

Um Método para Modelagem de Requisitos não Funcionais em *Ambient Assisted Living*

Maurício Manoel Coelho Júnior¹, Willy Silva Coutinho¹, Robson Lima de Alencar¹, and Fernanda Alencar¹

Universidade Federal de Pernambuco, Departamento de Eletrônica e Sistemas,
50.740-560, Recife, Pernambuco, Brasil
{mauricio.coelhojr,willy.coutinho,robson.limaalencar,fernanda.ralencar}@ufpe.br

Resumo O aumento da expectativa de vida da população mundial é uma realidade, e quando combinado a quedas acentuadas e taxas de fertilidade, esses avanços na expectativa de vida podem levar a um rápido envelhecimento da população em todo o mundo. Tecnologias como *Ambient Assisted Living* (AAL) fornecem serviços que permitem às pessoas idosas viver uma vida independente, segura e saudável. De forma a atender às necessidades do idoso, faz-se necessário superar as percepções de riscos, incertezas relacionadas à tecnologia, usabilidade e contexto de uso, itens que exigem o cumprimento de diversos requisitos não funcionais (*Non Functional Requirements* - NFR). Nesse contexto, este artigo apresenta um método para auxiliar na modelagem de requisitos não funcionais no contexto de AAL.

Keywords: Requisitos não Funcionais · Ambientes de Vida Assistida · AAL · Modelagem · NFR Framework

1 Introdução

Segundo a Organização Mundial de Saúde (OMS) o número de pessoas com idade superior a 60 anos chegará a 2 bilhões de pessoas até 2050 [20], representando um quinto da população mundial. Segundo projeções do IBGE [14] a partir de 2040, 17% da população brasileira será de idosos. Na perspectiva de tecnologias direcionada a este público, pode-se destacar Ambientes de Vida Assistida (*Ambient Assisted Living* - AAL) cujo objetivo é melhorar a qualidade de vida dos idosos, pacientes com condições crônicas e pessoas com deficiência, oferecendo autonomia/independência, conforto e segurança [7].

Do ponto de vista de soluções em AAL destinadas ao público idoso é importante garantir que os *NFRs* sejam modelados, especificados e validados durante todo o ciclo de desenvolvimento. Assim, é apresentado um método para auxiliar na modelagem de requisitos não funcionais no contexto de AAL, como uma extensão no *NFR Framework* [9]. Este artigo é organizada da seguinte maneira: A Seção 2 apresenta a base conceitual para o desenvolvimento deste artigo e relaciona as principais conceitos: Engenharia de Requisitos, *Ambient Assisted Living* e *NFR Framework*. Na Seção 3 apresenta-se uma Revisão Sistemática

da Literatura (RSL) com 22 trabalhos onde é possível observar os principais NFR em AAL e propostas para modelagem de *NFRs* em AAL. Na Seção 4 é apresentado o método proposto e um estudo de caso na área da saúde; e por fim, a Seção 6 conclui o artigo e apresenta sugestões para trabalhos futuros.

2 Fundamentação Teórica

2.1 Engenharia de Requisitos

De acordo com Abbott [5], um requisito pode ser uma função, uma restrição ou outra característica que deve ser fornecida ou atendida para satisfazer as necessidades do usuário do sistema que será desenvolvido. Já Sommerville [22] define que os requisitos de um sistema é uma descrição do que o sistema deve fazer, os serviços que ele fornece, e as suas restrições. O autor ainda classifica os requisitos de três maneiras: requisitos funcionais, requisitos não funcionais e de domínio, porém muitos autores preferem classificá-los em dois principais tipos: (i) Requisitos funcionais: Como o sistema deve reagir às entradas e como deve se comportar em algumas situações. Aplicam-se a funções específicas do sistema. (ii) Requisitos não-funcionais: São restrições das funções e serviços oferecidos pelo sistema. Podem ser restrições de tempo, de desenvolvimento, de processos e por normas, aplicando-se ao sistema com um todo.

2.2 Ambient Assisted Living

O conceito de AAL está relacionado com um conjunto de produtos, serviços e sistemas que procuram manter a qualidade de vida e independência da população sênior e de indivíduos com necessidades de cuidados específicos [13]. Para Almeida [6], AAL é um sistema crítico, e um dos fatores de sucesso para esses sistemas, são uma boa modelagem de requisitos não-funcionais [16]. Uma classificação dos sistemas AAL foi proposto no projeto BRAID [1] focando em quatro subdomínios: (i) Vida independente: auxilia as atividades da vida diária (por exemplo, lembretes médicos, monitoramento do estado de vida) e oferece suporte à mobilidade das pessoas (por exemplo, assistência para compras, cadeiras de rodas inteligentes); (ii) Saúde e Cuidados na Vida: auxilia os pacientes em atividades relacionadas à saúde, por exemplo, monitoramento remoto da saúde, assistência de emergência, assistência de exercícios físicos; (iii) Ocupação na Vida: auxilia os idosos na continuidade de suas atividades profissionais; e (iv) Recreação na Vida: facilita a socialização e a participação dos idosos em atividades sociais, culturais e de lazer.

2.3 NFR Framework

O *NFR Framework* foi proposto por Mylopoulos, Chung e Nixon [19]; e desenvolvido em Chung et al. [8]. O *NFR Framework* concentra-se na modelagem e análise de requisitos não-funcionais, com objetivo de colocar os *NFRs* mais importantes na mente do desenvolvedor [8]. O Framework oferece uma estrutura

para representar e registrar o processo de *design* e raciocínio em gráficos, chamados de *Softgoal Interdependency Graphs* (SIGs), que ajuda os desenvolvedores a produzir soluções personalizadas, considerando-se as características do domínio e do sistema a ser desenvolvido. Estas características, incluindo os requisitos não-funcionais, requisitos funcionais, prioridades e carga de trabalho, influenciam a escolha do desenvolvimento de alternativas para um sistema específico. Para lidar com o grande número de possíveis alternativas de desenvolvimento, os desenvolvedores podem consultar os catálogos. Estes catálogos organizam experiências do passado, técnicas padrões, e conhecimento sobre *NFRs* particulares com suas interdependências [10]. Chung apresenta várias etapas principais do seu processo, que não são necessariamente sequenciais podendo ser necessário iterá-las várias vezes durante o processo de *design*. Como a proposta é para uso de domínio geral, não foi possível encontrar no *NFR Framework*: (i) uma etapa para elicitação de requisitos, onde seja possível capturar também o contexto de uso e as preferências do usuário, fatores importantes para os sistemas AAL; (ii) Na etapa "identificar *NFRs* específicos" do *NFR Framework*, deve ser considerada um suporte de outros artefatos como normas legais ou taxonomia AAL, que devem ser apoiadas por um catálogo ou base de dados que seja transversal em todo o processo de *design*; (iii) Como representar o impacto do contexto de uso e preferências do usuário no SIG. Esses pontos foram resolvidos no método proposto.

3 Revisão Sistemática da Literatura - RSL

Para apoiar esta proposta, foi realizado uma RSL, com o objetivo de identificar os trabalhos relacionados em modelagem e especificação de *NFRs* para AAL. Os arquivos produzidos nesta RSL estão disponíveis em um repositório online [3]. Foi adotado o processo descrito por Mian et al. [18], segundo a qual estabelecem três fases do processo de condução: (i) Planejamento da revisão, (ii) execução da revisão e (iii) análise e divulgação dos resultados. A construção dos termos de busca foi realizada baseada nas seguintes etapas: (i) A partir das variáveis independentes e dependentes foram identificados as principais palavras-chave; (ii) Sinônimos são identificados com base em artigos conhecidos e relevantes na área de pesquisa; e (iii) A *string* de pesquisa é gerada a partir da combinação dessas palavras-chave e sinônimos (**Ambient Assisted Living, Requisitos não Funcionais, Especificação e Modelagem**), conforme protocolo disponível online [3].

3.1 Fase de execução

Nessa etapa, foi realizada a condução e análise dos estudos primários no período compreendido entre 01/02/2021 e 15/02/2021 com o total de 1071 trabalhos. Foram aplicados os critérios de inclusão e exclusão definidos no protocolo [3]. A *string* de busca (Tabela 1) foi aplicada nos repositórios: IEEE Xplorer (1), ACM Digital Library (77), Sciencedirect (487), PMC (107), Engineering Village (10), PubMed (24), Springer (352), Scopus (13), e os resultados passaram por cinco etapas de seleção, conforme protocolo disponível online [3].

Tabela 1. String de Busca

```

(("AAL"OR "Ambient Assisted Living"OR "ambient assisted"OR "ambient assistance"OR "assisted environment"OR "assistive environment"OR "AAL environment"OR "independent living"OR "assisted life"OR "intelligent living"OR "pervasive living"OR "assistive environments"OR "AAL environments"OR "assisted environments"OR "Assistive Software") AND ("quality model"OR "quality attribute"OR "non-functional property"OR "Non-Functional Requirements"OR "quality requirement"OR "quality models"OR "quality attributes"OR "non-functional properties"OR "non-functional requirements"OR "quality requirements"OR "NFR") AND ("specification"OR "model*"))

```

3.2 Fase de análise e divulgação dos resultados

Nesta fase, os estudos primários que atendiam ao propósito da RSL foram analisados criticamente e sintetizados no formulário de aprovação dos trabalhos que preencheram os critérios de inclusão. Apresenta-se uma visão das características gerais dos estudos, bem como os resultados das questões de pesquisa [21]. Os gráficos foram extraídos a partir do *Dashboard* [4] criado para a RSL. A Tabela 2 apresenta os estudos selecionados após a fase de avaliação de qualidade. Foi disponibilizado um repositório online [3] com todos os artigos em suas respectivas fases.

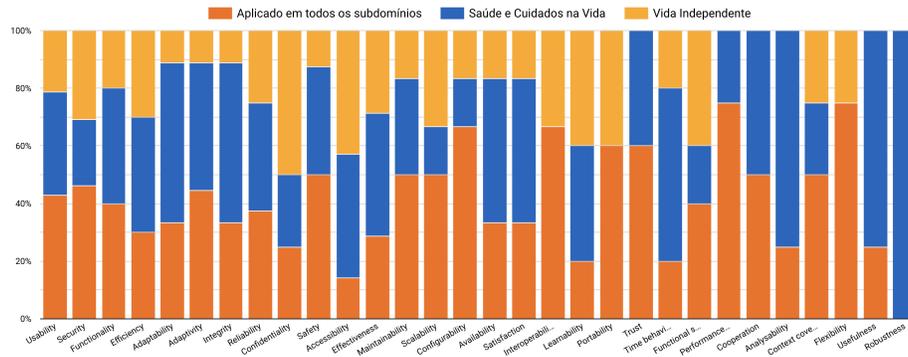
RQ1 – Quais os subdomínios de AAL que o estudo suporta? O resultado desta questão de pesquisa é importante para identificar os diferentes domínios propostos para a abordagem AAL. A partir dos resultados, Saúde e Cuidado na Vida apresenta 40,90%, dos trabalhos, seguido por Vida Independente com 30,56%. Um dado interessante é que 22,70% dos trabalhos que podem ser aplicados em todos os subdomínios, com destaque para o trabalho EP_12 que define padrões e um processo formal de decomposição dos requisitos do sistema, de modo a obter requisitos básicos para uma validação composicional do sistema.

RQ2 – Quais os requisitos não funcionais identificados? A Figura 1 apresenta uma relação entre os *NFRs* [15] e subdomínios de AAL presentes nos estudos. É possível notar que todos os *NFRs* estão presentes em algum dos estudos, independente de domínio, exceto o *NFR Robustness*, presente apenas em trabalhos na área da Saúde e Cuidado na vida. Para este subdomínio, podem-se destacar os *NFRs Usability, Adaptability, Integrity, Usefulness e Integrity*, que dentro da ISO 25010 [15] são classificadas como qualidade do produto. Destacam-se os trabalhos EP_01, EP_10, EP_17 e EP_20 que apresentam características de qualidade para AAL.

RQ3 - Quais técnicas / métodos são usados para apoiar a modelagem e/ou especificação de NFRs em AAL? Foi identificado que 45,50% dos estudos utilizam *Unified Modeling Language (UML)*, como o trabalho EP_15,

Tabela 2. Trabalhos Seleccionados

ID	Trabalho	Autor	Ano	Fonte
EP_01	A Quality Model for AAL Software Systems	Lina Garcés et al.	2016	IEEE
EP_02	Data and Information Quality Issues in Ambient Assisted Living Systems	McNaull et al.	2012	ACM
EP_03	Patterns for Identification of Trust Concerns and Specification of Trustworthiness Requirements	Mohammadi, Nazila Gol and Heisel, Maritta	2016	ACM
EP_04	Exploratory Study on the Use of Software Product Lines in the Development of Quality Assistive Technology Software	Martins et al.	2018	ACM
EP_05	Applying model-driven engineering to a method for systematic treatment of NFRs in Aml systems	Ruiz-López et al.	2013	Scopus
EP_06	Runtime models based on dynamic decision networks: Enhancing the decision-making in the domain of ambient assisted living applications	García-Panear, Luis-Hernán and Bencomo, Nelly	2016	Scopus
EP_07	A semantic approach for designing Assistive Software Recommender systems	Elena et al.	2015	Scienccdirect
EP_08	Activities of Daily Living Ontology for Ubiquitous Systems: Development and Evaluation	Woznowski et al.	2018	PMC
EP_09	The SmartHabit: An Intelligent Privacy-Aware Home Care Assistance System	Grgurić et al.	2019	PMC
EP_10	Quality attributes and quality models for ambient assisted living software systems: A systematic mapping	Lina Garcés et al.	2017	Scienccdirect
EP_11	Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes	Memon et al.	2014	PMC
EP_12	Enabling correct design and formal analysis of Ambient Assisted Living systems	Kawtar Benghazi et al.	2012	Scienccdirect
EP_13	A tailored smart home for dementia care	Amiribesheli, Mohsen and Bouchachia, Hamid	2018	Springer
EP_14	Using RELAX, SysML and KAOS for Ambient Systems Requirements Modeling	Manzoor Ahmad et al.	2012	Scienccdirect
EP_15	REUBI: A Requirements Engineering method for ubiquitous systems	Tomás Ruiz-López et al.	2013	Scienccdirect
EP_16	Development and evaluation of SOA-based AAL services in real-life environments: A case study and lessons learned	Erlend Stav et al.	2013	Scienccdirect
EP_17	A Quality Model for the Evaluation AAL Systems	Madjid Kara et al.	2017	Scienccdirect
EP_18	Quality Parameters as Modeling Language Abstractions for Context-Aware Applications: An AAL Case Study	Hoyos et al.	2017	Springer
EP_19	Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces	Kleinberger et al.	2007	Springer
EP_20	Elicitation of Quality Characteristics for AAL Systems and Services	Omerovic et al.	2013	Springer
EP_21	Quality Assessment Approaches for Ambient Assisted Living Systems: A Systematic Review	Erazo-Garzon et al.	2020	Springer
EP_22	Ambient Assisted living system's models and architectures: A survey of the state of the art	Amina et al.	2020	Scienccdirect

**Figura 1.** Requisitos não Funcionais x Subdomínios AAL

que introduz um método de engenharia de requisitos para a análise de Sistemas Onipresentes, baseado em metas que representa a influência do contexto e das situações adversas. Já o trabalho EP_16, permite uma derivação semi-automática de modelos de *design* de *software* de qualidade a partir de uma especificação de *NFRs* do sistema, com foco em arquiteturas orientadas a agentes. Já *Goal-Oriented Requirements Engineering* (GORE) esta presente em 31,80% dos estudos, destacando-se o estudo EP_14 que apresenta uma linguagem de

Engenharia de Requisitos para sistemas adaptativos, com a possibilidade de introduzir flexibilidade nos *NFRs* para se adaptarem a quaisquer mudanças nas condições ambientais.

RQ4 - Quais ferramentas são utilizadas na modelagem e/ou especificação de NFRs? Foi identificado que 27,30% dos estudos não utilizam nenhuma ferramenta para auxiliar na construção de sua proposta. Já 22,7% dos trabalhos, utilizam ferramentas *desktop* (EP_06, EP_12, EP_15, EP_04, EP_18), que pode ser um problema para modelagem e/ou especificação de *NFRs* de forma colaborativa e online. Ferramenta Web (EP_07, EP_09, EP_16, EP_17) e Análise Literária (EP_03, EP_10, EP_11, EP_21) aparecem em 18,20% dos estudos, onde não foi possível identificar nenhum estudo com ferramenta colaborativa para modelagem ou especificação, mas a utilização de revisões sistemáticas para auxiliar na compreensão dos requisitos para modelagem e especificação de sistemas para AAL.

RQ5 - Quais são os problemas em aberto relacionados à área de pesquisa? As respostas a essa questão abrem caminho para novas pesquisas nessa área. Observou-se que a maior parte dos estudos projetou trabalhos futuros com objetivo de aprimorar os modelos, técnicas ou sistemas, de forma a validar seu estudo em outros cenários, destacando-se os estudos EP_11, EP_19. Três estudos (EP_03, EP_12, EP_18) tiveram como trabalhos em aberto estruturar técnicas auxiliares para aquele sistema ou metodologia desenvolvido. Dois dos estudos (EP_13, EP_11) afirmaram que a falta de materiais na literatura relacionados podem ter limitado o estudo, de forma que trabalhos semelhantes ou na mesma linha seriam feitos futuramente.

3.3 Ameaças à validade

Como ameaças à validade desta revisão, é possível citar as limitações apresentadas pelas ferramentas de apoio a RLS que suportem apenas arquivos no formato bibtex, excluindo, por exemplo, bases como PubMed e PMC. De modo a padronizar todos os arquivos, foi desenvolvido uma ferramenta *web* para suportar arquivos de diferentes formatos (CVS, NBIB, TXT, BIB) e convertidos para o formato JSON (*JavaScript Object Notation*), que possibilita trabalhar de forma nativa com linguagens de programação (PHP, Python, Kotlin, Java) e base de dados (relacional ou não relacional). Outra ameaça foi a adaptação da **String** para cada base de dados, que pode mudar o foco da pesquisa caso não seja feita de forma precisa. Para mitigar esta ameaça, foi solicitado a um especialista que avaliasse a **String** para validá-la e melhorar sua efetividade. Por fim, a última ameaça é o número de pesquisadores. Para evitar o risco de vieses presente no trabalho, todo o processo foi revisado pelos autores em reunião online, reduzindo o risco de interpretações pessoais presentes na revisão.

4 O Método Proposto

Nesta seção introduzimos o método para modelagem de requisitos não funcionais para *Ambient Assisted Living*.

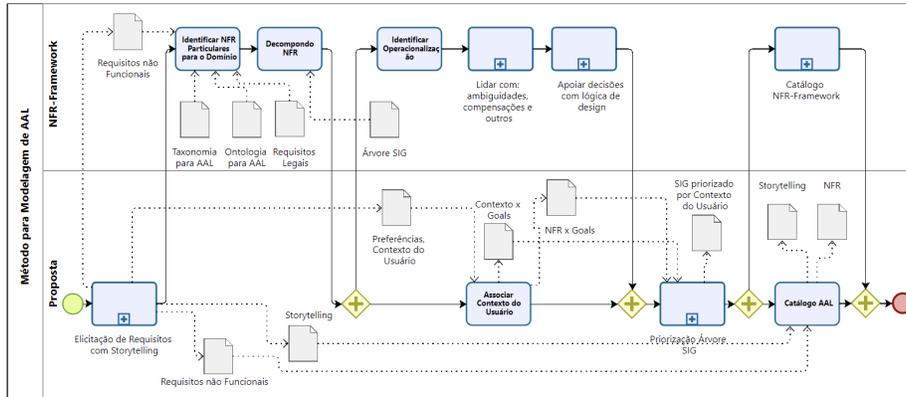


Figura 2. Visão Geral do Método Proposto como Extensão para o NFR Framework

A Figura 2 apresenta a visão geral do processo (*layer NFR Framework* e Método Proposto). A *layer* do método proposto, consiste em dois subprocessos (Elicitação de Requisitos com *Storytelling* e Priorização Árvore SIG) e dois processos (Associar Contexto do Usuário e Catálogo AAL), cada um deles com o objetivo de atender os aspectos que foram retratados acima. O método começa com uma captura inicial de requisitos utilizando *Storytelling*; um subprocesso com passos para descobrir requisitos não funcionais, contextos de uso e preferências do usuário. Como saída desse subprocesso deve gerar artefatos, tais como: *NFRs*, *storytelling*, contexto de uso e preferências que são usadas como entrada em outros processos e subprocessos do método. No processo "Associar Contexto de Uso", é estudada a influência do contexto e as preferências do idoso sobre os requisitos, onde realiza-se o relacionamento entre os *NFRs* e contexto do usuário, atribuindo um peso para cada situação. No subprocesso "Priorização Árvore SIG" é apresentado trechos de código em Python para gerar uma árvore SIG priorizada por cores com base no contexto de uso. Por fim, no processo "Catálogo AAL", além do catálogo proposto por Chung, é adicionado outros artefatos que contribuem para o desenvolvimento com reuso, tais como: *Storytelling*, *NFR*, Contextos de Uso, Taxonomias, Requisitos Legais e árvores SIG's.

4.1 Elicitação de Requisitos com Storytelling

A Figura 3 apresenta o detalhamento das atividades referentes a "Elicitação de Requisitos com *Storytelling*". É iniciado com o processo "Adquirir informações sobre o sistema", onde o Analista de Requisitos (AR) deverá obter informações do sistema a ser desenvolvido. No catálogo de AAL (Subseção 4.4), o AR tem

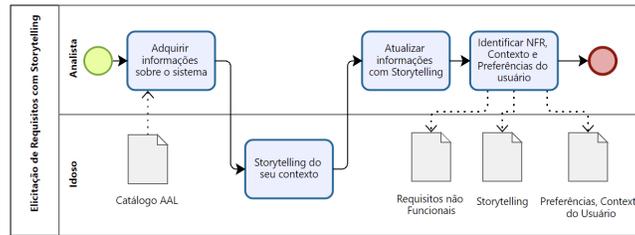


Figura 3. Detalhamento do Subprocesso Elicitação de Requisitos com *Storytelling*

acesso a informações como descrição, requisitos funcionais, *NFRs* e perguntas importantes que devem ser feitas aos *stakeholders* de modo a auxiliar na captura de informações importantes para o sistema. Com as informações, o próximo passo é capturar informações do idoso no processo "*Storytelling* do seu contexto", direcionando sempre o idoso a narrar suas histórias (*Storytelling*). Esta é uma etapa importante para o levantamento de outras informações como as preferências do usuário, contexto, requisitos não funcionais, questões culturais e seu conhecimento com tecnologias. O trabalho de Gausepohl [12] faz o uso de *Storytelling* e utiliza 5W1H para capturar os requisitos para sistemas médicos, é sugerido ao AR a utilização desta técnica quando for necessário capturar informações importantes do contexto de uso. Com informações capturadas na *Storytelling*, o AR deve atualizar o projeto, passando a atribuir tags para os requisitos funcionais, requisitos não funcionais, preferências do usuário e contexto de uso identificados, assim, gerando os seguintes artefatos: Lista de Requisitos Não funcionais, *Storytelling*, contexto de uso e preferências do usuário. Esses artefatos serão usados como entrada em outros processos, principalmente no catálogo AAL, necessário na atividade de desenvolvimento para reuso[17]. Este subprocesso substitui o processo "Adquirindo Conhecimento de Domínio" proposto por Chung. No processo "Identificando *NFRs* Particulares para o Domínio", são adicionadas outras informações que podem auxiliar nessa identificação, por exemplo, Garcés [11] propõe uma taxonomia para AAL, e *AAL Guidelines* [2] apresentam diretrizes da AAL para ética, privacidade e segurança de dados do projeto Europeu de AAL. Com as entradas (características do sistema, requisitos, entre outros) do subprocesso anterior, é possível exibir possíveis *NFRs* ou restrições que não foram identificados anteriormente. Essas informações são agrupadas por subdomínios do AAL e sistemas similares (disponíveis no catálogo). Após finalizar este processo, é seguido o fluxo do *NFR Framework*, com a decomposição de *NFR*, aqui sendo apoiado pelo catálogo, onde é gerado um SIG parcial com informações capturadas nos processos anteriores.

4.2 Associar Preferências do Usuário

Este processo pretende relacionar os requisitos não funcionais, contexto de uso e preferências do usuário com os elementos do gráfico de interdependência, definindo uma prioridade nessa associação: Baixo, Normal e Importante. Por exemplo, na etapa de *Storytelling*, o AR identificou "Conforto do Usuário" como

um *NFR*, "Localização do Usuário" e "Tarefas Urgentes" como contexto de uso, assim o AR deve relacionar o *NFR* encontrado a cada *goal* e *softgoal*. No contexto "Localização do Usuário" foi identificado os valores "Casa" ou "Externo", e para "Tarefas Urgentes" os valores "Verdadeiro" ou "Falso". Nesta etapa o AR deverá definir expressões lógicas para calcular um peso w (um número real no intervalo $[0,1]$), para cada *NFR*, como é apresentado na Figura 4.

LocalizacaoUsuario=Externo, 0
 LocalizacaoUsuario=Casa \vee TarefaUrgente=Falso, 0.8
 LocalizacaoUsuario=Casa \vee TarefaUrgente=Verdadeiro, 0.5
 LocalizacaoUsuario=Casa \vee Temperatura=Frio, 0.7

Figura 4. Expressão Lógica do NFR com Contexto do Usuário

Portanto, o NFR "Conforto do Usuário" pode assumir três valores, baixo ($0.0 < w < 0.4$), normal ($0.5 < w < 0.7$) ou importante ($0.8 < w < 1.0$) a depender do contexto do usuário. Ao final desta etapa, o AR terá a visão do impacto do contexto do uso no SIG, representando cada prioridade por cores.

4.3 Priorização Árvore SIG

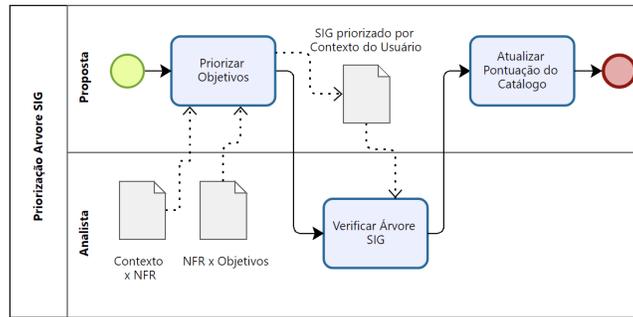


Figura 5. Detalhamento do Subprocesso Priorização Árvore SIG

A Figura 5 apresenta o subprocesso "Priorização da Árvore SIG". Uma vez modelado o gráfico interdependência, uma avaliação deve ser realizada para verificar o cumprimento das metas que estão presentes dentro dele. Esse cumprimento de metas é utilizado conforme descrito no NFR Framework [8], mas para exibir uma árvore SIG priorizada é executado duas regras representadas na Figura 6. Durante a execução do código `codigo1.py`, para cada objetivo (linhas 6–9) a satisfação é estimada conforme as regras de avaliação. Se algum dos requisitos não estiver satisfeito, o AR é solicitado a adicionar algumas modificações no subconjunto de operacionalização, de modo a tentar satisfazer os objetivos insatisfeitos. Depois disso, a satisfação do objetivo é verificada (linhas 10–15). Se algum dos objetivos críticos e importante não for atendido, a avaliação é considerada falha, o AR

é avisado. Esse código deve ser executado até que todos os objetivos estejam satisfeitos. Na execução do `codigo2.py` é verificado se todos os *NFRs* e contexto de uso (linhas 20–25) foram relacionados, caso existe algum *NFR* sem contexto, será apresentado uma mensagem de falha para ser resolvida esta pendência. Para cada contexto de uso, o AR deverá informar os valores possíveis, por exemplo, no Contexto de Uso "LocalizacaoUsuario", deverá ser informado Externo ou Casa, para ser atualizada os pesos de cada *NFR* (linhas 28–32). Com todos os pesos atualizados, para cada objetivo e operacionalização do objetivo, será atribuído uma cor conforme a prioridade do *NFR* (linhas 35–38), onde por padrão são usadas as cores vermelho, amarelo e cinza. Ao final será exibido o SIG com indicações de cores para os objetivos e operacionalizações que melhor representam a seleção do contexto feita pelo AR.

```

codigo1.py
1 # Lendo JSONs
2 treeSIG = json.loads(dataTreeSIG) # json Arvore SIG
3
4 # Definição de variáveis
5 objetivos = treeSIG.objetivos
6 for objetivo in objetivos:
7     # Regras Sugeridas pelo NFR-Framework
8     objetivo.atualizarRegraAvaliacao()
9     if (len(objetivo.operationalizations) == 0 && objetivo.status != SATISFIED):
10        corrigirInferenciaSatisfacao(objetivo)
11    elif (objetivo.status != SATISFIED || objetivo.prioridade == CRITICAL):
12        output(objetivo, FAILURE)
13    elif (objetivo.status != SATISFIED || objetivo.prioridade == IMPORTANT):
14        output(objetivo, WARNING)
15    else:
16        output(objetivo, SUCCESS)

codigo2.py
1 # Lendo JSONs
2 contextoUso = json.loads(dataUC) # json UC - Contexto de Uso
3 nrfs = json.loads(dataNRF) # json NRF
4 UCNRF = json.loads(dataUCNRF) # json UCxNRF
5 treeSIG = json.loads(dataTreeSIG) # json Arvore SIG
6
7 # Funcao que atualiza pesos dos NRF
8 def atualizarPesoRequisitos(elemento, contexto):
9     # Funcao de atualizacao de cores por prioridade
10    def atualizarPrioridadePorCor(elemento, contexto):
11        # Função que verifica se cada NRF foi associado a um ou mais contextos
12        def verificarAssociacaoNRFContexto(nrfs):
13            # Verifica erros na associacao NRF x Contexto
14            error = verificarAssociacaoNRFContexto(nrfs)
15
16        if (error == false):
17            for contexto in contextoUso:
18                contexto.valor = input()
19                # Atualizar pesos de cada requisito
20                atualizarPesoRequisitos(contextoUso)
21            # Para cada objetivo, atualizar o elemento por cor
22            objetivos = treeSIG.objetivos
23            for objetivo in objetivos:
24                atualizarPrioridadePorCor(objetivo, contexto)
25                for opers in objetivo.operationalizations:
26                    atualizarPrioridadePorCor(opers, contexto)
27
28 #Exibir a arvore SIG com indicacoes de prioridade por cor.
29 showGraphical(treeSIG)

```

Figura 6. Trechos dos Códigos 1 e 2

4.4 Catálogo AAL

Segundo Chung, para trazer o conhecimento relevante à atenção do AR em cada ponto do processo de *design*, esse precisa ser representado em uma estrutura de catálogo flexível. Propomos adicionar ao catálogo, outros artefatos do projeto para auxiliar no desenvolvimento com reuso: *Storytelling*, requisitos não funcionais, árvore SIG, taxonomia, ontologias e requisitos legais. Assim, ao iniciar um novo projeto, com base em parâmetros é possível listar artefatos do catálogo para auxiliar no *setup* inicial do projeto. A modelagem do catálogo, bem como os conceitos do desenvolvimento com reuso são implementados conforme descrito por Manoel [17].

5 Estudo de Caso

Nesta seção é apresentado um estudo de caso, nos quais o processo proposto foi utilizado integralmente com o objetivo de modelar *NFRs* para AAL. Para o estudo de caso, o AR vai interagir com a Mariana, uma viúva de 65 anos,

com pressão alta e níveis de colesterol elevados. Durante o processo "Elicitação de Requisitos com *Storytelling*" o AR faz a seguinte transcrição: Todos os dias, eu acordo às 7h ouvindo minha rádio preferida, e quando não tenho nenhuma atividade urgente, gosto de abrir a janela para sentir a luz do sol. Enquanto vejo a paisagem, gosto de ler a bíblia até às 7h20 enquanto a água do banheiro vai aquecendo, assim, posso tomar banho enquanto meu café é feito pela máquina que ganhei do meu filho. Depois, vou para a cozinha tomar meu café quente enquanto ouço a previsão do tempo. Eu preciso tomar muitos remédios e às vezes acabo esquecendo. Quando eu tenho alguma atividade externo, eu gostaria de ir no meu carro, mas tenho medo de dirigir sozinha. Acabo utilizando o transporte público, porque nunca acerto chamar esses carros por aplicativos.

A partir da *Storytelling* o AR identificou:

- Requisitos não funcionais: conforto do Usuário, Eficiência do Usuário e Eficiência Energética;
- Contexto de Uso: Localização, Tarefas Urgentes e Temperatura;
- Preferências do usuário: Luz Natural, Leitura e Controle de Medicamentos;
- Observações: Mariana tem dificuldades para acordar em dias mais frios, por isso, ela às vezes acaba usando um despertador para poder acordar na hora certa; doenças identificadas em Mariana: Pressão Alta e Colesterol Elevado.

O AR indicou o subdomínio "Saúde e Cuidados na vida" do sistema que precisa desenvolver, assim, no processo de "Identificar NFRs particulares para o Domínio" o catálogo identificou informações sobre taxonomia [11] e requisitos legais [2] conforme itens abaixo:

- Requisitos não funcionais sugeridos: Satisfação e Usabilidade;
- Requisitos legais - Padrões internacionais: ISO/TC 314 *Ageing Societies* e ISO/PRF TS 82304-2 *Health software*;
- Requisitos legais - Soluções éticas [2]: (i) Definir o que as pessoas (profissionais de saúde e família) fazem quando um alarme tocar; (ii) com base na idade, devem ser utilizadas cores mais vibrantes, de modo a serem facilmente identificáveis.

Verificadas as informações, o processo "Decompondo NFR" vai receber como entrada uma árvore SIG gerada a partir do catálogo. Um trecho desta árvore SIG é apresentada na Figura 7.

Em paralelo com os demais processos, proposto pelo *NFR Framework* o AR vai executando o processo "Associar Preferências do Usuário", para construir uma expressão lógica e ajudar na definição dos pesos, conforme apresentado na Figura 8 o trecho da expressão lógica e pesos para cada *NFR* com base no contexto do usuário.

Após modelagem do SIG no cenário 1, são atribuídos para o contexto do usuário os seguintes valores: *LocalizacaoUsuario* = "Casa", *TarefaUrgente* = "falso" e *Temperatura* = "Quente". Seguindo os passos, após a execução dos códigos da Figura 6, é exibida a árvore SIG priorizada por cores, conforme Figura 9.

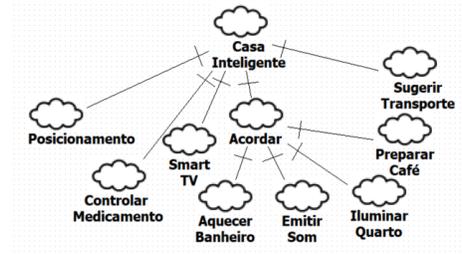


Figura 7. Árvore SIG gerada a partir do Catálogo

- LocalizacaoUsuario=Externo, 0
- LocalizacaoUsuario=Casa v TarefaUrgente=Verdadeiro, 1
- LocalizacaoUsuario=Casa v TarefaUrgente=Falso, 0.3
- LocalizacaoUsuario=Casa v Temperatura=Frio, 0.9

Eficiência do Usuário

- LocalizacaoUsuario=Externo, 0.9
- LocalizacaoUsuario=Casa, 0.4
- LocalizacaoUsuario=Casa v Temperatura=Frio, 0.8

Usabilidade

- LocalizacaoUsuario=Externo, 0
- LocalizacaoUsuario=Casa, 1

Figura 8. Trecho da Expressão Lógica do NFR com Contexto do Usuário para o Estudo de Caso

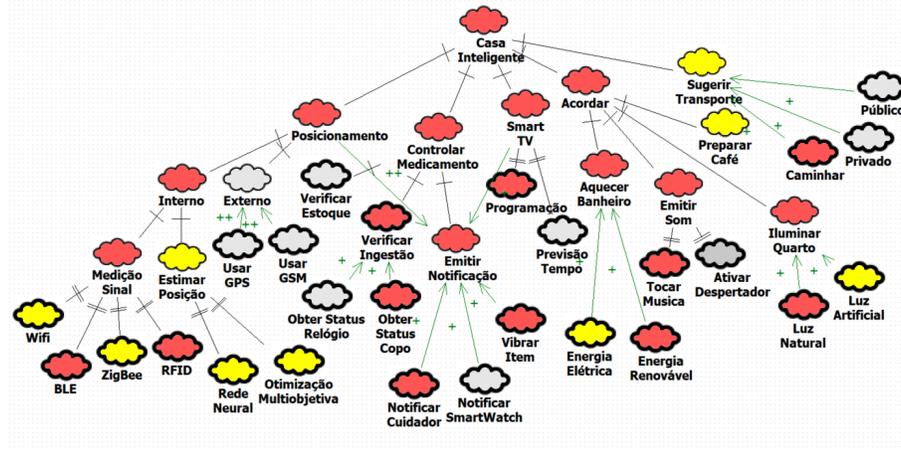


Figura 9. Trecho da Árvore SIG Priorizada por Cores no Cenário 1

Para um novo cenário, se fosse atribuído os valores LocalizacaoUsuario = "Casa", TarefaUrgente = "verdadeiro" e Temperatura = "Frio", seria gerada uma nova árvore SIG priorizada com novas cores.

6 Conclusão e Trabalhos Futuros

Uma proposta para um Método para Modelagem de Requisitos não Funcionais em *Ambient Assisted Living*, foi apresentada neste artigo. Fornece um tratamento sistemático de Requisitos Não Funcionais, em simultâneo, prestando atenção

especial às características que podem impactar o sistema. Ao criar uma extensão para o *NFR Framework*, foi possível tratar as preferências e o contexto de uso do usuário apoiando outras etapas propostas no *framework*. As principais contribuições deste estudo são: (i) Fornece uma orientação base para a elicitación de requisitos, além da captura de *NFR*, preferências do usuário e contexto de uso; (ii) Modela as contribuições de potenciais decisões arquitetônicas e de *design*, e suas combinações, para a satisfação de objetivos; (iii) Mostra o impacto do contexto nos objetivos que precisam ser cumpridos sempre que for necessário; (iv) Fornece um procedimento algoritmo fim de determinar quais decisões são as mais adequadas para a satisfação dos objetivos sob certas restrições (contexto de uso e preferências do usuário); (v) Fornece um catálogo com artefatos para auxiliar no início e em todas as fases do projeto em AAL; (vi) Fornece suporte a outros elementos que impactam na modelagem dos *NFRs*, a exemplificar as normativas legais que variam conforme a legislação vigente na região de utilização do sistema.

Como trabalhos futuros sugere-se: (i) Avaliar o uso de algoritmos inteligentes como árvore de decisão, otimização multiobjetiva ou redes neurais com múltiplas classes para identificar *NFRs*, contexto de uso e no processo de priorização da seleção e avaliação do impacto de decisão; (ii) Validar o método em outros estudos de caso em AAL com colaboração da indústria; (iii) Desenvolver uma ferramenta *web* para automatizar as etapas do método, catálogo e geração de cenários de SIG priorizado por cores; (iv) Criar uma categoria para os requisitos legais que considerem, por exemplo, a região onde o sistema AAL será implementado; (v) Publicar a versão completa da RSL realizada neste estudo.

Agradecimentos

Agradecemos a CAPES por disponibilizar os recursos necessários à elaboração do trabalho e ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE/UFPE).

Referências

1. Braid project - bridging research in ageing and ict development (2012), <https://cordis.europa.eu/project/id/248485>
2. Aal guidelines for ethics, data privacy and security (2020), <https://bit.ly/3t8cU0P>, [Online; accessed 01-março-2021]
3. Arquivos da Revisão Sistemática da Literatura. <https://bit.ly/3t6DhEN> (2021), [Online; accessed 01-Fev-2021]
4. Dashboard da RSL. <https://bit.ly/39U1jxV> (2021), [Online; accessed 01-março-2021]
5. Abbott, R.J.: An integrated approach to software development. John Wiley & Sons, Inc. (1986), <https://dl.acm.org/doi/10.5555/1098691>
6. Almeida, A., Mulero, R., Rametta, P., Urošević, V., Andrić, M., Patrono, L.: A critical analysis of an iot—aware aal system for elderly monitoring. *Future Generation Computer Systems* (2019). <https://doi.org/10.1016/j.future.2019.03.019>

7. van den Broek, G., Cavallo, F., Wehrmann, C.: AALIANCE ambient assisted living roadmap (2010), <https://dl.acm.org/doi/10.5555/1805924>
8. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering, vol. 5 (01 2000). <https://doi.org/10.1007/978-1-4615-5269-7>
9. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering, vol. 5. Springer Science & Business Media (2012)
10. Chung, L., do Prado Leite, J.C.S.: On non-functional requirements in software engineering. In: Conceptual modeling: Foundations and applications, pp. 363–379. Springer (2009)
11. Garcés, L., Ampatzoglou, A., Avgeriou, P., Nakagawa, E.Y.: Quality attributes and quality models for ambient assisted living software systems: A systematic mapping. Information and Software Technology (2017). <https://doi.org/10.1016/j.infsof.2016.10.005>
12. Gausepohl, K.: Investigation of storytelling as a requirements elicitation method for medical devices (2008), <http://hdl.handle.net/10919/36401>
13. Grguric, A.: Ict towards elderly independent living. Research and Development Centre, Ericsson Nikola Tesla (2012)
14. IBGE: Projeções e estimativas da população do Brasil e das Unidades da Federação. <https://bit.ly/3uBmWbt> (2021), [Online; accessed 01-march-2021]
15. ISO/IEC 25010: ISO/IEC 25010:2011, systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models (2011)
16. de Macedo, P.C., de Cássia Catini, R., Neto, C.C.: Sistemas críticos, um guia para elicitação de requisitos de software. UNIVERSITAS (16) (2016)
17. Manoel, M., Lencastre, M.: M-4reuse: Reusing use cases specifications. Association for Computing Machinery (2015). <https://doi.org/10.1145/2695664.2695947>
18. Mian, P., Conte, T., Natali, A., Biolchini, J., Travassos, G.: A systematic review process for software engineering. In: ESELAW05: 2nd Experimental Software Engineering Latin American Workshop (2005)
19. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Transactions on software engineering **18**(6), 483–497 (1992)
20. OMS: “Ageing well” must be a global priority. <https://bit.ly/3s0U7DJ> (2014), [Online; accessed 01-August-2019]
21. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology (2015)
22. Sommerville, I.: Software engineering 9th edition (2011)