracasso do projeto.

Diante desse contexto e do conhecimento sobre o processo de rastreabilidade conceituado em bases científicas, entende-se que requisitos guiam a criação de testes para o sistema. Mas o que acontece no contexto de um projeto ágil quando a evolução do software e as respostas às mudanças não estão sendo acompanhadas pelo processo de rastreio e atualização da documentação, é que se torna incabível a geração de testes e o projeto evolui sem validações.

Para solucionar este problema, propomos uma abordagem para a rastreabilidade de requisitos guiada por testes, onde testes criados a partir de uma referência de documentação

(de preferência a mais atual) auxiliam nesta rastreabilidade para com outros artefatos existentes, porém desatualizados; incompletos e inconsistentes do software. Entende-se que para construir testes que mantenham a cobertura de qualidade para todas as regras de negócio e funcionalidades do sistema, os requisitos precisam estar atualizados e em conformidade com o produto.

Figura 5: Visão geral da abordagem proposta para a rastreabilidade de requisitos guiadas por testes



Fonte: Elaborada pelo autor (2019).

De acordo com a abordagem descrita na Figura 5, os *inputs* são as entradas de dados necessárias para início da abordagem proposta, neste caso são utilizados os documentos de requisitos como entrada, e que foram ilustrados como requisitos novos e antigos para fazer

referência a um software de desenvolvimento ágil que diante das mudanças produz mais de um artefato. Toda a abordagem resultará em dados de saída (*output*) que serão uma Matriz de Rastreabilidade, uma reescrita de requisitos resultando em uma versão de documento completo com os requisitos elicitados, e um *report* de testes como resultado de um plano de execução de validação.

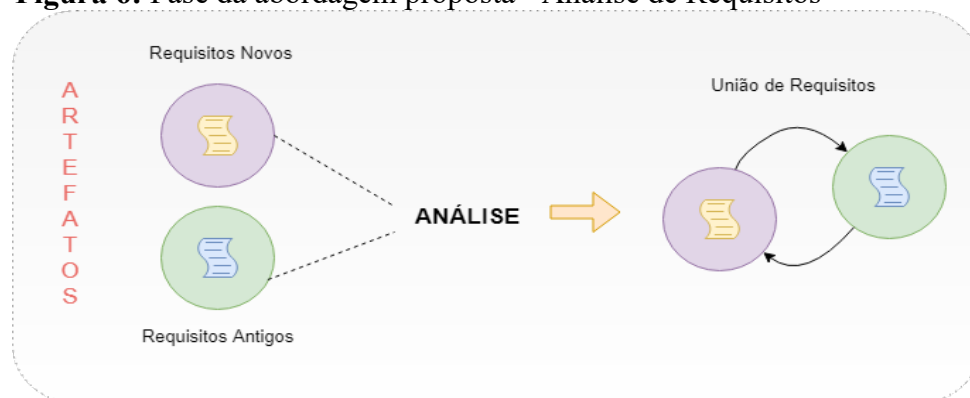
O objetivo é garantir a cobertura dos requisitos, características e funcionalidades do software para execução de testes, mantendo a rastreabilidade de seus requisitos para assegurar a evolução do software com segurança, estabilidade e principalmente qualidade. Cada fase da abordagem será identificada e explicado seu funcionamento para uma melhor compreensão, são elas:

- Análise de requisitos - onde será filtrado os artefatos, principalmente versões de documentos de requisitos e separados para análise de compatibilidade, dependência, ausências.
- Definição de Casos de Teste através de Mapa Mental - onde com base em um documento principal, casos de testes podem ser criados para cobrir seus requisitos e somado a isto o resultado da análise de requisitos influi na criação de mais casos que podem ser melhor compreendidos na diagramação de um mapa mental.
- Matriz de Rastreabilidade - onde o elo de rastreio serão os casos de testes criados no mapa mental que fazem referência aos requisitos das versões dos documentos elicitados, tornando-se possível mapear o relacionamento entre eles.
- Testes - onde a qualidade do software poderá ser validada, após obter o conhecimento dos requisitos essenciais para seu bom funcionamento, durante as etapas anteriores.
- Reescrita de requisitos - por fim será possível identificar através da Matriz o relacionamento de dependência entre requisitos definidos em versões diferentes do documento, e a importância de atualizar outros requisitos esquecidos e assim promover a criação de uma reescrita para geração de um documento atualizado, completo e útil.

3.1.1 Análise de Requisitos

A abordagem proposta neste capítulo envolve inicialmente a análise de Requisitos, como descrito na Figura 6 a seguir:

Figura 6: Fase da abordagem proposta - Análise de Requisitos



Fonte: Elaborada pelo autor (2019).

A Figura 6 revela a aplicação da abordagem na análise de diferentes versões de documentos de requisitos, artefatos que definem o software e suas funcionalidades. É evidenciado, a análise entre requisitos novos e antigos do projeto, quando para este existe sobreposição de documentações, caso que acontece corriqueiramente em projetos ágeis e distribuídos (quando o escopo de um projeto é dividido em partes, e estas são desenvolvidas por equipes diferentes, que podem estar separadas geograficamente) que acontece de estarem com versões diferentes de documentações, por desvio da comunicação e entre outros fatores.

Logo, havendo mais de uma documentação é necessário analisá-las, para investigar diferenças e semelhanças, comparar com as funcionalidades desenvolvidas, certificar que regras continuando existindo, se foram modificadas, ou ainda descartadas na evolução do processo de software. Após a análise, é previsto a união de requisitos, onde é feito o levantamento de identificação dos requisitos em uso, e os de importância para o negócio do projeto.

3.1.2 Definição de Casos de Teste através de Mapa Mental

A partir da análise de Requisitos, já é possível identificar casos de testes que podem ser criados, em conformidade a requisitos antigos que foram esquecidos, porém que possuem relação com o software e, em conformidade a requisitos novos (atuais, gerados pela evolução do processo de desenvolvimento).

Para facilitar então a criação dos casos, uma melhor visualização foi pensada através de Mapas Mentais, como citado por Otaduy e Diaz (2017), são benéficos quanto ao trabalho em colaboração. Os mapas promovem uma visão dos cenários que estão sendo descritos pelos

responsáveis, mas que podem ser compartilhados de forma assíncrona com demais *stakeholders*, possibilitando o feedback rápido, a priorização de casos de teste, a reformulação de cenários, entre outros. Se o cliente demonstrar interesse poderá ter acesso ao mapa criado para entender o que está sendo desenvolvido e os fluxos da sua aplicação.

Figura 7: Exemplo de criação de casos de testes utilizando mapa mental



Fonte: Elaborada pelo autor (2019).

A Figura 7 revela um exemplo prático de como pode ser elaborado um mapa mental e sua estrutura. Podemos indicar um projeto como o destacado no centro do mapa, e este contendo folhas que são os requisitos do sistema, ou as funcionalidades da aplicação, como melhor convém ao estilo do projeto. E cada folha de requisito ou funcionalidade se tornará “pai” de outras folhas que expressam os casos de testes para sua cobertura e validação.

3.1.3 Matriz de Rastreabilidade

A matriz de rastreabilidade de requisitos foi montada a partir da análise dos requisitos (a união destes com relação em conformidade as funcionalidades do sistema) e do mapa mental de casos de testes levantados, que fazem a cobertura destes requisitos.

Figura 8: Fase da abordagem proposta - Geração da Matriz de rastreabilidade

Fonte: Elaborada pelo autor, (2019).

A matriz então objetiva construir o elo entre os requisitos (antigos e novos) através de casos de testes que o entrecortam, facilitando assim o rastreio dos requisitos que precisam ser atualizados, reescritos, e unidos para uma nova versão do documento de contrato, descrita mais à frente na seção 3.1.5 de “Reescrita de Requisitos”. O elo entre os requisitos são os casos de teste citados no mapa mental, o modelo de matriz pode ser descrito como mostra a Tabela 1 a seguir:

Tabela 1: Modelo de matriz de rastreabilidade entre requisitos guiada por casos de teste

MATRIZ DE RASTREABILIDADE				
RASTREABILIDADE POR TESTES DE ACEITAÇÃO		REQUISITOS NOVOS		
		Req_novo_1	Req_novo_2	Req_novo_3
REQUISITOS ANTIGOS	Req_antigo_1			
	Req_antigo_2	CT-001, CT-002		CT-005, CT-006, CT-007
	Req_antigo_3			
	Req_antigo_4		CT-004, CT-002	
	Req_antigo_5			CT-008

Fonte: Elaborada pelo autor (2019).

Logo, de acordo com a tabela os requisitos são mapeados, estando na vertical os requisitos antigos e na horizontal os requisitos novos (sendo estes o ponto de partida para

geração dos casos de testes). Feito o descrito, e tendo o conhecimento da análise de requisitos já realizada é mais fácil visualizar os casos de testes que entrecortam ambos, conectando os requisitos novos a algo descrito nos antigos, por se tratar da mesma funcionalidade, ou citar o mesmo critério de aceitação, entre outros. Esta visão da Matriz é um ponto chave no processo porque facilitará organizar todo o processo de qualidade para um projeto ágil, distribuído que tem sido falho na engenharia de Requisitos, gerando expectativas melhores para o potencial sucesso de um projeto.

3.1.4 Testes

Nesta etapa como descreve a Figura 9, após o levantamento de casos de testes cadastrados na visão do mapa mental é possível direcioná-los para especificação de testes. Especificar significa construir o cenário de teste, colocando os itens necessários como pré-condições, passos sequenciais e os resultados esperados para cada ação, de acordo com o requisito e funcionalidade do Software. Os casos de testes especificados podem ser executados de forma manual ou transformados em testes automáticos e quando executados geram como saída um report de resultados, com bugs relatados e dentre outras informações de relevância para o projeto.

Figura 9: Fase da abordagem proposta - Aplicação de testes para validação a partir do mapa mental



Fonte: Elaborada pelo autor, (2019).

O processo de especificação pode ser feito em ferramentas gratuitas disponíveis na web, uma delas é o TestLink que oferece toda a gerência de testes que um projeto necessita (TESTLINK, 2018). Trabalha com permissões concedidas ao usuário, quando se tem um time de especialistas que dividem atividades e níveis de tarefas (como líder de testes,

testador, entre outros), e com isso a visão das funcionalidades da ferramenta mudam para cada perfil, o que é totalmente adaptável pela equipe de trabalho.

É necessário ter um projeto criado na ferramenta, para dar início ao processo de especificação. O Testlink também trabalha com suítes de teste, cada suíte guarda um conjunto de casos de testes que o definem.

Na abordagem proposta utilizamos o mapa mental como guia (Figura 10), as folhas ‘pai’ são transformadas em suítes de testes, e os filhos destas folhas são os casos de testes a serem especificados.

Figura 10: Casos de testes utilizando mapa mental



Fonte: Elaborada pelo autor (2019).

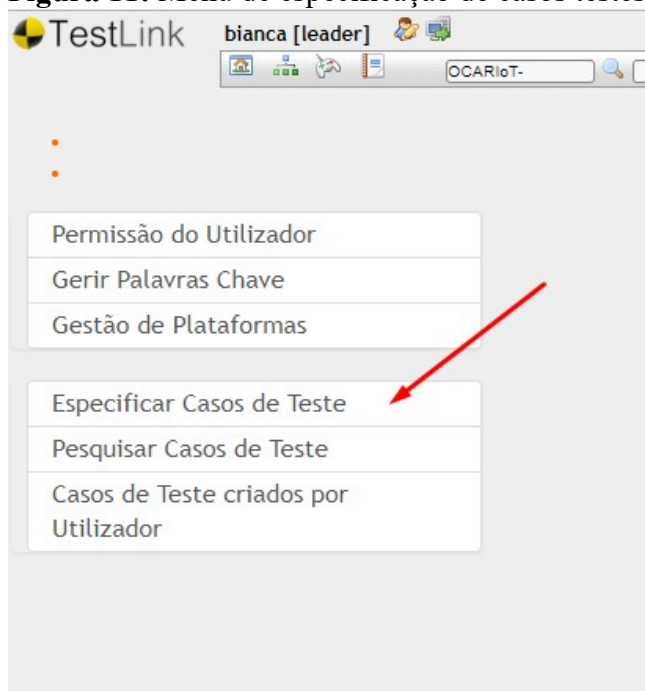
Exemplificando aplicação da Figura 10 no TestLink teríamos:

- **Projeto_Teste** = Nome do projeto criado no TestLink;
- **Requisito_1 / Requisito_2 / Requisito_3** = Suítes de testes, criadas dentro do projeto;
- **CT 1 / CT 2 / CT 3** = Casos de testes criados dentro da suíte de testes “Requisito_3”;
- **CT 1 / CT 2** = Casos de testes criados dentro da suíte de testes “Requisito 2” .

Exemplificando o processo de especificação dos casos de testes na ferramenta, a seguir:

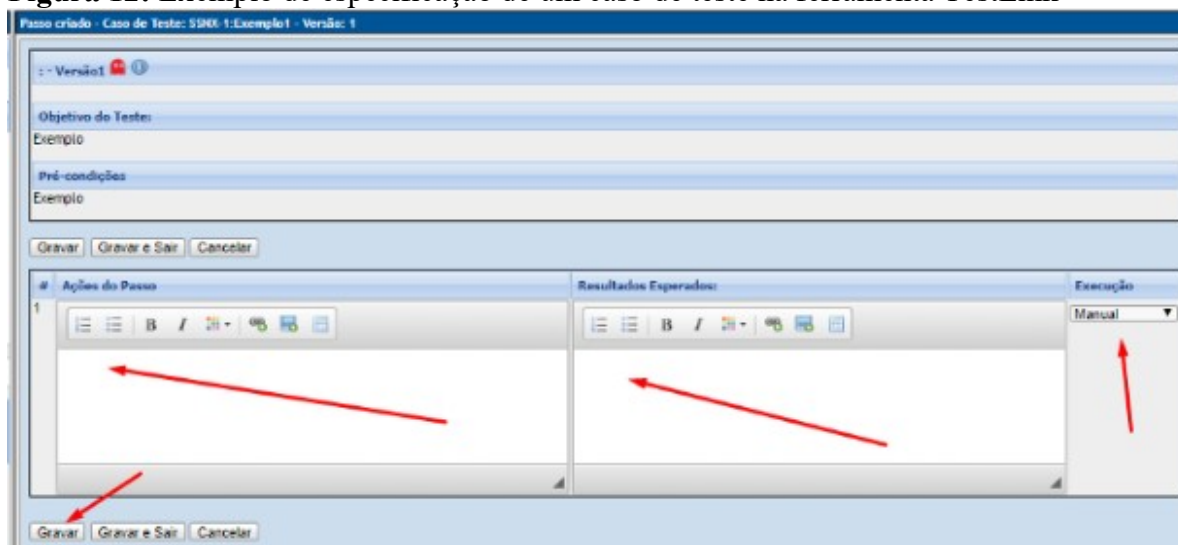
- Para especificar casos de testes, clique em “Especificar casos de testes” do TestLink (Figura 11).

Figura 11: Menu de especificação de casos testes na ferramenta TestLink



Fonte: Elaborada pelo autor (2019), com acesso ao Testlink.

- Após isso (Figura 11) será redirecionado para a página de configurações de casos de testes, e dentro da pasta do projeto criado é possível adicionar suítes de testes para o mesmo. As suítes geralmente representam um requisito, ou funcionalidade específica do sistema a ser testado.
- Os casos de testes são cadastrados dentro da suíte específica, basta clicar sobre a suíte cadastrada e selecionar para adicionar um caso de teste por vez, preenchendo as informações de obrigatoriedade, como nome do teste, objetivos, pré-condições, entre outros.
- Para adicionar a especificação de um caso de teste, contendo os passos a serem executados, basta clicar nele dentro da suíte de teste e especificá-lo, cada passo necessita de um comportamento esperado e pode ser informado o tipo de execução (manual ou automática), clique em “Gravar” ou “Gravar e sair” para que ele adicione outro passo em seguida (Figura 12).

Figura 12: Exemplo de especificação de um caso de teste na ferramenta TestLink

Fonte: <https://medium.com/@luanasreis/testlink-uma-ferramenta-de-gerenciamento-de-testes-de-software-44001b816f64>.

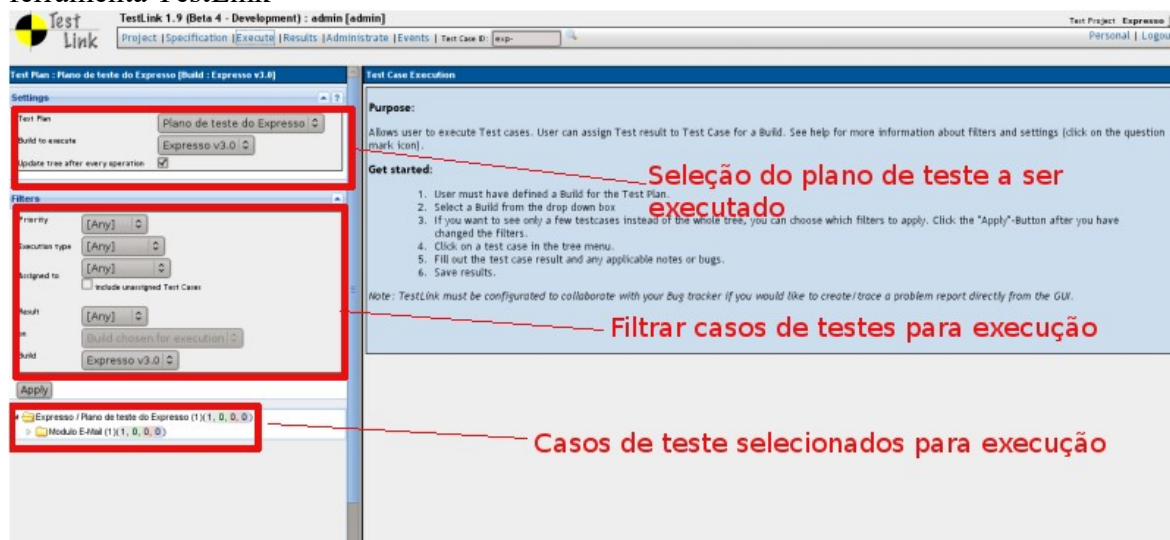
Este é o processo de criação de casos de testes manuais na ferramenta TestLink.

Para a execução de casos de testes é necessário:

- Plano de Teste: cada Plano de Teste está relacionado à um Projeto de Teste, é a base para execução de casos de testes, nele contém nome, descrição, conjunto de casos de teste, versão, resultados de teste, atribuição de teste e definição de prioridade.
- Build / Release: A execução do teste precisa de uma versão (Build) para acontecer, isto porque durante o desenvolvimento do software muitas builds podem ser liberadas para verificação de qualidade. Cada build precisa ser versionada e adicionada casos de testes para sua execução.

Tendo um plano de teste e uma Build é possível associar casos de testes para execução e a visualização dos casos será da seguinte forma no Testlink (Figura 13):

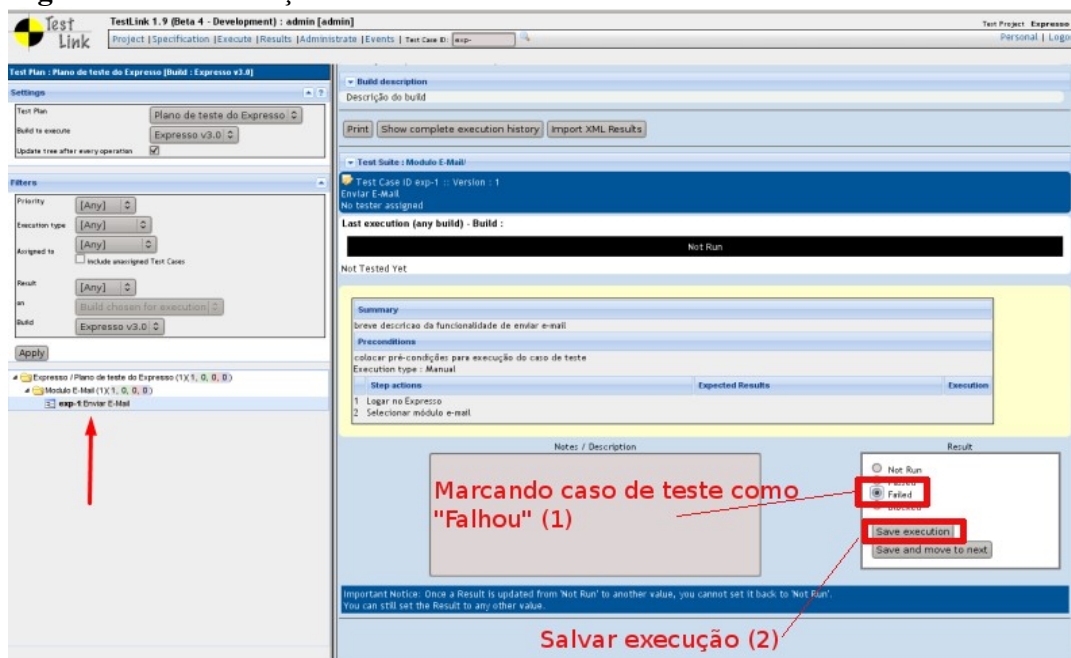
Figura 13: Exemplo de visualização de casos de testes disponíveis para execução na ferramenta TestLink



Fonte: <https://medium.com/@luanasreis/testlink-uma-ferramenta-de-gerenciamento-de-testes-de-software-44001b816f64>.

Na árvore à esquerda (Figura 13) no inferior da tela, mostram-se os casos de testes disponíveis para a execução. Ao clicar na suíte e em seguida clicar em um caso de teste específico, a visualização será (Figura 14):

Figura 14: Visualização de um caso de teste a ser executado na ferramenta TestLink



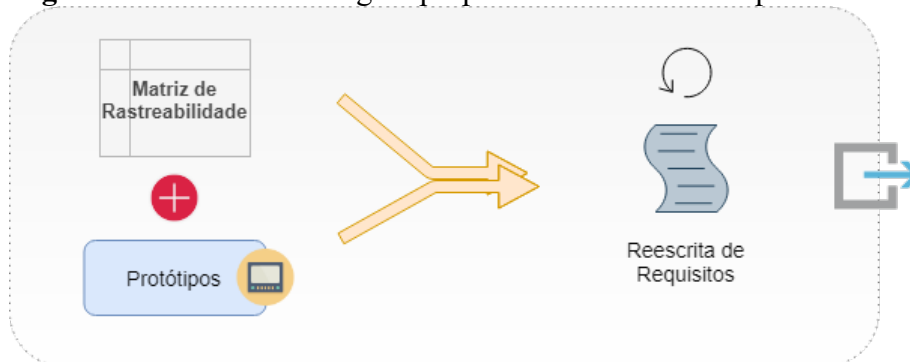
Fonte: <https://medium.com/@luanasreis/testlink-uma-ferramenta-de-gerenciamento-de-testes-de-software-44001b816f64>.

Executando a sequência de passos (Figura 12) de forma manual basta selecionar o resultado do teste (1) e salvar a execução (2). Os resultados são salvos automaticamente para os casos de testes que vão sendo executados e que podem ser: passou; falhou; bloqueado. E no próprio Testlink é possível gerar os relatórios (*report*) de testes, com gráficos, métricas e entre outros recursos disponíveis, facilitando o feedback para com a equipe de desenvolvimento, tendo o resultado das *builds* que estão sendo geradas, podendo aplicar o controle de *bugs* e as correções.

3.1.5 Reescrita de Requisitos

A Reescrita de Requisitos (Figura 15) é um resultado para a abordagem aplicada, ela tem o objetivo de estabelecer para o projeto a gerência de Requisitos. Ambos artefatos utilizados: requisitos antigos, requisitos novos e protótipos de telas auxiliaram a reformulação de um documento de contrato de requisitos mais preciso, completo e que acompanha a evolução de desenvolvimento do software. Assim, a gerência poderá ser aplicada eficazmente.

Figura 15: Fase da abordagem proposta - Reescrita de requisitos



Fonte: Elaborada pelo autor (2019).

De acordo com a Figura 15, a matriz apresenta a visão completa de elos entre os requisitos, o que levará a equipe saber quais os requisitos estão em uso, quais são essenciais no negócio da aplicação, e levará a visualizar também os que não foram mapeados com os casos de testes, por não possuírem relacionamento (Requisito antigo x Requisito novo), mas que tem vínculo com o sistema e precisam ser considerados. Os protótipos de telas, caso exista como artefato do projeto, também agregam valor na remodelação para a reescrita de requisitos, eles mostram o design das telas e a interação entre elas, exemplificando seu fluxo.

Na prática a reescrita servirá como documento oficial atualizado, contendo todos os requisitos que o sistema necessita para o bom funcionamento e para está em conformidade com as necessidades do cliente. Para isso, toda a fase (Figura 15) é aplicada e levada a feedback dos envolvidos.

3.2 Considerações Finais

Neste capítulo foi apresentada a abordagem proposta para o problema discutido anteriormente, onde em projetos de desenvolvimento ágeis respostas rápidas às mudanças também podem refletir em negligências para com o processo de rastreabilidade de requisitos. A abordagem proposta é definida pela rastreabilidade de requisitos sendo guiada por testes e é composta das seguintes fases: Análise de requisitos, Definição de casos de testes através do mapa mental, Matriz de rastreabilidade, Testes e Reescrita de requisitos.

4 ESTUDO DE CASO: RASTREABILIDADE DE REQUISITOS GUIADA POR TESTES NO OCARIoT DASHBOARD

Este capítulo descreve um estudo de caso no projeto OCARIoT-Dashboard, onde foi aplicada a abordagem proposta no capítulo anterior. Sendo então o nosso objeto de experimentação, apresentaremos os resultados obtidos e os seus benefícios no contexto da aplicação.

4.1 OCARIoT Dashboard

O OCARIoT “é um projeto co-financiado pelo Programa HORIZON 2020 da União Europeia e pelo Ministério da Ciência, Tecnologia e Inovação do Brasil através da Rede Nacional de Ensino e Pesquisa (RNP)” (OCARIoT, 2017).

Tem o objetivo de desenvolver uma solução personalizada de *coaching* baseada na Internet das coisas - IoT, que orienta as crianças a adotar comportamentos saudáveis, e que permitirá observar padrões de atividades na vida cotidiana das crianças, a evolução da sua saúde, parâmetros fisiológicos e comportamentais e ainda dados ambientais. Todas estas informações combinadas com os padrões médicos permitirá que o OCARIoT forneça um plano personalizado e individual de *coaching* para obesidade, concedendo as crianças permanecerem ativas e engajadas em seu bem-estar e gerenciamento de hábitos saudáveis. (OCARIoT, 2017).

Figura 16: Como funciona o projeto OCARIoT



Fonte: OCARIoT (2017).

A Figura 16 apresenta como funciona a solução OCARIoT da seguinte forma:

- A rede IoT sincronizará os dados provenientes das ferramentas que compõem a solução: *SmartBand* conectado através do *bluetooth*, o aplicativo-App OCARIoT, e o Dashboard OCARIoT.
- A criança sendo o alvo da solução utilizará a *SmartBand* conectada via *bluetooth* no celular para monitoramento de suas atividades físicas e qualidade do sono, e o App para oferecer missões e tarefas mediadas por um treinador virtual.
- O Responsável que terá no Dashboard a visão da sincronização de dados do APP utilizado pela criança, monitorando assim o seu desempenho.

Tendo conhecimento da solução em sua visão geral, apresentamos a seguir a visão específica do nosso estudo de caso que foi aplicado especificamente para uma das ferramentas que compõem o OCARIoT, o Dashboard.

O foco do painel OCARIoT é a promoção de mudanças comportamentais em direção a hábitos saudáveis das crianças dos locais piloto, que são eles: Espanha, Grécia e Brasil. O objetivo é permitir o uso e compartilhamento de dados gerados para profissionais de saúde, educadores e famílias (pais, parentes ou tutores), para que eles possam ter acesso a dados reais a qualquer momento. O Dashboard está fornecendo suporte para as seguintes funcionalidades: acompanhamento de alimentação, rastreamento de atividades e atividades educacionais.

4.2 Rastreabilidade de Requisitos Guiada por Testes no OCARIoT Dashboard

A abordagem proposta para rastrear requisitos através de testes foi aplicada visando atender as necessidades de nosso objeto de estudo, OCARIoT Dashboard. Ele faz parte de um projeto de desenvolvimento ágil, e distribuído, onde no seu entendimento foi descoberto que a engenharia de requisitos era falha, consequentemente a gerência de requisitos também. Com isso o processo de qualidade e validação para o Dashboard não podia ser iniciado, era inviável a criação de casos de testes que cobrisse todos os cenários, pois existia conflitos entre os artefatos de desenvolvimento, e conflitos entre comunicação das partes distribuídas. As regras de negócio estavam espalhadas e não havia sincronia com as prioridades das funcionalidades a serem desenvolvidas.

Visando solucionar este problema, a abordagem (Figura 5) foi proposta e aplicada ao OCARIoT Dashboard, a fim de reorganizar o processo de desenvolvimento para validação do produto que está em andamento, garantindo a cobertura dos requisitos, características e funcionalidades do software para execução de testes. E manter a rastreabilidade dos seus requisitos para assegurar a evolução do mesmo. Então a partir da versão mais atual de um documento de requisitos foi possível gerar os primeiros casos de testes, estes foram utilizados como guia para rastrear os relacionamentos de dependências com outros requisitos de diferentes versões, identificando os requisitos em uso na aplicação.

Nas subseções a seguir explicaremos como a abordagem proposta foi aplicada ao OCARIoT Dashboard, seguindo todos os passos definidos na Figura 5 (a figura da abordagem do Cap. 3): Análise de Requisitos, Definição de Casos de Teste através de Mapa Mental, Matriz de rastreabilidade, Testes e Reescrita de Requisitos.

4.2.1 Análise de Requisitos

A análise de requisitos é a primeira etapa da abordagem descrita (Figura 6), nela a análise dos artefatos de regras de negócio precisam ser feitas e como dados de entrada (*inputs*) teremos um documento de requisitos novos (atuais) e um documento de requisitos antigos. Isto porque no OCARIoT tivemos conflitos com as documentações existentes, e por ser um projeto distribuído aconteceu que ambas as partes utilizavam artefatos como modelo de desenvolvimento, distintos. Foi na etapa de análise que observou-se as inconsistências entre funcionalidade e documentações e o processo de rastreabilidade começou a ser implantado através da criação dos primeiros casos de testes.

Inicialmente verificamos os artefatos disponíveis, os citados requisitos antigos (Apêndice A) e os requisitos novos (Apêndice B) voltados para nossa aplicação OCARIoT Dashboard. Depois fizemos um emparelhamento das informações, tendo em vista a primeira versão funcional também entregue para a qualidade, foi um processo árduo e custoso, porém necessário para aplicação de uma gerência de requisitos e garantia de qualidade.

Seguimos alinhando os requisitos antigos que estavam esquecidos, porém eram considerados no funcional da aplicação, e destacamos as informações que existiam no antigo e no novo em conformidade, ou que se completavam. Sabendo que os requisitos novos eram atualmente o documento parâmetro de desenvolvimento, nada poderia lhe alterar. Então poderíamos apenas lhe acrescentar, somando a ele informações necessárias de negócio que os

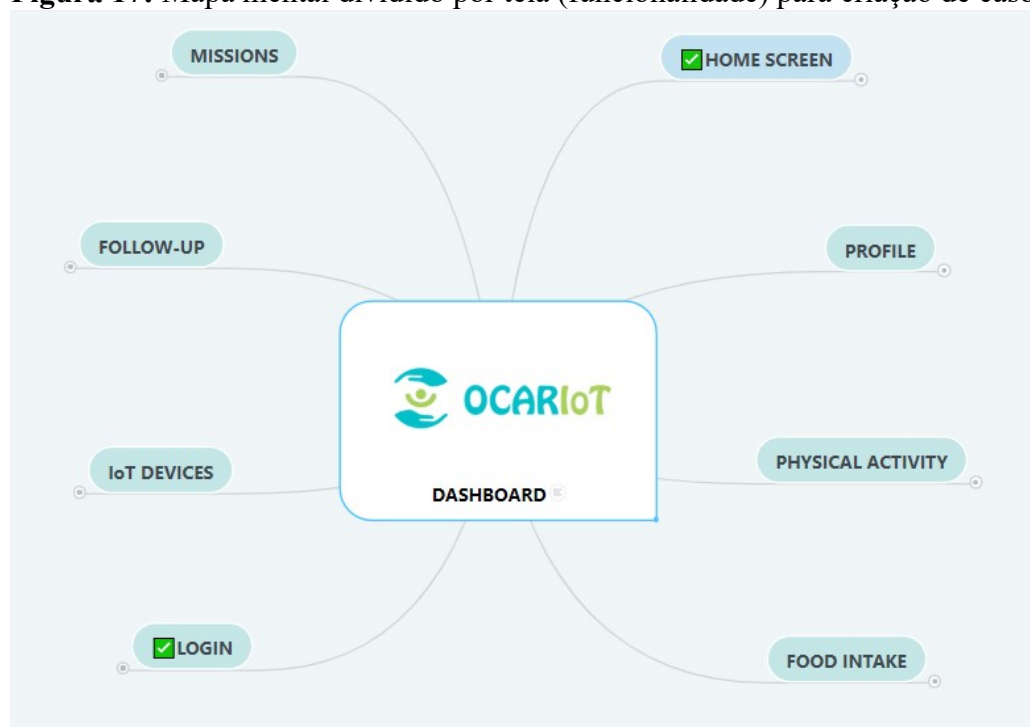
requisitos antigos descreviam, e que eram essenciais para o projeto. Nessa etapa, apenas destacamos nos próprios documentos as regras que se integravam, e as que eram lógicas essenciais para o projeto. Facilitando a visão de análise para dar continuidade ao processo de rastreabilidade.

4.2.2 Definição de Casos de Teste através de Mapa Mental

A definição dos casos de testes através do mapa mental foi possível após a fase de análise dos requisitos, e quando construídos tornou as especificações de requisitos mais compreensíveis, obtendo o entendimento das regras de negócio do sistema.

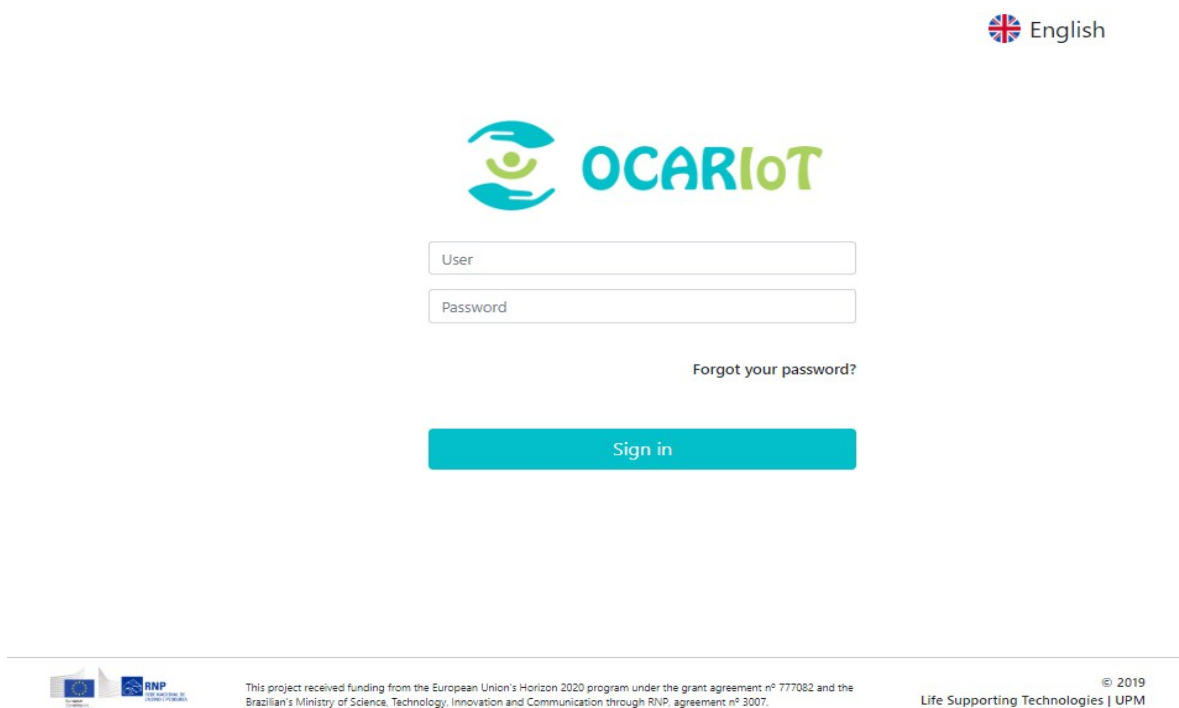
O documento parâmetro de referência para o desenvolvimento dos casos de testes foi o documento de requisitos mais atual (Apêndice B), como este é organizado por funcionalidades da aplicação, seguimos o mesmo padrão para especificação dos testes, pelas funcionalidades. Ou seja, a partir desse documento, foram definidos casos de teste como sugestão a cobertura dos requisitos, e foram estruturados em forma de mapa mental, seguindo a divisão de telas (funcionalidades) do dashboard citados nas imagens a seguir:

Figura 17: Mapa mental dividido por tela (funcionalidade) para criação de casos de testes



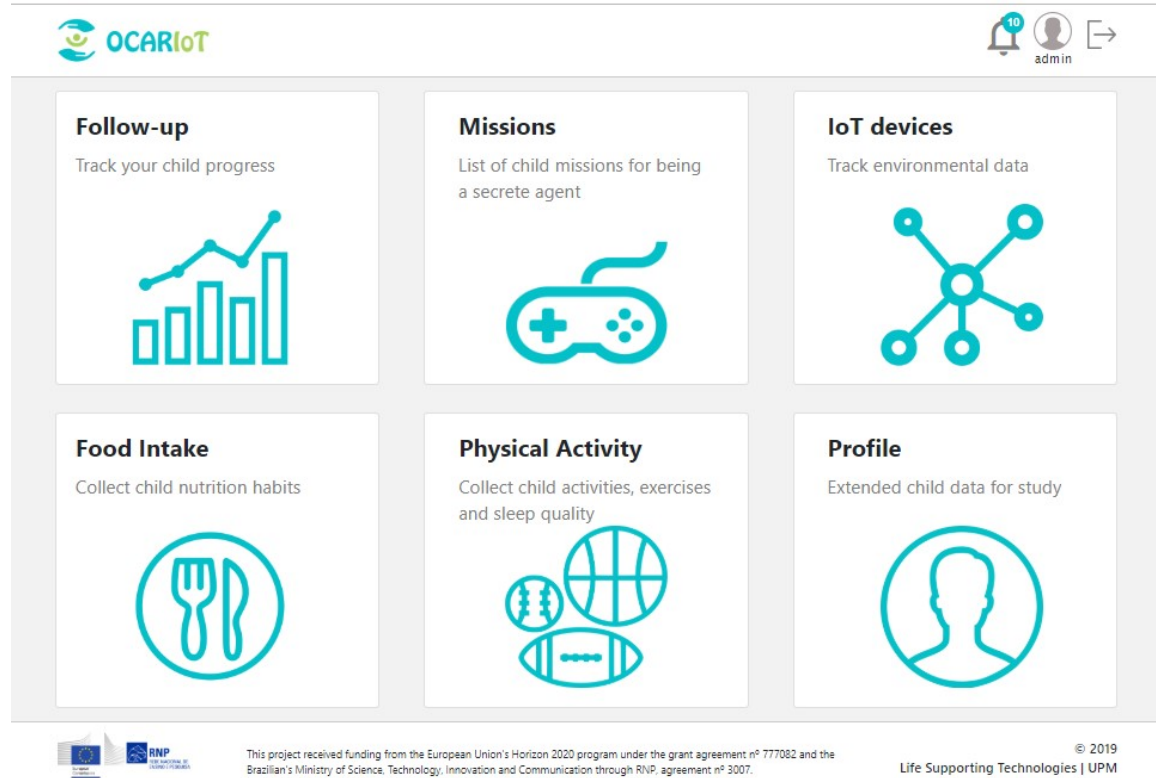
Fonte: Elaborada pelo autor (2019).

Figura 18: Tela de Login da aplicação OCARIoT- Dashboard



Fonte: OCARIoT (2019).

Figura 19: Tela Home da aplicação OCARIoT- Dashboard



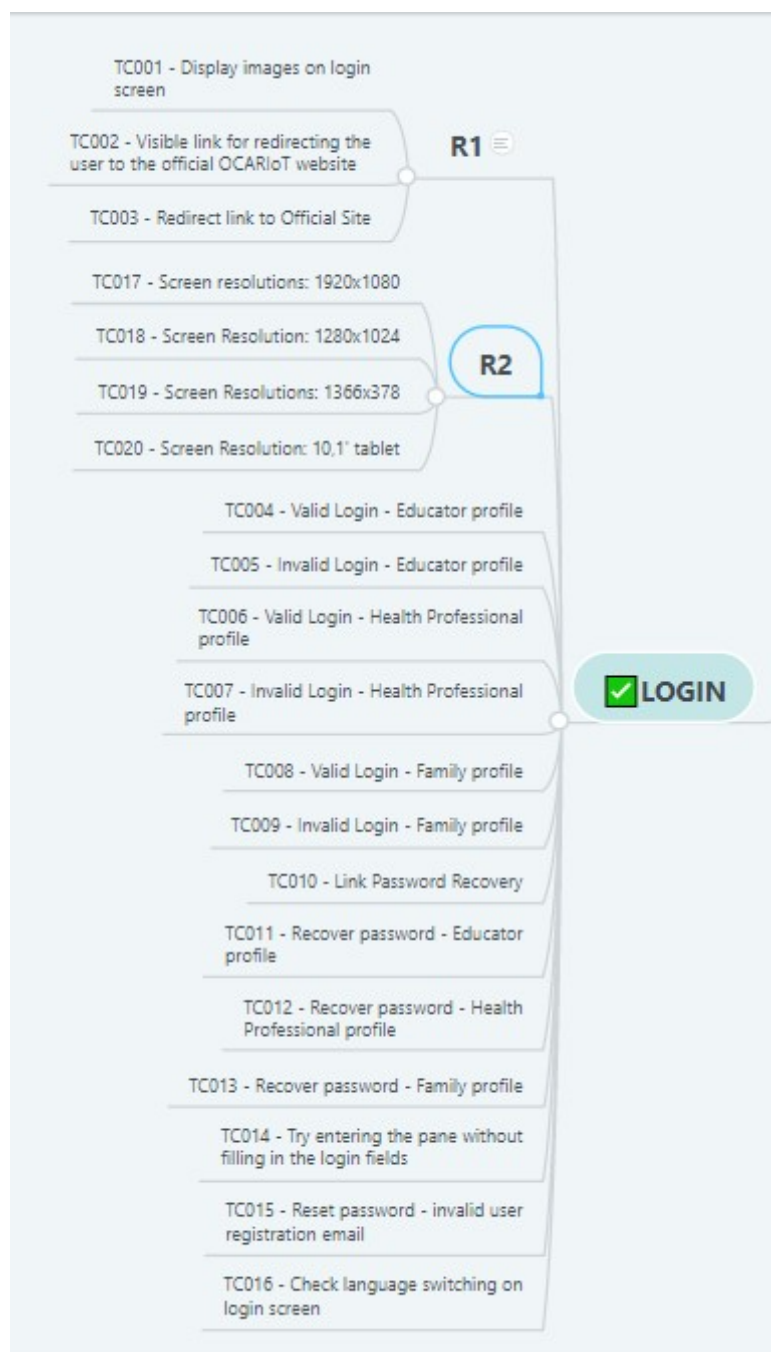
Fonte: OCARIoT (2019).

O OCARIoT se divide em oito funcionalidades principais: Login (Figura 19), Home Screen (Figura 17), Follow-Up, Missions, IoT Devices, Food Intake, Physical Activity e Profile. Ambos possuem seus requisitos citados no documento atual de regras de negócio. Estruturado o nosso diagrama, iniciou a elaboração dos casos de testes para cobertura dos requisitos que eram os parâmetros atuais de desenvolvimento. Finalizado, foi percebido a ausência de validações importantes para a funcionalidade, e após a análise era sabido os requisitos antigos que concordavam e completavam os requisitos novos, estes então precisavam ser mapeados em casos de testes para a completude da lógica de rastreabilidade.

A seguir é exemplificado a funcionalidade de Login e Home e os casos de testes que foram criados para estes, em linguagem natural e compreensível para os envolvidos no negócio. Os resultados para as demais funcionalidades da aplicação podem ser consultadas no Apêndice C. Todavia, o somatório total de casos de testes desenhados no mapa mental, como resultado para todas as funcionalidades do sistema foram 188 casos. Fragmentado nas imagens por funcionalidade nesta seção e no Apêndice C em continuidade.

- Login - Primeiro contato com a aplicação, atividade de credenciamento para acesso ao sistema

Figura 20: Definição de casos de testes por requisito para a funcionalidade de Login da aplicação

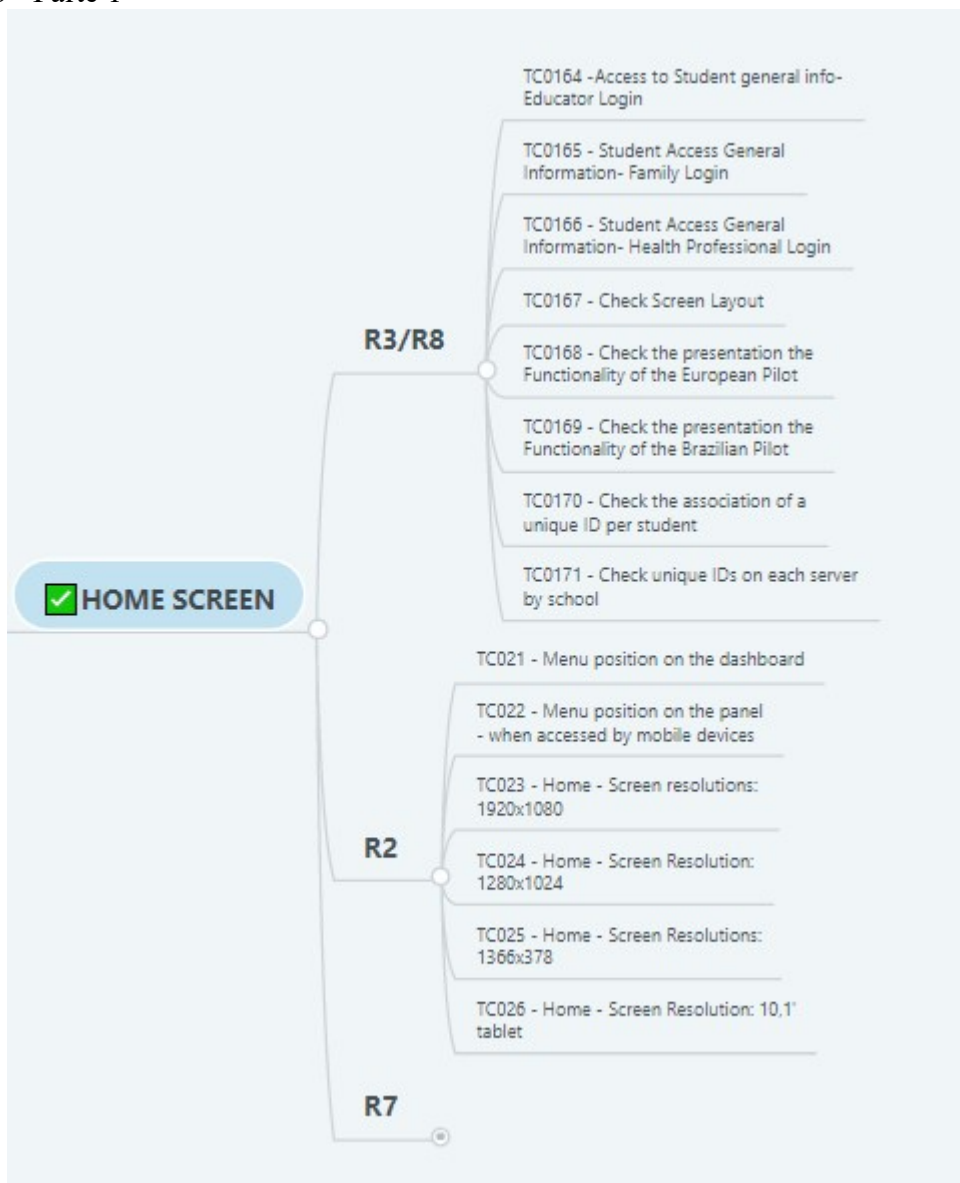


Fonte: Elaborada pelo autor, (2019).

De acordo com a Figura 20, após a definição do projeto (nó pai) seguem as folhas dependentes (LOGIN), os filhos são os requisitos (R1, R2 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. Já os casos de testes que estão soltos dentro da dependência do nó 'pai' são os que não estão mapeados dentro desses requisitos atuais, e que podem ser resultados da completude com requisitos antigos e características essenciais da própria tela funcional.

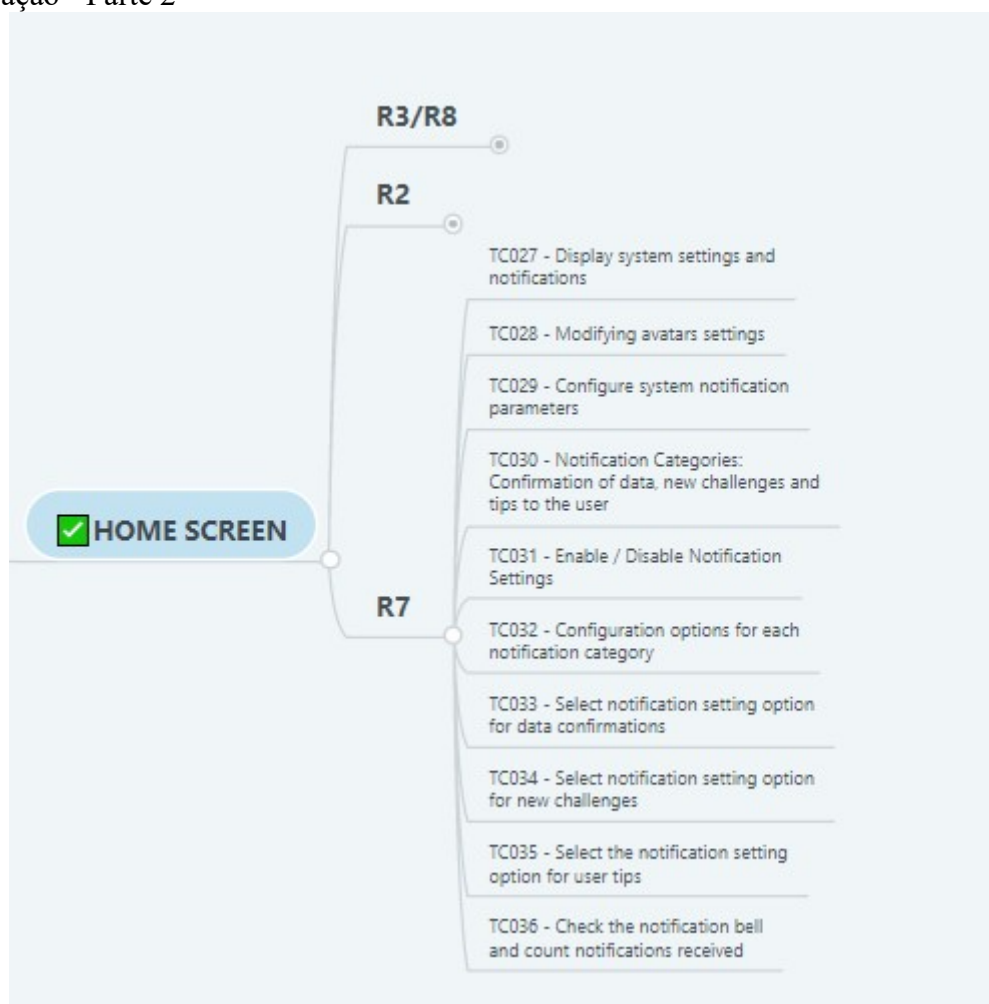
- Home Screen - é a tela principal do sistema, nela estão disponíveis todas as funcionalidades do mesmo.

Figura 21: Definição de casos de testes por requisito para funcionalidade da Home Screen da aplicação - Parte 1



Fonte: Elaborada pelo autor (2019).

Figura 22: Definição de casos de testes por requisito para a funcionalidade da Home Screen da aplicação - Parte 2



Fonte: Elaborada pelo autor (2019).

De acordo com as Figuras 21 e 22, após a definição do projeto (nó pai) seguem as folhas dependentes (Home Screen), os filhos são os requisitos (R3/ R8, R2, R7- Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento.

Assim foram mapeados casos de testes que fazem a cobertura por funcionalidade da aplicação, utilizando como parâmetro de desenvolvimento o documento de Requisitos novos (Apêndice B) e a completude com os requisitos antigos (Apêndice A) que lhe são comuns também quanto às suas funcionalidades.

4.2.3 Matriz de Rastreabilidade

A matriz de rastreabilidade foi derivada a partir da escrita dos casos de testes, ou seja, a medida que um teste precisava ser escrito tendo como referência o documento mais atual de requisitos, se fazia a ligação entre os requisitos através deste.

Os testes proporcionaram a análise conjunta dos requisitos e sua completude, logo estabeleceram um relacionamento de ligação entre eles. Para isso, a matriz auxilia na visão macro dos requisitos que se conectam por uma relação de dependência, e os testes potencializam essa conexão ao tornar claro a composição dos requisitos.

Tabela 2: Matriz de rastreabilidade de requisitos guiada por testes para o OCARIoT

MATRIZ DE RASTREABILIDADE DE REQUISITOS											
Rastreabilidade por: Testes de Aceitação	R1	R2	R3	R4	REQUISITOS ATUAIS			R7	R8	R9	R10
	Req_01				TC0172...TC0179						
Req_02				TC0113...TC0119							TC0121...TC0133
Req_03											
Req_06											
Req_07			TC0164...TC0166			TC0162		TC0165...TC0166	TC037...TC039, TC044...TC045, TC052 ...TC053	TC0124...TC0125, TC0134, TC0136, TC0138	
Req_10											
Req_11					TC064 ... TC073					TC043	TC0121...TC0133
Req_12					TC069 ...TC073					TC041 ...TC042, TC051...TC054	TC0134...TC0138, TC0141
Req_14											
Req_15/ Req_26			TC0164...TC0171					TC0164...TC0171			
Req_17		TC017 ... TC020, TC023...TC026, TC055...TC058, TC079...TC082, TC097...TC0100, TC0104 ...TC0107, TC0142...TC0145, TC0154 ...TC0157								TC037...TC038, TC044...TC045, TC052...TC053	TC0123...TC0124, TC0134, TC0136
Req_18											
Req_19											
Req_22				TC0113...TC0120, TC0172...TC0179							TC0121...TC0138, TC0141
Req_23											
Req_24											TC0128...TC0133
Req_25						TC064 ... TC073				TC043	TC0121...TC0133
Req_27											
Req_28	TC001...TC003										
Req_29/ Req_30/ Req_31											

Fonte: Elaborada pelo autor (2019).

A matriz definida na Tabela 2 foi dividida por requisitos novos (Apêndice B) e requisitos antigos (Apêndice A), o cruzamento do relacionamento foi através dos casos de testes definidos no mapa mental.

Deve-se compreender que na Matriz são representados apenas os casos de testes que fazem elo de ligação com ambas versões de requisitos. Para demonstrar que é possível rastreá-los por dependência para com a regra de negócio do sistema, e que ambos precisam ser mapeados em um documento oficial.

O que não impede compreender, que houveram casos de testes que não foram utilizados no rastreamento, porque provavelmente só especificava a um dos versionamentos de requisitos, não havendo relacionamento entre ambos. Como também casos de testes se repetiram dentro do mapeamento da rastreabilidade, justamente porque não é um relacionamento 1 para 1, mas há compartilhamento entre um ou mais requisitos.

Um dos fatores para este compartilhamento é os requisitos novos terem sido mapeados para uma funcionalidade ou mais de uma, e os requisitos antigos são a nível global do

sistema. Então, é de se esperar que um requisito de uma versão global possa estar se relacionando com mais de um requisito escrito por funcionalidade e virse-versa.

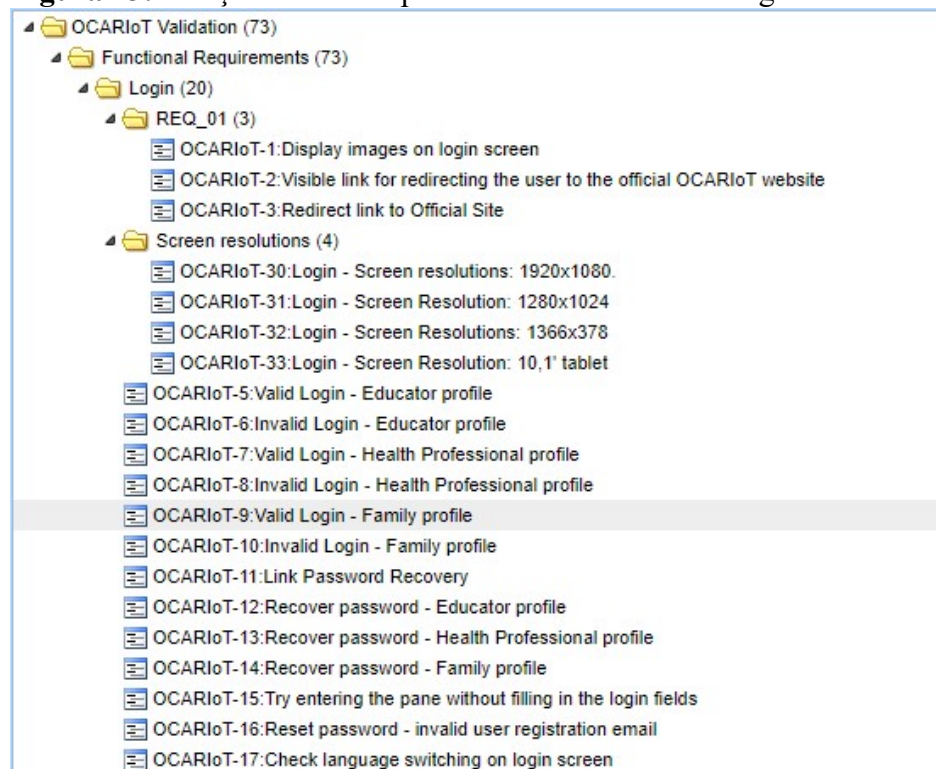
4.2.4 Testes

Os testes desenhados no mapa mental foram especificados utilizando a ferramenta já descrita, TestLink. A partir deste momento, a validação da qualidade funcional começa a ser executada com a escrita das especificações para os cenários do mapa. Aplicamos o primeiro experimento para duas funcionalidades do sistema, o login e a home screen.

Na ferramenta criamos um projeto, e dentro deste projeto criamos suítes de testes uma para representar a pasta de requisitos funcionais, e dentro dela suítes para as duas funcionalidades em experimento, como mostra as figuras a seguir:

- LOGIN:

Figura 23: Criação dos testes para a funcionalidade de Login do OCARIoT-Dashboard

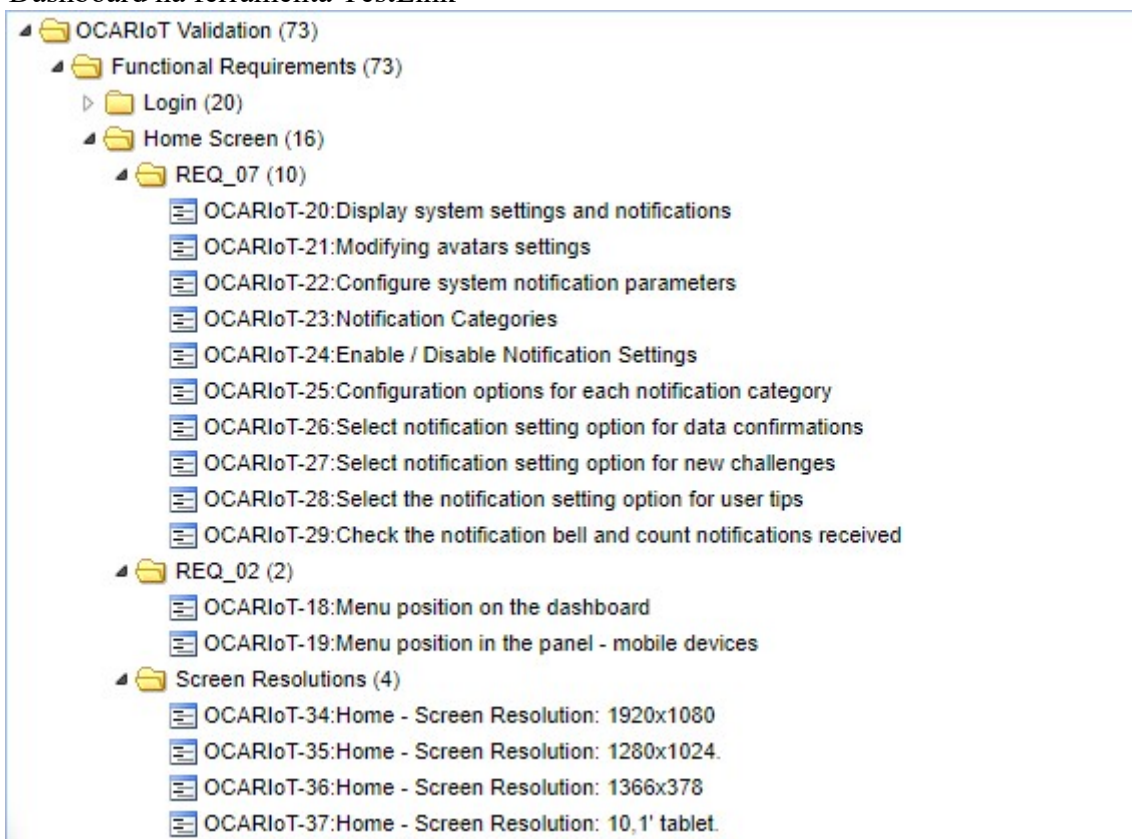


Fonte: Elaborada pelo autor (2019).

De acordo com a Figura 23, a suíte **REQ_01** representa o **R1** e a suíte ‘*Screen resolutions*’ representa o **R2**, ambos da diagramação do mapa mental para esta funcionalidade. Todos os casos de testes foram originados do mapa mental.

- HOME SCREEN:

Figura 24: Criação dos testes para a funcionalidade de Home Screen do OCARIoT-Dashboard na ferramenta TestLink



Fonte: Elaborada pelo autor (2019).

De acordo com a Figura 24, a suíte de teste **REQ_07** representa o **R7** e as suítes **REQ_02** e ‘*Screen Resolutions*’ representam o **R2** na diagramação do mapa mental para esta funcionalidade. Todos os casos de testes foram originados do mapa mental.

4.2.4.1 Especificação

A especificação como já definida descreve um cenário, ou fluxo da aplicação. É criada através de uma sequência de passos, cada um com um resultado esperado do sistema para verificar um comportamento da aplicação.

Nosso objeto de experiência foram as funcionalidades de Login e Home screen do dashboard IscarIoT e estão exemplificadas a seguir:

- LOGIN

Figura 25: Exemplo 1 - cenário de caso de teste construído para a funcionalidade de Login do dashboard

OCARIoT-2 : Visible link for redirecting the user to the official OCARIoT website - Versão1			
Objetivo do Teste:			
Verify that in the dashboard you can see the redirect link to the official project page			
Pré-condições			
1. Get access to the OCARIoT Dashboard			
#	Ações do Passo	Resultados Esperados:	Execução
1	Given I am accessing the OCARIoT Dashboard site	Then I'll see the app login screen	Manual
2	And on the login page you should see a link to the official project website	Then the dashboard displays link in the login, to guide the user to the official OCARIoT website	Manual
Estado : Final Prioridade : Média Tipo de Execução : Automatizado Estimativa da Execução (min) : 5.00			

Fonte: Elaborada pelo autor (2019).

Esta é a visão (Figura 25) de um caso de teste construído na ferramenta TestLink. O título do cenário é: “*Visible link for redirecting the user to the official OCARIoT website*”. Logo, o objetivo do teste é verificar se um link que redireciona o usuário para o site oficial do OCARIoT é visível para o mesmo na tela de login do dashboard. A pré-condição para executar este teste é ter acesso ao Dashboard.

A sua verificação é feita pela execução de dois passos definidos no caso de teste. E este tem as características definidas no final da especificação. Como, o seu estado, prioridade, tipo de execução e estimativa da execução.

Figura 26: Exemplo 2 - cenário de caso de teste construído para a funcionalidade de Login



OCARIoT-5 : Valid Login - Educator profile - Versão1			
Objetivo do Teste:			
Log in with a valid user in the educator profile.			
Pré-condições			
1. Access the OCARIoT dashboard 2. Have an educator login			
#	Ações do Passo	Resultados Esperados:	Execução
1	Given I am accessing the OCARIoT Dashboard site	Then I'll see the app login screen	Manual
2	I type a valid user with educator profile	Then I visualize the filled user field	Manual
3	I fill in the valid profile password	And I view the password field filled anonymously	Manual
4	I click on sign in	Then I enter the panel and see the application start screen	Manual
Estado : Final Prioridade : Média Tipo de Execução : Automatizado Estimativa da Execução (min) :			

Fonte: Elaborada pelo autor (2019).

Este exemplo (Figura 26) é um cenário de teste para verificar o login válido de um perfil de educador no sistema. A pré-condição para executá-lo é acessar o Dashboard e ter ou conhecer um login válido de educador na aplicação. Foi definido em uma sequência de 4 passos, possuindo o resultado esperado para cada um para verificar que com o login válido é possível acessar e visualizar a tela inicial do Dashboard.

- HOME SCREEN

Figura 27: Exemplo 1 - cenário de caso de teste construído para a funcionalidade de Home screen do dashboard

OCARIoT-20 : Display system settings and notifications - Versão1  			
Objetivo do Teste:			
Check icon that displays system notifications and settings.			
Pré-condições			
1. Have access to the panel			
2. Have an valid login			
#	Ações do Passo	Resultados Esperados:	Execução
1	Given I am accessing the OCARIoT Dashboard site	Then I'll see the app login screen	Manual
2	I type a valid user and password and click in sign in	Then I enter the panel and see the application start screen	Manual
3	And check the panel for the presence of the system notifications icon	The alerts icon is displayed at the top right of the application	Manual
4	And click in notifications icon	Then on the same screen, a modal is displayed with system notifications	Manual
5	And I see in the modal a gear icon	The gear icon is displayed to configure notifications and system customization.	Manual
Estado : Rascunho Prioridade : Média Tipo de Execução : Manual Estimativa da Execução (min) : 5.00			

Fonte: Elaborada pelo autor (2019).

O título do cenário (Figura 27) é: “Display system settings and notifications”. Logo, o objetivo do teste é verificar se o Dashboard exibe as configurações se notificações do sistema. A pré-condição para executar este teste é ter acesso ao Dashboard e ter um login válido para acesso a tela inicial da aplicação.

A sua verificação é feita pela execução de cinco passos definidos no caso de teste, cada um com seu resultado esperado.

Figura 28: Exemplo 2 - cenário de caso de teste construído para a funcionalidade de Home screen do dashboard

OCARIoT-25 : Configuration options for each notification category - Versão1			
Objetivo do Teste:			
Check Configuration Options for each notification category:			
<ul style="list-style-type: none"> • Once a day • Twice a week • Once a week 			
Pré-condições			
1. Have access to the panel			
2. Have an valid login			
#	Ações do Passo	Resultados Esperados	Execução
1	Given I am accessing the OCARIoT Dashboard site	Then I'll see the app login screen	Manual
2	I type a valid user and password and click in sign in	Then I enter the panel and see the application start screen	Manual
3	And click no notifications icon	Then on the same screen, a modal is displayed with system notifications	Manual
4	And I see in the modal a gear icon	The gear icon is displayed to configure notifications and system customization.	Manual
5	And click the gear icon for settings	Then a modal configuration is presented for customizing notifications	Manual
6	And I see the notification configuration categories and their default system options	Then, for each notification category is presented as default customization options:	Manual
<ul style="list-style-type: none"> • Once a day • Twice a week • Once a week 			
Estado : Rascunho Prioridade : Média			
Tipo de Execução : Manual			
Estimativa da Execução (min) : 5.00			

Fonte: Elaborada pelo autor (2019).

Este segundo exemplo (Figura 28) objetiva testar o cenário que valida as opções de configuração para cada categoria de notificações que são apresentadas no Dashboard. É descrita em uma sequência de 6 passos para validação do resultado esperado.

4.2.4.2 Execução

Como visto anteriormente, para se executar testes especificados na ferramenta TestLink é necessário criar um plano de teste, e uma build que envolva os casos a serem testados. Na nossa experimentação duas *builds* foram criadas para a funcionalidade de login, uma que testou suas funcionalidades na cobertura dos seus Requisitos, e outra destinada apenas a verificação do comportamento da tela em diferentes resoluções. (Screen Resolutions).

A Evidência demonstra os resultados por Plano de teste, gerando a matriz por Suítes de teste e dentro destas os casos de testes especificados.

- Build destinada ao plano de testes do LOGIN - Matriz de resultados de testes da execução:

Figura 29: Matriz de resultado de execução de testes

Projecto de Teste: OCARIoT Validation						
Plano de Testes : Login Test Plan v1.2: 77ba1c77234fba8941aa1ce29ce80aed7e10a6a7						
Caso de Teste	Priori...	Login - v1.2: 77ba1c...	[Última Build] Login - ...	Exec Notes for Late...	Última Execução	Notas (última Execu...
Suite de Testes: Functional Requirements/Login (13 Items)						
OCARIoT-5:Valid Login - E...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-6:Invalid Login - ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-7:Valid Login - H...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-8:Invalid Login - ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-9:Valid Login - F...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-10:Invalid Login ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-11:Link Passwor...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-12:Recover pass...	Média	Bloqueado [v1]	Bloqueado [v1]	Not-implemented pa...	Bloqueado [v1]	Not-implemented pa...
OCARIoT-13:Recover pass...	Média	Bloqueado [v1]	Bloqueado [v1]	Not-implemented pa...	Bloqueado [v1]	Not-implemented pa...
OCARIoT-14:Recover pass...	Média	Bloqueado [v1]	Bloqueado [v1]	Not-implemented pa...	Bloqueado [v1]	Not-implemented pa...
OCARIoT-15:Try entering t...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-16:Reset passw...	Média	Bloqueado [v1]	Bloqueado [v1]	Not-implemented pa...	Bloqueado [v1]	Not-implemented pa...
OCARIoT-17:Check langua...	Média	Falhado [v1]	Falhado [v1]	Dashboard behavior...	Falhado [v1]	Dashboard behavior...
Suite de Testes: Functional Requirements/Login/REQ_01 (3 Items)						
OCARIoT-1:Display image...	Média	Falhado [v1]	Falhado [v1]	There are no images...	Falhado [v1]	There are no images.
OCARIoT-2:Visible link for ...	Média	Falhado [v1]	Falhado [v1]	There is no redirect l...	Falhado [v1]	There is no redirect l...
OCARIoT-3:Redirect link to...	Média	Bloqueado [v1]	Bloqueado [v1]	Not implemented the...	Bloqueado [v1]	Not implemented the...

Fonte: Elaborada pelo autor (2019).

É possível ver o resultado da execução de cada especificação, e caso exista, mostra a anotação que descreve uma situação ocorrida durante a execução. Que normalmente serve como justificativa para o comportamento do teste. Porque o mesmo pode ter sido falho ou bloqueado, facilitando a descrição para futuros testes de Regressão.

- Build destinada ao plano de testes do LOGIN screen resolutions - Matriz de resultados de testes da execução:

Figura 30: Matriz de resultado de execução de testes para um plano do login em específico os requisitos de Screen Resolutions

Projecto de Teste: OCARIoT Validation									
Plano de Testes : Login - Screen Resolutions									
Caso de Teste	Pri...	Build Login - ...	Login - Scre...	Login - Scre...	Login - Scre...	Última Build...	Exec Notes f...	Última Exec...	Notas (últim...
Suite de Testes: Functional Requirements/Login/Screen resolutions (4 Items)									
OCARIoT-30:L...	M...	Passou [...]	Passou [...]	Passou [...]	Não Exec...	Não Exec...		Passou [...]	
OCARIoT-31:L...	M...	Passou [...]	Passou [...]	Passou [...]	Não Exec...	Não Exec...		Passou [...]	
OCARIoT-32:L...	M...	Falhado [...]	Falhado [...]	Falhado [...]	Não Exec...	Não Exec...		Falhado [...]	Considering ...
OCARIoT-33:L...	M...	Passou [...]	Passou [...]	Passou [...]	Não Exec...	Não Exec...		Passou [...]	

Fonte: Elaborada pelo autor (2019).

- Build destinada ao plano de testes da HOME SCREEN -Matriz de resultados de testes da execução:

Figura 31: Matriz de resultado de execução de testes para um plano da Home Screen

Caso de Teste	Prioridade	Home Screen - Build	Última Build] Home Sore...	Exec Notes for Latest (ore...	Última Execução	Notas (Última Execução)
Suite de Testes: Functional Requirements/Home Screen/REQ_02 (2 Items)						
OCARIoT-18:Menu position on the ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-19:Menu position in the p...	Média	Falhado [v1]	Falhado [v1]	According to the specifica...	Falhado [v1]	According to the specifica...
Suite de Testes: Functional Requirements/Home Screen/REQ_07 (9 Items)						
OCARIoT-20:Display system settin...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-22:Configure system noti...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-23:Notification Categories	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-24:Enable / Disable Notifi...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-25:Configuration options ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-26>Select notification sett...	Média	Bloqueado [v1]	Bloqueado [v1]	Test impossible to comple...	Bloqueado [v1]	Test impossible to comple...
OCARIoT-27>Select notification sett...	Média	Bloqueado [v1]	Bloqueado [v1]	Test impossible to comple...	Bloqueado [v1]	Test impossible to comple...
OCARIoT-28>Select the notification ...	Média	Bloqueado [v1]	Bloqueado [v1]	Test impossible to comple...	Bloqueado [v1]	Test impossible to comple...
OCARIoT-29:Check the notification ...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
Suite de Testes: Functional Requirements/Home Screen/Screen Resolutions (4 Items)						
OCARIoT-34:Home - Screen Resol...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-35:Home - Screen Resol...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-36:Home - Screen Resol...	Média	Passou [v1]	Passou [v1]		Passou [v1]	
OCARIoT-37:Home - Screen Resol...	Média	Passou [v1]	Passou [v1]		Passou [v1]	

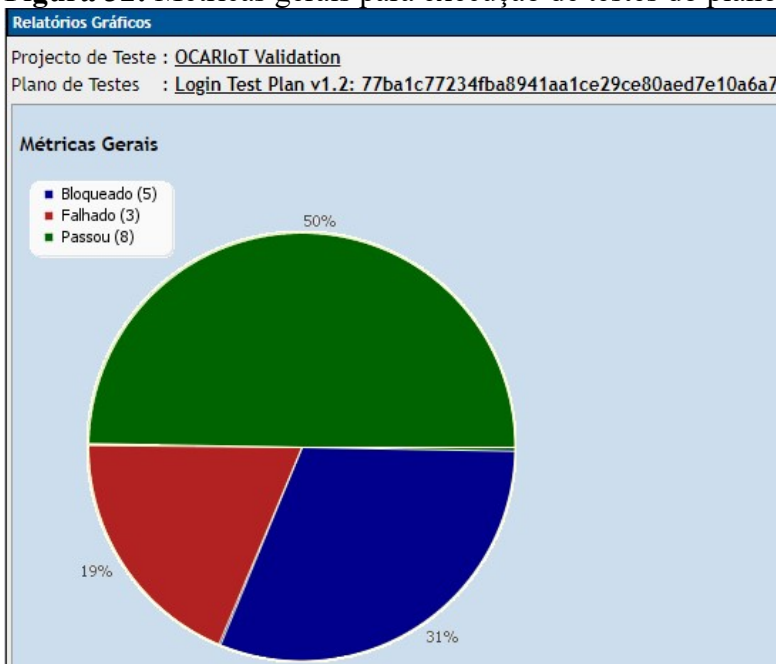
Fonte: Elaborada pelo autor (2019).

4.2.4.3 Resultados

É possível visualizar de forma gráfica o resultado da execução da *Build* para os Plano de testes exemplificados, demonstrando as métricas gerais de casos de testes Bloqueados, Falhados, e que passaram na execução. Facilitando o feedback rápido para as partes interessadas, com o *report* das execuções.

- Métricas Gerais do plano de teste: LOGIN

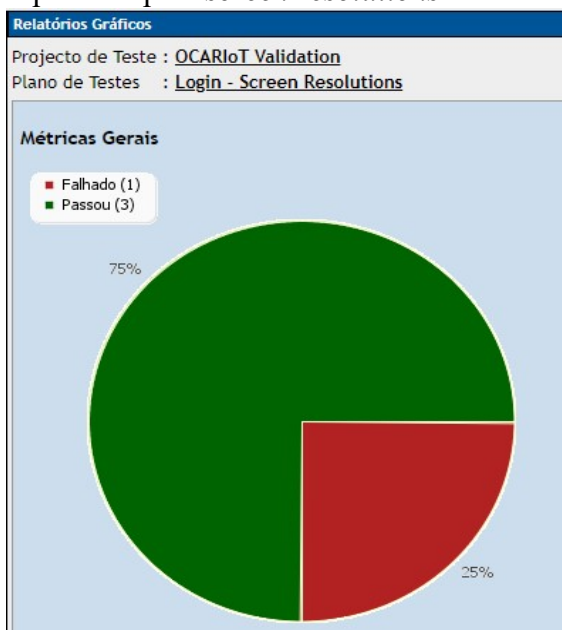
Figura 32: Métricas gerais para execução de testes do plano de Login



Fonte: Elaborada pelo autor (2019).

- Métricas Gerais do plano de teste: LOGIN - SCREEN RESOLUTIONS

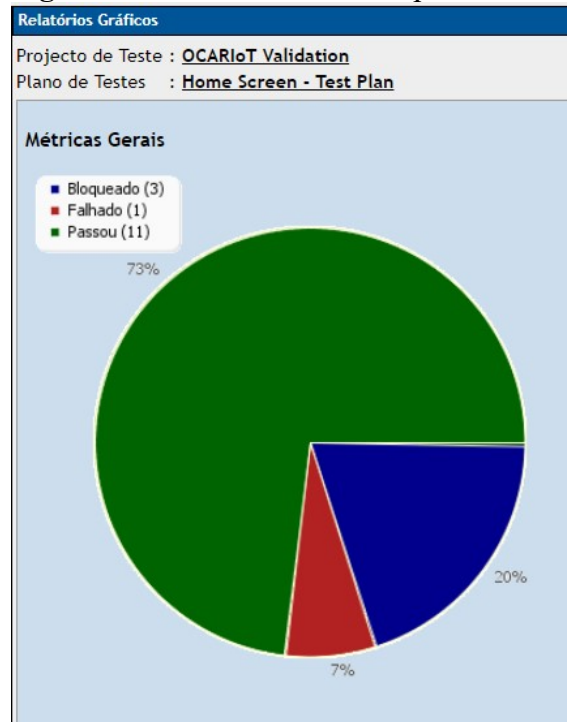
Figura 33: Métricas gerais para execução de testes do plano de Login específico para *screen resolutions*



Fonte: Elaborada pelo autor (2019).

- Métricas Gerais do plano de teste: HOME SCREEN

Figura 34: Métricas Gerais do plano de teste específico para home screen



Fonte: Elaborada pelo autor (2019).

- Relatório Geral de métrica de execução para os casos de testes que contemplam LOGIN e HOME do sistema:

Figura 35: Relatório Geral de métrica de execução para os casos de testes que contemplam LOGIN e HOME

Screen	#Total	#Passing	#Failing	#Blocked
Login	20	55.00%	20.00%	25.00%
Home	15	73.33%	6.67%	20.00%
TOTAL	35	66.00%	12.00%	22.00%

Fonte: Elaborada pelo autor (2019).

A coluna vertical do #Total representa a quantidade de casos de testes executados nas *builds* para cada funcionalidade (Login = 20 casos especificados e Home = 15 casos). E a porcentagem destes casos que tiveram na sua execução, resultados #Passing (passaram na

execução), *#Failing* (falharam na execução) e *#Blocked* (foram bloqueados). E o *#total* na horizontal da tabela faz um somatório geral das execuções.

No caso do OCARIoT os resultados *#Failing* são justificados pela presença de falhas no software na execução de casos de testes, como uma funcionalidade não estar respondendo ao comportamento esperado consolidado ao requisito e entre outros. Já os resultados *#Blocked* foram casos de testes impossibilitados de serem executados, por na maioria das vezes, não ser uma funcionalidade ainda desenvolvida no software.

Desta forma obtém-se um relatório completo e rápido das execuções de testes para uma aplicação de software.

4.2.5 *Reescrita de Requisitos*

A reescrita de requisitos foi elaborada como uma proposta sugestiva para o OCARIoT, com a formulação de um novo documento de requisitos que guiasse a evolução do seu desenvolvimento com mais precisão, acompanhando as mudanças ocorridas e podendo ser atualizado sempre que preciso. Como explicado na **seção 3.1.5**, a reescrita de requisitos foi resultado de uma abordagem de avaliação entre os requisitos novos e antigos do sistema, cujo relacionamento foi mapeado na Matriz (Tabela 2) através dos casos de testes, e entre os protótipos atuais disponíveis como artefato para o desenvolvimento do OCARIoT.

Para seu desenvolvimento foi estruturado um modelo de *template* (Tabela 3), que descreve como foi feita a reescrita dos requisitos para o OCARIoT Dashboard.

Tabela 3: *Template* gerado para estruturação da reescrita de requisitos

TEMPLATE PARA REESCRITA DO REQUISITO	
REESCRITA (valor referenciando o requisito) obs: a reescrita deve ser feita seguindo o padrão do documento atual de requisitos utilizado como referência.	
Base de Dados para preenchimento do template	Casos de Testes da Matriz por requisito
	Protótipo de tela por Requisito
	Design funcional implementado

Fonte: Elaborada pelo autor (2019).

A reescrita então, seguiu a estrutura do último documento que continha os requisitos novos (Apêndice B), e foi definida utilizando a mesma sequência (R1, R2, R3, R4, R5, R6, R7, R8, R9, R10), porém agora agregando os requisitos rastreados pela matriz, e outros que sejam importantes para o bom funcionamento do software. Logo, como descrito no template (Tabela 3) seu resultado convém da união de requisitos cujos casos de testes são os elos de relacionamento na matriz de rastreabilidade, e a comparação entre telas de protótipo e design funcional implementado já na aplicação.

Desta forma, segue a proposta de como resultou a reescrita de Requisitos para o Dashboard-OCARIoT nas funcionalidades de Login e Home Screen, podendo ser, acompanhado pela visualização da Matriz de rastreabilidade (Tabela 2). A reescrita das demais funcionalidades podem ser consultadas no Apêndice D, bem como a visualização do processo de compilação (Apêndice E).

Reescrita R1

Base de dados: União dos escopos do R1 (requisito novo) com o Req_28. Os casos de testes referentes a essa união compreendem: CASOS DE TESTES DA MATRIZ

Casos de Testes mapeados na Matriz

Req_28 + R1 = TC001...TC003

Porém na rastreabilidade percebe-se que pode haver uma maior união de requisitos novos e antigos que englobam o mesmo cenário, neste caso o recurso de Login. Identifica-se então os Req_06, Req_10, como participante do mesmo cenário.

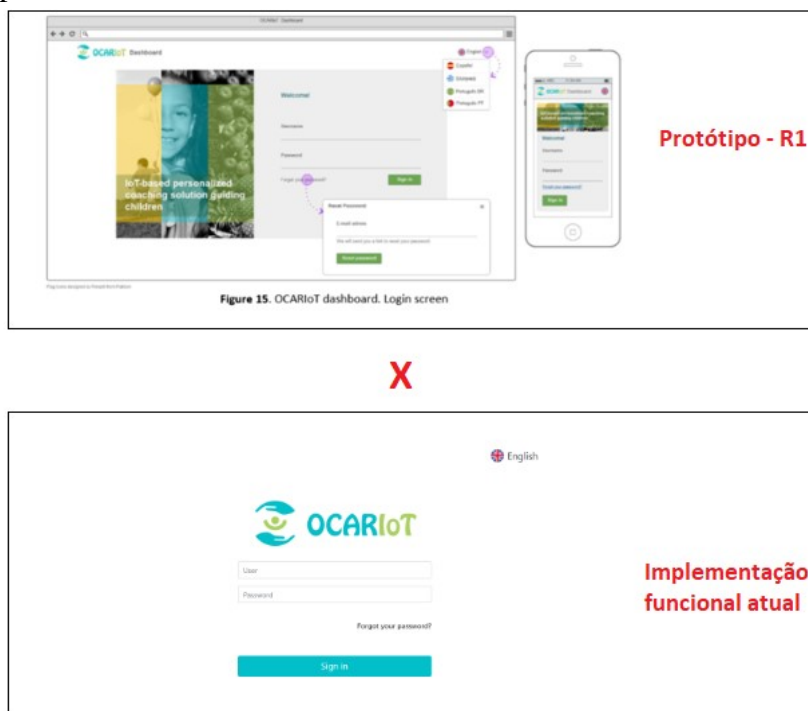
A cobertura de testes passa a ser:

Casos de Testes Gerais:

Req_28 + R1 = TC001...TC003
 Req_06 + R1 = TC004 ... TC009 , TC014
 Req_10 + R1 = TC010 ... TC013, TC015

Base de dados: Protótipo referente ao R1 (requisito novo) versus Tela funcional implementada no OCARIoT.

Figura 36: Comparação entre protótipo e tela funcional implementada no OCARIoT - referente ao R1



Fonte: Elaborada pelo autor (2019).

A análise do protótipo ao funcional, somado ao resultado da Matriz de rastreabilidade faz-se necessária para uma reescrita composta e adequada ao negócio do sistema, e ao que já está sendo desenvolvido e acompanhado pelo cliente.

Como resultado desta rastreabilidade e das comparações de design e implementação é desenvolvida a seguinte proposta de Reescrita para união de ambos os requisitos:

R1

Este requisito está estritamente relacionado à percepção pessoal do usuário sobre a atratividade do login do painel e a sua qualidade. Para isto alguns componentes devem ser adicionados para auxiliar na percepção dos usuários, como:

- 1) Criar uma landing Page;
- 2) Incluir um link para o site OCARIoT oficial para visualização de material multimídia (vídeo, áudio, entrevistas, histórias de sucesso) gerado pelos sites piloto.

Facilitando assim o envolvimento do usuário na aplicação quando este pode compreender o seu objetivo.

Para as funcionalidades do Login o sistema deve se comportar da maneira mais coerente seguindo as especificações a seguir:

1. Permite ao usuário acessar as funcionalidades customizadas de acordo com o seu papel. Sabendo que os papéis no painel OCARIoT se qualificam pelo perfil dos Stakeholders e são eles: Educador, Profissional de Saúde, Familiar (Parente).
2. Para efetuar Login no painel deve ser inserido pelo usuário cadastrado: Login e senha válidos e ativo na plataforma.
3. É necessário que o sistema faça validação das credenciais ao tentar fazer login na plataforma. E ao inserir credenciais que não são válidas o sistema deve exibir erro de login.

É comum os usuários perderem o controle ou o gerenciamento de suas contas. Portanto, para permitir o uso contínuo, é necessário que haja um recurso de recuperação de senha, descrita desta forma:

1. Caso o usuário tenha esquecido sua senha, ele pode recuperá-la através da página de login .
2. O sistema deve enviar uma mensagem (ou email) para recuperação de senha
3. O Usuário deve escolher uma nova senha para atualização.
4. A senha deve ser validada de acordo com as regras de segurança do sistema.

Para uma melhor visualização da reescrita de requisitos está exposta a figura 49 que demonstra a compilação para a reescrita de um requisito final, a partir da visão dos requisitos rastreados pela matriz com elo de ligação.

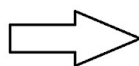
Figura 37: Processo de compilação de requisitos para a versão de um requisito R1 final

CÓDIGO	DESCRIÇÃO	FEEDBACK ENDEREÇADO
R1	Refine o Login: 1) crie uma landing page e 2) inclua um link para o site OCARIoT oficial para visualizar o material multimídia (vídeo, áudio, entrevistas, histórias de sucesso) gerado pelos sites piloto.	EF1, EF15

Requisito novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_28	Manutenção e Suporte	Provedor técnico	Introduzir usuários iniciantes no aplicativo e no painel	Apresentar informações de embarque no início do aplicativo / painel	Todos
Req_06	Funcional	Crianças, família, educador e profissional de Saúde	Recurso de Login	Permite ao usuário acessar as funcionalidades customizadas de acordo com o seu papel	Todos
Req_10	Funcional	Crianças, família, educador e profissional de Saúde	Recuperação de Senha	É comum os usuários perderem o controle ou o gerenciamento de suas contas. Portanto, para permitir o uso contínuo, é necessário que haja um recurso de recuperação de senha	Todos

Requisitos Antigos



Processo de Compilação para reescrita de um Requisito final

Reescrita - Requisito

R1: Este requisito está estritamente relacionado à percepção pessoal do usuário sobre a atratividade do login do painel e a sua qualidade. Para isto alguns componentes devem ser adicionados para auxiliar na percepção dos usuários, como:

- 1) crie uma landing page e,
- 2) incluir um link para o site OCARIoT oficial para visualização de material multimídia (vídeo, áudio, entrevistas, histórias de sucesso) gerado pelos sites piloto.

Facilitando assim o envolvimento do usuário na aplicação quando este pode compreender o seu objetivo.

Para as funcionalidades do Login o sistema deve se comportar da maneira mais coerente seguindo as especificações a seguir:

- 1) Permite ao usuário acessar as funcionalidades customizadas de acordo com o seu papel. Sabendo que os papéis no painel OCARIoT se qualificam pelo perfil dos Stakeholders e são eles: Educador, Profissional de Saúde, Familiar (Parente).
- 2) Para efetuar Login no painel deve ser inserido pelo usuário cadastrado: Login e senha válidos e ativo na plataforma.
- 3) É necessário que o sistema faça validação das credenciais ao tentar fazer login na plataforma. E ao inserir credenciais que não são válidas o sistema deve exibir erro de login.

É comum os usuários perderem o controle ou o gerenciamento de suas contas. Portanto, para permitir o uso contínuo, é necessário que haja um recurso de recuperação de senha, descrita desta forma:

- 1) Caso o usuário tenha esquecido sua senha, ele pode recuperá-la através da página de login .
- 2) O sistema deve enviar uma mensagem (ou email) para recuperação de senha
- 3) O Usuário deve escolher uma nova senha para atualização.
- 4) A senha deve ser validada de acordo com as regras de segurança do sistema.

Fonte: Elaborada pelo autor (2019).

Reescrita R2

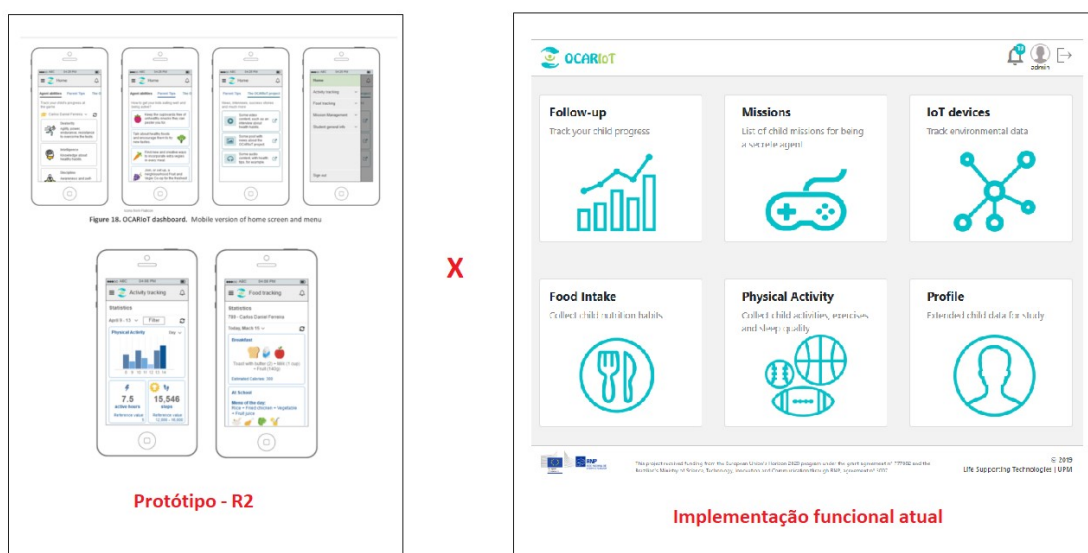
Base de Dados: Somatório do escopo do R2 (requisito novo) mais o Req_17 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_17 + R2 = TC017 ...TC020, TC023...TC026, TC055...TC058, TC079...TC082, TC097...TC0100, TC0104 ...TC0107, TC0142...TC0145, TC0154 ...TC0157

Base de dados: Protótipo referente ao R2 (requisito novo) versus Tela funcional implementada no OCARIoT.

Figura 38: Comparação entre protótipo e tela funcional implementada no OCARIoT - referente ao R2



Fonte: Elaborada pelo autor (2019).

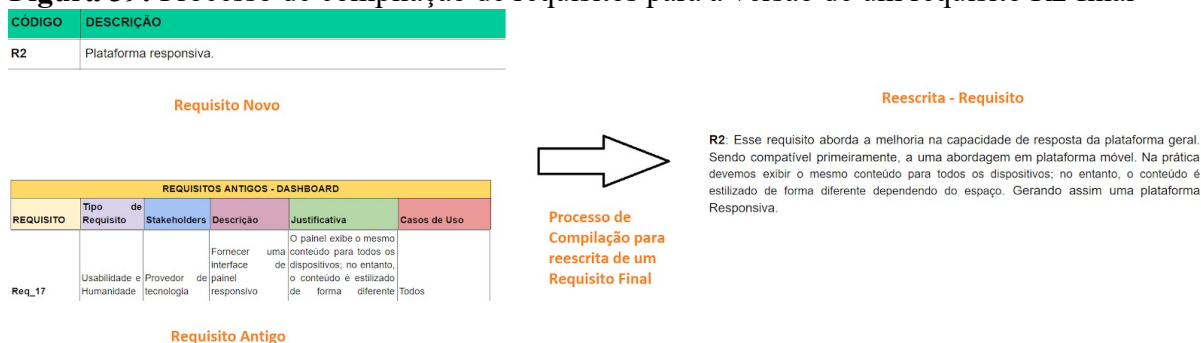
Proposta de Reescrita como junção dos requisitos rastreados:

R2

Esse requisito aborda a melhoria na capacidade de resposta da plataforma geral. Sendo compatível primeiramente, a uma abordagem em plataforma móvel. Na prática devemos exibir o mesmo conteúdo para todos os dispositivos; no entanto, o conteúdo é estilizado de forma diferente dependendo do espaço. Gerando assim uma plataforma Responsiva.

Para uma melhor visualização da reescrita de requisitos está exposta a figura 49 que demonstra a compilação para a reescrita de um requisito final, a partir da visão dos requisitos rastreados pela matriz com elo de ligação.

Figura 39: Processo de compilação de requisitos para a versão de um requisito R2 final



Fonte: Elaborada pelo autor (2019).

4.3 Considerações finais

A abordagem proposta (Figura 5) aplicada ao objeto de estudo OCARIoT, resultou na rastreabilidade de requisitos através de casos de testes criados a partir de uma versão do documento como referência. A rastreabilidade é resultado da análise de requisitos e definição de casos de testes em mapa mental, que geraram uma matriz com elo de ligação sendo os testes. Possibilitando reescrever um novo documento completo e preciso para o projeto com os requisitos que o definem e a aplicação de um processo de qualidade com design, modelagem e execução de testes para garantia da qualidade.

5 CONCLUSÃO

Diante do cenário atual o uso de metodologias ágeis em processos de desenvolvimento de software tem se expandido no meio, mais que as metodologias tradicionais, com seu estilo colaborativo, agregador de valor entre as partes de interesse e equipe de desenvolvimento e rápido na resposta às mudanças. Apesar de facilitadora, essa metodologia também precisa de um acompanhamento íntegro durante o ciclo de vida do software, e este envolvendo as fases principais: especificação, implementação, validação e evolução. Se a fase de especificação não for bem trabalhada, os demais resultados serão afetados por ela, portanto, a especificação reflete as regras de negócio e funcionalidades do sistema, e precisam ser definidas com exatidão, alinhadas às necessidades do cliente.

Contudo, é sabido que em projetos ágeis existe interação entre as fases de um ciclo de desenvolvimento e quando somado a mudanças de requisitos, solicitações do cliente, novas regras, o processo de acompanhamento do software deve ser adaptável para retornar às especificações elicitadas e em uma atividade de gerência manter os seus requisitos rastreáveis. Se isto não acontece, facilmente o controle do projeto e de suas funcionalidades é postergado. Logo, requisitos sem rastreamento e desatualizados, resulta em software incabível de validação.

Com a abordagem proposta é possível rastrear os requisitos de versões de artefatos do software, que são importantes e necessários para o sistema, através de casos de testes criados para uma versão como referência. Assim, o rastreamento de requisitos é implementado e o processo de validação do software é eficaz.

O estudo aplicado ao Dashboard OCARIoT obtiveram resultados palpáveis e assertivos, por conseguir gerar uma matriz de rastreabilidade que evidenciasse os requisitos essenciais para aplicação sendo o elo de relacionamento os casos de testes, que posteriormente especificados resultaram na execução e report dos seus comportamentos para as partes de interesse. E como resultado final a proposta de reescrita de requisitos, contribuindo para elicitação de apenas um artefato de valor completo e consistente, documento de requisitos. Possibilitando o bom funcionamento do ciclo de vida do software guiado por uma metodologia ágil, rastreável e com validações.

A abordagem utilizada possibilita, em oportunidades futuras, promover a execução contínua do seu fluxo em caso de novas atualizações de requisitos para o OCARIoT, criação de novos testes e garantia de sua cobertura na validação. Além de viabilizar a aplicação de outras técnicas de validação no software, como testes de regressão, que se preocupa em verificar que mudanças ocorridas não tenham afetado o bom funcionamento do software.

Permitindo ainda ser aplicada e utilizada no contexto de qualquer desenvolvimento de software que se assemelha ao descrito nesta pesquisa, contribuindo positivamente para a evolução do software sob a ótica da rastreabilidade de requisitos guiada por testes.

REFERÊNCIAS

BLANCO, Mariana Zanuzzio. *Documentação de teste baseado na Norma IEEE 829 – estudo de caso: “sistema de apoio a tomada de decisão”*. Tecnologias, Infraestrutura e Software. Departamento de Computação – Universidade Federal de São Carlos (UFSCar). ISSN 2316-2872 T.I.S. São Carlos, v. 1, n. 1, p. 91-97, jul. 2012.

BSTQB, Brazilian Software Testing Qualifications Board. *Certified Tester Foundation Level Syllabus – Agile Tester Extended do ISTQB*. Comissão Internacional para qualificação de teste de software. Tradução realizada pela TAG01 – Documentação do BSTQB baseada na Versão 2014 do Certified Tester Foundation Level Syllabus – Agile Tester Extended do ISTQB. Versão 2014br, 2014.

_____, Brazilian Software Testing Qualifications Board. *Certified Tester Foundation Level Syllabus*. Tradução realizada pelo WG-Traduções do BSTQB do syllabus do ISTQB: Certified Tester Foundation Syllabus. Versão 2018br, 2018.

COCKBURN, Alistair; HIGHSMITH, Jim. *Agile Software Development: The People Factor*. Computer, 34(11), 131–133. doi:10.1109/2.963450. November, 2001. Disponível em: <http://sci-hub.tw/10.1109/2.963450>. Acesso em: 15 de Ago. de 2019.

DELAMARO, M. E.; MALDONADO, J.C.; JINO, M. *Introdução ao Teste de Software*. Campus, 1. ed. Rio de Janeiro, RJ, Brasil: Elsevier Editora Ltda, 2007.

_____, Márcio Eduardo; MALDONADO, José Carlos, Mario Jino. *Introdução ao Teste de Software*. editora Elsevier, 2016. 2ª Edição.

DIJKSTRA, Edsger W. *Structured programming*. Capítulo I: Notes on structured programming, p.1–82. Academic Press Ltd., London, UK, 1972. Disponível em: <http://dl.acm.org/citation.cfm?id=1243380.1243381>. Acesso em: 14 de Ago. de 2019.

DEVMEDIA. *Engenharia de Software - Introdução à Engenharia de Requisitos*. 2008. Disponível em: <https://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-engenharia-de-requisitos/8034>. Acesso em: 15 de Ago. de 2019.

ESPINDOLA, Rodrigo Santos de; MAJDENBAUM, Ariel; AUDY, Jorge Luiz Nicolas. *Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software*. Programa de Pós-Graduação em Ciência da Computação. Pontifícia Universidade Católica do Rio Grande do Sul. 2014. Disponível em: http://www.metodoiron.com.br/iron/wp-content/uploads/2014/03/Artigo_EngReq_ManutencaoSoftware.pdf. Acesso em: 13 de Ago. de 2019.

GENVIGIR, Elias Canhadas. *Um Modelo para Rastreabilidade de Requisitos de Software Baseado em Generalização de Elos e Atributos*. Tese de Doutorado do Curso de Pós-graduação em Computação Aplicada, orientada pelo Dr. Nandamudi Lankalapalli Vijaykumar, aprovada em 31 de março de 2009. Instituto Nacional de Pesquisas Espaciais – INPE. São José dos Campos, 2009. Disponível em: <http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2009/03.02.14.17/doc/publicacao.pdf>. Acesso em: 06 de Ago. de 2019.

GONÇALVES, Marcus Vinicius Guedes. *O Uso da Metodologia Ágil como Processo de Desenvolvimento Participativo de Sistemas*. Dissertação (mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Ciências Humanas, letras e Artes. Programa de Pós Graduação em Design, Natal, 2017. Disponível em: https://repositorio.ufrn.br/jspui/bitstream/123456789/25022/1/UsometodologiaAgil_Goncalves_2017.pdf. Acesso em: 02 de Ago. de 2019.

IEEE Standard Glossary of Software Engineering Terminology, Standard 610.12. IEEE Press, 1990.

LEAL, Igor Gonçalves. *Requisitos de Metodologias de Teste de software para processos ágeis*. Departamento de Ciência da Computação – Universidade Federal de Minas Gerais Belo Horizonte – MG, Brasil, 2009. Disponível em: <https://homepages.dcc.ufmg.br/~rodolfo/dcc823-1-09/Entrega2Pos/igor2.pdf>. Acesso em 03 de Ago. de 2019.

LUCASSEN, G.; DALPIAZ, F.; VAN DER WERF, J. M. E. M.; BRINKKEMPER, S. Behavior – Driven Requirements Traceability via Automated Acceptance Tests. 2017.

MENEZES, P. M.; CORREIA, T.A.; COUTO, T.; SANTANA, L. *A Engenharia de Requisitos: Um Caso de Implementação de um Sistema para Engenharia de Requisitos*. Revista Interfaces Científicas – Exatas e Tecnologia, Aracaju, V.1, N.3, p. 43 – 54, Out. 2015.

NETO, Arilo Claudio Dias. *Engenharia de Software Magazine - Introdução a Teste de Software*. All content following this page was uploaded by Arilo Claudio Dias Neto on 14 May 2015. Disponível em: https://edisciplinas.usp.br/pluginfile.php/3503764/mod_resource/content/3/Introducao_a_Test_e_de_Software.pdf. Acesso em: 02 de Ago. de 2019.

NETO. José Gameleira do Rêgo. *Usando Técnicas de Mineração de Repositórios Software para Apoiar a Automação de Testes de Software*. Monografia de Graduação. Universidade Federal do Rio Grande do Norte – UFRN. Departamento de Informática e Matemática Aplicada – DIMAp. Natal-RN, Junho, 2019. Disponível em:

https://monografias.ufrn.br/jspui/bitstream/123456789/8949/6/UsandoTecnicasMineracaoTestesSoftware_RegoNeto_2019.pdf. Acesso em 02 de Ago. de 2019.

OCARIOT. *Solução inteligente para cuidar da obesidade infantil usando o potencial da IoT*. 2017. Disponível em: <http://www.ocariot.com.br/>. Acesso em: 05 de Ago. de 2019.

OTADUY, I.; DIAZ, O. *User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps*. ONEKIN Research Group, University of the Basque Country (UPV/EHU), P. Manuel Lardizabal 1, 20018 San Sebastián, Spain. *The Journal of Systems and Software* 133. 212–229. 2017.

PRESSMAN, Roger S.; MAXIM, Bruce R. *Engenharia de Software, Uma Abordagem Profissional*. AMGH Editora Ltda. 8ª . ed. São Paulo, 2016.

RAMESH, Balasubramaniam; JARKE, Matthias. *Toward Reference Models for Requirements Traceability*. *IEEE Transactions on Software Engineering*, Vol. 27, No. 1, January 2001. Disponível em: <http://sci-hub.tw/https://ieeexplore.ieee.org/abstract/document/895989>. Acesso em 14 de Ago. de 2019.

SAYÃO, Miriam; LEITE, Julio Cesar Sampaio do Prado. *Rastreabilidade de Requisitos*. RITA, Volume XII, Número 1, 2005. Disponível em: <http://www-di.inf.puc-rio.br/~julio/rastreabilidade5.pdf>. Acesso em 10 de Ago. de 2019.

SOARES, Michel dos Santos. *Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software*. **INFOCOMP Journal of Computer Science**, [S.l.], v. 3, n. 2, p. 8-13, nov. 2004. Disponível em: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68>. Acesso em: 14 de Ago. de 2019.

SOMMERVILLE, Ian. *Engenharia de Software*. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves ; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo : Pearson Prentice Hall, 2011. Disponível em: https://www.di.ubi.pt/~sebastiao/Ensino/UBI/2017-2018/ES/ApoioEstudo/Engenharia_Software_3Edicao.pdf. Acesso em: 14 de Ago. de 2019.

TESTLINK. *TestLink Open Source Test Management*. 2018. Disponível em: <http://testlink.org/>. Acesso em: 09 de Ago. de 2019.

TESTLINK. *TestLink 1.9.20 Raijin - Read me. Abr*, 2019. Disponível em: https://github.com/TestLinkOpenSourceTRMS/testlink-code/blob/testlink_1_9/README.md. Acesso em: 09 Ago. 2019.

TOMELIN Marcio. *Testes de Software a partir da Ferramenta Visual Test*. Blumenau, Junho. 2001.

TURINE, Marcelo Augusto Santos; MASIERO, Paulo Cesar. *Especificação de Requisitos: Uma Introdução*. Instituto de Ciências Matemáticas de São Carlos Universidade de São Paulo São Carlos, Março. 1996. Disponível em:
http://www2.unemat.br/rhycardo/download/engenharia_de_requisitos.pdf. Acesso em: 03 de Ago. de 2019.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. *Engenharia de Requisitos: software orientado ao negócio*. Rio de Janeiro: Brasport, 2016. Disponível em:
https://books.google.com.br/books?hl=pt-BR&lr=&id=gA7kDAAAQBAJ&oi=fnd&pg=PA74&dq=Ger%C3%Aancia+de+requisitos&ots=sNcq4MW3oU&sig=29nicdM0HeOfDmSrNjOMBUZUb0c&redir_esc=y#v=onepage&q=Ger%C3%Aancia%20de%20requisitos&f=false. Acesso em 04 de Ago. de 2019.

APÊNDICE A – REQUISITOS ANTIGOS DO DASHBOARD

Apêndice destinado a exposição dos requisitos antigos produzidos pelo projeto OCARIoT para o desenvolvimento do sistema.

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_01	Funcional	Profissional de Saúde	Rastreamento de atividade física básica: Medição de continuidade, deslocamento e intensidade	Para atividade física o sistema deve ser capaz de estimar pelo menos a continuidade, o deslocamento e a intensidade do que as crianças estão fazendo durante o dia.	Monitoramento de Atividade Física
Req_02	Funcional	Profissional de Saúde	Rastreamento básico de alimentos: entender as categorias de alimentos, a maneira de cozinhar e a estimativa de quantidade em cada relatório de alimentos	O Sistema não deve ser um contador de calorias, deve ser capaz de classificar as crianças em perfis nutricionais agregados	Rastreamento de Alimentos
Req_03	Funcional	Provedor de Tecnologia	Para criar um painel para educadores e famílias	Necessário para gestão do piloto	Todos
Req_06	Funcional	Crianças, família, educador e profissional de Saúde	Recurso de Login	Permite ao usuário acessar as funcionalidades customizadas de acordo com o seu papel	Todos
Req_07	Funcional	Educador, família	Recursos do painel OCARIoT personalizados	Sistema personalizado para melhor envolvimento:	Todos

			de acordo com o perfil ou função do usuário	interface do usuário e recursos	
Req_10	Funcional	Crianças, família, educador e profissional de Saúde	Recuperação de Senha	É comum os usuários perderem o controle ou o gerenciamento de suas contas. Portanto, para permitir o uso contínuo, é necessário que haja um recurso de recuperação de senha	Todos
Req_11	Funcional	Família, educador e profissional de saúde	Informação estatística do participante BI	Visualização geral dos participantes e sua interação com o aplicativo, o que permite que a equipe de saúde e a equipe técnica realizem estudos dos parâmetros coletados.	Todos
Req_12	Funcional	Criança, família e provedor técnico	Sincronização de dispositivos	Processo de sincronização do aplicativo com o wearable (e / ou outros dispositivos)	Todos
Req_14	Funcional	Provedor de Tecnologia	Dados vestíveis devem ser processados por gateways piloto	Os dados provenientes de diferentes wearables serão processados nos gateways para serem compatíveis com o modelo de dados OCARIoT.	Monitoramento de Atividade Física

Req_15	Usabilidade e Humanidade	Provedor de tecnologia	Gestão de consentimento	Para permitir ou revogar o consentimento total / parcial dos dados pessoais	Todos
Req_17	Usabilidade e Humanidade	Provedor de tecnologia	Fornecer uma interface de painel responsivo	O painel exibe o mesmo conteúdo para todos os dispositivos; no entanto, o conteúdo é estilizado de forma diferente dependendo do espaço disponível	Todos
Req_18	Usabilidade e Humanidade	Provedor de tecnologia	Para fornecer diferentes tipos de visualização para a família e educador	Visualizações de UI diferentes por função	Todos
Req_19	Usabilidade e Humanidade	Provedor de tecnologia	Sincronização de dados transparente entre wearable e a plataforma	Espera-se que a sincronização de dados seja transparente e não exija intervenção humana, se possível	Monitoramento de atividade física
Req_22	Desempenho	Profissional de saúde	Reforço da atividade física e rastreamento de alimentos com validação cruzada de dados entre diferentes usuários (crianças / famílias, crianças / educadores)	O processo de validação cruzada melhora a qualidade dos dados coletados e os resultados no final do projeto	Todos
Req_23	Desempenho	Provedor de tecnologia	Integração com sensores ambientais (por exemplo, temperatura,	Melhore a integração da IoT	Todos

			umidade, etc.)		
Req_24	Desempenho	Crianças e família	Introdução de comida o mais fácil possível	Desde que eles têm que fazer isso diário é importante para simplificar essa ação	Rastreamento de alimentos
Req_25	Operacional e Ambiental	Provedor de tecnologia	Visualização de dados estatísticos por usuário	Fornecer ferramentas de visualização de dados para os diferentes usuários	Todos
Req_26	Operacional e Ambiental	Profissional de saúde e provedor técnico	ID exclusivo por criança, incluindo um prefixo piloto (SP, GR, BR)	Para garantir a privacidade em dados pessoais sensatos	Todos
Req_27	Operacional e Ambiental	Profissional de saúde e provedor técnico	Associar o ID da criança ao ID vestível	para conceder privacidade sensível a pessoa	Todos
Req_28	Manutenção e Suporte	Provedor técnico	Introduzir usuários iniciantes no aplicativo e no painel	Apresentar as informações de embarque no início do aplicativo / painel	Todos
Req_29	Segurança	Provedor de tecnologia	Segurança e privacidade por design	Para garantir a privacidade em dados pessoais sensatos	Todos
Req_30	Segurança	Provedor de tecnologia	Anonimizado e Pseudo Anonimizado	Permitir a troca de dados pessoais sensíveis para o estudo de projetos	Todos
Req_31	Segurança	Provedor de tecnologia	Acesso Restrito ao recurso System (RAS)	Impede usuários não autorizados de assinar	Todos

APÊNDICE B – REQUISITOS NOVOS DO DASHBOARD

Apêndice destinado a exposição dos requisitos atuais (novos) produzidos pelo projeto OCARIoT para continuidade do desenvolvimento do sistema.

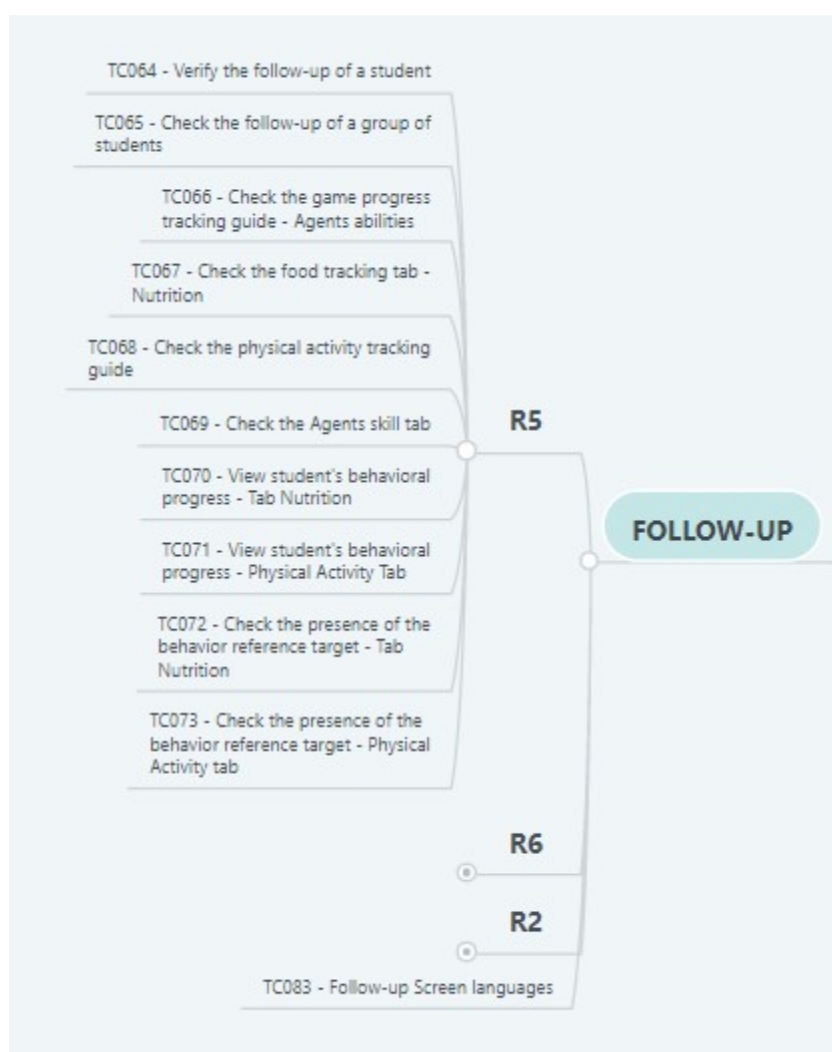
CÓDIGO	DESCRIÇÃO
R1	Refine o Login: 1) crie uma landing page e 2) inclua um link para o site OCARIoT oficial para visualizar o material multimídia (vídeo, áudio, entrevistas, histórias de sucesso) gerado pelos sites piloto.
R2	Plataforma responsiva.
R3	Estender o formulário nas exibições de professores e pais para incluir um novo elemento (rótulo de texto), caso eles possam adicionar um apelido de filho / aluno, embora a plataforma não deva armazenar dados que identifiquem o aluno
R4	Melhore as telas que contêm guias, incluindo paginação, estado de progresso, etc. Os formulários de entrada devem poder salvar os dados coletados, embora o questionário não esteja totalmente preenchido e dê a possibilidade de concluí-lo posteriormente
R5	Aprimore as visualizações de estatísticas mostrando o status atual em relação à média da população e / ou hábitos de referência
R6	Adicione recursos de pesquisa e filtros para facilitar que educadores e profissionais de saúde possam gerenciar de maneira mais fácil as estatísticas, a coleta de dados e o gerenciamento de usuários.

R7	Adicionar menu de configurações para personalizar: notificações (ativar / desativar, frequência para cada desafio), componentes do avatar, como cor da pele, sexo etc.
R8	Por padrão, cada aluno terá um ID único que não pode ser vinculado à identidade pessoal em nenhum caso e deve ser armazenado apenas no servidor das escolas.
R9	Crie um painel de gerenciamento de missões (MMD) para vincular dados provenientes do DSS e do aplicativo OCARIoT.
R10	Inclua mais detalhes no menu de acompanhamento de alimentos: 1) questionários; 2) criação de cardápio para educadores e famílias; 3) entrada proveniente do aplicativo OCARIoT (estudantes).

APÊNDICE C – CASOS DE TESTES DESENHADOS NO MAPA MENTAL PARA AS FUNCIONALIDADES DO OCARI6T

Este apêndice é parte do conjunto de resultados produzidos para a aplicação da abordagem apresentada neste trabalho. Contém o desenho de casos de testes para todas as outras funcionalidades do sistema.

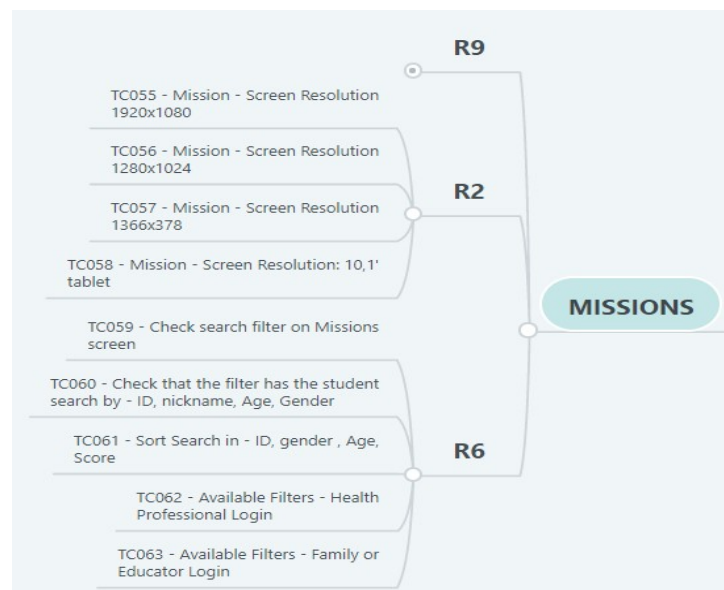
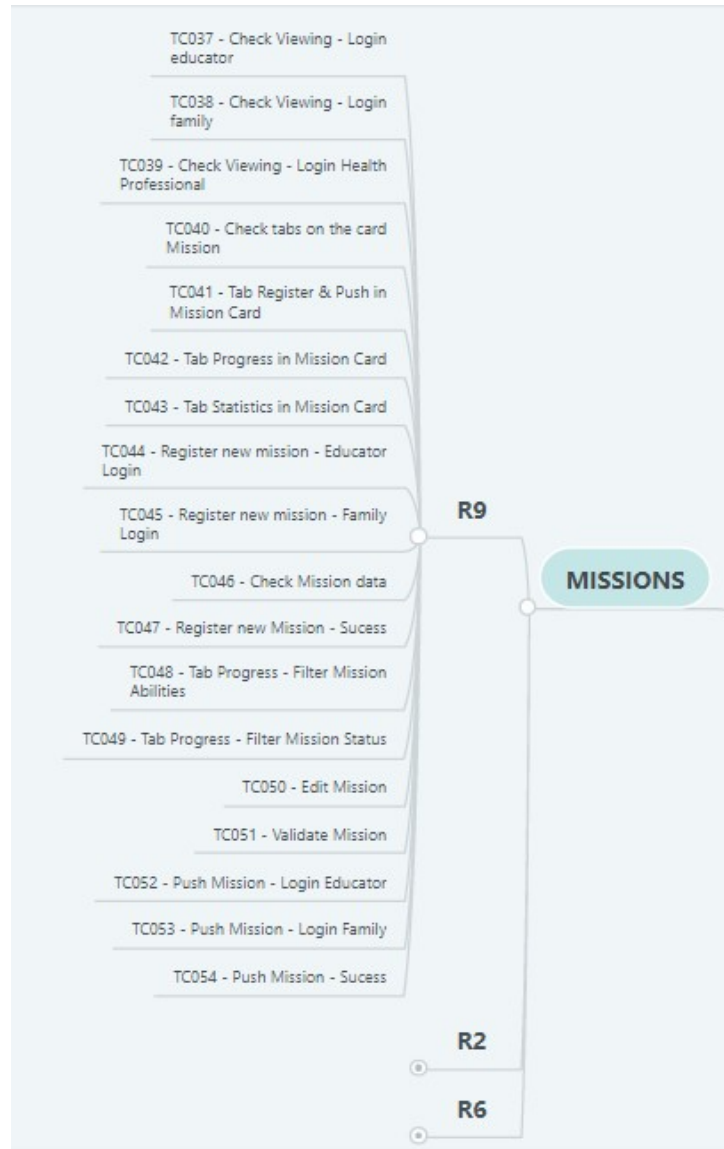
- **FOLLOW-UP** - Tela de acompanhamento do progresso da criança no jogo, na sua alimentação e nas atividades físicas.





Após a definição do projeto (nó pai) segue a folha dependente (Follow-Up), os filhos são os requisitos (R5, R6, R2 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. E o caso de teste solto, completa a cobertura como requisito essencial do negócio do projeto.

- MISSIONS - lista, visualiza detalhes, cria missões para as crianças e que são enviadas pelo agente secreto. Além de acompanhar o progresso das crianças na resposta às missões e suas estatísticas.



Após a definição do projeto (nó pai) segue a folha dependente (Missions), os filhos são os requisitos (R9, R2, R6 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento.

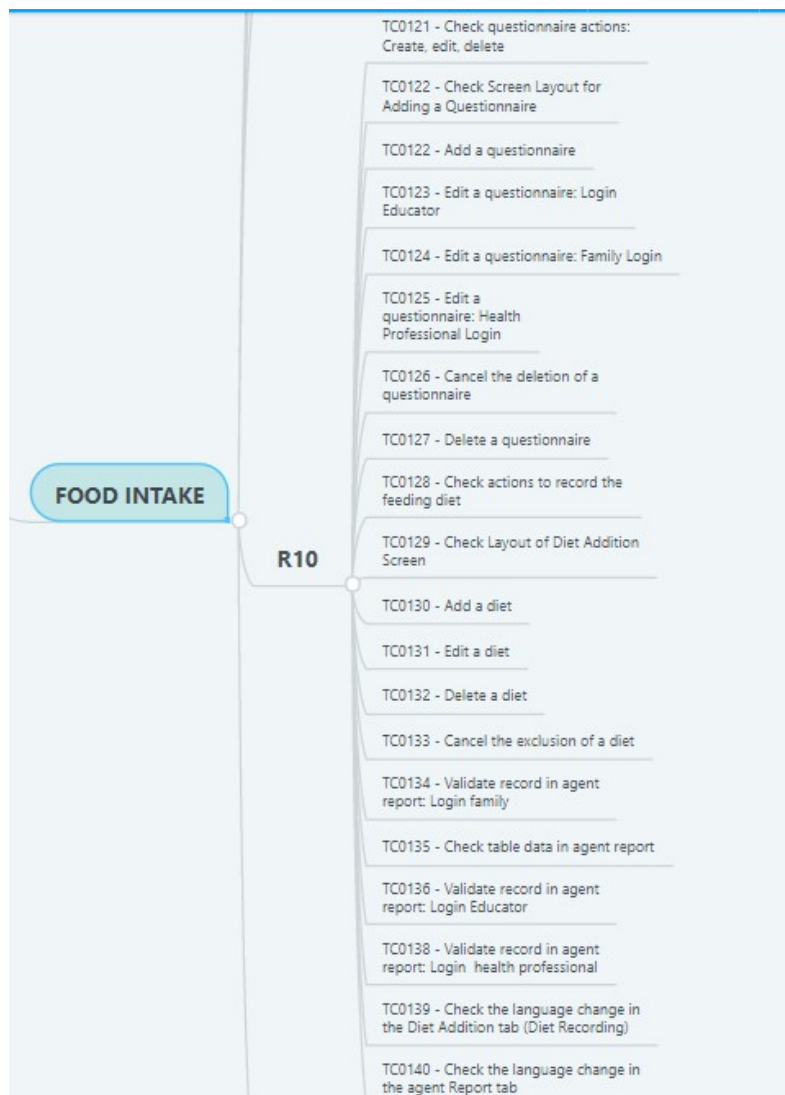
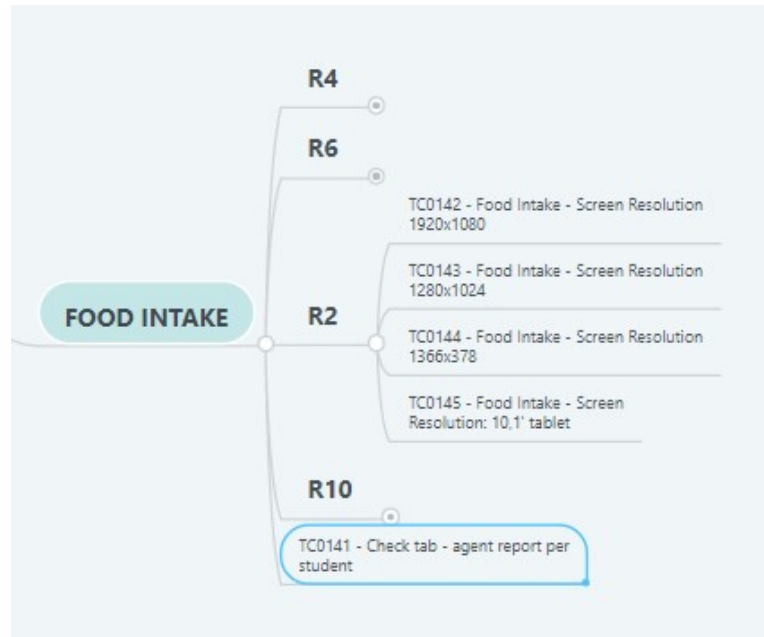
- Iot DEVICES - Para o acompanhamento de dados ambientais, através de outros dispositivos sincronizados.



Após a definição do projeto (nó pai) segue a folha dependente (Iot Devices), os filhos são os requisitos (Req_23 - requisito antigo, R2, R6 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. E o caso de teste solto, completa a cobertura como requisito essencial do negócio do projeto.

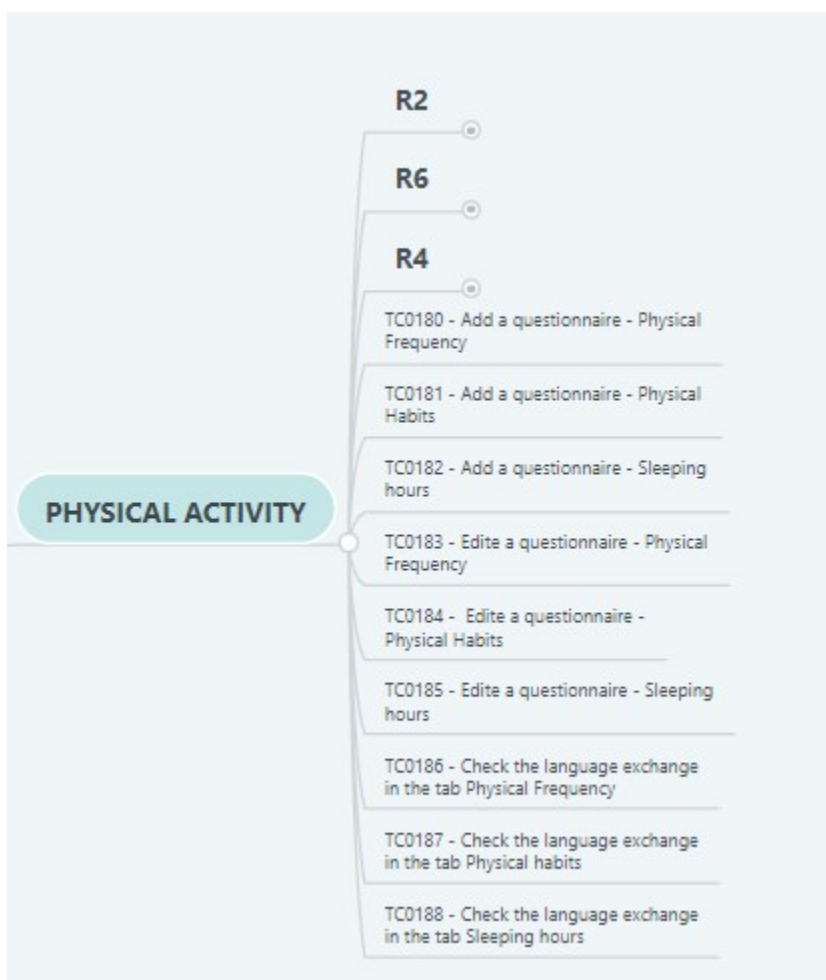
- Food Intake - Recolher e acompanhar os hábitos de nutrição infantil.





Após a definição do projeto (nó pai) segue a folha dependente (Food Intake), os filhos são os requisitos (R4, R6, R2, R10 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. E o caso de teste solto, completa a cobertura como requisito essencial do negócio do projeto.

- **PHYSICAL ACTIVITY** - Para recolher e acompanhar atividades infantis, exercícios e qualidade do sono.





Após a definição do projeto (nó pai) segue a folha dependente (Physical Activity), os filhos são os requisitos (R2,R4,R6 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. E os casos de testes soltos, completam a cobertura como requisito essencial do negócio do projeto.

- PROFILE - Dados das crianças estendidos para o estudo.





Após a definição do projeto (nó pai) segue a folha dependente (Profile), os filhos são os requisitos (R2, R6 - Requisitos Novos), e os filhos destes são os casos de testes que visam garantir seu funcionamento. E os casos de testes soltos, completam a cobertura como requisito essencial do negócio do projeto.

APÊNDICE D – PROCESSO DE REESCRITA DE REQUISITOS PARA AS FUNCIONALIDADES DO OCARIoT

O processo de reescrita de requisitos foi desenvolvido pelo autor, com o objetivo de alcançar a cobertura de todas as funcionalidades do sistema OCARIoT-Dashboard, nosso objeto de estudo, a partir do documento de requisitos mais atual e da abordagem proposta nesse trabalho. A reescrita utiliza como base de dados os requisitos atuais como citado, e o resultado da matriz de rastreabilidade para fazer o elo a outros requisitos já existentes para o sistema. Além da comparação com o protótipo e a tela funcional implementada, resultando em uma reformulação do documento de requisitos, os quais são essências para o desenvolvimento do projeto. A seguir estão as demais reescritas construídas para alcançar as funcionalidades do sistema:

Reescrita R3

Base de Dados: Somatório do escopo do R3 (requisito novo) mais o Req_07, Req_15, Req_26 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_07 + R3 = TC0165...TC0166

Req_15 + R3 = TC0164...TC0171

Req_26 + R3 = TC0170...TC0171

Base de dados: Protótipo referente ao R3 (requisito novo), não possuindo ainda implementação no sistema.

The screenshot shows a web browser window with the OCARIoT D logo in the top right corner. The browser's address bar contains a search icon. The main content area is titled 'Student general info > General information'. On the left, there is a sidebar menu with options: Home, Activity tracking, Food tracking, Education and motivation, and Student general info (which is highlighted). The main form contains the following fields:

- ID: 674
- Name/Nickname: Carlos Daniel
- Gender: Radio buttons for Female and Male
- Date of Birth: 12/04/2008 (with a calendar icon)
- School: Select the School (dropdown menu)
- Class: Select the Class (dropdown menu)

A green 'Save' button is located at the bottom of the form.

Proposta de Reescrita como junção dos requisitos rastreados:

R3

A plataforma não deve armazenar dados que identifiquem o aluno, De acordo com o regulamento GDPR, dados sensíveis como nome, sexo, data de nascimento, não serão armazenados nos pilotos europeus. Mas para os brasileiros, deve-se estender o formulário nas exibições de professores e pais para incluir um novo elemento (rótulo de texto), caso eles possam adicionar informações extras ao perfil do usuário, como um apelido de filho / aluno. A fim de identificá-los na plataforma, melhorando a usabilidade e legibilidade de seus pontos de vista.

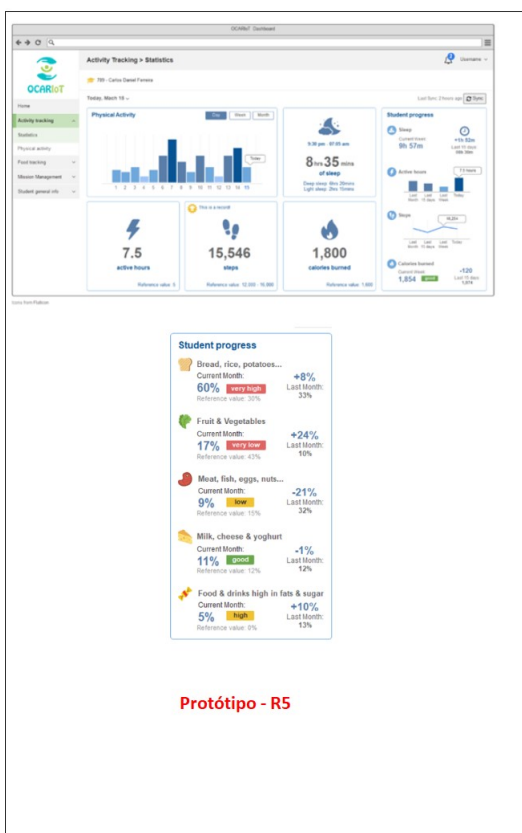
Reescrita R4

Base de Dados: Somatório do escopo do R4 (requisito novo) mais o Req_01, Req_02, Req_22 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

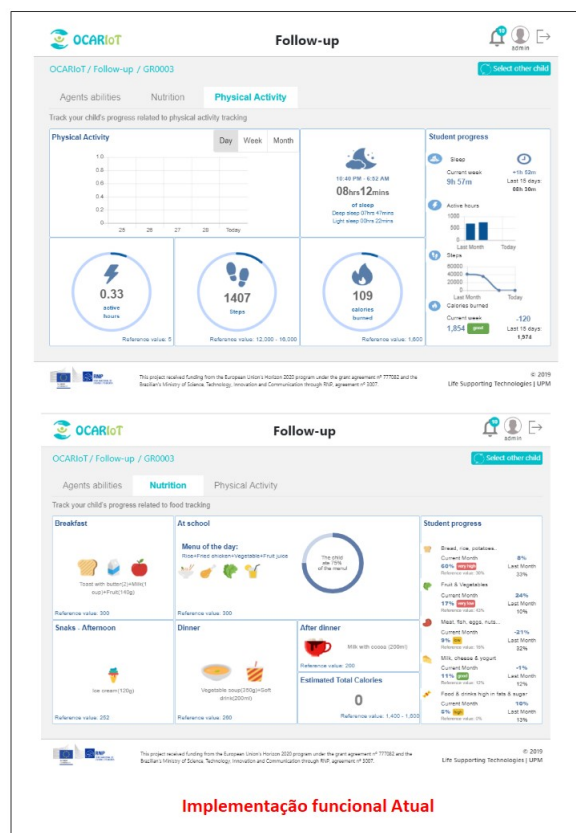
Casos de Testes mapeados na Matriz

Req_01 + R4 = TC0172...TC0179
 Req_02 + R4 = TC0113...TC0119
 Req_22 + R4 = TC0113...TC0120, TC0172...TC0179

Base de dados: Protótipo referente ao R4 (requisito novo) versus Tela funcional implementada no OCARIoT.



X



Proposta de Reescrita como junção dos requisitos rastreados:

R4

O painel também é uma ferramenta para coleta de dados baseada em questionários padronizados sobre hábitos de atividade física, hábitos alimentares e frequência alimentar usados em estudos clínicos. Os questionários comentados são grandes e complexos, para melhorar essa percepção é ideal estabelecer um padrão para dividi-los em uma estrutura mais hierárquica, desta forma deve-se:

1. Incluir paginação em todas as guias

2. Inserir também componentes que mostram o progresso do usuário ao responder o questionário e o quão próximo estão de sua conclusão
3. E somando a isto a funcionalidade de salvar e retornar posteriormente de onde o usuário parou.

Essa estrutura deve ser aplicada e validada a toda e qualquer funcionalidade que tenha a inclusão de questionários. Descritas a seguir:

- **Rastreamento de Atividade física básica:** Medição de continuidade, deslocamento e intensidade das atividades coletadas das crianças. Pelo acompanhamento de frequência de atividade física, hábitos e horas de sono. Esse escopo será mensurado pela resposta de questionários, onde:
 1. É possível cadastrar(responder) um novo questionário
 2. Editar um questionário anteriormente respondido
 3. É possível visualizar os questionários respondidos para uma criança específica.
- **Rastreamento básico de alimentos: Tem o objetivo de coletar hábitos de nutrição das crianças,** entender as categorias de alimentos que consomem e adicionar gravação de alimentação, a estimativa de quantidades, a frequência da alimentação, hábitos e o seu comportamento. Esse escopo será mensurado pela resposta de questionários, onde:
 1. É possível mensurar (gravar) uma nova alimentação que pode ser na categoria: Café da manhã, lanche da manhã / tarde, almoço e jantar
 2. Adicionar o tipo da comida e selecionar a quantidade a ser ingerida.
 3. Editar a alimentação gravada
 4. Responder a questionários nas funcionalidades: Frequência, hábitos e Comportamento.
 5. Editar questionários
 6. Visualizar questionários para uma criança específica.
- Reforço da atividade física e rastreamento de alimentos com validação cruzada de dados entre diferentes usuários. (Criança/família, criança/educador).

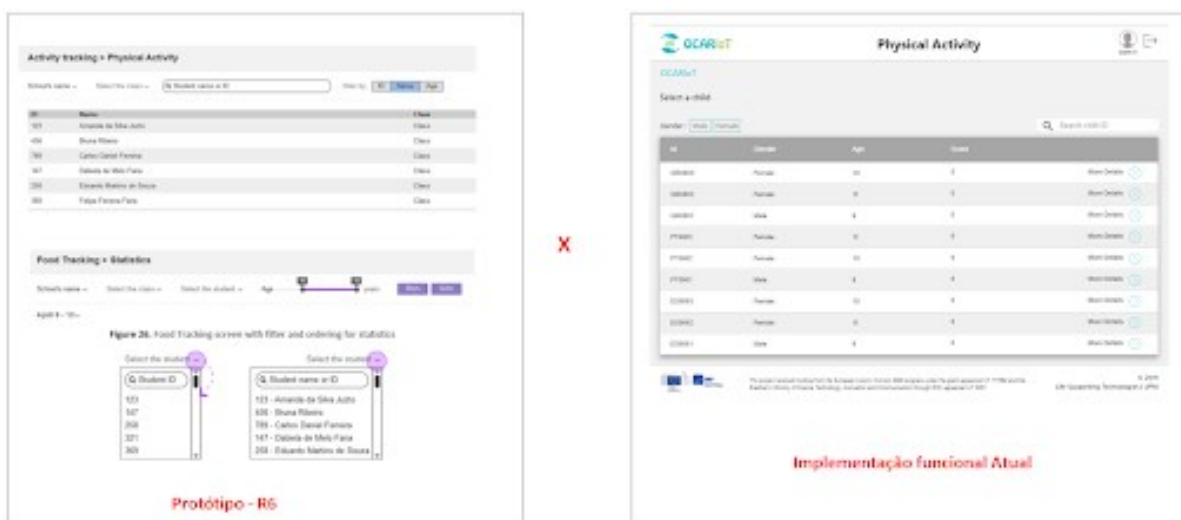
Reescrita R5

Base de Dados: Somatório do escopo do R5 (requisito novo) mais o Req_11, Req_12, Req_25 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_11 + R5 = TC064 ... TC073
 Req_12 + R5 = TC069 ... TC073
 Req_25 + R5 = TC064 ... TC073

Base de dados: Protótipo referente ao R5 (requisito novo) versus Tela funcional implementada no OCARIoT.



Proposta de Reescrita como junção dos requisitos rastreados:

R5

Em um processo de sincronização, as estatísticas irão juntar dados provenientes de wearable, questionários e missões (ver R9) em várias visões que, através de gráficos analíticos, fornecerão informações sobre o comportamento de um estudante ou uma agregação de

estudantes. Sendo assim, se torna a visualização geral dos participantes e sua interação com o aplicativo.

Estas estatísticas devem poder mostrar:

- Progresso comportamental do aluno, isso significa quanta atividade física ele está praticando ou a comida está comendo, etc.
- Alvo de referência do comportamento, isso significa que um gráfico específico deve representar também o destino esperado para esse comportamento específico. Por exemplo, quanto de fruta um estudante tem que comer ou quanta atividade física ele tem que praticar. Esse tipo de dado virá do Painel de Gerenciamento de Missões.
- Recomendações para o alvo de referência do comportamento, isso significa que, a partir das estatísticas, o usuário tem que entender a ação que o aluno deve realizar para alcançar o alvo de referência.
- Na implementação do dashboard, os valores de referência relacionados ao consumo de alimentos serão encaminhados para a frequência de consumo por semana ou dia; enquanto a porcentagem se referirá à ingestão de calorias.

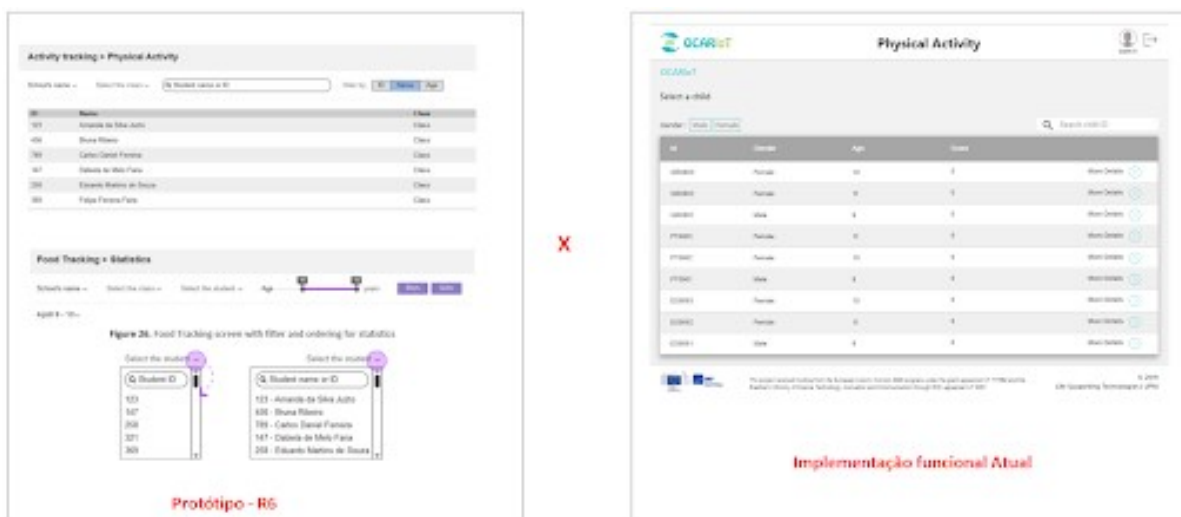
Reescrita R6

Base de Dados: Somatório do escopo do R6 (requisito novo) mais o Req_07 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_07 + R6 = TC0162

Base de dados: Protótipo referente ao R6 (requisito novo) versus Tela funcional implementada no OCARIoT.



Proposta de Reescrita como junção dos requisitos rastreados:

R6

Para melhorar a usabilidade do painel no novo design, é previsto adicionar um formulário de pesquisa para os usuários que associaram vários alunos. Eles devem ser capazes de procurar um aluno por id, gênero, idade, pontuação, etc... O formulário de busca irá filtrar em tempo real a lista completa dos alunos, a fim de fornecer ao usuário apenas os alunos que correspondem aos critérios de pesquisa.

Esta funcionalidade do sistema deve se comportar de forma personalizada, dependendo do perfil do usuário:

- Em Food Intake - Filtrar por ID para profissionais de saúde e por nome / ID para educadores e familiares.

Reescrita R7

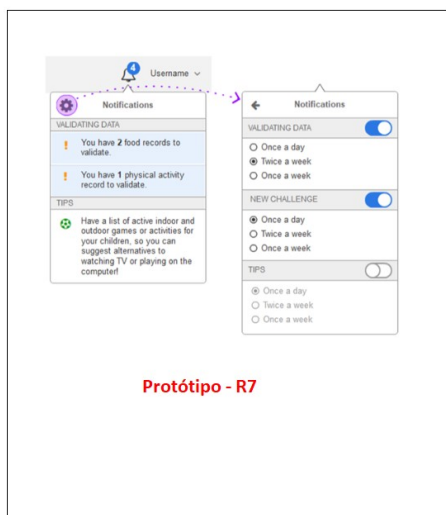
Base de Dados: Replicação da especificação para o R7.

Não houve rastreabilidade com requisitos antigos, pois o mesmo foi adicionado a versão atual de regras de negócio do sistema.

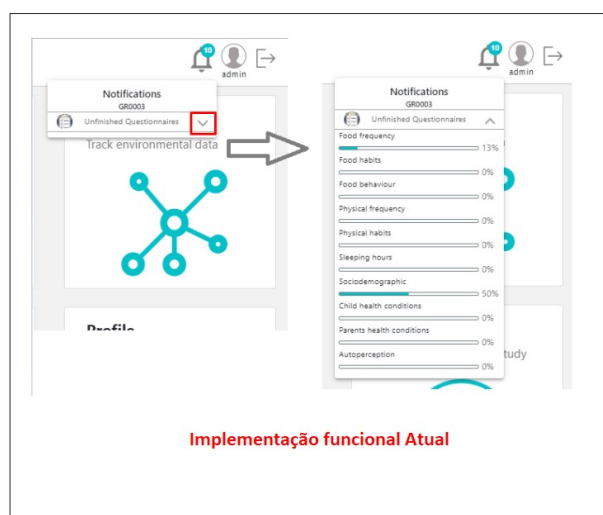
Casos de Testes Gerais:

TC027...TC036

Base de dados: Protótipo referente ao R7 (requisito novo) versus Tela funcional implementada no OCARIoT.



X



Proposta de Reescrita do Requisito:

R7

Na nova versão do painel, será prevista uma interface de usuário para personalizar a aparência do sistema e alguns parâmetros, como a frequência das notificações. Eles devem receber notificações para as funcionalidades do sistema, que precisam ser refinadas e alinhadas ao Cliente, necessitando de uma melhor construção.

Reescrita R8

O requisito 8 não será reescrito, pois o mesmo é definido da mesma forma do R3, simplificando a documentação em apenas uma única descrição.

Reescrita R9

Base de Dados: Somatório do escopo do R9 (requisito novo) mais o Req_07, Req_11, Req_12, Req_18, Req_25 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_07 + R5 = TC037...TC039, TC044...TC045, TC052 ...TC053

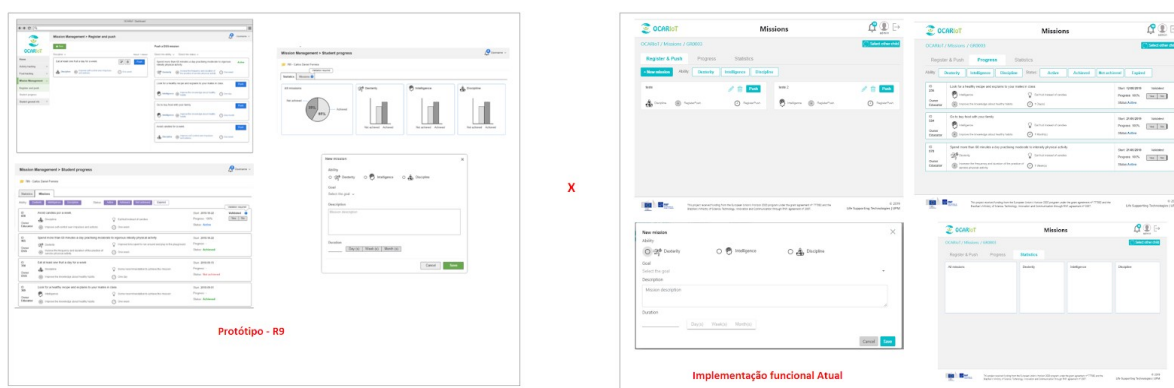
Req_11 + R5 = TC043

Req_12 + R5 = TC041 ...TC042, TC051...TC054

Req_18 + R5 = TC037...TC038, TC044...TC045, TC052...TC053

Req_25 + R5 = TC043

Base de dados: Protótipo referente ao R9 (requisito novo) versus Tela funcional implementada no OCARIoT.



Proposta de Reescrita do Requisito:

R9

Nesta seção será descrita a parte do dashboard chamada Painel de gerenciamento de missão (MMD). Ele descreve uma nova interface do usuário projetada para corresponder aos serviços fornecidos pelo DSS do WP4 que implementa as missões propostas aos alunos no aplicativo OCARIoT, havendo assim sincronização de dados com o dispositivo. As recomendações serão disparadas na plataforma OCARIoT, a fim de ajudar pais e educadores a apoiarem os

alunos a alcançarem o objetivo proposto por uma missão e, ao mesmo tempo, que estarão cientes dos conceitos propostos pelo sistema.

Para esclarecer esses conceitos, usaremos as seguintes definições:

- Missão alvo - o número a ser atingido para considerar a missão como concluída, por exemplo, 15 min / dia.
- Missão - é a atividade específica que a criança deve realizar para completar o comportamento proposto e o desafio, e a quantificação a ser usada serão baseadas na faixa de metas definida para o desafio associado. (Exemplo: gasta pelo menos 15 minutos / dia praticando atividade física de intensidade moderada a vigorosa)
- Recomendação do painel, um plano de ação proposto para pais / educadores para apoiar a criança a alcançar a missão, a ser mostrado através do painel. (Exemplo: incentivar a criança para caminhar juntos depois da escola)
- Perguntas do painel (questionários) - um conjunto de perguntas a serem feitas aos pais / educadores (não mais que 3) e as respostas para a seleção (não mais que 4) quando eles estiverem validando uma missão como concluída no painel. (Exemplo: por exemplo qual atividade física sua criança praticou durante a semana? Futebol, natação, corrida, outros).
- Habilidades - uma lista das habilidades associadas à missão, a primeira é a habilidade principal e o restante são habilidades secundárias (opcional). (Exemplo. Destreza, disciplina).
- As missões, a missão alvo e as recomendações estão agrupadas nas três habilidades do agente: destreza, inteligência e disciplina. Essas habilidades serão as mesmas do aplicativo OCARIoT.

O conceito principal nesta parte do painel é a missão (conceito compartilhado entre painel, DSS e aplicativo. Para cada aluno será possível acessar uma visão que mostrará:

- O id da missão
- O identificador do criador da missão, educador ou DSS
- A habilidade associada à missão
- A descrição da missão
- A duração da missão é o tempo esperado da missão
- Hora de início, o momento em que a missão começa
- O alvo da missão

- O estado (alcançado ou não alcançado) da missão e, eventualmente, o % do progresso se aplicável.
- As recomendações associadas ao alvo da missão
- O estado de validação da missão, caso precise ser revisado pelos pais e / ou educadores.

A especificação segue-se com o recursos personalizados para funcionalidade dependendo do perfil ou função do usuário, segue-se:

- Os educadores também devem ser capazes de criar e enviar novas missões ao aplicativo OCARIoT.
- O MMD permitirá, apenas educadores, a criar missões. Porém apenas criam missões para o piloto (escola) no qual fazem parte.
- Uma missão criada envolve todo o piloto da escola.
- O MMD permitirá atualizar e validar missões
- A maioria das missões será validada de forma transparente pelo DSS, algumas missões específicas devem poder ser validadas pelos pais e educadores, para uma criança ou um grupo de crianças.
- Deverá ter um conjunto de estatísticas disponível no MMD, ou seja, haverá informações estatísticas dos participantes.

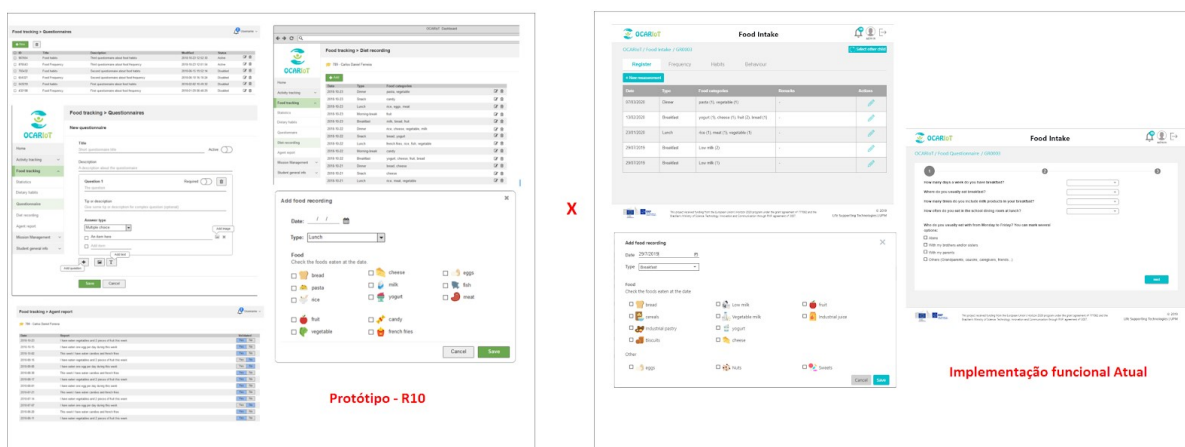
Reescrita R10

Base de Dados: Somatório do escopo do R10 (requisito novo) mais o Req_02, Req_07, Req_11, Req_12, Req_18, Req_22, Req_24, Req_25 ambos cobertos pelos casos de testes referenciados na Matriz respectivamente:

Casos de Testes mapeados na Matriz

Req_02 + R10 = TC0121...TC0133
 Req_07 + R10 = TC0124...TC0125, TC0134, TC0136, TC0138
 Req_11 + R10 = TC0121..TC0133
 Req_12 + R10 = TC0134...TC0138, TC0141
 Req_18 + R10 = TC0123...TC0124, TC0134 , TC0136
 Req_22 + R10 = TC0121...TC0138, TC0141
 Req_24 + R10 = TC0128..TC0133
 Req_25 + R10 = TC0121..TC0133

Base de dados: Protótipo referente ao R10 (requisito novo) versus Tela funcional implementada no OCARIoT.



Proposta de Reescrita do Requisito:

R10

Dentro do painel, o rastreamento de alimentos será gerenciado em 3 níveis diferentes.

Haverá recursos da funcionalidade personalizados de acordo com o perfil ou função do usuário:

- Somente pais e educadores poderão editar o rastreamento da comida das crianças.
- Os profissionais de saúde só poderão revisar os dados.
- **Nível 1: Linha de base (questionários)**

A linha de base é como uma fotografia do estado da criança com base em questionários validados relacionados à frequência alimentar e hábitos alimentares. Eles serão preenchidos periodicamente pelos alunos e pais pelo menos 3 vezes por período de avaliação. Esses questionários devem poder ser:

- Revisados
- Criados,
- excluídos e atualizados.

As estatísticas provenientes desses dados serão frequências de consumo de alimentos e categoria relacionada aos hábitos alimentares das crianças.

- **Nível 2: registro de dieta**

A gravação da dieta será a parte do rastreamento de alimentos relacionado ao registro diário de alimentos. A gravação da dieta permitirá:

- Criar,
- Atualizar,
- Excluir e validar o registro da dieta.
- Um registro alimentar deve ocorrer da maneira mais simplificada possível, este será a representação qualitativa de um café da manhã, Lanche da manhã/ tarde, almoço, ou jantar de uma criança.

Incluirá:

- Data
- Tipo: café da manhã, manhã, almoço, lanche ou jantar de uma criança
- Um conjunto de bandeiras, uma para cada categoria de alimentos, como frutas, vegetais, peixe, carne, peixe frito, etc, que será selecionado pelo usuário. Conjuntamente a quantidade de ingestão deve ser mensurada.
- Dados estatísticos dos participantes provenientes dessa funcionalidade, devem ser comparáveis com a linha de base.

- **Nível 3: relatório do agente**

Nesse nível deve haver sincronização dos dispositivos, estando vinculado o rastreamento de alimentos no painel do OCARIoT ao aplicativo OCARIoT.

- No aplicativo, os alunos precisam preencher um "relatório de agente secreto" que incluirá informações sobre comida.
- Esta parte do painel informa todas as informações coletadas pelo relatório do agente secreto relacionadas à comida em uma exibição de tabela. Os pais têm que interagir com essa tabela e validar as respostas das crianças.

APÊNDICE E – COMPILAÇÃO DA REESCRITA DE REQUISITOS

Este apêndice destina-se a expor o resultado desenvolvido pelo autor, para melhor visualização do processo de reescrita. Onde mostra a compilação dos requisitos que foram identificados seus relacionamentos pela matriz de rastreabilidade e a reformulação destes para uma nova escrita de apenas um único requisito.

• R3

CÓDIGO	DESCRIÇÃO
R3	Estender o formulário nas exibições de professores e pais para incluir um novo elemento (rótulo de texto), caso eles possam adicionar um apelido de filho / aluno, embora a plataforma não deva armazenar dados que identifiquem o aluno
R8	Por padrão, cada aluno terá um ID único que não pode ser vinculado à identidade pessoal em nenhum caso e deve ser armazenado apenas no servidor das escolas.

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_07	Funcional	Educador, família	Recursos do painel OCARIOT personalizados	Sistema personalizado para melhor envolvimento: interface	Todos
Req_15	Usabilidade e Humanidade	Provedor de tecnologia	Gestão de consentimento	Para permitir ou revogar o consentimento total / parcial dos dados pessoais	Todos
Req_26	Operacional e Ambiental	Profissional de saúde e provedor técnico	ID exclusivo por criança, incluindo um prefixo piloto (SP, GR, BR)	Para garantir a privacidade em dados pessoais sensatos	Todos



Processo de compilação para reescrita de um Requisito final

R3

A plataforma não deve armazenar dados que identifiquem o aluno. De acordo com o regulamento GDPR, dados sensíveis como nome, sexo, data de nascimento, não serão armazenados nos pilotos europeus. Mas para os brasileiros, deve-se estender o formulário nas exibições de professores e pais para incluir um novo elemento (rótulo de texto), caso eles possam adicionar informações extras ao perfil do usuário , como um apelido de filho / aluno. A fim de identificá-los na plataforma, melhorando a usabilidade e legibilidade de seus pontos de vista.

• R4

CÓDIGO	DESCRIÇÃO
R4	Melhora as telas que contém guias, incluindo paginação, estado de progresso, etc. Os formulários de entrada devem poder salvar os dados coletados, embora o questionário não esteja totalmente preenchido e dê a possibilidade de concluí-lo posteriormente

Requisito Novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_01	Funcional	Profissional de Saúde	Rastreamento de atividade física básica: Medição de continuidade e a de deslocamento e intensidade do que as crianças estão fazendo durante o dia	Para atividade física o sistema deve ser capaz de estimar pelo menos a continuidade, o deslocamento e a intensidade	Monitoramento de Atividade Física
Req_02	Funcional	Profissional de Saúde	Rastreamento básico de alimentos: entender as categorias de alimentos, a maneira de cozinhar e a um contador de calorias, estimativa de leve ser capaz de classificar as crianças em perfis nutricionais agregados	O Sistema não deve ser capaz de calcular a quantidade de alimentos em cada relatório de alimentos	Rastreamento de Alimentos
Req_22	Desempenho	Profissional de saúde / educadores	Reforço da atividade física e rastreamento de alimentos com validação cruzada de dados entre diferentes usuários (crianças / famílias, crianças / educadores)	O processo de validação cruzada melhora a qualidade dos dados coletados e os resultados no final do projeto	Todos

Requisitos Antigos



Processo de compilação para reescrita de um Requisito final

R4

O painel também é uma ferramenta para coleta de dados baseada em questionários padronizados sobre hábitos de atividade física, hábitos alimentares e frequência alimentar usados em estudos clínicos. Os questionários comumente são grandes e complexos, para melhorar essa percepção é ideal estabelecer um padrão para dividi-los em uma estrutura mais hierárquica, desta forma deve-se:

1. Incluir paginação em todas as guias
2. Inserir também componentes que mostram o progresso do usuário ao responder o questionário e o qual próximo estado de sua conclusão
3. É somando a isto a funcionalidade de salvar e retornar posteriormente de onde o usuário parou.

Essa estrutura deve ser aplicada e validada a toda e qualquer funcionalidade que tenha a inclusão de questionários. Descritas a seguir:

- **Rastreamento de Atividade física básica:** Medição de continuidade, deslocamento e intensidade das atividades coletadas das crianças. Pelo acompanhamento de frequência de atividade física, hábitos e horas de sono. Esse escopo será mensurado pela resposta de questionários, onde:
 1. É possível cadastrar(responder) um novo questionário
 2. Editar o questionário anteriormente respondido
 3. É possível visualizar os questionários respondidos para uma criança específica.

- **Rastreamento básico de alimentos:** Tem o objetivo de coletar hábitos de nutrição das crianças, entender as categorias de alimentos que consomem e adicionar gravação de alimentação, a estimativa de quantidades, a frequência da alimentação, hábitos e o seu comportamento. Esse escopo será mensurado pela resposta de questionários, onde:
 1. É possível mensurar (gravar) uma nova alimentação que pode ser na categoria: Café da manhã, lanche da manhã / tarde, almoço e jantar
 2. Adicionar o tipo da comida e selecionar a quantidade a ser ingerida.
 3. Editar a alimentação gravada
 4. Responder a questionários nas funcionalidades: Frequência, hábitos e Comportamento.
 5. Editar questionários
 6. Visualizar questionários para uma criança específica.

- Reforço da atividade física e rastreamento de alimentos com validação cruzada de dados entre diferentes usuários. (Criança/família, criança/educador).

- R5

CÓDIGO	DESCRIÇÃO
R6	Adicione recursos de pesquisa e filtros para facilitar que educadores e profissionais de saúde possam gerenciar de maneira mais fácil as estatísticas, a coleta de dados e o gerenciamento de usuários.

Requisito novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_07	Funcional	Educador, família	Recursos do painel OCARIoT personalizados de acordo com o perfil ou função do usuário	Sistema personalizado para melhor envolvimento: interface do usuário e recursos	Todos

Requisito Antigo



Processo de compilação para reescrita de um Requisito final

R6

Para melhorar a usabilidade do painel no novo design, é previsto adicionar um formulário de pesquisa para os usuários que associaram vários alunos. Eles devem ser capazes de procurar um aluno por id, gênero, idade, pontuação, etc... O formulário de busca irá filtrar em tempo real a lista completa dos alunos, a fim de fornecer ao usuário apenas os alunos que correspondem aos critérios de pesquisa.

Esta funcionalidade do sistema deve se comportar de forma personalizada, dependendo do perfil do usuário:

- Em Food Intake - Filtrar por ID para profissionais de saúde e por nome / ID para educadores e familiares.

- R6

CÓDIGO	DESCRIÇÃO
R6	Adicione recursos de pesquisa e filtros para facilitar que educadores e profissionais de saúde possam gerenciar de maneira mais fácil as estatísticas, a coleta de dados e o gerenciamento de usuários.

Requisito novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_07	Funcional	Educador, família	Recursos do painel OCARIoT personalizados de acordo com o perfil ou função do usuário	Sistema personalizado para melhor envolvimento: interface do usuário e recursos	Todos

Requisito Antigo



Processo de compilação para reescrita de um Requisito final

R6

Para melhorar a usabilidade do painel no novo design, é previsto adicionar um formulário de pesquisa para os usuários que associaram vários alunos. Eles devem ser capazes de procurar um aluno por id, gênero, idade, pontuação, etc... O formulário de busca irá filtrar em tempo real a lista completa dos alunos, a fim de fornecer ao usuário apenas os alunos que correspondem aos critérios de pesquisa.

Esta funcionalidade do sistema deve se comportar de forma personalizada, dependendo do perfil do usuário:

- Em Food Intake - Filtrar por ID para profissionais de saúde e por nome / ID para educadores e familiares.

- R7

CÓDIGO	DESCRIÇÃO
R7	Adicionar menu de configurações para personalizar: notificações (ativar / desativar, frequência para cada desafio), componentes do avatar, como cor da pele, sexo etc.

Requisito novo



Processo de Compilação para reescrita de um Requisito Final

R7

Na nova versão do painel, será prevista uma interface de usuário para personalizar a aparência do sistema e alguns parâmetros, como a frequência das notificações. Eles devem receber notificações para as funcionalidades do sistema, que precisam ser refinadas e alinhadas ao Cliente, necessitando de uma melhor construção.

- R9

CÓDIGO	DESCRIÇÃO
R9	Crie um painel de gerenciamento de missões (MMD) para vincular dados provenientes do DSS e do aplicativo OCARIoT.

Requisito Novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_07	Funcional	Educador, família	Recursos do painel OCARIoT personalizados de acordo com o perfil ou função do usuário	Sistema personalizado para melhor envolvimento interface do usuário e recursos	Todos
Req_11	Funcional	Família, educador, profissional de saúde	Informação estatística do participante BI	Visualização geral dos participantes e sua interação com o aplicativo, o que permite que a equipe de saúde e a equipe técnica realizem estudos dos parâmetros coletados.	Todos
Req_12	Funcional	Criança, família, provedor técnico	Sincronização de dispositivos	Processo de sincronização do aplicativo com o wearable (e / ou outros dispositivos)	Todos
Req_18	Usabilidade e Humanidade	Provedor de tecnologia	Para fornecer diferentes tipos de visualização para a família e educador	Visualizações de UI diferentes por função	Todos
Req_25	Operacional e Ambiental	Provedor de tecnologia	Visualização de dados estatísticos por usuário	Fornecer ferramentas de visualização de dados para os diferentes usuários	Todos



Processo de compilação para reescrita de um Requisito final

Requisito Antigo

- R10

CÓDIGO	DESCRIÇÃO
R10	Inclua mais detalhes no menu de acompanhamento de alimentos: 1) questionários; 2) criação de cardápio para educadores e famílias; 3) entrada proveniente do aplicativo OCARIoT (estudantes).

Requisito Novo

REQUISITOS ANTIGOS - DASHBOARD					
REQUISITO	Tipo de Requisito	Stakeholders	Descrição	Justificativa	Casos de Uso
Req_02	Funcional	Profissional de Saúde	Rastreamento básico de alimentos: entender as categorias de alimentos, a maneira de cozinhar e a estimativa de quantidade em cada relatório de alimentos	O Sistema não deve ser capaz de classificar as crianças em perfis nutricionais agregados	Rastreamento de alimentos
Req_07	Funcional	Educador, família	Recursos do painel OCARIoT personalizados de acordo com o perfil ou função do usuário	Sistema personalizado para melhor envolvimento interface do usuário e recursos	Todos
Req_11	Funcional	Família, educador, profissional de saúde	Informação estatística do participante BI	Visualização geral dos participantes e sua interação com o aplicativo, o que permite que a equipe de saúde e a equipe técnica realizem estudos dos parâmetros coletados.	Todos
Req_12	Funcional	Criança, família, provedor técnico	Sincronização de dispositivos	Processo de sincronização do aplicativo com o wearable (e / ou outros dispositivos)	Todos
Req_18	Usabilidade e Humanidade	Provedor de tecnologia	Para fornecer diferentes tipos de visualização para a família e educador	Visualizações de UI diferentes por função	Todos
Req_22	Desempenho	Profissional de saúde	Reforço da atividade física e rastreamento de alimentos com validação cruzada de dados entre diferentes usuários (crianças / famílias, crianças / educadores)	O processo de validação cruzada melhora a qualidade dos dados coletados e os resultados no final do projeto	Todos
Req_24	Desempenho	Crianças e família	Introdução de comida o mais fácil possível	Desde que eles têm que fazer isso, é importante para simplificar essa ação	Rastreamento de alimentos
Req_25	Operacional e Ambiental	Provedor de tecnologia	Visualização de dados estatísticos por usuário	Fornecer ferramentas de visualização de dados para os diferentes usuários	Todos



Processo de compilação para reescrita de um Requisito final

Requisitos Antigos

R9

Nesta seção será descrita a parte do dashboard chamada Painel de gerenciamento de missão (MMD). Ele descreve uma nova interface do usuário projetada para corresponder aos serviços fornecidos pelo DSS do WPI que implementa as missões propostas aos alunos no aplicativo OCARIoT, havendo assim sincronização de dados com o dispositivo. As recomendações serão disparadas na plataforma OCARIoT, a fim de ajudar pais e educadores a apoiar os alunos a alcançarem o objetivo proposto por uma missão e, ao mesmo tempo, que estarão cientes dos exercícios propostos pelo sistema.

Para esclarecer esses conceitos, faremos as seguintes definições:

- **Missão alvo** - o número a ser atingido para considerar a missão como concluída, por exemplo, 15 min / dia
- **Missão** - é a atividade específica que a criança deve realizar para completar o comportamento proposto e o desafio, e a quantificação a ser usada serão baseadas na faixa de metas definidas para o desafio associado. (Exemplo: gasta pelo menos 12 minutos / dia praticando atividade física de intensidade moderada a vigorosa)
- **Recomendação do painel**, um plano de ação proposto para pais / educadores para apoiar a criança a alcançar a missão, a ser mostrado através do painel. (Exemplo: incentivar a criança para caminhar junto depois da escola)
- **Perguntas do painel (questionários)** - um conjunto de perguntas a serem feitas aos pais / educadores (não mais que 3) e as respostas para a seleção (não mais que 4) quando eles estiverem validando uma missão como concluída no painel. (Exemplo: por exemplo qual atividade física sua criança praticou durante a semana? Futebol, natação, corrida, outros).
- **Habilidades** - uma lista das habilidades associadas à missão, a primeira é a habilidade principal e o restante são habilidades secundárias (opcionais). (Exemplo: Destreza).
- **As missões**, a missão alvo e as recomendações estão agrupadas nas três habilidades do agente: destreza, inteligência e disciplina. Essas habilidades serão as mesmas do aplicativo OCARIoT.

O conceito principal nesta parte do painel é a missão (conceito compartilhado entre painel, DSS e aplicativo). Para cada aluno será possível acessar uma visão que mostrará:

- O ID da missão
- O identificador do criador da missão, educador ou DSS
- A habilidade associada à missão
- A descrição da missão
- A duração da missão, e o tempo esperado da missão
- Hora de início, o momento em que a missão começa
- O alvo da missão
- O estado (alcançado ou não alcançado) da missão e, eventualmente, o % do progresso se aplicável.
- As recomendações associadas ao alvo da missão
- O estado de validação da missão, caso precise ser revisado pelos pais e / ou educadores.

A especificação segue-se com o recurso personalizado para funcionalidade dependente do perfil ou função do usuário, segue-se:

- Os educadores também devem ser capazes de criar e enviar novas missões ao aplicativo OCARIoT.
- O MMD permitirá, apenas educadores, a criar missões. Porém apenas criam missões para o piloto (exceto no qual foram parte).
- Uma missão criada envolve todo o piloto da escola.
- O MMD permitirá atualizar e validar missões
- A maioria das missões será validada de forma transparente pelo DSS, algumas missões específicas devem poder ser validadas pelos pais e educadores, para uma criança ou um grupo de crianças.
- Deverá ter um conjunto de estatísticas disponível no MMD, ou seja, haverá informações estatísticas dos participantes.

R10

Devido do painel, o rastreamento de alimentos está gerenciado em 3 níveis diferentes. Haverá recursos da funcionalidade personalizadas de acordo com o perfil ou função do usuário:

- Somente pais e educadores poderão editar o rastreamento da comida das crianças.
- Os profissionais de saúde só poderão revisar os dados.

UI - Nível 1: Linha de base (questionários)

A linha de base é como uma fotografia do estado da criança com base em questionários validados relacionados à frequência alimentar e hábitos alimentares. Eles serão preenchidos periodicamente pelos alunos e pais pelo menos 3 vezes por período de avaliação. Esses questionários devem poder ser:

- Revisados
- Criados
- Excluídos e atualizados.

As estatísticas provenientes desses dados serão frequências de consumo de alimentos e categoria relacionada aos hábitos alimentares das crianças.

UI - Nível 2: registro de dieta

A gravação da dieta será a parte do rastreamento de alimentos relacionado ao registro diário de alimentos. A gravação da dieta permitirá:

- Criar,
- Analisar,
- Excluir e validar o registro da dieta.
- Um registro alimentar deve ocorrer da maneira mais simplificada possível, este será a representação qualitativa de um café da manhã, Lanche da manhã / tarde, almoço, ou jantar de uma criança.

- Incluirá
 - Data
 - Tipo café da manhã, manhã, almoço, lanche ou jantar de uma criança
 - Um conjunto de bandeiras, uma para cada categoria de alimento, como frutas, vegetais, carne, peixe, ovo, etc, que será selecionado pelo usuário. Conjuntamente a quantidade de ingestão deve ser mensurada.

• Dados estatísticos dos participantes provenientes dessa funcionalidade, devem ser comparáveis com a linha de base.

UI - Nível 3: relatório de agente

Nesse nível deve haver sincronização dos dispositivos, estando vinculado o rastreamento de alimentos no painel do OCARIoT ao aplicativo OCARIoT.

- No aplicativo, os alunos precisam preencher um "relatório de agente secreto" que incluirá informações sobre comida.
- Esta parte do painel informa todas as informações coletadas pelo relatório do agente secreto relacionadas à comida em uma exibição de tabela. Os pais têm que interagir com essa tabela e validar as respostas das crianças

