



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

ALEX FELIPE FERREIRA COSTA

**ESTUDO EXPLORATÓRIO SOBRE O DESIGN DA USABILIDADE INTEGRADO
AO DESIGN DA ARQUITETURA DO SOFTWARE**

RUSSAS

2018

ALEX FELIPE FERREIRA COSTA

ESTUDO EXPLORATÓRIO SOBRE O DESIGN DA USABILIDADE INTEGRADO AO
DESIGN DA ARQUITETURA DO SOFTWARE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Orientadora: Profa. Dra. Anna Beatriz
dos Santos Marques

RUSSAS

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C87e Costa, Alex Felipe Ferreira.
Estudo exploratório sobre o design da usabilidade integrado ao design da arquitetura do software / Alex Felipe Ferreira Costa. – 2018.
101 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2018.

Orientação: Profa. Dra. Anna Beatriz dos Santos Marques.

1. Usabilidade. 2. Arquitetura de Software. 3. Design. 4. Estudo Exploratório. I. Título.

CDD 005.1

ALEX FELIPE FERREIRA COSTA

ESTUDO EXPLORATÓRIO SOBRE O DESIGN DA USABILIDADE INTEGRADO AO
DESIGN DA ARQUITETURA DO SOFTWARE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Engenharia de Software
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Profa. Dra. Anna Beatriz dos Santos
Marques (Orientadora)
Universidade Federal do Ceará (UFC)

Profa. Dra. Marília Soares Mendes
Universidade Federal do Ceará (UFC)

Profa. Dra. Valéria Lelli Leitão Dantas
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha mãe, Verônildes
Ferreira, por todo amor, carinho e por sempre
acreditar em mim.

AGRADECIMENTOS

Primeiramente a Deus, por ter me dado a força necessária para enfrentar todas as dificuldades encontradas durante a minha jornada em busca dos meus sonhos.

À minha mãe, Verônildes Ferreira, por sempre apoiar, incentivar e acreditar em mim. Obrigado por todo o seu empenho, dedicação, por sempre investir e dar duro por mim. Te amo! Você sempre vai ser a minha maior inspiração.

Ao meu pai, Francisco Dantas, e aos meus irmãos, Arthur Felipe e Alan Felipe. Obrigado por todos os incentivos e momentos divertidos que vocês me proporcionaram durante essa jornada.

À minha orientadora, Anna Beatriz, que se esforçou ao máximo para transmitir seus conhecimentos. Por ter me apresentado uma das melhores áreas em que eu poderia estar. Obrigado por confiar em mim e agradecer imensamente por todo o trabalho, dedicação, atenção e amizade. Quero sempre continuar cooperando, aprendendo e crescendo com seu conhecimento. Me inspiro na professora, pesquisadora, profissional e pessoa que você é. Obrigado! :)

Aos meus melhores amigos desses últimos quatro anos, Liana Mara (Lheanna), Gabriel Gonçalves, Lavínia Matoso (Lavinha) e Thiago Hellen. A amizade de vocês com certeza foi o melhor presente que a UFC poderia ter me dado. Obrigado por todos os momentos que estivemos juntos, por terem rido e sofrido comigo nos momentos de grande desespero e de vergonhas que tivemos durante o curso e, claro, por todos os memes sem graça. Vocês são os melhores!

Lavínia, obrigado por todos os momentos de conversas, brincadeiras, conselhos e troca de ideias. Obrigado por ter me aguentado durante esses quatro anos e pelos melhores trabalhos que fizemos juntos. Você é a melhor!

À minha tia, Ozani Dantas, e à minha prima, Aurilene. Obrigado por me receberem na casa de vocês durante boa parte dessa jornada. Sem o apoio de vocês, talvez nada disso teria sido realidade.

À minha tia, madrinha, guerreira e segunda mãe, Verônica Dantas (Dedé), por sempre me incentivar e me apoiar.

À minha tia, prima, madrinha, pseudo irmã, Verilene Dantas (Lene). Obrigado por todos os momentos que estivemos juntos e por ter me aguentado durante todo esse tempo.

Ao Laboratório INterdisciplinar de Computação e Engenharia de Software – LINCE, por ter me dado a oportunidade de ingressar no mundo da pesquisa. Ao meu primeiro orientador,

Marcos Vinícius e a todos os meus colegas de pesquisa, por terem contribuído de forma direta ou indireta no incentivo à pesquisa.

A toda minha família, que durante a graduação, ficaram felizes e apoiaram em minhas conquistas. Em especial a minha avó, Maria Mirtes, e meu avô, Joaquim Sabino (*in memoriam*) que sempre se preocupou comigo e me apoiou por durante toda minha vida.

Enfim, agradeço a todos que contribuíram na realização deste trabalho e na minha graduação, seja de forma direta ou indireta. Não deixaria de agradecer também, a todos que duvidaram do meu potencial, que de certa forma me deram forças para sempre seguir em frente e mostrar que sou capaz de atingir meus objetivos.

Gratidão a todos!

“Eu sei que é difícil quando você está caindo, e é um longo caminho quando você atinge o chão. Levante-se agora”

(Imagine Dragons)

RESUMO

A usabilidade é um atributo da qualidade dos sistemas que são fáceis de serem entendidos, usados e atraentes aos usuários finais. Incorporar recomendações de usabilidade durante o desenvolvimento de sistemas não é uma tarefa trivial, pois sua incorporação impacta em aspectos que vão além da interface com o usuário, afetando diretamente as funcionalidades do sistema e a sua arquitetura. Para isso, é necessário que a usabilidade também seja considerada durante o design da arquitetura do sistema, visto que alterações futuras exigem altos custos, algumas vezes inviáveis de serem realizadas. Nesse sentido, realizou-se um estudo exploratório para investigar como a usabilidade pode ser representada no design da arquitetura de sistemas. O estudo foi conduzido em ambiente acadêmico durante a condução de projetos da arquitetura de aplicações web e mobile, realizados por dez equipes de estudantes de uma disciplina de Arquitetura de Software. O estudo foi subdividido em 4 etapas: (i) seleção de padrões, onde foram selecionados trabalhos da literatura que apresentam padrões para o design da usabilidade integrado à arquitetura de software; (ii) treinamento com os participantes, onde foram realizadas aulas e discussões sobre os padrões selecionados; (iii) execução dos projetos de design arquitetural, na qual os participantes realizaram a etapa de design da arquitetura de seus sistemas incorporando aspectos da usabilidade; (iv) análise quantitativa e qualitativa dos resultados dos projetos realizados pelos participantes. Os resultados indicaram que há uma necessidade de melhorias em diretrizes de usabilidade apresentadas na literatura e de um maior direcionamento para a tomada de decisões na integração da usabilidade na arquitetura de software. Além disso, foi possível explorar soluções alternativas de como representar a usabilidade na arquitetura e percepções das equipes sobre motivações e desafios ao integrar a usabilidade na arquitetura de software.

Palavras-chave: Usabilidade. Arquitetura de Software. Design. Estudo Exploratório.

ABSTRACT

Usability is an attribute of the systems quality that are easy to understand, to use, and attractive to end users. Incorporating usability recommendations during system development is not a trivial task. Since its incorporation impacts on aspects that go beyond the user interface, directly affecting the system's functionalities and architecture. For this, it is necessary that usability also be considered during the design of the system architecture. Since future changes demand high costs, sometimes unviable to be realized. In this sense, an exploratory study was conducted to investigate how usability can be represented in the system architecture design. The study was conducted in an academic environment during the conduction of web and mobile application architecture projects carried out by ten teams of students of a Software Architecture discipline. The study was conducted into four stages: (i) selection of standards, where papers were selected from the literature that present standards for usability design integrated with the software architecture; (ii) training the participants, where classes and discussions were held on the selected standards; (iii) execution of the architectural design projects, in which the participants carried out the design stage of their systems architecture incorporating usability aspects; (iv) quantitative and qualitative analysis of the projects results carried out by the participants. The results indicated that there is a need for improvements in usability guidelines presented in the literature and a greater direction for decision making in the integration of usability in the software architecture. In addition, it was possible to explore alternative solutions of how to represent usability in architecture and team perceptions about motivations and challenges when integrating usability into software architecture.

Keywords: Usability. Software Architecture. Design. Exploration Study.

LISTA DE FIGURAS

Figura 1 – Representação do modelo de visão 4 + 1	24
Figura 2 – Imagem com a distribuição dos participantes por equipe	34
Figura 3 – Fotografia com a representação de mecanismos de usabilidade em diagrama de caso de uso durante a etapa de treinamento	35
Figura 4 – Gráfico com o número de requisitos de usabilidade especificados que não estão associados aos mecanismos de usabilidade	38
Figura 5 – Gráfico com a porcentagem de uso dos meta-modelos e de soluções alternativas para representação da usabilidade	39
Figura 6 – Gráfico com os resultados da análise quantitativa	40
Figura 7 – Diagramas com as soluções alternativas propostas pela equipe E02	45
Figura 8 – Diagrama com as soluções alternativas propostas pela equipe E06	53
Figura 9 – Diagrama com a solução alternativa proposta pela equipe E08	58
Figura 10 – Diagramas com as soluções alternativas propostas pela equipe E09	62
Figura 11 – Rede de visualização sobre a opinião do uso das diretrizes no projeto arquitetural	66
Figura 12 – Rede de visualização sobre a influência na decisão por adotar ou não as diretrizes	68
Figura 13 – Rede de visualização sobre o impacto positivo na facilidade de uso das diretrizes	70
Figura 14 – Rede de visualização sobre a percepção de facilidade de uso das diretrizes	71
Figura 15 – Meta-modelo para o mecanismo desfazer no diagrama de casos de uso	79
Figura 16 – Meta-modelo para o mecanismo feedback do progresso no diagrama de casos de uso	81
Figura 17 – Meta-modelo para o mecanismo feedback do status do sistema no diagrama de casos de uso	82
Figura 18 – Meta-modelo para o mecanismo passo a passo no diagrama de casos de uso	83
Figura 19 – Meta-modelo para o mecanismo abortar no diagrama de casos de uso	85
Figura 20 – Meta-modelo para o mecanismo área de objetos pessoais no diagrama de casos de uso	86
Figura 21 – Meta-modelo para o mecanismo ajuda multinível no diagrama de casos de uso	88
Figura 22 – Meta-modelo para o mecanismo preferências no diagrama de casos de uso	89
Figura 23 – Meta-modelo para o mecanismo alerta no diagrama de casos de uso	90

Figura 24 – Meta-modelo para o mecanismo agregação de comandos no diagrama de casos de uso	92
Figura 25 – Meta-modelo para o mecanismo favoritos no diagrama de casos de uso . . .	94
Figura 26 – Meta-modelo para o mecanismo desfazer no diagrama de classes	95
Figura 27 – Meta-modelo para o mecanismo feedback do progresso no diagrama de classes	95
Figura 28 – Meta-modelo para o mecanismo feedback do status do sistema no diagrama de classes	96
Figura 29 – Meta-modelo para o mecanismo passo a passo no diagrama de classes . . .	96
Figura 30 – Meta-modelo para o mecanismo abortar no diagrama de classes	97
Figura 31 – Meta-modelo para o mecanismo alerta no diagrama de classes	97
Figura 32 – Meta-modelo para o mecanismo área de objetos pessoais no diagrama de classes	98
Figura 33 – Meta-modelo para o mecanismo ajuda multinível no diagrama de classes . .	99
Figura 34 – Meta-modelo para o mecanismo preferências no diagrama de classes	99
Figura 35 – Meta-modelo para o mecanismo favoritos no diagrama de classes	100

LISTA DE TABELAS

Tabela 1 – FUFs e seus mecanismos de usabilidade	27
Tabela 2 – Impacto médio das FUFs no design do sistema	30
Tabela 3 – Requisitos de usabilidade especificados pela equipe E01	43
Tabela 4 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E01	43
Tabela 5 – Requisitos de usabilidade especificados pela equipe E02	44
Tabela 6 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E02	45
Tabela 7 – Requisitos de usabilidade especificados pela equipe E03	46
Tabela 8 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E03	47
Tabela 9 – Requisitos de usabilidade especificados pela equipe E04	48
Tabela 10 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E04	49
Tabela 11 – Requisitos de usabilidade especificados pela equipe E05	50
Tabela 12 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E05	51
Tabela 13 – Requisitos de usabilidade especificados pela equipe E06	52
Tabela 14 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E06	53
Tabela 15 – Requisitos de usabilidade especificados pela equipe E07	55
Tabela 16 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E07	55
Tabela 17 – Requisitos de usabilidade especificados pela equipe E08	57
Tabela 18 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E08	58
Tabela 19 – Requisitos de usabilidade especificados pela equipe E09	60
Tabela 20 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E09	61
Tabela 21 – Requisitos de usabilidade especificados pela equipe E10	63
Tabela 22 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E10	63
Tabela 23 – Especificação dos casos de uso do meta-modelo para o mecanismo desfazer	80
Tabela 24 – Especificação dos casos de uso do meta-modelo para o mecanismo desfazer	81
Tabela 25 – Especificação dos casos de uso do meta-modelo para o mecanismo feedback do status do sistema	82
Tabela 26 – Especificação dos casos de uso do meta-modelo para o mecanismo passo a passo	84
Tabela 27 – Especificação dos casos de uso do meta-modelo para o mecanismo abortar .	85
Tabela 28 – Especificação dos casos de uso do meta-modelo para o mecanismo área de objetos pessoais	87

Tabela 29 – Especificação dos casos de uso do meta-modelo para o mecanismo ajuda multinível	88
Tabela 30 – Especificação dos casos de uso do meta-modelo para o mecanismo preferências	89
Tabela 31 – Especificação dos casos de uso do meta-modelo para o mecanismo alerta . .	91
Tabela 32 – Especificação dos casos de uso do meta-modelo para o mecanismo agregação de comandos	93
Tabela 33 – Especificação dos casos de uso do meta-modelo para o mecanismo favoritos	94

LISTA DE ABREVIATURAS E SIGLAS

DT	<i>Design Thinking</i>
FUF	<i>Functional Usability Features</i>
MVC	<i>Model-view-controller</i>
UML	<i>Unified Modeling Language</i>
USINN	<i>USability-oriented INteraction and Navigation model</i>

SUMÁRIO

1	INTRODUÇÃO	17
2	OBJETIVOS	20
2.1	Objetivo geral	20
2.2	Objetivos específicos	20
3	PROCEDIMENTOS METODOLÓGICOS	21
3.1	Revisão bibliográfica	21
3.2	Estudo exploratório	21
3.3	Análise dos dados	22
4	FUNDAMENTAÇÃO TEÓRICA	23
4.1	Usabilidade	23
4.2	Arquitetura de Software	24
4.3	Functional Usability Features	25
5	TRABALHOS RELACIONADOS	28
6	ESTUDO EXPLORATÓRIO	32
6.1	Planejamento	32
6.2	Preparação	32
6.3	Especificação de requisitos de usabilidade	34
6.4	Modelagem arquitetural	35
6.5	Coleta da percepção dos participantes	36
7	ANÁLISE QUANTITATIVA	38
8	ANÁLISE DAS SOLUÇÕES ARQUITETURAIS PARA MODELAGEM DE ASPECTOS DE USABILIDADE	42
8.1	Solução arquitetural para a representação da usabilidade no projeto da equipe E01	42
8.2	Solução arquitetural para a representação da usabilidade no projeto da equipe E02	44
8.3	Solução arquitetural para a representação da usabilidade no projeto da equipe E03	46
8.4	Solução arquitetural para a representação da usabilidade no projeto da equipe E04	47

8.5	Solução arquitetural para a representação da usabilidade no projeto da equipe E05	50
8.6	Solução arquitetural para a representação da usabilidade no projeto da equipe E06	51
8.7	Solução arquitetural para a representação da usabilidade no projeto da equipe E07	54
8.8	Solução arquitetural para a representação da usabilidade no projeto da equipe E08	56
8.9	Solução arquitetural para a representação da usabilidade no projeto da equipe E09	59
8.10	Solução arquitetural para a representação da usabilidade no projeto da equipe E10	62
9	ANÁLISE QUALITATIVA DOS RELATOS DOS PARTICIPANTES . .	65
9.1	Percepção sobre o uso das diretrizes de usabilidade para o projeto arquitetural do software	66
9.2	O que influenciaria na decisão para adotar ou não as diretrizes de usabilidade	68
9.3	Percepção sobre a facilidade de utilizar as diretrizes no projeto arquitetural de software	69
9.4	Impacto positivo na facilidade de uso das diretrizes	71
10	DISCUSSÃO DOS RESULTADOS	73
11	CONCLUSÕES E TRABALHOS FUTUROS	75
	REFERÊNCIAS	76
	APÊNDICES	78
	APÊNDICE A – Questionário utilizado para extração de relatos dos participantes sobre o uso das diretrizes	78
	APÊNDICE B – Meta-modelos para diagrama de casos de uso	79
	APÊNDICE C – Meta-modelos para diagrama de classes	95
	ANEXOS	100

1 INTRODUÇÃO

A usabilidade é um atributo de qualidade de software que retrata a capacidade de um sistema ser entendido, usado e atraente ao usuário sob condições específicas (ISO 9126, 2000). Quando a usabilidade é considerada em sistemas interativos, diversos benefícios podem ser obtidos, como a redução de custos com treinamento, aumento da satisfação dos usuários e a redução de possíveis erros no uso (ABRAN *et al.*, 2003). Quando não considerada, resulta em manutenções adaptativas que consomem uma quantidade significativa de recursos para a correção de problemas no uso (FOLMER *et al.*, 2003).

Estudos citados por Folmer *et al.* (2003) identificaram que entre os custos totais gastos com manutenções adaptativas, envolvendo problemas entre os usuários e o sistema, 64% deles são relacionados a correções de problemas de usabilidade. Em casos onde a correção de problemas de usabilidade exige modificações na arquitetura do sistema, o custo pode ser ainda mais elevado. Vilela *et al.* (2015) citam em seu estudo que, em alguns desses casos, os envolvidos no desenvolvimento do software optam em não corrigir esses problemas, tendo em vista um alto custo para a realização de modificações na arquitetura do sistema, impondo aos usuários finais a ação de contornar esses problemas no uso.

Nesse sentido, Juristo *et al.* (2007a) identificaram quais das principais recomendações de usabilidade, apresentadas na literatura, ao serem implantadas afetam diretamente a arquitetura do software. Essas recomendações de usabilidade foram nomeadas por *Functional Usability Features* (FUF). Recomendações como o exibir feedback do progresso de uma tarefa demorada e permitir que usuário desfaça ou cancele operações em andamento, são exemplos de recomendações de usabilidade que afetam a arquitetura do sistema.

Para realizar a implantação dessas FUFs é necessária a inserção de componentes arquiteturais que possam identificar, por exemplo, o progresso da execução das tarefas, as alterações no status do sistema e quais tarefas foram realizadas anteriormente pelo usuário. No entanto, sua incorporação durante o *design* da arquitetura do software está longe de ser uma tarefa trivial, devido à diversidade de fatores que impactam na inclusão dessas recomendações como, por exemplo, o alto nível de abstração em que elas são apresentadas e o quão distante elas estão da implementação real do software (CARVAJAL, 2012).

No intuito de identificar trabalhos que relacionem essas duas áreas, Vilela *et al.* (2015) realizaram uma revisão sistemática da literatura buscando identificar os principais estudos que analisaram a relação entre a usabilidade e a arquitetura do software. Foram encontrados 22

estudos datados entre 1995 e 2013, onde foram analisados os impactos, os principais problemas, relatos de casos de sucesso ou de falha e as contribuições dessas soluções para a academia e a indústria.

Dentre os trabalhos identificados pela revisão, foram analisados aqueles que apresentavam padrões ou diretrizes para o design da usabilidade integrado ao design da arquitetura. Um desses estudos foi realizado por Carvajal *et al.* (2013). As autoras apresentam diretrizes para a incorporação das FUFs no desenvolvimento de software e meta-modelos em diagramas de casos de uso, classes e sequência da *Unified Modeling Language* (UML) para a representação das FUFs. Entretanto, o estudo não investigou vantagens e desvantagens da aplicação das diretrizes ou alternativas divergentes, deixando em aberto questões que precisam ser discutidas, como:

- Quais os principais desafios para a incorporação das FUFs no design da arquitetura?
- Quais os impactos das diretrizes nas decisões arquiteturais?
- Como as diretrizes podem ser adotadas por equipes de desenvolvimento de software?
- O que influenciaria na decisão para adotar ou não as diretrizes de usabilidade?

Portanto, fornecer evidências experimentais sobre como as FUFs podem ser integradas no design da arquitetura do sistema se faz necessária para auxiliar aos arquitetos de software a tomar melhores decisões durante o projeto da arquitetura de sistemas com boa qualidade de uso. Além disso, explorar como as soluções apresentadas na literatura são abordadas e qual o impacto e a percepção de uso delas, deve contribuir com melhorias nessas soluções, apoiar o seu uso e incentivar o surgimento de novas pesquisas que visem a incorporação da usabilidade durante fases iniciais de desenvolvimento, evitando altos custos com correções de problemas de usabilidade e promovendo o desenvolvimento de sistemas com boa qualidade de uso.

Diante desse cenário, foi realizado um estudo exploratório com o objetivo de investigar a integração das FUFs nas fases iniciais de desenvolvimento do sistema, com foco no *design* da arquitetura. O estudo foi conduzido em ambiente acadêmico durante o desenvolvimento de projetos para a especificação da arquitetura de aplicações *web* e *mobile*. Dez equipes participaram do estudo, totalizando 50 estudantes. Cada equipe projetou a arquitetura de uma aplicação durante um trabalho para disciplina de Arquitetura de Software, em um curso de graduação em Engenharia de Software. As soluções arquiteturais adotadas pelas equipes foram analisadas, assim como os relatos dos participantes. Os resultados dessas análises são descritas neste estudo.

O restante deste documento está organizado como segue: No Capítulo 2 estão descritos os objetivos do estudo. No Capítulo 3 estão descritos os procedimentos metodológicos usados neste estudo. O Capítulo 4 apresenta os principais termos que fundamentam este trabalho. No Capítulo 5 estão descritos os trabalhos que relacionam a incorporação da usabilidade no design da arquitetura. No Capítulo 6 são apresentados os procedimentos realizados durante a execução do projeto. No Capítulo 7 estão apresentados os resultados da análise quantitativa. No Capítulo 8 contém os resultados da análise das soluções arquiteturais. No Capítulo 9 estão os resultados da análise qualitativa dos relatos. Por fim, no Capítulo 10 é apresentada uma discussão dos resultados e no Capítulo 11 estão as conclusões e os trabalhos futuros obtidos por este estudo.

2 OBJETIVOS

2.1 Objetivo geral

O objetivo geral desta pesquisa consiste em investigar como os mecanismos de usabilidade podem ser modelados no *design* da arquitetura do software.

2.2 Objetivos específicos

Os objetivos específicos são:

- Analisar como as diretrizes propostas por Carvajal *et al.* (2013) foram adotadas no *design* arquitetural de aplicativos móveis e aplicações *web*;
- Analisar como as decisões arquiteturais foram influenciadas pelas diretrizes propostas por Carvajal *et al.* (2013);
- Investigar soluções alternativas para a proposta feita por Carvajal *et al.* (2013).

3 PROCEDIMENTOS METODOLÓGICOS

Nesta seção estão descritos os procedimentos metodológicos organizados a partir dos objetivos traçados, que foram realizados para ajudar na investigação desta pesquisa. A pesquisa foi conduzida em três etapas: (i) revisão bibliográfica de literatura; (ii) aplicação do estudo exploratório; (iii) análise dos dados.

3.1 Revisão bibliográfica

Na etapa de revisão bibliográfica, foram realizadas pesquisas voltadas para identificação de trabalhos da literatura que relacionam a usabilidade com a arquitetura de software. Foi utilizado inicialmente o mecanismo de busca Google Acadêmico, que imediatamente resultou na identificação do estudo de Vilela *et al.* (2015), na qual foi realizado uma revisão sistemática sobre a relação entre esses dois assuntos. Logo, não seria necessário o retrabalho de realizar outra revisão sistemática da literatura.

Nesse sentido, foi realizada a leitura da revisão sistemática apresentada por Vilela *et al.* (2015) e a análise dos trabalhos relevantes para este estudo. Dentre os trabalhos apresentados na revisão, apenas aqueles que citavam padrões para o *design* da usabilidade na arquitetura do software foram considerados como trabalhos relacionados a este estudo. Dentre os trabalhos selecionados, foi identificado o estudo de Carvajal *et al.* (2013) como o principal trabalho relacionado ao assunto a ser abordado neste estudo, tendo em vista que as autoras apresentam meta-modelos para o *design* da usabilidade no *design* da arquitetura.

3.2 Estudo exploratório

Após a revisão bibliográfica, foi conduzido um estudo exploratório em ambiente acadêmico com a participação de cinquenta estudantes de Engenharia de Software, matriculados na disciplina de Arquitetura de Software. O estudo foi conduzido em cinco etapas: (1) planejamento; (2) preparação; (3) especificação de requisitos de usabilidade; (4) modelagem arquitetural e (5) coleta da percepção dos participantes.

Inicialmente, foi realizado um planejamento de como o estudo seria conduzido. Em seguida, foi realizado um treinamento com os participantes do estudo, com o objetivo de apresentar o conteúdo a ser abordado no estudo e apresentar o trabalho da Carvajal *et al.* (2013). Após o treinamento, os participantes se dividiram em dez equipes e realizaram a etapa de

especificação dos requisitos do projeto que cada equipe iria projetar. Com os requisitos definidos, as equipes realizaram a etapa de projeto da arquitetura integrando os aspectos de usabilidade que eles identificaram como essenciais para o projeto.

Ao final do prazo para modelagem arquitetural, os participantes responderam um questionário sobre percepção de impacto e facilidade (Apêndice A) e o documento arquitetural elaborado por cada equipe foi disponibilizado para a análise.

3.3 Análise dos dados

Para a análise do dados, foram realizadas três tipos de análises. A análise quantitativa da representação dos mecanismos especificados e de uso de soluções alternativas as diretrizes. A análise de soluções alternativas, na qual foram analisadas as soluções adotadas pelas equipes e que estavam divergentes das propostas pelas diretrizes, e a análise qualitativa dos relatos, onde foi usado o método de Grounded Theory proposto por Strauss e Corbin (1990). Esse método utiliza um conjunto de procedimentos sistemáticos para gerar teorias substantivas sobre fenômenos essencialmente sociais (STRAUSS; CORBIN, 1990).

4 FUNDAMENTAÇÃO TEÓRICA

Com o propósito de auxiliar no entendimento deste estudo, nesta seção são apresentados os fundamentos teóricos relacionados aos assuntos abordados nesta pesquisa. Tendo em vista que esta pesquisa aborda conteúdos referentes ao design da usabilidade integrado ao design da arquitetura do software, são apresentados os conceitos relacionados a: usabilidade; arquitetura de software; *Functional Usability Features* e mecanismos de usabilidade.

4.1 Usabilidade

Existem diferentes definições para o termo usabilidade, o que ainda torna o termo um pouco confuso, especialmente para desenvolvedores de software (SEFFAH; METZKER, 2004). Folmer e Bosch (2004) citam que muitos autores gastaram esforço para definir o termo usabilidade, mas que todas essas definições estão fortemente relacionadas a quatro abordagens que, segundo os autores, são definições amplamente reconhecidas e utilizadas na prática. As definições citadas pelos autores foram apresentadas por: Shackel (1991); Nielsen (1994); ISO 9241-11 (1998); ISO 9126 (2000).

Shackel (1991) foi um dos primeiros autores a reconhecer a importância da engenharia de usabilidade e a relatividade da definição desse termo (FOLMER; BOSCH, 2004). Segundo Shackel (1991), a usabilidade é definida como a capacidade de usar um sistema com facilidade e eficácia, em termos funcionais humanos, dado um treinamento específico e o suporte aos usuários, sendo possível que o usuário possa atender a uma faixa específica de tarefas em uma faixa específica de cenários.

Nielsen (1994) define a usabilidade como um conjunto de cinco atributos: (i) aprendizagem (facilidade de aprender a usar o sistema); (ii) eficiência (após aprender a usar o sistema, a produtividade deve ser possível em um alto nível); (iii) memorabilidade (facilidade de usuários casuais lembrarem-se de como usar o sistema, mesmo após algum período sem a utilização); (iv) prevenção de erros (a taxa de erros do sistema deve ser baixa e erros catastróficos não devem ocorrer); (v) satisfação (o sistema deve ser agradável de usar, tornando os usuários subjetivamente satisfeitos ao usá-lo).

A norma ISO 9241-11 (1998) define a usabilidade como a capacidade do produto de software ser usado por usuários específicos para atingir suas metas específicas de forma eficaz, eficiente e satisfatória em um contexto de uso específico.

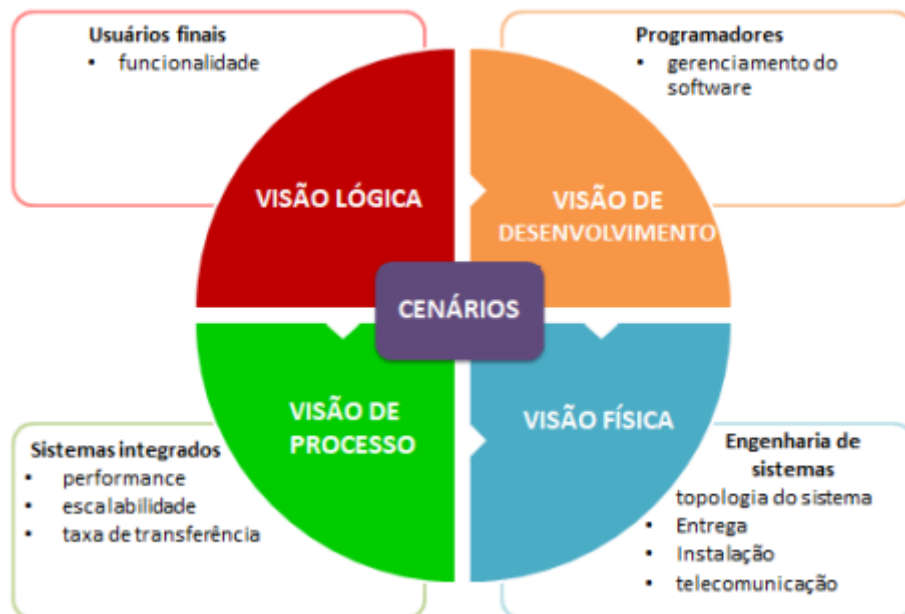
Para a ISO 9126 (2000), a usabilidade é definida como a capacidade do produto de software ser compreendido, aprendido, operado e atraente ao usuário, mediante condições específicas.

4.2 Arquitetura de Software

A arquitetura de software é o conceito de nível mais alto de um sistema em seu ambiente (IEEE, 1998) e vai além dos algoritmos e estruturas de dados (GARLAN; SHAW, 1993). A arquitetura de software pode ser definida como o conjunto de decisões significativas para a organização de um software, desde a seleção dos elementos estruturais que irão compor o sistema, até o comportamento e estilo arquitetural que irá orientar esta organização (BIEL; GRUHN, 2009). No entanto, a arquitetura de software não se limita apenas a um enfoque interno desta organização, mas também a um enfoque externo, lidando com aspectos como à adequação a integridade do sistema, restrições econômicas, preocupações estéticas e estilo (IEEE, 1998).

Kruchten (1995) descreve a arquitetura de software em cinco visualizações simultâneas. Cada uma dessas visões abrange um conjunto específico de preocupações/interesses para as diferentes partes interessadas no sistema. Esse modelo de visão é denominado por “4 + 1” e é subdividido em cinco visões: (i) visão lógica; (ii) visão de desenvolvimento; (iii) visão de processos; (iv) visão física e a (v) visão de cenários

Figura 1 – Representação do modelo de visão 4 + 1



Fonte: Kruchten (1995), adaptada pelo autor.

A visão lógica apresenta os serviços que o sistema deve fornecer aos usuários finais, com base nos requisitos funcionais do sistema. Nessa visão, os arquitetos do software representam o sistema através de um conjunto de abstrações fundamentais do domínio do problema. Essas abstrações podem ser representadas por objetos ou classes de objetos em um diagrama de classes a nível de domínio da UML.

A visão de desenvolvimento apresenta a organização estática do software em seu ambiente, para auxiliar os desenvolvedores do sistema. Nessa visão, os arquitetos do software representam a organização do sistema por meio de relacionamentos entre componentes (subsistemas, módulos, camadas, bibliotecas, pacotes, dentre outros). Esses relacionamentos podem ser representados por um diagrama de componentes ou um diagrama de classes detalhado da UML.

A visão de processo descreve a organização dos processos que irão compor o sistema, assim como a simultaneidade e a sincronização dos aspectos de design. Nessa visão, os arquitetos do software ilustram a decomposição de processos envolvidos no sistema e os relacionamentos entre eles. Esses processos podem ser representados em diagramas de sequência ou em diagramas de atividades da UML.

A visão física descreve o mapeamento dos componentes de software para o hardware. Nessa visão, os arquitetos do software ilustram os componentes físicos do sistema que estão distribuídos. Esses componentes podem ser representados por um diagrama de implantação da UML.

Com essas visões, os arquitetos de software podem organizar a descrição de suas decisões arquiteturais em torno delas e, em seguida, ilustrá-las com alguns casos de uso ou cenários selecionados, constituindo a quinta visão, a visão de cenários. Desta forma, nesta pesquisa, será utilizado o modelo arquitetural “4 + 1” para representar a arquitetura dos sistemas a serem projetados pelos participantes do estudo. Com este modelo, os participantes poderão representar a arquitetura do software para as diferentes visões, atendendo a conjuntos específicos de interesses dos envolvidos no sistema.

4.3 Functional Usability Features

Algumas recomendações de usabilidade envolvem a construção de certas funcionalidades no software para melhorar a interação com o usuário (JURISTO *et al.*, 2007b). O recurso “cancelar o progresso de uma tarefa” é um exemplo de recomendação de usabilidade proposta por Nielsen (1994), que afirma:

Se o tempo para terminar o processamento do comando for maior que 10s, juntamente com a informação de *feedback*, deve-se fornecer uma opção de cancelamento da operação para interromper o processamento e voltar para o estado anterior. (NIELSEN, 1994).

Para realizar essa ação de cancelamento de progresso em um sistema, será necessário que o software realize passos como: interromper a execução do comando, estimar o tempo para cancelar, informar o usuário sobre o progresso do cancelamento, dentre outras ações. Portanto, além das mudanças a serem realizadas na interface do usuário para adicionar o botão de cancelamento, componentes específicos devem ser incorporados ao *design* do software para lidar com essas responsabilidades.

Essas recomendações de usabilidade que impactam na criação de novas funcionalidades a serem incorporadas em um sistema, foram denominadas por *Functional Usability Features* (FUF) por (JURISTO *et al.*, 2007b). As FUFs representam os aspectos da usabilidade com maior impacto no design da arquitetura do software Carvajal *et al.* (2013). Além disso, esses aspectos possuem variedades de funcionalidades, denominadas por mecanismos de usabilidade (JURISTO *et al.*, 2007a; JURISTO *et al.*, 2007b). Cada aspecto funcional da usabilidade identificado é um produto das recomendações de usabilidade apresentados por especialistas em Interação Humano-Computador (IHC). Cada recomendação de usabilidade identifica os diferentes subtipos de uma FUF. Cada subtipo está referido como um mecanismo de usabilidade que indica o nome da funcionalidade (RODRÍGUEZ *et al.*, 2015). Na Tabela 1, estão descritos os mecanismos de usabilidade e suas respectivas FUFs identificadas no estudo de Juristo *et al.* (2007b).

Tabela 1 – FUFs e seus mecanismos de usabilidade

FUF	Mecanismo de usabilidade	Descrição
Feedback	Status do sistema	Informar os usuários sobre o estado interno do sistema
	Interação	Informar os usuários que houve registro de uma interação
	Alerta	Informar os usuários sobre qualquer ação com consequências importantes
	Feedback sobre o progresso	Informar os usuários quando o sistema estiver processando uma ação que poderá levar algum tempo
Desfazer / Cancelar	Desfazer ação global	Desfazer ação do sistema em vários níveis
	Desfazer ações em um objeto específico	Desfazer várias ações em um objeto
	Abortar operações	Cancelar a execução de uma ação ou de toda a aplicação
	Voltar	Retornar a um determinado estado em uma sequência de execução de comandos
Entrada de dados e Prevenção / Correção de erros	Entrada de texto estruturada	Prevenir que usuários cometam erros de entrada de dados
Wizard	Execução passo-a-passo	Auxiliar os usuários a realizar tarefas que requerem diferentes passos com entrada de dados corretas
Perfil do usuário	Preferências	Registrar as opções do usuário no uso das funções do sistema
	Áreas de objetos	Registrar as opções do usuário no uso da interface do sistema
	Favoritos	Registrar partes do sistema e do conteúdo que são interesses do usuário
Ajuda	Ajuda multinível	Prover diferentes níveis de ajuda para diferentes usuários
Agregação de comandos	Agregação de comandos	Expressar possíveis ações do usuário através de comandos obtidos a partir da agregação de partes menores

Fonte: Juristo *et al.* (2007b), Marques *et al.* (2017), adaptada pelo autor.

5 TRABALHOS RELACIONADOS

Foi realizada uma revisão bibliográfica da literatura em busca de trabalhos que apresentassem padrões para o design da usabilidade na arquitetura do software. Durante esta etapa, foi encontrada a revisão sistemática de Vilela *et al.* (2015) identificou trabalhos que relacionem a usabilidade e a arquitetura de software.

A revisão sistemática de Vilela *et al.* (2015) foi realizada nas bases eletrônicas *Science Direct* e *IEEEExplore* e foram utilizadas as técnicas de busca automática e a técnica *snowball* (análise das referências presente nos estudos selecionados na pesquisa automática). A *string* de busca utilizada na revisão foi baseada nas palavras-chave: “usabilidade” e “arquitetura de software” e foi aplicada diretamente no *abstract* dos trabalhos. A busca automática resultou em 171 artigos. Os critérios de exclusão usados na revisão foram trabalhos que: (i) não estivessem escritos na língua inglesa; (ii) não estivessem disponíveis através da rede CIN/UFPE; (iii) não relacionavam a usabilidade com a arquitetura de software e (iv) apresentassem resultados incompletos ou que (v) não fossem artigos científicos. Após a aplicação desses critérios, a base foi reduzida a 17 artigos. Posteriormente, foi realizada a técnica *snowball*, resultando na inclusão de mais 5 estudos. Logo, a revisão apresentou 22 estudos relevantes sobre a relação da usabilidade com a arquitetura de software.

Kaartinen *et al.* (2007) apresentam um modelo de processo de design orientado por padrão genérico, e como aplicar a usabilidade neste padrão, obtendo assim um modelo de processo de design centrado na usabilidade. Para o estudo de caso, foi elaborado uma nova arquitetura para um sistema de máquinas operacionais. A nova arquitetura foi projetada com foco na modificabilidade e usabilidade.

Os padrões relacionados à usabilidade utilizados no estudo de caso foram Architecture Sensitive Usability Patterns (ASUP) proposto por Folmer e Bosch (2005) e Usability Supporting Architecture Patterns (USAP) proposto por Bass e John (2003), e o Model-View-Controller (MVC) foi selecionado como o principal estilo arquitetônico do sistema. As ASUPs representam um conjunto de padrões existentes, identificadas a partir da análise de diferentes estudos de caso (FOLMER; BOSCH, 2005). As USAPs descrevem um problema de usabilidade e um conjunto de responsabilidades que devem ser cumpridas para solucionar esse problema (BASS; JOHN, 2003). No projeto do sistema foram utilizados parte da documentação das USAPs, e para medir a capacidade de suporte a usabilidade, foi utilizada o método SALUTA. O método SALUTA permite a medição da capacidade em que a arquitetura suporta a usabilidade.

Foram usados sete cenários do método para avaliar a arquitetura. Os resultados da avaliação indicaram que quase todos os cenários foram fortemente aceitos pelos padrões.

Seffah *et al.* (2008) identificaram e modelaram cenários específicos que ilustram como os componentes de software podem afetar a usabilidade do sistema. Para cada um dos cenários propostos é sugerido um padrão de design de software existente ou aprimorado. Esses padrões são documentados e sua aplicação em um modelo de arquitetura *Model-view-controller* (MVC) é detalhada. Quando identificados esses cenários, os autores propõem soluções baseadas em recomendações de usabilidade apresentados na literatura. Também é proposto um modelo das relações causa-efeito entre elementos de software e a usabilidade, identificando os tipos mais prováveis de relações entre a usabilidade e arquitetura de software.

Bass e John (2003) identificam em 27 cenários quais os possíveis benefícios de usabilidade que um usuário poderia obter e qual padrão arquitetural suportaria esse benefício. Os autores organizaram os cenários em duas hierarquias emergentes, na primeira hierarquia são apresentados os benefícios potenciais para os usuários do sistema, e na segunda hierarquia as táticas arquiteturais utilizada em padrões de suporte. O objetivo do trabalho é apresentar técnicas que permitem identificar problemas de usabilidade que devem ser abordados de maneira proativa no tempo de design da arquitetura, em vez de retroativamente após o teste do usuário.

Juristo *et al.* (2007b) realizaram uma análise de diferentes recomendações de usabilidade e qual o impacto que elas teriam no design de sistemas. Essas recomendações de usabilidade são nomeadas em seu estudo como Functional Usability Features (FUF). O estudo foi realizado em diferentes sistemas orientados a objetos, na qual foram aplicadas diferentes FUFs em seu design. Para medir o impacto que a inserção das FUFs teria no design do sistema, foram analisadas as seguintes métricas: (i) FUF-Functionality – número de funcionalidades afetadas pela FUF (em termos de casos de uso expandidos); (ii) FUF-Classes – número de classes criadas no design como resultado da adição de uma FUF; (iii) FUF-Methods Complexity – nível de complexidade em que os métodos são criados a partir de uma FUF; (iv) FUF- Interaction – nível de acoplamento das classes derivadas de uma FUF com as outras classes de domínio. Na Tabela 2 são apresentados os resultados obtidos através da análise do impacto médio que a inserção das FUFs afetou no design do sistema.

Com base nos resultados da Tabela 2, os arquitetos de software, em conjunto com os stakeholders, podem ter uma base para determinar quais FUFs irão exigir um maior esforço para ser inserida no sistema e conseqüentemente definir quais delas serão incorporadas. Nesse

Tabela 2 – Impacto médio das FUFs no design do sistema

FUF	FUF- Functionality	FUF- Classes	FUF- Methods Complexity	FUF-Interaction
Feedback	Alto 90%	Baixo 27%	Médio	Médio / Alto 66%
Desfazer	Médio 40%	Baixo 10%	Alto	Médio / Alto 66%
Cancelar	Alto 95%	Baixo 8%	Alto	Médio / Alto 66%
Prevenção e correção de erros de entrada de dados	Médio 36%	Baixo 11%	Médio	Baixo 6%
Wizard	Baixo 7%	Baixo 10%	Baixo	Alto 70%
Perfil do usuário	Baixo 8%	Médio 37%	Médio	Baixo 10%
Ajuda	Baixo 7%	Baixo 6%	Baixo	Alto 68%
Uso de diferentes idiomas	Médio 51%	Baixo 10%	Médio	Alto 70%
Alerta	Baixo 27%	Baixo 7%	Baixo	Médio / Alto 66%

Fonte: Juristo *et al.* (2007b).

sentido, o estudo fornece evidências quantitativas de que (i) a usabilidade pode ser descrita em termos de aspectos funcionais que não impactam somente a interface, mas também a interação do usuário com o sistema e (ii) os artefatos de design devem representar aspectos de usabilidade, refutando a visão de que a usabilidade afeta somente a interface do usuário (JURISTO *et al.*, 2007b; MARQUES *et al.*, 2017). Entretanto, o trabalho não apresenta evidências experimentais de como esses aspectos de usabilidade podem ser modelados no design da arquitetura.

Carvajal *et al.* (2013) elaboraram um conjunto de diretrizes para a inclusão de FUFs no processo de desenvolvimento do software. As autoras apresentam meta-modelos para o design de 11 mecanismos de usabilidade, são eles: **abortar, desfazer, agregação de comandos, feedback do progresso, feedback do status do sistema, alerta, wizard, ajuda multinível, favoritos, área de objetos pessoais, preferências, passo a passo**. Para cada mecanismo de usabilidade são apresentados meta-modelos em diagramas de casos de uso, classes e sequência da UML. As autoras realizaram um experimento com a aplicação dessas diretrizes no desenvolvimento de um software. Os resultados preliminares do estudo indicaram a redução de tempo de desenvolvimento e a redução da complexidade percebida das FUFs.

Marques *et al.* (2017) propõe um modelo para representação visual dos mecanismos de usabilidade, assim como da interação e navegação de sistemas interativos. Esse modelo é denominado *USability-oriented INteraction and Navigation model* (USINN) e tem como objetivo representar diversos elementos da interação e navegação do sistema como, por exemplo,

ações e transições. Além disso, o modelo permite a representação de mecanismos de usabilidade como, por exemplo, o feedback do progresso de uma tarefa e alertas sobre o status do sistema.

Com o objetivo de avaliar e refinar o modelo, as autoras realizaram um estudo empírico com graduandos de um curso de Análise e Desenvolvimento de Sistemas. O experimento foi dividido em duas etapas. Na primeira etapa os participantes usaram o USINN para modelar a interação e navegação de um sistema e, na segunda, como base para construir *mockups*. Os resultados do estudo demonstram a viabilidade de usar o modelo e oportunidades para melhorar a facilidade do seu uso.

De forma geral, os trabalhos identificados não apresentam evidências experimentais de como o *design* da usabilidade pode ser incorporado ao *design* da arquitetura de software. Além disso, não são realizados estudos exploratórios sobre a incorporação de mecanismos de usabilidade durante o *design* arquitetural.

6 ESTUDO EXPLORATÓRIO

O estudo exploratório foi conduzido com o intuito de investigar como os mecanismos de usabilidade podem ser incorporados ao *design* da arquitetura do software. Para a realização deste estudo foram conduzidas as seguintes etapas: (1) planejamento; (2) preparação; (3) especificação de requisitos de usabilidade; (4) modelagem arquitetural e (5) coleta da percepção dos participantes.

O estudo foi realizado em ambiente acadêmico com a participação de 50 graduandos em Engenharia de Software, matriculados na disciplina de Arquitetura de Software. Os estudantes conduziram projetos da arquitetura de aplicações *web* e/ou *mobile*, incorporando, nesses projetos, requisitos funcionais de usabilidade identificados.

6.1 Planejamento

O estudo exploratório foi conduzido ao longo de um projeto prático da disciplina de Arquitetura de Software. O projeto requisitado pela disciplina exigia a realização das seguintes etapas: 1) definição do escopo e requisitos das aplicações; 2) seleção de padrões arquiteturais; 3) modelagem arquitetural; 4) avaliação da qualidade da arquitetura. No entanto, o foco deste estudo estava voltado principalmente para as etapas (1) e (3) do projeto, visto que as atividades de elicitación e especificação de requisitos funcionais de usabilidade e sua incorporação no *design* da arquitetura estavam diretamente relacionadas a essas etapas.

Diante desse cenário, este estudo pôde ser estruturado nas seguintes etapas: 1) preparação; 2) especificação de requisitos de usabilidade; 3) modelagem arquitetural; 4) coleta da percepção dos participantes. Na qual as etapas (2) e (3) do estudo exploratório foram realizadas em paralelo as etapas (1) e (3) do projeto da disciplina. Nas próximas seções estão detalhadas cada etapa do estudo.

6.2 Preparação

Com a identificação de trabalhos que relacionem a usabilidade e a arquitetura de software, foi realizada uma seleção dos trabalhos que apresentaram padrões, diretrizes ou modelos de como representar a usabilidade no design da arquitetura. Dentre os trabalhos relacionados que foram analisados (Capítulo 5), o trabalho apresentado por Carvajal (2012) foi considerado para a aplicação no estudo, visto que seu trabalho apresenta diretrizes e meta-

modelos para a incorporação dos diferentes mecanismos de usabilidade para algumas das visões arquiteturais. Além disso, os meta-modelos utilizam da UML para representar as decisões de design arquitetural, o que difere de alguns outros trabalhos. Com isso, foi produzido um material de apoio para os participantes da pesquisa (Apêndice B e Apêndice C). O material aborda adaptações nas diretrizes propostas por Carvajal (2012) para a língua portuguesa e melhorias nas resoluções gráficas dos meta-modelos.

Inicialmente, todos os participantes se organizaram em equipes com 5 ou 6 integrantes. Cada equipe foi organizada por afinidade, não havendo influência dos pesquisadores. Todas as equipes estavam responsáveis por idealizar e projetar arquiteturas de aplicações *web* e/ou *mobile*.

As aplicações teriam por objetivo auxiliar os novos habitantes da cidade de Russas. Tendo em vista que Russas é uma cidade universitária, durante todos os semestres letivos há uma grande número de novos habitantes, como funcionários e estudantes, que enfrentam problemas em encontrar informações sobre a cidade. Portanto, as aplicações teriam como objetivo auxiliar os novos moradores a encontrar, por exemplo, imóveis para aluguel, pontos de lazer, supermercados, ou algumas outras problemáticas enfrentadas por esses habitantes.

Para identificar as oportunidades de desenvolvimento de aplicações para esse contexto, cada uma das equipes realizou atividades iniciais da metodologia de *Design Thinking* (DT). O DT é uma técnica centrada no usuário, que tem por objetivo impulsionar aspectos para a inovação de produtos, considerando métodos como a observação, aprendizado e prototipação de ideias (BROWN, 2008; MARQUES *et al.*, 2015). Essa técnica é desenvolvida em três etapas, a (i) imersão, (ii) ideação e a (iii) prototipagem (MARQUES *et al.*, 2015). Entretanto, durante o estudo, as equipes realizaram apenas as duas fases iniciais da técnica DT, a imersão e a ideação. Na etapa inicial de imersão, os participantes executaram atividades para o entendimento inicial do problema a ser abordado. Durante essa etapa, cada equipe realizou pesquisas exploratórias para identificar o público-alvo e suas necessidades. Após a etapa de imersão, na fase de ideação, as equipes realizaram um *brainstorm* para reunir ideias que levem a uma solução eficaz para o problema.

Na Figura 2 são apresentadas as subdivisões das equipes e seus respectivos integrantes. Os códigos "E" e "P", contidos na figura, referenciam respectivamente a equipe e o participante do estudo. Esses códigos foram criados para preservar a identidade dos participantes e apoiar a contextualização dos resultados que serão apresentados posteriormente.

Figura 2 – Imagem com a distribuição dos participantes por equipe



Fonte: Elaborada pelo autor.

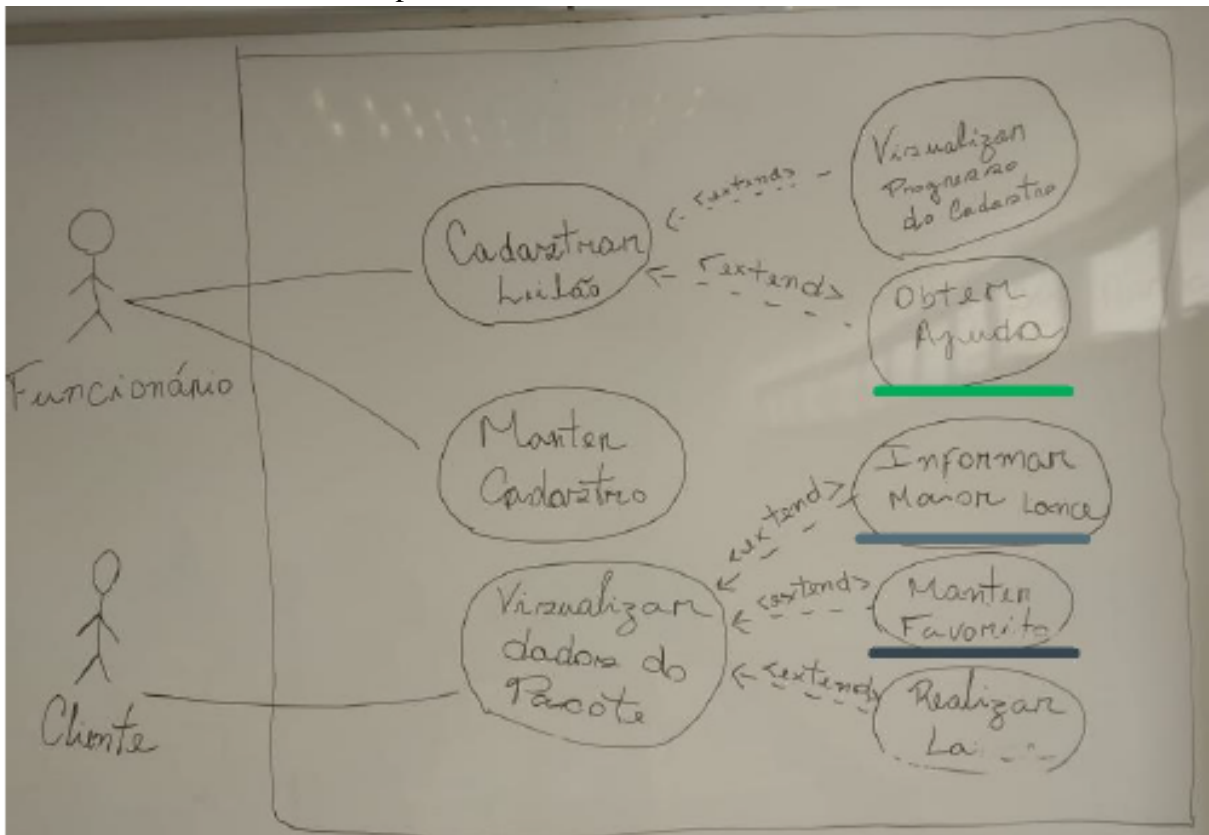
6.3 Especificação de requisitos de usabilidade

Após obter uma ideia sobre qual aplicação projetar, cada equipe realizou atividades de elicitação e especificação de requisitos. Foi requisitado para as equipes que considerassem a usabilidade durante todo o projeto da aplicação.

Com intuito de familiarizar as equipes, foi conduzida uma aula, com duração de 2 horas, sobre o *design* orientado à usabilidade, abordando as FUFs e os mecanismos de usabilidade. Para direcionar a fase de elicitação e especificação dos requisitos de usabilidade, foram apresentados os *guidelines* propostos por Juristo *et al.* (2007a). As equipes poderiam usar as diretrizes como base para elicitação dos requisitos funcionais de usabilidade. Após essa aula, os alunos tiveram um período de 20 dias para a entrega dos requisitos e decisões sobre o uso de padrões arquiteturais.

Em seguida, realizamos um treinamento sobre modelagem arquitetural durante seis aulas, intercalando aulas de modelagem tradicional (sem mecanismos de usabilidade) e modelagem da usabilidade (meta-modelos de Carvajal e colegas). Contando da primeira aula sobre modelagem arquitetural, os alunos tiveram 29 dias para a entrega da modelagem das diferentes visões arquiteturais de seus projetos. Na Figura 3 é apresentada uma das soluções desenvolvidas pela equipe E08 durante o treinamento.

Figura 3 – Fotografia com a representação de mecanismos de usabilidade em diagrama de caso de uso durante a etapa de treinamento



Fonte: Lavínia Matoso. Acervo pessoal.

6.4 Modelagem arquitetural

Com os requisitos especificados, as equipes puderam realizar a modelagem arquitetural de suas aplicações, incorporando os aspectos funcionais da usabilidade indicados como requisitos para suas aplicações. Nesse sentido, foram conduzidas aulas expositivas sobre as diretrizes e meta-modelos propostos por Carvajal *et al.* (2013). Foram duas aulas que ocorreram de forma intercalada com aulas de modelagem tradicional (sem mecanismos de usabilidade), inicialmente uma aula sobre modelagem tradicional e a outra aula sobre modelagem com as diretrizes. Durante as aulas ofertadas pela disciplina, foram apresentados os diagramas da notação UML direcionados a modelar as diferentes perspectivas arquiteturais como a estrutura, o comportamento e a execução. Em seguida, durante as aulas sobre modelagem arquitetural considerando a usabilidade, foram apresentados os meta-modelos para os diagramas de casos de uso e de classes da UML e, além disso, foram apresentados cenários onde as diretrizes poderiam ser incorporadas.

Para auxiliar no uso dos meta-modelos apresentados por Carvajal *et al.* (2013), foram

elaborados dois documentos auxiliares. Nesses documentos estão contidos os meta-modelos e suas descrições traduzidas para a língua portuguesa, visando facilitar o entendimento pelos estudantes. No primeiro documento estavam descritos os meta-modelos para diagrama de caso de uso e suas descrições (Apêndice A). No segundo documento, os meta-modelos para diagrama de classes (Apêndice B). Além disso, também foram disponibilizados as apresentações de *slide* usadas nas aulas sobre as diretrizes. Essas apresentações continham uma breve descrição de cada caso de uso e de cada classe.

As equipes ficaram livres para usar ou não as diretrizes durante o *design* da arquitetura de suas aplicações, entretanto, em ambos os casos os participantes deveriam apresentar a motivação de suas decisões. Além disso, os requisitos de usabilidade especificados deveriam ser modelados nas soluções arquiteturais, com ou sem a adoção dos meta-modelos.

Após as aulas expositivas e a disponibilização dos documentos, foi disponibilizado um período de três semanas para as equipes realizarem as atividades de modelagem arquitetural das aplicações. Cada uma das equipes elaborou um documento contendo os requisitos funcionais e não funcionais de suas aplicações, além dos requisitos funcionais de usabilidade.

Durante as aulas da disciplina foi requisitado a representação da arquitetura utilizando o modelo 4 + 1 proposto por Kruchten (1995). O modelo é subdividido em cinco visões: (i) visão lógica; (ii) visão de desenvolvimento; (iii) visão de processos; (iv) visão física e (v) visão de cenários. Cada visão pode ser representada por diagramas da UML e abrange um conjunto específico de preocupações/interesses.

No entanto, para a análise dos dados do estudo o foco estava apenas nas visões de cenário e a visão lógica, tendo em vista que essas são contempladas pelas diretrizes da Carvajal *et al.* (2013). Portanto, foram analisados apenas os diagramas de casos de uso e o diagramas de classes modelados pelas equipes.

Contando a partir da primeira aula sobre modelagem arquitetural, os alunos tiveram um período de 29 dias para a entrega da modelagem das diferentes visões arquiteturais de seus projetos.

6.5 Coleta da percepção dos participantes

Ao final de todas as etapas, foi realizada uma pesquisa de opinião com os participantes. A pesquisa foi conduzida durante uma das aulas que ocorreram após a entrega do documento arquitetural pelas equipes. As questões foram apresentadas no quadro branco, onde

cada estudante respondeu em uma folha de papel e puderam entregar o seu relato ao fim da aula. Foi esclarecido que a percepção dos estudantes não ia interferir nas notas das disciplinas. Nessa pesquisa de opinião foram apresentadas as seguintes questões:

1. Qual sua percepção sobre o uso das diretrizes de usabilidade para o projeto arquitetural do software?
2. O que influenciaria na sua decisão para adotar ou não as diretrizes de usabilidade em um projeto de arquitetura de software?
3. Qual sua percepção sobre a facilidade de utilizar no projeto arquitetural de software?
4. O que poderia impactar de maneira positiva na facilidade de uso das diretrizes?
5. Quais aspectos de usabilidade as diretrizes de usabilidade permitem representar nas visões arquiteturais?

O documento de projeto arquitetural e os relatos dos participantes foram recolhidos e foi realizada análises sobre esses dados. Os resultados dessas análises são discutidas nos próximos capítulos.

7 ANÁLISE QUANTITATIVA

Nesta seção estão detalhados os resultados da análise quantitativa que foi realizada visando identificar fatores como: o percentual de equipes que utilizaram as diretrizes apresentadas no treinamento; número de equipes que apresentaram soluções alternativas ou que não conseguiram modelar determinado mecanismo de usabilidade com ou sem o uso das diretrizes.

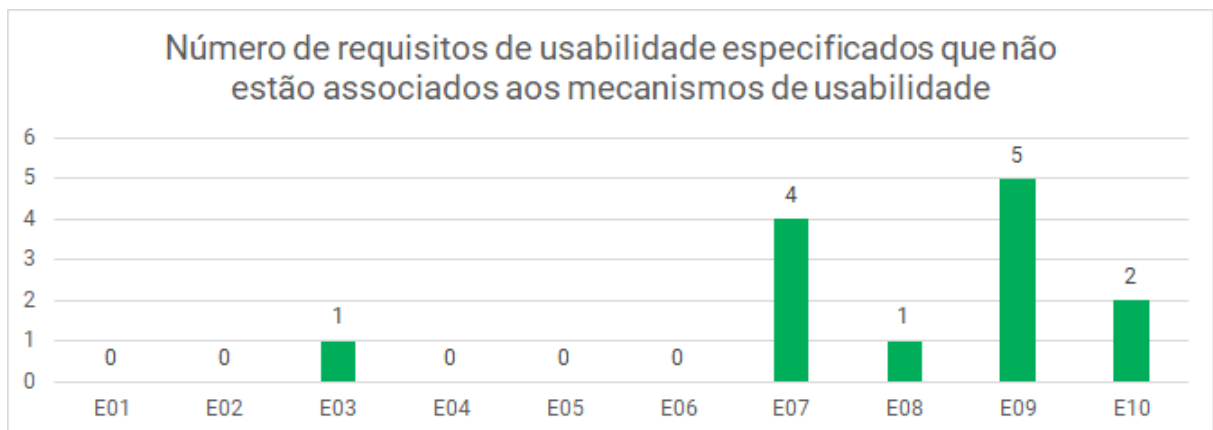
Os resultados dessa análise visam contribuir com a identificação de mecanismos de usabilidade que não são facilmente incorporados e que merecem uma avaliação mais cuidadosa durante o projeto. Além disso, os dados irão contribuir para a análise de viabilidade do uso das diretrizes em outros projetos.

Inicialmente foi realizada uma análise na lista de requisitos funcionais da usabilidade especificados por cada equipe. Os resultados indicaram que algumas equipes especificaram requisitos que não estão associados a aspectos funcionais da usabilidade e que estavam muitas vezes associados a aspectos da usabilidade relacionados à interface do sistema.

Um dos requisitos de usabilidade especificado pela equipe E07, por exemplo, indicou que *"a opção de busca por locais, setores ou funcionalidades no app, deve ser feita ao clicar em uma lupa, já que o usuário já possui a experiência, pois a busca em outros sistema usa a imagem da lupa"*. No entanto, esse tipo de requisito não aborda nenhum dos aspectos de usabilidade apresentados por Juristo *et al.* (2007b) e está relacionado com aspectos da interface com o usuário, sem afetar diretamente a arquitetura do sistema.

Na Figura 4 estão apresentados os resultados da análise dos requisitos de usabilidade que não estão associados aos mecanismos.

Figura 4 – Gráfico com o número de requisitos de usabilidade especificados que não estão associados aos mecanismos de usabilidade

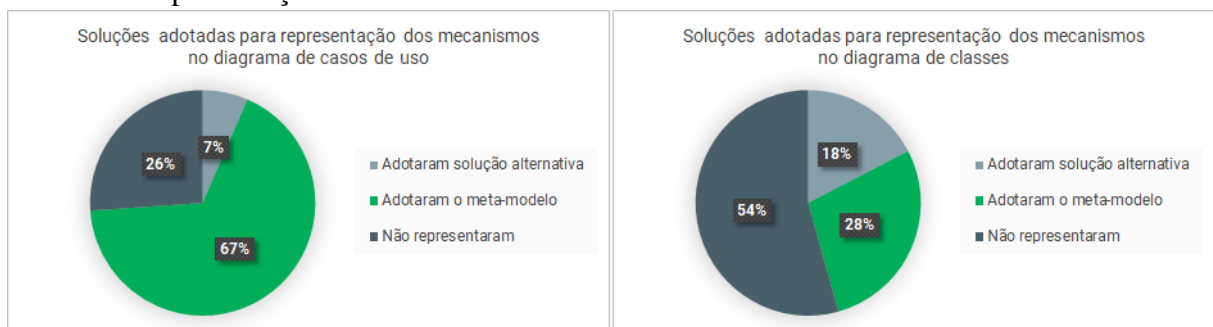


Fonte: Elaborada pelo autor.

Esses resultados indicam que os mecanismos de usabilidade ainda não são tão bem compreendidos e existe a necessidade de abordar mais sobre o assunto com os participantes e com a comunidade de Engenharia de Software de forma geral.

Na Figura 5 estão descritos o resultados do percentual de uso das diretrizes e da adoção de soluções alternativas para os requisitos de usabilidade especificados. Os resultados indicaram que 67% dos requisitos especificados foram representados nos diagramas de casos de uso com o uso dos meta-modelos, no entanto, apenas 28% foram representados desta forma no diagrama de classes. Em contrapartida, a porcentagem de não representação dos mecanismos aumentou de 26%, no diagrama de casos de uso, para 56%, no diagrama de classes. Da mesma forma, a porcentagem de uso de soluções alternativas aumentou de 7%, no diagrama de casos de uso, para 18%, no diagrama de classes.

Figura 5 – Gráfico com a porcentagem de uso dos meta-modelos e de soluções alternativas para representação da usabilidade



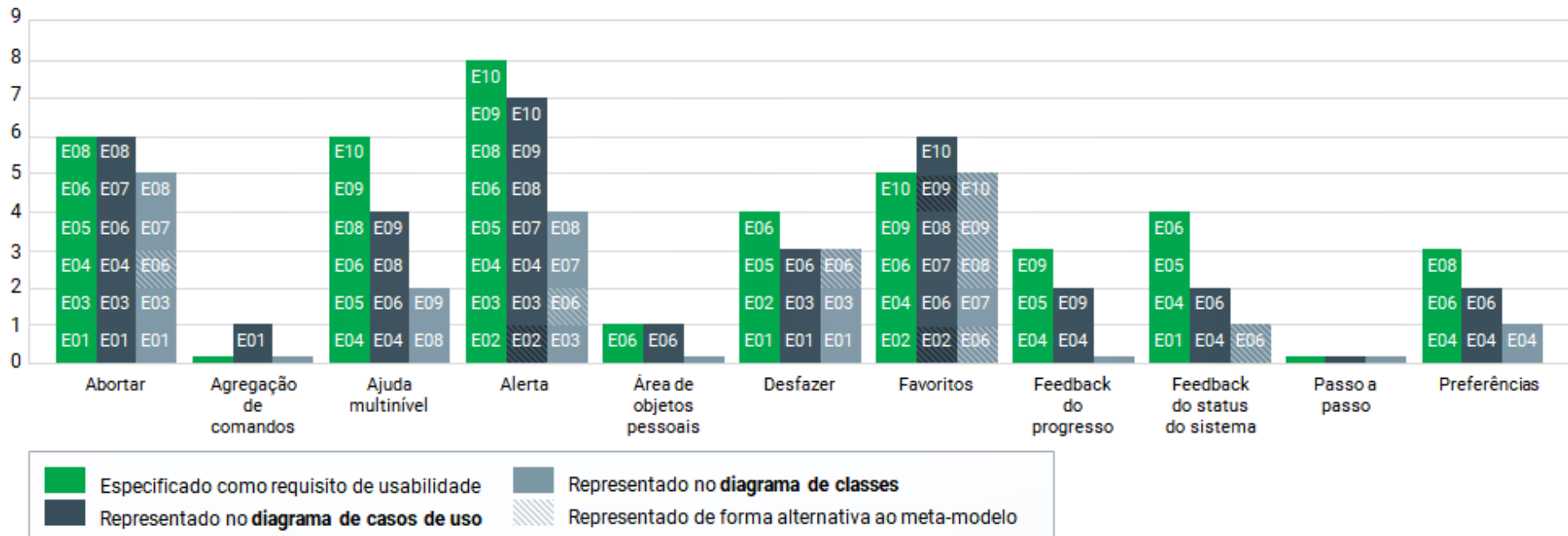
Fonte: Elaborada pelo autor.

Esses resultados podem indicar que os meta-modelos para casos de uso são bem aceitos e apoiam a representação dos mecanismos de usabilidade, entretanto, para o diagrama de classes, ainda existe um receio em adota-los implicando na adoção de soluções alternativas ou da não representação dos mecanismos explicitamente. Além disso, os resultados também indicaram que o mecanismo mais especificado foi o **alerta**, seguido dos mecanismos **abortar** e **ajuda multinível**. Em contrapartida, os mecanismos **agregação de comandos** e **passo a passo** não foram especificados por nenhuma das equipes.

Na Figura 6 é apresentado um gráfico contendo os dados obtidos pela análise quantitativa. Os dados estão distribuídos por mecanismos de usabilidade. No gráfico, são indicadas as equipes que especificaram o mecanismo como requisito de usabilidade, representaram no diagrama de casos de uso e de classes, além de quais representações foram adotadas soluções alternativas.

Figura 6 – Gráfico com os resultados da análise quantitativa

Gráfico com os resultados da análise quantitativa



Fonte: Elaborada pelo autor.

Mesmo sendo o mecanismo mais representado, o **alerta** teve a maior redução proporcional de não representação no diagrama de classes, seguido pelo mecanismo **ajuda multinível**, em um total de 8 equipes especificaram e apenas 4 representaram no diagrama de classes e 6 para 2 respectivamente. Esses dados apoiam os resultados obtidos pelo estudo de Juristo *et al.* (2007b), descritos na Tabela 2. Os resultados do estudo de Juristo *et al.* (2007b) indicaram que os mecanismos **alerta** e **ajuda multinível** são os que possuem o menor impacto nas classes do projeto, eles afetam as classes do projeto em 7% e 6% respectivamente. Portanto, a não representação desses mecanismos pode ser justificada pelo pouco impacto que possuem nas classes e possivelmente este seja o motivo de não serem representados por estas equipes.

Outro ponto observado foi que o mecanismo **favoritos** teve o maior número de representações de forma alternativa ao meta-modelo. Essas formas alternativas indicam que os participantes optaram em não usar as soluções propostas pelas diretrizes e optaram em elaborar uma própria solução para representação dos mecanismos. Esses resultados mostram indícios de que as equipes adotaram uma nova solução provavelmente por razões relacionadas a complexidade de como o meta-modelo representa o mecanismo ou por uma solução mais simples e melhor aceita. Essas soluções alternativas serão melhor discutidas no Capítulo 8.

Quanto aos mecanismos **agregação de comandos**, **área de objetos pessoais** e **passo a passo**, a pouca ou nenhuma representação pode estar relacionada com o contexto específico na qual esses mecanismos são incorporados, assim como o contexto das aplicações que foram projetadas pelos participantes.

De forma geral, os resultados dessa análise demonstram que os aspectos funcionais da usabilidade são comuns em diversas aplicações e, quando especificados como requisitos funcionais, tendem a ser representados na arquitetura. Pôde-se perceber que a representação dos mecanismos nos diagramas é bem aceita, visto que ocorreram casos onde as equipes representaram mecanismos que não estavam especificados como requisitos de suas aplicações. Além disso, o uso dos meta-modelos para diagramas de casos foram bem aceitos e que para diagramas de classes parece ainda existir receios quanto ao uso. Para isso, foi realizada uma análise das soluções arquiteturais adotadas por cada equipe. Os resultados dessa análise estão detalhados no próximo capítulo.

8 ANÁLISE DAS SOLUÇÕES ARQUITETURAIS PARA MODELAGEM DE ASPECTOS DE USABILIDADE

Algumas equipes elaboraram soluções alternativas ao uso do meta-modelo proposto por Carvajal *et al.* (2013), apresentado durante o período de treinamento. Nesse sentido, foi efetuada uma análise nos documentos arquiteturais elaborados por cada equipe.

Inicialmente, foram identificados os requisitos de usabilidade especificados para cada projeto. Em seguida, foi realizada uma análise nos diagramas de casos de uso e classes, visando identificar se a equipe representou explicitamente o requisito de usabilidade em ambos os diagramas. Com isso, foi identificado se a representação usada pela equipe condiz com o meta-modelo apresentado no treinamento.

As soluções alternativas ao meta-modelo são descritas nesta seção, assim como os relatos de alguns dos participantes sobre a motivação para adotar ou não o meta-modelo. Para isso, são apresentados, para cada equipe, duas tabelas contendo os requisitos de usabilidade identificados e qual deles foram representados nos diagramas.

Para a tabela com os requisitos de usabilidade, são apresentados os requisitos especificados e qual mecanismo de usabilidade associado. Em alguns casos, foram identificados mecanismos de usabilidade representados que não foram especificados. Esses mecanismos são apresentados nas tabelas como requisitos, no entanto, sua especificação não foi descrita e foi indicada pelo símbolo (-).

Para a tabela que indica os requisitos que a equipe representou no diagrama de classes ou casos de uso, os requisitos são categorizados de três formas: "Sim", para aqueles que usaram o meta-modelo para representar o requisito; "Não", para os que usaram uma solução alternativa ao meta-modelo e, "Não representou", para aqueles que não representaram o requisito de nenhuma forma.

8.1 Solução arquitetural para a representação da usabilidade no projeto da equipe E01

O projeto proposto pela equipe E01 é o desenvolvimento de uma aplicação mobile nomeada por SOU NOVO AQUI. A aplicação tem por objetivo auxiliar novos moradores da cidade de Russas a encontrar imóveis que estejam disponíveis para o aluguel. Em um contexto geral, a aplicação irá disponibilizar uma lista com dados de imóveis da região que estejam disponíveis para serem alugados e os usuários da aplicação poderão registrar os seus interesses nas ofertas que mais lhe agradam.

Na Tabela 3 estão apresentados os requisitos funcionais de usabilidade especificados pela equipe E01 e qual o respectivo mecanismo de usabilidade associado a cada um deles. A equipe especificou três requisitos de usabilidade associados aos mecanismos de **feedback do status do sistema, desfazer e abortar**.

Tabela 3 – Requisitos de usabilidade especificados pela equipe E01

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E01-RU01	"Fornecer Feedback de operações: O sistema deve fornecer informação clara quando uma operação é realizada ou não"	Feedback do status do sistema
E01-RU02	"Fornecer operação de desfazer: Para qualquer operação do sistema, o sistema deve fornecer ao usuário a funcionalidade de desfazer uma operação realizada"	Desfazer
E01-RU03	"Cancelar operação: O sistema deve ter uma funcionalidade de cancelar uma operação em andamento"	Abortar

Fonte: O Autor.

Na Tabela 4 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 4 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E01

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E01-RU01	Não representou	Não representou
E01-RU02	Sim	Sim
E01-RU03	Sim	Sim

Fonte: O Autor.

Dentre os requisitos especificados, o requisito E01-RU01, associado ao mecanismo **feedback do status do sistema**, não foi representado em nenhum dos diagramas. No entanto, os requisitos E01-RU02 (**desfazer**) e E01-RU03 (**abortar**) foram representados com características similares aos apresentados no meta-modelo, havendo apenas simplificações em alguns casos de uso e classes que já são previstas pelas diretrizes.

Com a análise dos relatos dos integrantes da equipe E01, foi possível validar que as diretrizes foram realmente usadas apenas com simplificações necessárias para o contexto da

aplicação projetada pela equipe. O participante P03, integrante da equipe E01, indicou que *"os componentes do meta-modelo) que foram utilizados, foram simplificados de acordo com as guidelines"*. Entretanto, não foram identificados relatos do integrantes sobre a não representação do requisito E01-RU01.

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação do requisito E01-RU01 é o esquecimento de representar o mecanismo, pelo fato de ainda não ser comum sua representação pelos participantes, ou a mudança nos requisitos e a não atualização da documentação.

8.2 Solução arquitetural para a representação da usabilidade no projeto da equipe E02

O projeto proposto pela equipe E02 é o desenvolvimento de uma aplicação mobile nomeada por HELLO RUSSAS. A aplicação tem por objetivo auxiliar novos moradores da cidade de Russas a encontrar estabelecimentos comerciais da cidade. O administrador da aplicação será responsável por registrar os estabelecimentos e categorizá-los, além de registrar anúncios sobre eles. Os novos moradores poderão buscar e visualizar os estabelecimentos cadastrados e favoritar aqueles que mais lhe agradam.

Na Tabela 5 estão apresentados os requisitos funcionais de usabilidade especificados pela equipe E02 e qual o respectivo mecanismo de usabilidade associado a cada um deles. A equipe especificou três requisitos de usabilidade associados aos mecanismos de **alerta, favoritos e desfazer**.

Tabela 5 – Requisitos de usabilidade especificados pela equipe E02

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E02-RU01	"O sistema deve notificar o usuário assim que um anúncio for cadastrado"	Alerta
E02-RU02	"O sistema deve permitir ao usuário criar uma lista de estabelecimentos favoritos"	Favoritos
E02-RU03	"O sistema deve permitir ao usuário retornar à páginas anteriores"	Desfazer

Fonte: O Autor.

Na Tabela 6 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 6 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E02

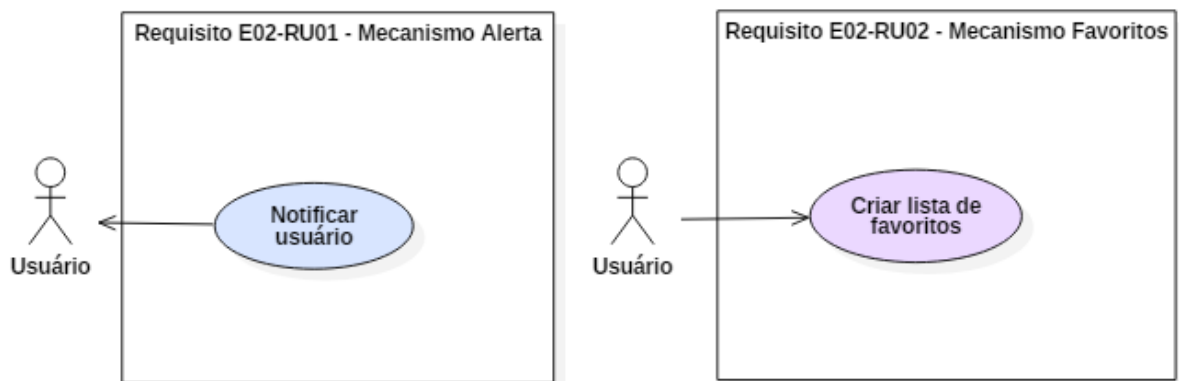
Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E02-RU01	Não	Não representou
E02-RU02	Não	Não representou
E02-RU03	Não representou	Não representou

Fonte: O Autor.

Dentre os requisitos especificados, o requisito E02-RU03, associado ao mecanismo **desfazer**, não foi representado em nenhum dos diagramas. Além disso, a equipe não representou nenhum dos requisitos de usabilidade no diagrama de classes do projeto. No entanto, os requisitos E02-RU01 e E02-RU02 foram representados no diagrama de casos de uso de forma alternativa ao meta-modelo.

A Figura 7 apresenta as soluções alternativas elaboradas pela equipe E02. Para o requisito E02-RU01 (**alerta**), a equipe optou em utilizar apenas um único caso de uso com a informação "notificar usuário", indicando que o usuário receberá uma notificação/alerta quando um anúncio for cadastrado. Da mesma forma, para o requisito E02-RU02 (**favoritos**), a equipe usou apenas um caso de uso com a informação "criar lista de favoritos", sugerindo que o usuário possa criar uma lista com todos os estabelecimentos categorizados como favorito.

Figura 7 – Diagramas com as soluções alternativas propostas pela equipe E02



Fonte: Elaborada pelo autor.

Com o intuito de identificar as motivações que levaram os integrantes da equipe E02 a adotar soluções alternativas e não representar alguns dos requisitos especificados, foi realizada uma análise nos relatos dos participantes, obtidos pelo questionário aplicado durante o estudo.

O participante P05, integrante da equipe E02, indicou que "os (componentes do

meta-modelo) que não foram ideais não colocamos, pois tentamos fazer um projeto bem simples de acordo com o que foi pedido, ou seja, as funcionalidades que iam deixar mais complexo e que poderia fugir do escopo inicial foi descartado"

Não foram identificados relatos sobre a não representação do requisito E02-RU03. Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação dos requisitos de usabilidade no diagrama de classes e do requisito E02-RU03 no diagrama de casos de uso, podem estar associadas ao receio em aumentar a complexidade dos diagramas, como foi relatado pelo integrante P05, ou o fato de não possuir experiência com a representação da usabilidade em projetos arquiteturais. Além disso, o esquecimento e a falta de atualização na documentação também são fatores que podem ter ocasionado a não representação dos requisitos nesses diagramas.

8.3 Solução arquitetural para a representação da usabilidade no projeto da equipe E03

O projeto proposto pela equipe E03 é o desenvolvimento de um sistema WEB nomeado por RUSSAS TOUR. O sistema tem por objetivo auxiliar novos moradores da cidade de Russas a encontrar estabelecimentos comerciais da cidade. O administrador da aplicação será responsável por manter o registro dos estabelecimentos e categorizá-los. Os usuários do sistema poderão buscar e visualizar os estabelecimentos cadastrados, favoritar aqueles que mais lhe agradam e avaliar o estabelecimento de acordo com o seu grau de satisfação.

Na Tabela 7 estão apresentados os requisitos funcionais de usabilidade especificados pela equipe E03 e qual o respectivo mecanismo de usabilidade associado a cada um deles.

Tabela 7 – Requisitos de usabilidade especificados pela equipe E03

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E03-RU01	"O sistema deve abrir uma tela informando que o cadastro feito com sucesso"	Alerta
E03-RU02	"O usuário poderá editar seus comentários sobre os locais"	-
E03-RU03	"Cancelar cadastro e exclusão"	Abortar
E03-RU04	-	Desfazer

Fonte: O Autor.

A equipe especificou dois requisitos funcionais de usabilidade associados aos me-

canismos **alerta** e **abortar**. Foi observado que um requisito associado a ação de editar foi especificado como requisito funcional de usabilidade, no entanto, não está associado a um dos mecanismos de usabilidade indicado por Juristo *et al.* (2007b). Além disso, a equipe representou o mecanismo de usabilidade **desfazer** em seus diagramas, mas não o especificou no documento.

Na Tabela 8 estão apresentados os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 8 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E03

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E03-RU01	Sim	Sim
E03-RU02	-	-
E03-RU03	Sim	Sim
E03-RU04	Sim	Sim

Fonte: O Autor.

Dentre os requisitos especificados, o E03-RU02 não está associado à um mecanismo de usabilidade, portanto, não foi considerado na análise de soluções alternativas. No entanto, todos os outros requisitos de usabilidade foram representados de acordo com o meta-modelo apresentado.

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a especificação do requisito E03-RU02 como requisito funcional de usabilidade, pode estar associado ao fato de não possuir experiência com a abordagem de mecanismos de usabilidade.

8.4 Solução arquitetural para a representação da usabilidade no projeto da equipe E04

O projeto proposto pela equipe E04 é o desenvolvimento de uma aplicação mobile nomeada por VAI E VEM RUSSAS. A aplicação tem por objetivo auxiliar os moradores da cidade de Russas a encontrar anúncios sobre produtos ofertados na região. Em um contexto geral, a aplicação irá disponibilizar o registro, busca e classificação de anúncios de produtos. Os anúncios são categorizados por tipo, e podem ser favoritos e avaliados pelos usuários. Além disso, a aplicação irá dispor de um chat para a comunicação entre o usuário interessado e o anunciante e, para os anunciantes, será disponibilizado um sistema de premiação, na qual ele receberá premiações pela popularidade do seu anúncio publicado.

Na Tabela 9 estão apresentados os requisitos funcionais de usabilidade especificados

pela equipe E04 e qual o respectivo mecanismo de usabilidade associado a cada um deles. A equipe especificou seis requisitos de usabilidade associados aos mecanismos de **feedback do status do sistema, feedback do progresso, ajuda multinível, abortar, alerta, favoritos e preferências**.

Tabela 9 – Requisitos de usabilidade especificados pela equipe E04

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E04-RU01	"O sistema deve manter os usuários informados sobre o status do sistema, pois caso a conexão do cliente com o servidor caia o status do sistema irá mudar de online para offline, e em casos de operações feitas pelo servidor que sejam demoradas o sistema deve informar que estar processando algo"	Feedback do status do sistema e feedback do progresso
E04-RU02	"O sistema deve fornecer diferentes níveis de ajuda a seus usuários, diminuindo assim a chance de erro no uso da aplicação, e permitindo que a aprendizagem do usuário seja mais rápida"	Ajuda multinível
E04-RU03	"O sistema deve fornecer a opção de cancelar qualquer operação que esteja fazendo, oferecendo mais controle ao usuário na aplicação"	Abortar
E04-RU04	"O sistema deve informar aos usuários consequências de ações que possam ter grande impacto sobre suas ações atuais ou no sistema como um todo"	Alerta
E04-RU05	"O sistema irá permitir que os usuários adicionem determinadas categorias ou anúncios aos favoritos para que eles possam ser notificados de possíveis alterações do conteúdo dos anúncios ou novos anúncios de determinada categoria"	Favoritos
E05-RU06	"O usuário poderá definir algumas configurações básicas sobre a interface do sistema, como o tema escuro ou claro da aplicação, o tamanho da fonte, entre outras"	Preferências

Fonte: O Autor.

Na Tabela 10 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 10 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E04

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E04-RU01	Sim	Não representou
E04-RU02	Sim	Não representou
E04-RU03	Sim	Não representou
E04-RU04	Sim	Não representou
E04-RU05	Não representou	Não representou
E04-RU06	Sim	Não representou

Fonte: O Autor.

Dentre os requisitos especificados, apenas o requisito E04-RU05, associado ao mecanismo de **favoritos**, não foi representado no diagrama de casos de uso. Além disso, a equipe não representou nenhum dos requisitos de usabilidade para o diagrama de classes do projeto.

Não foram identificados relatos sobre a não representação dos requisitos de usabilidade no diagrama de classes, porém, os integrantes da equipe indicaram algumas soluções alternativas para a representação dos mecanismos nas descrições dos casos de uso. Para os integrantes da equipe E04, os mecanismos **abortar** e **feedback do progresso** não precisam ser representados explicitamente no diagrama de casos de uso, sendo indicado apenas como descrições nos fluxos alternativos. Segundo o integrante P19, da equipe E04, *"algumas operações podem ser feitas no fluxo de exceção de uma operação. Além disso, o cancelar (**abortar**) e o **feedback do progresso** também podem ser representados na descrição dos casos de uso, principalmente o cancelar (**abortar**). Para o integrante P21, "o fornecer feedback, **desfazer** e refazer são perfeitamente representados, deixando claro a estrutura necessária para tais requisitos. Em contrapartida, o cancelar e o abortar pode ser representado nos fluxos de exceção (...)"*

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação dos requisitos de usabilidade no diagrama de classes e do requisito E04-RU05 no diagrama de casos de uso, podem estar associadas ao receio em aumentar a complexidade dos diagramas ou o fato de não possuir experiência com a representação da usabilidade em projetos arquiteturais. Além disso, o esquecimento e a falta de atualização na documentação também são fatores que podem ter ocasionado a não representação dos requisitos nesses diagramas.

8.5 Solução arquitetural para a representação da usabilidade no projeto da equipe E05

O projeto proposto pela equipe E05 é o desenvolvimento de uma aplicação mobile nomeada por RUSSAS AP. A aplicação irá permitir que os usuários possam encontrar ou anunciar imóveis para locação ou venda, na cidade de Russas. Além disso, a aplicação também permitirá a utilização de filtros de busca e disponibilizará uma área destinada para os usuários trocarem mensagens com os anunciantes.

Na Tabela 11 estão apresentados os requisitos funcionais de usabilidade especificados pela equipe E05 e qual o respectivo mecanismo de usabilidade associado a cada um deles. A equipe especificou seis requisitos de usabilidade associados aos mecanismos de **feedback do progresso, feedback do status do sistema, alerta, ajuda multinível, desfazer e abortar**.

Tabela 11 – Requisitos de usabilidade especificados pela equipe E05

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E05-RU01	"O sistema deve fornecer feedback sobre o andamento da tarefa atual"	Feedback do progresso
E05-RU02	"O sistema deve informar sobre os seus diferentes estados"	Feedback do status do sistema
E05-RU03	"O sistema deve fornecer alertas para prevenir que o usuário erre"	Alerta
E05-RU04	"O sistema deve fornecer textos de ajuda ao longo da ferramenta"	Ajuda multinível
E05-RU05	"O sistema deve permitir que ações executadas sejam revertidas"	Desfazer
E05-RU06	"O sistema deve possibilitar ações em andamento sejam canceladas"	Abortar

Fonte: O Autor.

Na Tabela 12 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 12 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E05

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E05-RU01	Não representou	Não representou
E05-RU02	Não representou	Não representou
E05-RU03	Não representou	Não representou
E05-RU04	Não representou	Não representou
E05-RU05	Não representou	Não representou
E05-RU06	Não representou	Não representou
E05-RU07	Não representou	Não representou

Fonte: O Autor.

A equipe E05 optou em não representar nenhum dos requisitos de usabilidade. Segundo os integrantes da equipe, as diretrizes eram confusas e aumentariam a complexidade dos diagramas, portanto, não foram usadas. O integrante P22 informou que *"preferia representar as diretrizes de outra forma"* e que *"ficou confuso o conceito das diretrizes"*. Para o integrante P25, *"representar (os requisitos de usabilidade) foi confuso para a equipe, então foi deixado de lado para não prejudicar os demais itens"*. Além disso, o integrante P26 informa que não representou os mecanismos *"devido ao tempo, complexidade e a qualidade"* e, portanto, tiveram que *"focar em funcionalidades do sistema"*.

8.6 Solução arquitetural para a representação da usabilidade no projeto da equipe E06

O projeto proposto pela equipe E06 é o desenvolvimento de uma aplicação mobile nomeada por HOMEMATE APP. A aplicação tem por objetivo auxiliar os novos moradores da cidade de Russas a encontrar moradores que desejam compartilhar um imóvel. Em um contexto geral, a aplicação irá permitir que os usuários possam visualizar o perfil de outros usuários e do imóvel que ele pretende compartilhar. Os usuários poderão favoritar o perfil de outros usuários, caso os dois usuários favoritem uma ao outro (match) o aplicativo irá permitir que eles possam se comunicar por um chat com mensagens instantâneas.

Na Tabela 13 estão apresentados os requisitos de usabilidade especificados pela equipe E06 e qual o respectivo mecanismo de usabilidade associado a cada um deles. A equipe especificou nove requisitos de usabilidade associados aos mecanismos de **desfazer**, **abortar**, **feedback do status do sistema**, **alerta**, **favoritos**, **preferências**, **ajuda multinível** e **área de objetos pessoais**.

Tabela 13 – Requisitos de usabilidade especificados pela equipe E06

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E06-RU01	"O usuário poderá desfazer o match com qualquer usuário em que tenha sido realizado"	Desfazer
E06-RU02	"O sistema deve cancelar alterações realizadas nas informações de moradia, onde as informações que podem ser visualizadas são as que já estavam cadastradas no sistema"	Abortar
E06-RU03	"O sistema deve apresentar o progresso do envio de uma mensagem no chat (enviada, recebida, lida)"	Feedback do status do sistema
E06-RU04	"O sistema deve emitir uma notificação (representando a quantidade) caso hajam mensagens recebidas de outros usuários"	Feedback do status do sistema
E06-RU05	"O sistema deve apresentar as notificações globais (alertas) na barra de notificação do sistema operacional"	Alerta
E06-RU06	"O usuário deve poder definir uma lista de matches de sua preferência, denominado como lista de favoritos"	Favoritos
E06-RU07	"O sistema deve permitir que o usuário informe quais dados pessoais (não obrigatórios) serão visíveis por outros usuários"	Preferências
E06-RU08	O sistema deve apresentar uma área onde o usuário possa acessar um guia que detalha as funcionalidades do sistema"	Ajuda multinível
E06-RU09	"O sistema deve permitir que o usuário reorganize a lista de candidatos que deram match de acordo com sua preferência"	Área de objetos pessoais

Fonte: O Autor.

Na Tabela 14 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

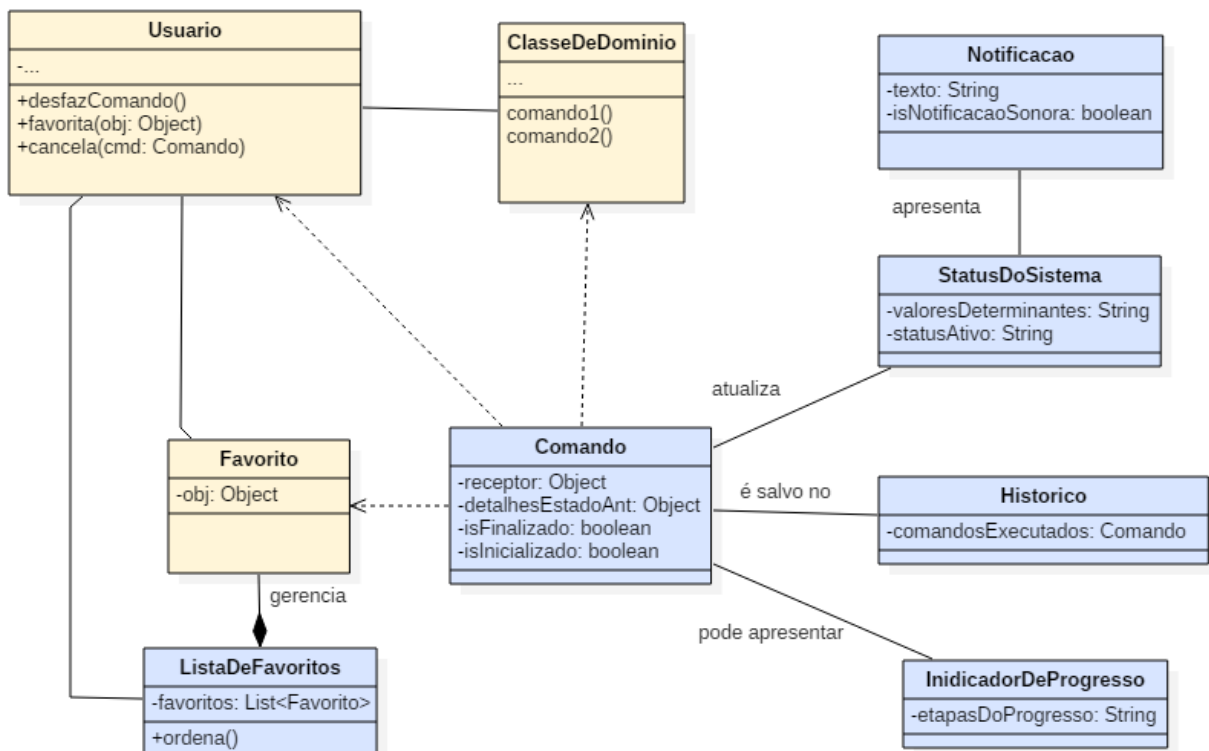
Tabela 14 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E06

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E06-RU01	Sim	Não
E06-RU02	Sim	Não
E06-RU03	Sim	Não
E06-RU04	Sim	Não
E06-RU05	Não representou	Não
E06-RU06	Sim	Não
E06-RU07	Sim	Não representou
E06-RU08	Sim	Não representou
E06-RU09	Sim	Não representou

Fonte: O Autor.

Dentre os requisitos especificados, todos foram representado no diagrama de casos de uso utilizando o meta-modelo. No entanto, a equipe propôs soluções alternativas para a representação dos mecanismos no diagrama de classes. Na Figura 8 é apresentada a solução adotada pela equipe.

Figura 8 – Diagrama com as soluções alternativas propostas pela equipe E06



Fonte: Elaborada pelo autor.

A solução proposta pela equipe para a representação dos mecanismos no diagrama

de classes foi utilizar uma classe denominada "Comando" que irá representar todos os comandos com consequências importantes no sistema. As classes de domínio serão as responsáveis por executar esses comandos. De acordo com o que foi representado pela equipe, cada comando executado poderá atualizar o status do sistema e por sua vez exibir ou não uma notificação. Todo comando realizado é salvo no histórico e pode ou não apresentar um indicador de progresso. Além disso, a solução indicou que os usuários terão uma lista responsável por armazenar todos os seus objetos favoritos.

Ao analisar os relatos dos integrantes da equipe E06, foi identificado que alguns deles citaram a priorização em representar requisitos de usabilidade essenciais. Segundo o integrante P27, *"todos (os requisitos de usabilidade) foram representados em pelo menos uma visão, no entanto, alguns deles não se tornou extremamente necessário representar em outros diagramas"*. O integrante P29 informou que *"todos (os requisitos de usabilidade) foram possíveis de representar, mas o mecanismo alerta foi desconsiderado por conta da priorização dos requisitos de usabilidade"*.

A não representação dos requisitos de usabilidade E06-RU07, E06-RU09 e E06-RU09 no diagrama de classes podem estar associadas a priorização dos requisitos, como informado pelos integrantes, ou o esquecimento e o receio de aumentar a complexidade, visto que um dos integrantes informou que alguns dos mecanismos não tornou-se necessário a representação.

8.7 Solução arquitetural para a representação da usabilidade no projeto da equipe E07

O projeto proposto pela equipe E07 é o desenvolvimento de um aplicativo mobile nomeado por RusSOS. O sistema tem por objetivo auxiliar novos moradores da cidade de Russas a encontrar estabelecimentos comerciais da cidade. O administrador da aplicação será responsável por manter o registro dos estabelecimentos e categorizá-los. Os usuários da aplicação poderão buscar e visualizar os estabelecimentos cadastrados e o ranking deles para cada categoria. Além disso, os usuários poderão favoritar aqueles que mais lhe agradam e avaliar o estabelecimento de acordo com o seu grau de satisfação.

Na Tabela 15 estão apresentados os requisitos de usabilidade especificados pela equipe E07 e qual o respectivo mecanismo de usabilidade associado. A equipe especificou quatro requisitos de usabilidade, porém, foi observado que nenhum desses requisitos retrata uma das características de um mecanismo de usabilidade indicada por Juristo *et al.* (2007b). Entretanto, a equipe representou os mecanismos de usabilidade **favoritos**, **abortar** e **alerta** nos diagramas de

casos de uso e classes.

Tabela 15 – Requisitos de usabilidade especificados pela equipe E07

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E07-RU01	"O login pode ser feito pelo Facebook ou Gmail para facilitar a entrada do usuário no sistema"	-
E07-RU02	"As avaliações de qualidade e preço serão mostradas ao usuário por meio de estrelas para facilitar a visualização, deixando o mesmo mais satisfeito"	-
E07-RU03	"A opção de busca por locais, setores ou funcionalidades no app, deve ser feita ao clicar em uma lupa, já que o usuário já possui a experiência, pois a busca em outros sistema usa a imagem da lupa."	-
E07-RU04	"O menu com os setores dos locais será apresentado como se fosse um catálogo para ser melhor de o usuário visualizar e interagir"	-
E07-RU05	-	Favoritos
E07-RU06	-	Abortar
E07-RU07	-	Alerta

Fonte: O Autor.

Na Tabela 16 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes. Todos os mecanismos representados pela equipe foram baseados nos meta-modelos apresentados, havendo apenas algumas simplificações.

Tabela 16 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E07

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E07-RU01	-	-
E07-RU02	-	-
E07-RU03	-	-
E07-RU04	-	-
E07-RU05	Sim	Sim
E07-RU06	Sim	Sim
E07-RU07	Sim	Sim

Fonte: O Autor.

Segundo o integrante P33, foram consideradas "*apenas 3 (mecanismos), por que foram os que haviam necessidade para o projeto*" e o integrante P36 informa que "*foram permitidas de aplicar o **abortar, favoritos e alerta***".

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não especificação dos requisitos de usabilidade E07-RU05, E07-RU06 e E07-RU07 podem estar associadas ao esquecimento e a falta de atualização na documentação arquitetural da equipe. No entanto, os fatores que podem ter ocasionado a especificação de requisitos não associados aos mecanismos devem estar relacionados a falta de experiência com a especificação dos requisitos funcionais da usabilidade.

8.8 Solução arquitetural para a representação da usabilidade no projeto da equipe E08

O projeto proposto pela equipe E08 é o desenvolvimento de um aplicativo mobile nomeado por **SEGUE SEGURO APP**. A aplicação tem por objetivo auxiliar os usuários a identificar o nível de segurança/insegurança e eventuais incidentes em horários distintos para determinadas rotas de trânsito. As rotas serão classificadas por meio de uma avaliação dos próprios moradores da localidade e os usuários poderão acessar essas informações por uma busca rápida pela rota no aplicativo.

Na Tabela 17 estão apresentados os requisitos de usabilidade especificados pela equipe E08 e qual o respectivo mecanismo de usabilidade associado. A equipe especificou seis requisitos de usabilidade associados aos mecanismos **abortar, alerta, ajuda multinível e preferências**.

Tabela 17 – Requisitos de usabilidade especificados pela equipe E08

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E08-RU01	"O sistema deve possibilitar que o usuário cancele em tempo de processamento o login ou cadastro que esteja efetuando"	Abortar
E08-RU02	O sistema deve averiguar se o usuário deseja salvar eventuais alterações realizadas sempre que o mesmo abortar alguma operação, como quando o usuário estiver editando seus dados pessoais, classificações ou registros já feitos e optar por sair do sistema antes de concluir o processamento da operação/comando"	Alerta
E08-RU03	"O sistema deve auxiliar o usuário no primeiro acesso fornecendo "dicas" pontuais orientando-os sobre como realizar classificações e registros de rotas"	Ajuda multinível
E08-RU04	"O sistema deve emitir um alerta ao usuário quando o mesmo estiver prestes a realizar uma operação que não poderá ser desfeita, no caso, quando o usuário estiver prestes a excluir seu perfil pessoal o sistema deve avisá-lo que se trata de uma decisão irreversível"	Alerta
E08-RU05	"O sistema deve permitir que o usuário altere a busca por uma rota a qualquer momento, para isto, sempre que o usuário interromper esse processo de busca o sistema deve oferecer-lhe a opção de fazer uma nova busca"	-
E08-RU06	"O sistema deve alertar o usuário sempre que forem adicionados em suas rotas salvas (preferenciais) novos registros de incidentes"	Alerta
E08-RU07	"O sistema deve permitir que o usuário altere as configurações padrões, desta forma, permitindo que opcionalmente o usuário bloqueie funções como: alertas de novos registros de incidentes para determinadas rotas em suas preferências"	Preferências
E08-RU08	-	Favoritos

Fonte: O Autor.

Na Tabela 18 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

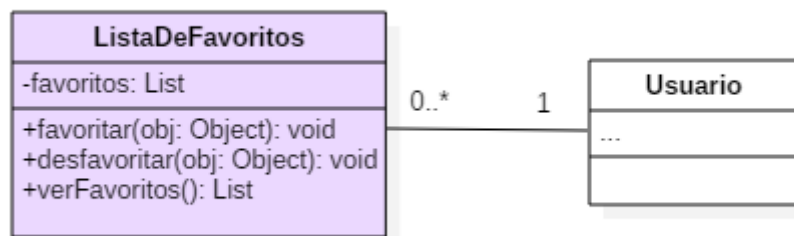
Tabela 18 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E08

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E08-RU01	Sim	Sim
E08-RU02	Sim	Sim
E08-RU03	Sim	Sim
E08-RU04	Sim	Sim
E08-RU05	-	-
E08-RU06	Sim	Sim
E08-RU07	Não representou	Não representou
E08-RU08	Sim	Não

Fonte: O Autor.

Dentre os requisitos especificados, apenas o requisito E08-RU08, associado ao mecanismo de **favoritos**, foi representado no diagrama de classes de forma alternativa ao meta-modelo. A Figura 9 apresenta a solução alternativa adotada pela equipe.

Figura 9 – Diagrama com a solução alternativa proposta pela equipe E08



Fonte: Elaborada pelo autor.

Para representar o mecanismo de **favoritos** a equipe optou em utilizar apenas uma classe com uma lista de objetos favoritados e métodos para o gerenciamento dessa lista. No entanto, não foram identificados relatos do integrantes da equipe sobre o uso dessa abordagem.

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação do requisito de usabilidade E08-RU07, podem estar associadas ao esquecimento e a falta de atualização na documentação arquitetural.

8.9 Solução arquitetural para a representação da usabilidade no projeto da equipe E09

O projeto proposto pela equipe E09 é o desenvolvimento de um aplicativo mobile nomeado por ME GUIA. A aplicação tem por objetivo permitir que novos moradores e visitantes da cidade de Russas tenham acesso à informações gerais sobre a cidade. Os usuários do sistema terão acesso a conteúdos relacionados aos estabelecimentos comerciais, estudantis e governamentais, servindo como um guia rápido online. Além disso, o usuário terá acesso a um fórum para tirar dúvidas, fazer reclamações e apresentar sugestões de estabelecimentos.

Na Tabela 19 estão apresentados os requisitos de usabilidade especificados pela equipe E09 e qual o respectivo mecanismo de usabilidade associado. A equipe especificou treze requisitos de usabilidade associados aos mecanismos **feedback do progresso, alerta, ajuda multinível e favoritos**.

Tabela 19 – Requisitos de usabilidade especificados pela equipe E09

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E09-RU01	"Enquanto o sistema realiza a busca por locais deve ser exibido o progresso da operação"	Feedback do progresso
E09-RU02	"Textos e imagens devem possuir descrições possibilitando a leitura por aplicações de leitura de tela."	-
E09-RU03	"Ao utilizar o mecanismo de busca deve ser apresentado sugestões de acordo com o que é digitado pelo o usuário"	-
E09-RU04	"A transação de uma tela para outra deve ser de apenas 0.5 segundos"	-
E09-RU05	"Ao realizar uma pesquisa o usuário poderá ordenar o resultado da pesquisa em ordem crescente ou decrescente de acordo com nota, popularidade e adicionado recentemente"	-
E09-RU06	"Quando um cadastro de localização é realizado de ser apresentado mensagem de confirmação, sucesso ou falhas"	Alerta
E09-RU07	"Ao Selecionar um item deve aparecer um opção de adicionar o item aos favoritos, onde será inserido em uma lista, onde se encontra todos os favoritos do usuário"	Favoritos
E09-RU08	"Este caso de uso permite que o usuário possa ter acesso a um guia geral de uso das ferramentas e funções dentro do aplicativo"	Ajuda multinível
E09-RU09	"Este caso de uso permite que o usuário através do guia geral do uso das ferramentas e funções que o aplicativo disponibiliza, possa ter acesso a seções específicas de ajuda"	Ajuda multinível
E09-RU10	"Este caso de uso permite que o usuário adicione mais localizações as quais deseja-se ter informações, formando uma lista de suas localizações"	-
E09-RU11	"Este caso de uso é invocado quando o usuário cria uma localização favorita"	Favoritos
E09-RU12	"Este caso de uso permite que o usuário visualize a lista de localizações que ele criou como favoritas"	Favoritos
E09-RU13	"Este caso de uso permite que o usuário remova uma localização da lista de favoritas que ele criou"	Favoritos

Fonte: O Autor.

Na Tabela 20 estão descritos os resultados da análise sobre a verificação da representação dos requisitos de usabilidade nos diagramas de casos de uso e de classes.

Tabela 20 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E09

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E09-RU01	Sim	Não representou
E09-RU02	-	-
E09-RU03	-	-
E09-RU04	-	-
E09-RU05	-	-
E09-RU06	Sim	Não representou
E09-RU07	Não	Não
E09-RU08	Sim	Sim
E09-RU09	Sim	Sim
E09-RU10	-	-
E09-RU11	Não	Não
E09-RU12	Não	Não
E09-RU13	Não	Não

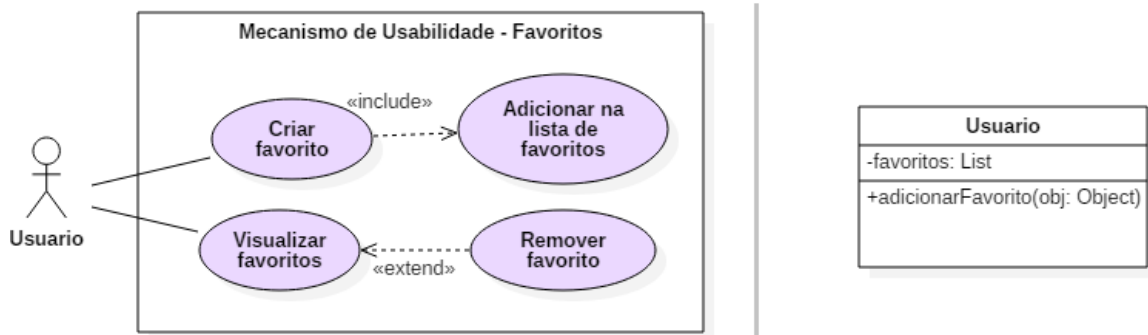
Fonte: O Autor.

Dentre os requisitos especificados, apenas o requisito E08-RU08, associado ao mecanismo de **favoritos**, foi representado de forma alternativa ao meta-modelo. Para representar o mecanismo de **favoritos** no diagrama de classes a equipe optou em utilizar apenas a classe do "Usuario" contendo uma lista com os objetos favoritados e métodos para o gerenciamento dessa lista. Por sua vez, o mecanismo de **favoritos** foi representado no diagrama de casos de uso usando o meta-modelo, porém, foi adicionado o caso de uso "remover favorito", que não está especificado no meta-modelo. A Figura 10 apresenta a solução alternativa adotada pela equipe.

Os relatos dos integrantes da equipe indicaram apenas confirmações sobre a adoção de uma solução alternativa ao meta-modelo para à representação do mecanismo **favoritos**. Segundo o integrante P45, *"no diagrama de classes (o mecanismo de favoritos) foi representado com um único atributo, uma lista"*.

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação dos requisitos de usabilidade E09-RU01 e E09-RU06, podem estar associadas ao esquecimento e a falta de atualização na documentação arquitetural.

Figura 10 – Diagramas com as soluções alternativas propostas pela equipe E09



Fonte: Elaborada pelo autor.

8.10 Solução arquitetural para a representação da usabilidade no projeto da equipe E10

A sistema WEB proposto pela equipe E10 tem por objetivo permitir que os novos moradores e visitantes da cidade de Russas tenham acesso a localização de estabelecimentos comerciais da região. O sistema também irá permitir que os usuários possam anunciar o seu estabelecimento/serviço e avaliar outros.

Na Tabela 21 estão apresentados os requisitos de usabilidade especificados pela equipe E10 e qual o respectivo mecanismo de usabilidade associado. A equipe especificou cinco requisitos de usabilidade associados aos mecanismos **abortar**, **alerta**, **ajuda multinível** e **favoritos**. Além disso, a equipe representou o mecanismo de *agregação de comandos* no diagrama de casos de uso, entretanto, não especificou como requisito de usabilidade.

Tabela 21 – Requisitos de usabilidade especificados pela equipe E10

ID	Requisito de usabilidade especificado	Mecanismo de usabilidade associado
E10-RU01	"O usuário deverá ser capaz de editar suas publicações mesmo antes de serem analisadas pelo moderador, em caso de erros"	-
E10-RU02	"No cadastro de uma publicação ou de um usuário, deve existir um botão 'cancelar'"	-
E10-RU03	"Deverá ser informado para o usuário quando uma publicação sua é aprovada pela moderação, alterada ou excluída"	Alerta
E10-RU04	"O sistema deverá ter uma página com a legenda dos ícones"	Ajuda multinível
E10-RU05	"O sistema deve permitir que o usuário possa marcar seus lugares favoritos"	Favoritos
E10-RU06	-	Agregação de comandos

Fonte: O Autor.

Na Tabela 22 estão descritos os resultados da análise sobre a verificação da representação do requisito de usabilidade nos diagramas de casos de uso e de classes.

Tabela 22 – Resultado da análise sobre uso dos meta-modelos nos diagramas da equipe E10

Requisito de usabilidade	Usou o meta-modelo no diagrama de casos de uso?	Usou o meta-modelo no diagrama de classes?
E10-RU01	-	-
E10-RU02	-	-
E10-RU03	Sim	Não representou
E10-RU04	Não representou	Não representou
E10-RU05	Sim	Não
E10-RU06	Sim	Não representou

Fonte: O Autor.

Dentre os requisitos especificados, apenas o requisito E10-RU05 (**favoritos**) foi representado de forma alternativa ao meta-modelo. Assim como a equipe E09, a equipe E10 optou em representar o mecanismo de **favoritos** no diagrama de classes usando apenas a classe do "Usuario" contendo uma lista com os objetos favoritados e métodos para o gerenciamento dessa lista.

Os relatos dos integrantes indicaram apenas confirmações sobre a adoção de uma

solução alternativa ao meta-modelo para à representação do mecanismo **favoritos**. Para os participantes P48 e P50, os mecanismos "*alerta, favoritos e agregação de comandos*" foram possíveis de representar.

Alguns dos possíveis motivos, levantados pelos pesquisadores deste estudo, para a não representação do requisito de usabilidade E10-RU04 em ambos os diagramas e a não representação dos requisitos E10-RU03, E10-RU04 e E10-RU06 podem estar associadas ao esquecimento e a falta de atualização na documentação arquitetural. Além disso, a falta de experiência e de motivação podem ter ocasionado a não representação dos mecanismos, visto que os diagramas continham diversos defeitos de sintaxe.

9 ANÁLISE QUALITATIVA DOS RELATOS DOS PARTICIPANTES

Nesta seção estão detalhados os resultados da análise qualitativa. A análise qualitativa, utilizando o método de Grounded Theory (STRAUSS; CORBIN, 1990), foi realizada com base nas respostas obtidas por um questionário sobre o uso das diretrizes (Apêndice A). Dentre os 50 participantes da pesquisa, apenas 42 deles responderam ao questionário. As questões analisadas foram:

1. Qual sua percepção sobre o uso das diretrizes de usabilidade para o projeto arquitetural do software?
2. O que influenciaria na sua decisão para adotar ou não as diretrizes de usabilidade?
3. Qual sua percepção sobre a facilidade de utilizar no projeto arquitetural de software?
4. O que poderia impactar de maneira positiva na facilidade de uso das diretrizes?

Para cada questão foi realizada uma análise seguindo as etapas de codificação aberta (identificação de códigos e categorias), codificação axial (relacionamento entre categorias e subcategorias) e seletiva (identificação de categorias centrais da teoria). A codificação aberta envolve a divisão, análise, comparação e categorização dos dados. Inicialmente, na codificação aberta, o pesquisador realiza uma exploração dos dados com um exame detalhado do que se considera relevante ao assunto por meio de uma leitura intensiva dos textos transcritos. Na codificação axial, os códigos são agrupados em categorias através do comparação entre os códigos (MENDES *et al.*, 2018).

Para auxiliar na realização deste processo, foi utilizado o software de análise de dados qualitativos Atlas.ti¹. Além disso, para validar os códigos resultantes, um pesquisador realizou a codificação e um segundo pesquisador validou os resultados para identificar se os códigos realmente refletiam os dados coletados.

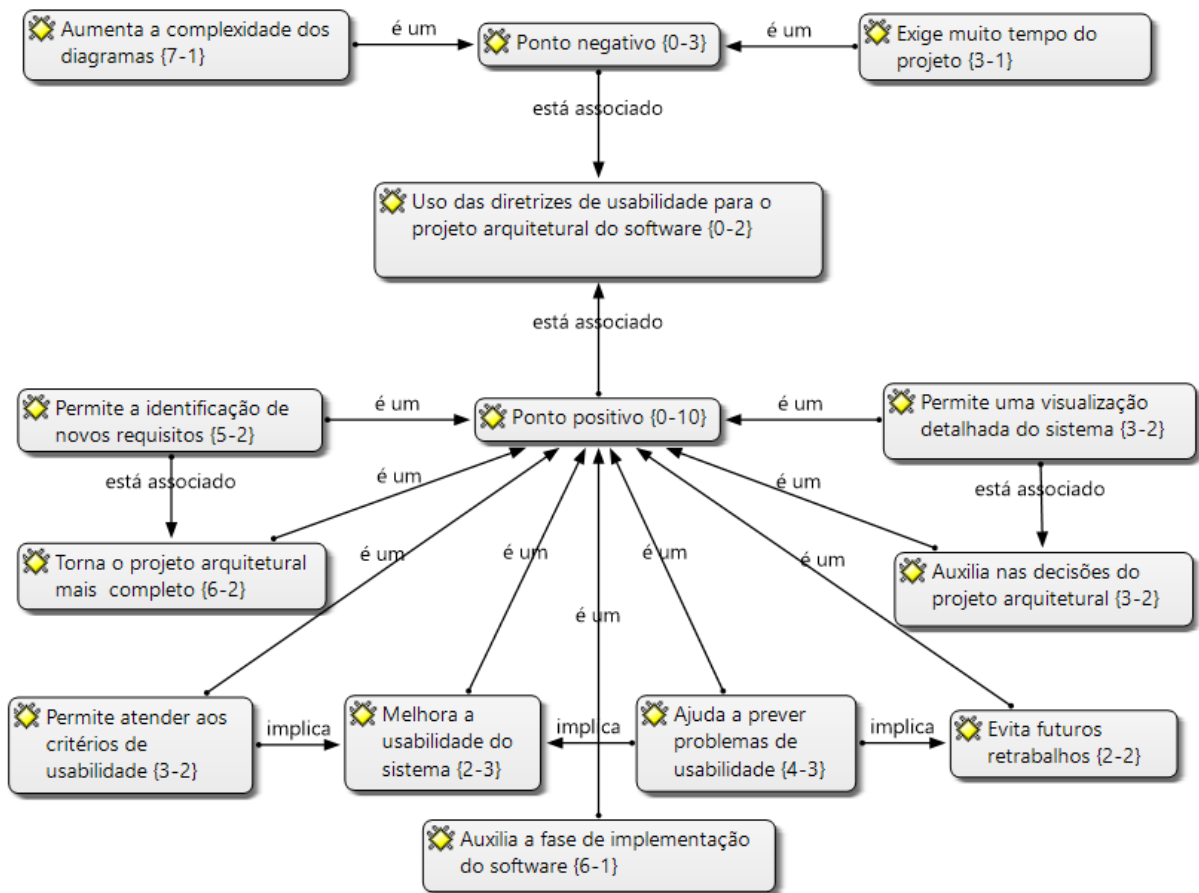
Nas subseções seguintes são apresentados os resultados para cada questão. Os códigos são descritos seguidos de dois números que representam respectivamente o grau de fundamentação (número de citações do código) e o de densidade teórica (número de relacionamentos do código com outros códigos).

¹ Link: <https://atlasti.com/>

9.1 Percepção sobre o uso das diretrizes de usabilidade para o projeto arquitetural do software

Nesta seção estão apresentados os resultados obtidos para a análise da opinião dos participantes do estudo sobre o uso das diretrizes propostas por Carvajal *et al.* (2013). Na Figura 11 é apresentado a rede de visualização dos códigos e categorias obtidos pela análise.

Figura 11 – Rede de visualização sobre a opinião do uso das diretrizes no projeto arquitetural



Fonte: Elaborada pelo autor.

Os relatos dos participantes sobre o uso das diretrizes foram classificados em duas categorias, pontos positivos e negativos. Foram identificados nove códigos associados à categoria de pontos positivos e dois códigos para a categoria de pontos negativos. Os principais pontos negativos identificados indicaram que o uso das diretrizes aumenta a **complexidade dos diagramas** e **exige muito tempo do projeto** para a sua incorporação. Em contrapartida, os pontos positivos apontam diversos benefícios quanto ao uso das diretrizes. Os principais pontos positivos citados pelos participantes indicaram que o uso das diretrizes **auxilia a fase de implementação, torna o projeto arquitetural mais completo, permite a identificação de**

novos requisitos e ajuda a prever problemas de usabilidade.

Dentre os códigos identificados, foram encontrados relacionamentos entre alguns deles. Os códigos **permite atender aos critérios de usabilidade** e **ajuda a prever problemas de usabilidade** implicam em **melhorar a usabilidade do sistema**, visto que quando os critérios de usabilidade são atingidos e os futuros problemas são previstos, os problemas com o uso do sistema tendem a ser reduzidos, conseqüentemente melhorando a usabilidade do sistema.

As principais vantagens citadas sobre o uso das diretrizes está associada ao apoio durante as fases iniciais do desenvolvimento, a etapa de elicitação de requisitos, projeto e implementação do software. Associado ao código **permite a identificação de novos requisitos**, P04 afirmou que *"obter detalhes de decisões arquiteturais antes do desenvolvimento/implementação do sistema, possibilita listar de forma prematura funcionalidades não levantadas na etapa de requisitos de software"*. Quanto ao código sobre o **apoio na fase de implementação**, P50 afirma que as diretrizes *"auxiliam os desenvolvedores a terem uma visão melhor do que deve ser feito"*. Para P44, o uso das diretrizes *"é útil para tornar de fácil implementação a usabilidade de um sistema, mesmo que sua modelagem seja um pouco complexa"*.

Dentre os códigos categorizados como pontos negativos, foram identificados sete relatos sobre o uso das diretrizes **aumentar a complexidade dos diagramas**. Dentre esses relatos, P25 afirma que *"embora facilitem no desenvolvimento, (as diretrizes) deixam complexo a modelagem"*. No mesmo sentido, P24 indicou que *"perde-se muito tempo modelando (...) e acaba aumentando a complexidade dos diagramas"*

Para alguns dos participantes, as diretrizes **tornam o projeto arquitetural mais completo**. P07 citou que *"a principal utilidade é que padroniza os diversos projetos arquiteturais que o engenheiro venha a fazer, evitando o esquecimento de alguma funcionalidade e permitindo a elaboração de um projeto arquitetural completo com funcionalidade essenciais, importantes e até desejáveis"*.

Relacionado ao código **ajuda a prever problemas de usabilidade**, P46 relatou que *"ao utilizar essas diretrizes, um ponto positivo é que ela previne erros futuros, bem como erros com a programação"*. Nesse mesmo sentido, P07 afirmou que o uso das diretrizes *"impacta diretamente na ausência de erros, onde a lista de funcionalidades de usabilidade lhe faça lembrar de todas as funções (...)"*.

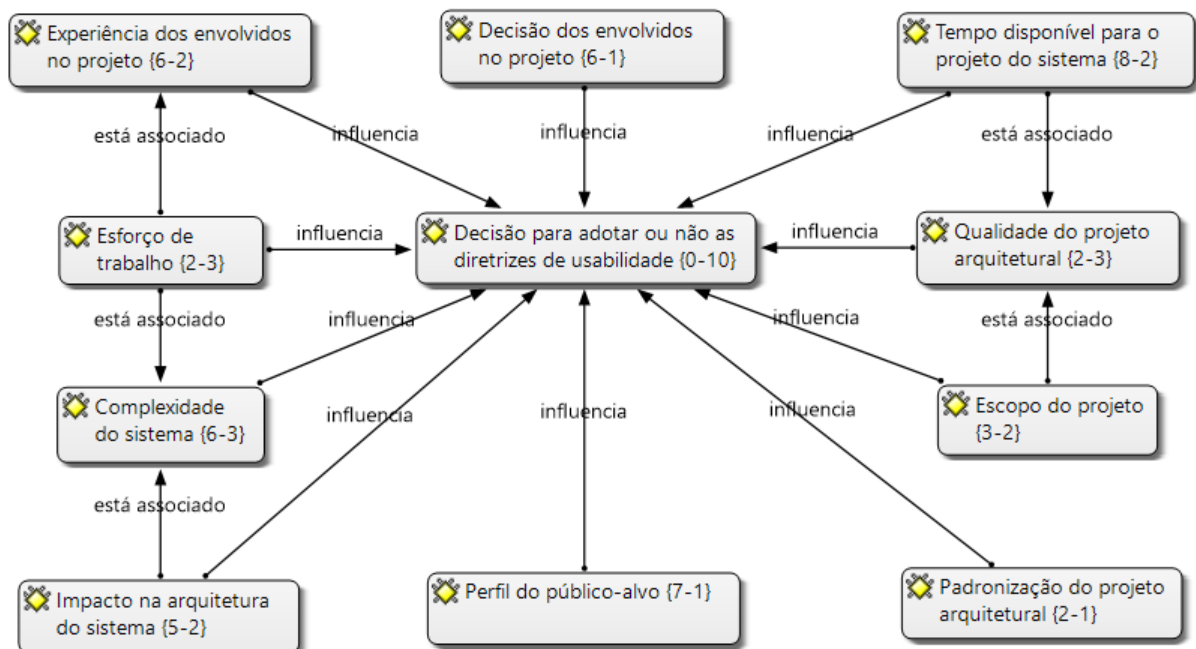
De forma geral, as opiniões dos participantes indicaram que o uso das diretrizes apoia as fases iniciais de desenvolvimento e tornam o projeto arquitetural mais completo,

permitindo atender os critérios que irão prover uma boa qualidade de uso ao sistema. No entanto, é necessário mais tempo para projeto arquitetural e o uso das diretrizes pode aumentar a complexidade percebida dos diagramas.

9.2 O que influenciaria na decisão para adotar ou não as diretrizes de usabilidade

Nesta seção estão apresentados os resultados obtidos para os principais fatores, indicados pelos participantes, que influenciariam na tomada de decisão sobre o uso ou não das diretrizes proposta por Carvajal *et al.* (2013). Na Figura 12 é apresentada a rede de visualização dos códigos e categorias obtidos pela análise.

Figura 12 – Rede de visualização sobre a influência na decisão por adotar ou não as diretrizes



Fonte: Elaborada pelo autor.

Os principais fatores que influenciam na tomada de decisão sobre o uso das diretrizes, indicados pelos participantes, foram: **o tempo disponível para o projeto do sistema; o perfil do público-alvo; a experiência dos envolvidos no projeto; a complexidade do sistema e a decisão dos envolvidos no projeto.**

Os participantes relataram que adotar as diretrizes pode exigir muito tempo com o design e, portanto, **o tempo disponível para o projeto do sistema** é um dos principais fatores que motivariam a adota-las ou não no projeto. Dentre os relatos, P47 indicou que *"a falta de tempo incentivaria a não usar (as diretrizes)"*. Já P48 *"consideraria (as diretrizes) se o*

cronograma do projeto permitir" e P29 indicou que *"provavelmente, adotaria uma parte das diretrizes que parecem mais úteis"*.

Outro fator bastante citado nos relatos sobre a influência no uso ou não das diretrizes é o **perfil do público-alvo**. Os participantes indicaram que suas decisões são influenciadas pela quantidade e perfil dos usuários finais do sistema. P47 relatou que *"muitos usuários com diferentes papéis incentivam a usar (as diretrizes)"* e no mesmo sentido, P41 afirma que *"o que mais influenciaria seria o público-alvo do projeto"*. Complementarmente, P01 detalha que *"a facilidade de uso do sistema e como ele se comporta diante das diferentes categorias de pessoas que irão utilizar"* é também um fator que afeta na tomada da decisão.

Além disso, a **experiência dos envolvidos no projeto** também é um fator que influencia na tomada de decisão sobre o uso das diretrizes. Os relatos indicaram que a experiência da equipe de desenvolvimento em usar as diretrizes ou em incorporar a usabilidade durante o projeto arquitetural pode influenciar de maneira positiva em querer adotar as diretrizes no projeto. Segundo os participantes, *"o conhecimento dos membros envolvidos no projeto sobre os mecanismos de usabilidade e sobre a modelagem"* - P10, além da *"capacidade dos desenvolvedores em implementar uma boa interface"* - P20 são variáveis que devem ser observadas antes de adotar as diretrizes. Logo, a *"experiência e maturidade da equipe"* - P13 deve ser analisada durante a tomada da decisão.

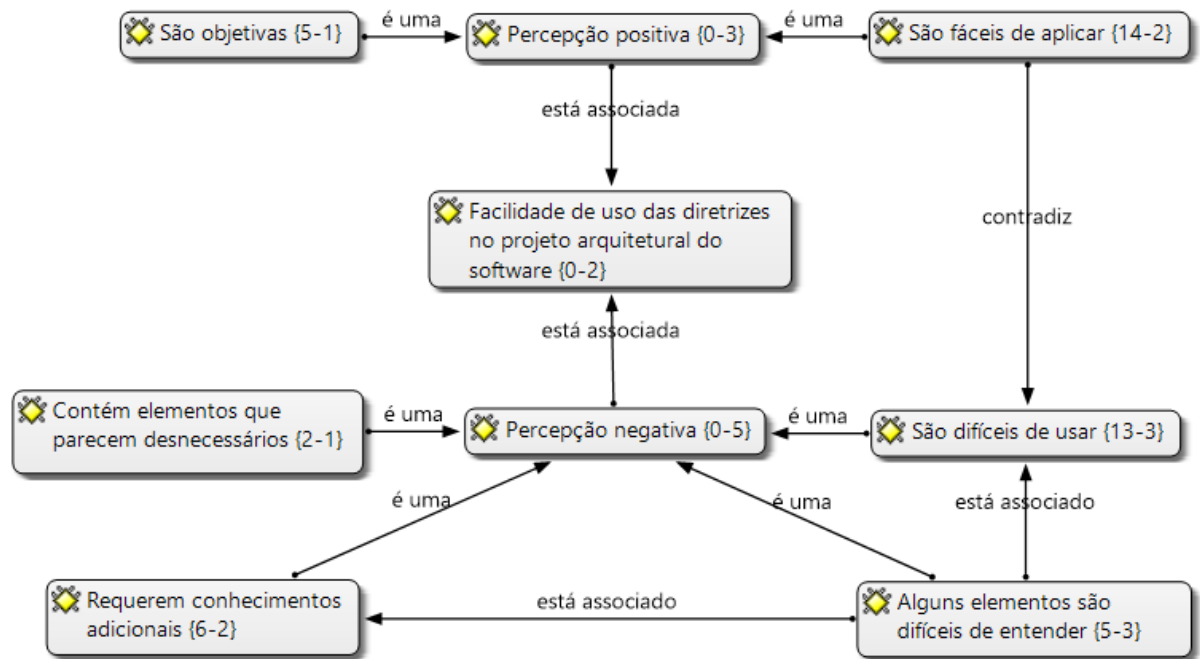
Da mesma forma, a **complexidade do sistema** e a **decisão dos envolvidos no projeto** irá impactar diretamente na escolha em adotar as diretrizes. O participante P35 relatou que *"se o projeto for simples e pequeno não acho que seja necessário o uso das diretrizes, mas se for algo maior e complexo é muito importante até mesmo para manter a usabilidade no projeto, além da sua organização"*.

Outros fatores que influenciariam no tomada de decisão em adotar ou não as diretrizes são: o **impacto na arquitetura do sistema**; o **escopo do projeto**; a **padronização e qualidade do projeto arquitetural** e o **esforço de trabalho**.

9.3 Percepção sobre a facilidade de utilizar as diretrizes no projeto arquitetural de software

Nesta seção estão apresentados os resultados obtidos para a análise da percepção dos participantes sobre a facilidade de uso das diretrizes no projeto arquitetural. Na Figura 13 é apresentada a rede de visualização dos códigos e categorias obtidos pela análise.

Figura 13 – Rede de visualização sobre o impacto positivo na facilidade de uso das diretrizes



Fonte: Elaborada pelo autor.

Os participantes relataram percepções positivas e negativas sobre o uso das diretrizes. Os principais pontos positivos identificados indicaram que a facilidade de uso das diretrizes está associada ao fato de serem **objetivas** e **fáceis** de aplicar. Em contrapartida, foram identificados quase a mesma quantidade de relatos que afirmam que as diretrizes são **difíceis de usar** devido a fatores como a **difículdade de entender alguns elementos** e **requerem conhecimentos adicionais**.

Dentre os códigos da categoria de **percepção positiva**, relacionada ao código **são fáceis de aplicar**, P21 afirmou que "*a utilização das diretrizes*) foi simples e não envolveu algo muito complexo". P27 também afirmou que "*as diretrizes são facilmente entendidas e aplicadas e, portanto, não exige um grande esforço*". Para o código **são objetivas**, P09 indicou que "*são objetivas, o que facilita a utilização*".

Para os códigos pertencentes a categoria de **percepção negativa**. Associados ao código **são difíceis de usar**, P19 informou que "*por ser minha primeira vez usando, não foi fácil de utilizar, tendo que estava acostumando só com as funcionalidade do sistema e não na usabilidade*". Para P33, "*é fácil de entender, porém na prática é diferente (...)*". P47 relata que "*não é uma tarefa trivial e é melhor executado quando existe experiência por parte dos envolvidos*".

Para o código **alguns elementos são difíceis de entender**, P44 citou que "*algumas*

(das diretrizes) são intuitivas, outras são difíceis de entender (...)". No mesmo sentido, P50 relatou que "algumas delas são fáceis de representar através dos diagramas, outras não".

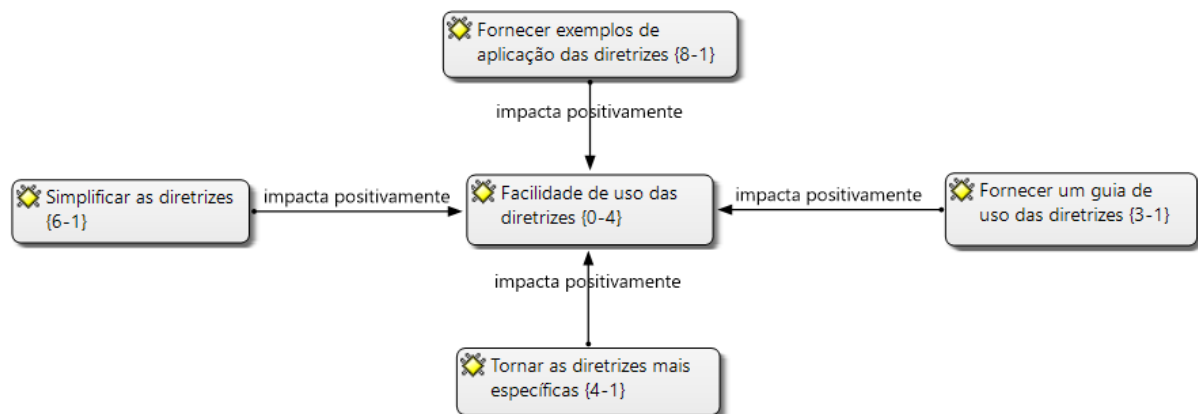
Dentre os relatos associados ao código **requerem conhecimentos adicionais**, P10 afirmou que "utilizar as diretrizes requer conhecimento sobre aspectos da usabilidade e também modelagem de sistemas, para quem já tem esses conhecimentos é fácil utilizar as diretrizes". P26 relata que "por ter conhecimentos prévios em usabilidade é fácil de identificar onde e o que deve ser utilizado (...)".

De forma geral, a percepção dos participantes indica que usar as diretrizes é fácil devido ao fato de serem objetivas, no entanto, é necessária experiência no uso, pois, caso contrário, o uso delas se torna difícil devido a complexidade de alguns elementos e a necessidade de conhecimentos específicos sobre a usabilidade.

9.4 Impacto positivo na facilidade de uso das diretrizes

Nesta seção estão apresentados os resultados obtidos para a análise da opinião dos participantes sobre o que impactaria positivamente na facilidade de uso das diretrizes. Na Figura 14 é apresentada a rede de visualização dos códigos e categorias obtidos pela análise.

Figura 14 – Rede de visualização sobre a percepção de facilidade de uso das diretrizes



Fonte: Elaborada pelo autor.

Os participantes relataram que **fornecer um guia de uso** acompanhado de **exemplos de aplicação das diretrizes**, além de **simplificá-las** e **torná-las mais específicas**, impacta de forma positiva na sua facilidade de uso.

Associado ao código **exemplos de aplicação das diretrizes**, o relato de P28 indicou que "a apresentação de alguns exemplos de aplicação das diretrizes podem facilitar em como e

quais situações as diretrizes podem ser aplicadas" impacta positivamente na facilidade de uso delas.

Relacionado ao código **fornecer um guia de uso**, P47 sugere que exista *"um repositório de informações sobre como usar e como resolver problemas sobre a escolha e utilização das diretrizes"*. Nesse mesmo sentido, P46 sugere que *"um modo de facilitar a utilização (das diretrizes) é possuindo uma espécie de guia de aplicação"*.

Associados ao código de **tornar as diretrizes mais específicas**, P50 relata que existe a necessidade de uma *"maior adaptabilidade das diretrizes e modelos mais resumidos"*. P40 cita que *"os meta-modelos serem mais desagregados, possibilitando sua utilização mais especificamente"* impacta de forma positiva na facilidade de uso das diretrizes.

Para o código **simplificar as diretrizes**, P23 relata que a *"representação deve ser mais resumida e simples"*. Para P27, *"(...) deve ser necessário uma revisão nas diretrizes a fim de simplificar e permitir uma melhor facilidade na inserção delas no projeto"*.

De forma geral, os participantes relataram que existe uma necessidade de revisão nas diretrizes, visando torna-las mais simples e menos genéricas, além de fornecer um guia de uso com exemplos práticos.

10 DISCUSSÃO DOS RESULTADOS

Os participantes do estudo reconheceram a importância de considerar a usabilidade durante as fases de especificação de requisitos e design da arquitetura, entretanto, apresentaram dificuldades em incorporá-los e indicaram aumento da complexidade dos diagramas. Essas dificuldades, em incorporar os mecanismos, podem estar relacionadas ao pouco conhecimento dos participantes quanto a estes aspectos, indicando a importância de apresentar para a comunidade de Engenharia de Software estudos sobre a incorporação desses aspectos da usabilidade, permitindo o aprimoramento do conhecimento desses profissionais.

Para os participantes do estudo, os principais fatores que influenciam na decisão de usar as diretrizes está relacionado com o tempo e escopo do projeto, além da experiência dos envolvidos e o perfil do público alvo. Essas características implicam na necessidade de um maior apoio na incorporação das diretrizes e de formas que aprimorem o conhecimento dos profissionais sobre este assunto.

Com a análise das soluções arquiteturais pôde-se identificar que alguns componentes das diretrizes ainda precisam ser adicionados como, por exemplo, o caso de uso "remover favorito" no meta-modelo para o mecanismo **favoritos**. Portanto, torna-se necessária uma revisão e evolução das diretrizes, visando torná-las mais completas.

Percebeu-se também que existem algumas ameaças que podem afetar a validade do estudo. Como o uso dos meta-modelos era uma atividade opcional no projeto das aplicações, alguns participantes ficaram relutantes em adotá-los devido à percepção de sua complexidade. Além disso, observou-se que algumas das equipes, embora não tenham especificado determinados requisitos funcionais de usabilidade, representaram os mecanismos na fase de design com o uso dos meta-modelos.

Quanto a experiência dos estudantes, não foi realizado um estudo detalhado para analisar o perfil deles, no entanto, os estudantes possuíam conhecimentos sobre conteúdos relacionados à programação orientada a objetos, análise e projeto de sistemas e padrões de projeto, pois é necessária a aprovação nessas disciplinas para cursar a disciplina de Arquitetura de Software. Além do que as diretrizes usadas no experimento foram propostas para serem utilizadas por não especialistas em usabilidade, caracterizando os estudantes como participantes adequados para a pesquisa.

Novos resultados poderiam ser encontrados com a aplicação de um estudo exploratório na indústria. O principal fator a ser considerado, seria a experiência dos profissionais

atuantes na área. Porém, este tipo de estudo é propenso a grandes problemas em sua execução, principalmente quanto ao comprometimento e tempo das empresas na realização das tarefas. Um problema encontrado para sua aplicação, no âmbito da cidade de Russas, foi a falta de empresas atuantes na área de TI.

Por fim, os resultados apresentados são considerados indícios de que a incorporação dos requisitos funcionais de usabilidade ainda não é facilmente integrada ao design da arquitetura de software. Nesse sentido, percebe-se que existe a necessidade de um maior direcionamento para a tomada de decisões na integração da usabilidade na arquitetura para que se desenvolvam sistemas orientados a qualidade de uso.

11 CONCLUSÕES E TRABALHOS FUTUROS

Esta monografia apresentou os resultados de um estudo exploratório que teve por objetivo investigar como os mecanismos de usabilidade podem ser incorporados no design da arquitetura de software. Os resultados indicaram tendências em especificar requisitos de usabilidade voltados a **abortar**, **alerta** e **ajuda multinível**. No entanto, os participantes relataram dificuldades em incorporar esses mecanismos no design da arquitetura e indicaram aumento na complexidade dos diagramas.

Com o estudo, também foi possível identificar que as diretrizes propostas por Carvajal *et al.* (2013) foram entendidas mas não aplicadas por todas as equipes participantes, visto que, segundo alguns deles, aumentam a complexidade percebida dos diagramas. Portanto, existem ainda algumas barreiras quanto à complexidade dessas diretrizes e o impacto que elas provocam na qualidade dos diagramas. Nesse sentido, como um trabalho futuro, pretende-se investigar novas soluções alternativas para incorporação dos mecanismos no design da arquitetura e identificar quais pontos das diretrizes podem ser aprimorados. Além disso, pretende-se aplicar o estudo na indústria, assim como, identificar FUFs e mecanismos de usabilidade ainda não reconhecidos e apresentando o seu impacto no *design* da arquitetura.

Um dos pontos que pode contribuir com a facilidade de uso das diretrizes, sugerido pelos participantes, é a elaboração de um guia que apoie a incorporação das diretrizes, abordando exemplos práticos do uso delas. Outra solução que pode ser implementada é uma ferramenta que permita incorporar de forma automática os meta-modelos nos diagramas do projeto, reduzindo o tempo gasto com sua incorporação. Uma ferramenta nesse sentido foi apresentada no trabalho de Carvajal (2012), entretanto, não foi disponibilizada para uso.

Espera-se que os resultados deste estudo possa contribuir com a melhoria da qualidade dos softwares, permitindo que os futuros sistemas possam agregar a usabilidade sem grandes custos e que resultem no desenvolvimento de sistemas com mais qualidade. Além disso, os resultados e lacunas apresentados nesse estudo irão contribuir com incentivos para novas pesquisas sobre esse assunto.

REFERÊNCIAS

- ABRAN, A.; KHELIFI, A.; SURYN, W.; SEFFAH, A. Usability meanings and interpretations in iso standards. **Software quality journal**, Springer, v. 11, n. 4, p. 325–338, 2003.
- BASS, L.; JOHN, B. E. Linking usability to software architecture patterns through general scenarios. **Journal of Systems and Software**, Elsevier, v. 66, n. 3, p. 187–197, 2003.
- BIEL, B.; GRUHN, V. Towards a method for analyzing architectural support levels of usability. In: IEEE. **Software Architecture, 2009 European Conference on Software Architecture (WICSA/ECSA) 2009**. Cambridge, 2009. p. 273–276.
- BROWN, T. Design thinking. **Harvard business review**, v. 86, p. 84–92, 141, 07 2008.
- CARVAJAL, L.; MORENO, A. M.; SANCHEZ-SEGURA, M.-I.; SEFFAH, A. Usability through software design. **IEEE Transactions on Software Engineering**, IEEE, v. 39, n. 11, p. 1582–1596, 2013.
- CARVAJAL, L. E. **Usability-Oriented Software Development Process**. Tese (Doutorado) — Informatica, 2012.
- FOLMER, E.; BOSCH, J. Architecting for usability: a survey. **Journal of systems and software**, Elsevier, v. 70, n. 1-2, p. 61–78, 2004.
- FOLMER, E.; BOSCH, J. Analyzing software architectures for usability. In: LECTURE NOTES IN COMPUTER SCIENCE. **Proceedings of the EuroMicro Conference on Software Engineering**. Porto August, 2005.
- FOLMER, E.; GURP, J. V.; BOSCH, J. A framework for capturing the relationship between usability and software architecture. **Software Process: Improvement and Practice**, Wiley Online Library, v. 8, n. 2, p. 67–87, 2003.
- GARLAN, D.; SHAW, M. An introduction to software architecture. In: **Advances in software engineering and knowledge engineering**. River Edge: World Scientific, 1993. p. 1–39.
- IEEE. Ieee recommended practice for architectural description. **IEEE**, Genebra, v. 1471, p. 102–110, 1998.
- ISO 9126. **ISO/IEC 9126-1:2000, Software engineering: Product quality - Part 1: Quality model**. Genebra, 2000.
- ISO 9241-11. **ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability**. Genebra, 1998.
- JURISTO, N.; MORENO, A.; SANCHEZ-SEGURA, M.-I. Guidelines for eliciting usability functionalities. **IEEE Transactions on Software Engineering**, IEEE, n. 11, p. 744–758, 2007.
- JURISTO, N.; MORENO, A. M.; SANCHEZ-SEGURA, M.-I. Analysing the impact of usability on software design. **Journal of Systems and Software**, Elsevier, v. 80, n. 9, p. 1506–1516, 2007.
- KAARTINEN, J.; PALVIAINEN, J.; KOSKIMIES, K. A pattern-driven process model for quality-centered software architecture design—a case study on usability-centered design. In: IEEE. **Software Engineering Conference, 2007. ASWEC 2007. 18th Australian**. Melbourne, 2007. p. 17–26.

- KRUCHTEN, P. B. The 4+ 1 view model of architecture. **IEEE software**, IEEE, v. 12, n. 6, p. 42–50, 1995.
- MARQUES, A. B.; BARBOSA, S. D. J.; CONTE, T. Evaluating the usability expressiveness of a usability-oriented interaction and navigation model. In: ACM. **Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems**. Joinville, 2017. p. 24.
- MARQUES, A. B.; CAVALCANTE, E.; RIVERO, L.; LOPES, A.; CONTE, T. Aplicando design thinking para melhorar a qualidade de um aplicativo web móvel. **XIV Simpósio Brasileiro de Qualidade de Software**, 2015.
- MARQUES, A. B. d. S. *et al.* Usinn: um modelo de interação e navegação orientado à usabilidade. Universidade Federal do Amazonas, 2017.
- MENDES, E.; VIANA, D.; VISHNUBHOTLA, S. D.; LUNDBERG, L. Realising individual and team capability in agile software development: A qualitative investigation. In: IEEE. **2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. Prague, 2018. p. 183–190.
- NIELSEN, J. **Usability engineering**. Boston: Elsevier, 1994.
- RODRÍGUEZ, F. D.; ACUÑA, S. T.; JURISTO, N. Design and programming patterns for implementing usability functionalities in web applications. **Journal of Systems and Software**, Elsevier, v. 105, p. 107–124, 2015.
- SEFFAH, A.; METZKER, E. The obstacles and myths of usability and software engineering. **Communications of the ACM**, ACM, v. 47, n. 12, p. 71–76, 2004.
- SEFFAH, A.; MOHAMED, T.; HABIEB-MAMMAR, H.; ABRAN, A. Reconciling usability and interactive system architecture using patterns. **Journal of Systems and Software**, Elsevier, v. 81, n. 11, p. 1845–1852, 2008.
- SHACKEL, B. Usability-context, framework, definition, design and evaluation. **Human factors for informatics usability**, Cambridge University Press Cambridge, p. 21–37, 1991.
- STRAUSS, A.; CORBIN, J. M. **Basics of qualitative research: Grounded theory procedures and techniques**. Thousand Oaks: Sage Publications, Inc, 1990.
- VILELA, J.; FIGUEIREDO, B.; CASTRO, J.; SOARES, M.; GONÇALVES, E. Usability and software architecture: A literature review. In: IEEE. **Components, Architectures and Reuse Software (SBCARS), 2015 IX Brazilian Symposium on**. Belo Horizonte, 2015. p. 80–89.

APÊNDICE A – QUESTIONÁRIO UTILIZADO PARA EXTRAÇÃO DE RELATOS DOS PARTICIPANTES SOBRE O USO DAS DIRETRIZES

Questão 1. Qual sua opinião sobre o uso das diretrizes de usabilidade para o projeto arquitetural do software?

Questão 2. O que influenciaria na sua decisão para adotar ou não as diretrizes de usabilidade?

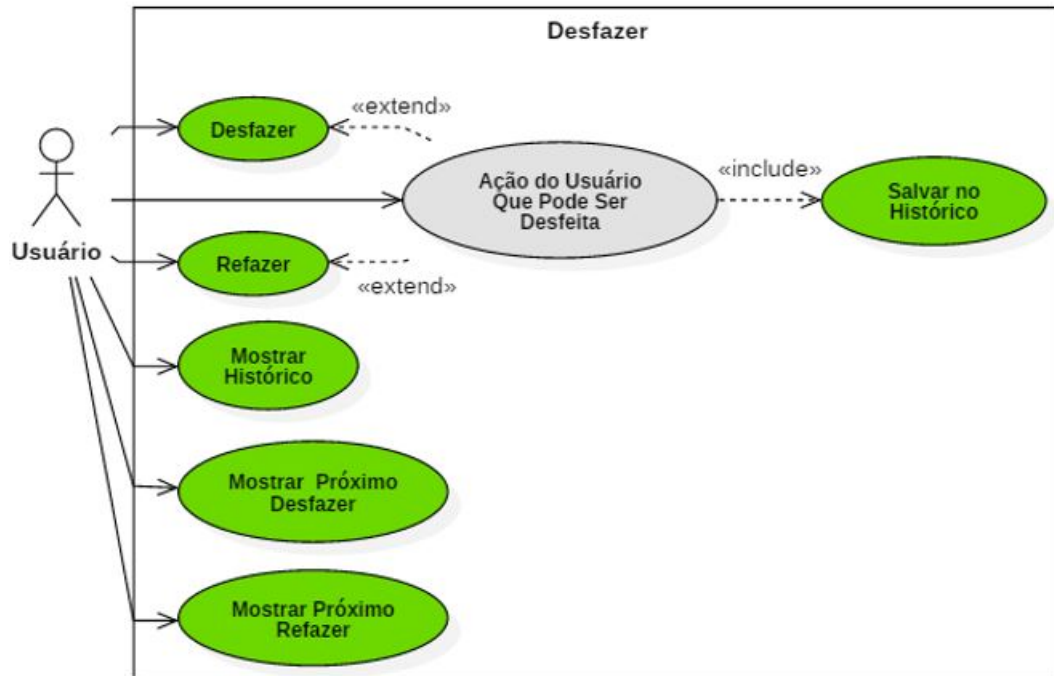
Questão 3. Qual sua percepção sobre a facilidade de utilizar no projeto arquitetural de software?

Questão 4. O que poderia impactar de maneira positiva na facilidade de uso das diretrizes?

Questão 5. Quais aspectos de usabilidade as diretrizes de usabilidade permitem representar nas visões arquiteturais?

APÊNDICE B – META-MODELOS PARA DIAGRAMA DE CASOS DE USO

Figura 15 – Meta-modelo para o mecanismo desfazer no diagrama de casos de uso



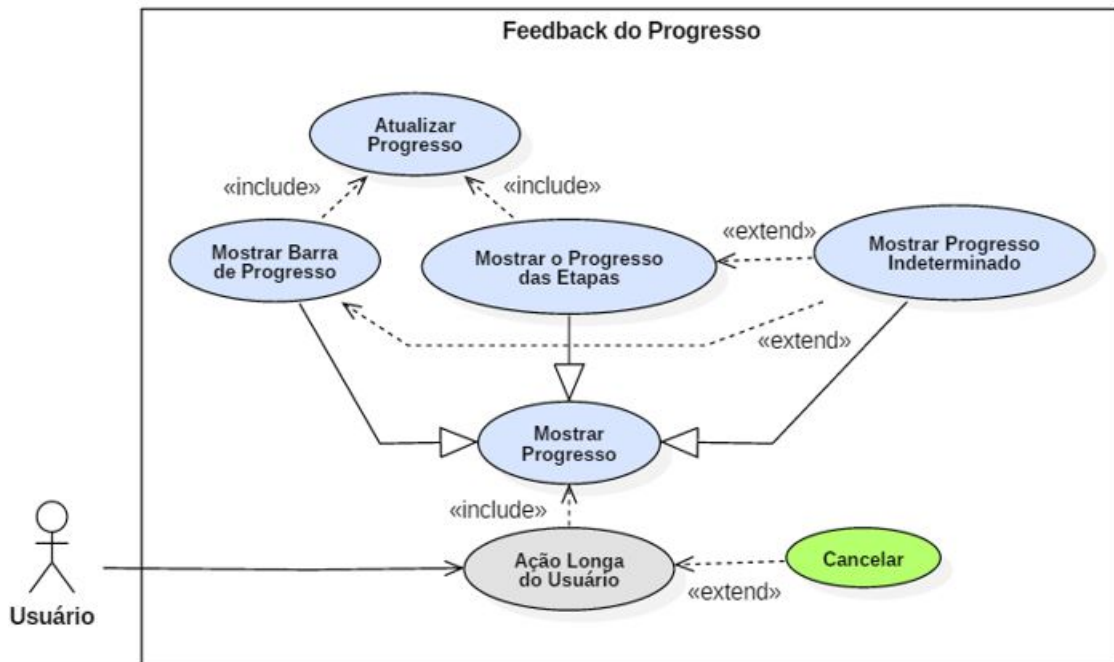
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 23 – Especificação dos casos de uso do meta-modelo para o mecanismo desfazer

Caso de uso	Descrição
Desfazer	O usuário solicita que o sistema reverta os efeitos da última ação invocada. Isso pode ser feito globalmente ou por objeto. Ao fazer isso globalmente, o usuário chama Desfazer sem especificação adicional, o que significa que a última ação que pode ser desfeita será revertida. Ao fazer isso por objeto, o Desfazer será chamado sobre o objeto, o que irá reverter os efeitos da última ação reversível que foi realizada nele.
Refazer	O usuário solicita que o sistema reverta os efeitos do último Desfazer invocado, seja global ou específico do objeto.
Mostrar Histórico	O usuário solicita que o sistema exiba todos os elementos que estão na fila do desfazer/refazer (se houver). Esta fila ou histórico (consulte Salvar no Histórico) contém a sequência de ações executadas pelo usuário que podem ser desfeitas.
Mostrar Próximo Desfazer	O usuário solicita (por meio de 'menus inteligentes', por exemplo) que o sistema informe os nomes das ações que seriam desfeitas se o Desfazer fosse executado naquele momento específico.
Mostrar Próximo Refazer	O usuário solicita (por meio de 'menus inteligentes', por exemplo) que o sistema informe os nomes das ações que seriam refeitas se o Refazer fosse executado naquele momento específico.
Ação do Usuário Que Pode Ser Desfeita	Este caso de uso é representado em cinza para ilustrar que é um "Caso de Uso Modelo" e deve ser substituído pela ação real que pode ser desfeita / refeita, se conhecido. O usuário ordena a execução de uma ação que pode ser desfeita dentro do sistema. Este caso de uso representa as muitas ações que podem ser desfeitas na qual um sistema particular pode suportar. Essas ações podem ser invocadas pelo usuário diretamente, ou pelos comandos Desfazer ou Refazer. Por exemplo, o usuário pode chamar diretamente uma ação chamada "ligar a luz", ou essa ação pode ser chamada como parte da invocação do Desfazer na qual o usuário desfaz a ação de "desligar a luz".
Salvar no Histórico	Sempre que uma Ação do Usuário Que Pode Ser Desfeita é executada, ela é salva na fila ou histórico do desfazer. Esse caso de uso é acionado por toda execução de uma ação de Desfazer, portanto, este caso de uso é incluído no caso de uso Ação do Usuário Que Pode Ser Desfeita.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 16 – Meta-modelo para o mecanismo feedback do progresso no diagrama de casos de uso



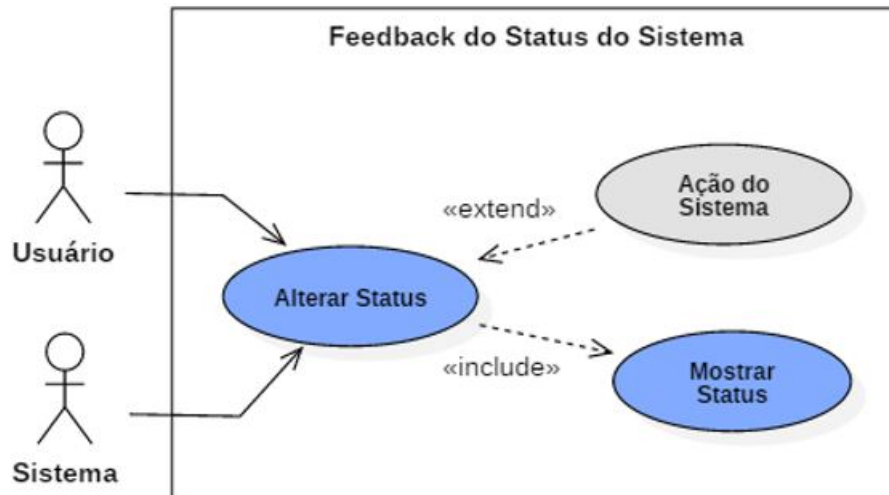
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 24 – Especificação dos casos de uso do meta-modelo para o mecanismo desfazer

Caso de uso	Descrição
Mostrar Progresso	Quando o usuário solicita uma Longa Ação do Usuário, o caso de uso Mostrar Progresso é acionado ao longo de sua execução. O caso de uso Mostrar Progresso pode ser do tipo Mostrar Barra de Progresso, Mostrar o Progresso das Etapas ou Mostrar Progresso Indeterminado, que estão descritos logo abaixo.
Mostrar Barra de Progresso	Inicialmente é exibida uma barra vazia, que é continuamente preenchida conforme a tarefa progride. Informações textuais, como a porcentagem do progresso ou o número de partes concluídas da tarefa pode acompanhar essa barra.
Mostrar o Progresso das Etapas	Enquanto a tarefa que requer esse tipo de feedback progride, são exibidas informações sobre os 'passos' dos quais ela é composta. Estas ações normalmente têm um número definido e finito de etapas, ao contrário daquelas representadas pelas barras de progresso.
Mostrar Progresso Indeterminado	Neste caso, a partir do momento em que a ação começa até o seu fim, um elemento gráfico representando o progresso indeterminado é exibido (ou seja, spinning wheel, relógio, etc.). Este caso de uso também pode ser uma extensão de Mostrar Barra de Progresso e Mostrar o Progresso das Etapas, quando a execução da tarefa é iniciada e as informações de progresso ainda não tenham sido calculadas.
Ação Longa do Usuário	Este caso de uso representa a ação do sistema que é diretamente invocada pelo usuário, que leva vários segundos para ser executada, necessitando de informações de progresso a serem exibidas para ele.
Cancelar	Em qualquer momento de sua execução, o usuário pode cancelar qualquer Ação Longa do Usuário, conforme descrito no recurso Abortar.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 17 – Meta-modelo para o mecanismo feedback do status do sistema no diagrama de casos de uso



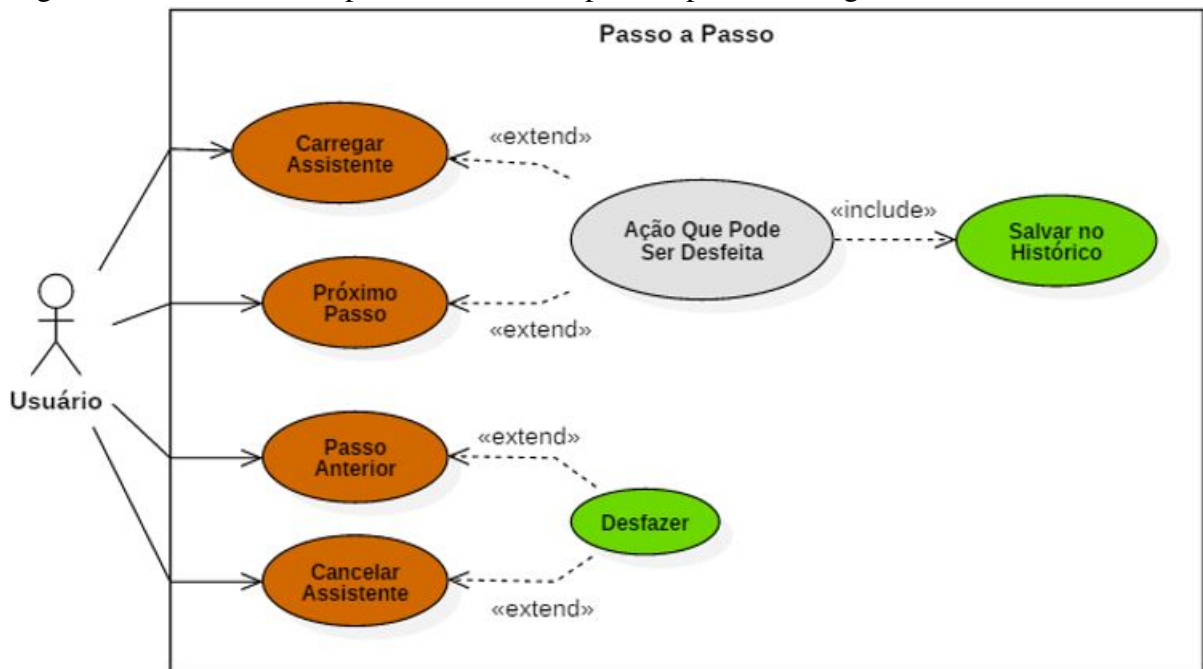
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 25 – Especificação dos casos de uso do meta-modelo para o mecanismo feedback do status do sistema

Caso de uso	Descrição
Alterar Status	Este caso de uso começa quando solicitado à alteração do valor de um determinado status do sistema (por exemplo, 'online' para 'offline'). Após a recepção dessa solicitação, o sistema altera o valor do status e normalmente o exibe (ou seja, ícone de conexão indo de verde para vermelho) e executa uma ação associada (ou seja, bloqueia o acesso à Internet). Os casos de uso Mostrar Status e Ação do Sistema representam essas duas últimas ações respectivamente e são descritas logo abaixo. Além disso, as mudanças de status podem ser acionadas pelo usuário ou pelo próprio sistema.
Mostrar Status	Sempre que um status é modificado, o sistema deve exibir visualmente a mudança para o usuário através da GUI associada.
Ação do Sistema	Como mencionado na descrição do caso de uso de Alterar Status, esta outra ação do sistema é o que deve ser executado pelo sistema uma vez que um status tenha sido alterado, além da exibição dessa alteração. Por exemplo, desligando a conexão sem fio em um notebook, não apenas altera o ícone de status como também executa uma série de chamadas para o sistema operacional bloquear o acesso ao hardware das redes Wi-Fi.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 18 – Meta-modelo para o mecanismo passo a passo no diagrama de casos de uso



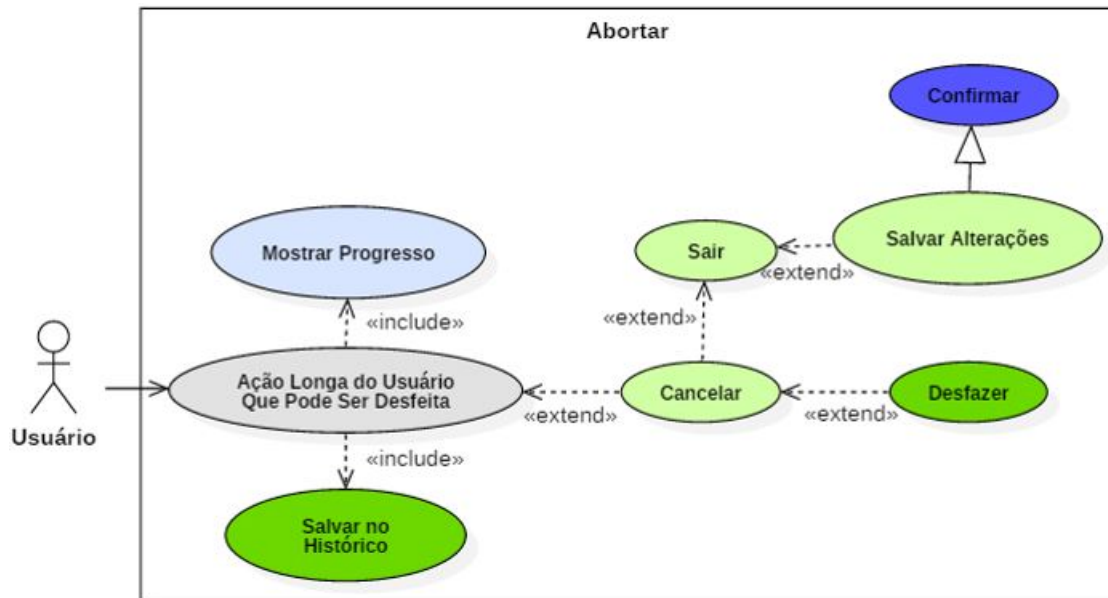
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 26 – Especificação dos casos de uso do meta-modelo para o mecanismo passo a passo

Caso de uso	Descrição
Carregar Assistente	O usuário solicita que o assistente seja carregado, na qual será apresentado o primeiro passo no assistente. Carregar um assistente pode ou não desencadear uma ação do sistema, representada por Ação Que Pode Ser Desfeita.
Próximo Passo	Ao selecionar a qualquer momento a opção 'próximo', o usuário solicita ao assistente que realize o próximo passo. O sistema apresentará ao usuário o próximo passo e com isso também poderá ser desencadeada uma ação do sistema, representada pela Ação Que Pode Ser Desfeita. O próximo passo é definido com base na estrutura da sequência (linear, árvore, etc) e qualquer entrada dada pelo usuário na etapa atual.
Passo Anterior	Ao selecionar a opção 'voltar', o usuário solicita ao assistente que realize o passo anterior. O sistema apresentará ao usuário a etapa realizada anteriormente. Para isso, pode ser necessário desfazer ações que foram executadas enquanto visitava pela primeira vez este passo (consulte Desfazer). Independentemente da estrutura da sequência que está sendo percorrido, o passo anterior é sempre o último que foi visitado antes de selecionar a opção 'voltar'.
Cancelar Assistente	A qualquer momento durante a navegação de um assistente, o usuário pode solicitar a opção de "cancelá-lo", saindo efetivamente da sequência. Assim como quando vai ao passo anterior, o cancelamento de um assistente pode incorrer em desfazer quaisquer ações que foram executados anteriormente dentro dele.
Ação Que Pode Ser Desfeita	Este caso de uso modelo representa qualquer ação do sistema que possa ocorrer quando vai de um passo de um assistente para outro. Por exemplo, quando clicando em 'próximo' em um assistente de instalação, certos componentes de software estão sendo instalados. Eles precisarão ser desinstalados se o usuário optar por Cancelar Assistente ou para percorrê-lo para trás, por exemplo, fazer escolhas diferentes, daí a necessidade de que essas ações sejam ações desfeitas e tratadas como tal.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 19 – Meta-modelo para o mecanismo abortar no diagrama de casos de uso



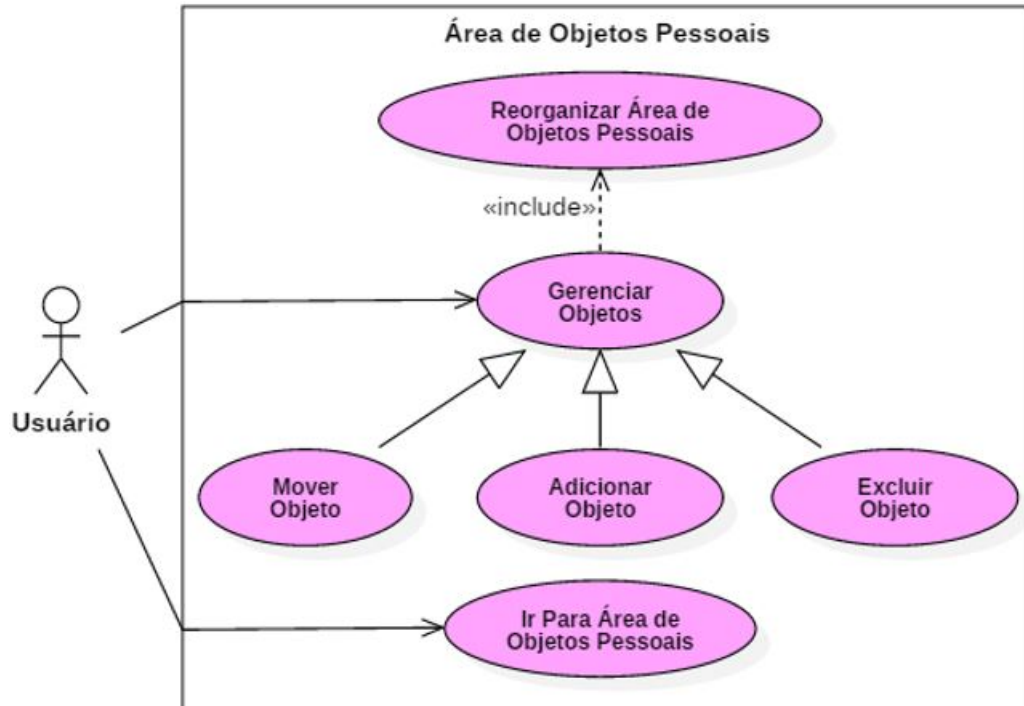
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 27 – Especificação dos casos de uso do meta-modelo para o mecanismo abortar

Caso de uso	Descrição
Cancelar	O usuário escolhe a opção para cancelar um comando em andamento (a Ação Longa do Usuário Que Pode Ser Desfeita). Cancelar uma ação significa desfazer as alterações que podem ter ocorrido (ver extensão Desfazer)
Sair	O usuário escolhe a opção para sair do aplicativo. Esta ação pode solicitar que usuário salve quaisquer alterações pendentes (ver extensão Salvar Alterações). Isso também pode desencadear o cancelamento de quaisquer ações em andamento (sejam elas comandos ou operações. Veja a extensão Cancelar).
Salvar Alterações	Ao sair do aplicativo, o usuário é solicitado a salvar quaisquer alterações pendentes.
Ação Longa do Usuário Que Pode Ser Desfeita	Qualquer ação invocada pelo usuário que é considerada 'longa' e que precisa ser mostrada de alguma forma a informação de progresso (ver Mostrar Progresso) e que possa ser desfeita, precisando ser salvo na fila ou histórico do desfazer/refazer (consulte o caso de uso incluído Salvar no Histórico).
Mostrar Progresso	Mostra o progresso de uma ação. Veja o recurso Feedback do Progresso.
Desfazer	Reverte os efeitos de uma ação executada. Veja o recurso Desfazer.
Confirmar	Solicita que o usuário prossiga (OK) ou cancela uma ação. Veja o recurso Alerta.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 20 – Meta-modelo para o mecanismo área de objetos pessoais no diagrama de casos de uso



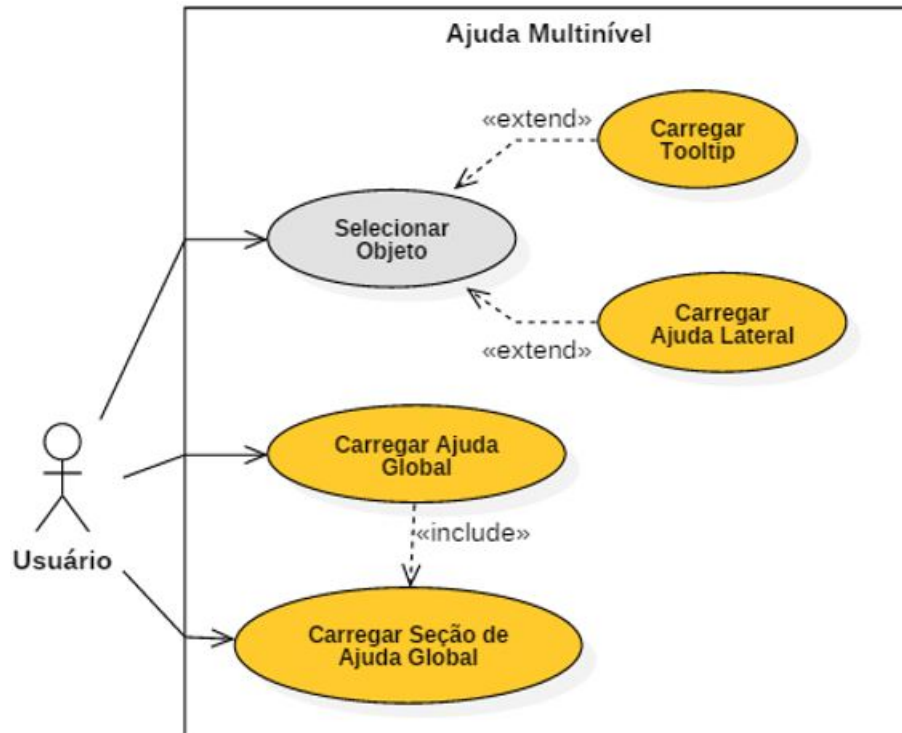
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 28 – Especificação dos casos de uso do meta-modelo para o mecanismo área de objetos pessoais

Caso de uso	Descrição
Gerenciar Objetos	Este é o caso de uso pai para mover um objeto (Mover Objeto), adicionar um objeto (Adicionar Objeto) e excluir um objeto (Excluir Objeto) do espaço de objetos pessoais. Todos esses casos de uso têm em comum o fato de que após cada uma de suas respectivas operações realizadas (movimentação, adição, exclusão), em todos os casos, o espaço em que eles foram executados deve ser rearranjado em torno da mudança, conforme será explicado abaixo.
Reorganizar Área de Objetos Pessoais	Reorganizar a área de objetos pessoais implica em mover os objetos existentes do espaço para que eles estejam em conformidade com um conjunto de regras. Rearranjo só acontece depois de um objeto no espaço ter sido "gerenciado" de alguma forma (veja o caso de uso acima), possivelmente deixando o espaço em um estado instável, precisando ser rearranjado para essa estabilidade ser atingida mais uma vez. Por exemplo, se um novo objeto for excluído de um espaço, outros objetos nesse espaço podem precisar ser reposicionados para preencher este local recém-desocupado. Cenários semelhantes são reproduzidos ao mover ou adicionar objetos.
Mover Objeto	Este caso de uso começa quando o usuário pega um objeto do espaço e move ele para uma nova posição. Esta nova posição pode estar ocupada ou vazia. Depois que o objeto é movido, os elementos restantes do espaço são reorganizados, se for necessário.
Adicionar Objeto	Este caso de uso é iniciado quando o usuário seleciona a opção para adicionar um novo objeto para o espaço. O objeto pode ser adicionado a uma posição específica do espaço (ocupado ou não) ou para a posição 'padrão' no espaço onde os novos objetos são remanejados.
Excluir Objeto	A exclusão começa quando o usuário seleciona o objeto que deseja excluir e depois escolhe a opção de eliminá-lo do espaço. O resto dos objetos no espaço pode ter que ser reorganizado em torno da exclusão (ou seja, para preencher a posição recentemente desocupada).
Ir Para Área de Objetos Pessoais	Quando múltiplos espaços são permitidos, este caso de uso descreve como o usuário pode alternar de um para outro.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 21 – Meta-modelo para o mecanismo ajuda multinível no diagrama de casos de uso



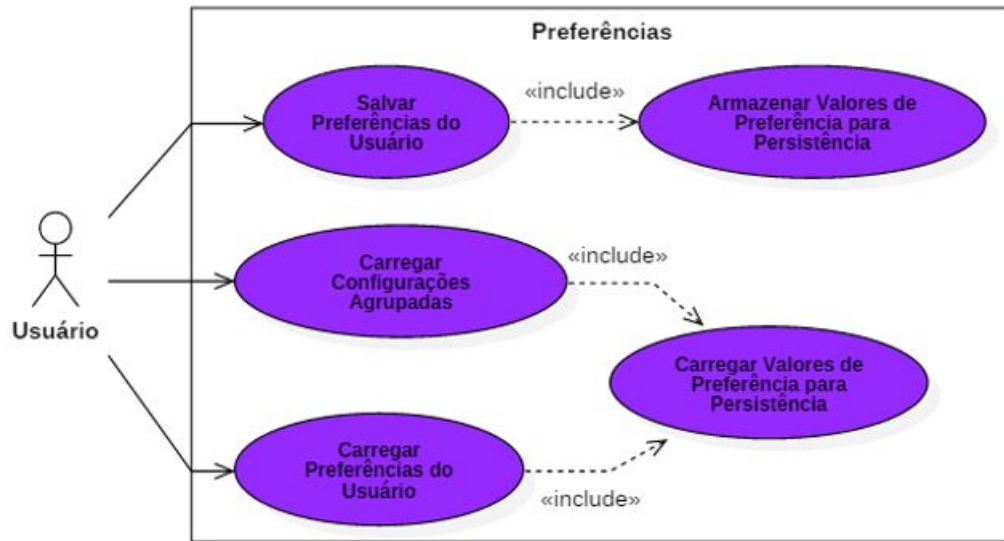
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 29 – Especificação dos casos de uso do meta-modelo para o mecanismo ajuda multinível

Caso de uso	Descrição
Selecionar Objeto	Este caso de uso modelo representa a seleção de um objeto da interface que pode mostrar um tooltip (como por exemplo, passar o mouse sobre um link).
Carregar Tooltip	Este caso de uso sempre fará parte de outro caso de uso, Selecionar Objeto nesse caso, que determinará se será mostrada uma “dica de ferramenta” (tooltip).
Carregar Ajuda Lateral	Assim como no Carregar Tooltip, esse caso de uso é invocado de dentro do Selecionar Objeto, quando o objeto selecionado requer ajuda lateral para ser carregado.
Carregar Ajuda Global	Este caso de uso começa quando o usuário solicita uma ajuda global, sem indicar nenhuma seção específica para carregar (o ‘home’ da ajuda global será exibido).
Carregar Seção de Ajuda Global	Este caso de uso pode ser iniciado por um usuário que deseja carregar uma seção específica da ajuda global. Também é chamado pelo uso do Carregar Ajuda Global continuamente, conforme os usuários se movem de uma seção para outra dentro dela.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 22 – Meta-modelo para o mecanismo preferências no diagrama de casos de uso



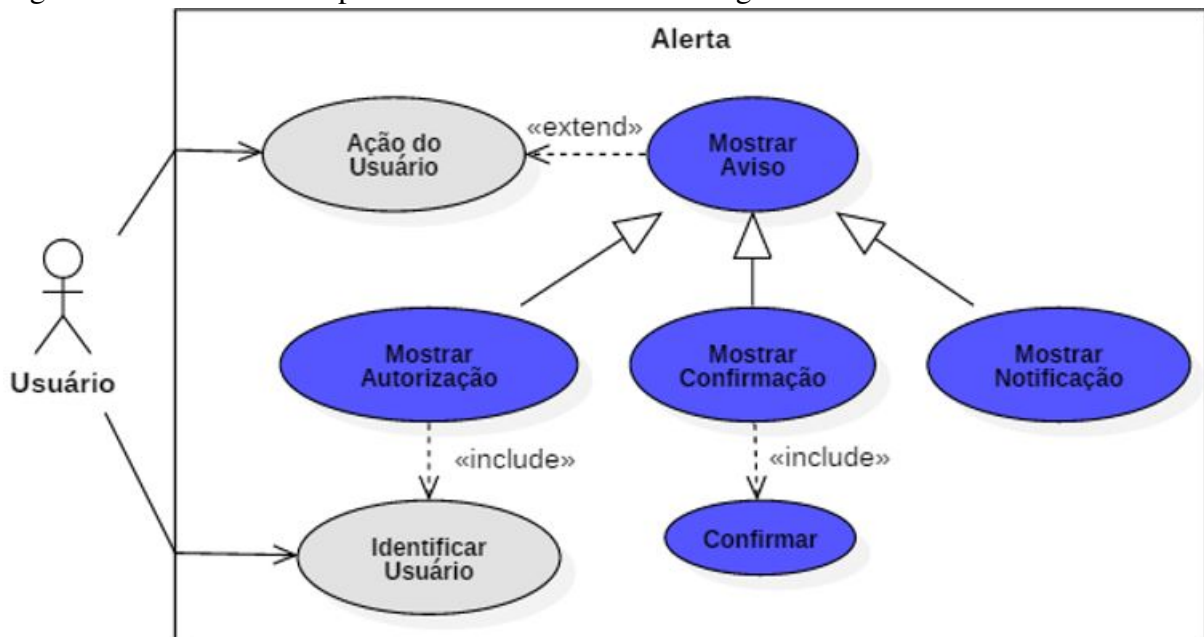
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 30 – Especificação dos casos de uso do meta-modelo para o mecanismo preferências

Caso de uso	Descrição
Salvar Preferências do Usuário	Ao fazer alterações em uma ou mais preferências, o usuário solicita que elas sejam salvas. Isso acionará o caso de uso incluído Armazenar Valores de Preferência para Persistência
Carregar Configurações Agrupadas	O usuário solicita que um grupo de configurações prontas seja carregado, permitindo-lhe definir certo número de preferências de uma só vez. Este caso de uso aciona Carregar Valores de Preferência Para Persistência, permitindo o carregamento dos valores armazenados que serão utilizados.
Carregar Preferências do Usuário	O usuário solicita que suas preferências sejam carregadas, diretamente ou indiretamente. Por exemplo, iniciar o aplicativo é uma maneira indireta de solicitar que as preferências sejam carregadas em determinados sistemas. Este caso de uso também aciona Carregar Valores de Preferência Para Persistência, permitindo que cada uma das preferências carregadas seja 'preenchida' com seu valor armazenado em persistência.
Armazenar Valores de Preferência Para Persistências	Este caso de uso é acionado por Salvar Preferências do Usuário toda vez que um usuário escolher salvar suas preferências atuais. Ele grava os valores de preferência no meio físico predeterminado.
Carregar Valores de Preferência Para Persistência	Este caso de uso é acionado por Carregar Configurações Agrupadas ao carregar uma configuração agrupada. Cada um dos valores padrão para as preferências contidas nessa configuração precisam ser carregadas para a persistência.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 23 – Meta-modelo para o mecanismo alerta no diagrama de casos de uso



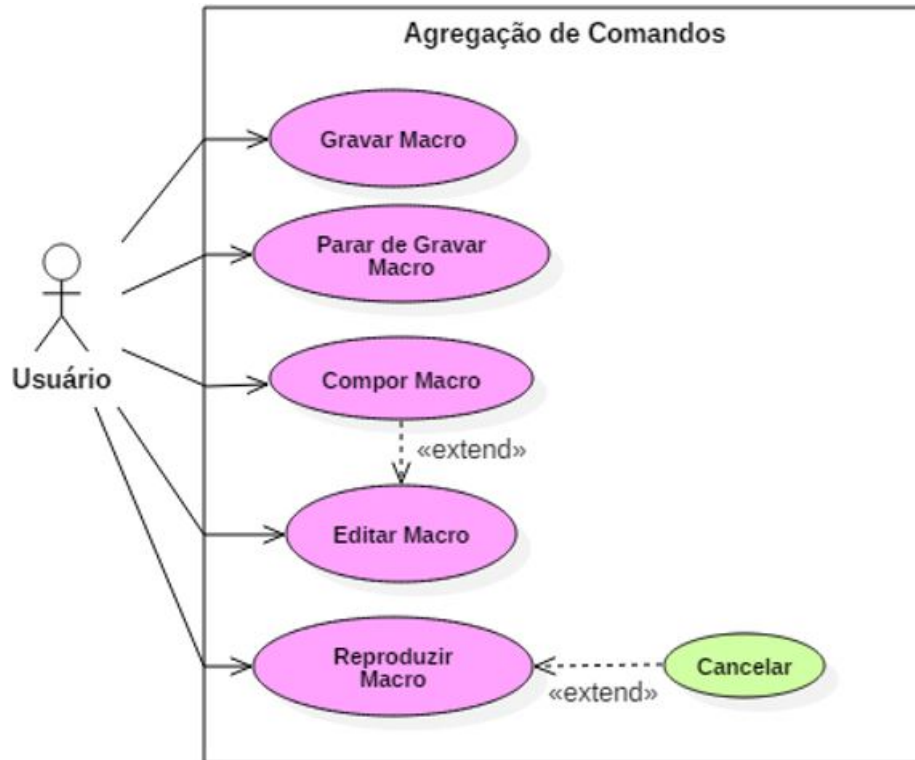
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 31 – Especificação dos casos de uso do meta-modelo para o mecanismo alerta

Caso de uso	Descrição
Ação do Usuário	Este caso de uso modelo representa qualquer ação do usuário no sistema que pode desencadear um aviso.
Mostrar Aviso	É o caso de uso pai para exibir um aviso. Esses casos de uso são extensões da Ação do Usuário, e podem ou não desencadear dentro dele se assim for determinado durante a elicitação. Por exemplo, se um recurso "exportar vídeo" em um aplicativo de vídeo fornecesse aviso ao exportar vídeos com tamanho superior a 100 MB, o uso da "exportação de vídeo" acionaria o aviso apropriado somente se essa condição fosse atendida e não de outra forma.
Mostrar Autorização	Uma autorização é o tipo de aviso que requer do usuário não apenas o 'ok' em uma ação, será necessário também introduzir alguma forma de validação dos dados para prosseguir (como, por exemplo, uma combinação de login e senha).
Mostrar Confirmação	Uma confirmação é o tipo de aviso em que o usuário deve apenas confirmar uma ação antes da execução (por exemplo, 'Tem certeza de que deseja esvaziar a lixeira? Sim/ Não').
Mostrar Notificação	Uma notificação é simplesmente uma mensagem transmitida ao usuário sobre uma ação que já ocorreu. O usuário não tem meios de interromper a ação e só está sendo informado que a ação foi executada.
Identificar Usuário	Este caso de uso modelo representa o caso de uso específico do domínio que o usuário introduz suas credenciais para serem identificadas pelo sistema. Como isso é inerente a cada sistema em particular, o caso de uso real deve ser substituído por esse caso de uso modelo quando o modelo real for projetado. Este caso de uso está incluído dentro do aviso de autorização.
Confirmar	Confirmar é um caso de uso incluído no aviso correspondente, que representa uma confirmação do usuário, que pode ser simplesmente um clique em um botão 'ok' ou outra opção, deixando o meta-modelo aberto a alternativas mais elaboradas.

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 24 – Meta-modelo para o mecanismo agregação de comandos no diagrama de casos de uso



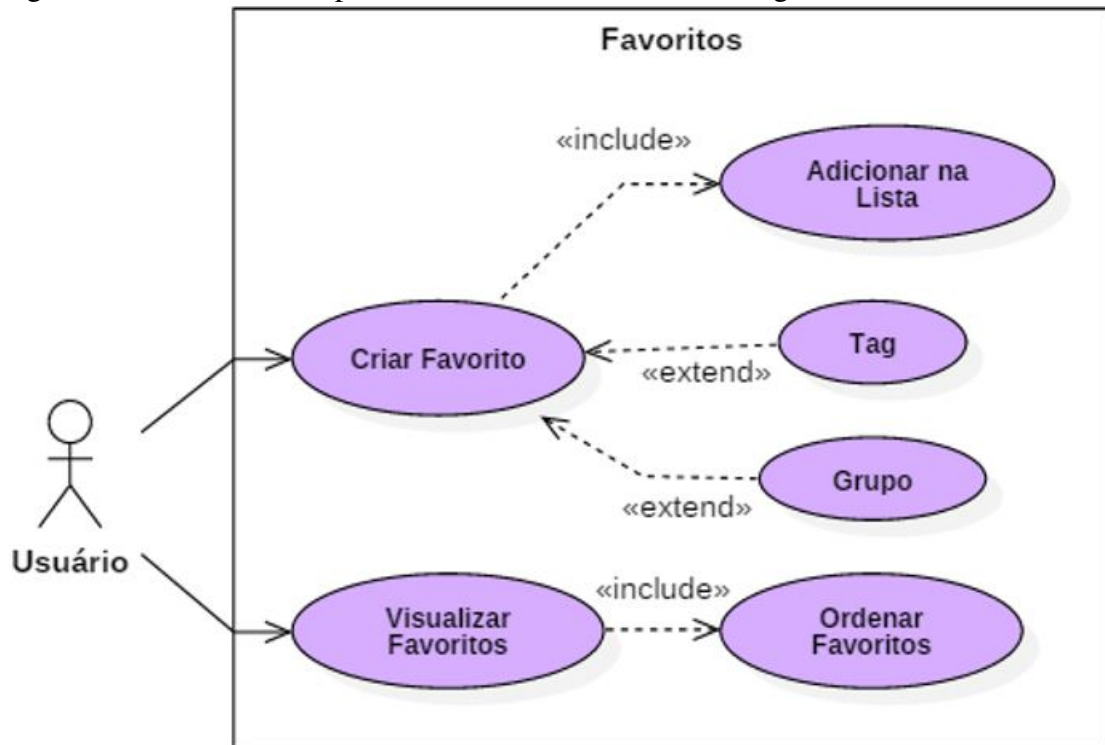
Fonte: Carvajal (2012), adaptada pelo autor.

Tabela 32 – Especificação dos casos de uso do meta-modelo para o mecanismo agregação de comandos

Caso de uso	Descrição
Gravar Macro	O usuário seleciona a opção para iniciar a gravação de uma macro. Imediatamente depois de fazer isso, o sistema registra cada ação (registro adequado) realizada pelo usuário até que a gravação da macro seja interrompida.
Parar de Gravar Macro	O usuário seleciona a opção para interromper a gravação de uma macro. O sistema interrompe a gravação das ações, salva as ações gravadas e solicita ao usuário que nomeie a macro recém-criada.
Compor Macro	O usuário seleciona a opção para compor duas ou mais macros, após a conclusão, salva os resultados em uma nova macro.
Editar Macro	O usuário seleciona a opção para editar uma macro. Ao fazer isso, ele pode compor com outras macros existentes. Após a conclusão da edição, o usuário salva os resultados na macro original.
Reproduzir Macro	O usuário seleciona uma macro existente e ordena que ela seja reproduzida. A macro executa todas as ações salvas nela (cada uma representada por um caso de uso diferente da Ação do Usuário. Ver abaixo). Quando a execução estiver concluída, o sistema informa ao usuário que a macro terminou a reprodução.
Ação do Usuário	Este caso de uso modelo representa as ações que compõem qualquer sistema macro

Fonte: Carvajal (2012), adaptada pelo autor.

Figura 25 – Meta-modelo para o mecanismo favoritos no diagrama de casos de uso



Fonte: Carvajal (2012), adaptada pelo autor.

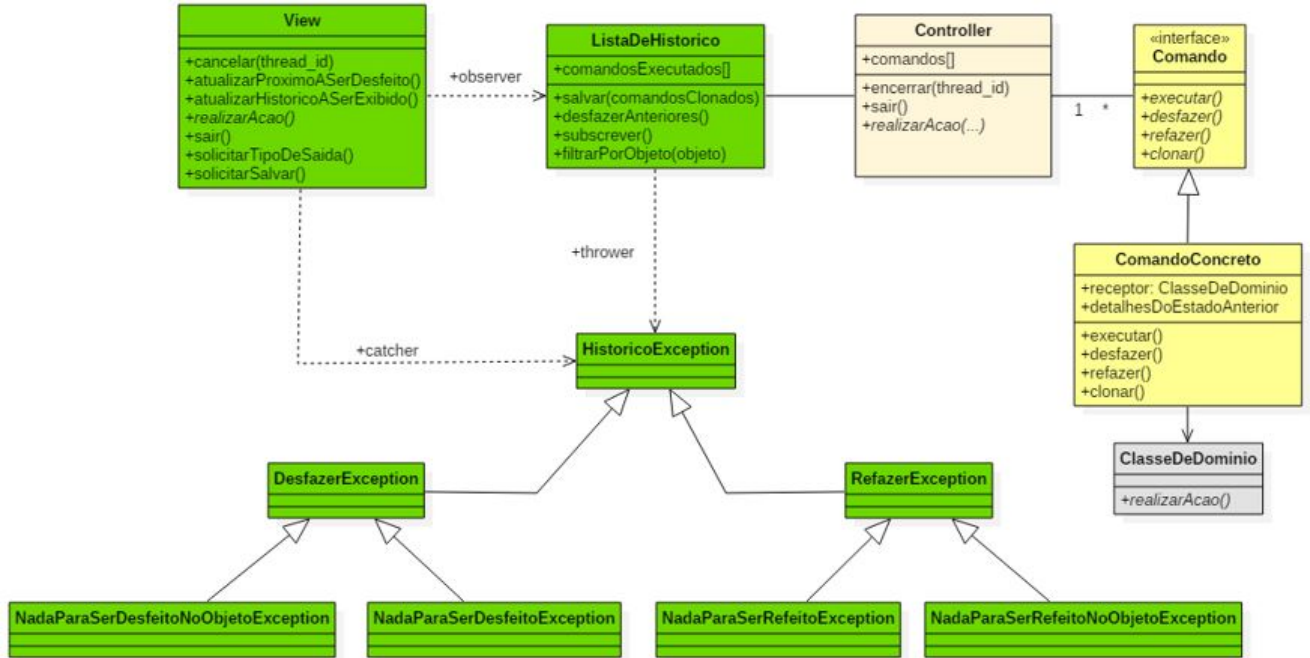
Tabela 33 – Especificação dos casos de uso do meta-modelo para o mecanismo favoritos

Caso de uso	Descrição
Criar Favorito	Este caso de uso começa quando o usuário solicita marcar um objeto como favorito. Isto implica em adicioná-lo na lista de favoritos (Adicionar na Lista) e pode implicar na ação de adicioná-lo em um Grupo ou em marca-lo com uma Tag.
Adicionar na Lista	O caso de uso Criar Favorito aciona este caso de uso. Ele implica em adicionar um favorito, criado recentemente, para a lista de favoritos.
Tag	Uma tag ou mais tags estão associadas ao marcador recém-criado. Muitas tags podem ser associadas a um ou mais marcadores.
Grupo	O marcador recém-criado é adicionado a um grupo existente. Um marcador só pode ser adicionado a um único grupo de cada vez.
Visualizar Favoritos	O usuário solicita ver uma lista completa de todos os favoritos salvos. Uma vez que a lista é exibida, o Visualizar Favoritos deve primeiro pedir que os favoritos sejam classificados, conforme determinado no tempo de elicitação (se aplicável).
Ordenar Favoritos	Este caso de uso é invocado a partir do Visualizar Favoritos e compreende na ação de classificar/ordenar os marcadores conforme é solicitado, para que posteriormente possam ser exibidos.

Fonte: Carvajal (2012), adaptada pelo autor.

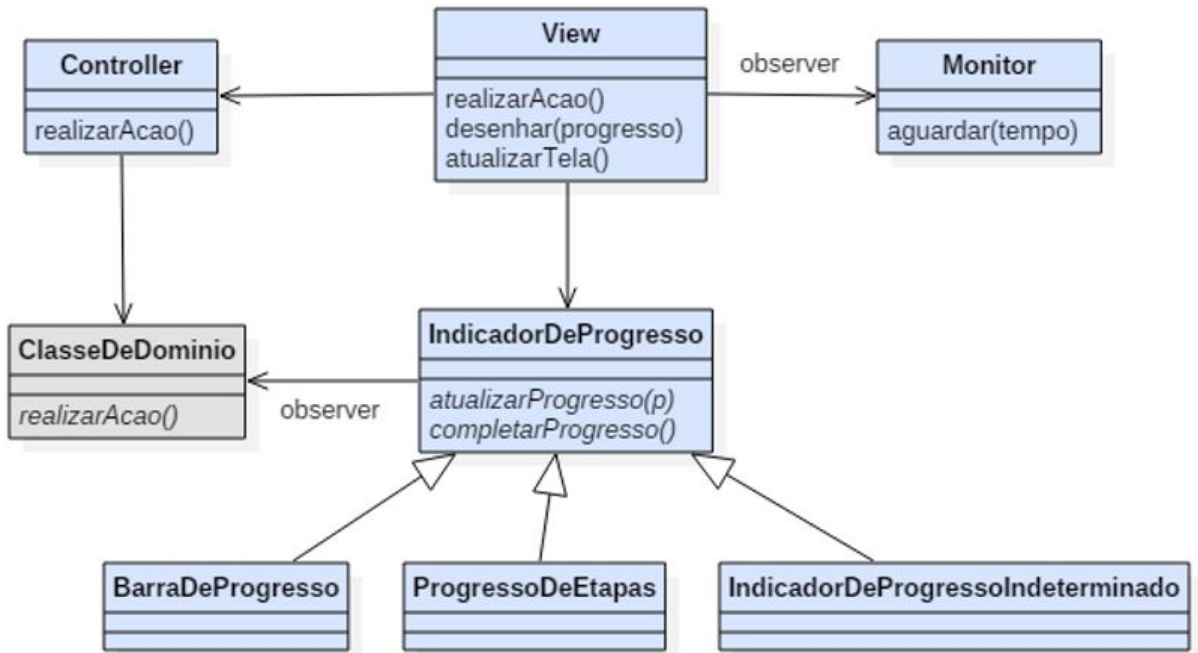
APÊNDICE C – META-MODELOS PARA DIAGRAMA DE CLASSES

Figura 26 – Meta-modelo para o mecanismo desfazer no diagrama de classes



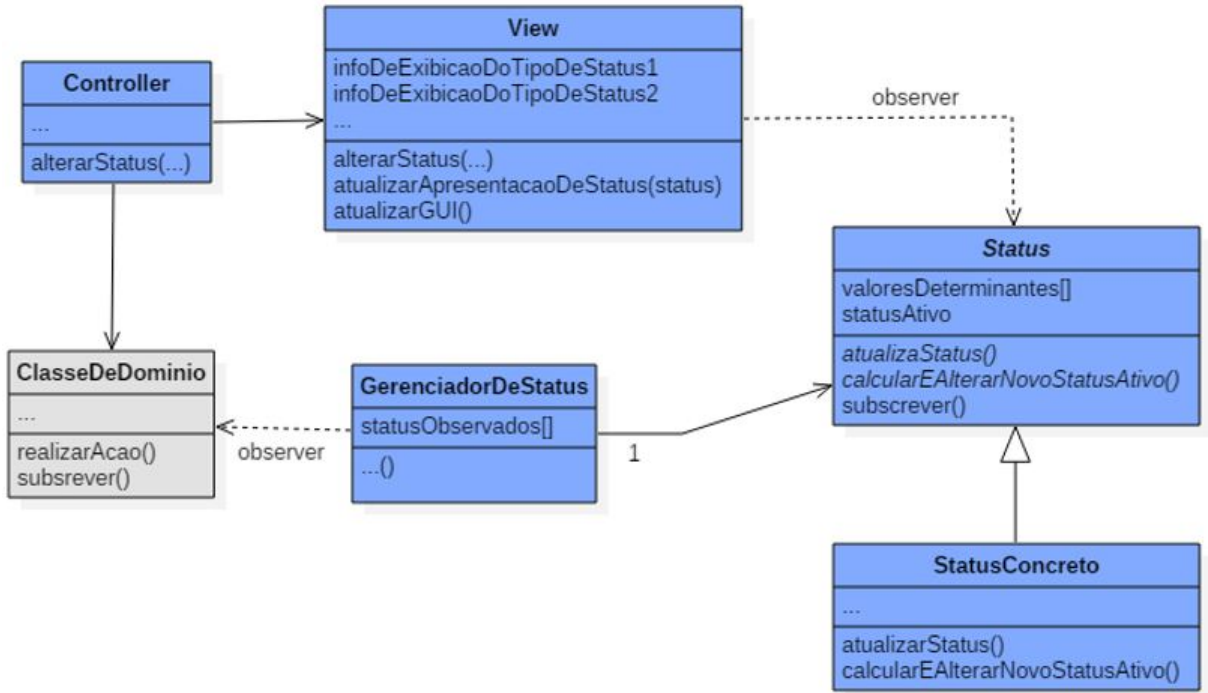
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 27 – Meta-modelo para o mecanismo feedback do progresso no diagrama de classes



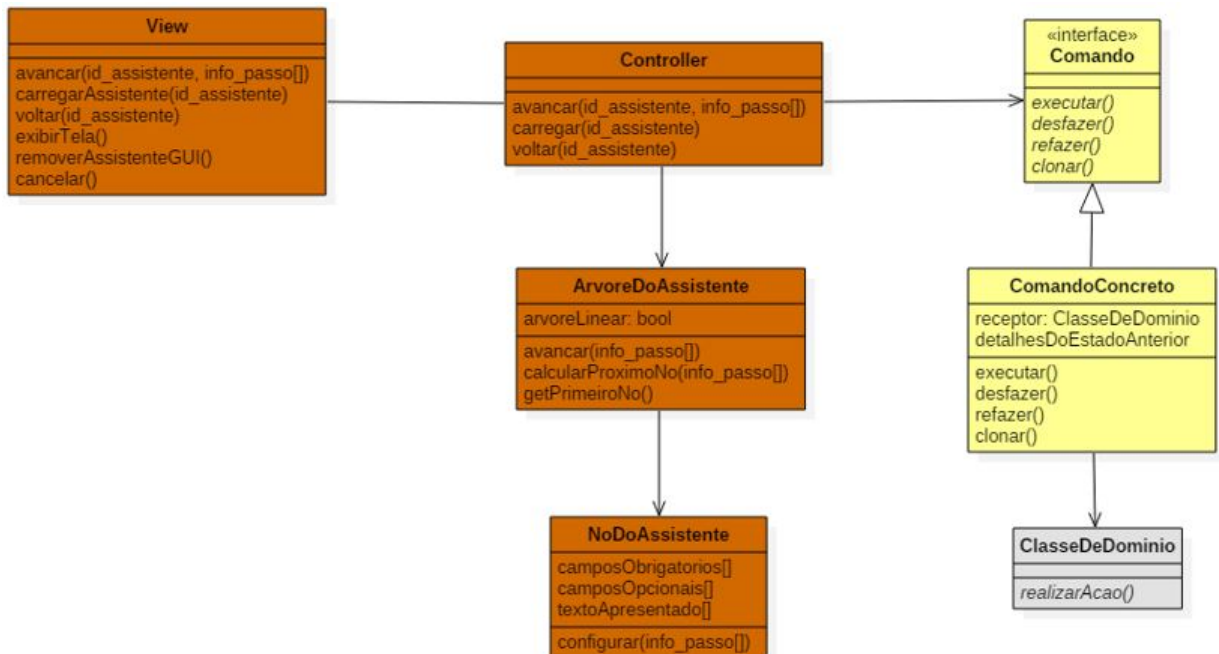
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 28 – Meta-modelo para o mecanismo feedback do status do sistema no diagrama de classes



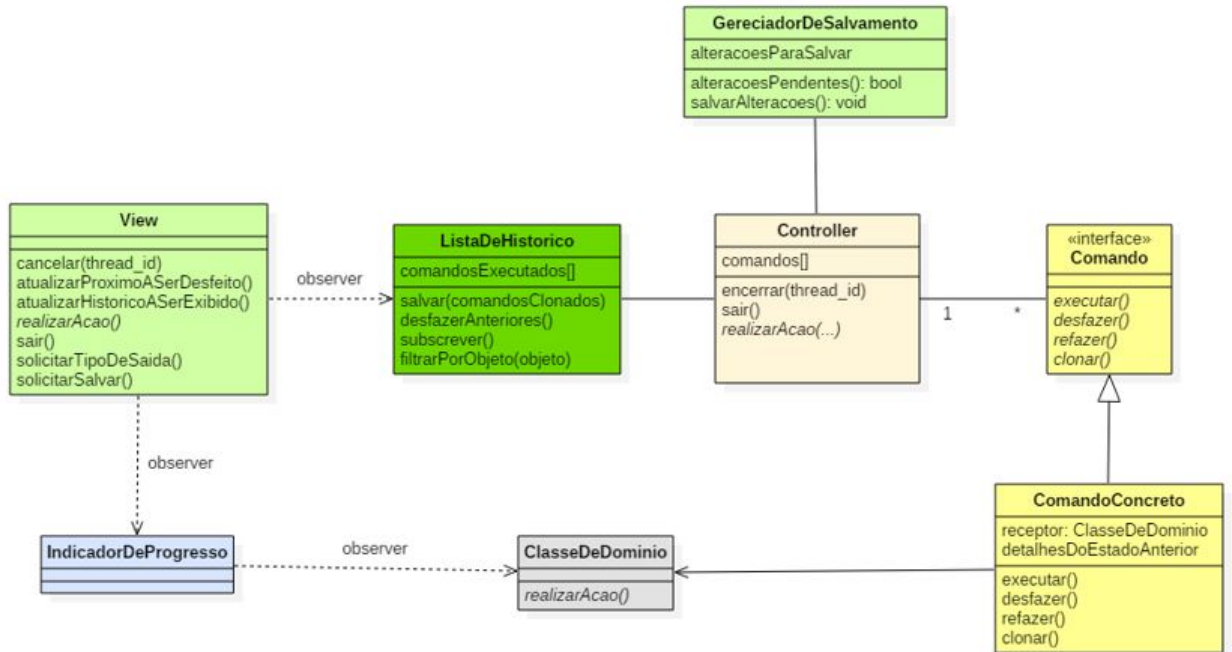
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 29 – Meta-modelo para o mecanismo passo a passo no diagrama de classes



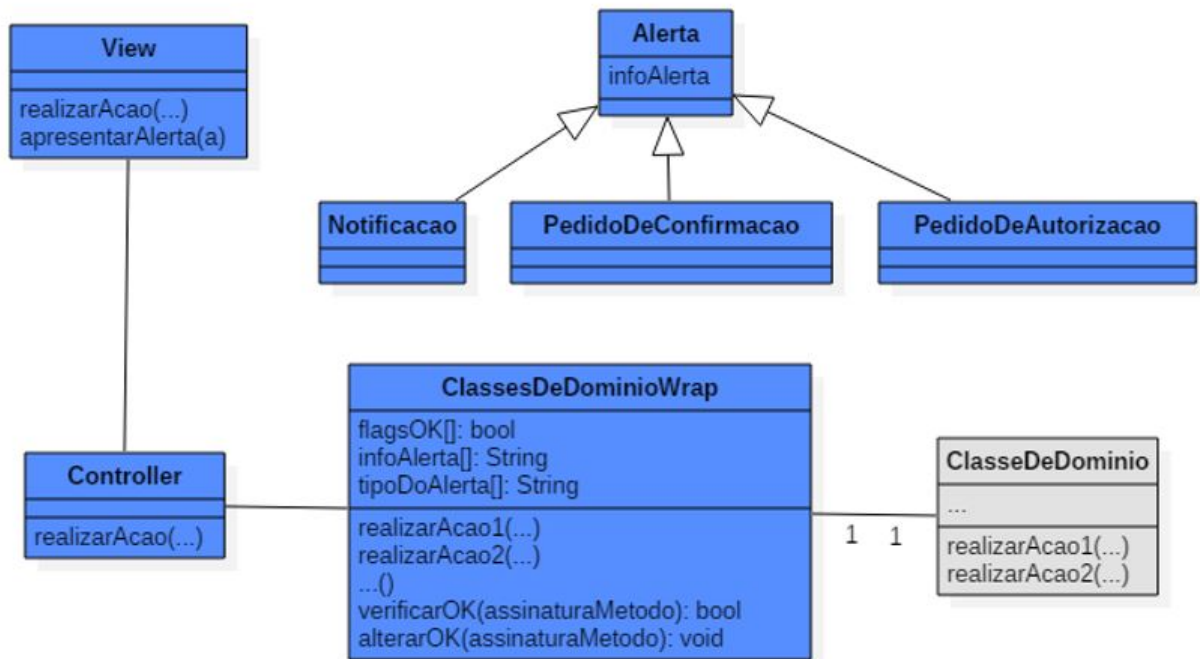
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 30 – Meta-modelo para o mecanismo abortar no diagrama de classes



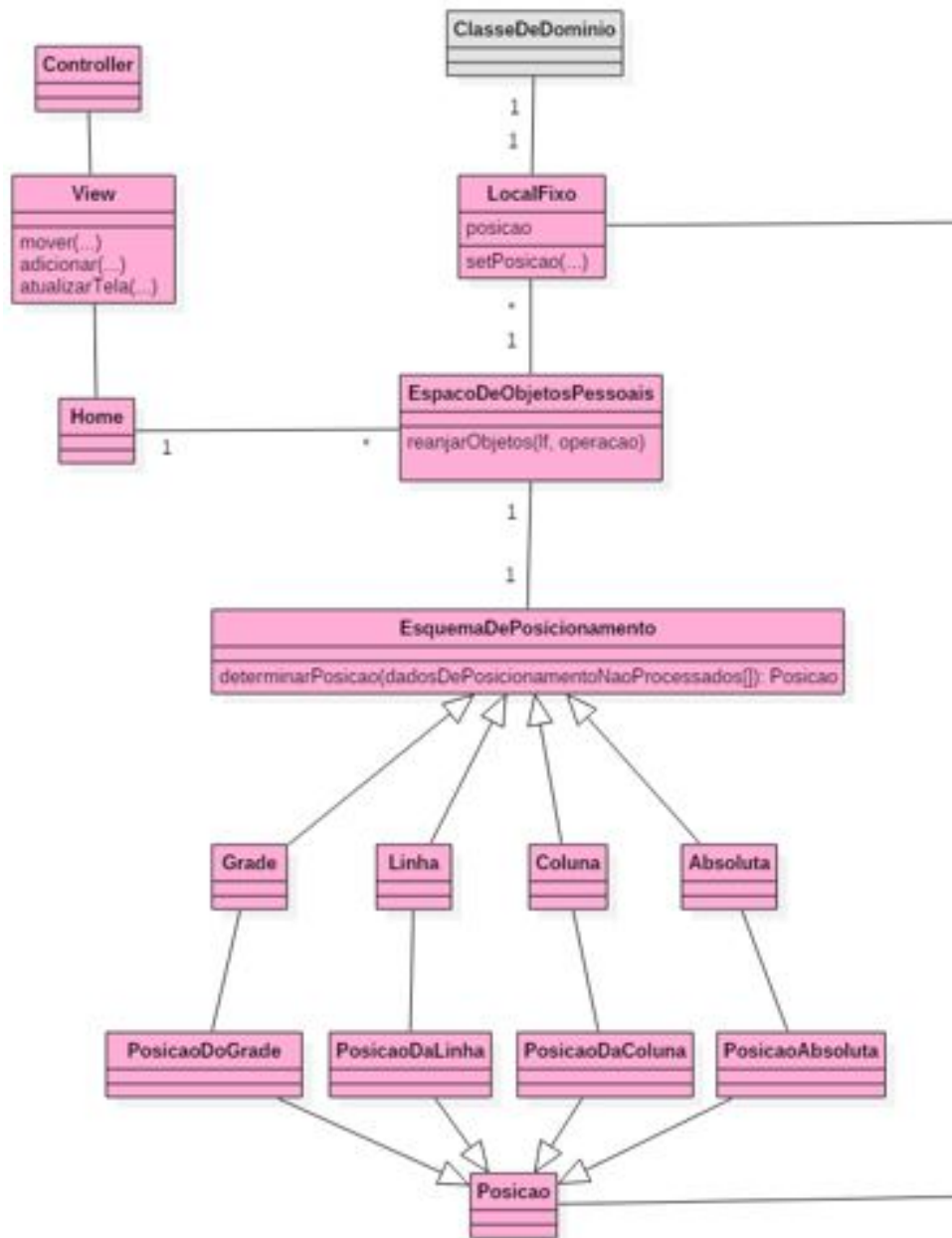
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 31 – Meta-modelo para o mecanismo alerta no diagrama de classes



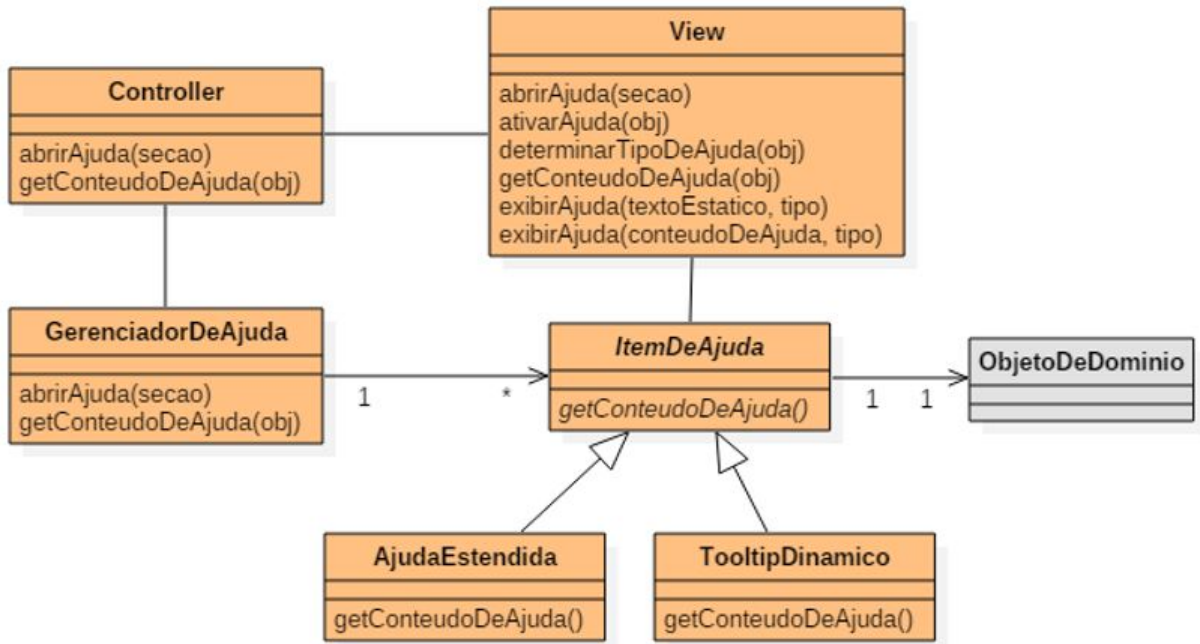
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 32 – Meta-modelo para o mecanismo área de objetos pessoais no diagrama de classes



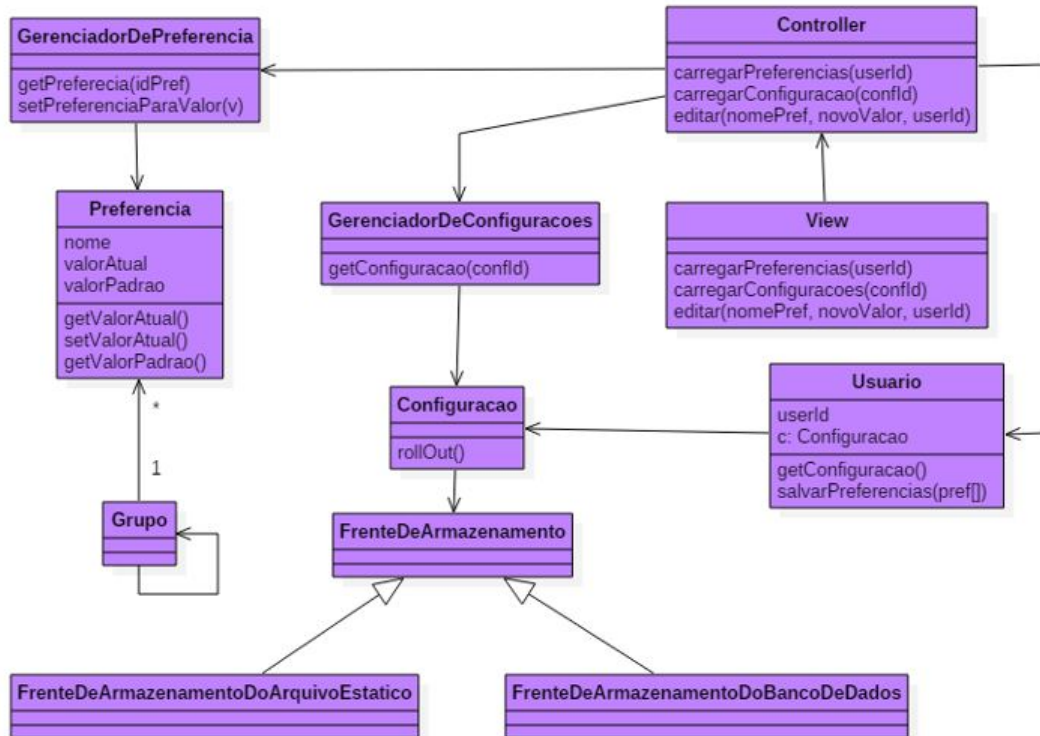
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 33 – Meta-modelo para o mecanismo ajuda multinível no diagrama de classes



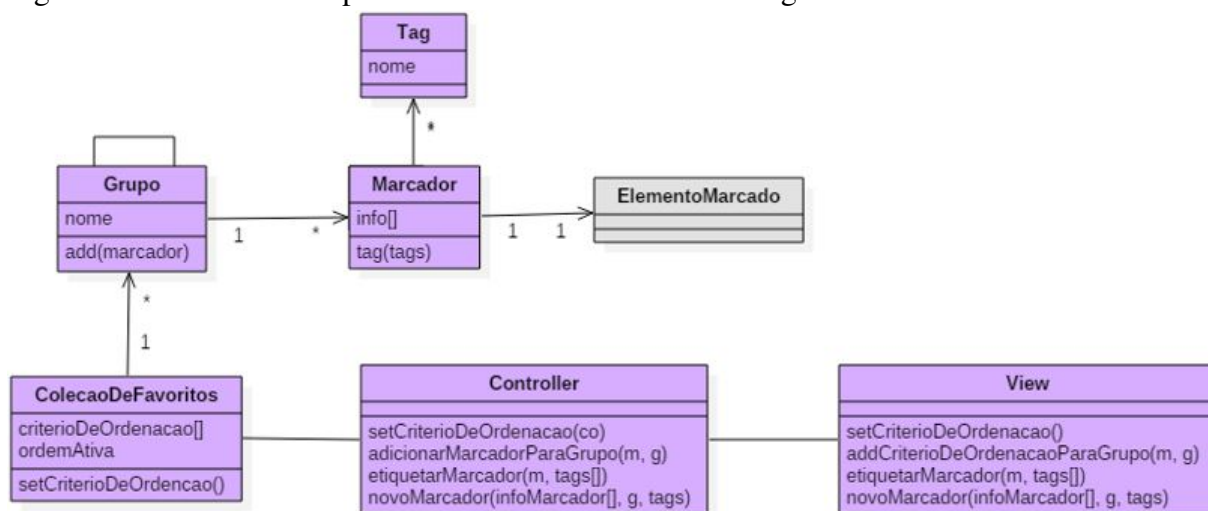
Fonte: Carvajal (2012), adaptada pelo autor.

Figura 34 – Meta-modelo para o mecanismo preferências no diagrama de classes



Fonte: Carvajal (2012), adaptada pelo autor.

Figura 35 – Meta-modelo para o mecanismo favoritos no diagrama de classes



Fonte: Carvajal (2012), adaptada pelo autor.