



**UNIVERSIDADE FEDERAL DE PERNAMBUCO**  
**CENTRO DE INFORMÁTICA - CIN**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

Júlia Carneiro de Almeida

**Especificação de Requisitos da Experiência de Aprendizagem em**  
**Plataformas Ubíquas**

Trabalho de Conclusão de Curso

Recife,

2017.

JÚLIA CARNEIRO DE ALMEIDA

**Especificação de Requisitos da Experiência de Aprendizagem em  
Plataformas Ubíquas**

Trabalho de graduação apresentado à banca examinadora composta pelos Professores Alex Sandro Gomes e Carla Taciana Lima Lourenco Schuenemann como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação da Universidade Federal de Pernambuco.

**Orientador:** Alex Sandro Gomes.

Aprovado em \_\_\_\_\_ de \_\_\_\_\_ de 2017.

**BANCA EXAMINADORA**

Alex Sandro Gomes

---

Carla Taciana Lima Lourenco Schuenemann

---

Recife,  
2017.

## **Agradecimentos**

Agradeço primeiramente a Deus, por me conceder forças e capacidade de chegar até onde cheguei e concluir o meu curso.

Aos meus pais, Sandoval Malta de Almeida Neto e Angélica Bianca Costa Carneiro por me apoiarem sempre e principalmente durante a minha graduação para que eu pudesse continuar e chegar até onde cheguei. Por todas as idas e vindas ao Centro de Informática, as esperas até a hora de largar da aula.

Aos meus avós, Maria da Glória, Licurgo Almeida e Josefa Francinete que me acompanharam e me aconselharam sempre!

Aos meus irmãos, Glenda Almeida, minha plateia, revisora e conselheira, e os gêmeos, Sandoval Almeida Filho e Licurgo Almeida Neto, por pararem de fazer barulho para que eu pudesse me concentrar e estudar.

Ao meu namorado e companheiro, Tiago Nogueira, por toda a paciência em me ouvir falando várias vezes sobre o meu tema, pelos finais de semanas juntos fazendo o TCC, pelas dúvidas tiradas e por todo apoio, pois quando eu mais precisei você estava lá.

À minha família, por terem acreditado no meu potencial e que eu conseguiria sim concluir o meu ensino superior em uma Universidade Federal.

Ao meu sogro, Aloísio Nogueira por todo incentivo e apoio sempre!

Ao meu orientador, Alex Sandro Gomes, que me auxiliou e apoiou durante essa caminhada. Pelo um brilhante empenho e dedicação à elaboração deste trabalho.

Aos colegas do CCTE, Luma Seixas, Hugo Souza, Izautino Oliveira, Ivanildo Melo, por todo o auxílio durante o desenvolvimento deste trabalho.

A todos os amigos do Centro de Informática, que me ajudaram e contribuíram para a minha formação, seja no desenvolvimento de trabalhos em grupo, como atividades do dia dia, como pelas conversas nos corredores no centro de informática.

Aos colegas de estágio na PROCIT - UFPE que sempre me ajudaram e apoiaram desde o início.

A UFPE e ao Centro de Informática, por ter me proporcionado um ambiente impecável de estudo e apoio acadêmico.

A todos que contribuíram de uma maneira geral na minha vida pessoal, acadêmica e profissional.

*“Never let the fear of striking out,  
keep you from playing the game”.*

A Cinderella Story.

## **Resumo**

Em consequência aos avanços tecnológicos, o número de dispositivos aumentou, surgindo diferentes *smartphones*, *tablets* e *smartwatches* presentes tanto na realização de atividades pessoais como profissionais. Diante desses avanços surgiu a computação ubíqua, a qual une características da computação móvel com suas próprias características de onipresença e sensibilidade ao contexto. Desse modo, em decorrência desse novo paradigma, surgiram os *softwares* ubíquos e com eles a necessidade de técnicas e métodos da engenharia de *software* e engenharia de requisitos que proporcionem suporte para a realização das atividades de desenvolvimento e concepção de um *software* ubíquo. O presente trabalho objetivou-se em identificar através da literatura abordagens que destacam preocupações e soluções para o processo de especificação de requisitos de *softwares* ubíquos. Tais abordagens proporcionam fundamentação teórica para a identificação da proposta de atividades para o processo de especificação. Dessa forma, as atividades encontradas forneceram base para criação da proposta, as quais foram aplicadas ao contexto real de uma aplicação ubíqua de aprendizagem, o *Youubi Heritage Education*. Diante disso, obtiveram-se resultados e artefatos como, os diagramas de caso de uso e diagrama de atividades, os quais contribuem para a realização de outras atividades do ciclo de desenvolvimento.

**Palavras-chave:** Computação ubíqua, engenharia de requisitos, aprendizagem ubíqua.

## **Abstract**

As a result of technological advances, the number of devices increased, resulting in different smartphones, tablets and smartwatches present in both personal and professional activities. In the face of these advances, ubiquitous computing has emerged, which unites characteristics of mobile computing with its own characteristics of omnipresence and context sensitivity. Thus, as a result of this new paradigm, ubiquitous software appeared and with them the need for techniques and methods of software engineering and requirements engineering that provide support for the development and design activities of ubiquitous software. The present work aimed to identify through the literature approaches that highlight concerns and solutions for the process of specifying requirements of ubiquitous software. Such approaches provide theoretical basis for the identification of the proposed activities for the specification process. Thus, the activities found provided the basis for proposal creation, which were applied to the real context of a ubiquitous learning application, Youubi Heritage Education. In this way, results and artifacts were obtained, such as case-of-use diagrams and activity diagram, which contribute to other activities in the development cycle.

**Keywords:** Ubiquitous computing, requirements engineering, ubiquitous learning.

## **Lista de Figuras**

**Figura 1** - Interface das aplicações - BCAULS e *PerZoovasive*;

**Figura 2** - Exemplo de requisito presente no DDR do *Youubi Heritage Education*;

**Figura 3** - Casos de uso dos requisitos funcionais comuns;

**Figura 4** - Casos de uso dos requisitos funcionais de educação patrimonial - Parte 1;

**Figura 5** - Casos de uso dos requisitos funcionais de educação patrimonial - Parte 2;

**Figura 6** - Casos de uso dos requisitos funcionais de educação patrimonial - Parte 3;

**Figura 7** - Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 1;

**Figura 8** - Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 2;

**Figura 9** - Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 3;

**Figura 10** - Diagrama de atividade da manipulação de interesses;

**Figura 11** - Diagrama de atividade da manipulação de desafios;

**Figura 12** - Diagrama de atividade da manipulação de memória;

**Figura 13** - Diagrama de atividade da manipulação de rotas;

**Figura 14** - Recorte do diagrama de atividade da manipulação de interesses;

**Figura 15** - Tela de login do protótipo de baixa fidelidade do *Youubi*;

**Figura 16** - Requisito atualizado no DDR - *Login* do aprendiz;

**Figura 17** - Possibilidade de incluir o requisito “*Enviar notificação*”.

## **Lista de Quadros**

**Quadro 1** - Abordagens identificadas;

**Quadro 2** - Mapeamento do processo de identificação e especificação de requisitos.

## **Lista de Abreviaturas e Siglas**

**PDA** - Assistente Pessoal Digital (*Personal Digital Assistant*);

**GPS** - Sistema De Posicionamento Global (*Global Positioning System*);

**IEEE** - Instituto de Engenheiros Eletricistas e Eletrônicos;

**EES** - Especificação de Exigências de *Software* (*Software Requirements Specification - SRS*);

**ES** - Engenharia de *Software*;

**SOA** - Arquitetura Orientada a Serviços (*Service-Oriented Architecture*);

**DAS** - Desenvolvimento de um *Software* Adaptável;

**UML** - Linguagem de Modelagem Unificada (*Unified Modeling Language*);

**DDR** - Documento de Definição de Requisitos;

**RF** - Requisitos Funcionais;

**RNF** - Requisitos Não Funcionais;

**UX** - Experiência do usuário (*User Experience*).

## Sumário

1.	Introdução	13
1.1.	Motivação	13
1.2.	Justificativa	14
1.3.	Objetivos	14
1.3.1.	Objetivo Geral	14
1.3.2.	Objetivos Específicos	15
1.4.	Metodologia da Pesquisa	15
1.5.	Estrutura do Trabalho	16
2.	Experiência de Aprendizagem em Plataformas Ubíquas	18
2.1.	Computação Ubíqua	18
2.2.	Experiência de Aprendizagem	18
2.3.	Experiência de Aprendizagem em Plataformas Ubíquas	19
2.4.	Características da Computação Ubíqua	20
2.5.	Plataformas Ubíquas de Aprendizagem	21
3.	Especificação de Requisitos da Experiência de Aprendizagem em Plataformas Ubíquas	24
3.1.	Contextualização sobre a Computação Ubíqua e Engenharia de Requisitos	24
3.2.	Técnicas e Práticas da Especificação de Requisitos	25
3.3.	Desafios da Engenharia de <i>Software</i> para a Computação Ubíqua	26
3.4.	Estratégias para especificação e documentação de requisitos para <i>softwares</i> ubíquos	26
3.4.1.	Mapeamento para especificação de requisitos para <i>softwares</i> ubíquos	28
4.	Método	31
4.1.	Contextualização	31
4.2.	Etapas realizadas na pesquisa	31
4.3.	Ferramentas utilizadas	32
4.4.	Estudo de caso - <i>Youubi Heritage Education</i>	33
5.	Resultados	34
5.1.	Proposta de atividades para o processo de especificação de requisitos em plataformas ubíquas	34
5.2.	Documento de Definição de Requisitos (DDR)	36
5.3.	Diagrama de caso de uso	37
5.4.	Diagrama de atividade	44
5.4.1.	Experiência de manipular interesses	45
5.4.2.	Experiência de manipular desafios	46

5.4.3.	Experiência de manipular memórias	47
5.4.4.	Experiência de manipular rotas	48
5.5.	Análise dos resultados	50
6.	Discussão	51
6.1.	Como os diagramas complementam o entendimento do funcionamento dos protótipos de baixa fidelidade	51
6.2.	Como os diagramas orientam os métodos de avaliações e evolução dos modelos conceituais	52
6.3.	Como a combinação do uso de UML e <i>wireframe</i> ampliar a capacidade de perceber requisitos	53
6.4.	A combinação das técnicas de UML e <i>wireframe</i> no processo de concepção	54
7.	Considerações Finais	56
7.1.	Conclusão	56
7.2.	Trabalhos Futuros	56
	Referências	58

## 1. Introdução

Neste capítulo será apresentada a motivação para a realização deste trabalho, bem como a justificativa e a definição do problema. Por fim, será demonstrada a estrutura que compõe as seções deste trabalho.

### 1.1. Motivação

Com o passar dos anos, o número de dispositivos portáteis está cada vez maior devido aos avanços na fabricação de componentes eletrônicos. Tais dispositivos podem ser encontrados com uma considerável capacidade de processamento e recursos para a comunicação sem fio. Esses dispositivos se popularizaram como *handhelds*, *PDA*s, *smartphones*, os quais possuem diversas funcionalidades e interfaces como GPS, rádio e TV, tocadores de áudio e câmeras fotográficas digitais, podendo ser utilizado para atividades pessoais como profissionais (LOUREIRO, *et al*, 2009).

Em consequência aos avanços tecnológicos de *hardware* e *software* surgiu a computação ubíqua, a qual integra a mobilidade em larga escala com a funcionalidade da computação pervasiva. O potencial das aplicações ubíquas é limitado apenas pela imaginação, já que pode permitir a conexão, monitoramento e coordenação de dispositivos localizados em casas, edifícios e carros inteligentes, por meio de redes sem fio locais e de longa distância com alta largura de banda, realizando diversas atividades, tais como o controle de temperatura, luzes e umidade de uma residência (ARAÚJO, 2017).

Diante desse novo cenário, a engenharia de *software* evoluiu para apoiar as atividades de desenvolvimento fornecendo técnicas e processos que estão presentes em todo o ciclo de vida. Dentre esses processos, se encontra a engenharia de requisitos. Tais requisitos são a base de cada projeto, pois definem as necessidades dos *stakeholders* e o que sistema deve fazer. Normalmente são expressos em linguagem natural objetivando-se capturar uma solicitação ou problema de maneira clara, fornecer base para o planejamento do desenvolvimento do sistema e permitir o gerenciamento dos riscos, impactos e efeitos, muito antes do *software* ser desenvolvido (DICK, HULL, JACKSON, 2017).

Dessa forma o processo de engenharia de requisitos apoia diversas categorias de *software*, dentre elas, os *softwares* ubíquos.

## **1.2. Justificativa**

Com o surgimento da computação ubíqua, novos desafios apareceram relacionados aos processos da engenharia *software* e engenharia de requisitos. Dessa forma, um dos principais problemas de pesquisa está centralizado no processo de especificação de requisitos de *softwares* ubíquos.

De acordo com Spínola (2010), a engenharia de *software* possui metodologias que dão suporte a construção de um *software*. Dessa forma, diante deste novo cenário surgiram novas abordagens para os projetos de *software* ubíquos, os quais são elaborados de acordo com os conhecimentos práticos e com os problemas vivenciados.

Nesta situação, iniciativas como a de Mota (2013) e Spínola (2010) podem ser observadas nos últimos anos destacando o apoio ao processo de especificação de requisitos para *softwares* ubíquos. Entretanto, Mota (2013) destaca dificuldades, riscos e preocupações relacionadas à existência de poucas abordagens na literatura que relatam e proporcionam apoio para a construção desse tipo de *software*.

De acordo com Mota (2013), Rocha, et al (2011) e Hong, Chiu, Shen (2005), deve-se existir uma preocupação para que durante o processo de desenvolvimento seja levada em consideração os diversos fatores e características da computação ubíqua, já que essa categoria de *software* possui diversas características específicas e a utilização de abordagens tradicionais da engenharia de *software* podem não tratá-las da maneira adequada (MOTA, 2013).

## **1.3. Objetivos**

Neste tópico serão apresentados o objetivo geral e os objetivos específicos.

### **1.3.1. Objetivo Geral**

Este trabalho tem como objetivo realizar uma análise sobre o processo de engenharia de requisitos ou especificação de *software*, com objetivo identificar e determinar recomendações para o processo de especificação dos requisitos de *softwares* ubíquos que proporcionem a experiência de utilizar uma aplicação ubíqua, como estudo de caso a plataforma *Youubi Heritage Education*.

Sendo assim, serão representadas as demandas, necessidades e desafios do processo de especificação, assim como exposto por Mota (2013). De maneira que através da criação da documentação fundamentada nas práticas e modelos encontrados na literatura, poderão ser obtidos artefatos consistentes e que auxiliem aos desenvolvedores e partes envolvidas durante todo o ciclo de vida do projeto.

### **1.3.2. Objetivos Específicos**

Os objetivos específicos deste trabalho são os seguintes:

- Identificar técnicas e procedimentos utilizados no processo de especificação dos requisitos para *softwares* ubíquos;
- Destacar atividades para o processo de especificação de requisitos de *softwares* ubíquos;
- Criar artefatos para construção da documentação de requisitos da plataforma *Youubi*;
- Estabelecer mecanismos e atividades para o desenvolvimento desse tipo de *software*.
- Compreender os efeitos que os artefatos criados causam na criação dos protótipos e modelos conceituais do *Youubi*;

### **1.4. Metodologia da Pesquisa**

Esta pesquisa foi realizada em cinco etapas. A primeira etapa compreende na realização de uma revisão bibliográfica, a qual teve como objetivo identificar relatos sobre o processo de engenharia de requisitos no contexto de plataformas ubíquas. Buscou-se também entender os conceitos relacionados como sobre a engenharia de requisitos, computação ubíqua e *u-learning*.

Na segunda etapa foi realizado um estudo para identificar características e desafios relacionados ao contexto ubíquo.

Na terceira etapa buscou-se entender a aplicação da computação ubíqua no ambiente de sala de aula, ou seja, o que o *u-learning* ou aprendizagem ubíqua proporciona de desafios e benefícios.

Posteriormente, foram identificados métodos e atividades para realização da especificação de requisitos. A partir das atividades recomendadas modelaram-se alguns artefatos, como o caso de uso e diagramas de atividade, considerando todas as informações obtidas anteriormente sobre o contexto. Buscou-se também entender o contexto das plataformas de aprendizagem ubíqua e o *Youubi*.

E por fim, podem-se observar algumas discussões sobre os impactos e resultados obtidos através das recomendações das atividades, destacando os benefícios e desafios para o contexto da aplicação *Youubi*.

### **1.5. Estrutura do Trabalho**

Este trabalho além de demonstrar inicialmente seu contexto, justificativa, objetivo e metodologia, também desenvolverá o seguinte escopo:

No capítulo 2 expõe a importância da experiência de aprendizagem em um contexto geral e experiência de aprendizagem ubíqua, demonstrando como se inicia e provoca a aprendizagem nos seres humanos. Também serão apontados conceitos da computação ubíqua, suas características e exemplos de plataformas ubíquas de aprendizagem.

No capítulo 3 será apresentada uma breve contextualização sobre a engenharia de requisitos na computação ubíqua, bem como técnicas e práticas da especificação de requisitos. Além de levantar estratégias e os principais desafios enfrentados no desenvolvimento da engenharia de *software* aplicado ao contexto da computação ubíqua.

No capítulo 4 será demonstrado o método utilizado para desenvolvimento da pesquisa e obtenção dos resultados, bem como as técnicas, ferramentas e contextualizar sobre o estudo de caso.

No capítulo 5 serão apresentados os resultados, expondo as atividades recomendadas que foram propostas para o processo de especificação de requisitos para *softwares* ubíquos e a criação de artefatos UML, que compreende parte da documentação.

No capítulo 6 serão apontadas as discussões sobre os benefícios ou desafios obtidos pelos artefatos criados para a aplicação *Youubi*.

Por fim, no capítulo 7 apresenta as conclusões obtidas pela pesquisa e a perspectiva de trabalhos futuros.

## **2. Experiência de Aprendizagem em Plataformas Ubíquas**

Neste capítulo será apresentada uma breve contextualização sobre a experiência da aprendizagem em plataformas ubíquas, bem como suas características e desafios de acordo com a literatura.

### **2.1. Computação Ubíqua**

Com o avanço da tecnologia surgiu a computação ubíqua, a qual integra algumas características da computação móvel, da mobilidade em larga escala e da computação pervasiva (ARAÚJO, 2003).

Ranga (2015) define a computação ubíqua como um modelo pós-*desktop* da interação homem-computador. Suas atividades envolvem a utilização simultânea de uma série de dispositivos e sistemas de computação de maneira onipresente, a fim de que seus serviços ou facilidades possam através do uso de dispositivos comuns se materializarem em qualquer momento ou lugar de forma transparente (SPÍNOLA, et. al., 2006), como havia sido previsto por Weiser (1999).

Diante disso, a computação ubíqua compreende em uma nova tendência de tecnologias da informação e comunicação. Caracterizada pela incorporação de uma grande quantidade de pequenos dispositivos imersos de forma invisível na vida cotidiana, permitindo o acesso a qualquer coisa, por qualquer pessoa, a qualquer hora e em qualquer lugar (SAKAMURA E KOSHIZUKA, 2005).

Em consequência da computação ubíqua surgiram as plataformas e aplicações ubíquas, as quais podem ser encontradas e aplicadas a diversos contextos, como por exemplo, na assistência domiciliar à saúde como abordado por Carvalho, Copetti e Loques Filho (2011) e na área educacional através de plataformas de *u-learning*, como abordado por Monteiro (2015).

### **2.2. Experiência de Aprendizagem**

De acordo com o dicionário Michaelis o termo “aprendizagem” compreende em um “*processo por meio do qual uma nova informação é incorporada à estrutura cognitiva do*

*indivíduo, por se relacionar a um aspecto relevante dessa estrutura. Esse novo conteúdo poderá modificar aquele já existente, dando-lhe outros significados”.*

De acordo Varela, Thompson, Rosch (2003) a experiência compreende na percepção e interação de maneira atenta com o mundo e com tempo suficiente para se obter *insights* sobre sua própria natureza e funcionamento.

Teixeira (1955) destaca que o processo de experiência compreende em uma atividade contínua de investigação, aprendizagem e descoberta. Dessa maneira que suas necessidades e suas dificuldades fazem-se problemas, os quais são resolvidos pela instituição, pelos hábitos, pelas crenças, pelas artes e pelos conhecimentos construídos e obtidos anteriormente.

Dessa forma, verificou-se que uma situação indeterminada se tornar determinada, em virtude da aquisição do conhecimento. Tal que o saber não é aprender noções já conhecidas, nem é se familiarizar com uma bagagem de informações e conhecimentos, mas sim é descobri-las de novo, como se fosse os primeiros descobridores (TEIXEIRA, 1955).

Dessa forma, a experiência de aprendizagem está diretamente relacionada com o ato de experimentar algo de maneira de gere conhecimento e que esse conhecimento possa ser utilizado posteriormente.

### **2.3. Experiência de Aprendizagem em Plataformas Ubíquas**

A tecnologia está sendo utilizada em diversos contextos, dentre eles no ambiente da sala de aula. Diante disso, a transmissão do conhecimento vem se tornando um grande desafio para os professores, principalmente para aqueles que estudaram e aprenderam a ensinar em uma época em que não se contava com recursos capazes de interagir e colaborar de maneira conectada com os estudantes e sociedade (OLIVEIRA, PORTO E ALVES, 2017).

Diante deste novo cenário, as plataformas de aprendizagem ubíqua proporcionam para seus alunos um ambiente de aprendizagem onipresente, capaz de ler e transitar entre as formas, volumes, massas, movimentos, forças, direções, traços, cores, luzes, ou seja, está presente e circulando pelos ambientes físicos, sejam eles, casas, ruas, parques, avenidas, estradas, realizando a leitura de sinais emitidos em qualquer lugar (SANTAELLA, 2013).

Portanto, a experiência obtida com a utilização de plataformas de aprendizagem ubíqua, oferece um novo olhar diante das novas formas de educar. Além disso, toda essa

circulação de novos dispositivos de informação e comunicação inaugura novas discussões para a educação não escolar, além propor novas formas de educar fora dela (OLIVEIRA, PORTO E ALVES, 2017).

#### 2.4. Características da Computação Ubíqua

Com a preocupação de criar *softwares* ubíquos de qualidade, deve-se buscar entender as características deste novo ambiente de computação (HONG, CHIU e SHEN, 2005). De acordo com Spínola (2010), as características das aplicações ubíquas podem influenciar no processo de desenvolvimento do *software*, tal que se deve destacar uma preocupação na organização dos conceitos, relações associadas às características e fatores de ubiquidade de forma detalhada (SPÍNOLA, 2010 e MOTA, 2013).

Dessa forma, Spínola, Travassos e Massollar (2007) realizou uma revisão sistemática com o objetivo de obter características comuns aos *softwares* ubíquos, além disso, Abowd (1999), Kim e Mirusmonov (2012) e Pinto (2009) também destacaram as principais características que um *software* ubíquo pode possuir. Dessa forma se expõe as seguintes características:

- **Onipresença do serviço:** permite a movimentação física do usuário e proporciona a percepção que os serviços computacionais estão sendo levados consigo;
- **Invisibilidade:** consiste na capacidade de estar presente nos objetos utilizados no dia-a-dia de maneira invisível. Pode gerar uma pequena percepção sobre os objetos ou dispositivos que provêm serviços, já que a interface é minimamente percebida pelo usuário;
- **Sensibilidade ao contexto:** permite a captura de informações sobre o ambiente e gerar algum efeito sobre isso;
- **Comportamento adaptável ou dinamismo de tarefas:** compreende na capacidade de se adaptar ao contexto envolvido, entretanto deve-se ter em vista as limitações;
- **Captura de experiências:** capacidade de capturar experiências, as quais podem ser utilizadas posteriormente;

- **Descoberta de serviços:** compreende na construção de serviços de maneira proativa de acordo com o ambiente em que se encontra. Dessa forma, a aplicação deve interagir com o ambiente e possibilitar que usuário descubra novos serviços ou informações para atingir um objetivo;
- **Composição de funcionalidades:** permite que a partir de serviços básicos, monte-se uma funcionalidade solicitada pelo usuário;
- **Interoperabilidade espontânea:** permite interagir com outros dispositivos durante a movimentação;
- **Heterogeneidade de dispositivos:** provê a migração das suas funcionalidades para outros dispositivos heterogêneos, como por exemplo, um computador e um *tablet*;
- **Tolerância a falhas:** permite a capacidade de tolerar falhas que podem ocorrer durante a utilização das funcionalidades;
- **Personalização:** provê facilidade no preenchimento das preferências do usuário, tornando a utilização mais rica sobre gostos e interesses dos usuários;
- **Escalabilidade:** permite que o sistema aumente a capacidade com a inserção de novos dispositivos no ambiente;
- **Qualidade de Serviço:** capacidade do sistema conseguir manter um nível de serviço predeterminado;
- **Privacidade e Confiança:** permite que o sistema preserve a integridade e confidencialidade das informações.

## 2.5. Plataformas Ubíquas de Aprendizagem

Em consequência a evolução da tecnologia, aumentou-se no número de dispositivos conectados. Atualmente, utilizamos a maioria desses dispositivos para a realização das atividades corriqueiras do dia-dia, de maneira tão natural como já havia sido previsto por Weiser (1999).

O aprendizado ubíquo ou *u-learning* tem como principal objetivo realizar a construção de um ambiente de aprendizagem onipresente, que permita o aprendizado de qualquer pessoa,

em qualquer lugar e a qualquer momento (YAHYA, AHMAD E ABD JALIL, 2010). De acordo Jones e Jo (2004), o desenvolvimento deste ambiente de aprendizagem combina as vantagens da aprendizagem adaptativa, com os benefícios da computação ubíqua e a flexibilidade de dispositivos móveis.

Contudo, ainda é contexto distante de ser aproveitado para as práticas pedagógicas em sala de aula, já que atualmente os dispositivos normalmente são utilizados de forma pessoal e isolados do contexto das interações sociais que auxiliam as práticas colaborativas de aprendizagem com tecnologia (GOMES, BRITO e AMORIM, 2015). Sendo assim, os ambientes educacionais precisam evoluir de modo que permita acessos aos processos educacionais, de maneira que possibilite a realização de mudanças expressivas na sala de aula e que também permita a integração com as novas funcionalidades contidas nos dispositivos móveis, proporcionando uma interação com o meio e a captura de imagens, sons, localização e vídeos de acordo com o ambiente (GALENO, 2010).

Atualmente, já podemos encontrar algumas aplicações *u-learning* utilizadas e documentadas na literatura. A aplicação *BCAULS* é um sistema de aprendizagem ubíquo para museus, o qual teve o seu estudo e experimento conduzido no *National Museum of Natural Science*, Taiwan (CHIU, TSENG, HSU, 2017). De outro modo Barbosa, Hahn, Barbosa e Geyer (2007) descrevem uma aplicação móvel e ubíqua chamada de *Undergraduate Course of Reference*, o *GRefe*, que possui quatro programas de aprendizagem e tem como propósito estimular a interação do grupo em diversas áreas do conhecimento. Em outro contexto Rouillard e Laroussi (2008) demonstram o *PerZoovasive*, uma aplicação móvel que fornece conteúdo informativo sobre o Zoológico de Londres, por meio da leitura de códigos de barra em placas ao longo do caminho. Essas plataformas possuem características em comum, elas permitem que seus usuários se conectem aos sensores, *QR codes*, *RFIDs* e obtenham conteúdo informativo através das aplicações móveis, baseadas na geolocalização. Além disso, possuem características da gamificação, formadas por *quizzes* e desafios, como pode ser visto na figura 1.

**Figura 1:** Interface das aplicações - *BCAULS* e *PerZoovasive*.



**Fonte:** Chiu, Tseng e Hsu, 2017 e Rouillard e Laroussi (2008), respectivamente.

Diante dessas aplicações, o *Youubi Heritage Education* apresenta características em comum as ferramentas citadas acima, entretanto destaca-se pelo fato do seu modelo conceitual está compreendido em seis entidades, são elas: pessoas, postagens, desafios, lugares, grupos e eventos (MONTEIRO, 2015). Tal que irá permitir que os seus usuários tenham a experiência de manipular rotas, memórias, desafios e interagir com outros usuários de maneira que promova o aprendizado colaborativo.

### **3. Especificação de Requisitos da Experiência de Aprendizagem em Plataformas Ubíquas**

Neste capítulo será apresentada uma breve contextualização da relação entre a computação ubíqua e a engenharia de requisitos aplicada ao contexto de aprendizagem.

#### **3.1. Contextualização sobre a Computação Ubíqua e Engenharia de Requisitos**

A computação ubíqua estabelece uso de um conjunto de dispositivos de tamanhos variados, formatos e funções diferentes, que auxiliam as pessoas na realização de diversas atividades cotidianas de forma coordenada e autônoma (SATYANARAYANAN, 2001).

De acordo com Weiser (1999) e Satyanarayanan (2001), cada vez mais estamos tendo uma maneira diferente de pensar sobre computadores e os diversos dispositivos eletrônicos. Tais dispositivos que já que passaram a ser naturais, sob medida e embutidos em todos os locais e nos mais variados e inusitados objetos do cotidiano, permitindo que os próprios computadores desapareçam ao fundo assim como ocorre com a energia elétrica.

Dessa forma, com a computação ubíqua, surgiram componentes de *software* que fornecem suporte as novas formas de interação, os quais são capazes de reconhecer a face e a voz dos usuários, rastrear a sua localização e prover formas de incorporar realidade aumentada. Em decorrência a esse novo cenário surgiu a necessidade do apoio dos processos de engenharia de *software* durante o desenvolvimento, exigindo a adoção de novas estratégias devido ao aumento da complexidade (ROCHA, *et al*, 2011).

Diante disso, a engenharia de *software* necessitou evoluir, para que seus processos pudessem apoiar o desenvolvimento de diferentes categorias de *software*, dentre eles os *softwares* ubíquos. Além disso, despertou a preocupação sobre os requisitos, já que eles possuem um papel central no processo de desenvolvimento, pois geram a especificação das estimativas, modelagem, projeto, implementação e testes, ou seja, estão presentes ao longo de todo o ciclo de desenvolvimento (MOTA, 2013 e ROCHA, *et al*, 2011).

Entretanto como abordado por Spínola (2010), a literatura possui poucas investigações sobre como realizar o processo de engenharia de *software* e engenharia de requisitos para *softwares* ubíquos dificultando o desenvolvimento de um *software* ubíquo.

### 3.2. Técnicas e Práticas da Especificação de Requisitos

De acordo com Sommerville (2011), o processo de especificação de requisitos objetiva-se na compreensão e escrita dos requisitos de sistema e de usuário de maneira clara, completa, consistentes e de fácil entendimento.

O IEEE Std 610.12-1990 (IEEE, 1990) define o processo de especificação como um documento que especifica os requisitos para um sistema ou parte de dele. Tal documento representa uma declaração oficial e detalhada de todos os aspectos a serem construídos, determinando o que deve ser implementado (PRESSMAN, 2011 e SOMMERVILLE, 2011).

De acordo com o IEEE Std 830-1998 (IEEE, 1998), propõe um processo de especificação de *software* de uma maneira geral deve considerar os seguintes pontos:

- **Natureza do documento e ambiente do documento de EES (Especificação de exigências de *software*):** deve conter todas as informações referentes a criação do documento, funcionalidades, interfaces externas, performance, atributos, restrições e quem criou o documento;
- **Características de um bom documento de EES:** é fundamental possuir as seguintes características: ser correto, não ambíguo, completo, consistente, classificável por importância e/ou estabilidade, verificável, modificável e rastreável;
- **Preparação conjunta do documento de EES:** para criação do documento de requisitos deve ser preparado de forma conjunta, ou seja, o cliente e o desenvolvedor do projeto de *software*;
- **Evolução do documento de EES:** objetiva-se em auxiliar e documentar as possíveis necessidades de mudança;
- **Prototipagem:** é utilizada normalmente na fase de desenvolvimento de exigências, de forma que facilite a exposição de algumas características do *software*;
- **Incorporação de design no documento de EES:** incorporar os pontos que descrevem um sistema ou um subcomponente particular de um sistema e/ou suas interfaces com outros subcomponentes, de acordo com a visão de um design;
- **Incorporação dos requisitos do projeto no documento de EES:** determina que o documento de EES deve focar no produto de *software* e não no processo de produção do produto de *software*.

### **3.3. Desafios da Engenharia de *Software* para a Computação Ubíqua**

Com o surgimento da computação ubíqua, diversas abordagens surgiram nas últimas décadas relacionando-a a engenharia de *software*. Dessa forma, essas abordagens foram empregadas visando à diminuição da complexidade do desenvolvimento, seja através da utilização de *frameworks*, modelos de componentes e/ou arquitetura orientada a serviços (SOA). Entretanto, alguns problemas e desafios persistem dificultando a implantação de uma abordagem mais sistemática de desenvolvimento para esse tipo de *software* (ROCHA, *et al*, 2011).

Diante disso, Costa, Yamin e Geyer (2008) aponta que os desafios enfrentados no processo de desenvolvimento de *software* ubíquo podem ser classificados em duas categorias: os desafios técnicos e desafios metodológicos. Os desafios técnicos estão relacionados aos requisitos difíceis de realizar, tais como a interoperabilidade, descoberta, mobilidade, entre outros. Os desafios metodológicos estão relacionados aos processos, técnicas e ferramentas da ES, utilizados para apoiar o desenvolvimento de *software* ubíquo em larga escala.

Outros desafios podem ser encontrados no processo de criação de um *software* ubíquo, Viana e Andrade (2005) apontam para algumas preocupações para a obtenção da heterogeneidade dos dispositivos e das plataformas de programação, já que o contexto da computação ubíqua envolve diversos dispositivos que fazem o uso de tecnologias diferentes.

Dessa forma, Kindberg e Fox (2002), Rocha *et al* (2011), COSTA, YAMIN e GEYER (2008), Viana e Andrade (2005), destacam os seguintes desafios: interação desacoplada, coordenação, interoperabilidade, descoberta, composição dinâmica de serviços, mobilidade, sensibilidade ao contexto, adaptação, autonomicidade, dependabilidade e segurança, integração, estrutura de programação e robustez. Desafios os quais normalmente estão diretamente ligados a alguma característica dos *softwares* ubíquos.

### **3.4. Estratégias para especificação e documentação de requisitos para *softwares* ubíquos**

Com o surgimento de novos tipos de *software*, novas formas de realizar os processos da engenharia de *software* e de engenharia de requisitos vêm sendo atualizadas e decompostas de maneira que auxiliem o processo de desenvolvimento de *software*.

Entretanto, diante desse cenário poucos artigos podem ser encontrados na literatura abordando as atividades do processo desenvolvimento e concepção desse tipo de *software*. Dessa maneira, aumentam-se as dificuldades, riscos e problemas que podem surgir durante a realização das atividades do ciclo desenvolvimento, dentre elas, o processo de especificação de requisitos.

Sendo assim, a partir desse contexto buscou-se identificar artigos que fornecessem fundamentação teórica e apoiassem o processo de especificação de requisitos para os *softwares* ubíquos.

Portanto, através da revisão da literatura realizada por Spínola (2010), verificou-se a existência de poucas abordagens e recomendações para o processo de identificação e especificação de requisitos, os quais podem ser observados através do Quadro 1. Foram procuradas novas abordagens, entretanto não obteve-se nenhum artigo novo. Dessa forma, Spínola (2010), identificou apenas nove artigos que abordam o processo aplicado a contextos diferentes, permitindo-se analisar os benefícios e desafios enfrentados, de maneira sirvam de base para criação da proposta de recomendação deste presente trabalho.

**Quadro 1:** Abordagens identificadas.

<b>Contexto</b>	<b>Trabalhos Identificados</b>	<b>Abordagem Proposta</b>	<b>Características de Ubiquidade</b>
Definição de requisitos do usuário com base em uma ontologia	(BO et al., 2007)	Identificação	Sensibilidade ao contexto
Metodologia para definir requisitos de sistemas adaptáveis e sensíveis ao contexto	(CHIU et al., 2007)	Identificação e especificação	Sensibilidade ao contexto
Exemplos de requisitos de ubiquidade do projeto ENQUETE-BAISE	(CHENG et al., 2007)	Especificação	Não especificado
Exemplos de requisitos de ubiquidade de um sistema multimodal multimédia	(CHERIF et al., 2007)	Especificação	Não especificado
Mecanismo para definição de requisitos de serviços	(XIANG et al., 2007)	Identificação	Onipresença de Serviços

Metodologia para definir requisitos de sistemas adaptáveis e sensíveis ao contexto	(HONG et al., 2005)	Identificação e especificação	Sensibilidade ao contexto
Modelagem de requisitos com a linguagem KAOS	(GOLDSBY e CHENG, 2007) (CHENG et al., 2005)	Identificação e especificação	Sensibilidade ao contexto e Comportamento adaptável
Casos de uso executáveis	(JORGENSEN e BOSSEN, 2003)	Especificação	Não especificado
Processo sistemático para análise de requisitos de segurança em sistemas embarcados	(MARKOSE et al., 2008)	Identificação e especificação	Não especificado
Processo para definição de requisitos com base nas intenções dos usuários	(OYAMA et al., 2008)(MING et al., 2008)	Identificação	Não especificado

**Fonte:** Spínola (2010) - Adaptado.

### 3.4.1. Mapeamento para especificação de requisitos para *softwares* ubíquos

O mapeamento das atividades para a especificação dos requisitos de *softwares* ubíquos é de suma importância, pois através delas identifica-se os passos para especificação dos requisitos e para a obtenção dos resultados assim como nos artigos.

Sendo assim, a metodologia adotada para efetuar o mapeamento das atividades foi baseada na metodologia proposta por SOUZA (2012), a qual consiste na identificação de atividades ou práticas equivalentes do processo de especificação de requisitos para *softwares* ubíquos de acordo com a literatura. Dessa forma, reúne de forma emparelhada um conjunto de referências de como deve-se fazer em uma determinada situação e cenário (SOUZA, 2012).

Tal conjunto de referências terá como entrada os artigos obtidos através da revisão sistemática realizada por Spínola (2010), observada anteriormente. Dessa forma, o quadro 2 apresenta três colunas contendo a metodologia ou abordagem, o processo ou fase equivalentes e as atividades e práticas:

**Quadro 2:** Mapeamento do processo de identificação e especificação de requisitos.

<b>Metodologia/Abordagem/ Exemplos/Definição</b>	<b>Processo/Fase Equivalente</b>	<b>Atividade/Práticas</b>
<b>A1.</b> Definição de requisitos do usuário com base em uma ontologia	Identificação	-Definir contexto; -Identificar requisitos com base em ontologia; -Decompor os requisitos em sub-requisitos; -Construir uma árvore de requisitos e sub-requisitos.
<b>A2.</b> Exemplos de requisitos de ubiquidade do projeto ENQUETE-BAISE	Especificação	-Listar requisitos (Sem Técnica); -Definir requisitos funcionais através das características genéricas (Requisitos concretos).
<b>A3.</b> Exemplos de requisitos de ubiquidade de um sistema multimodal multimídia	Especificação	-Listar requisitos (Sem Técnica).
<b>A4.</b> Mecanismo para definição de requisitos de serviços	Identificação	-Definir a ontologia para a modelagem de requisitos; -Elicitar detalhes de requisitos e informações estruturais; -Integrar as instâncias dos requisitos para diferentes usuários.
<b>A5.</b> Metodologia para definir requisitos de sistemas adaptáveis e sensíveis ao contexto	Identificação e especificação	-Determinar participantes da definição do software; -Identificar os contextos de aplicação do sistema; -Listar os requisitos específicos; -Investigar os impactos nos diferentes contextos de interações; -Detalhar as funcionalidades para cada característica; -Verificar o atendimento das necessidades; -Projetar visões do processo, capturando características, fonte de dados e interface; -Validar a consistência entre as visões.
<b>A6.</b> Modelagem de requisitos com a linguagem KAOS	Identificação e especificação	-Identificar os requisitos; -Identificar comportamento e variáveis da utilização do sistema; -Definir melhor técnica de mapeamento dos requisitos;
<b>A7.</b> Casos de uso executáveis	Especificação	-Analisar o contexto; -Definir os requisitos através dos casos de uso; -Modelar interações entre o usuário e o sistema;

		-Elaborar conjunto de protótipos; -Verificar e validar requisitos.
<b>A8.</b> Processo sistemático para análise de requisitos de segurança em sistemas embarcados	Identificação e especificação	-Analisar objetos do contexto do sistema com componentes externos; -Representar os subcomponentes com os contextos; -Especificar através de casos de uso as funcionalidades; -Identificar e tratar os casos de uso.
<b>A9.</b> Processo para definição de requisitos com base nas intenções dos usuários	Identificação	-Observar padrões de comportamento dos usuários; -Observar relatos do usuário; -Associar os padrões de comportamento e relatos do usuário aos requisitos do sistema.

**Fonte:** Spínola (2010) – Adaptado; Atividades e práticas obtidas através da metodologia – SOUZA (2012) e análise das atividades – Própria.

Existem diversas formas de identificar e especificar os requisitos, entretanto quando aplicadas ao contexto de *softwares* ubíquos ainda existem poucas abordagens. Algumas atividades comuns entre abordagens foram encontradas, como realização de uma análise do contexto e a listagem dos requisitos, porém ainda se encontram algumas diferenças entre elas. Em suma, essa listagem permite verificar e destacar os pontos mais interessantes entre elas, que se utilizadas de maneira combinada podem provavelmente gerar bons resultados.

## 4. Método

Neste capítulo será apresentada uma justificativa sobre a metodologia a ser utilizada e seus principais conceitos. Além disso, será descrito as etapas necessárias para a realização deste trabalho.

### 4.1. Contextualização

De acordo com Lima e Mioto (2007), uma metodologia pode ser considerada como uma forma de discurso, a qual apresenta o método escolhido como uma direção para o encaminhamento da pesquisa, bem como procedimento realizado para a investigação da solução ou proposta.

O método está dividido em 3 seções: Na seção 4.2 são identificadas as técnicas utilizadas para a criação das atividades propostas, bem como as utilizadas para a identificação dos requisitos do *Youubi*. As ferramentas utilizadas para criação dos artefatos e criação dos diagramas serão apresentadas na seção 4.3. E finalmente, na seção 4.4 é identificada a plataforma ubíqua de aprendizagem *Youubi Heritage Education*, o estudo de caso do presente trabalho.

### 4.2. Etapas realizadas na pesquisa

Inicialmente, para a identificação dos requisitos de *software* buscou-se através da técnica baseada no reuso de requisitos, identificar através da literatura os requisitos que normalmente um *software* ubíquo possui. Além disso, através da colaboração do grupo focal do *Youubi* identificaram-se requisitos dentre os diferentes pontos de vista, de maneira que ouviu-se as opiniões de todos os envolvidos sobre os requisitos já identificados e sobre os novos.

Inicialmente, foi realizada uma análise sobre as recomendações propostas pelo IEEE, conforme pode ser observado na seção 3.2, bem como a verificação de outras estratégias encontradas na literatura, para identificar pontos importantes para a criação da proposta de recomendações.

Dessa forma, objetivando-se gerar de resultados e documentação de qualidade criou-se uma proposta de recomendações para o processo de especificação de requisitos para *softwares* ubíquos. Tais recomendações foram criadas através da utilização de uma adaptação da

metodologia criada e aplicada por Souza (2012), tornando-a mais simples e aplicada apenas para auxiliar na organização e identificação dos passos de cada atividade. Dessa forma, contribuiu para seleção e agrupamento das atividades investigadas, fornecendo insumos suficientes para a gerar as recomendações e os artefatos.

Durante a realização do presente trabalho pode-se utilizar a modelagem de casos de uso, visando identificar novos requisitos, descrever os cenários e mostrar as funcionalidades do *Youubi*. Assim, seria obtido ao final do processo uma versão do documento de definição de requisitos (DDR).

A partir das recomendações propostas também se analisou o que poderiam ser aplicado para o contexto da aplicação *Youubi*. Sendo assim, foram criadas algumas modelagens e artefatos a partir dos requisitos identificados e organizados no DDR, de maneira que foi utilizada a notação UML (*Unified Modeling Language*). Tal notação compreende em uma linguagem gráfica que permite a visualização, especificação, construção e documentação dos artefatos de um determinado *software*. Além disso, permitem incluir alguns aspectos conceituais como processo de negócios, funções do sistema, classes, esquemas de banco de dados e componentes de software (BOOCH, RUMBAUGH, JACOBSON, 2006).

#### **4.3. Ferramentas utilizadas**

Para a criação dos artefatos e modelagem do sistema foi necessário utilizar ferramentas que auxiliasse e facilitasse o processo de construção. Dessa forma, para a criação dos diagramas de caso de uso foi utilizado a ferramenta *Astah Professional*<sup>1</sup>, um *software* para modelagem UML, que possui interface amigável e permite exportação dos diagramas completos nos formatos JPEG e PNG.

Além dessa ferramenta, foi utilizada a ferramenta *online Lucidchart*<sup>2</sup>, a qual permite a criação de fluxogramas, *wireframes* e diagramas, tais como os diagramas de atividades. Optou-se por sua utilização pelo fato de permitir o compartilhamento com outras pessoas e o fácil manuseio, já que também possui uma interface amigável e com exportação também em alguns formatos de imagens.

---

<sup>1</sup> Astah Professional - <http://astah.net/>

<sup>2</sup> Lucidchart – <https://www.lucidchart.com/>

#### 4.4. Estudo de caso - *Youubi Heritage Education*

O *Youubi Heritage Education* objetiva-se em promover o aprendizado geolocalizado através da possibilidade de realizar micro-aprendizagens utilizando um aplicativo móvel. Esta nasceu no Centro de Informática - UFPE, desenvolvido pelo grupo de pesquisa de Ciências Cognitivas e Tecnologia Educacional (CCTE), através da arquitetura de *software* criada por Monteiro (2015). Tal arquitetura de *software* permite criar aplicações clientes com características de *u-learning*, dentre eles o *Youubi* (MONTEIRO, 2015)

Diante disso, atualmente está sendo desenvolvida a aplicação *mobile* do *Youubi* para *android* e *IOS*, e com ele todos os artefatos necessários para o desenvolvimento de um *software*. Dessa forma, teve-se a oportunidade de aplicar alguns dos conhecimentos obtidos em um cenário real de desenvolvimento durante a realização da presente pesquisa. Além disso, pode-se como criar algumas recomendações para o novo contexto de engenharia de requisitos e computação ubíqua, como também criar alguns artefatos necessários e identificados que compõem o processo de especificação de requisitos.

## 5. Resultados

Neste capítulo serão apresentados os resultados obtidos da realização de algumas atividades do processo de engenharia de requisitos aplicadas ao contexto de *softwares* ubíquos. Além disso, serão descritos e apresentados os principais artefatos e modelos criados sobre a plataforma de *e-learning* o *Youubi Heritage Education*.

### 5.1. Proposta de atividades para o processo de especificação de requisitos em plataformas ubíquas

Diante do novo paradigma da computação, a engenharia de *software* precisou evoluir de maneira que proporcionasse auxílio para o desenvolvimento dos *softwares* ubíquos, como pode ser observado anteriormente na seção 3.1.

Para o desenvolvimento de *softwares* tradicionais o IEEE definiu e forneceu um guia de recomendações que auxiliam na análise e descrição dos sistemas, como pode ser observado na seção 3.2 com as recomendações propostas IEEE Std 830 - 1998. Entretanto, essas recomendações buscam abranger diversos tipos de *software* de uma maneira geral, podendo assim não conseguir capturar as diferentes características dos *softwares* ubíquos.

Dessa forma, verificou-se também ao longo desse trabalho que realizar o processo de especificação de requisitos para *softwares* ubíquos é um grande desafio, como pode-se observar na seção 3.3. Sendo assim, criar uma proposta de atividades para o processo, permitiu-se não apenas auxiliar no desenvolvimento das atividades para o *Youubi*, mas como também poderá ajudar na realização da especificação de outros *softwares* ubíquos.

Dessa maneira, através dos resultados obtidos pela revisão da literatura realizada por Spínola (2010) e adaptação e simplificação da metodologia proposta por Souza (2012), pode-se criar a proposta de recomendações de atividades para o processo de especificação de requisitos em plataformas ubíquas.

Através dos artigos e abordagens obtidas pela revisão sistemática (SPÍNOLA, 2010), identificaram-se atividades realizadas pelos autores e através delas verificaram-se quais as atividades importantes e que se enquadrava com a fase que o *Youubi* se encontrava.

Dessa forma, abaixo será apresentada atividades do processo de especificação de

requisitos em plataformas ubíquas, numerada de acordo com as atividades encontradas nos artigos, presentes na seção 3.4.1:

- 1. Definir contexto de aplicação do sistema (1) e determinar participantes da definição do *software* (5)** - pelo fato de estar se tratando de um *software* ubíquo, o processo de identificação se difere em relação aos *softwares* convencionais. Dessa forma, inicialmente devem-se verificar informações e características relevantes sobre o contexto da computação ubíqua, para obtenção de base para o levantamento dos requisitos. Outro fator importante é entender e determinar quais os diferentes *stakeholders* participantes da definição do *software*;
- 2. Identificar os requisitos de maneira tradicional (6) e definir requisitos funcionais através das características genéricas da computação ubíqua (2)** - recomenda-se identificar os requisitos comuns do *software*, ou seja, requisitos que normalmente são encontrados nos softwares, como por exemplo, “Efetuar login”, de maneira tradicional ou utilizando a técnica que o engenheiro de *software* achar mais conveniente, porém devem ser consideradas algumas das características genéricas das plataformas ubíquas, como podem ser observadas na seção 2.4;
- 3. Listar os requisitos específicos para o contexto (5) e detalhes de requisitos e informações estruturais (4)** - após realizar o levantamento dos requisitos através da etapa anterior, recomenda-se elicitar e listar os requisitos específicos do contexto ubíquo levando em consideração as características de ubiquidade. Além de descrever os detalhes referentes aos requisitos e funcionalidades;
- 4. Identificar/especificar através dos casos de uso as funcionalidades (8)** - após a definição dos requisitos, deve-se iniciar o processo de captura das funcionalidades, as quais podem ser capturadas ou atualizadas através da modelagem de caso de uso;
- 5. Modelar interações entre o usuário e o sistema (7)** - através das modelagens de interações pode-se visualizar as interações que o usuário poderá realizar, bem como a identificação de novos e/ou a atualização dos requisitos já identificados. De modo que auxilie posteriormente na criação dos protótipos de *interface*;
- 6. Elaborar conjunto de protótipos (7)** - permite gerar visões de como será a *interface* do sistema;

**7. Verificar e validar requisitos (7)** - por fim, através dos protótipos e artefatos criados anteriormente, pode-se verificar e validar os requisitos identificados e elicitados com os *stakeholders*.

## **5.2. Documento de Definição de Requisitos (DDR)**

O documento de requisitos do *Youubi Heritage Education* foi criado com o objetivo de fornecer informações relacionadas às necessidades de um determinado contexto, propondo o entendimento do *software* que será desenvolvido e mantido.

Antes da criação do documento de requisitos, realizou-se uma análise na literatura para identificar informações e características do cenário em que o *Youubi* está envolvido, bem como sugerido pela atividade 1. Dessa forma, o documento inicia descrevendo e apresentando o projeto do *Youubi Heritage Education*. Logo depois, são descritos informações sobre o público alvo, apontando quem criou e utilizará o documento, ou seja, gerente, *owner*, *UX design team*, programadores, *web design* e *front-end*. Também especifica quais os canais de comunicação, repositórios, horários de reuniões de *sprint*, além de determinar como será o processo de desenvolvimento.

Posteriormente, iniciou-se descrição do escopo funcional sendo realizado de maneira mais tradicional utilizando algumas técnicas de reuso de requisitos e analisando o ponto de vista do grupo focal envolvido no desenvolvimento do *Youubi*, realizando assim a atividade 2. Em seguida, pode-se destacar os módulos e funcionalidades serão gerenciadas, como previstas pela atividade 3. Sendo seguido pelos requisitos não funcionais (RNF) e requisitos funcionais (RF), os quais se dividem em: requisitos funcionais comuns, requisitos funcionais de educação patrimonial e requisitos funcionais de *self-directed learning*.

Os requisitos funcionais comuns compreendem nos requisitos que normalmente todo sistema deve conter, por exemplo: “Efetuar login”. Já os requisitos funcionais de educação patrimonial compreendem nos requisitos que buscam a aprendizagem patrimonial, de acordo com a manipulação das rotas, desafios, memórias e dicas. Por fim, os requisitos funcionais de *self-directed learning*, no qual compreende nos requisitos relacionados a geração de interesses e auto aprendizado.

Dessa forma, para a criação do *template* dos requisitos do DDR foi criada a seguinte estrutura dos requisitos listados como pode ser observado na Figura 2:

**Figura 2:** Exemplo de requisito presente no DDR do *Youubi Heritage Education*.

**2.3 Cadastrar o aprendiz**

O cadastro pode ser realizado de forma simples solicitando apenas um e-mail, senha, nome do usuário e interesse inicial.

**Fluxo alternativo:** Caso o campo senha estiver vazio o servidor cria senha aleatória e envia por email. Caso o campo senha seja preenchido, a senha será a informada pelo usuário e não envia e-mail.

**Telas:**

**Tempo (em horas):**

**Prioridade:**  Essencial  Importante  Desejável

**Dependências:** Nenhuma dependência

**Youubi API:** createPerson

**Fonte:** Documento de Definição de Requisitos (DDR) - *Youubi Heritage Education*.

Os requisitos são numerados para facilitar a rastreabilidade e dependências, sendo seguido pelo nome e descrição. Além disso, se houver fluxo alternativo, deve-se adicionar uma breve descrição determinando a ação a ser realizada. Por fim relacionam os requisitos as telas dos protótipos criados, quanto tempo em horas será necessário para desenvolvê-lo, qual o nível de prioridade, as dependências e qual a chamada da classe na API do *Youubi*.

### 5.3. Diagrama de caso de uso

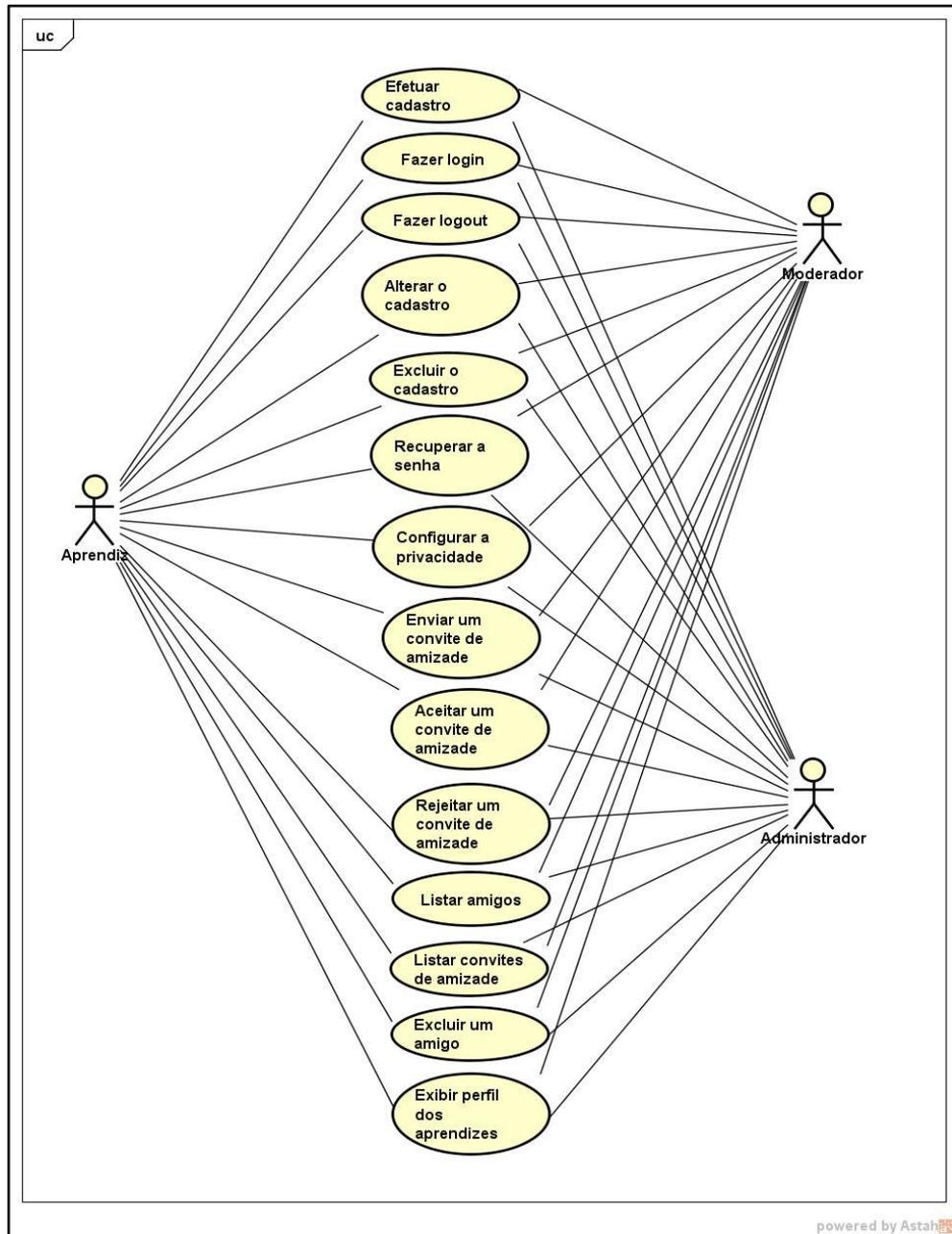
A partir da identificação dos requisitos organizados no DDR, permitiu-se iniciar a criação dos artefatos e modelagens, dentre elas o diagrama de caso de uso, conforme havia sido sugerido pela recomendação 4.

Para criação dos diagramas foi necessário identificar quais as informações essenciais do contexto da modelagem, tal como a definição do ator(es). O *Youubi* possui apenas três atores, o aprendiz, administrador e o moderador. O aprendiz pode realizar diversas atividades dentro do sistema, sendo responsável pelo carregamento e consumo de informações dentre as funcionalidades. Já o moderador pode realizar as mesmas atividades que o aprendiz, entretanto ele se destaca dentre os outros aprendizes por ser um usuário ativo e que contribui diariamente. Por fim, o administrador pode realizar as mesmas atividades dos outros usuários, entretanto ele também é responsável por gerenciar algumas atividades, como o reporte e denúncia de um desafio e/ou rota.

Dessa forma, foram criados os diagramas de caso de uso sobre os requisitos funcionais:

- **Casos de uso para requisitos funcionais comuns:** se encontram as funcionalidades comuns que normalmente todo o sistema deve conter como podem ser visto na Figura 3.

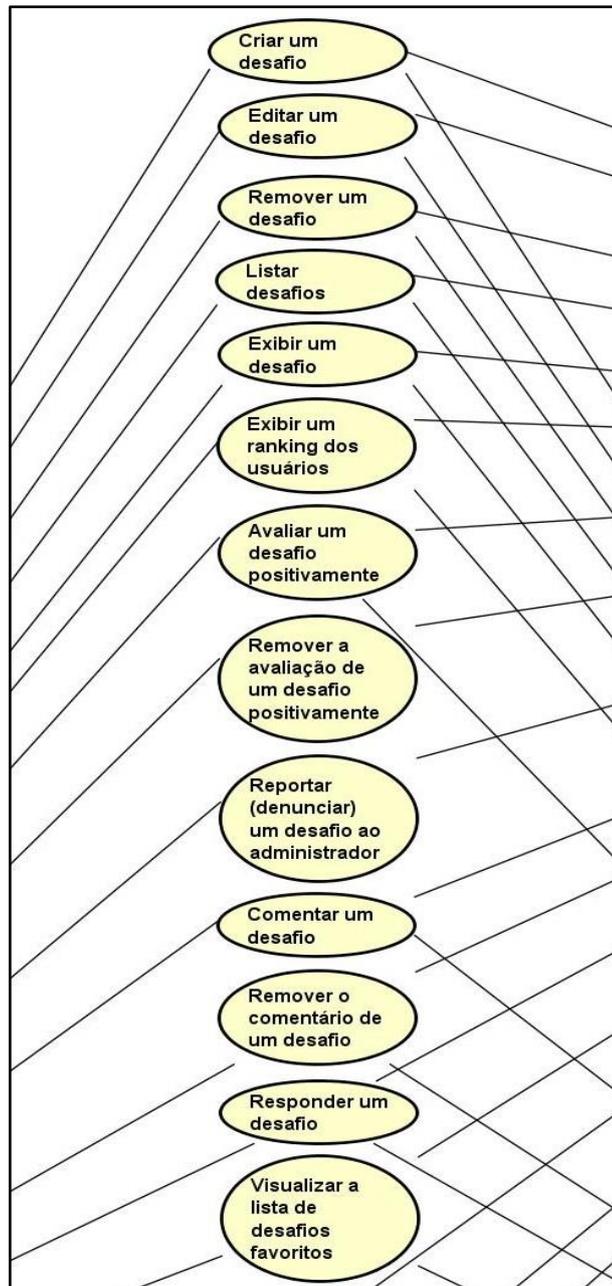
**Figura 3:** Casos de uso dos requisitos funcionais comuns.



Fonte: Própria.

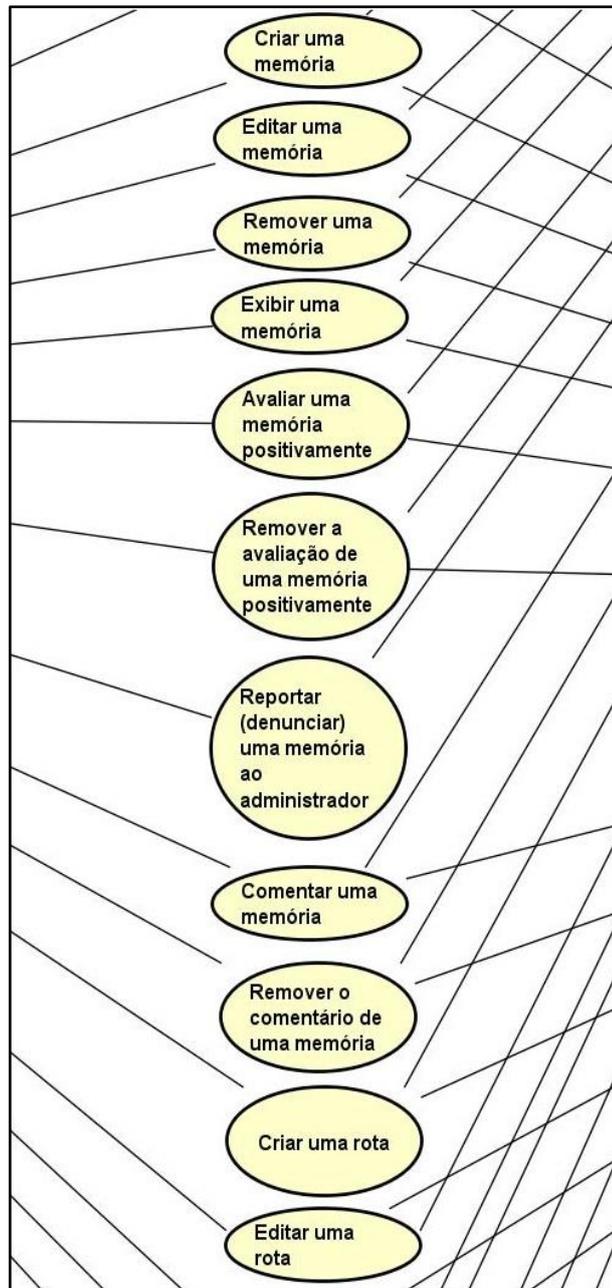
- **Casos de uso dos requisitos funcionais de educação patrimonial:** se encontram os casos de usos que objetiva-se na geração de aprendizagem para o aprendiz, moderador e o controle por parte do administrador, como podem ser vistos nas Figuras 4, 5 e 6.

**Figura 4:** Casos de uso dos requisitos funcionais de educação patrimonial - Parte 1.



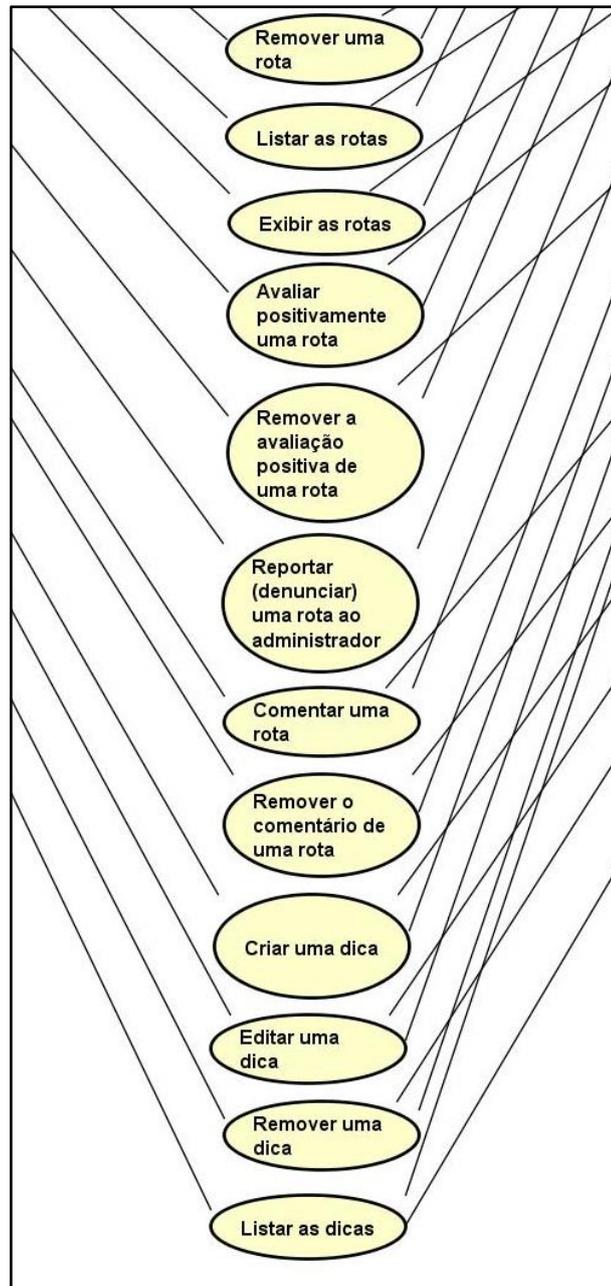
**Fonte:** Própria.

**Figura 5:** Casos de uso dos requisitos funcionais de educação patrimonial - Parte 2.



**Fonte:** Própria.

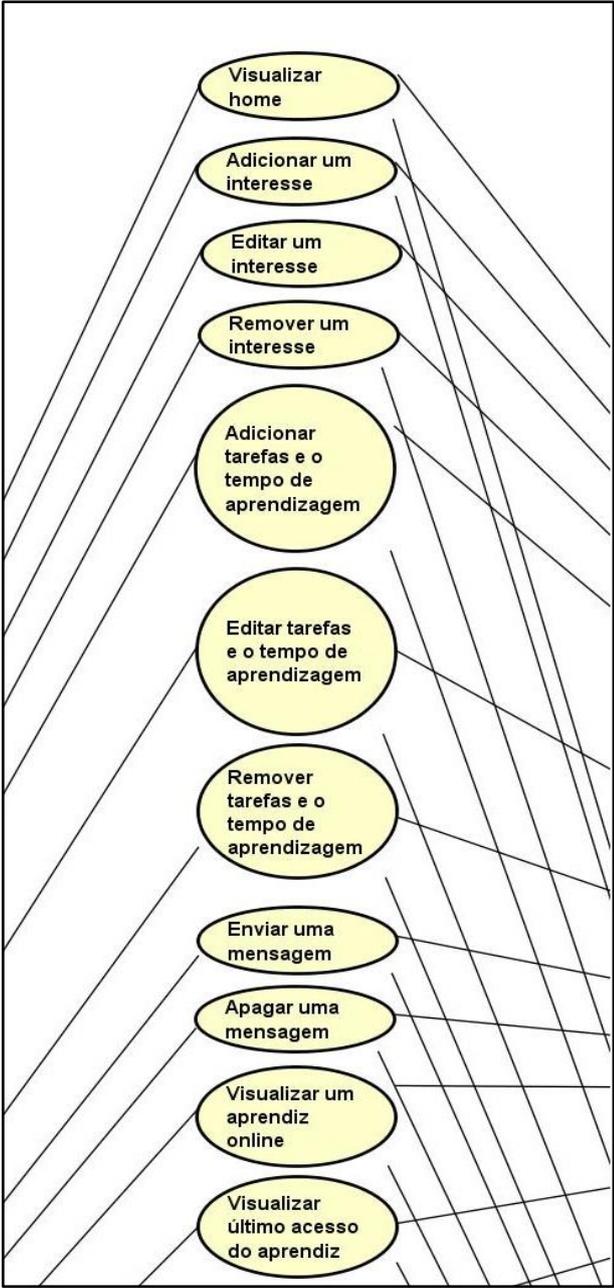
**Figura 6:** Casos de uso dos requisitos funcionais de educação patrimonial - Parte 3.



Fonte: Própria.

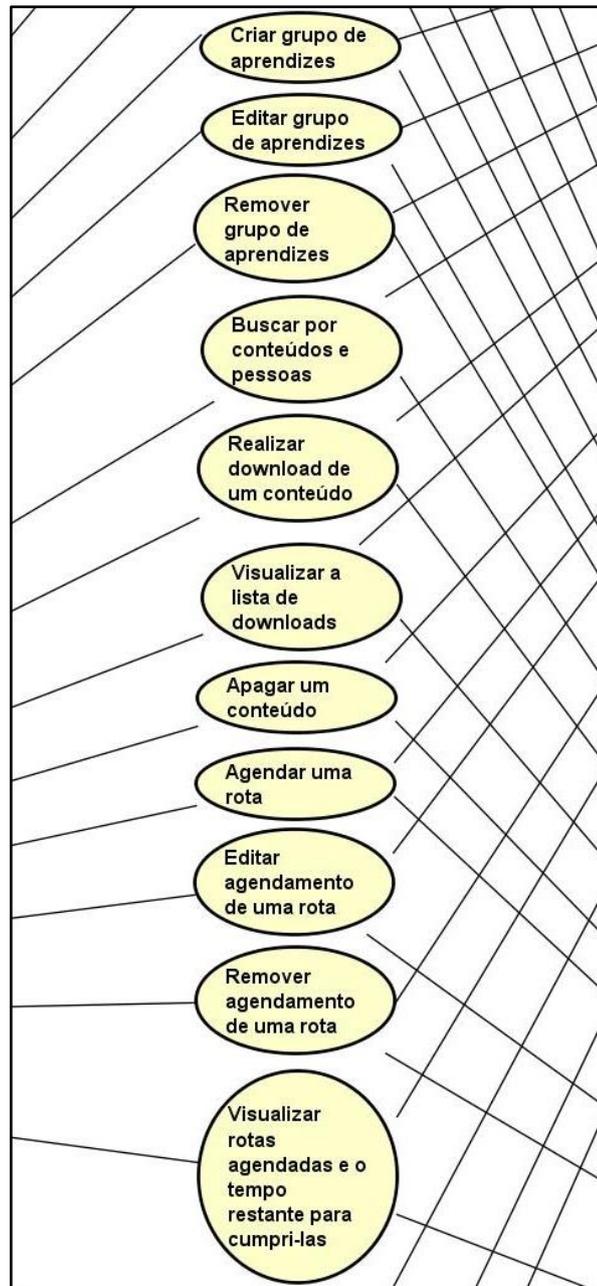
- **Casos de uso para requisitos funcionais de *self-directed learning*:** se encontram os requisitos relacionados ao auto-aprendizado, como podem ser vistos nas Figuras 7, 8 e 9.

**Figura 7:** Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 1.



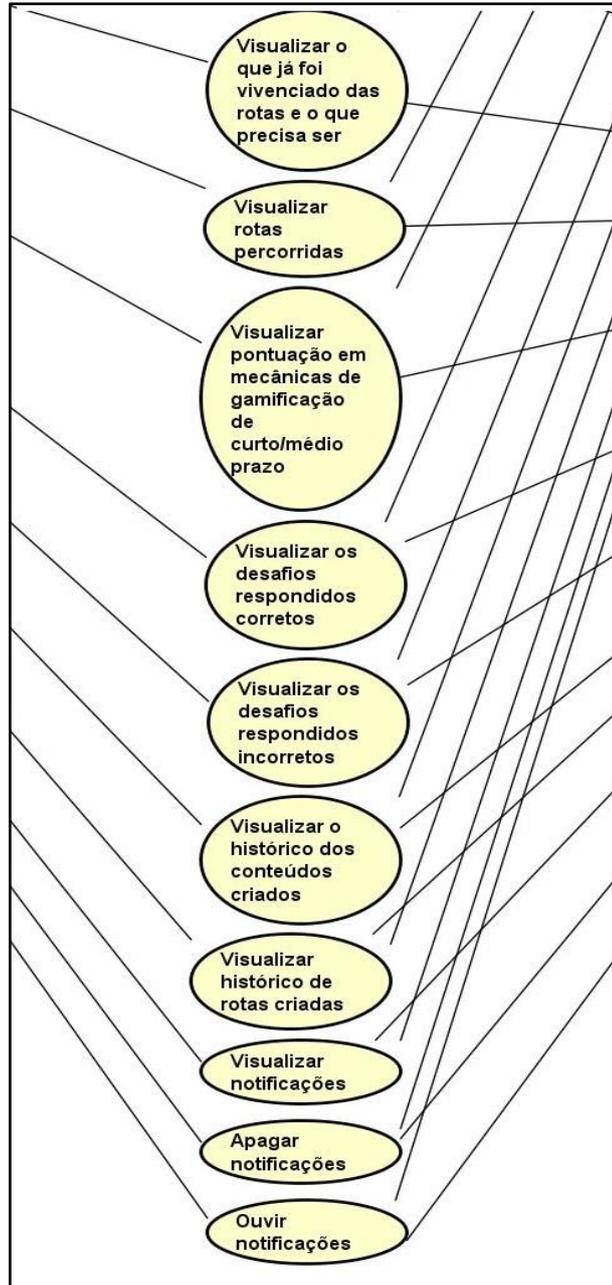
**Fonte:** Própria.

**Figura 8:** Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 2.



**Fonte:** Própria.

**Figura 9:** Casos de uso dos requisitos funcionais de *self-directed learning* - Parte 3.



Fonte: Própria.

#### 5.4. Diagrama de atividade

O diagrama de atividade é um gráfico de fluxo, o qual determina o fluxo dinâmico entre as atividades. Tais diagramas foram criados buscando identificar formas de interação do usuário com a aplicação, de modo que capture a experiência que ele terá quando utilizar o *Youubi*.

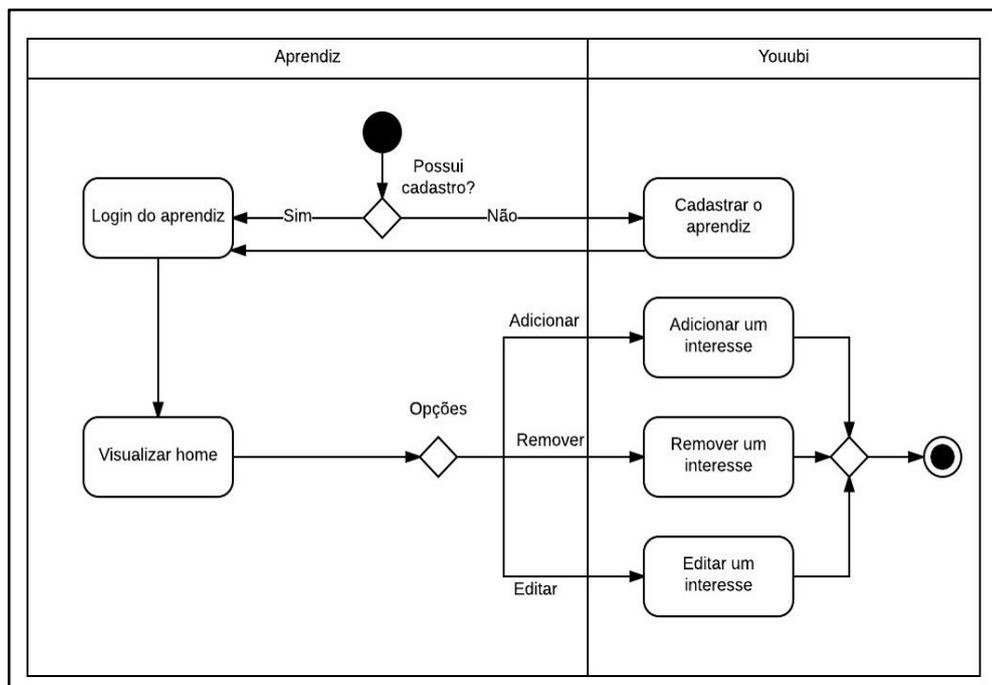
Para obtenção dos resultados foi necessário realizar uma imersão sobre os requisitos, de modo que permitisse visualizar uma aplicação e sua sequência de cliques. Outro fator contribuidor foi visualização de alguns *wireframes* criados anteriormente pelo grupo do *Youubi*, pois possibilitou identificar novas ações e obter os diagramas abaixo.

Dessa forma, através da utilização dos diagramas de atividade buscou-se identificar a experiência que os usuários teriam utilizando a aplicação do *Youubi* e a partir disso analisar formas de melhorar a experiência que o usuário teria manipulando alguns casos de uso. Dessa forma, foi possível identificar como se daria a experiência de manipular interesses, memórias, interesses e desafios.

#### 5.4.1. Experiência de manipular interesses

Para que o usuário defina os seus interesses, foram identificadas as interações que permitam aos usuários selecionarem os interesses de aprendizagem sobre patrimônio escolhido. Dessa forma, obteve-se que o usuário poderá adicionar, remover e editar um interesse de acordo com suas escolhas, apresentados na Figura 10:

**Figura 10:** Diagrama de atividade da manipulação de interesses.

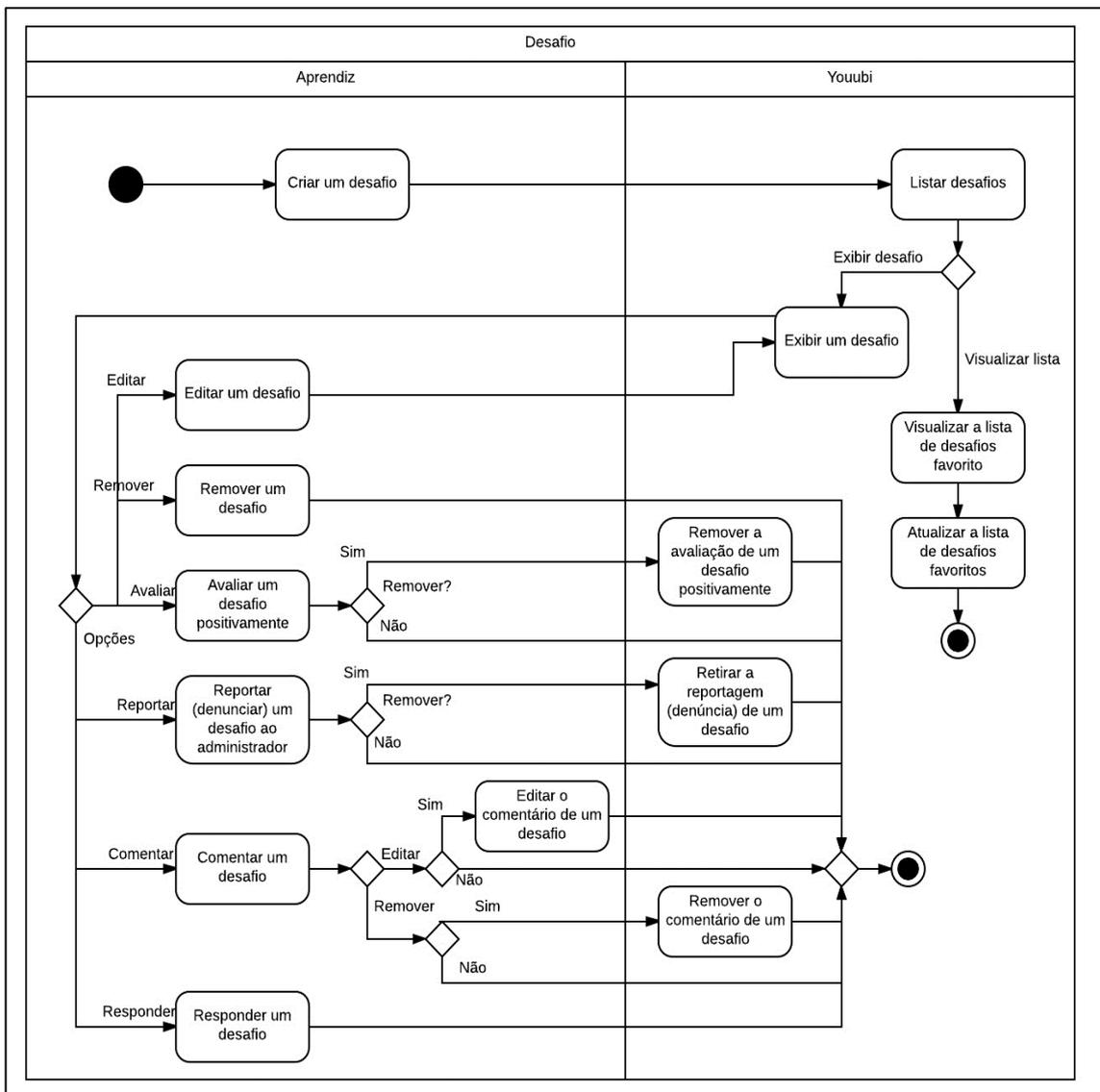


Fonte: Própria.

### 5.4.2. Experiência de manipular desafios

Por fim, pode-se capturar que o usuário poderá realizar as seguintes atividades: criar, editar, remover, avaliar, reportar, comentar e responder um desafio de acordo com sua necessidade e escolha. Além disso, pode-se visualizar e atualizar a lista de desafios favoritos. Dessa forma, obtiveram-se os seguintes resultados modelados, apresentados na Figura 11:

**Figura 11:** Diagrama de atividade da manipulação de desafios.

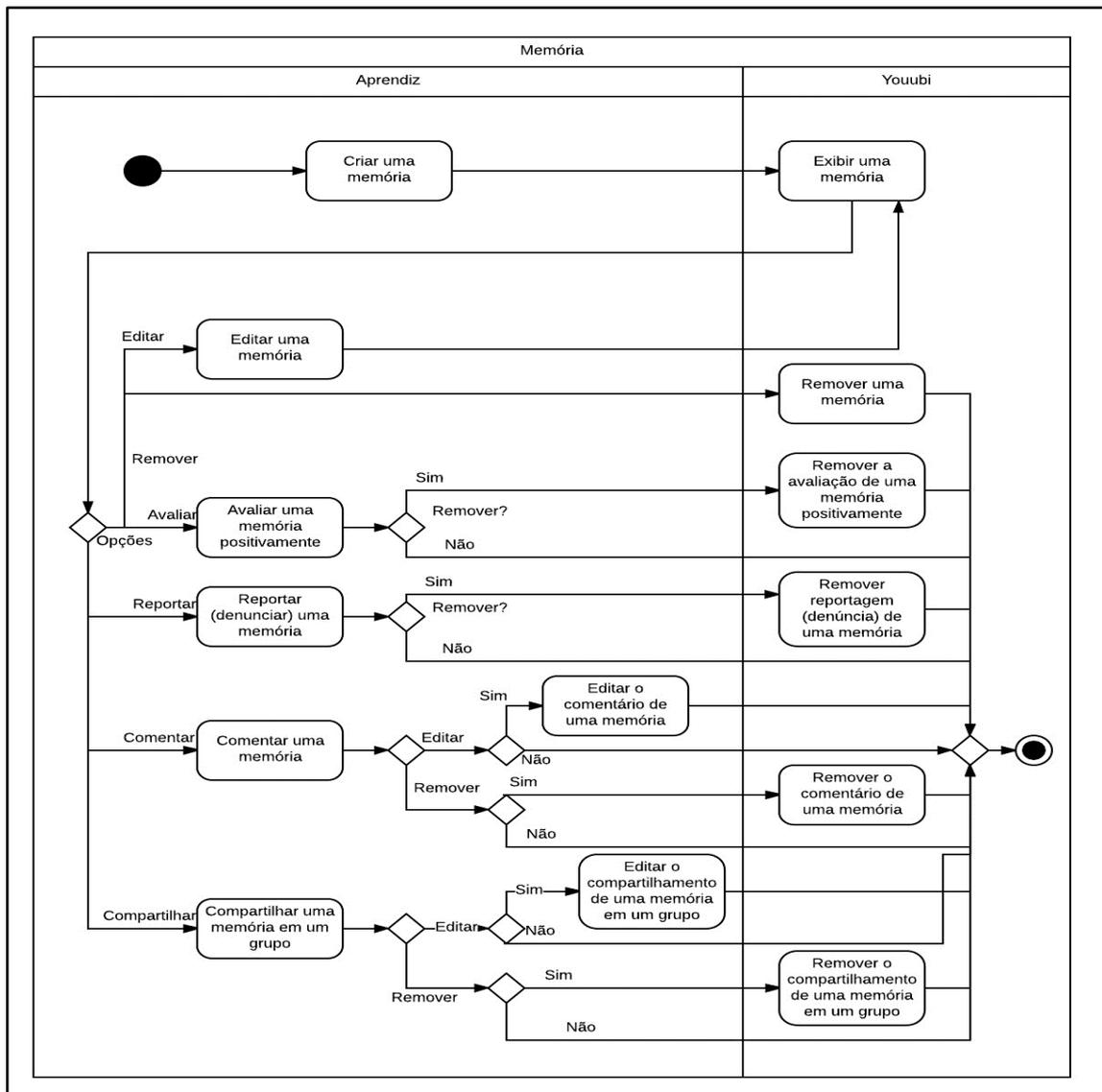


Fonte: Própria.

### 5.4.3. Experiência de manipular memórias

Para criação do diagrama de atividades buscando capturar a experiência do usuário na manipulação das memórias e obteve as seguintes interações: criar, editar, remover, avaliar, reportar, comentar e compartilhar. De modo que resultou nos seguintes resultados, apresentados na Figura 12:

**Figura 12:** Diagrama de atividade da manipulação de memória.



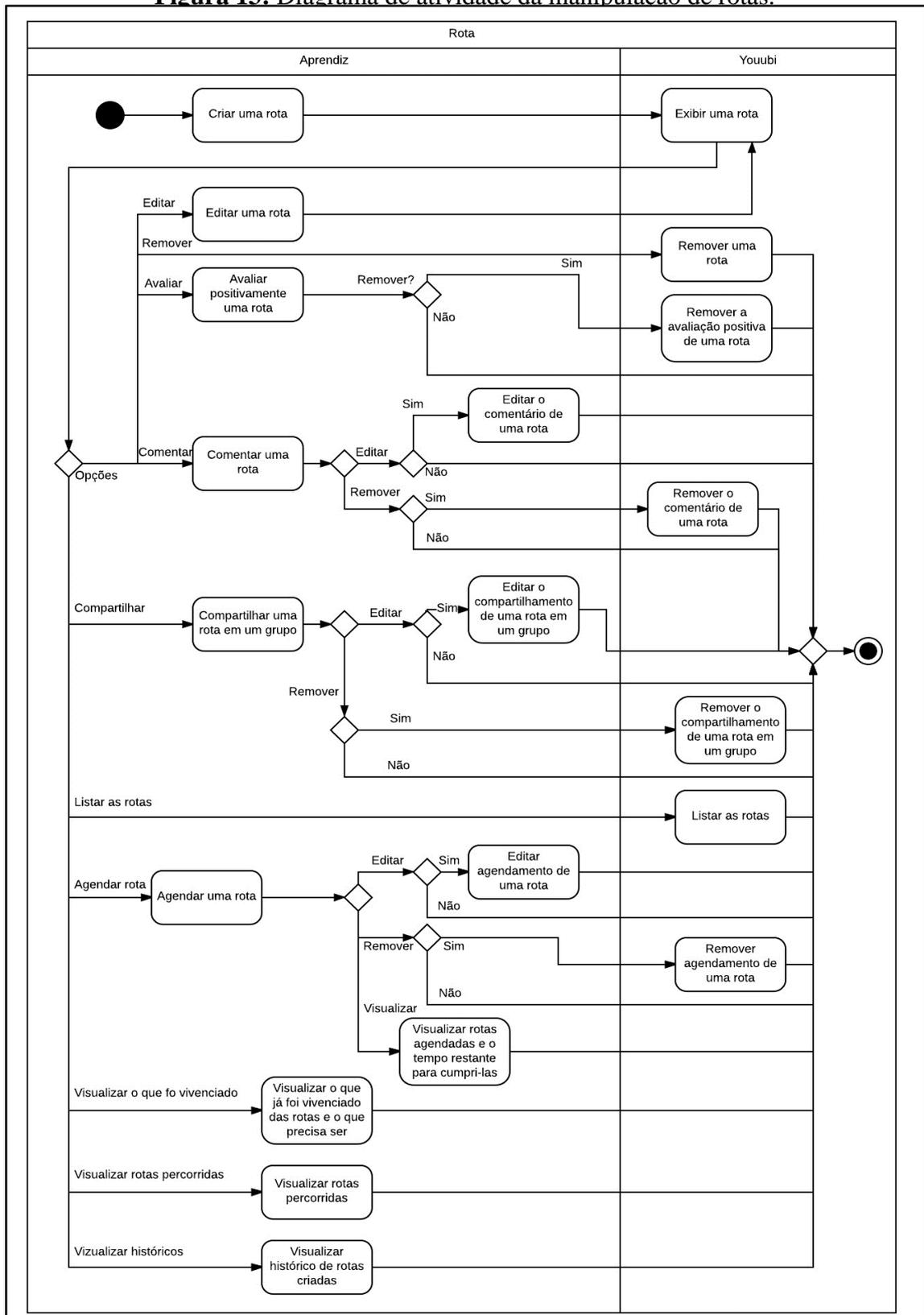
Fonte: Própria.

#### **5.4.4. Experiência de manipular rotas**

Para criação do diagrama de atividades buscando capturar a experiência do usuário na manipulação das rotas obteve-se os seguintes resultados, apresentados na Figura 13.

Dessa maneira, o usuário poderá a partir da criação de uma rota, visualizar, editar, remover, avaliar, comentar, compartilhar, listar, agendar e visualizar o que foi vivenciado, visualizar suas rotas percorridas e visualizar o histórico de rotas criadas.

**Figura 13:** Diagrama de atividade da manipulação de rotas.



Fonte: Própria.

### 5.5. Análise dos resultados

Durante a realização das atividades propostas, percebeu-se a agilidade no desenvolvimento das atividades de especificação do *Youubi*, pelo fato do processo está bem definido, facilitando o entendimento sobre qual o próximo passo a ser realizado. Além disso, observou-se um alinhamento entre as atividades e modelagens, por exemplo, ao realizar a criação dos diagramas de caso de uso, verificou-se a necessidade de incluir novos requisitos para a plataforma, da mesma forma ocorreu com os diagramas de atividades, os quais facilitou o entendimento entre as interações e possíveis fluxos que o usuário poderá realizar quando estiver utilizando a aplicação.

Em suma, o processo proposto conseguiu auxiliar na obtenção da documentação do processo de especificação de requisitos da plataforma *Youubi*, facilitando não apenas durante o levantamento dos requisitos, mas como também na capacidade de perceber novos, bem como realizar a validação para confirmar se o *software* está sendo desenvolvido de acordo com o esperado e requerido.

Portanto, as recomendações podem ser aplicadas ao contexto de outras plataformas ubíquas, de maneira que se obtenham resultados positivos, assim como os obtidos para o presente estudo de caso.

## 6. Discussão

Neste capítulo serão apresentadas algumas discussões sobre como os diagramas obtidos complementam e/ou orientam a realização de outras atividades. Dessa forma, serão apresentados os principais benefícios ou desafios identificados durante a criação dos artefatos.

### 6.1. Como os diagramas complementam o entendimento do funcionamento dos protótipos de baixa fidelidade

Através da proposta de recomendações apresentada na seção 5.1 verificou-se a necessidade de realizar dentre as atividades sugeridas, a criação de alguns artefatos e modelagens importantes para obtenção de uma documentação de especificação de requisitos de qualidade.

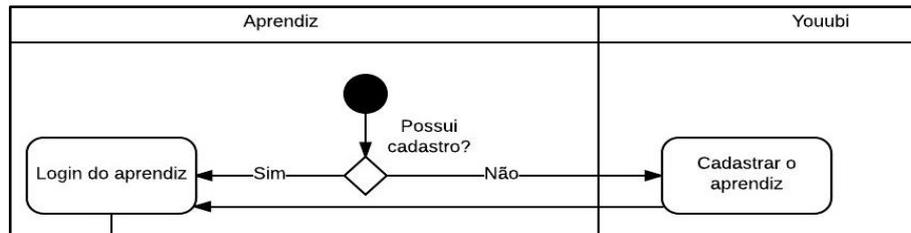
Os diagramas de caso de uso e de atividades desempenham papéis importantes para a especificação de requisitos, conforme descrito nas seções 5.3 e 5.4, eles foram criados com o propósito de capturar de novas informações sobre os cenários e as experiências dos usuários.

Dessa forma, durante a criação das modelagens observou-se que por mais que o documento de especificação de requisitos do *Youubi* abrangesse vários aspectos, visualizar como os requisitos se tornam funcionalidades e *interfaces* é uma tarefa difícil, principalmente estando na fase de concepção. Além disso a criação dos protótipos é de suma importância, já que permite tanto gerar visões de como será a *interface* do sistema, como também identificar as formas de interação que o usuário realizará, conforme foi verificado através da recomendação 6 proposta na seção 5.1.

Portanto, através das modelagens dos diagramas de atividade, pode-se entender como ocorreria funcionamento, a visualização de características que um protótipo deve possuir, as ligações e as interações entre as telas dos protótipos e os possíveis fluxos das atividades e subatividades de uma interação. Tal que comprovou que as informações obtidas através das modelagens são imprescindíveis para completar e capturar o processo de criação e atualização dos protótipos de baixa fidelidade.

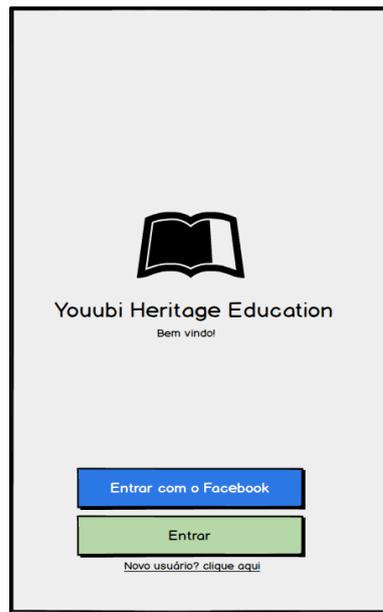
Dessa forma, pode-se observar abaixo através das Figuras 14 e 15, verifica-se uma ligação do diagrama de atividade com os protótipos, demonstrando a maior facilidade no entendimento dos fluxos alternativos que podem acontecer através interação do usuário.

**Figura 14** - Recorte do diagrama de atividade da manipulação de interesses.



**Fonte:** Própria.

**Figura 15** - Tela de login do protótipo de baixa fidelidade do *Youubi*.



**Fonte:** CCTE.

## 6.2. Como os diagramas orientam os métodos de avaliações e evolução dos modelos conceituais

Durante o ciclo de vida de um *software* é de suma importância realizar verificações e validações para identificar a corretude do que está sendo desenvolvido. Diante disso na seção 5.1 observa-se que as recomendações propostas no presente trabalho também se objetivam na criação de artefatos que orientam e validam o *software* que está sendo desenvolvido.

Dessa forma percebeu-se que os artefatos e protótipos auxiliaram na identificação de características erradas, incompletas, de alta ou baixa prioridade de desenvolvimento, reunindo material consistente para avaliações dos possíveis usuários e partes interessadas.

Diante da aplicação da recomendação 7 no contexto do *Youubi* pode-se observar que após a criação dos artefatos e protótipos, principalmente a partir dos diagramas de atividades,

se permitiu verificar e validar as informações obtidas nos passos anteriores, gerando discussões sobre a necessidade, prioridade e ausência de alguns requisitos.

Por exemplo, na figura 15 acima, verifica-se a possibilidade de realizar o *login* com o *Facebook*, entretanto a equipe de desenvolvimento da aplicação *Youubi*, analisou a viabilidade da criação dessa funcionalidade e concluiu que o requisito “*Login do aprendiz*” deveria ser modificado, permitindo apenas que o usuário efetue o *login* a partir do e-mail e senha. Dessa forma, realizou a atualização dos protótipo e do DDR, resultando no requisito indicado através da Figura 16:

**Figura 16** - Requisito atualizado no DDR - *Login do aprendiz*.

<p><b>2.1 Login do aprendiz</b> O login pode ser realizado quando o aprendiz precisa informar o seu e-mail e uma senha [RF 2.2]. <b>Fluxo alternativo:</b> não existe fluxo alternativo <b>Telas:</b> <b>Tempo (em horas):</b> <b>Prioridade:</b> [ X ] Essencial [ ] Importante [ ] Desejável <b>Dependências:</b> RF-2.3 <b>Youubi API:</b> login</p>
---

**Fonte:** Própria.

### 6.3. Como a combinação do uso de UML e *wireframe* ampliar a capacidade de perceber requisitos

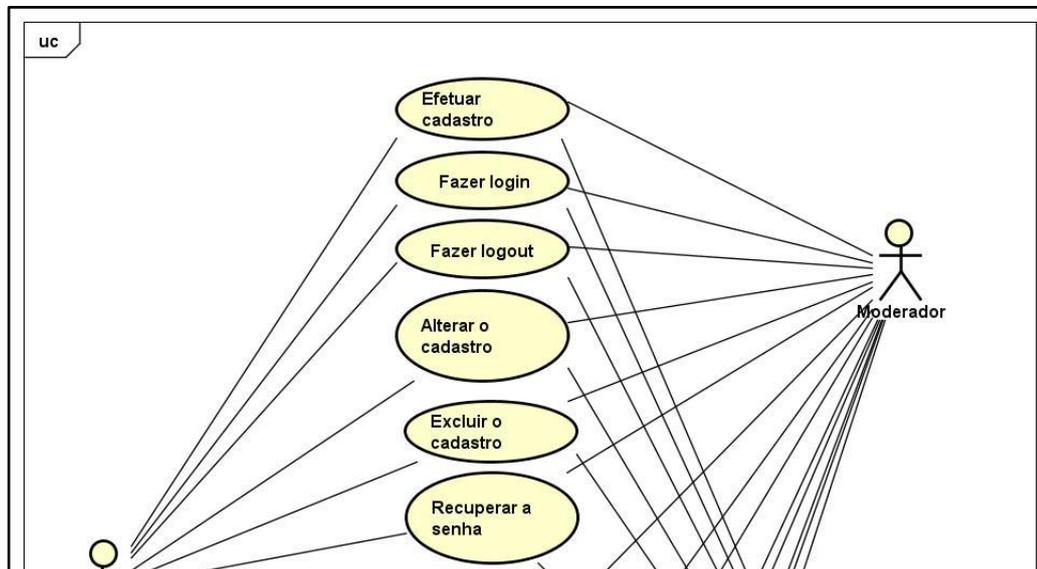
Diante da necessidade de capturar informações para o desenvolvimento, o diagrama de caso de uso permitiu a visualização do cenário da aplicação, destacando o relacionamento entre os outros casos de uso e os atores. Além disso, pode-se identificar a prioridade entre os requisitos e também observar necessidade de incluir alguns outros, para o desenvolvimento das telas previstas nos *wireframes*.

Dessa forma, analisando os diagramas e *wireframes*, podem-se perceber os possíveis cliques e requisitos que geram funcionalidades para uma determinada tela que não havia sido pensada anteriormente.

Na figura 17 abaixo aponta a possibilidade de criação de um requisito de *software*, pois observou-se a necessidade do envio de notificações ou e-mail quando o usuário realizasse determinadas ações com o *Youubi*. Essas notificações, e-mail ou mensagem de texto poderiam ser disparadas quando, por exemplo, um aprendiz realizasse o cadastro ou recuperação de

senha, informando o procedimento, senha alternativa de acesso e/ou mensagem de boas vindas.

**Figura 17** - Possibilidade de incluir o requisito “*Enviar notificação*”.



Fonte: Própria.

#### 6.4. A combinação das técnicas de UML e *wireframe* no processo de concepção

Assim como a combinação das técnicas de UML e *wireframes* para captar novos requisitos, também pode-se identificar uma relação importante para o processo de concepção, contribuindo para a descrição e organização das funções e requisitos chaves do *Youubi*.

Dessa forma, através dos diagramas de casos de uso, percebeu-se quais os requisitos que exigiram mais cuidado e atenção durante as interações. Permitiu-se também a possibilidade de gerar uma listagem com as funcionalidades extraídas do documento de requisitos para um determinado cenário, bem como analisar gerar formas visuais e criativas de telas/*wireframe* do *Youubi* durante o processo de concepção.

O exercício de concepção através de protótipos é muito importante, pois permite que a aplicação seja testada antes do desenvolvimento do código fonte, reduzindo um possível retrabalho e erro, já que modificações podem ser realizadas mais facilmente, sem gerar maiores custos. Portanto, através da união dos diagramas UML e *wireframes* do *Youubi* possibilitou o cruzamento das informações e gerar uma maior sincronia do que está sendo desenvolvido nos protótipos com a documentação do *software*.

Dessa forma, também permitiu capturar a experiência ubíqua que um usuário teria ao interagir com a aplicação Youubi, de maneira que permitiu validar e verificar se o *software* que está sendo desenvolvido precisa ou não de ajustes.

## **7. Considerações Finais**

Neste capítulo serão apresentadas as considerações finais, destacando pontos importantes observados durante a realização do presente trabalho, bem como as possibilidades de trabalhos futuros.

### **7.1. Conclusão**

Com a realização deste trabalho ficou bastante claro que pelo fato da computação ubíqua ser um novo paradigma da computação, ainda existe muitos desafios para o desenvolvimento de *softwares*, como puderam ser visto na seção 3.3.

Dessa forma, ficou constatado que antes de iniciar o desenvolvimento de um *software* ubíquo, as partes envolvidas devem está ciente do contexto, como também entender suas características e os desafios que estão atrelados a elas e que podem ser enfrentados. Uma das principais preocupações identificadas foi a de permitir que o *software* criado seja capaz de ser sensível ao contexto e garanta a interoperabilidade entre as tecnologias.

Além disso, a literatura possui poucas abordagens sobre o processo de especificação de requisitos e criar uma proposta do processo de especificação de requisitos representou um grande desafio, já que é um contexto novo. Dessa forma, a partir das atividades foram utilizadas algumas para a criação e obtenção dos resultados aplicados ao estudo de caso, o *Youubi Heritage Education*, uma plataforma ubíqua de aprendizagem que está sendo desenvolvida e precisa de uma atenção especial nas fases iniciais de concepção.

Diante disso, pode-se criar artefatos que permitiram identificar discussões sobre quais os benefícios e desafios foram proporcionados para as atividades de desenvolvimento de *software* ubíquo.

Contudo, realizar foi bastante desafiador e satisfatório realizar o desenvolvido do presente trabalho.

### **7.2. Trabalhos Futuros**

Mesmo conseguindo atingir aos objetivos propostos por essa pesquisa, com o foco de realizar e identificar o processo de especificação de requisitos da experiência de aprendizagem em plataformas ubíquas tem como trabalhos futuros as seguintes sugestões:

- Aplicar as atividades como um todo no desenvolvimento de outro cenário de *software* ubíquo;
- Criar novos artefatos para que possam ser identificados outros olhares sobre o processo de especificação de requisitos para obter a experiência ubíqua.

## Referências

- ABOWD, G. D. Software Engineering Issues for Ubiquitous Computing. Software Engineering, 1999. Proceedings of the 1999 International Conference. Los Angeles, USA, 1999.
- ARAUJO, R. B. de. Computação Ubíqua: Princípios, Tecnologias e Desafios. Disponível em: <[http://professordiovani.com.br/rw/monografia\\_araujo.pdf](http://professordiovani.com.br/rw/monografia_araujo.pdf)>. Data de Acesso: 15 de Setembro de 2017.
- BARBOSA, J.; HAHN, R.; BARBOSA, D. N. F.; GEYER, C. F. R. Mobile and Ubiquitous Computing in an Innovative Undergraduate Course, SIGCSE'07, Covington, p. 379-383, 2007.
- BOOCH, B.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Elsevier Brasil, 2006.
- C. A. da COSTA, A. C. YAMIN, AND C. F. R. GEYER, "Toward a general software infrastructure for ubiquitous computing,"IEEE Pervasive Computing, vol. 7, no. 1, pp. 64–73, 2008.
- CARVALHO, S. T. DE; COPETTI, A.; LOQUES FILHO, O. G.. Sistema de computação ubíqua na assistência domiciliar à saúde. J. Health Inform, 2011.
- CHIU, C.; TSENG, J. C. R.; HSU, T. Blended context-aware ubiquitous learning in museums: environment, navigation support and system development. Personal and Ubiquitous Computing (2017). pp 355–369, 2017.
- DICK, J.; HULL, E; JACKSON, K. Requirements Engineering. Springer, p. 239, 2017.
- GALENO, A dos S. Concepção de módulo para dispositivos móveis de gestão da aprendizagem pessoal integrado ao sistema de gestão da aprendizagem Amadeus. Dissertação de mestrado - UFPE, 2010.
- GOMES, A. S.; BRITO, J. A.; AMORIM, R. Effectiveness of educational practices through the extension of a framework ubiquitous learning environment. Information Systems and Technologies (CISTI), 2015 10th Iberian Conference, Aveiro, Portugal, 2015.
- HONG, D.; CHIU, D. K. W; SHEN, V. Y.; Requirements Elicitation for the Design of Context-aware Applications in a Ubiquitous Environment. ICEC '05 Proceedings of the 7th international conference on Electronic commerce, p. 590-596, Xi'an, China, 2005.
- IEEE. IEEE Std 610.12 – 1990. IEEE Standard Glossary of Software Engineering Terminology, 1990.

IEEE. IEEE Std 830 – 1998. IEEE Recommended Practice for Software Requirements Specifications, 1998.

JONES, V.; JO, J. H. Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology. Disponível em: <<https://www.ascilite.org/conferences/perth04/procs/jones.html>>. Data de acesso: 21 de Outubro de 2017.

JORGENSEN, J. B., BOSSEN, C. “Requirements engineering for a pervasive health care system”. Proceedings of. 11th IEEE International Requirements Engineering Conference, p. 55-64, 2003.

KIM, C.; MIRUSMONOV, M. Factors Influencing the Usage of Ubiquitous-Based Mobile Devices. Computing Technology and Information Management (ICCM), 8th International Conference, Seoul, South Korea, 2012.

KINDBERG, T.; FOX, A. System Software for Ubiquitous Computing, IEEE Pervasive Computing archive. v. 1, p. 70-81, 2002.

LIMA, T. C. S. DE; MIOTO, R. C. T. Procedimentos metodológicos na construção do conhecimento científico: a pesquisa bibliográfica. Rev. Katál. Florianópolis v. 10 n. esp. p. 37-45 2007.

LOUREIRO, A. A. F.; OLIVEIRA, R. A. R.; SILVA, T. R. DE M. B.; PIRES JÚNIOR, W. R.; OLIVEIRA, L. B. R. DE; MOREIRA, R. A.; SIQUEIRA, R. G.; ROCHA, B. P. S.; RUIZ, L. B.. Computação Ubíqua Ciente de Contexto: Desafios e Tendências. 27o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC 2009, p.99 - 149, Minicurso, 2009.

MONTEIRO, B. de S. Ambiente de aprendizado ubíquo youubi: design e avaliação. Tese de Doutorado - UFPE 2015.

MOTA, L. da S. Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software. 2013. 149 p. Dissertação (Mestrado) - UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação - Rio de Janeiro, 2013.

OLIVEIRA, K. E. de J.; PORTO, C. de M.; ALVES, A. L.. Dispositivos Digitais E Aprendizagem Ubíqua: Um Experimento Com A Inversão De Sala De Aula. Interfaces Científicas - Educação, Aracaju, v. 5, n.2, p. 53 - 62, 2017.

PINTO, F. C. do R. “Uma Abordagem Para Apoiar Especificação De Requisitos Para Projetos De Software Ubíquo”. Dissertação de Mestrado. COPPE/UFRJ. Rio de Janeiro, 2009.

PRESSMAN, R. S.; MAXIM, B. R.. Software Engineering A Practitioner’s Approach, 8ª ed. 59

McGraw-Hill Education, 2015.

PRESSMAN, Roger. Engenharia de Software. 7.ed. Porto Alegre: AMGH, 2011.

RANGA, K. K. Ubiquitous Computing: Fastest Emerging Technology. Computing for Sustainable Global Development (INDIACom), 2nd International Conference, New Delhi, India. 2015.

ROCHA, L. S.; FERREIRA F., J. B.; LIMA, F. F. P.; MAIA, M. E. F.; VIANA, W.; CASTRO, M. F. DE; R. M. C. ANDRADE. Ubiquitous Software Engineering: Achievements, Challenges and Beyond. Brazilian Symposium on Software Engineering. SBES, p. 132-137, 2011.

ROUILLARD, J.; LAROUSSE, M. PerZoovasive: contextual pervasive QR codes as tool to provide an adaptive learning support. Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology, França, 2008.

SAKAMURA, K., KOSHIZUKA, N. Wireless and Mobile Technologies in Education. WMTE 2005. IEEE International Workshop, 2005.

SANTAELLA, L. Desafios da ubiquidade para a educação. Novas mídias e o Ensino Superior - Revista Ensino Superior Unicamp, 2013.

SATYANARAYANAN, M.. "Pervasive computing: Vision and challenges," IEEE Personal Communications, vol. 8, no. 4, pp. 10 –17, August 2001.

SOMMERVILLE, I. Engenharia de Software. Pearson Prentice Hall, 2011.

SOUZA, H. V. L. de. Guidelines para implantação de processos híbridos de software: Uma proposta para a definição de um ciclo de vida com um padrão ISO e metodologia ágeis. Dissertação de mestrado - UFPE, 2012.

SPÍNOLA, R. O., SILVA, J. L. M, TRAVASSOS, G. H. "Towards a Conceptual Framework to Classify Ubiquitous Software Projects". Proceedings of the 8th International Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, USA, 2006.

SPINOLA, R. O.; Travassos, G. H.; MASSOLLAR, J. Checklist to Characterize Ubiquitous Software Projects. SBES 2007 - XXI Simpósio Brasileiro de Engenharia de Software, 2007.

SPÍNOLA, R.O. (2010). "Apoio à Especificação e Verificação de Requisitos Funcionais de Ubiquidade em Projetos de Software". Tese de Doutorado. COPPE/UFRJ. Rio de Janeiro.

TEIXEIRA, A. Bases Da Teoria Lógica De Dewey. Revista Brasileira De Estudos Pedagógicos - Instituto Nacional de Estudos Pedagógicos, Ministério da Educação e Cultura. v. XXIII, n. 57, 1955.

VARELA, F. J.; THOMPSON, E.; ROSCH, E. A Mente Incorporada: Ciências Cognitivas e

Experiência Humana. Artmed, 2003.

VIANA, W.; R. M. C. ANDRADE. Uma proposta para a geração semi-automática de aplicações adaptativas para dispositivos móveis. XX Simpósio Brasileiro de Engenharia de Software, 2005.

WEISER, M. The computer for the 21st century. ACM SIGMOBILE mobile computing and communications review, vol 3(3). ACM Press, New York, p. 3–11, 1999.

YAHYA, S.; AHMAD, E. A.; JALIL, K. A.. International Journal of Education and Development using Information and Communication Technology; Bridgetown v. 6, i. 1, p. 11, 2010.